University of Zakho

Faculty of Science

Computer Department

Mobile Application

MiniProject : [Restaurant application]

Prepared by:

Dleen Ameen Hassan

Gazi Salah Ramadan

Sidra Abdulsatar Ali

Jin Hishyar Mohamad salim

Group :B

superviser:

Yousif Arshak

Contents

What is Xamarin.forms?

Xamarin.Forms is a feature of Xamarin. Xamarin is the popular mobile development framework which extends the .Net Development Platform with tools and libraries for building mobile apps. Xamarin.Forms is an open-source, cross-platform framework acquired by Microsoft for building Android, iOS, and windows app with .NET from a single shared codebase. We use Xamarin. Forms built-in Pages, layouts, and controls to build and design mobile apps from a single API that is highly extensible. Subclass any control to customize the behavior or to define our controls, layouts, pages, and cells to make our apps pixel perfect.

How does Xamarin.Forms work?

A Xamarin.Forms application is designed similarly as a traditional cross-platform application. Shared code is usually placed in a .NET Standard library and platform-specific applications consume the shared code. Xamarin.Forms applications have a single class named App that instantiates the application on each platform. The AppShell class defines the visual hierarchy of the application. Shell uses this visual hierarchy to produce the user interface of the app.

At runtime, Xamarin.Forms uses platform renderers to convert the cross-platform UI elements into native controls on Android, iOS and Windows devices. It enables the developer to give the native look, feel and user experience to the application while using the feature of code-sharing across platforms.

What are the benefits of Xamarin.Forms?

It allows developers to create native UI layout and designs across platforms using a single shared codebase which reduces the time and money deployed in the development process. It allows developers to share code, test and business logic of app across multiple platforms. It reduces the overhead of managing different codebases for app due to separate code for different platforms .It's been claimed that while Xamarin enables 70% code sharing, Xamarin.Forms allows up to 95% code sharing. It's best for apps where code sharing is more important than custom UI. One team reported that with Xamarin.Forms they had only 135 lines of platform-specific code out of 3744 lines (96% reuse). Earlier with Xamarin the team had achieved 80% code reuse .PGS Software implemented European airline Volotea's mobile app in Xamarin.Forms. With Forms, consistent branding and styling were achieved across iOS and Android. With Forms, 70% of custom renderers were removed.

What are the key building blocks of Xamarin.Forms?

The key building blocks of a Xamarin.Forms application are as follows:

- **Xamarin.Essentials**: Essentials provides a single cross-platform API that can work with any Xamarin.Forms, Android, iOS, or UWP application accessible from shared code regardless of how the user interface is created.
- **eXtensible Application Markup Language (XAML)**: XAML is used for instantiating and initializing objects and organizing those objects in a hierarchical way. In XAML, developers can use all the Xamarin.Forms views, layouts, pages as well as custom classes to create the UI. It is very brief and readable than the equivalent code.
- **Behaviors**: Behaviors allows to add functionality to the user interface controls without extending to any other class. The functionality is already implemented in the behavior class so developers only need to attach it to the controls. They are created by deriving from the Behavior or Behavior<T> class.
- **Data binding**: It is used to keep the objects synchronized so that changes in one object's property are automatically reflected in other related object's property. It is an important part of the Model-View-ViewModel (MVVM) application architecture.

What are platform-specific features and how to create them?

A platform may have features and functionalities that doesn't exist on other platforms. For instance, Android has a functionality of fast scrolling in a list view but iOS doesn't have the same. Xamarin.Forms platform-specifics allow developers to utilize such platform-specific functionality that is only available on a specific platform without creating custom renderers or effects. There are some built-in platform-specifics in Xamarin.Forms, developers can use them or create the platform-specifics by themselves. Below are the steps for creating a platform-specific :

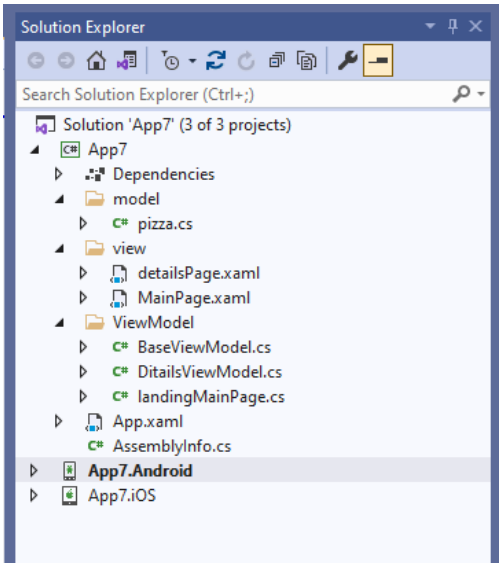
- Implement the specific functionality as an Effect.
- Create a platform-specific class that will use the Effect.
- Make an attached property in the platform-specific class to allow the platform-specific to be used through XAML.
- Implement extension methods in the platform-specific class, to allow the platform-specific to be used through a fluent code API.
- Set the effect implementation to be applied only when the platform-specific has been invoked on the same platform as the Effect.

Using an Effect as a platform-specific makes the effect easily consumable through XAML and a fluent code API.

Design and implementation of our application:

First of all our application is about pizza's restaurants and in this app it show us different type of pizza with their name and price, to create this app first we create three folders Called (model , view and viewModel) and inside each of them we create files for different class.

Our folders and files:



Pizza class:

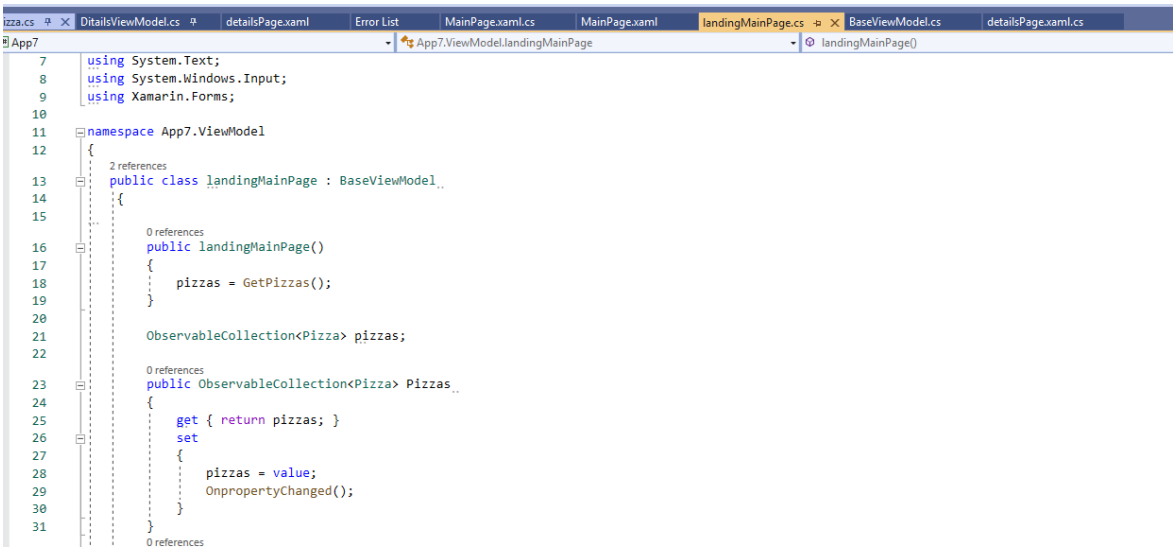
This class it contains information about pizza such as name ,price and image .

```
namespace App7.model
{
    14 references
    public class Pizza : landingMainPage
    {
        6 references
        public string Name { get; set; }
        6 references
        public float Price { get; set; }
        6 references
        public string Image { get; set; }

    }
}
```

LandingviewModel.cs:

In this class we create an object for who wants to get our products, also we have selection command .



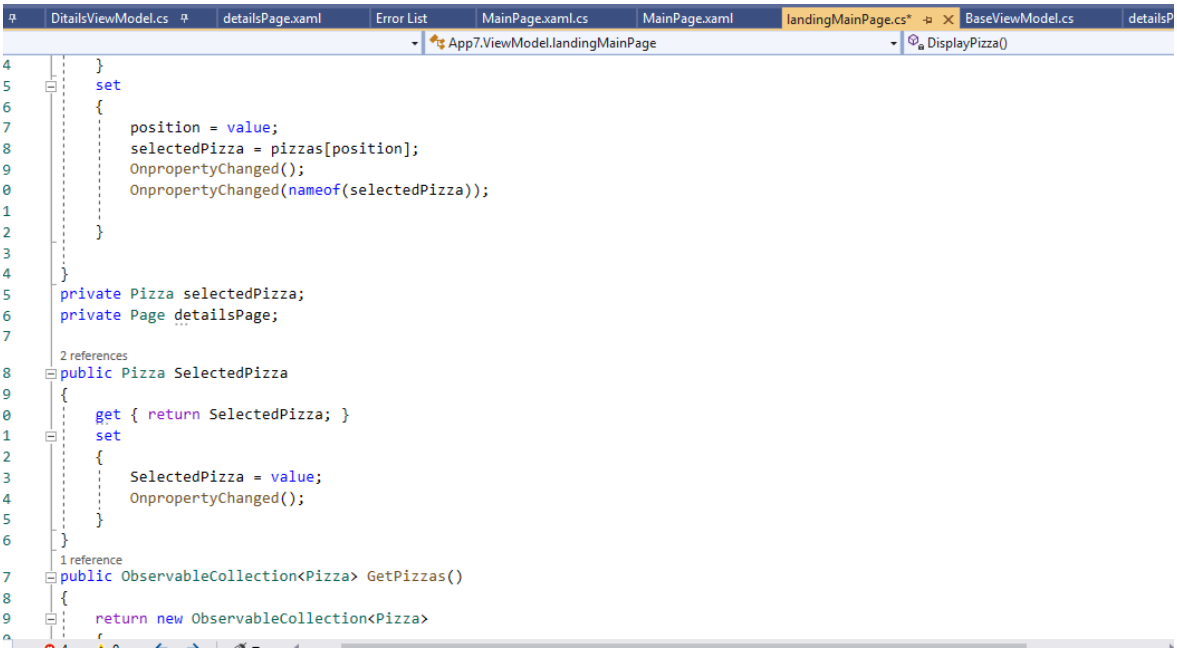
```
using System.Text;
using System.Windows.Input;
using Xamarin.Forms;

namespace App7.ViewModel
{
    2 references
    public class landingMainPage : BaseViewModel
    {

        0 references
        public landingMainPage()
        {
            pizzas = GetPizzas();
        }

        ObservableCollection<Pizza> pizzas;

        0 references
        public ObservableCollection<Pizza> Pizzas
        {
            get { return pizzas; }
            set
            {
                pizzas = value;
                OnpropertyChanged();
            }
        }
        0 references
```

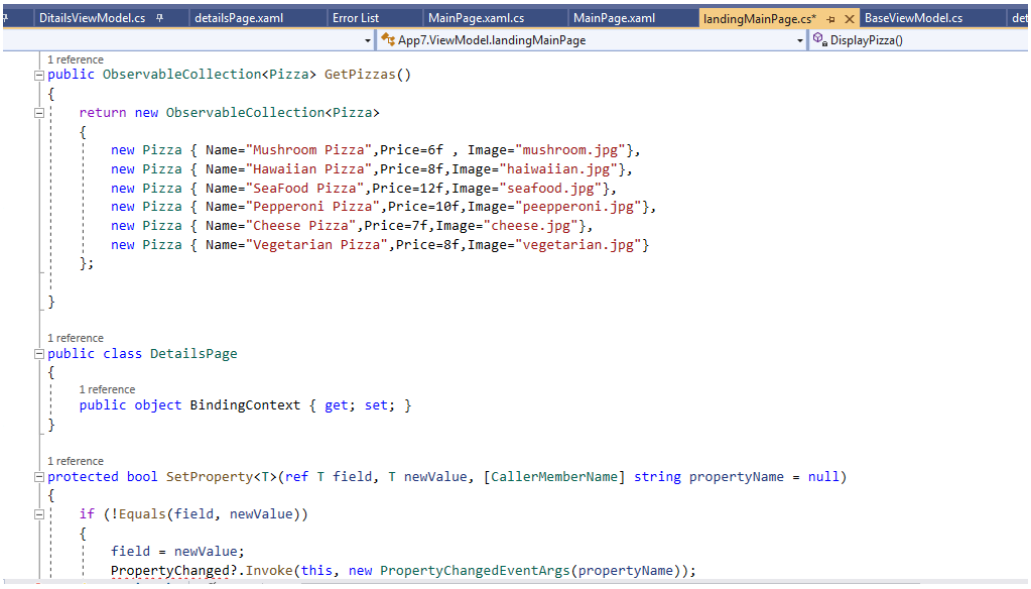


```
        }
        set
        {
            position = value;
            selectedPizza = pizzas[position];
            OnpropertyChanged();
            OnpropertyChanged(nameof(selectedPizza));


        }

    }
    private Pizza selectedPizza;
    private Page detailsPage;

    2 references
    public Pizza SelectedPizza
    {
        get { return SelectedPizza; }
        set
        {
            SelectedPizza = value;
            OnpropertyChanged();
        }
    }
    1 reference
    public ObservableCollection<Pizza> GetPizzas()
    {
        return new ObservableCollection<Pizza>
```



BaseviewModel.cs :

Mainpage.xaml:

This is desing code for our application we use different task such as (Grid, stackLayout ,Image,….ect) and other task for showing different thing like background color and name of our restaurant and text color and other things.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:vm="clr-namespace:App7.ViewModel"
               NavigationPage.HasNavigationBar="False"
             x:Class="App7.MainPage">
    <ContentPage.BindingContext>
        <vm:LandingMainPage/>
    </ContentPage.BindingContext>


    <Grid Padding="30" BackgroundColor="Black" HorizontalOptions="FillAndExpand" VerticalOptions="FillAndExpand">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <StackLayout Spacing="20" HorizontalOptions="Start" VerticalOptions="Center">
            <Image Source="searcg.jpg" Aspect="AspectFit" WidthRequest="18" HeightRequest="18"/>
            <Image Source="user.png" Aspect="AspectFit" WidthRequest="18" HeightRequest="18"/>
        </StackLayout>
        <StackLayout Spacing="0" HorizontalOptions="Center" VerticalOptions="Center">
            <Label Text="PIZZA" TextColor="White" FontFamily="ThemeFont" FontSize="50" HorizontalOptions="Center"/>
            <Label Text="DLEVERY" Margin="0,-5,0,0" TextColor="white" Opacity="0.5" FontFamily="LightFont" FontSize="25" HorizontalOptions="Center"/>
        </StackLayout>
```

```
        <CollectionView x:Name="pizzaList" Margin="0,40,0,0" Grid.Row="1" ItemsSource="{Binding Pizza}" SelectionMode="Single"
                  SelectedItem="{Binding Pizza}" VerticalScrollBarVisibility="Never"
                  SelectionChangedCommand="{Binding SelectionCommand }">
            <CollectionView.ItemsLayout>
                <GridItemsLayout Orientation="Vertical" VerticalItemSpacing="20" HorizontalItemSpacing="20" Span="2"/>
            </CollectionView.ItemsLayout>
            <CollectionView.ItemTemplate>
                <DataTemplate>
                    <Grid Padding="10" Background="gray" WidthRequest="150" HeightRequest="165">
                        <Grid.RowDefinitions>
                            <RowDefinition Height="*"/>
                            <RowDefinition Height="auto"/>
                        </Grid.RowDefinitions>
                        <Image Aspect="AspectFill" Source="{Binding Image}"
                               Margin="10" WidthRequest="125" HeightRequest="110"
                               HorizontalOptions="Center" VerticalOptions="Center"/>
                        <Label Grid.Row="1" Text="{Binding Name}" TextColor="Yellow"
                               FontFamily="ThemeFont" VerticalOptions="End" HorizontalOptions="Start"/>
                        <Label Grid.Row="1" Text="{Binding price,StringFormat='${0}'}" TextColor="White"
                               FontFamily="ThemeFont" VerticalOptions="End" HorizontalOptions="End"/>
                    </Grid>

                </DataTemplate>
            </CollectionView.ItemTemplate>
        </CollectionView>
    </Grid>
</ContentPage>
```

Detailspage.xaml:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml" xmlns:viewmodel="clr-namespace:App7.ViewModel"
             x:DataType="viewmodel:landingMainPage"
             NavigationPage.HasNavigationBar="True">

    <Grid Padding="30" BackgroundColor="Black" HorizontalOptions="FillAndExpand" VerticalOptions="FillAndExpand">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <Grid Padding="0.10" HorizontalOptions="FillAndExpand" VerticalOptions="FillAndExpand">
            <StackLayout Spacing="20" HorizontalOptions="Start" VerticalOptions="Center">
                <Image Source="searcg.jpg" Aspect="AspectFit" WidthRequest="18" HeightRequest="15"/>
                <Image Source="user.png" Aspect="AspectFit" WidthRequest="18" HeightRequest="15"/>
            </StackLayout>
            <StackLayout Spacing="0" HorizontalOptions="End" VerticalOptions="Start">
                <Label Text="Pizza" TextColor="White" FontFamily="ThemeFont" FontSize="25" HorizontalOptions="End"/>
                <Label Text="SPOT"  TextColor="white" Opacity="0.5" FontFamily="LightFont" FontSize="12" HorizontalOptions="End"/>
            </StackLayout>
        </Grid>
            <ScrollView Grid.Row="1" HorizontalOptions="FillAndExpand">
                <Grid RowSpacing="50" HorizontalOptions="FillAndExpand">
                    <Grid.RowDefinitions>
                        <RowDefinition Height="auto"/>
                        <RowDefinition Height="auto"/>
                        <RowDefinition Height="auto"/>
                        <RowDefinition Height="auto"/>
                    </Grid.RowDefinitions>
                    <StackLayout Spacing="0" HorizontalOptions="Center" VerticalOptions="Start">
                        <Label Text="{Binding SelectedPizza.Price,StringFormat='${0}'}" TextColor="#f9b522"
                               FontFamily="themefont" FontSize="50" HorizontalOptions="Center"/>
                    </StackLayout>
                </Grid>
            </ScrollView>
            <CarouselView Grid.Row="1" ItemsSource="{Binding Pizzas}" HeightRequest="260" HorizontalOptions="FillAndExpand"
                          CurrentItem="{Binding SelectedPizza}" IsScrollAnimated="True" IsSwipeEnabled="True">
                <CarouselView.ItemTemplate>
                    <DataTemplate>
                        <Grid HorizontalOptions="FillAndExpand">

                            <Label Text="{Binding Name}" TextColor="White" FontSize="105" FontFamily="themefont"
                                   HorizontalOptions="Center"/>
                            <Image Source="{Binding Image}" Aspect="Fill" Margin="0,-50,0,0" HeightRequest="210"
                             WidthRequest="235" HorizontalOptions="Center" VerticalOptions="End"/>

                        </Grid>
                    </DataTemplate>
                </CarouselView.ItemTemplate>


            </CarouselView>
    </Grid>
</ContentPage>
```

This is the result of our code it show like this: