**Computing in Software Development**

*Year 2*

Module: *Internet & Mobile Application*

Chris Brennan

G00212584

Table of contents

1: Overview, Test data

2: Xaml Code – Mainpage

3,4,5: C# Code – Mainpage

6: Xaml Code – AnswerPage

7,8: C# Code – AnswerPage

Overview:

The project consists of two pages

Data to test for outcome:

*Birthday*

12-09-1987 *equates to* 1

31-12-1993 *equates to* 11

28-09-1955 *equates to* 22

31-02-1954 (error)

32-01-1855(error)

No entry (error)

1

Code:

MainPage.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:local="clr-namespace:Numerology"

x:Class="Numerology.MainPage"

BackgroundImage="Assets\mainbackground.png" >

<StackLayout VerticalOptions="Center">

<!--fields for value entry -->

<Label Text="Enter Date of Birth" TextColor="LightCyan" FontAttributes="Italic" FontSize="25" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />

<Grid HorizontalOptions="Center" VerticalOptions="Center" WidthRequest="300">

<!--Day -->

<Grid HorizontalOptions="Start" WidthRequest="96">

<Entry BackgroundColor="Ivory" x:Name="entryDay" Placeholder="D/DD" HorizontalTextAlignment="Center" Keyboard="Numeric" TextChanged="DayValid" />

</Grid>

<!--Month-->

<Grid HorizontalOptions="Center" WidthRequest="90" >

<Entry BackgroundColor="Ivory" x:Name="entryMonth" Placeholder="M/MM" HorizontalTextAlignment="Center" Keyboard="Numeric" TextChanged="MonthValid"/>

</Grid>

<!--Year-->

<Grid HorizontalOptions="End" WidthRequest="100">

<Entry BackgroundColor="Ivory" x:Name="entryYear" Placeholder="YYYYY" HorizontalTextAlignment="Center" Keyboard="Numeric" TextChanged="YearValid"/>

</Grid>

</Grid>

<Button TextColor="AliceBlue" Text="Answer" FontSize="20" HorizontalOptions="Center" VerticalOptions="Start" Clicked="Date\_Of\_Birth" />

<!--Numerology calculation return-->

<Label x:Name="numerologyCalc" FontSize="1" TextColor="Black" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />

<!--Previous Date-->

<Label x:Name="prevDate" TextColor="Orange" FontAttributes="Italic" FontSize="15" HorizontalOptions="Center" />

</StackLayout>

2

Code:

MainPage.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

namespace Numerology

{

public partial class MainPage : ContentPage

{

public MainPage()

{

InitializeComponent();

}

private async void AnswerPage()

{

await Navigation.PushAsync(new AnswerPage(numerologyCalc.Text));

}

//validate Day Entry

private void DayValid( object sender,EventArgs text)

{

//eliminates backspace crash on computer operation

if (!string.IsNullOrWhiteSpace(entryDay.Text))

{

int day = 0;

day = Convert.ToInt16(entryDay.Text);

if (day < 01 || day > 31)

{

DisplayAlert("Day", "Invalid Entry\n Enter between 1 - 31", "Retry");

}//innerIf

}//if

}

//validate Month Entry

private void MonthValid(object sender, TextChangedEventArgs text)

{

int dayCheck = 0;

if (!string.IsNullOrWhiteSpace(entryMonth.Text))

{

int month = 0;

month = Convert.ToInt16(entryMonth.Text);

if (month < 01 || month > 12)

{

DisplayAlert("Month", "Invalid Entry\n Enter between 1 - 12", "Retry");

}//innerIf

//further validation of Day/Month compatible

dayCheck = Convert.ToInt16(entryDay.Text);

switch (month)

{

case 2:

if (dayCheck > 29)//for leap year if necessary

{

DisplayAlert("Incorrect", "Please enter correct date with month\n" +

" " + dayCheck + " of February\n does not exist", "Retry");

InitializeComponent();

}

break;

case 4:

if (dayCheck > 30)

{

DisplayAlert("Incorrect", "Please enter correct date with month\n" +

" " + dayCheck + "st of April\n does not exist", "Retry");

InitializeComponent();

}//if

break;

case 6:

if (dayCheck > 30)

{

DisplayAlert("Incorrect", "Please enter correct date with month\n" +

" " + dayCheck + "st of June\n does not exist", "Retry");

InitializeComponent();

}//if

break;

3

case 9:

if (dayCheck > 30)

{

DisplayAlert("Incorrect", "Please enter correct date with month\n" +

" " + dayCheck + "st of September\n does not exist", "Retry");

InitializeComponent();

}//if

break;

case 11:

if (dayCheck > 30)

{

DisplayAlert("Incorrect", "Please enter correct date with month\n" +

" " + dayCheck + "st of Novemebr\n does not exist", "Retry");

InitializeComponent();

}//if

break;

}//switch

}//largeIf

}

//validate Year Entry

private void YearValid()

{

if (!string.IsNullOrWhiteSpace(entryYear.Text))

{

int year = 0;

year = Convert.ToInt16(entryYear.Text);

if (year < 0|| year > 2020)

{

DisplayAlert("Year", "Invalid Entry\n Enter between 1 - 2020", "Retry");

}//innerIf

}//if

}

//Main Calculation

private void Date\_Of\_Birth(object sender, EventArgs args)

{

// declaration of variables

int day = 0, day2 = 0, month = 0, month2 = 0;

int centuryYear = 0, centuryYear2 = 0, decadeYear = 0, decadeYear2 = 0;

int numerology = 0, breakdown=0, breakdown2=0;

if (string.IsNullOrWhiteSpace(entryDay.Text) || string.IsNullOrWhiteSpace(entryMonth.Text) || string.IsNullOrWhiteSpace(entryYear.Text))

{

DisplayAlert("No Entry", "Please enter all values", "Retry");

InitializeComponent();

}

else

{

//conversion of String to interger

day = Convert.ToInt16(entryDay.Text);

day2 = day % 10;

day = (day - day2) / 10;

day += day2;

switch (day)

{

case 10:

day = 1;

break;

case 11:

day = 2;

break;

}

//assign to numerology

numerology += day;

//calculate month

month = Convert.ToInt16(entryMonth.Text);

month2 = month % 10;

if (month > 9)

{

month = 1;

}

4

else

{

month = 0;

}

month += month2;

//assign to numerology

numerology += month;

centuryYear = Convert.ToInt16(entryYear.Text);

//breaking down to individual digits

centuryYear -= decadeYear = centuryYear % 100;

centuryYear2 = ((centuryYear / 100) % 10);

centuryYear /= 1000;

decadeYear2 = decadeYear % 10;

decadeYear = (decadeYear - decadeYear2) / 10;

numerology += centuryYear + centuryYear2 + decadeYear + decadeYear2;

if (numerology > 9 && !(numerology == 11) && !(numerology == 22) && !(numerology == 33))

{

breakdown2 = numerology % 10;

breakdown = numerology / 10;

numerology = breakdown + breakdown2;

if (numerology == 10)

{

numerology = 1;

}//innerIf

if (numerology >= 12)

{

breakdown2 = numerology % 10;

breakdown = numerology / 10;

numerology = breakdown + breakdown2;

}//innerIf

}//if

//return String format

numerologyCalc.Text = " " + numerology;

//return previous date

prevDate.Text = " \nLast Entered\n " + day + " : " + month + " : " + entryYear.Text;

//call answerPage

AnswerPage();

}//Large If/Else

}

}

}

5

Code:

AnswerPage.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="Numerology.AnswerPage"

BackgroundImage="Assets\answerpage.png" >

<ContentPage.Content>

<StackLayout>

<Grid >

<Grid.RowDefinitions>

<RowDefinition Height="50" />

<RowDefinition Height="75"/>

<RowDefinition Height="30" />

<RowDefinition Height="\*" />

<RowDefinition Height="30" />

<RowDefinition Height="\*" />

<RowDefinition Height="30" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="150" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<!--Calculated number from Main page-->

<Label x:Name="calcNumber" FontSize="50" FontAttributes="Italic" HorizontalTextAlignment="Center" TextColor="Orange" Grid.Row="1" Grid.Column="1" />

<!--Personality return-->

<Label Text="Personality" Grid.Row="2" Grid.Column="1" FontSize="22" FontAttributes="Italic" HorizontalTextAlignment="Center" TextColor="Orange" />

<Label x:Name="personality" FontSize="18" FontAttributes="Italic" HorizontalTextAlignment="Center" TextColor="white" Grid.Row="3" Grid.Column="1" />

<!--Work/Career return-->

<Label Text="Work &amp; Career" Grid.Row="4" Grid.Column="1" FontSize="22" FontAttributes="Italic" HorizontalTextAlignment="Center" TextColor="Orange"/>

<Label x:Name="work\_career" FontSize="18" FontAttributes="Italic" HorizontalTextAlignment="Center" TextColor="white" Grid.Row="5" Grid.Column="1"/>

<!--Challenges return-->

<Label Text="Challenges" Grid.Row="6" Grid.Column="1" FontSize="22" FontAttributes="Italic" HorizontalTextAlignment="Center" TextColor="Orange"/>

<Label x:Name="challenges" FontSize="18" FontAttributes="Italic" HorizontalTextAlignment="Center" TextColor="white" Grid.Row="7" Grid.Column="1"/>

</Grid>

</StackLayout>

</ContentPage.Content>

</ContentPage>

6

Code:

AnswerPage.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Numerology

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class AnswerPage : ContentPage

{

//Data passing Main to Answer

public AnswerPage (string number)

{

InitializeComponent ();

calcNumber.Text = number;

Explanation(calcNumber.Text);

}

//Definitions of calculated number

private void Explanation(string number)

{

int numerologyNum = 0;

string person = " ", work = " ", challenge = " ";

numerologyNum = Convert.ToInt16(number);

switch (numerologyNum)

{

case 1:

person = "Independent\nCreative ";

work = "Natural Leader\nAmbitious";

challenge = "Self doubt\nRisk taker ";

break;

case 2:

person = "Mediator\nDiplomatic";

work = "Team player\nVisionary";

challenge = "Oversensitive\nIndecisiveness ";

break;

case 3:

person = "Innovative\nCommunicative";

work = "Educator\nEntertainer";

challenge = "Frustration\nEscapist ";

break;

case 4:

person = "Dedicated\nTrustworthy";

work = "Managerial\nEntrepeneur";

challenge = "Stubborn\nObsessive ";

break;

case 5:

person = "Progressive\nAdventerous";

work = "Humanitarian\nCommunicator";

challenge = "Inflexible\nDistractible ";

break;

case 6:

person = "Nurturer\nRighteousness ";

work = "Councillor\nAuthority ";

challenge = "Overwhelmed\nSelf-righteousness ";

break;

case 7:

person = "Analytical\nObserver ";

work = "Scientic\nStudious";

challenge = "Pessimistic\nSecretive ";

break;

7

case 8:

person = "Organizational\nGovern ";

work = "Executive\nGovernment";

challenge = "Work-a-olic\nMaterialist ";

break;

case 9:

person = "compassionate\nGenerous";

work = "Artistic\nPhilosophical ";

challenge = "selflessness\nOver-burdened ";

break;

case 11:

person = "Enlighteners \n";

work = "Spiritual Balance ";

challenge = "overambitious\nVanity ";

break;

case 22:

person = "Creators \n ";

work = "Grandiose thinker ";

challenge = "Underachievement\nSelf-imposed pressure ";

break;

case 33:

person = "Master teachers\n ";

work = "Mover & shaker";

challenge = "Impulsive\nIdealist ";

break;

default:

person = "Please Return \n& Enter \nBirthdate";

work = "\n\n";

challenge = "\n\n";

break;

}

//Return definitions

personality.Text = person;

work\_career.Text = work;

challenges.Text = challenge;

}

}

}

8

Code:

App.xaml.cs

using System;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

[assembly: XamlCompilation(XamlCompilationOptions.Compile)]

namespace Numerology

{

public partial class App : Application

{

public App()

{

InitializeComponent();

// MainPage = new MainPage();

//page navigation

MainPage = new NavigationPage(new MainPage());

}

protected override void OnStart()

{

// Handle when your app starts

}

protected override void OnSleep()

{

// Handle when your app sleeps

}

protected override void OnResume()

{

// Handle when your app resumes

}

}

}