Lab 8 – MVVM

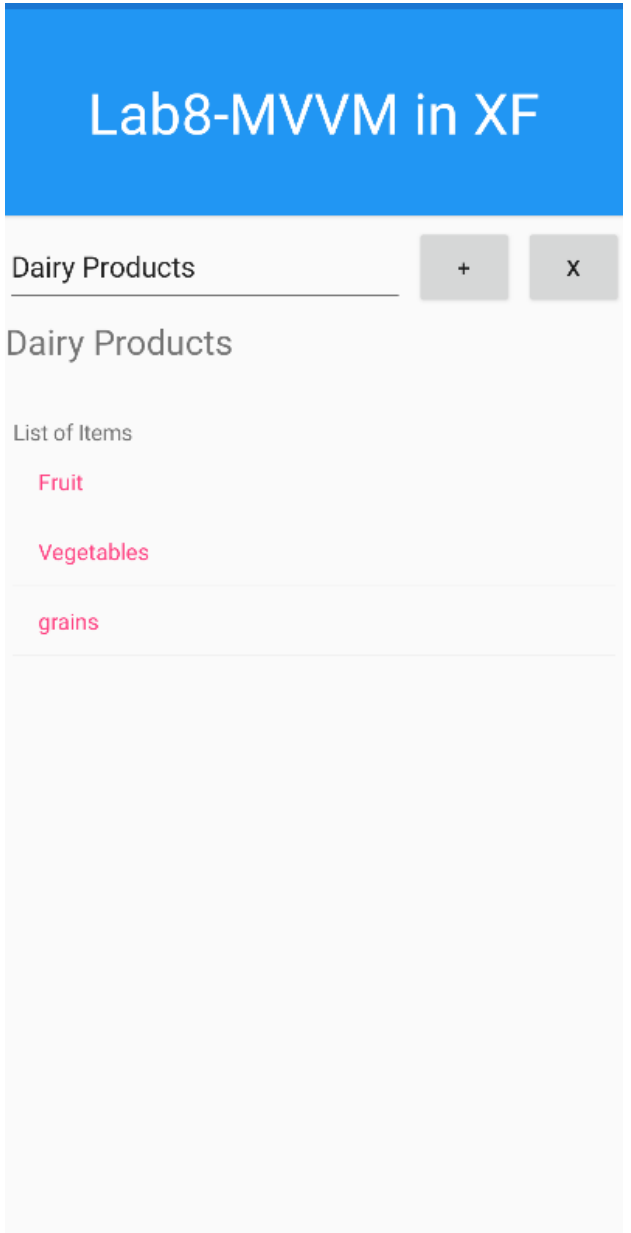
Mr. Yousif Garabet Arshak
Computer Science Department
University of Zakho
yousif.arshak@uoz.edu.krd

OUTLINES

- MVVM Example

MVVM Example

We are going to create this application using MVVM



View - Design Application with 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"
             x:Class="Lab8_MVVM.MainPage">
    <Grid RowDefinitions="1*,0.4*,0.4*,4*">
        <Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0" Grid.Row="0">
            <Label Text="Lab8-MVVM in XF" VerticalOptions="Center" HorizontalTextAlignment="Center"
                   TextColor="White" FontSize="36"/>
        </Frame>
        <Grid ColumnDefinitions="4*,*,*" Grid.Row="1">
            <Entry Placeholder="Type an item name" Grid.Column="0" Text="{Binding txtSearch}"/>
            <Button Text="+" Grid.Column="1" Command="{Binding searchCommand}"/>
            <Button Text="X" Grid.Column="2" Command="{Binding clearCommand}"/>
        </Grid>
        <Label Text="{Binding txtSearch}" Grid.Row="2" FontSize="Large"/>
        <ListView Margin="5" Header="List of Items" Grid.Row="3" ItemsSource="{Binding lstItems}">
        </ListView>
    </Grid>
</ContentPage>
```

ViewModel – Create viewmodel class

```
2 references
public class MainPageViewModel: INotifyPropertyChanged
{
    string textSearch;
    2 references
    public string txtSearch { get=>textSearch;
        set
        {
            textSearch = value;
            var args = new PropertyChangedEventArgs(nameof(txtSearch));
            PropertyChanged?.Invoke(this, args);
        } }
    3 references
    public ObservableCollection<string> lstItems { get; set; }
    1 reference
    public Command searchCommand { get; set; }
    1 reference
    public Command clearCommand { get; set; }
```

Add Constructor to the viewmodel class

```
public MainPageViewModel()
{
    lstItems = new ObservableCollection<string>();
    searchCommand = new Command(() => // using labda function to items to the lstItems
    {
        lstItems.Add(txtSearch);
    });
    clearCommand = new Command(clear); // calling clear funtion
}
```

Add function and implement event

```
    void clear() // class to clear items from lstItems
    {
        lstItems.Clear();
    }
    public event PropertyChangedEventHandler PropertyChanged; // implementing PropertyChangedEventHandler event
    0 references
    public void OnPropertyChanged(string propertyName) // we use this function to tell view to update itself
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }

}
```

Add our viewmodel to the MainPage

```
5 references
public partial class MainPage : ContentPage
{
    1 reference
    public MainPage()
    {
        InitializeComponent();
        // this will inform MainPageViewModel properties and commands to the view of MainPage
        BindingContext = new MainPageViewModel();
    }
}
```

Excercise

- Create Fruit app to show list of Fruits and each fruit has (Picture, Name) in listview and add (Add, Delete, Update) items to the listview using SQLite & MVVM