



**UNIVERZITET U SARAJEVU**  
**FAKULTET ZA SAOBRAĆAJ I**  
**KOMUNIKACIJE**



**PROJEKTNI RAD IZ PREDMETA:**  
**Razvoj Aplikacija Mobilnih Uređaja**

<b>Tema rada:</b>	<b>Mobilno naručivanje u restoranu</b>
-------------------	--

<b>Predmetni nastavnik:</b>	V. prof. dr. Asmir Butković
<b>Asistent (saradnik):</b>	
<b>Student:</b>	Zuban Amila
<b>Broj indeksa:</b>	8179
<b>Usmjerenje:</b>	KIT
<b>Godina studija:</b>	Treća (III)
<b>Rezultat rada:</b>	

Datum: 07.07.2024.

## Contents

UVOD .....	1
1. Analiza zahtjeva.....	2
1.1. Identifikacija poslovnih zahtjeva za aplikaciju .....	2
1.2. Definicija korisničkih priča i funkcionalnosti .....	2
2. Dizajn i arhitektura .....	3
3. Implementacija.....	5
3.1. Tehnički detalji implementacije .....	5
4. Dodaci .....	6
4.1. MainPage.xaml i MainPage.xaml.cs .....	6
4.2. MenuPage.xaml i MenuPage.xaml.cs.....	7
4.3. CartPage.xaml i CartPage.xaml.cs .....	10
4.4. Modeli.....	12
ZAKLJUČAK .....	14

## UVOD

Mobilna aplikacija za naručivanje u restoranu je namijenjena kako bi se proces naručivanja hrane u restoranu modernizovao i kako bi se iskustvo klijenata poboljšalo pri samom naručivanju.

U ovoj aplikaciji, korisnici, nakon što upišu broj stola na kojem sjede, mogu pretraživati razne vrste jela kroz kategorije i odmah vidjeti cijene istih, kao i njihove kratke opise i slike.

Prilikom odabira hrane koju korisnici žele, omogućeno je da dodaju u korpu samim klikom na dugme “Dodaj u korpu” koje se nalazi ispod svakog artikla, te prilikom završetka imaju mogućnost da pogledaju korpu klikom na dugme “Pogledaj korpu” koje ih vodi na CartPage dio aplikacije.

Kada korisnik odluči da pogleda korpu, može da vidi svaki artikal posebno, te njegovu cijenu i može da završi narudžbu, da ukloni neko jelo iz korpe ili da se vrati na početni meni u slučaju da želi da počne ispočetka sa naručivanjem.

Koncept ove aplikacije je osmišljen tako da umjesto fizičkog menija u restoranu klijenti, odnosno mušterije istog imaju mobilnu aplikaciju preko koje naručuju svoju hranu i samim tim ubrzavaju čitav proces.

## 1. Analiza zahtjeva

Ova aplikacija je namijenjena digitalnim upravljanju menijem i narudžbama u restoranu. Cilj ove aplikacije je bio da se poboljša iskustvo klijenata, odnosno mušterija u restoranu omogućujući jednostavno pregledanje menija, odabir i narudžbu hrane i pića putem mobilnog uređaja.

### *1.1. Identifikacija poslovnih zahtjeva za aplikaciju*

1. Digitalizacija menija – omogućiti restoranima da digitalno prikažu svoj meni korisnicima
2. Narudžba i korpa – omogućiti korisnicima pregled artikala, dodavanje u korpu i zaključivanje narudžbe
3. Ubrzanje usluge – povećati efikasnost usluživanja omogućujući bržu i precizniju narudžbu.

### *1.2. Definicija korisničkih priča i funkcionalnosti*

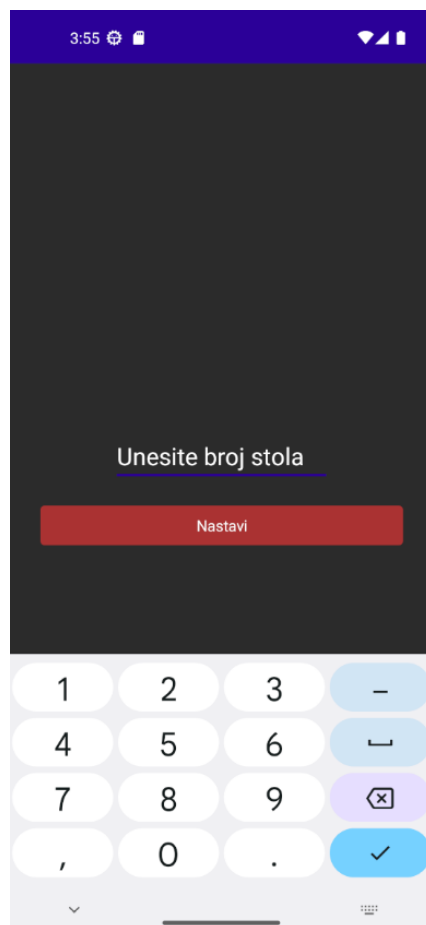
U ovoj aplikaciji, korisnik može:

1. Pregledati kategorije hrane i pića
2. Dodavati/brisati artikle iz korpe
3. Pregledati sadržaj korpe prije zaključenja narudžbe.

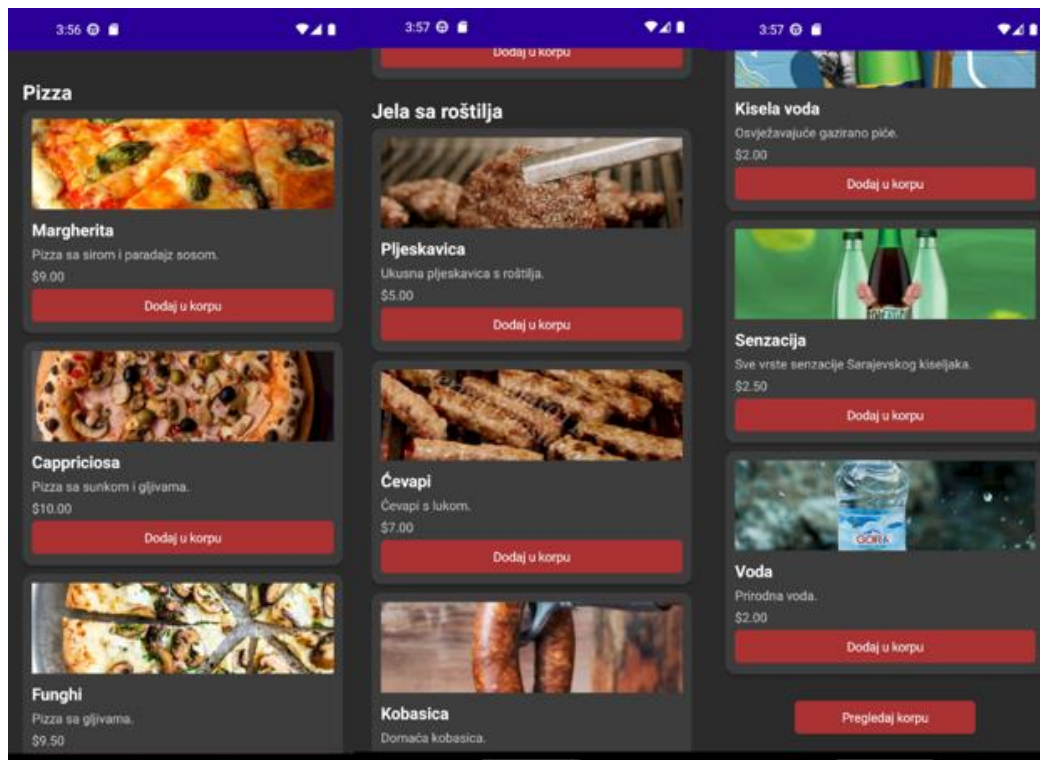
## 2. Dizajn i arhitektura

Svaka stranica ove aplikacije ima tamnosivu pozadinu, a artikli na meniju i u korpi su istaknuti svijetlosivom bojom. Tekst je bijele boje, dok su svi buttoni crvene boje.

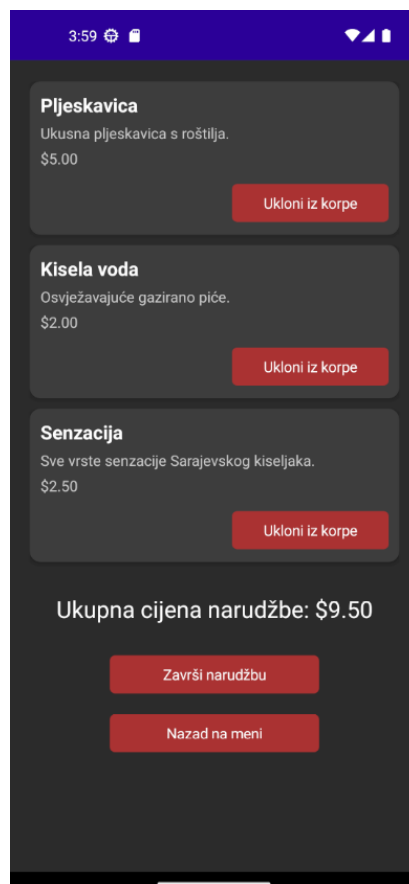
Arhitektura aplikacije organizuje kategorije na stranici menija jednu ispod druge. Svaka kategorija prikazuje artikle sa njihovim nazivom, slikom, kratkim opisom i cijenom, uz button "Dodaj u korpu". Na stranici korpe, artikli su raspoređeni jedan ispod drugog sa nazivom, kratkim opisom i cijenom. Za svaki artikl dostupno je button "Ukloni iz korpe" po potrebi. Ispod artikala nalaze se buttoni "Završi narudžbu" i "Nazad na meni".



Slika 1: Izgled glavne stranice (MainPage) aplikacije



Slika 2: Izgled menija (MenuPage)



Slika 3: Izgled korpe (CartPage)

## 3. Implementacija

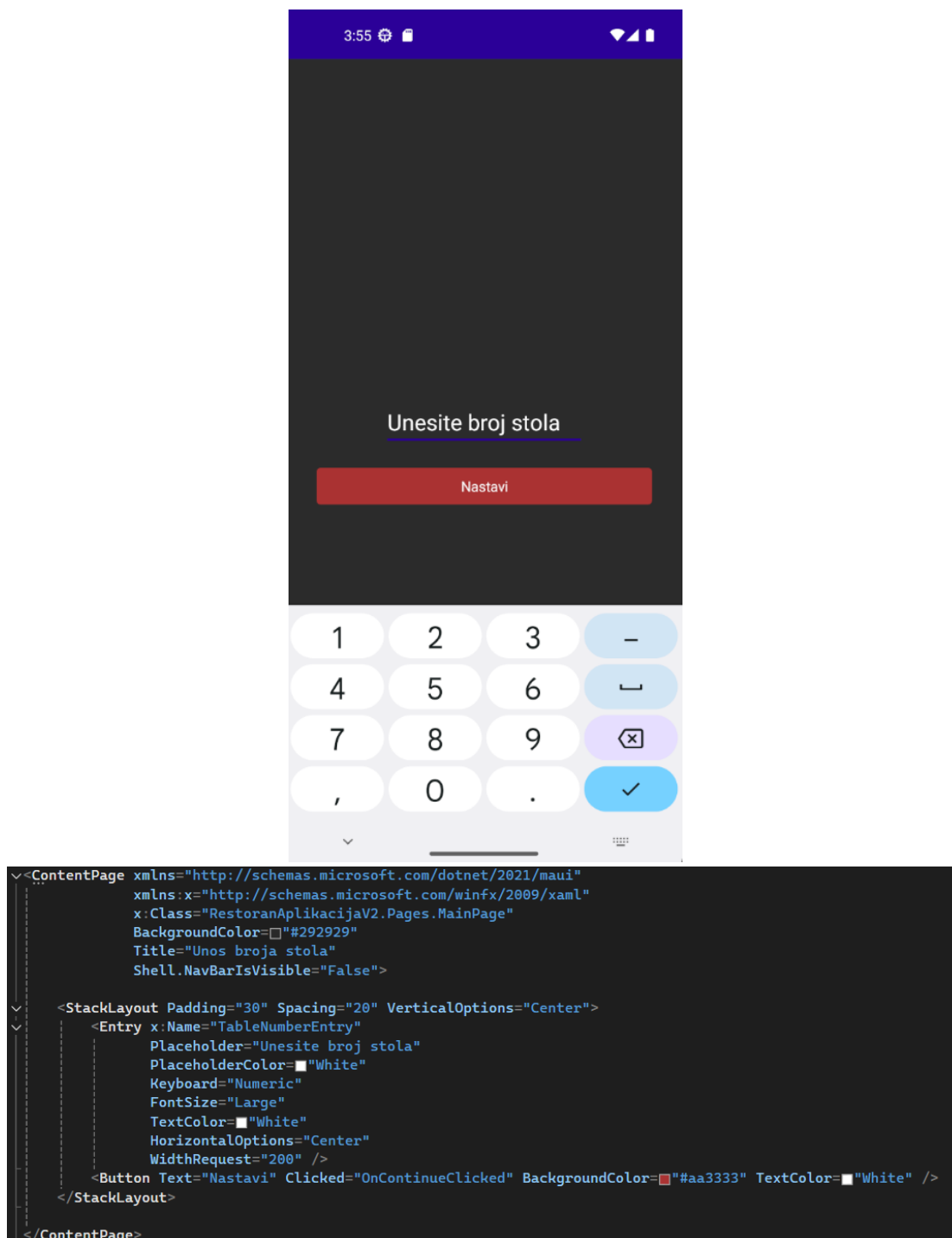
### 3.1. Tehnički detalji implementacije

Implementacija aplikacije je izvedena koristeći Microsoft MAUI platforu za razvoj mobilnih aplikacija.

1. Programski jezik – C# je korišten za implementaciju poslovne logike aplikacije.
2. Korisnički interface – XAML (eXtensible Application Markup Language) je korišten za definisanje korisničkog interface-a aplikacije.
3. Navigacija i strukturni elementi – Korišten je Shell pattern za organizaciju stranica i navigaciju unutar aplikacije. Stranice MainPage, MenuPage i CartPage implementirane su kao ContentPage u Shellu.
4. Korištenje modela podataka – Za organizaciju podataka korišteni su modelu poput 'Category' i 'Item' koji su omogućili razvijanje strukture kategorija i artikala unutar aplikacije.

## 4. Dodaci

### 4.1. MainPage.xaml i MainPage.xaml.cs



Slika 4: Izgled MainPage i MainPage.xaml kod korišten za uređivanje interface-a



```

namespace RestoranAplikacijaV2.Pages
{
    6 references
    public partial class MainPage : ContentPage
    {
        0 references
        public MainPage()
        {
            InitializeComponent();
        }

        0 references
        private async void OnContinueClicked(object sender, EventArgs e)
        {
            if (int.TryParse(TableNumberEntry.Text, out int tableNumber))
            {
                await Navigation.PushAsync(new MenuPage(tableNumber));
            }
            else
            {
                await DisplayAlert("Greška", "Molimo unesite validan broj stola.", "OK");
            }
        }
    }
}

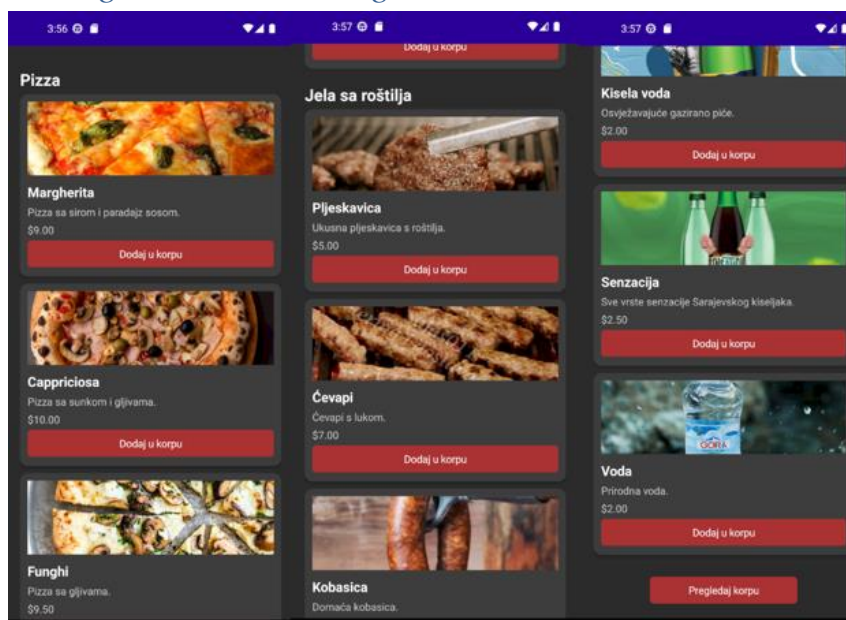
```

Slika 5: MainPage.xaml.cs kod

'MainPage.xaml' file definiše izgled početne stranice naše aplikacije (definiše izgled elemenata kao što su buttoni, slike, tekstualni elementi i ostali vizuelni sadržaji koji se prikazuju korisniku). XAML elementi su povezani s odgovarajućim funkcijama i svojstvima definisanim u 'MainPage.xaml.cs' file, što omogućava interakciju korisnika sa ovom aplikacijom.

Funkcija 'MainPage.xaml.cs' file-a je da upravlja logikom i interakcijom korisničkog interface-a definisanog u 'MainPage.xaml'. Nasljeđuje ContentPage i definiše metode koje obrađuje klik buttona i unos teksta, te navigira između stranica i druge interakcije korisnika.

## 4.2. MenuPage.xaml i MenuPage.xaml.cs



Slika 6: Izgled MenuPage

```

ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
x:Class="RestoranAplikacijaV2.Pages.MenuPage"
BackgroundColor="#292929"
Title="Meni"
Shell.NavBarIsVisible="False">
<ScrollView>
<StackLayout Padding="10" Margin="10">
<CollectionView ItemsSource="{Binding Categories}">
<CollectionView.ItemTemplate>
<DataTemplate>
<StackLayout>
<Label Text="{Binding Name}"
FontSize="Large"
TextColor="White"
FontAttributes="Bold"
Margin="0,10,0,5" />
<CollectionView ItemsSource="{Binding Items}">
<CollectionView.ItemsLayout>
<LinearItemsLayout Orientation="Vertical" />
</CollectionView.ItemsLayout>
<CollectionView.ItemTemplate>
<DataTemplate>
<Frame BackgroundColor="#3e3e3e"
CornerRadius="10"
Padding="10"
Margin="0,0,0,10"
HasShadow="True">
<StackLayout>
<Image Source="{Binding ImageSource}" Aspect="AspectFill" HeightRequest="100" Margin="0,0,0,10" />
<Label Text="{Binding Name}"
FontSize="Medium"
TextColor="White"
FontAttributes="Bold"
Margin="0,0,0,5" />
<Label Text="{Binding Description}"
FontSize="Small"
TextColor="White"
Margin="0,0,0,5" />
<Label Text="{Binding Price, StringFormat='{0:C}'}"
FontSize="Small"
TextColor="White"
Margin="0,0,0,5" />
<Button Text="Dodaj u korpu"
Clicked="OnAddToCartClicked"
CommandParameter="{Binding .}"
BackgroundColor="#aa3333"
TextColor="White"
CornerRadius="5" />
</StackLayout>
</Frame>
</DataTemplate>
</CollectionView.ItemTemplate>
</CollectionView>
</StackLayout>
</DataTemplate>
</CollectionView.ItemTemplate>
</CollectionView>
<Button Text="Pregledaj korpu"
Clicked="OnCartButtonClicked"
BackgroundColor="#aa3333"
TextColor="White"
CornerRadius="5"
WidthRequest="200"
HorizontalOptions="Center"
Margin="0,20,0,0" />
</StackLayout>
</ScrollView>
</ContentPage>

```

Slika 7: MenuPage.xaml kod

```

namespace RestoranAplikacijaV2.Pages
{
    7 references
    public partial class MenuPage : ContentPage
    {
        private int tableNumber;

        1 reference
        public List<Category> Categories { get; set; }

        1 reference
        public MenuPage(int tableNumber)
        {
            InitializeComponent();
            this.tableNumber = tableNumber;
            Categories = GetCategories();
            BindingContext = this;
        }

        1 reference
        private List<Category> GetCategories()
        {
            return new List<Category>
            {
                new Category
                {
                    Name = "Pizza",
                    Items = new List<Item>
                    {
                        new Item { Name = "Margherita", Description = "Pizza sa sirom i paradajz sosom.", Price = 9.0m, ImageSource = "Images/marga",
                        new Item { Name = "Capriciosa", Description = "Pizza sa sunkom i gljivama.", Price = 10.0m, ImageSource = "Images/kapricioz",
                        new Item { Name = "Funghi", Description = "Pizza sa gljivama.", Price = 9.5m, ImageSource = "Images/fungi.jpg" }
                    }
                },
                new Category
                {
                    Name = "Jela sa roštilja",
                    Items = new List<Item>
                    {
                        new Item { Name = "Pljeskavica", Description = "Ukusna pljeskavica s roštilja.", Price = 5.0m, ImageSource = "Images/pljeska",
                        new Item { Name = "Čevapi", Description = "Čevapi s lukom.", Price = 7.0m, ImageSource = "Images/cevapi.jpg" },
                        new Item { Name = "Kobasica", Description = "Domaća kobasica.", Price = 4.0m, ImageSource = "Images/kobasica.jpg" }
                    }
                },
                new Category
                {
                    Name = "Piće",
                    Items = new List<Item>
                    {
                        new Item { Name = "Kisela voda", Description = "Osvježavajuće gazirano piće.", Price = 2.0m, ImageSource = "Images/kisela.j",
                        new Item { Name = "Senzacija", Description = "Sve vrste senzacije Sarajevskog kiseljaka.", Price = 2.5m, ImageSource = "Ima",
                        new Item { Name = "Voda", Description = "Prirodna voda.", Price = 2.0m, ImageSource = "Images/voda.jpg" }
                    }
                }
            };
        }

        0 references
        private void OnAddToCartClicked(object sender, EventArgs e)
        {
            var button = sender as Button;
            var item = button?.BindingContext as Item;

            if (item != null)
            {
                ((App)Application.Current).AddToCart(item);
            }
        }

        0 references
        private async void OnCartButtonClicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new CartPage(((App)Application.Current).GetCartItems(), tableNumber));
        }
    }
}

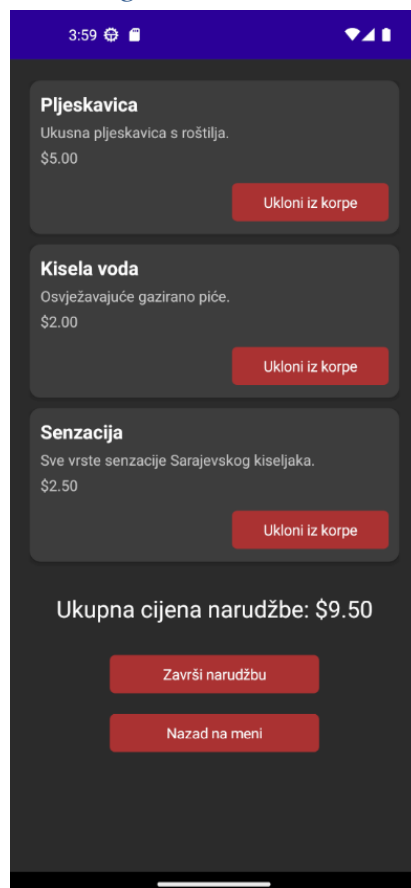
```

Slika 8: MenuPage.xaml.cs kod

'MenuPage.xaml' definiše korisnički interface stranice menija. Koristi se za prikazivanje kategorija i artikala jednog restoranskog menija i sadrži XAML kod koji opisuje raspored elemenata na stranici, kao što su slike, nazivi artikala, te njihove cijene i kratki opisi. Također, prikazuje buttone „Dodaj u korpu“ ispod svakog artikla.

'MenuPage.xaml.cs' je C# file koji implementira logiku ponašanja aplikacije i interakcije definisane u 'MenuPage.xaml'. Obrađuje događaje klikanja buttona za dodavanje artikala u korpu, te navigira korisnika na stranicu korpe za pregled klikom buttona i upravlja narudžbama.

### 4.3. *CartPage.xaml i CartPage.xaml.cs*



*Slika 9: Izgled CartPage*

```

ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
            xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
            x:Class="RestoranAplikacijaV2.Pages.CartPage"
            BackgroundColor="#292929"
            Title="Korpa"
            Shell.NavBarIsVisible="False">

    <ScrollView>
        <StackLayout Padding="10" Margin="10">
            <CollectionView ItemsSource="{Binding CartItems}">
                <CollectionView.ItemTemplate>
                    <DataTemplate>
                        <Frame BackgroundColor="#3e3e3e"
                            CornerRadius="10"
                            Padding="10"
                            Margin="0,0,0,10"
                            HasShadow="True">
                            <StackLayout>
                                <Label Text="{Binding Name}"
                                    FontSize="Medium"
                                    TextColor="White"
                                    FontAttributes="Bold"
                                    Margin="0,0,0,5" />
                                <Label Text="{Binding Description}"
                                    FontSize="Small"
                                    TextColor="White"
                                    Margin="0,0,0,5" />
                                <Label Text="{Binding Price, StringFormat='{0:C}'}"
                                    FontSize="Small"
                                    TextColor="White"
                                    Margin="0,0,0,5" />
                                <Button Text="Ukloni iz korpe"
                                    CommandParameter="{Binding .}"
                                    Clicked="OnRemoveFromCartClicked"
                                    BackgroundColor="#aa3333"
                                    BackgroundColor="#aa3333"
                                    TextColor="White"
                                    CornerRadius="5"
                                    WidthRequest="150"
                                    HorizontalOptions="End"
                                    Margin="0,10,0,0" />
                            </StackLayout>
                        </Frame>
                    </DataTemplate>
                </CollectionView.ItemTemplate>
            </CollectionView>

            <Label Text="{Binding TotalPrice, StringFormat='Ukupna cijena narudžbe: {0:C}'}"
                FontSize="Large"
                TextColor="White"
                HorizontalOptions="Center"
                Margin="0,20,0,10" />

            <Button Text="Završi narudžbu"
                Clicked="OnCheckoutButtonClicked"
                BackgroundColor="#aa3333"
                TextColor="White"
                CornerRadius="5"
                WidthRequest="200"
                HorizontalOptions="Center"
                Margin="0,20,0,0" />
            <Button Text="Nazad na meni"
                Clicked="OnBackToMenuButtonClicked"
                BackgroundColor="#aa3333"
                TextColor="White"
                CornerRadius="5"
                WidthRequest="200"
                HorizontalOptions="Center"
                Margin="0,20,0,0" />
        </StackLayout>
    </ScrollView>

```

Slika 10: CartPage.xaml kod

```

namespace RestoranAplikacijaV2.Pages
{
    7 references
    public partial class CartPage : ContentPage
    {
        3 references
        public List<Item> CartItems { get; set; }
        0 references
        public decimal TotalPrice => CartItems.Sum(item => item.Price);
        private int tableNumber;

        1 reference
        public CartPage(List<Item> cartItems, int tableNumber)
        {
            InitializeComponent();
            CartItems = cartItems;
            this.tableNumber = tableNumber;
            BindingContext = this;
        }

        0 references
        async void OnCheckoutButtonClicked(object sender, EventArgs e)
        {
            await DisplayAlert("Narudžba", $"Vaša narudžba je zaprimljena za stol broj {tableNumber}.", "OK");
        }

        0 references
        async void OnBackToMenuButtonClicked(object sender, EventArgs e)
        {
            try
            {
                ((App)Application.Current).ClearCart();
            }
            catch (Exception ex)
            {
                await DisplayAlert("Greška", $"Došlo je do greške: {ex.Message}", "OK");
            }
        }

        0 references
        async void OnRemoveFromCartClicked(object sender, EventArgs e)
        {
            var button = sender as Button;
            var item = button?.CommandParameter as Item;

            if (item != null)
            {
                CartItems.Remove(item);
                BindingContext = null;
                BindingContext = this;
            }
        }
    }
}

```

Slika 11: CartPage.xaml.cs kod

'CartPage.xaml' definiše korisnički interface korpe aplikacije. Koristi se za prikazivanje artikala koji su dodani u korpu za narudžbu. Sadrži XAML kod koji opisuje raspored elemenata na stranici, kao što su nazivi artikala, njihovi kratki opisi i cijene, te buttone za uklanjanje artikla iz korpe, za završavanje narudžbe i za vraćanje na meni.

'CartPage.xaml.cs' je C# datoteka koja implementira logiku ponašanja ove stranice. U ovom kodu možemo vidjeti logiku iza buttona za uklanjanje artikala iz korpe, za vraćanje na meni i za završavanje narudžbe.

#### 4.4. Modeli

Modeli 'Category' i 'Item' služe za reprezentaciju podataka o kategorijama i stavama u ovoj aplikaciji za naručivanje hrane.

Model 'Category' predstavlja kategoriju hrane u restoranu. 'Name' predstavlja kategorije hrane (predjelo, glavno jelo i desert) a 'Items' predstavlja listu stavki koje pripadaju svakoj kategoriji.

Model 'Item' predstavlja pojedinu stavku hrane u restoranu. Svaka stavka ima svoje imeime ('Name'), cijenu ('Price') u decimalnom formatu, kratki opis ('Description') i njegovu sliku ('ImageSource').

```

namespace RestoranAplikacijaV2.Models
{
    7 references
    public class Category
    {
        3 references
        public string? Name { get; set; }
        4 references
        public List<Item>? Items { get; set; }
    }
}

```

*Slika 12: Model 'Category'*

```

namespace RestoranAplikacijaV2.Models
{
    25 references
    public class Item
    {
        10 references
        public string? Name { get; set; }
        11 references
        public decimal Price { get; set; }
        9 references
        public string? Description { get; set; }
        9 references
        public string? ImageSource { get; set; }
    }
}

```

*Slika 13: Model 'Item'*

## ZAKLJUČAK

Ova aplikacija za naručivanje u restoranu je razvijena u .NET MAUI okruženju kako bi omogućila korisnicima interaktivno pregledanje restoranskog menija, dodavanje stavki u korpu i izvršavanje narudžbe. Temelji se na Shell objektnom modelu za organizaciju glavnih dijelova aplikacije, pri čemu ShellContent koristi se za prikaz različitih stranica poput MainPage, MenuPage i CartPage, čime olakšava navigaciju korisnicima.

MainPage korisniku omogućava jednostavan unos broja stola, te dalji prelazak na MenuPage. MenuPage prikazuje kategorije hrane pomoću 'CollectionView' komponente, gdje svaka kategorija sadrži listu stavki prikazanih u okvirima. To omogućava korisnicima da odaberu željenu stavku i dodaju je u korpu. CartPage prikazuje trenutni sadržaj korpe s dodanim stavkama, omogućujući korisnicima pregled svih stavki, uklanjanje neke stavke iz korpe, prikaz ukupne cijene narudžbe te opciju za završetak narudžbe ili povratak na glavni meni.

Implementacija asinhronih metoda osigurava fluidno iskustvo korisnika prilikom interakcije s aplikacijom, posebno prilikom dodavanja stavki u korpu i procesuiranja narudžbi. Data Binding u XAML-u omogućava dinamičko prikazivanje podataka o kategorijama i stavkama, što olakšava ažuriranje sučelja prema promjenama u sadržaju.

Ova aplikacija predstavlja moderni pristup razvoju mobilnih aplikacija koristeći .NET MAUI, s intuitivnim korisničkim sučeljem, jednostavnim upravljanjem i optimiziranim korištenjem asinhronih operacija. Pruža korisnicima jednostavan način za pregled i naručivanje hrane unutar restorana, uz potencijal za daljnje proširenje i poboljšanja. Ovi elementi čine aplikaciju prikladnom za stvarne primjene u restoranskom poslovanju.