using Xamarin.Forms;

namespace BirthdayCalendar.Behaviors

{

class WhitespaceCheckBehavior : Behavior<Entry>

{

protected override void OnAttachedTo(Entry entry)

{

entry.TextChanged += OnEntryTextChanged;

base.OnAttachedTo(entry);

}

protected override void OnDetachingFrom(Entry entry)

{

entry.TextChanged -= OnEntryTextChanged;

base.OnDetachingFrom(entry);

}

void OnEntryTextChanged(object sender, TextChangedEventArgs args)

{

string text = args.NewTextValue;

((Entry)sender).TextColor = !text.Trim().Contains(" ") ? Color.Default : Color.Red;

}

}

}

using System.Collections.Generic;

using System.Threading.Tasks;

using BirthdayCalendar.Models;

using SQLite;

namespace BirthdayCalendar.Data

{

public class PersonsDB

{

readonly SQLiteAsyncConnection db;

public PersonsDB(string connectionString)

{

db = new SQLiteAsyncConnection(connectionString);

db.CreateTableAsync<Person>().Wait();

}

public Task<List<Person>> GetPersonsAsync()

{

return db.Table<Person>().ToListAsync();

}

public Task<Person> GetPersonAsync(int id)

{

return db.Table<Person>()

.Where(i => i.ID == id)

.FirstOrDefaultAsync();

}

public Task<int> SavePersonAsync(Person person)

{

if(person.ID != 0)

{

return db.UpdateAsync(person);

}

else

{

return db.InsertAsync(person);

}

}

public Task<int> DeletePersonAsync(Person person)

{

return db.DeleteAsync(person);

}

public Task<int> GetCountAsync()

{

return db.Table<Person>().CountAsync();

}

}

}

using SQLite;

namespace BirthdayCalendar.Models

{

public class Person

{

[PrimaryKey, AutoIncrement]

public int ID { get; set; }

public string Name { get; set; }

public string Date { get; set; }

public string Image { get; set; }

}

}

<?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="BirthdayCalendar.Pages.AddPage"

NavigationPage.HasNavigationBar="false"

xmlns:local="clr-namespace:BirthdayCalendar.Behaviors">

<ContentPage.Content>

<Grid HorizontalOptions="FillAndExpand">

<Image x:Name="Background"

Source="AddPage\_background.png"

HorizontalOptions="FillAndExpand"

Aspect="AspectFill"/>

<StackLayout Padding="15, 0, 15, 10"

Spacing="0">

<StackLayout x:Name="NavBar"

Orientation="Horizontal"

HorizontalOptions="Center"

Spacing="0">

<Button x:Name="NavBtnBack"

Clicked="NavBtnBack\_Clicked"

BackgroundColor="#0000"

CornerRadius="58"

HorizontalOptions="Center"

VerticalOptions="End"/>

<Button x:Name="NavBtnAdd"

BackgroundColor="#0000"

CornerRadius="58" HorizontalOptions="Center"

VerticalOptions="End"/>

</StackLayout>

<Label x:Name="Title"

Text="Add your&#10;friends"

TextColor="#EFF2F6"

FontFamily="Comfortaa"

FontAttributes="Bold"/>

<Grid x:Name="InputPanel">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Label x:Name="NameLabel"

Text="Name"

TextColor="#000"

FontFamily="Comfortaa"

Grid.Column="0"

Grid.Row="0"/>

<Label x:Name="SurnameLabel"

Text="Surname"

TextColor="#000"

FontFamily="Comfortaa"

Grid.Column="0"

Grid.Row="1"/>

<Label x:Name="DateLabel"

Text="Date"

TextColor="#000"

FontFamily="Comfortaa"

Grid.Column="0"

Grid.Row="2"/>

<Label x:Name="PhotoLabel"

Text="Photo"

TextColor="#000"

FontFamily="Comfortaa"

Grid.Column="0"

Grid.Row="3"/>

<Entry x:Name="InputName"

Placeholder="Enter name here"

PlaceholderColor="#145535"

Grid.Column="1"

Grid.Row="0">

<Entry.Behaviors>

<local:WhitespaceCheckBehavior/>

</Entry.Behaviors>

</Entry>

<Entry x:Name="InputSurname"

Placeholder="Enter surname here"

PlaceholderColor="#145535"

Grid.Column="1"

Grid.Row="1">

<Entry.Behaviors>

<local:WhitespaceCheckBehavior/>

</Entry.Behaviors>

</Entry>

<DatePicker x:Name="InputDate"

TextColor="#000"

Format="dd.MM.yyyy"

Grid.Column="1"

Grid.Row="2"/>

<Grid Grid.Column="1" Grid.Row="3">

<Label x:Name="PhotoBtnLabel"

Text="select photo -->"

TextColor="#145535"

VerticalOptions="CenterAndExpand"/>

<Button x:Name="PhotoBtn"

Clicked="PhotoBtn\_Clicked"

BackgroundColor="#0000"

VerticalOptions="Center"

HorizontalOptions="Start"/>

</Grid>

</Grid>

<StackLayout Orientation="Horizontal"

HorizontalOptions="Center"

VerticalOptions="EndAndExpand"

Spacing="0"

x:Name="bottomPanel">

<Button x:Name="AddBtn"

Clicked="AddBtn\_Clicked"

Text="save"

TextColor="#EFF2F6"

BackgroundColor="#145535"

FontAttributes="Bold"

FontFamily="Comfortaa"

Padding="0, -2, 0, 2"/>

</StackLayout>

</StackLayout>

</Grid>

</ContentPage.Content>

</ContentPage>

using BirthdayCalendar.Models;

using System;

using System.Globalization;

using Xamarin.Forms;

namespace BirthdayCalendar.Pages

{

public partial class AddPage : ContentPage

{

int id = 0;

public AddPage()

{

InitializeComponent();

}

public AddPage(int ID)

{

InitializeComponent();

id = ID;

}

protected override void OnAppearing()

{

base.OnAppearing();

Background.SizeChanged += BackgroundSizeChanged;

ChangePageContent(GetScaleFactor());

ResizeObjects(GetScaleFactor());

}

private double GetScaleFactor()

{

const double oldBackgroundWidth = 320;

double backgroundWidth = Background.Width;

double scaleFactor = backgroundWidth / oldBackgroundWidth;

return scaleFactor;

}

private void BackgroundSizeChanged(object sender, EventArgs e)

{

Background.SizeChanged -= BackgroundSizeChanged;

ResizeObjects(GetScaleFactor());

}

private void ResizeObjects(double scaleFactor)

{

NavBar.Margin = new Thickness(0, 68d \* scaleFactor / 1.61d, 0, 0);

NavBtnBack.HeightRequest = 58d \* scaleFactor;

NavBtnBack.WidthRequest = 58d \* scaleFactor;

NavBtnAdd.HeightRequest = 58d \* scaleFactor;

NavBtnAdd.WidthRequest = 58d \* scaleFactor;

Title.FontSize = 26 \* scaleFactor;

Title.Margin = new Thickness(0, 22d \* scaleFactor / 1.61d, 0, 0);

InputPanel.Margin = new Thickness(0, 75d \* scaleFactor / 1.61d, 0, 0);

NameLabel.FontSize = 16d \* scaleFactor;

SurnameLabel.FontSize = 16d \* scaleFactor;

DateLabel.FontSize = 16d \* scaleFactor;

PhotoLabel.FontSize = 16d \* scaleFactor;

InputName.FontSize = 16d \* scaleFactor;

InputName.Margin = new Thickness(10d \* scaleFactor, -4d \* scaleFactor, 0, 0);

InputSurname.FontSize = 16d \* scaleFactor;

InputSurname.Margin = new Thickness(10d \* scaleFactor, -4d \* scaleFactor, 0, 0);

InputDate.Margin = new Thickness(10d \* scaleFactor, -2d \* scaleFactor, 0, 0);

PhotoBtnLabel.FontSize = 12d \* scaleFactor;

PhotoBtnLabel.Margin = new Thickness(15d \* scaleFactor, 5d \* scaleFactor, 0, 0);

PhotoBtn.HeightRequest = 20 \* scaleFactor;

PhotoBtn.WidthRequest = 100d \* scaleFactor;

PhotoBtn.Margin = new Thickness(10d \* scaleFactor, 5d \* scaleFactor, 0, 0);

AddBtn.FontSize = 16d \* scaleFactor;

}

private async void ChangePageContent(double scaleFactor)

{

if (id > 0)

{

Title.Text = "Change information\nabout friend";

Person person = new Person();

person = await App.PersonsDB.GetPersonAsync(id);

InputName.Text = person.Name.Split(' ')[0];

InputSurname.Text = person.Name.Split(' ')[1];

InputDate.Date = DateTime.ParseExact(person.Date, "dd.MM.yyyy", CultureInfo.InvariantCulture);

Button changeBtn = new Button

{

Text = "change",

TextColor = Color.FromHex("#EFF2F6"),

BackgroundColor = Color.FromHex("#145535"),

FontAttributes = FontAttributes.Bold,

FontFamily = "Comfortaa",

Padding = new Thickness(0, -2, 0, 2),

Margin = new Thickness(0, 0, 30d \* scaleFactor / 1.61d, 0)

};

changeBtn.Clicked += ChangeBtn\_Clicked;

Button deleteBtn = new Button

{

Text = "delete",

TextColor = Color.FromHex("#EFF2F6"),

BackgroundColor = Color.FromHex("#145535"),

FontAttributes = FontAttributes.Bold,

FontFamily = "Comfortaa",

Padding = new Thickness(0, -2, 0, 2)

};

deleteBtn.Clicked += DeleteBtn\_Clicked;

bottomPanel.Children.Clear();

bottomPanel.Children.Add(changeBtn);

bottomPanel.Children.Add(deleteBtn);

}

}

private async void PhotoBtn\_Clicked(object sender, EventArgs e)

{

await DisplayAlert("Warning", "In developing...", "Ok");

}

private async void AddBtn\_Clicked(object sender, EventArgs e)

{

string name, surname, date, image;

Person person = new Person();

if (InputName.Text == null)

{

await DisplayAlert("Error", "Invalid name entry", "Ok");

return;

}

if (!InputName.Text.Trim().Contains(" ") & InputName.Text.Trim() != "")

{

name = InputName.Text.Trim();

}

else

{

await DisplayAlert("Error", "Invalid name entry", "Ok");

return;

}

if (InputSurname.Text == null)

{

await DisplayAlert("Error", "Invalid surname entry", "Ok");

return;

}

if (!InputSurname.Text.Trim().Contains(" ") & InputSurname.Text.Trim() != "")

{

surname = InputSurname.Text.Trim();

}

else

{

await DisplayAlert("Error", "Invalid surname entry", "Ok");

return;

}

DateTime inputDate = InputDate.Date;

date = inputDate.ToString("dd.MM.yyyy");

image = "";

if (image == "")

{

image = "user.png";

}

person.Name = name + " " + surname;

person.Date = date;

person.Image = image;

await App.PersonsDB.SavePersonAsync(person);

await Navigation.PushAsync(new FriendsPage());

}

private async void ChangeBtn\_Clicked(object sender, EventArgs e)

{

string name, surname, date, image;

Person person = new Person();

if (InputName.Text == null)

{

await DisplayAlert("Error", "Invalid name entry", "Ok");

return;

}

if (!InputName.Text.Trim().Contains(" ") & InputName.Text.Trim() != "")

{

name = InputName.Text.Trim();

}

else

{

await DisplayAlert("Error", "Invalid name entry", "Ok");

return;

}

if (InputSurname.Text == null)

{

await DisplayAlert("Error", "Invalid surname entry", "Ok");

return;

}

if (!InputSurname.Text.Trim().Contains(" ") & InputSurname.Text.Trim() != "")

{

surname = InputSurname.Text.Trim();

}

else

{

await DisplayAlert("Error", "Invalid surname entry", "Ok");

return;

}

DateTime inputDate = InputDate.Date;

date = inputDate.ToString("dd.MM.yyyy");

image = "";

if (image == "")

{

image = "user.png";

}

person.ID = id;

person.Name = name + " " + surname;

person.Date = date;

person.Image = image;

await App.PersonsDB.SavePersonAsync(person);

await Navigation.PushAsync(new FriendsPage());

}

private async void DeleteBtn\_Clicked(object sender, EventArgs e)

{

Person person = await App.PersonsDB.GetPersonAsync(id);

await App.PersonsDB.DeletePersonAsync(person);

await Navigation.PushAsync(new FriendsPage());

}

private async void NavBtnBack\_Clicked(object sender, EventArgs e)

{

await Navigation.PushAsync(new FriendsPage());

}

}

}

<?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="BirthdayCalendar.Pages.FriendsPage"

NavigationPage.HasNavigationBar="false">

<ContentPage.Content>

<Grid HorizontalOptions="FillAndExpand">

<Image x:Name="Background"

Source="FriendsPage\_background.png"

HorizontalOptions="FillAndExpand"

Aspect="AspectFill"/>

<StackLayout Padding="15, 15, 15, 10"

Spacing="0">

<StackLayout Orientation="Horizontal">

<Frame x:Name="searchPanel"

BackgroundColor="#EFF2F6"

BorderColor="#145535"

CornerRadius="32"

Padding="0">

<StackLayout Orientation="Horizontal"

Spacing="0">

<Image x:Name="searchIcon"

Source="loupe\_gray.png"/>

<Entry x:Name="searchInput"

Placeholder="Search"

PlaceholderColor="#B3B3B3"

TextColor="#000"

FontFamily="Arial"/>

</StackLayout>

</Frame>

<Label x:Name="contactsScore"

TextColor="#EFF2F6"

FontFamily="Comfortaa"

FontAttributes="Bold"

HorizontalOptions="EndAndExpand"

VerticalOptions="Center"/>

<Grid>

<Image x:Name="addContactIcon"

Source="add\_contact.png"/>

<Button x:Name ="addContactBtn"

Clicked="AddContactBtn\_Clicked"

CornerRadius="24"

BackgroundColor="#0000"/>

</Grid>

</StackLayout>

<ScrollView x:Name="friendsBox">

<StackLayout x:Name="friendsList">

</StackLayout>

</ScrollView>

<StackLayout Orientation="Horizontal"

HorizontalOptions="Center"

VerticalOptions="EndAndExpand"

Spacing="0">

<Button x:Name="NavBtnHome"

Clicked="NavBtnHome\_Clicked"

BackgroundColor="#0000" CornerRadius="58"

HorizontalOptions="Center"

VerticalOptions="End"/>

<Button x:Name="NavBtnList"

BackgroundColor="#0000"

CornerRadius="58"

HorizontalOptions="Center"

VerticalOptions="End"/>

</StackLayout>

</StackLayout>

</Grid>

</ContentPage.Content>

</ContentPage>

using System;

using System.Collections.Generic;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

using BirthdayCalendar.Models;

using System.Linq;

namespace BirthdayCalendar.Pages

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class FriendsPage : ContentPage

{

public FriendsPage()

{

InitializeComponent();

}

protected override async void OnAppearing()

{

base.OnAppearing();

Background.SizeChanged += BackgroundSizeChanged;

ShowDB(GetScaleFactor());

ResizeObjects(GetScaleFactor());

int score = await App.PersonsDB.GetCountAsync();

contactsScore.Text = score.ToString();

}

private double GetScaleFactor()

{

const double oldBackgroundWidth = 320;

double backgroundWidth = Background.Width;

double scaleFactor = backgroundWidth / oldBackgroundWidth;

return scaleFactor;

}

private void BackgroundSizeChanged(object sender, EventArgs e)

{

Background.SizeChanged -= BackgroundSizeChanged;

ResizeObjects(GetScaleFactor());

}

private void ResizeObjects(double scaleFactor)

{

searchPanel.HeightRequest = 30d \* scaleFactor;

searchPanel.WidthRequest = 187d \* scaleFactor;

searchPanel.Margin = new Thickness(0, 0, 0, 10d \* scaleFactor / 1.61d);

searchIcon.HeightRequest = 16d \* scaleFactor;

searchIcon.WidthRequest = 16d \* scaleFactor;

searchIcon.Margin = new Thickness(9d \* scaleFactor, 7d \* scaleFactor, 7d \* scaleFactor, 7d \* scaleFactor);

searchInput.HeightRequest = 16d \* scaleFactor;

searchInput.WidthRequest = 148d \* scaleFactor;

searchInput.FontSize = 10d \* scaleFactor;

addContactIcon.HeightRequest = 24d \* scaleFactor;

addContactIcon.WidthRequest = 24d \* scaleFactor;

addContactBtn.HeightRequest = 24d \* scaleFactor;

addContactBtn.WidthRequest = 24d \* scaleFactor;

addContactBtn.Margin = new Thickness(0, 3d \* scaleFactor);

contactsScore.FontSize = 16d \* scaleFactor;

friendsBox.Margin = new Thickness(0, 10d \* scaleFactor);

NavBtnHome.HeightRequest = 58d \* scaleFactor;

NavBtnHome.WidthRequest = 58d \* scaleFactor;

NavBtnList.HeightRequest = 58d \* scaleFactor;

NavBtnList.WidthRequest = 58d \* scaleFactor;

}

private async void ShowDB(double scaleFactor)

{

List<Person> personsList = new List<Person>(await App.PersonsDB.GetCountAsync());

personsList = await App.PersonsDB.GetPersonsAsync();

personsList = personsList.OrderBy(p => p.Name).ToList();

foreach (Person person in personsList)

{

StackLayout friendStackLayout = new StackLayout

{

Orientation = StackOrientation.Horizontal,

Spacing = 0,

Margin = new Thickness(0, 0, 0, 20d \* scaleFactor / 1.61d)

};

Frame friendIconMask = new Frame

{

HeightRequest = 40d \* scaleFactor,

WidthRequest = 40d \* scaleFactor,

VerticalOptions = LayoutOptions.Center,

CornerRadius = Convert.ToInt32(40d \* scaleFactor),

Padding = new Thickness(0),

IsClippedToBounds = true

};

Image friendIcon = new Image

{

HorizontalOptions = LayoutOptions.Center,

VerticalOptions = LayoutOptions.Center,

Source = person.Image

};

Label friendInfo = new Label

{

FontSize = 16d \* scaleFactor,

FontAttributes = FontAttributes.Bold,

VerticalOptions = LayoutOptions.Center,

FontFamily = "Comfortaa",

TextColor = Color.FromHex("#EFF2F6"),

Padding = new Thickness(17d \* scaleFactor, -2d \* scaleFactor, 0, 0),

Text = person.Name + "\n" + person.Date

};

Grid friendBtnBox = new Grid

{

HeightRequest = 34d \* scaleFactor,

WidthRequest = 34d \* scaleFactor,

VerticalOptions = LayoutOptions.Center,

HorizontalOptions = LayoutOptions.EndAndExpand

};

Button friendBtn = new Button

{

HeightRequest = 34d \* scaleFactor,

WidthRequest = 34d \* scaleFactor,

BackgroundColor = Color.FromHex("#0000"),

TextColor = Color.FromHex("#0000"),

Text = person.ID.ToString()

};

friendBtn.Clicked += FriendBtn\_Clicked;

Image friendBtnIcon = new Image

{

HeightRequest = 34d \* scaleFactor,

WidthRequest = 34d \* scaleFactor,

Source = "person\_button.png"

};

friendIconMask.Content = friendIcon;

friendStackLayout.Children.Add(friendIconMask);

friendStackLayout.Children.Add(friendInfo);

friendBtnBox.Children.Add(friendBtnIcon);

friendBtnBox.Children.Add(friendBtn);

friendStackLayout.Children.Add(friendBtnBox);

friendsList.Children.Add(friendStackLayout);

}

}

private async void AddContactBtn\_Clicked(object sender, EventArgs e)

{

await Navigation.PushAsync(new AddPage());

}

private async void FriendBtn\_Clicked(object sender, EventArgs e)

{

string btnText = (sender as Button).Text;

int id = Convert.ToInt32(btnText);

await Navigation.PushAsync(new AddPage(id));

}

private async void NavBtnHome\_Clicked(object sender, EventArgs e)

{

await Navigation.PushAsync(new MainPage());

}

}

}

<?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="BirthdayCalendar.Pages.MainPage"

NavigationPage.HasNavigationBar="false">

<Grid HorizontalOptions="FillAndExpand">

<Image x:Name="Background"

Source="MainPage\_background.png"

HorizontalOptions="FillAndExpand"

Aspect="AspectFill"/>

<StackLayout Padding="15, 15, 15, 10"

Spacing="0">

<Label x:Name="Date"

FontFamily="Comfortaa"

FontAttributes="Bold"

TextColor="#EFF2F6"

HorizontalTextAlignment="End"/>

<Label x:Name="Title"

FontFamily="Comfortaa"

FontAttributes="Bold"

TextColor="#145535"

Text="Birthday!"

HorizontalTextAlignment="Center"/>

<ScrollView x:Name="birthdayListScroll"

Scrolled="BirthdayListScroll\_Scrolled"

Orientation="Horizontal">

<StackLayout x:Name="birthdayList"

Orientation="Horizontal"

Spacing="0"/>

</ScrollView>

<StackLayout x:Name="friendInfo"

Spacing="0">

<Label x:Name="friendName"

FontFamily="Comfortaa"

FontAttributes="Bold"

TextColor="#EFF2F6"

HorizontalTextAlignment="Center"/>

<Label x:Name="friendYearOld"

FontFamily="Comfortaa"

FontAttributes="Bold"

TextColor="#EFF2F6"

HorizontalTextAlignment="Center"/>

</StackLayout>

<ScrollView>

<StackLayout x:Name="upcomingBirthdayList"

Spacing="0"

Orientation="Horizontal"

HorizontalOptions="LayoutOptions.CenterAndExpand">

<Label x:Name="upcomingBirthdaysLabel"

FontFamily="Comfortaa"

FontAttributes="Bold"

TextColor="#EFF2F6"

Text="No upcoming birthdays :("

HorizontalTextAlignment="Center"/>

</StackLayout>

</ScrollView>

<StackLayout Orientation="Horizontal"

HorizontalOptions="Center"

VerticalOptions="EndAndExpand"

Spacing="0">

<Button x:Name="NavBtnHome"

BackgroundColor="#0000"

CornerRadius="58

" HorizontalOptions="Center"

VerticalOptions="End"/>

<Button x:Name="NavBtnList"

Clicked="NavBtnList\_Clicked"

BackgroundColor="#0000"

CornerRadius="58"

HorizontalOptions="Center"

VerticalOptions="End"/>

</StackLayout>

</StackLayout>

</Grid>

</ContentPage>

using BirthdayCalendar.Models;

using System;

using System.Collections.Generic;

using System.Globalization;

using Xamarin.Forms;

namespace BirthdayCalendar.Pages

{

public partial class MainPage : ContentPage

{

private double oldSclollPos = 0;

private bool system = false;

List<Person> birthdaysToday = new List<Person>();

public MainPage()

{

InitializeComponent();

}

protected override void OnAppearing()

{

base.OnAppearing();

Background.SizeChanged += BackgroundSizeChanged;

ResizeObjects(GetScaleFactor());

ShowBirthdayToday(GetScaleFactor());

Date.Text = DateTime.Now.ToString("dd.MM.yyyy");

}

private double GetScaleFactor()

{

const double oldBackgroundWidth = 320;

double backgroundWidth = Background.Width;

double scaleFactor = backgroundWidth / oldBackgroundWidth;

return scaleFactor;

}

private void BackgroundSizeChanged(object sender, EventArgs e)

{

Background.SizeChanged -= BackgroundSizeChanged;

ResizeObjects(GetScaleFactor());

ShowBirthdayToday(GetScaleFactor());

}

private void ResizeObjects(double scaleFactor)

{

Date.FontSize = 14d \* scaleFactor;

Date.Margin = new Thickness(0, 0, 0, 13d \* scaleFactor / 1.61d);

Title.FontSize = 26d \* scaleFactor;

Title.Margin = new Thickness(0, 0, 0, 65d \* scaleFactor / 1.61d);

birthdayList.HeightRequest = 115d \* scaleFactor;

birthdayList.Margin = new Thickness(0, 0, 0, 10d \* scaleFactor / 1.61d);

friendName.FontSize = 16d \* scaleFactor;

friendYearOld.FontSize = 16d \* scaleFactor;

friendInfo.Margin = new Thickness(0, 0, 0, 59 \* scaleFactor / 1.61d);

upcomingBirthdaysLabel.FontSize = 18d \* scaleFactor;

NavBtnHome.HeightRequest = 58d \* scaleFactor;

NavBtnHome.WidthRequest = 58d \* scaleFactor;

NavBtnList.HeightRequest = 58d \* scaleFactor;

NavBtnList.WidthRequest = 58d \* scaleFactor;

}

private async void ShowBirthdayToday(double scaleFactor)

{

try

{

List<Person> personsList = new List<Person>(await App.PersonsDB.GetCountAsync());

personsList = await App.PersonsDB.GetPersonsAsync();

DateTime dateNow = DateTime.ParseExact(Date.Text, "dd.MM.yyyy", CultureInfo.InvariantCulture);

birthdaysToday.Clear();

foreach (Person person in personsList)

{

DateTime personBirthday = DateTime.ParseExact(person.Date, "dd.MM.yyyy", CultureInfo.InvariantCulture);

if (personBirthday.Day == dateNow.Day && personBirthday.Month == dateNow.Month)

{

birthdaysToday.Add(person);

}

}

if (birthdaysToday.Count > 0)

{

ShowPersonInfo(birthdaysToday);

birthdayList.Padding = new Thickness(148d \* scaleFactor / 1.61d, 0, 148d \* scaleFactor / 1.61d, 0);

birthdayList.Children.Clear();

foreach (Person person in birthdaysToday)

{

Frame friendIconMask = new Frame

{

HeightRequest = 114d \* scaleFactor,

WidthRequest = 114d \* scaleFactor,

VerticalOptions = LayoutOptions.Center,

CornerRadius = Convert.ToInt32(144d \* scaleFactor),

Margin = new Thickness(0, 0, 10d \* scaleFactor / 1.61d, 0),

Padding = new Thickness(0),

IsClippedToBounds = true

};

Image friendIcon = new Image

{

HorizontalOptions = LayoutOptions.Center,

VerticalOptions = LayoutOptions.Center,

Source = person.Image

};

friendIconMask.Content = friendIcon;

birthdayList.Children.Add(friendIconMask);

}

}

else

{

Label birthdayTodayLabel = new Label

{

FontSize = 18d \* scaleFactor,

FontFamily = "Comfortaa",

FontAttributes = FontAttributes.Bold,

TextColor = Color.FromHex("#EFF2F6"),

Text = "There are no birthdays today :("

};

birthdayList.HorizontalOptions = LayoutOptions.CenterAndExpand;

birthdayList.Children.Clear();

birthdayList.Children.Add(birthdayTodayLabel);

}

}

catch (ArgumentNullException) { ShowBirthdayToday(GetScaleFactor()); }

}

private async void BirthdayListScroll\_Scrolled(object sender, ScrolledEventArgs e)

{

if (sender is ScrollView scrollView && !system)

{

system = true;

int x = (int)Math.Abs(e.ScrollX);

if (oldSclollPos > x)

{

// Прокрутка вправо

await scrollView.ScrollToAsync(x - (x % 155), 0, true);

oldSclollPos = x - (x % 155);

}

else

{

// Прокрутка влево

await scrollView.ScrollToAsync(x + (155 - (x % 155)), 0, true);

oldSclollPos = x + (155 - (x % 155));

}

system = false;

ShowPersonInfo(birthdaysToday);

}

}

private async void ShowPersonInfo(List<Person> list)

{

try

{

int id = (int)oldSclollPos / 155;

friendName.Text = list[id].Name;

friendYearOld.Text = (DateTime.Now.Year - DateTime.ParseExact(list[id].Date, "dd.MM.yyyy", CultureInfo.InvariantCulture).Year).ToString();

}

catch (IndexOutOfRangeException) { }

catch (ArgumentOutOfRangeException) { };

}

private async void NavBtnList\_Clicked(object sender, EventArgs e)

{

await Navigation.PushAsync(new FriendsPage());

}

}

}

<?xml version="1.0" encoding="utf-8" ?>

<Application xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="BirthdayCalendar.App">

<Application.Resources>

</Application.Resources>

</Application>

using System;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

using BirthdayCalendar.Pages;

using BirthdayCalendar.Data;

using System.IO;

namespace BirthdayCalendar

{

public partial class App : Application

{

static PersonsDB personsDB;

public static PersonsDB PersonsDB

{

get

{

if (personsDB == null)

{

personsDB = new PersonsDB(

Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData),

"PersonsDatabase.db3"));

}

return personsDB;

}

}

public App()

{

InitializeComponent();

MainPage = new NavigationPage(new MainPage());

}

protected override void OnStart()

{

}

protected override void OnSleep()

{

}

protected override void OnResume()

{

}

}

}