Lab 6 – CollectionView + Database

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

Outlines

- **CollectionView**
- **Text File**
- **SQLite**

CollectionView Introduction

- CollectionView is a view for presenting lists of data using different layout specifications. It aims to provide a more flexible, and performant alternative to ListView.

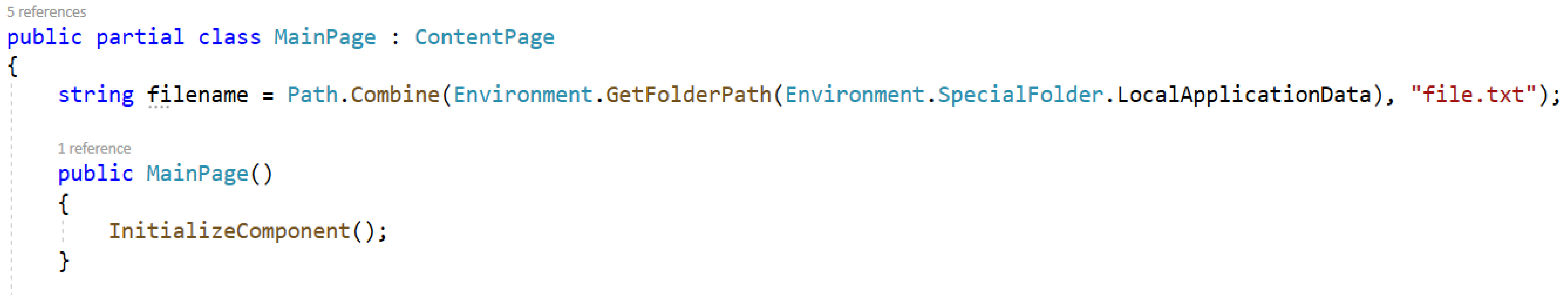
CollectionView and ListView differences

While the CollectionView and ListView APIs are similar, there are some notable differences:

- CollectionView has a flexible layout model, which allows data to be presented vertically or horizontally, in a list or a grid.
- CollectionView supports single and multiple selection.
- CollectionView has no concept of cells. Instead, a data template is used to define the appearance of each item of data in the list.
- CollectionView automatically utilizes the virtualization provided by the underlying native controls.
- CollectionView reduces the API surface of ListView. Many properties and events from ListView are not present in CollectionView.
- CollectionView does not include built-in separators.
- CollectionView will throw an exception if its ItemsSource is updated off the UI thread.

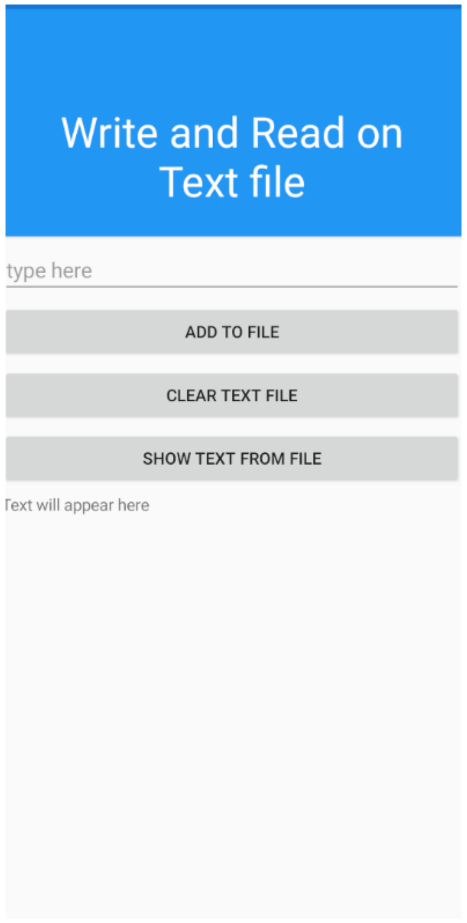
How to use Text File in your application

First Specify the path of the file in MainPage Class



```
5 references
public partial class MainPage : ContentPage
{
    string filename = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), "file.txt");

    1 reference
    public MainPage()
    {
        InitializeComponent();
    }
```

Lets make this application to read and write on Text file



Prepare your XAML Main page

```
<StackLayout>
    <Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0">
        <Label Text="Write and Read on Text file"
   HorizontalTextAlignment="Center" TextColor="White" FontSize="36"/>
    </Frame>
    <Entry Placeholder="type here" x:Name="txtfile"/>
    <Button Text="Add to File" x:Name="btnFile" Clicked="btnFile_Clicked"/>
    <Button Text="Clear Text file" x:Name="btnClear"
   Clicked="btnClear_Clicked"/>
    <Button Text="Show text from File" x:Name="btnRead"
   Clicked="btnRead_Clicked"/>
    <Label x:Name="fileView" Text="Text will appear here"/>
</StackLayout>
```

Write Your Button code

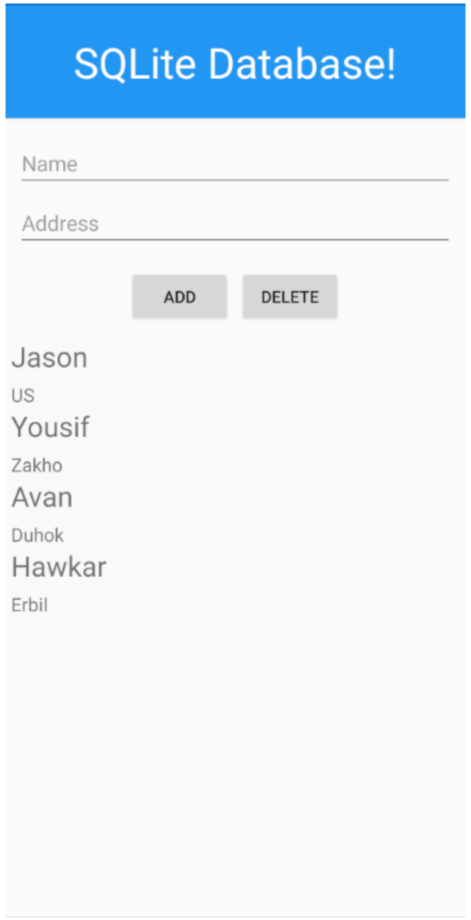
```
private void btnFile_Clicked(object sender, EventArgs e)
{
    // Check if the file is exist or not
    if (File.Exists(filename))
    {
        File.AppendAllText(filename, txtfile.Text);
    }
    else
    {
        File.Create(filename); // Create your file
        File.AppendAllText(filename, txtfile.Text); // Apped Text to the file
    }
}
```

```
private void btnRead_Clicked(object sender, EventArgs e)
{
    fileView.Text = File.ReadAllText(filename); // read text from text file
}
```

```
private void btnClear_Clicked(object sender, EventArgs e)
{
    if (File.Exists(filename)) // if file exist
    {
        File.WriteAllText(filename, ""); // Make text file empty
    }
    else //if file doesn't exisit
    {
        DisplayAlert("Attention", "There no file in your CellPhone", "OK Thanks");
    }
}
```

Form More Inform about Handling file go to bellow link
[File Handling in Xamarin.Forms - Xamarin | Microsoft Docs]

Let’s create an app and connect it with SQLite Database

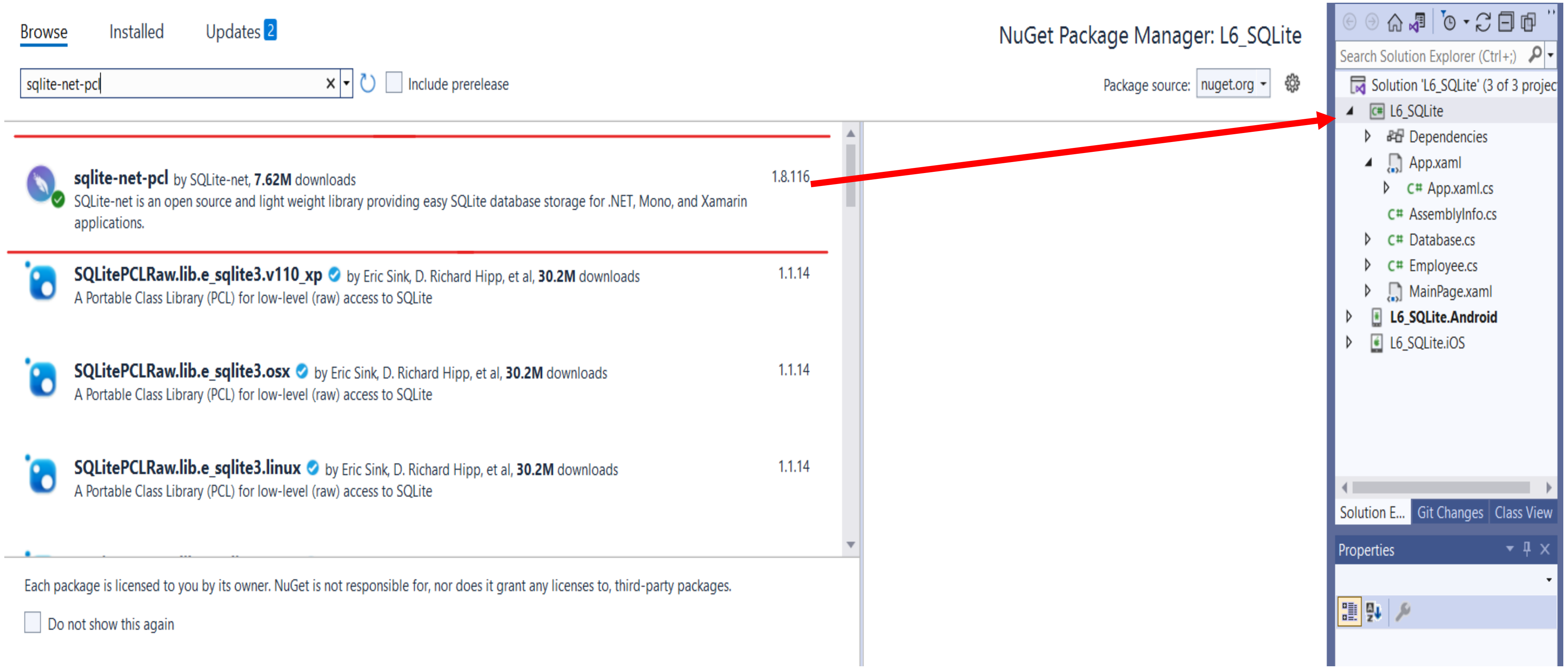

SQLite Database!

Name

Address

Jason
US
Yousif
Zakho
Avan
Duhok
Hawkar
Erbil

Add required library from Nuget Package Manager to your Xamarin Form App



Browse Installed Updates 2

NuGet Package Manager: L6_SQLite

sqlite-net-pcl

Include prerelease

Package source: nuget.org

sqlite-net-pcl by SQLite-net, **7.62M** downloads 1.8.116
SQLite-net is an open source and light weight library providing easy SQLite database storage for .NET, Mono, and Xamarin applications.

SQLitePCLRaw.lib.e_sqlite3.v110_xp by Eric Sink, D. Richard Hipp, et al, **30.2M** downloads 1.1.14
A Portable Class Library (PCL) for low-level (raw) access to SQLite

SQLitePCLRaw.lib.e_sqlite3.osx by Eric Sink, D. Richard Hipp, et al, **30.2M** downloads 1.1.14
A Portable Class Library (PCL) for low-level (raw) access to SQLite

SQLitePCLRaw.lib.e_sqlite3.linux by Eric Sink, D. Richard Hipp, et al, **30.2M** downloads 1.1.14
A Portable Class Library (PCL) for low-level (raw) access to SQLite

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

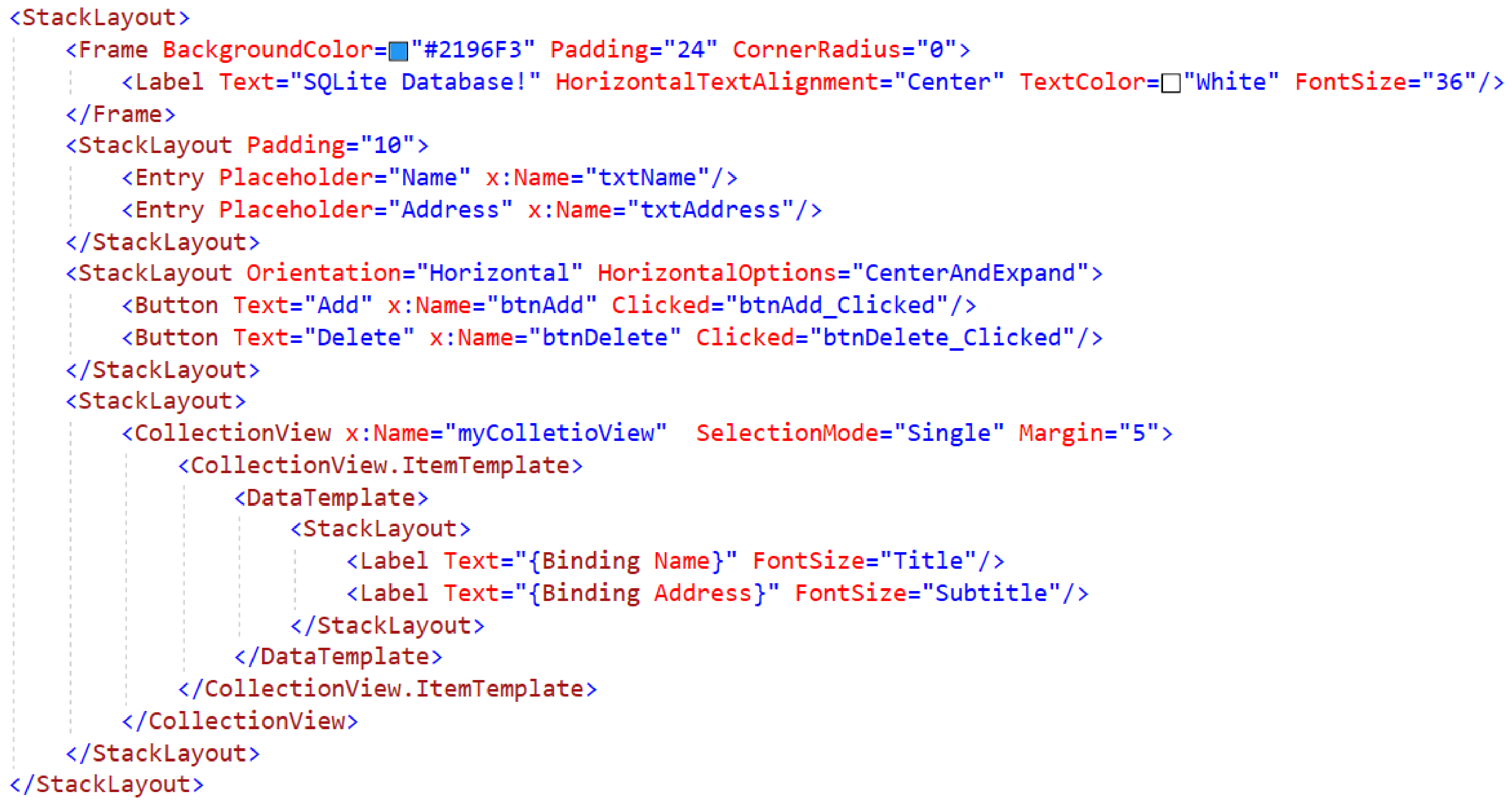
Search Solution Explorer (Ctrl+;)

```
Solution 'L6_SQLite' (3 of 3 projec
  L6_SQLite
    Dependencies
    App.xaml
      App.xaml.cs
    AssemblyInfo.cs
    Database.cs
    Employee.cs
    MainPage.xaml
  L6_SQLite.Android
  L6_SQLite.iOS
```

Solution E... Git Changes Class View

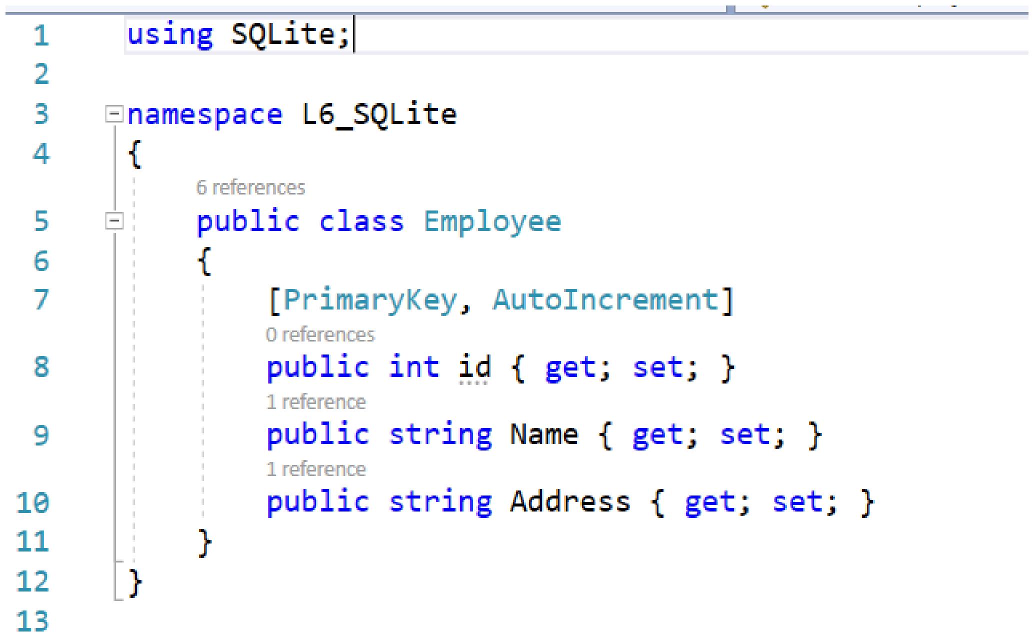
Properties

XAML page



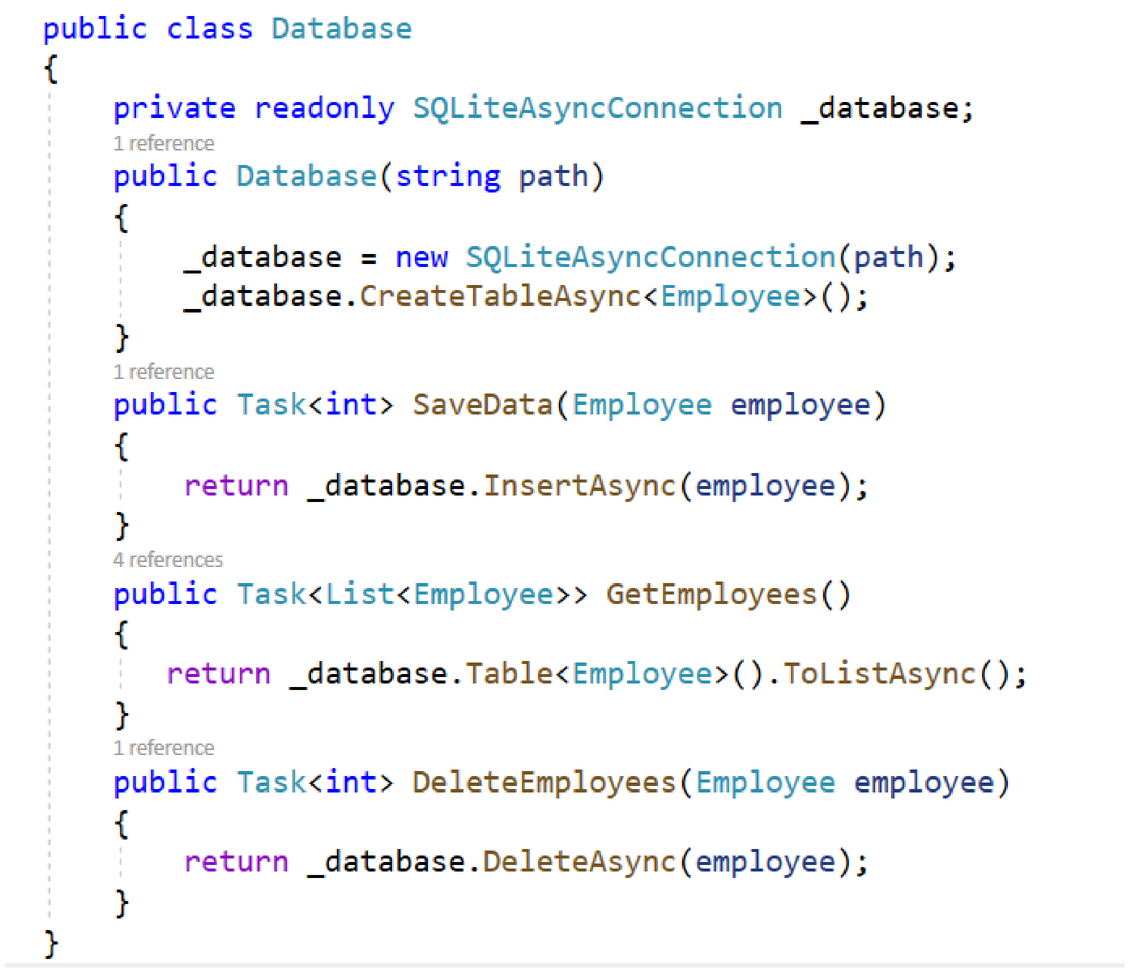
```
<StackLayout>
    <Frame BackgroundColor=🟦"#2196F3" Padding="24" CornerRadius="0">
        <Label Text="SQLite Database!" HorizontalTextAlignment="Center" TextColor=⬜"White" FontSize="36"/>
    </Frame>
    <StackLayout Padding="10">
        <Entry Placeholder="Name" x:Name="txtName"/>
        <Entry Placeholder="Address" x:Name="txtAddress"/>
    </StackLayout>
    <StackLayout Orientation="Horizontal" HorizontalOptions="CenterAndExpand">
        <Button Text="Add" x:Name="btnAdd" Clicked="btnAdd_Clicked"/>
        <Button Text="Delete" x:Name="btnDelete" Clicked="btnDelete_Clicked"/>
    </StackLayout>
    <StackLayout>
        <CollectionView x:Name="myColletioView"  SelectionMode="Single" Margin="5">
            <CollectionView.ItemTemplate>
                <DataTemplate>
                    <StackLayout>
                        <Label Text="{Binding Name}" FontSize="Title"/>
                        <Label Text="{Binding Address}" FontSize="Subtitle"/>
                    </StackLayout>
                </DataTemplate>
            </CollectionView.ItemTemplate>
        </CollectionView>
    </StackLayout>
</StackLayout>
```

Create your model in your project



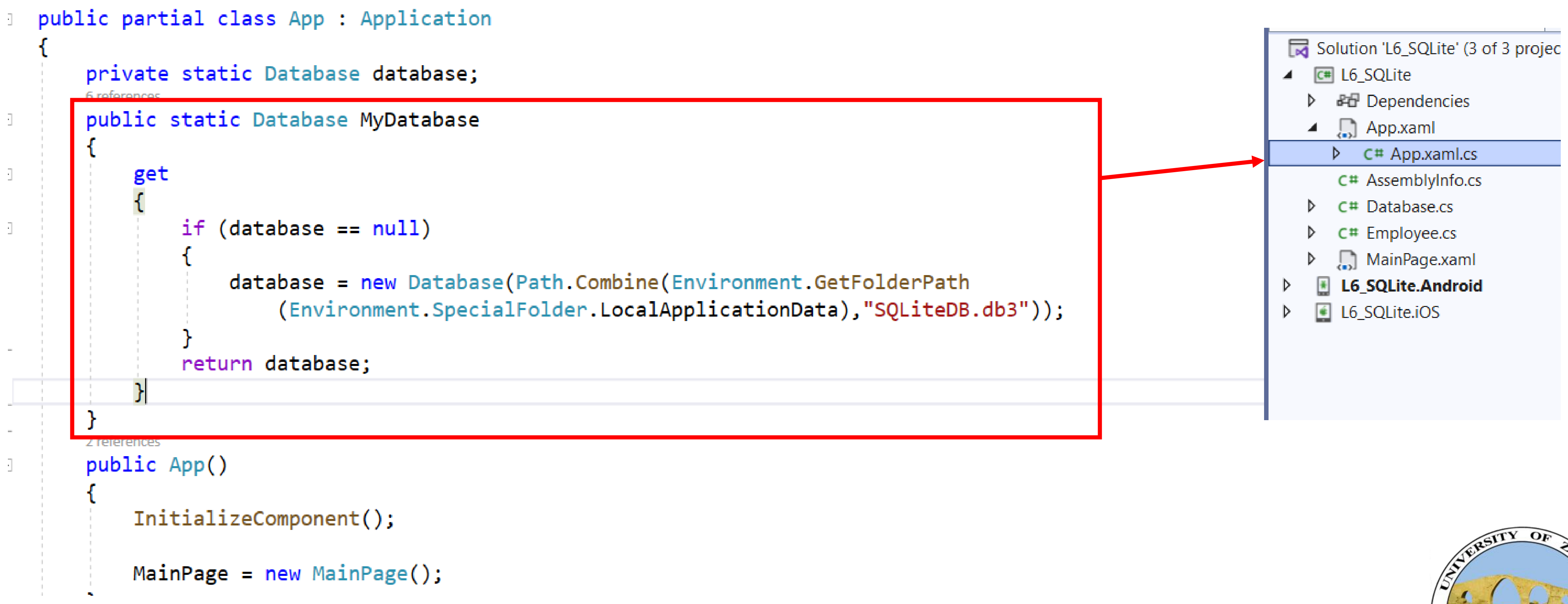
```
using SQLite;

namespace L6_SQLite
{
    6 references
    public class Employee
    {
        [PrimaryKey, AutoIncrement]
        0 references
        public int id { get; set; }
        1 reference
        public string Name { get; set; }
        1 reference
        public string Address { get; set; }
    }
}
```

Create Database services Class



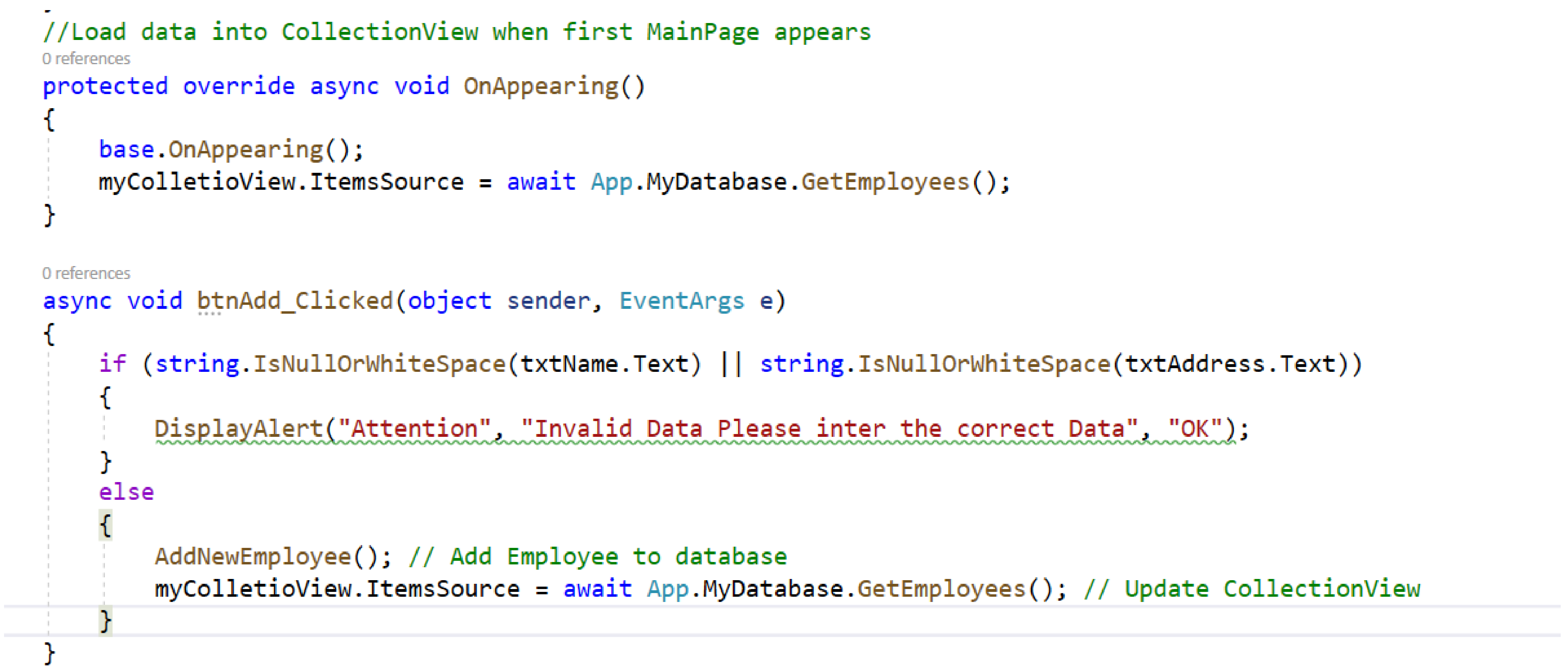
```
public class Database
{
    private readonly SQLiteAsyncConnection _database;
    1 reference
    public Database(string path)
    {
        _database = new SQLiteAsyncConnection(path);
        _database.CreateTableAsync<Employee>();
    }
    1 reference
    public Task<int> SaveData(Employee employee)
    {
        return _database.InsertAsync(employee);
    }
    4 references
    public Task<List<Employee>> GetEmployees()
    {
       return _database.Table<Employee>().ToListAsync();
    }
    1 reference
    public Task<int> DeleteEmployees(Employee employee)
    {
        return _database.DeleteAsync(employee);
    }
}
```

Add bellow code to App.xaml.cs file to create the database when first app loads

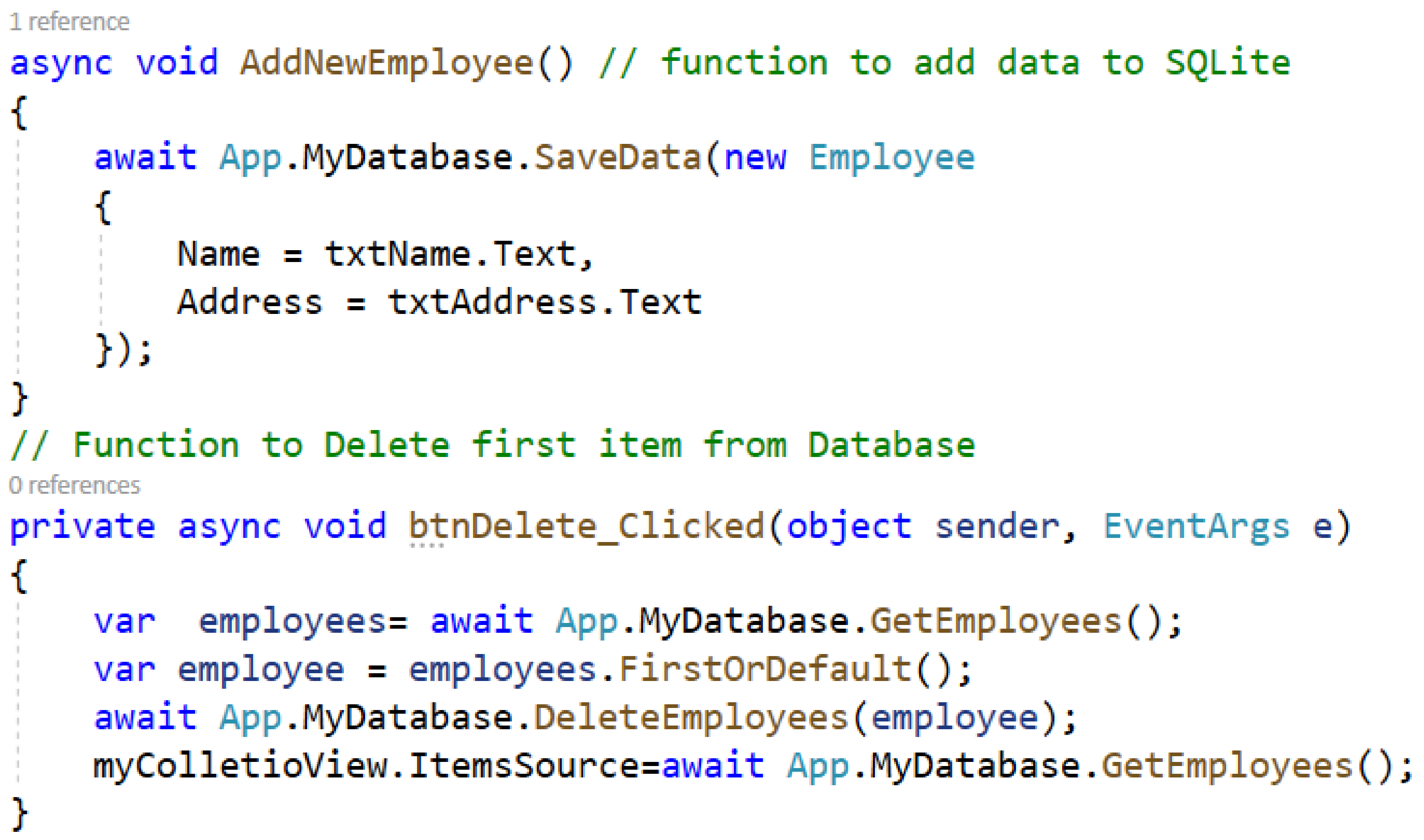


```
public partial class App : Application
{
    private static Database database;
    6 references
    public static Database MyDatabase
    {
        get
        {
            if (database == null)
            {
                database = new Database(Path.Combine(Environment.GetFolderPath
                    (Environment.SpecialFolder.LocalApplicationData),"SQLiteDB.db3"));
            }
            return database;
        }
    }
    2 references
    public App()
    {
        InitializeComponent();

        MainPage = new MainPage();
```

Write C# code in MainPage Class



```
//Load data into CollectionView when first MainPage appears
0 references
protected override async void OnAppearing()
{
    base.OnAppearing();
    myColletioView.ItemsSource = await App.MyDatabase.GetEmployees();
}

0 references
async void btnAdd_Clicked(object sender, EventArgs e)
{
    if (string.IsNullOrWhiteSpace(txtName.Text) || string.IsNullOrWhiteSpace(txtAddress.Text))
    {
        DisplayAlert("Attention", "Invalid Data Please inter the correct Data", "OK");
    }
    else
    {
        AddNewEmployee(); // Add Employee to database
        myColletioView.ItemsSource = await App.MyDatabase.GetEmployees(); // Update CollectionView
    }
}
```



```
1 reference
async void AddNewEmployee() // function to add data to SQLite
{
    await App.MyDatabase.SaveData(new Employee
    {
        Name = txtName.Text,
        Address = txtAddress.Text
    });
}
// Function to Delete first item from Database
0 references
private async void btnDelete_Clicked(object sender, EventArgs e)
{
    var  employees= await App.MyDatabase.GetEmployees();
    var employee = employees.FirstOrDefault();
    await App.MyDatabase.DeleteEmployees(employee);
    myColletioView.ItemsSource=await App.MyDatabase.GetEmployees();
}
```

Fore more info about SQLite Database go to the bellow link
[Xamarin.Forms Local Databases - Xamarin | Microsoft Docs]

Exercises

1- Create Fruit list in CollectionView.

2- Create Country list App in CollectionView and show Country Name, Capital with ability to Add, Edit, Update, Delete Items from CollectionView – Note: Use SQLite as your database to your data

Any Questions?