

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
RAČUNARSTVO I INFORMATIKA

Razvoj Windows 10 univerzalnih aplikacija
Završni rad

Mentor:

Red.prof. dr Dženana Đonko, dipl. el. ing.

Kandidat:

Ramiza Hošo, 16441

Sarajevo, 2017. godine

Postavka rada

Tema: Razvoj Windows 10 univerzalne aplikacije

Kurs: Objektno orijentirana analiza i dizajn

Cilj: Implementacija Windows 10 univerzalne aplikacije

Opis: Rad će obuhvatiti predstavljanje Windows 10 univerzalne aplikacije. Nakon toga slijedi razvoj Windows 10 univerzalne aplikacije u C# programskom jeziku. Na kraju će ta aplikacija poslužiti za prikaz mogućnosti koje se nude ovim operativnim sistemom.

Rad će se sastojati iz sljedećih poglavlja:

1. Uvod - U okviru ovog poglavlja predstavljena je tema rada, metodologija koja se primjenjivala kao i ciljevi rada.
2. Specifikacija aplikacije - Specificiranje aplikacije koja će se razviti u radu, a koja će pokazivati mogućnosti Windows 10 univerzalnih aplikacija i C# programskog jezika i kreiranje aplikacije u Visual Studio 2015 okruženju.
3. Arhitektura sistema - Predstavljanje MVVM arhitekturnog paterna i definicija arhitekture BeHealthy aplikacije.
4. Vizualizacija podataka - Opisivanje vizualizacije podataka i njena primjena na sistemu BeHealthy.
5. Zaključak - Sumiranje urađenog i smjernice za budući rad.

Polazna literatura:

- Dženana Đonko, Samir Omanović 2009. *Objektno orijentirana analiza i dizajn primjenom UML notacije*. 1.Izdanje. Sarajevo: Elektrotehnički fakultet u Sarajevu.
- Raffaele Garofalo, 2011. *Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern*, 1st Edition, Published with the authorization of Microsoft Corporation by O'Reilly Media, Inc., California

Mentor:

Prof. dr. Dženana Đonko

Univerzitet u Sarajevu

Naziv fakulteta/akademije: Elektrotehnički fakultet

Naziv odsjeka i/ili katedre: Računarstvo i informatika

Predmet: Objektno orijentisana analiza i dizajn

Izjava o autentičnosti radova

Seminarski rad, završni (diplomski odnosno magistarski) rad za I i II ciklus studija i integrirani studijski program I i II ciklusa studija, magistarski znanstveni rad i doktorska disertacija¹

Ime i prezime: Hošo Ramiza

Naslov rada: Razvoj Windows 10 univerzalne aplikacije

Vrsta rada: Završni rad I ciklusa studija

Broj stranica: 41

Potvrđujem:

- da sam pročitao/la dokumente koji se odnose na plagijarizam, kako je to definirano

Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;

- da sam svjestan univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio prisustvo citiranog ili parafraziranog materijala i da sam se referirao na sve izvore;
- da sam dosljedno naveo korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio svaku pomoć koju sam dobio pored pomoći mentorice i akademskih tutora.

Mjesto, datum _____

Potpis _____

¹ U radu su korišteni sljedeći dokumenti: *Izjava autora* koju koristi Elektrotehnički fakultet u Sarajevu; *Izjava o autentičnosti završnog rada* Centra za interdisciplinarnu studiju – master studij „Evropske studije“, *Izjava o plagijarizmu* koju koristi Fakultet političkih nauka u Sarajevu. *plagijarizmu* koju koristi Fakultet političkih nauka u Sarajevu.

Sadržaj

1.Uvod	6
1.1. Ciljevi rada	6
1.2. Metodologija rada	6
1.3. Struktura rada	7
1.4. Napomena	7
2. Specifikacija aplikacije “BeHealthy”	8
2.1. Namjena aplikacije	8
2.2. Funkcionalni zahtjevi	8
2.3. Prototip aplikacije BeHealthy	9
2.4.1. Kreiranje korisničkog računa/ Registracija	10
2.4.2. Prijavljivanje	10
2.4.3. Unošenje podataka	11
2.4.4. Upisivanje unesenih namirnica	11
2.4.5 Upisivanje obavljenih aktivnosti	12
2.4.6.Pregled stanja	12
2.5. Nefunkcionalni zahtjevi	13
2.6. Tehnologije za razvoj	13
2.7. Kreiranje aplikacije BeHealthy	13
2.8. Zaključak	14
3. Arhitektura sistema BeHealthy	15
3.1. Analiza MVVM (Model-View-ViewModel) arhitekturalnog paterna	15
3.2. Model sistema BeHealthy	17
3.4. Pogled modela sistema BeHealthy	21
3.5.Zaključak	23
4. Vizualizacija podataka sistema BeHealthy	24
4.1. Vizualizacija podataka	24
4.2. Uobičajni vizuelni prikazi	24
4.3. Kreiranje grafikona sistema BeHealthy	25
4.4. Zaključak	28
5. Zaključak	29
6. Reference	30
7. Popis slika	32
7.1. Popis slika ispod kodova	32
8. Popis tabela	34
	4

Sažetak

U okviru ovog rada je predstavljena i implementirana Windows 10 univerzalna aplikacija. Cilj ovog rada jeste detaljno analiziranje mogućnosti koje nude univerzalne aplikacije i njihovo kreiranje. Predstavljena je specifikacija, funkcionalnost i namjena aplikacije BeHealthy koja je implementirana u Visual Studio 2015 okruženju.

Prikazane su funkcionalnosti i dijelovi koda na odgovarajućim primjerima. Sama aplikacija je implementirana u svrhu ovog rada i moguća su poboljšanja i nadogradnja.

Abstract

Within this project, a Windows 10 Universal application is introduced and implemented. The aim of this project is to detail analyze in detail the possibilities that Universal applications offer, as well as their creating. The specification, functionality, and purpose of the BeHealthy application, that was implemented in the Visual Studio 2015 environment, is introduced.

The functionalities and parts of the code were shown on appropriate examples. The application itself is implemented for the purpose of this project, and there is possibility for improvements and upgrades.

1.Uvod

Kako operativni sistem Windows 8 i nije baš najbolje prihvaćen od strane korisnika, ubrzo je svijetlo dana ugledao Windows 10² operativni sistem koji je sa sobom donio mnoge novosti i time napravio jedan veoma važan korak. Naime, besplatna nadogradnja za sve korisnike Windows 7 i Windows 8 operativnog sistema u toku prve godine od datuma izdanja, osigurala je da veliki broj korisnika pređe na novi operativni sistem. S druge strane, Windows 10 omogućava jednostavniju sinhronizaciju zbog potpune kompatibilnosti sa svim uređajima koji ga koriste.

Međutim, jedna od najbitnijih novosti koje su došle sa Windows 8, a usavršene sa Windows 10 operativnim sistemom jesu univerzalne³ aplikacije kojima se sve više pridaje pažnja. Korisniku daju iskustvo korištenja jedne te iste aplikacije na različitim uređajima, različitih dimenzija i rezolucija, a samim softver inženjerima olakšavaju posao.

U ovom radu analizirana je detaljno struktura, dizajn i način rada Windows 10 univerzalne aplikacije, te da bi se što više približili njihovom radu razvijena je i prikazana aplikacija pod nazivom “BeHealthy”.

1.1. Ciljevi rada

Osnovni cilj rada jeste predstavljanje Windows 10 univerzalnih aplikacija, zatim implementacija aplikacije koristeći C# programski jezik.

Prije same implementacije aplikacije potrebno se upoznati sa mogućnostima koje nudi C# programski jezik i Windows 10 okruženje. Nakon toga, na primjeru razvoja aplikacije će se pokazati i sam rad aplikacije u navedenom okruženju.

1.2. Metodologija rada

Da bi se realizirali navedeni ciljevi potrebno je prije svega poznavanje rada u Windows 10 okruženju, a zatim proučiti odgovarajuću literaturu za rad u C# programskom jeziku. Nakon što se usvoje potrebna znanja pristupa se razvoju željene aplikacije. U toku ovog rada biće prikazan programski kod aplikacije kako bi se pokazali koraci u njenom razvoju.

1.3. Struktura rada

1. Uvod - U okviru ovog poglavlja predstavljena je tema rada, metodologija koja se primjenjivala kao i ciljevi rada.
2. Specifikacija aplikacije - Specificiranje aplikacije koja će se razviti u radu, a koja će pokazivati mogućnosti Windows 10 univerzalnih aplikacija i C# programskog jezika.
3. Arhitektura aplikacije - Opisivanje MVVM (Model - View - ViewModel) arhitekturnog paterna i njegove primjene na sistemu BeHealthy.
4. Vizualizacija podataka aplikacije - Opisivanje vizualizacije podataka i njena primjena na sistemu BeHealthy.
5. Implementacija aplikacije - Razvoj aplikacije i prikazivanje njenih mogućnosti.
6. Zaključak - Sumiranje urađenog i smjernice za budući rad.

1.4. Napomena

Pri izradi ovog rada korišten je font Times New Roman, dok je veličina fonta 12 pt. Za naslove poglavlja korišten je font veličine 14 pt i bold stil. Za referenciranje korištenih izvora prilikom pisanja rada korišten je font Arial veličine 12pt. Engleske riječi koje nisu prevedene pisane su italic stilom. Za imena klasa korišten je font Alegreya.

2. Specifikacija aplikacije “BeHealthy”

U okviru ovog poglavlja se prikazuje specifikacija aplikacije i njena namjena. Prikazan je prototip aplikacije, te su navedeni i opisani funkcionalni i nefunkcionalni zahtjevi aplikacije.

2.1. Namjena aplikacije

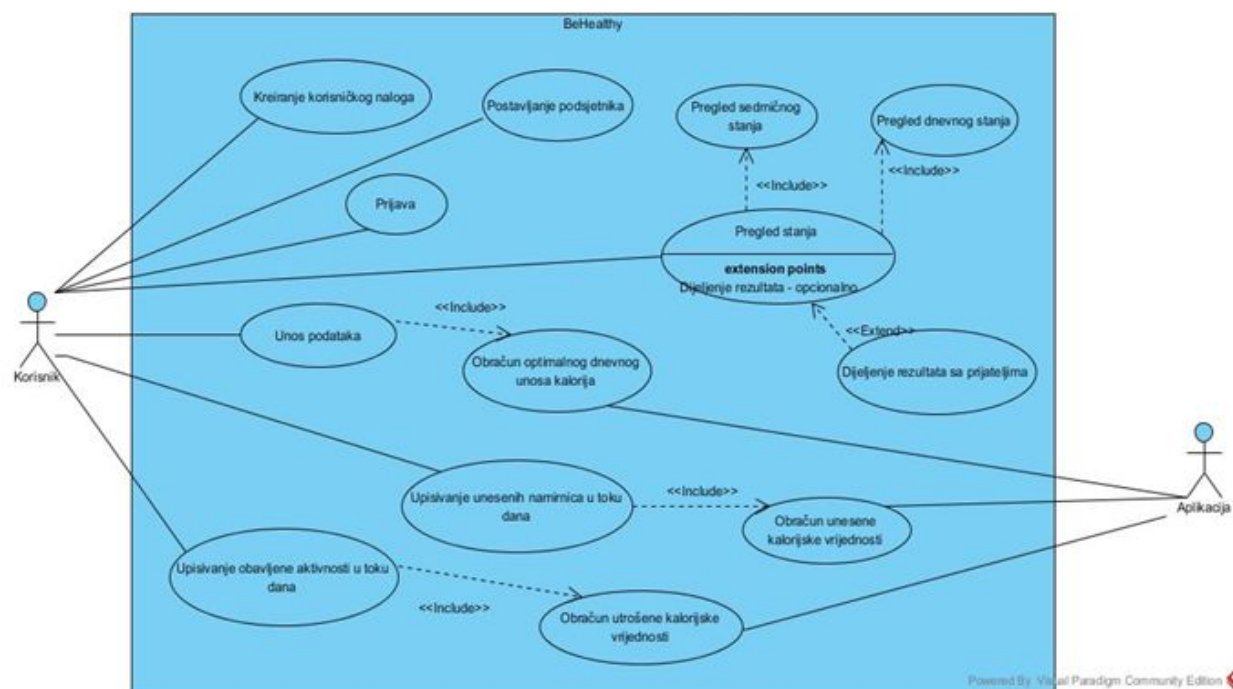
Namjena aplikacije jeste da omogući upoznavanje sa mogućnostima koje nude Windows 10 univerzalne aplikacije. Aplikacija će biti razvijena koristeći C# programski jezik. Prvo se definiše koja su očekivanja od same aplikacije, odnosno, funkcionalni i nefunkcionalni zahtjevi aplikacije. Nakon toga se pristupa kreiranju dijagrama slučajeva upotrebe i korisničkog interfejsa koristeći predstavljeni prototip.

2.2. Funkcionalni zahtjevi

U nastavku su nabrojane osnovne funkcionalnosti koje bi trebale biti omogućene aplikacijom:

- Kreiranje korisničkog računa
- Prijavljivanje
- Unošenje podataka
- Upisivanje unesenih namirnica tokom dana
- Upisivanje obavljenih aktivnosti tokom dana
- Pregled stanja

Nabrojane funkcionalnosti su predstavljene dijagramom slučajeva upotrebe koji je prikazan na slici 1.



Slika 1: Dijagram slučajeva upotrebe za aplikaciju “BeHealthy”

2.3. Prototip aplikacije BeHealthy

Na slici 2 prikazan je prototip aplikacije kako bi trebala da izgleda početna stranica aplikacije sa željenim rasporedom kontrola i sadržajem.



Slika2: Prototip aplikacije BeHealthy

2.4. Detaljna specifikacija funkcionalnih zahtjeva

U nastavku poglavlja detaljno su definisani funkcionalni zahtjevi aplikacije.

2.4.1. Kreiranje korisničkog računa/ Registracija

Glavni akteri	Korisnik
Interesi	Kreiranje korisničkog računa u cilju korištenja dostupnih opcija koje nudi aplikacija.
Preduslovi	-
Rezultat	Mogućnost korištenja dostupnih opcija.
Osnovni tok	Korisnik prvo instalira aplikaciju na željenom uređaju, a nakon toga pristupa kreiranju korisničkog računa. Pri kreiranju korisničkog naloga korisnik unosi ime, prezime, korisničko ime i šifru, pri čemu šifra mora sadržavati 4 znaka, visinu, težinu i starosnu dob.
Alternativni tok	Odbijeno kreiranje korisničkog računa u slučaju postojanja korisnika ili u slučaju pogrešno unesenih podataka.

Tabela1: Kreiranje korisničkog naloga

2.4.2. Prijavljivanje

Glavni akteri	Korisnik
Interesi	Korištenje opcija koje nudi aplikacija.
Preduslovi	Postojanje korisničkog računa.
Rezultat	Korištenje opcija i postizanje željenih rezultata.
Osnovni tok	Nakon uspješno kreiranog korisničkog računa, korisnik vrši prijavu tako što unosi korisničko ime i šifru unesenu pri kreiranju korisničkog računa.
Alternativni tok	Unos pogrešnog korisničkog imena ili šifre.

Tabela2: Prijava

2.4.3. Unošenje podataka

Glavni akteri	Korisnik
Interesi	Unos podataka u cilju određivanja idealne težine i postizanja željenih ciljeva.
Preduslovi	Korisnik mora ispravno unijeti korisničko ime i šifru.
Rezultat	Mogućnost postizanja željenog cilja.
Osnovni tok	Nakon što se korisnik uspješno prijavio on je u mogućnosti da mijenja lične podatke koje se odnose na visinu i težinu. Aplikacija na osnovu unesenih podataka vrši obračun optimalnog unosa kalorija u toku dana.
Alternativni tok	Pogrešno uneseni podaci onemogućavaju obračun optimalnog unosa kalorija.

Tabela3: Unos podataka

2.4.4. Upisivanje unesenih namirnica

Glavni akteri	Korisnik
Interesi	Upisivanje namirnica u cilju obračuna unesenih kalorija u toku dana.
Preduslovi	Korisnik mora biti prijavljen sa svojim korisničkim imenom i šifrom.
Rezultat	Korisnik ima uvid u unesene kalorije tokom dana.
Osnovni tok	Korisnik upisuje namirnice koje je konzumirao u toku dana, pri čemu se unose namirnice u vidu masti, bjelancevina i ugljikohidrata. Nakon toga aplikacija vrši računanje kalorija na osnovu unesenih vrijednosti.
Alternativni tok	Onemogućeno upisivanje unesenih kalorija zbog pogrešno unesenog korisničkog imena ili šifre.

Tabela4: Upis namirnica

2.4.5 Upisivanje obavljenih aktivnosti

Glavni akteri	Korisnik
Interesi	Upisivanje obavljenih aktivnosti u cilju obračuna utrošenih kalorija u toku dana.
Preduslovi	Korisnik mora biti prijavljen sa svojim korisničkim imenom i šifrom.
Rezultat	Korisnik ima uvid u utrošene kalorije tokom dana.
Osnovni tok	Korisnik upisuje aktivnosti kojima se bavio u toku dana i u kom vremenskom trajanju. Na osnovu unesenih aktivnosti aplikacija vrši obačun utrošenih kalorija u toku dana.
Alternativni tok	Onemogućeno upisivanje obavljenih aktivnosti zbog pogrešno unesenog korisničkog imena ili šifre.

Tabela5: Upis aktivnosti

2.4.6.Pregled stanja

Glavni akteri	Korisnik
Interesi	Pregled ostvarenog rezultata
Preduslovi	Korisnik mora biti prijavljen sa svojim korisničkim imenom i šifrom.
Rezultat	Korisnik ima uvid u utrošene kalorije tokom dana ili sedmice.
Osnovni tok	Korisnik ima mogućnost pregleda utrošenih i unesenih kalorija. Aplikacija prikazuje prekoračenu dozvoljenu dozu kalorija s obzirom na unesenu visinu, težinu, spol i starost korisnika ili pak koliko se još smije unijeti kalorija da bi se zadržala optimalna dnevna preporučena količina kalorija.
Alternativni tok	Onemogućeno upisivanje obavljenih aktivnosti zbog pogrešno unesenog korisničkog imena ili šifre.

Tabela6: Pregled stanja

2.5. Nefunkcionalni zahtjevi

Aplikacija je namjenjena za upotrebu od strane korisnika čije je znanje o korištenju pametnih uređaja nemoguće predvidjeti, pa dizajn korisničkog interfejsa treba biti jednostavan i prilagođen intuitivnom korištenju. S obzirom na to, dizajn mora podržavati sve navedene funkcionalnosti sistema, te korisnicima pružiti osjećaj povjerenja i sigurnosti kroz povratne poruke nakon izvršenih akcija. Aplikacija je dostupna svima i u svakom trenutku. Interakcija između korisnika i aplikacije se odvija brzo i bez predugog čekanja.

Izdvajaju se sljedeći nefunkcionalni zahtjevi:

- Aplikacija od korisnika zahtjeva prijavu u vidu korisničkog imena i šifre prije samog pristupa opcijama koje aplikacija nudi.
- Korisnički dizajn će biti na maternjem (bosanskom) jeziku.
- Korisniku će biti pružene adekvatne povratne poruke nakon izvršene akcije.
- Korisnički interfejs ne treba sadržavati suvišne detalje koji mogu zbuniti korisnika.
- Aplikacija će biti skalabilana i bilo kakvo proširenje neće utjecati na njenu upotrebljivost.
- Korisnički interfejs će biti nedvosmislen.
- Odziv aplikacije ne bi trebao biti duži od 5 sekunde.

2.6. Tehnologije za razvoj

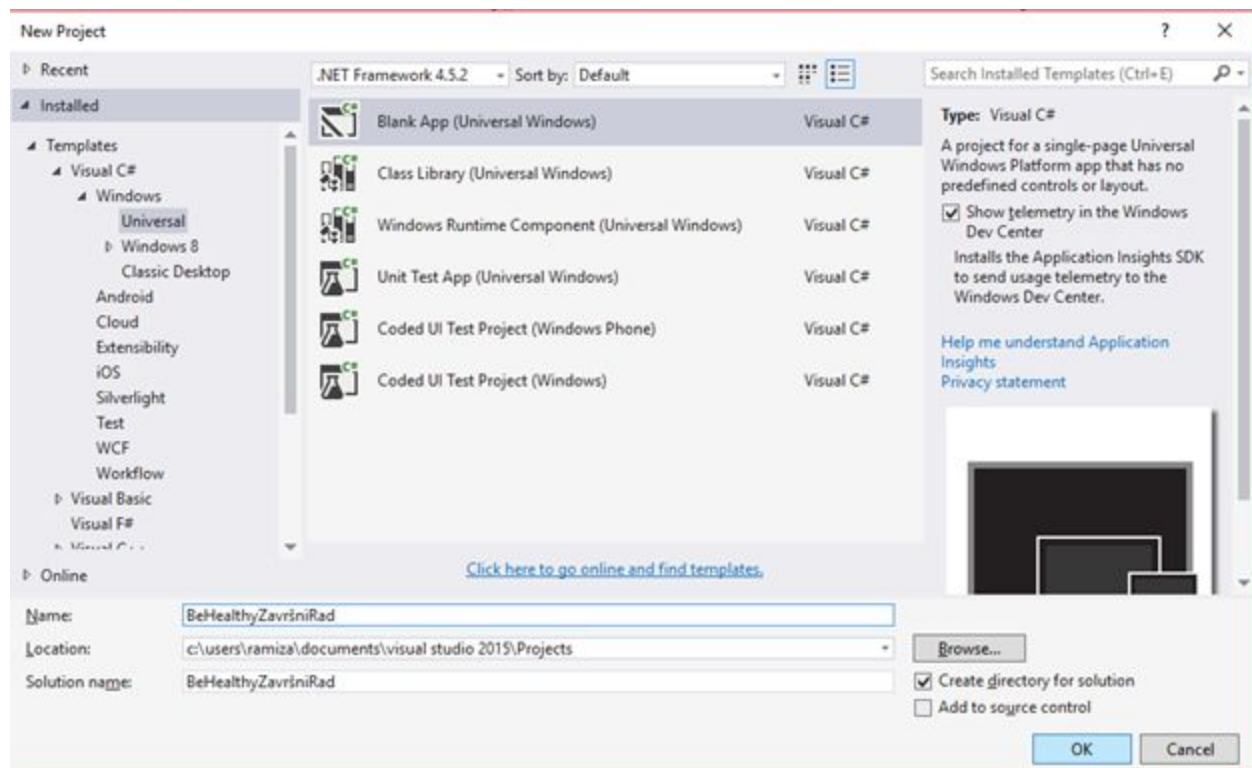
Razvoj Windows 10 univerzalne aplikacije BeHealthy korišteno je Visual Studio 2015 razvojno okruženje, pri čemu je korišten C# programski jezik. Za kreiranje korisničkog intereja koristi se XAML (Extensible Application Markup Language)⁴ deklarativni jezik koji može inicijalizirati objekte i mijenjati njihove osobine. On razdvaja korisnički interfejs od funkcionalnog dijela.

2.7. Kreiranje aplikacije BeHealthy

Da bi se kreirala Windows 10 univerzalna aplikacija potrebno je imati instaliranu posljednju verziju Windows operativnog sistema. Zatim, potrebno je instalirati razvojno okruženje kako bi implementirali i pokrenuli aplikaciju. Za potrebe BeHealthy aplikacije korišten je Visual Studio 2015 koji nudi mnoštvo pogodnosti za razvoj univerzalnih aplikacija. Dakle, kreira se jedan

projekat u okviru Visual Studio 2015 okruženja koji radi na uređajima definisanim XAML kodom aplikacije.

Da bi se kreirala Windows 10 univerzalna aplikacija u File meniju potrebno je izabrati New project, a zatim u meniju sa lijeve strane odabrati Universal. Nakon toga izabrati *Blank App (Universal Windows) Visual C#* kao što je prikazano na slici 3.



Slika3: Kreiranje aplikacije BeHealthy

Nakon toga je potrebno unijeti ime projekta i kliknuti na *OK*, pri čemu se naš projekat pojavljuje u *Solution Exploreru*.

2.8. Zaključak

U okviru ovog poglavlja definisana je namjena i specifikacija aplikacije BeHealthy. Detaljnije su definisani funkcionalni i nefunkcionalni zahtjevi, te tehnologije koje su odabrane za razvoj same aplikacije. Prikazan je i prototip aplikacije.

3. Arhitektura sistema BeHealthy

U okviru ovog poglavlja opisana je arhitektura aplikacije BeHealthy. Opisan je arhitekturni patern⁵ MVVM(Model-View-ViewModel) na kome je bazirana sama aplikacija, te su navedeni dijelovi aplikacije koji ga čine.

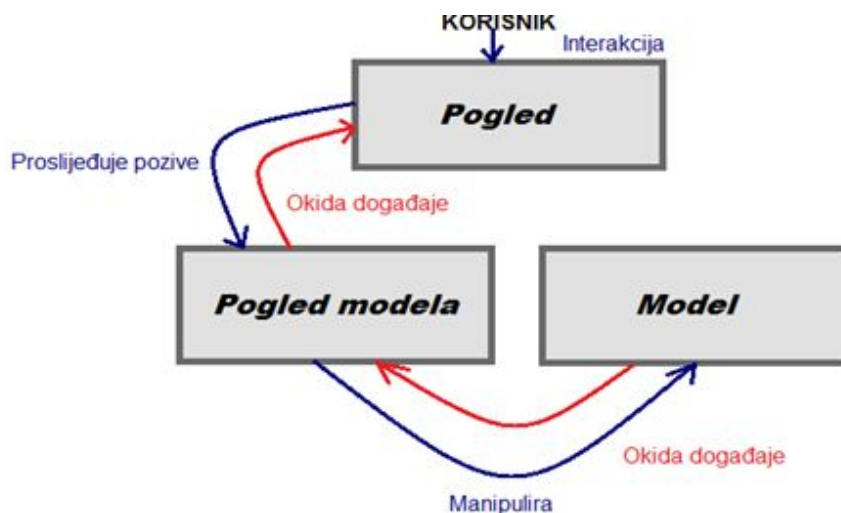
3.1. Analiza MVVM (Model-View-ViewModel) arhitekturnog patern

Model-View-ViewModel (MVVM) je arhitekturni *patern* prilagođen platformama koje podržavaju programiranje vođeno događajima kao što su WPF, Silverlight, UWP, itd. Njegova namjena je potpuno razdvajanje prezentacijskog dijela od same implementacije aplikacije. MVVM *patern* se sastoji iz tri osnovna dijela⁶ :

1. Model (Model) - Podaci i sloj pristupa podacima
2. Pogled (View) - Struktura i izgled podataka
3. Pogled modela (ViewModel) - Posrednik između modela i pogleda.

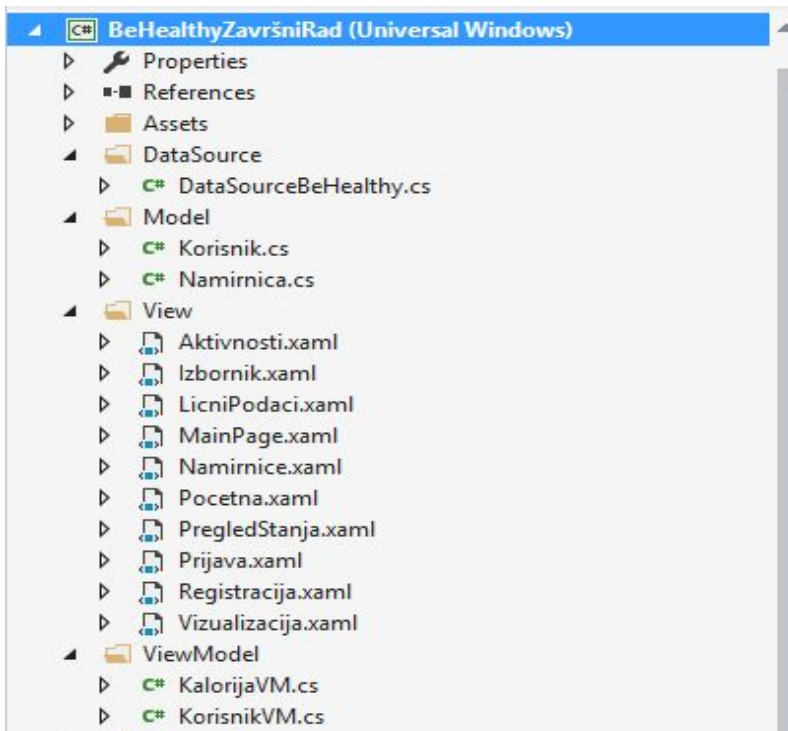
Na slici 4 prikazana je arhitektura MVVM *paterna* i način na koji međusobno komuniciraju njegove komponente.

⁵ Paterni (eng. patterns) nisu dio osnovnog UML ali se široko primjenjuju prilikom dizajna objektno orijentiranih sistema. Dizajn patern je solucija za uobičajene probleme u dizajnu softverskih sistema. Dizajn patern je dakle, ponovo iskoristivo rješenje za uobičajene probleme koji se javljaju prilikom dizajna softvera. Dizajn patern nije gotovo rješenje koje se može direktno pretvoriti u kôd. To je ustvari, opis ili templejt (šablon) za rješavanje problema, koji se mogu primijeniti u mnogo različitih situacija.



Slika4: MVVM arhitekturni patern

Prvo je potrebno izvršiti interakciju sa pogledom koji prikuplja sve potrebne informacije, nakon čega on komunicira sa pogledom modela koji vrši interakciju sa samim modelom⁷. Na slici 5 prikazana je organizacija MVVM *paterna* na arhitekturi BeHealthy aplikacije.



Slika5: Organizacija MVVM paterna u aplikaciji BeHealthy

U aplikaciji BeHealthy napravljeni su folderi u skladu sa MVVM arhitekturnim *paternom* čiji su sadržaji u nastavku detaljnije objašnjeni.

3.2. Model sistema BeHealthy

Model predstavlja domenski objekat koji je odgovoran za očuvanje sistema, te validaciju i poslovnu logiku. On sadrži sve informacije, otkriva funkcionalnosti aplikacije, odgovara o stanju aplikacije i obaviještava pogled o nastalim promjenama⁸.

Struktura aplikacije *BeHealthy* poprilično je jednostavna. Model se sastoji od dvije klase, Korisnik i Namirnica. U klasi Korisnik nalaze se svi potrebni podaci o korisnicima kako bi sistem ispravno funkcionirao, dok se u klasi Namirnica nalaze podaci o unesenim namirnicama radi ispravnog računanja unesenih kalorija. Isječak koda 1 prikazuje attribute i konstruktore klase Korisnik.

```
namespace BeHealthyZavršniRad.Model
{
    public class Korisnik : INotifyPropertyChanged
    {
        private string _ime;
        private string _prezime;
        private string _korisnickoIme;
        private string _sifra;
        private string _visina;
        private string _tezina;
        private DateTime _dob;

        public Korisnik() { }
        public Korisnik(string _ime, string _prezime, string _korisnickoIme, string _sifra, string _visina, string _tezina, DateTime _dob)
        {
            this.Ime = _ime;
            this.Prezime = _prezime;
            this.KorisnickoIme = _korisnickoIme;
            this.Sifra = _sifra;
            this.Visina = _visina;
            this.Tezina = _tezina;
            this.Dob = _dob;
        }
    }
}
```

Isječak koda 1: Klasa **Korisnik**

U svakom seteru za attribute klasa Korisnik i Namirnica se poziva metoda *OnPropertyChanged* koja je naslijeđena iz interfejsa *INotifyPropertyChanged* koja obaviještava pogled modela da je došlo do promjene. Implementacija metode *OnPropertyChanged* se vrši u klasi modela u kojoj se i koristi i prikazana je na isječku koda 2.

```

public event PropertyChangedEventHandler PropertyChanged;

protected void OnPropertyChanged(string propertyName)
{
    if (PropertyChanged != null)
    {
        PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}

```

Isječak koda 2: Implementacija metoda OnPropertyChanged

Kreirana je klasa DataSource koja omogućava inicijalizaciju klasa modela. Isječak koda 3 prikazuje dodavanje testnih korisnika klase Korisnik.

```

namespace BeHealthyZavršniRad.DataSource
{
    public class DataSourceBeHealthy
    {
        #region Korisnik-Dodavanje testnih korisnika
        private static List<Model.Korisnik> _korisnici = new List<Model.Korisnik>()
        {
            new Model.Korisnik()
            {
                Ime = "Ramiza",
                Prezime = "Hošo",
                KorisnickoIme = "rhoso1",
                Sifra = "rhoso1",
                Visina = "169",
                Tezina = "65",
                Dob = Convert.ToDateTime(23)
            }
        };
        #endregion
    }
}

```

Isječak koda 3: Inicijalizacija klase Korisnik

Neke od metoda koje se koriste u višim slojevima podataka su prikazane na isječku koda 4.

```

public static List<Model.Korisnik> dajSveKorisnike()
{
    return _korisnici;
}

public static Model.Korisnik ProvjeraKorisnika()
{
    Model.Korisnik _korisnik = new Model.Korisnik();
    foreach(var k in dajSveKorisnike())
    {
        if (k.KorisnickoIme == _korisnik.KorisnickoIme && k.Sifra == _korisnik.Sifra)
        {
            _korisnik = k;
        }
    }
    return _korisnik;
}
#endregion

```

Isječak koda 4: Metode koje se koriste u višim slojevima podataka

3.3. Pogled sistema BeHealthy

Pogled sistema zadužen je za definisanje strukture i izgleda, odnosno, onog što korisnik zapravo vidi. U nastavku je prikazan pogled početne stranice sistema BeHealthy i njegov XAML kod, dok je kod izuzetno jednostavan i ne sadrži nikakvu poslovnu logiku sistema. Najjednostavniji primjer u sistemu BeHealthy upravo je početna stranica koja vrši preusmjeravanje na prijavu/registraciju čiji je XAML kod prikazan na isječku koda 5.a i 5.b. Dakle, ovaj pogled je oslobođen pozadinskog koda.

```
<Grid>
  <VisualStateManager.VisualStateGroups>
    <VisualStateGroup>
      <VisualState x:Name="Phone">
        <VisualState.StateTriggers>
          <AdaptiveTrigger MinWindowWidth="300" />
        </VisualState.StateTriggers>
        <VisualState.Setters>
          <Setter Target="tekst.Width" Value="235" />
          <Setter Target="tekst.Height" Value="110" />
          <Setter Target="tekst.FontSize" Value="14" />
          <Setter Target="tekst.Margin" Value="16,285,0,0" />
          <Setter Target="prijavaBtn.Margin" Value="135,18,0,0" />
          <Setter Target="prijavaBtn.Width" Value="112" />
          <Setter Target="prijavaBtn.Height" Value="50" />
          <Setter Target="prijavaBtn.FontSize" Value="24" />
          <Setter Target="registracijaBtn.Margin" Value="245,22,0,0" />
          <Setter Target="registracijaBtn.Width" Value="190" />
          <Setter Target="registracijaBtn.Height" Value="47" />
          <Setter Target="registracijaBtn.FontSize" Value="24" />
        </VisualState.Setters>
      </VisualState>
    </VisualStateGroup>
  </VisualStateManager.VisualStateGroups>
</Grid>
```

Isječak koda 5.a: XAML kod pogleda Početna.xaml

```
<Grid.Background>
  <ImageBrush x:Name="pozadina" Stretch="Fill" ImageSource="ms-appx:///Assets/pocetna.jpg"/>
</Grid.Background>
<Grid HorizontalAlignment="Left" Height="700" Margin="10,10,0,0" VerticalAlignment="Top" Width="1260">

  <Button x:Name="prijavaBtn" Content="Prijava" HorizontalAlignment="Left" VerticalAlignment="Top"
    FontFamily="SketchFlow Print" FontWeight="Bold">
    <Button.BorderBrush>
      <ImageBrush Stretch="Fill"/>
    </Button.BorderBrush>
    <Button.Background>
      <ImageBrush Stretch="Fill"/>
    </Button.Background>
  </Button>
  <Button x:Name="registracijaBtn" Content="/ Registracija" HorizontalAlignment="Left"
    VerticalAlignment="Top" FontFamily="SketchFlow Print" FontWeight="Bold">
    <Button.Background>
      <ImageBrush Stretch="Fill"/>
    </Button.Background>
    <Button.BorderBrush>
      <ImageBrush Stretch="Fill"/>
    </Button.BorderBrush>
  </Button>
</Grid>
```

Isječak koda 5.b: XAML kod pogleda Početna.xaml

U XAML kodu pogleda Početna definisan je adaptivni⁹ *layout* koji se u potpunosti prilagođava svim rezolucijama uređaja bilo da se radi o mobilnim, tablet, desktop ili Surface uređajima. Na slici 6.a i 6.b prikazan je dizajn pogleda Početna za desktop i mobilne uređaje.



Slika 6.a: Dizajn pogleda Početna.xaml za desktop uređaje



Slika 6.a: Dizajn pogleda Početna.xaml za mobilne uređaje

U asinhronoj metodi **okReg_click** prikazanoj na isječku koda 6 vrši se preuzimanje podataka sa Registracija.xaml pogleda. Ukoliko uneseni korisnik već postoji vrši se preusmjerenje na prijavu, dok se u suprotnom uspješno izvrši registracija.

```
private async void okReg_Click(object sender, RoutedEventArgs e)
{
    string _sifra = Convert.ToString(passRegistracija.TextReadingOrder);
    var _korisnik = new Model.Korisnik(ImeTxt.Text, prezimeTxt.Text, kImeTxt.Text, _sifra, visinaReg.Text, tezinaReg.Text, dobreg.Text);

    if (DataSource.DataSourceBeHealthy.DajKorisnikaPoKorisnickomImenu(kImeTxt.Text) == _korisnik)
    {
        var dialog = new Windows.UI.Popups.MessageDialog("Korisnik vec postoji. Prijavite se");
        await dialog.ShowAsync();
        this.Frame.Navigate(typeof(Prijava));
    }
    else
    {
        ViewModel.KorisnikVM.dodajKorisnika(_korisnik);
        var dialog = new Windows.UI.Popups.MessageDialog("Korisnik uspješno dodan");
        await dialog.ShowAsync();
        this.Frame.Navigate(typeof(Izbornik));
    }
}
```

Isječak koda 6: Klasa Registracija.xaml.cs

Da bi se prikazala vrijednost podataka potrebno je podesiti osobinu *Text* kontrola *TextBox* tako što dodamo {*Binding Naziv*} kao što je prikazano na isječku koda 7.

```
<TextBox x:Name="ImeTxt" Text="{Binding Ime}" HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top"/>
<TextBox x:Name="prezimeTxt" Text="{Binding Prezime}" HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top"/>
<TextBox x:Name="kImeTxt" Text="{Binding KorisnickoIme}" HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top"/>
<TextBox x:Name="visinaReg" Text="{Binding Visina}" HorizontalAlignment="Left" VerticalAlignment="Top"/>
<TextBox x:Name="tezinaReg" Text="{Binding Tezina}" HorizontalAlignment="Left" VerticalAlignment="Top"/>
<TextBox x:Name="dobreg" Text="{Binding Dob}" HorizontalAlignment="Left" VerticalAlignment="Top"/>
```

Isječak koda 7: Dio XAML koda pogleda Registracija.xaml

3.4. Pogled modela sistema BeHealthy

Pogled modela je posrednik između modela i pogleda. Njegova uloga jeste da prilagodi informacije koje dobije iz modela na onaj način koji odgovara pogledu. Pogled modela implementira osobine i komande pomoću kojih se vrši povezivanje i obavještanje pogleda.

Pogled modela sistema BeHealthy se sastoji od dvije klase **KorisnikVM** i **KalorijaVM** koje posjeduju attribute, metode i događaje za povezivanje pogleda modela sa pogledom sistema.

Isječak koda 8.a prikazuje atribute i konstruktor klase *KorisnikVM*, a isječak koda 8.b prikazuje implementaciju metode *KorisniciViewModel* koja vrši mapiranje elemenata iz modela u pogled modela.

```
namespace BeHealthyZavršniRad.ViewModel
{
    public class KorisnikVM
    {
        private string _imeVM;
        private string _prezimeVM;
        private string _korisnickoImeVM;
        private string _sifraVM;
        private string _visinaVM;
        private string _tezinaVM;
        private string _dobVM;

        public KorisnikVM(string _imeVM, string _prezimeVM, string _korisnickoImeVM, string _sifraVM, string _visinaVM, string _tezinaVM, string _dobVM)
        {
            this.ImeVM = _imeVM;
            this.PrezimeVM = _prezimeVM;
            this.KorisnickoImeVM = _korisnickoImeVM;
            this.SifraVM = _sifraVM;
            this.VisinaVM = _visinaVM;
            this.TezinaVM = _tezinaVM;
            this.DobVM = _dobVM;
        }
    }
}
```

Isječak koda 8.a: Kod pogleda modela klase KorisnikVM

```
public static void dodajKorisnika(Korisnik korisnik)
{
    _korisnici.Add(korisnik);
}

public static List<Model.Korisnik> DajSveKorisnike()
{
    return _korisnici;
}

//metoda za mapiranje elemenata iz Modela u ViewModel koji se koriste u View-u
public static KorisnikVM KorisniciViewModel(Korisnik _korisnik)
{
    var _viewModel = new KorisnikVM();

    _viewModel.ImeVM = _korisnik.Ime;
    _viewModel.PrezimeVM = _korisnik.Prezime;
    _viewModel.KorisnickoImeVM = _korisnik.KorisnickoIme;
    _viewModel.SifraVM = _korisnik.Sifra;
    _viewModel.VisinaVM = _korisnik.Visina;
    _viewModel.TezinaVM = _korisnik.Tezina;
    _viewModel.DobVM = _korisnik.Dob;

    return _viewModel;
}
```

Isječak koda 8.b: Kod pogleda modela klase KorisnikVM

3.5.Zaključak

U okviru ovog poglavlja detaljnije je objašnjena arhitektura sistema BeHealthy. Analiziran je arhitekturni *patern* MVVM koji je korišten pri implementaciji sistema. Nakon detaljnije analize arhitekturnog *paterna* dolazi se do zaključka da korištenje *paterna* pri implementaciji sistema vodi ka kvalitetnijim temeljima sistema i izbjegavanju neočekivanim situacijama u budućnosti, jer se svi sistemi vremenom nadograđuju.

4. Vizualizacija podataka sistema BeHealthy

U okviru ovog poglavlja opisano je šta predstavlja vizualizacija podataka. Zatim je na sistemu BeHealthy prikazana vizualizacija podataka.

4.1. Vizualizacija podataka

Vizualizacija podataka¹⁰ jeste grafička prezentacija podataka i ona vodi ka njihovom jednostavnijem i bržem prikazu. Ljudski mozak brže pamti vizualne prikaze od tekstualnih i na osnovu vizuelnog prikaza on je u mogućnosti da zapamti ogromnu količinu podataka.

Podatke je moguće vizualizirati na različite načine pri čemu svaki od vizualnih prikaza ima svoje prednosti. Da bi se vizualizacija izvršila na pravilan način potrebno je poznavati podatke koje je potrebno vizualizirati i veze koje postoje između njih.

4.2. Uobičajni vizuelni prikazi

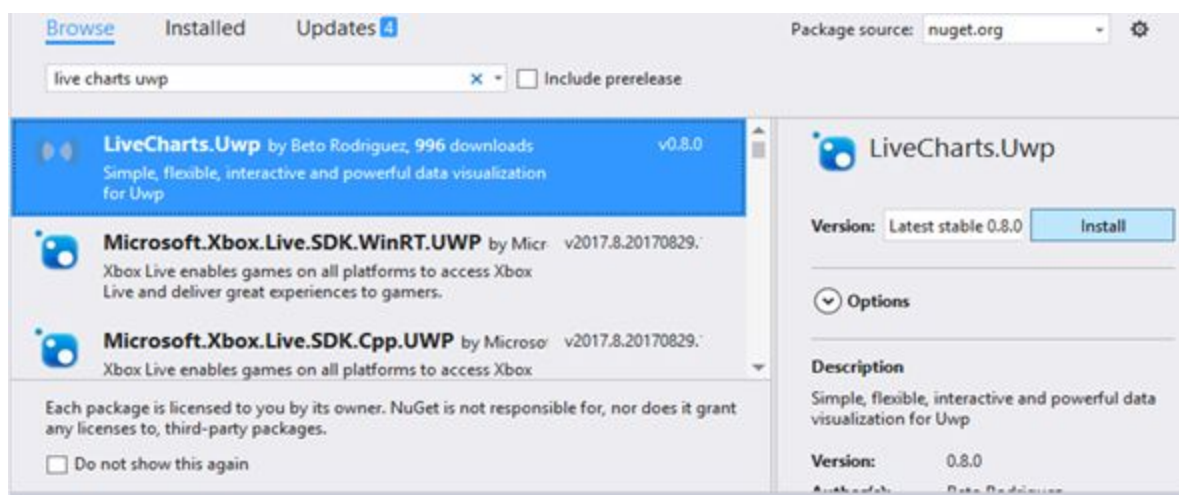
Prikupljeni podaci se vizualiziraju pomoću različitih vrsta dijagrama, tablica ili grafikona i na zanimljiviji način prezentiraju korisniku. Neki od načina vizualizacije podataka su¹¹:

- Stupčasti grafikoni (*eng Bar Charts*) - najbolji za prikaz i poređenje podataka koji se mijenjaju tokom vremena.
- Kružni grafikoni (*eng Pie Charts*) - su izuzetno jednostavni za prikaz dijelova koji čine jednu cjelinu.
- Linijski grafikoni (*eng Line Charts*) - koriste linije kako bi se pokazale promjene nad podacima kroz vrijeme koje se i dalje dešavaju.
- Raspršeni grafikon (*eng Scatter Plot*) - pogodni za prikaz velike količine podataka na jednom grafikonu. Slični su linijskim grafikonima.

4.3. Kreiranje grafikona sistema BeHealthy

Postoji niz alata koji omogućavaju vizualizaciju podataka. Međutim, najjednostavniji način vizualizacije podataka jeste pomoću grafikona. Biblioteka *LiveCharts* omogućava jednostavno kreiranje grafikona za Windows 10 univerzalnu aplikaciju.

Prvi korak za vizualizaciju podataka jeste dodavanje biblioteke u sistem BeHealthy, tako što se nakon desnog klika na *References* u *SolutionExplorer*-u BeHealthy aplikacije odabere *Manage NuGet Packages*, a zatim se vrši instaliranje kao što je prikazano na slici 7.



Slika 7: Instaliranje biblioteke LiveCharts

Nakon uspješne instalacije u XAML potrebno je dodati liniju koda koja omogućava korištenje biblioteke kao što je prikazano na isječku koda 9.

```
x:Class="BeHealthyZavršniRad.View.PregledStanja"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:local="using:BeHealthyZavršniRad.View"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:lvc="using:LiveCharts.Uwp"
mc:Ignorable="d" d:DataContext="{d:DesignInstance local:MainPage}">
```

Isječak koda 9: Dodavanje XAML koda

Nakon uspješne prijave/registracije korisniku sistema BeHealthy se nudi mogućnost unosa masti, bjelancevina i ugljikohidrata koje je konzumirao, nakon čega aplikacija vrši računanje koliko kalorija je uneseno u toku dana. Na osnovu tih unesenih podataka vrši se njihova vizualizacija pri

čemu korisnik ima uvid u unesene stvari od dana registracije do posljednjeg korištenja. Na osnovu tih vizualnih podataka korisnik je u mogućnosti da poboljša ili umanjí unos osnovnih potrebnih stvari u toku dana.

Biblioteka *LiveCharts* nudi različite klase za crtanje grafikona potrebnih za vizualizaciju podataka, dok je za potrebe sistema BeHealthy odabrana *CartesianChart* klasa koja je izuzetno jednostavna i nudi predefinisane vrijednosti.

Vizualizacija podataka sistema BeHealthy prikazana je u vidu grafikona koji sadrži dvije ose (eng *Axis*)¹², X i Y koje prikazuju željene podatke. Biblioteka *LiveCharts* posjeduje *Axis* klasu za iscrtavanje osa. Jedna osa, u ovom slučaju Y osa, prikazuje numeričku skalu, dok druga osa, odnosno, X osa prikazuje vrijednosti datuma kada su uneseni podaci. Ukoliko se nekoj od osa želi dodijeliti ime to je moguće pomoću osobine *Title*, a pozicija osa određena je osobinom *Location*.

Sve vrijednosti na grafičkom prikazu podataka se prikazuju uz pomoć labela (eng. Labels), pri čemu postoje formatirane i mapirane labele. Za potrebe sistema BeHealthy je korištena formatirana, jer postoji direktna veza između podataka koji su prikazani na grafikonu sa numeričkim vrijednostima osa X i Y.

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
  <lvc:CartesianChart FontFamily="SketchFlow Print" FontSize="14" Series="{x:Bind SeriesCollection}"
    LegendLocation="Left">
    <lvc:CartesianChart.AxisX>
      <lvc:Axis Width="5" FontFamily="SketchFlow Print" FontSize="14" Labels="{x:Bind Labels}"/>
    </lvc:CartesianChart.AxisX>
    <lvc:CartesianChart.AxisY>
      <lvc:Axis FontFamily="SketchFlow Print" FontSize="14" LabelFormatter="{x:Bind Formatter}"/>
    </lvc:CartesianChart.AxisY>
  </lvc:CartesianChart>
</Grid>
```

Isječak koda 10: XAML kod za crtanje grafikona

Isječak koda 10 prikazuje XAML kod za crtanje grafikona upotrebom osa i njihovih osobina. Pozadinski kod koji se koristi pri vizualizaciji podataka prikazan je na isječku koda 11.a i 11.b.

```

public sealed partial class Vizualizacija : Page
{
    public Vizualizacija()
    {
        InitializeComponent();

        SeriesCollection = new SeriesCollection
        {
            new ColumnSeries
            {
                Title = "Masti",
                Values = new ChartValues<double> { Double.Parse(String.Join(",", Namirnica.DajSveBjelancevine())) }
            }
        };
    }
}

```

Isječak koda 11.a: C# kod vizualizacije podataka

```

        Labels = new[] { String.Join(",", Namirnica.DajSveDatume()) };
        Formatter = value => value.ToString("N");

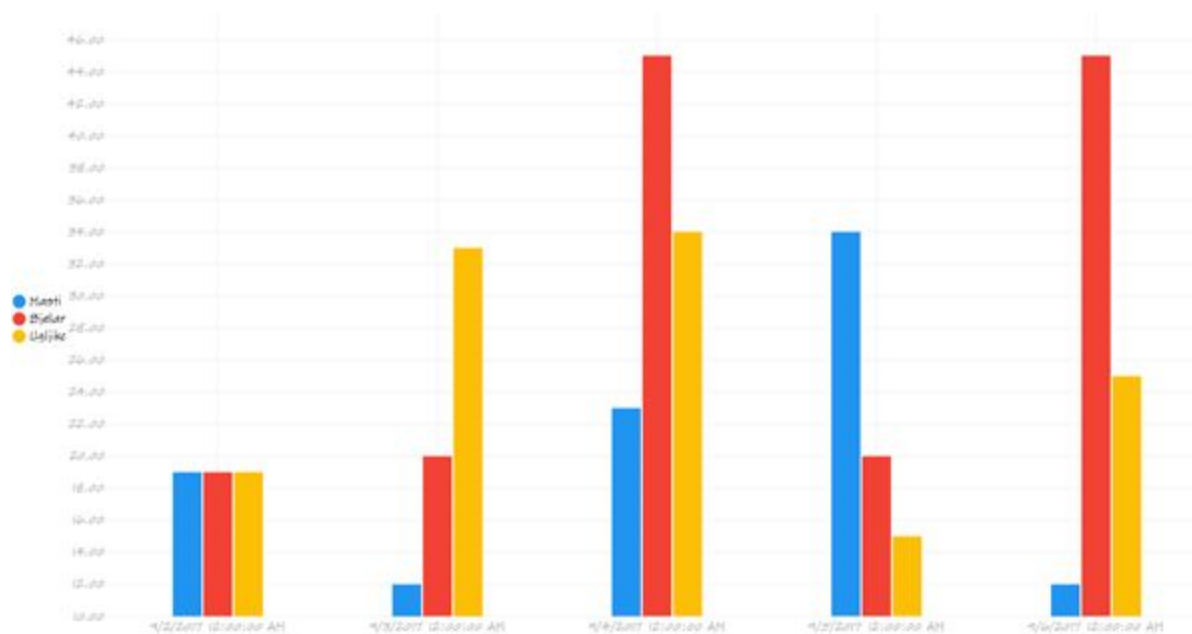
        DataContext = this;
    }

    public SeriesCollection SeriesCollection { get; set; }
    public string[] Labels { get; set; }
    public Func<double, string> Formatter { get; set; }
}
}

```

Isječak koda 11.b: C# kod vizualizacije podataka

Kolekcijom *SeriesCollection* se kreiraju kolone za grafički prikaz. U sistemu BeHealthy postoje tri različite kolone koje prikazuju vrijednosti za bjelancevine, masti i ugljikohidrate koje su unesene u toku dana. Na slici 8 prikazana je vizualizacija podataka koja je korisniku sistema BeHealthy dostupna ukoliko u meniju sistema odabere Vizualizacija.



Slika 8: Vizualni prikaz unesenih podataka

4.4. Zaključak

U okviru ovog poglavlja je opisana vizualizacija pomoću biblioteke *LiveCharts* u C# programskom jeziku, te je prikazana vizualizacija podataka sistema BeHealthy. Prikazane su klase biblioteke *LiveCharts* i njihove mogućnosti i osobine.

5. Zaključak

Cilj ovog rada jeste da prikaže mogućnosti koje nudi Windows 10 Univerzalna aplikacija, te će rad biti zaključen osvrtnjem na glavne prednosti koje se nudi.

Jedna od osnovnih prednosti koje nude univerzalne aplikacije jeste mogućnost prilagođavanja uređajima različitih rezolucija bilo da se radi o mobilnim, desktop, tablet uređajima ili pak uređajima velikih rezolucija kao što su Surface i Hub.

Pojava univerzalnih aplikacija je minimizirala posao inženjera. Dakle, dovoljno je pisati jedan programski kod aplikacije koja je zatim dostupna na svim uređajima sa instaliranim Windows 10 operativnim sistemom.

Također, implementirana je vizualizacija podataka koja grafički prikazuje vrijednosti tvari koje su unesene u toku dana. Vizualni prikaz podataka je izuzetno bitan za korisnike, jer se jednostavnije pamte podaci koji su zapisani na vizuelni način u odnosu na podatke koju se zapisani tekstualno. U okviru Visual Studio 2015 okruženja nudi se biblioteka LiveChartskoja nudi mnoštvo pogodnosti za dinamičko prikazivanje vizuelnih podataka kao što je prikazano u poglavlju 4.

Razvijena je Windows 10 univerzalna aplikacija koja služi za prikaz osnovnih pogodnosti koje su došle sa univerzalnim aplikacijama kao što je mogućnost prilagođavanja uređajima različitih rezolucija korištenjem adaptivnih layout-a. Na aplikaciji BeHealthy je prikazan i arhitekturni patern kao što je opisano u poglavlju 3, te tako predstavlja dobre temelje za buduća poboljšanja dodavanjem novih funkcionalnosti.

6. Reference

[1] - Izjava o autentičnosti rada

Izjava autora koju koristi Elektrotehnički fakultet u Sarajevu; *Izjava o autentičnosti završnog rada* Centra za interdisciplinarne studije – master studij „Evropske studije”, *Izjava o plagijarizmu* koju koristi Fakultet političkih nauka u Sarajevu. *plagijarizmu* koju koristi Fakultet političkih nauka u Sarajevu.

[2] – Windows 10

HyperLink: “https://en.wikipedia.org/wiki/Windows_10”

Datum preuzimanja: 15.07.2017.

[3] – Introduction to UWP app design

HyperLink: ” <https://docs.microsoft.com/en-us/windows/uwp/layout/design-and-ui-intro> “

Datum preuzimanja: 15.07.2017.

[4] - What is XAML?

HyperLink: “ <https://msdn.microsoft.com/en-us/library/cc295302.aspx> ”

Datum preuzimanja: 17.07.2017.

[5] - Dženana Đonko, Samir Omanović 2009. *Objektno orijentirana analiza i dizajn primjenom UML notacije*. 1.Izdanje. Sarajevo: Elektrotehnički fakultet u Sarajevu.

[6] – The MVVM Pattern

HyperLink: “ <https://msdn.microsoft.com/en-us/library/hh848246.aspx> ”

Datum preuzimanja: 02.08.2017.

[7] – Raffaele Garofalo, 2011. *Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern*, 1st Edition, Published with the authorization of Microsoft Corporation by O'Reilly Media, Inc., California

[8] - The MVVM Pattern

HyperLink: “ <https://msdn.microsoft.com/en-us/library/hh848246.aspx> ”

Datum preuzimanja: 02.08.2017.

[9] - Adaptive layout for Universal Windows Platform apps

HyperLink:

<https://developer.microsoft.com/en-us/windows/projects/events/build/2017/adaptive-layout-for-universal-windows-platform-apps>

Datum preuzimanja: 02.08.2017.

[10] – Alberto Cairo, 2012. *The Functional Art*, Published by New Riders Publishing, Berkley, California.

[11] – Data Visualisation 101: How to design charts and graphs

HyperLink:

“https://cdn2.hubspot.net/hub/53/file863940581pdf/Data_Visualization_101_How_to_Design_Charts_and_Graphs.pdf”

Datum preuzimanja: 09.09.2017.

[12] – Axis class

HyperLink: “<https://lvcharts.net/App/documentation/beta/uwp/LiveCharts-Uwp-Axis>”

[Datum preuzimanja: 22.08.2017.]

7. Popis slika

- [1] – Dijagram slučajeva upotrebe za aplikaciju BeHealthy
- [2] – Prototip aplikacije BeHealthy
- [3] – Kreiranje aplikacije BeHealthy
- [4] – MVVM arhitekturni *patern*
- [5] – Organizacija MVVM arhitekturnog *paterna* u aplikaciji BeHealthy
- [6.a] – Dizajn pogleda Početna.xaml za desktop uređaje
- [6.b] - Dizajn pogleda Početna.xaml za mobilne uređaje
- [7] – Instaliranje biblioteke *LiveCharts*
- [8] – Vizualni prikaz unesenih podataka

7.1. Popis slika ispod kodova

- [1] – Klasa Korisnik
- [2] – Implementacija metode *OnPropertyChanged*
- [3] – Inicijalizacija klase Korisnik
- [4] – Metode koje se koriste u višim slojevima podataka
- [5.a] – XAML kod pogleda Pocetna.xaml
- [5.b] - XAML kod pogleda Pocetna.xaml

[6] – Klasa Registracija.xaml.cs

[7] - Dio XAML koda pogleda Registracija.xaml

[8.a] – Kod pogleda modela klase KorisnikVM

[8.b] - Kod pogleda modela klase KorisnikVM

[9] – Dodavanje XAML koda

[10] – XAML kod za crtanje grafikona

[11.a] – C# kod vizualizacije podataka

[11.b] - C# kod vizualizacije podataka

8. Popis tabela

[1] - Kreiranje korisničkog naloga

[2] – Prijava

[3] - Unos podataka

[4] - Upis namirnica

[5] - Upis aktivnosti

[6] - Pregled stanja

Prilog - Prikaz pokazne aplikacije BeHealthy

Pokazna aplikacija BeHealthy je implementirana kako bi se pokazalo osnovne mogućnosti Windows 10 univerzalnih aplikacija.

Pažnja prilikom implementiranja posvećena je adaptivnom layout-u koji nudi mogućnost korištenja aplikacije na različitim uređajima.

U nastavku teksta biće prikazana aplikacija uz kratak opis funkcionalnosti.

Nakon što korisnik pokrene aplikaciju pojavljuje se početna stranica kao na narednoj slici:



Prilog - Slika1: Početna stranica aplikacije

Nakon toga, ukoliko korisnik prvi put koristi aplikaciju potrebno je da izvrši registraciju klikom na *Registracija* u gornjem desnom uglu. Ukoliko korisnik ima kreiran korisnički račun onda je potrebno da klikne na *Prijava* da bi pristupio funkcionalnostima sistema.

Ukoliko korisnik odabere registraciju, otvara se nova forma koja zahtjeva unos osnovnih podataka kako bi aplikacija obavila potrebne funkcionalnosti. Izgled forme za registraciju je prikazan na narednoj slici.

The screenshot shows a software window titled 'BeHealthyZavršniRad'. The background is a blue abstract design. At the top, the text 'Be Healthy' is written in a large, black, handwritten-style font. Below this, there are seven input fields arranged vertically. The first four are labeled 'Ime:', 'Prezime:', 'Korisničko ime:', and 'Lozinka:'. The next three are labeled 'Visina:', 'Težina:', and 'Dob:'. The 'Visina:' field has 'cm' to its right, 'Težina:' has 'kg' to its right, and 'Dob:' has 'god.' to its right. At the bottom right of the form area, there are two buttons: 'Nazad' and 'OK'.

Be Healthy

Ime:

Prezime:

Korisničko ime:

Lozinka:

Visina: cm

Težina: kg

Dob: god.

Nazad OK

Prilog - Slika2: Stranica za registraciju

Dakle, korisnik unosi potrebne podatke i nakon pritiska na dugme OK vrši se registracija novog korisnika. Ukoliko korisnički račun već postoji korisnik se o tome obavještava i preusmjerava na stranicu za prijavu. Nakon unosa podataka vrši se validacija podataka.

Forma za prijavu korisnika sa već postojećim korisničkim računom prikazana je na narednoj slici.

The screenshot shows a software window titled 'BeHealthyZavršniRad'. The background is a blue abstract design. At the top, the text 'Be Healthy' is written in a large, black, handwritten-style font. Below this, there are two input fields arranged vertically. The first is labeled 'Korisničko ime:' and the second is labeled 'Lozinka:'. At the bottom right of the form area, there are two buttons: 'Nazad' and 'OK'.

Be Healthy

Korisničko ime:

Lozinka:

Nazad OK

Prilog - Slika3: Stranica za prijavu korisnika

Nakon uspješne prijave/registracije otvara se prozor sa izbornikom, gdje se korisniku nude neke od funkcionalnosti.



Prilog - Slika4: Izbornik aplikacije

Pritiskom na dugme *Lični Podaci* otvara se forma sa podacima prijavljenog korisnika, pri čemu je moguće promijeniti šifru, visinu, težinu i dob.

The screenshot shows a window titled 'BeHealthyZavršniRad'. The main heading is 'Be Healthy'. Below it, the following information is displayed: 'Ime: Ramiza', 'Prezime: Hoso', and 'Korisničko ime: rhosol'. There are input fields for 'Lozinka:', 'Visina:', 'Težina:', and 'Dob:'. The 'Visina:' field is followed by 'cm', 'Težina:' by 'kg', and 'Dob:' by 'god.'. At the bottom right, there are two buttons labeled 'Nazad' and 'OK'.

Prilog - Slika5: Pregled ličnih podataka

Pritiskom na dugme *Nazad* vraća se na izbornik aplikacije, dok pritiskom na dugme *OK* vrši se ažuriranje promijenjenih podataka.

Korisnik ima mogućnost unosa materija koje je konzumirao u toku dana. Jedan gram bjelanjčevina kao i jedan gram ugljikohidrata ima vrijednost od četiri kalorije, dok jedan gram masti ima vrijednost od devet kalorija. Nakon što korisnik unese koliko grama je kojih tvari konzumirao i pritisne na dugme *Izračunaj kalorije* u textBlock kontrolu se upisuje koliko je ukupno kalorija konzumirao u toku tog dana.

Be Healthy

Masti: gr

Ugljikohidrati: gr

Bjelančevine: gr

Uneseno:

Prilog - Slika 6: Unos osnovnih materija

Također, za pravilno računanje koliko kalorija je uneseno, utrošeno i prekoračena doza na osnovu visine, težine i starosne dobi bitno je unijeti sportske aktivnosti kojima se korisnik bavio u toku dana. Tako korisnik ima mogućnost unosa sportske aktivnosti i vremenskog trajanju u cilju računanja koliko kalorija je utrošeno, što se također upisuje u *TextBlock* kontrolu.

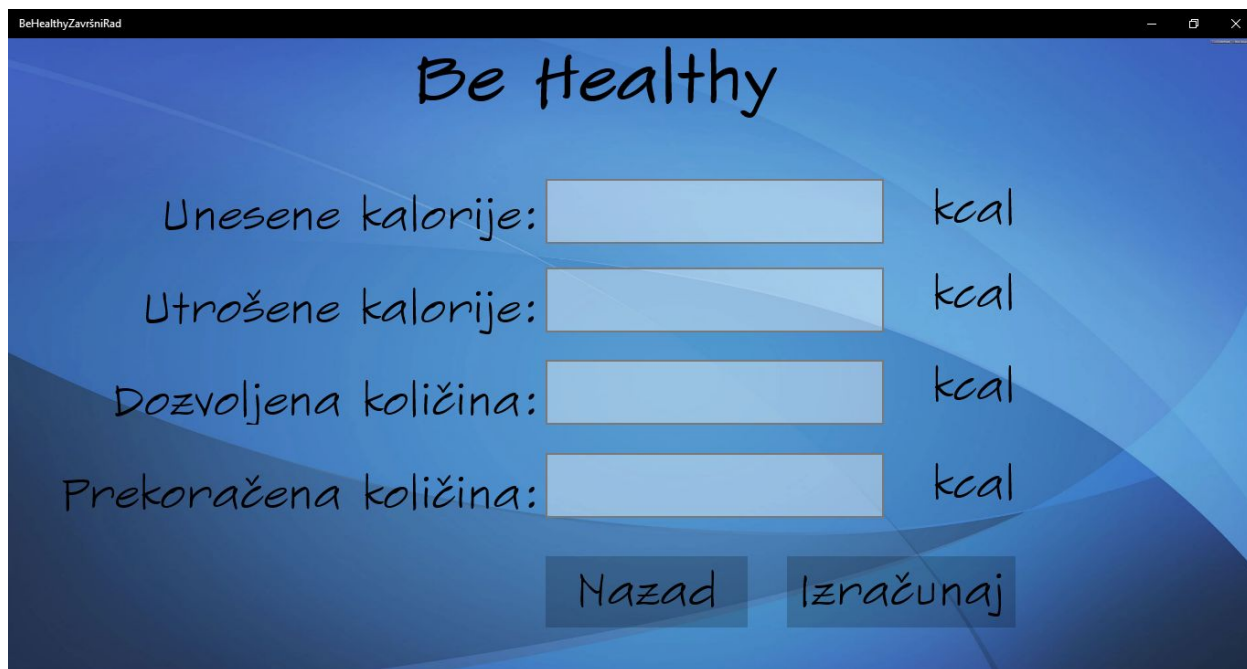
Be Healthy

Dnevna aktivnost:

Trajanje: minuta

Prilog - Slika 7: Unos sportskih aktivnosti

Zatim, korisnik ima mogućnost da na jednoj formi pregleda stanje za taj dan ukoliko odabere opciju *Pregled stanja*. Na osnovu visine, težine i starosne dobi se određuje dozvoljena količina kalorija u toku jednog dana, dok se prekoračena količina računa računom pritiskom na dugme *Izračunaj* na osnovu unesenih, utrošenih i dozvoljenih vrijednosti kalorija.



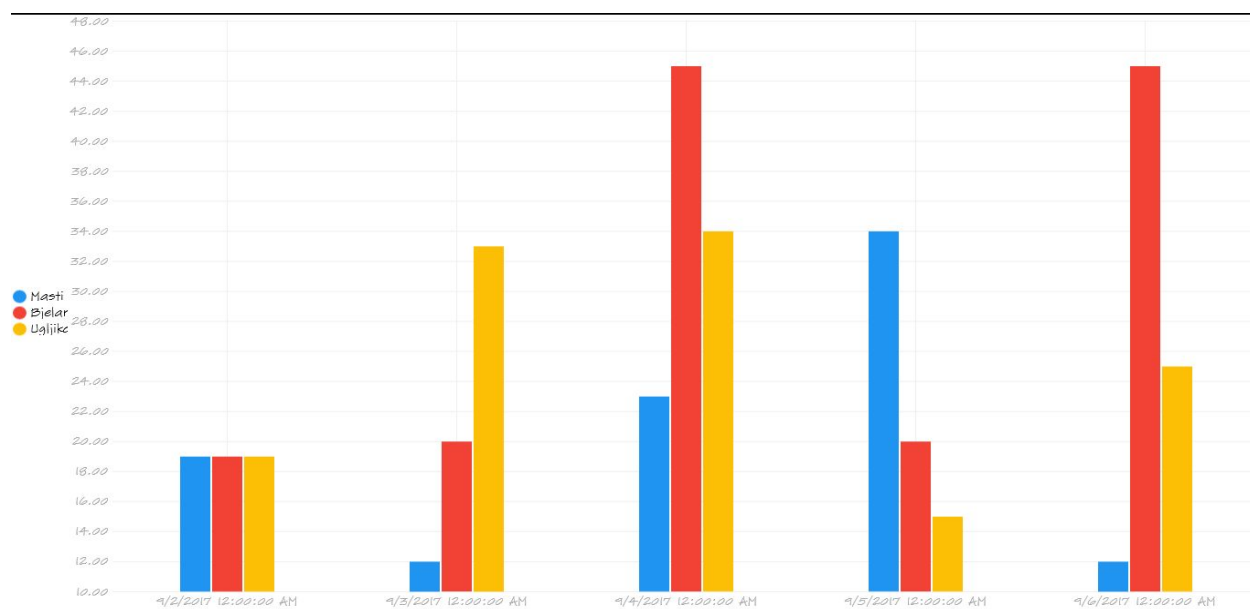
The screenshot shows a window titled "BeHealthyZavršniRad". The main heading is "Be Healthy". Below it, there are four input fields with labels and units:

- Unesene kalorije: kcal
- Utrošene kalorije: kcal
- Dozvoljena količina: kcal
- Prekoračena količina: kcal

At the bottom, there are two buttons: "Nazad" and "Izračunaj".

Prilog - Slika 8: Pregled stanja

Još jedna od funkcionalnosti koju nudi aplikacija BeHealthy jeste vizuelni prikaz unosa osnovnih materija od dana prvog korištenja aplikacije. Ukoliko korisnik odabere opciju *Vizualizacija* otvara se forma sa grafikonom na kojem su prikazane vrijednosti masti, bjelančevina i ugljikohidrata.



Prilog - Slika 9: Vizuelni prikaz materija

Kao što je pomenuto u poglavlju 3 i prikazano na slikama, prilikom implementacije aplikacije korišten je adaptivni layout koji nudi mogućnost prilagođavanja aplikacije, u slučaju aplikacije BeHealthy mobilnim i desktop uređajima, različitim rezolucijama. Na narednoj slici je prikazan izgled početne stranice na mobilnom uređaju koji je isti kao i na desktop uređaju što i jeste suština univerzalnih aplikacija.



Prilog - Slika 10: Izgled početne stranice na mobilnim uređajima

Kao što je i prethodno naglašeno, ova aplikacija nudi mogućnost buduće nadogradnje.

