**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Руководитель департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. М. Авдошин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |
| --- | --- |
| *Подп. и дата* |  |
| *Инв. № дубл.* |  |
| *Взам. инв. №* |  |
| *Подп. и дата* |  |
| *Инв. № подл* | *RU.17701729.04.01-01 12 01-1* |

**ПРОГРАММНЫЙ СИНТЕЗАТОР В ФОРМАТЕ ИНСТРУМЕНТА VST**

**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 12 01-1-ЛУ**

|  |  |
| --- | --- |
|  | Исполнитель:  студент группы БПИ182  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /М. А. Шакура /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

**2019**

|  |
| --- |
| УТВЕРЖДЕН  RU.17701729.04.01-01 12 01-1-ЛУ |

|  |  |
| --- | --- |
| *Подп. и дата* |  |
| *Инв. № дубл.* |  |
| *Взам. инв. №* |  |
| *Подп. и дата* |  |
| *Инв. № подл* | *RU.17701729.04.01-01 12 01-1* |

**ПРИЛОЖЕНИЕ АКТИВНОСТИ И УВЕДОИЛЕНИЯ ВШЭ**

**Текст программы**

**RU.17701729.04.01-01 12 01-1**

**Листов 20**

**2019**

# АННОТАЦИЯ

В данном программном документе представлен текст программы Приложение активности и уведомления ВШЭ». В разделе «Текст программы» находится текст программы, распределённый по файлам.

**СОДЕРЖАНИЕ**

[АННОТАЦИЯ 2](#_Toc8912161)

[1. ТЕКСТ ПРОГРАММЫ 4](#_Toc8912162)

[1.1. Models\AppData.cs 4](#_Toc8912163)

[1.2. Models\Item.cs 6](#_Toc8912164)

[1.3. Servises\IDataStore.cs 7](#_Toc8912165)

[1.4. Servises\MockDataStore.cs 7](#_Toc8912166)

[1.5. ViewModels\AboutViewModel.cs 8](#_Toc8912167)

[1.6. ViewModels\BaseViewModel.cs 8](#_Toc8912168)

[1.7. ViewModels\ItemDetailViewModel.cs 9](#_Toc8912169)

[1.8. ViewModels\ItemsViewModel.cs 9](#_Toc8912170)

[1.9. Views\AboutPage.xaml.cs 9](#_Toc8912171)

[1.10. Views\Favorite.xaml.cs 9](#_Toc8912172)

[1.11. Views\ItemDetailPage.xaml.cs 9](#_Toc8912173)

[1.12. Views\ItemsPage.xaml.cs 9](#_Toc8912174)

[1.13. Views\MainPage.xaml.cs 9](#_Toc8912175)

[1.14. App.xaml.cs 9](#_Toc8912176)

[1.15. Views\AboutPage.xaml 9](#_Toc8912177)

[1.16. Views\Favorite.xaml 9](#_Toc8912178)

[1.17. Views\ItemDetailPage.xaml 9](#_Toc8912179)

[1.18. Views\ItemsPage.xaml 9](#_Toc8912180)

[1.19. Views\MainPage.xaml 9](#_Toc8912181)

[1.20. App.xaml 9](#_Toc8912182)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 10](#_Toc8912183)

# ТЕКСТ ПРОГРАММЫ

## Models\AppData.cs

using System;

using System.Collections.Generic;

using Xamarin.Forms;

using System.IO;

using Newtonsoft.Json;

namespace App5.Models

{

/// <summary>

/// Sctruct of Data which writes to files

/// </summary>

struct FileStruct

{

public bool isrus { set; get; }

public bool IsThemeWhite { set; get; }

public List<string> Links;

}

static class AppData

{

static public Action SettingChanged;

static public List<string> Links = new List<string>();

/// <summary>

/// Langs

/// </summary>

static public string ru;

static public string en;

static public bool isrus { set; get; } = true;

/// <summary>

/// Colors

/// </summary>

static public bool IsThemeWhite { set; get; } = true;

static public Color BackgroundColor { get => IsThemeWhite ? Color.White : Color.FromRgb(16, 16, 16); }

static public Color BarBackgroundColor { get => IsThemeWhite ? Color.White : Color.FromRgb(10, 10, 10); }

static public Color FrontColor { get => IsThemeWhite ? Color.Black : Color.White; }

static AppData()

{

SettingChanged += Load;

/// <summary>

/// Read from file

/// </summary>

try

{

FileStruct a;

string json;

string path = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

string filePath = Path.Combine(path, "AppData.txt");

using (var file = File.Open(filePath, FileMode.Open, FileAccess.Read))

using (var strm = new StreamReader(file))

{

json = strm.ReadToEnd();

}

a = JsonConvert.DeserializeObject<FileStruct>(json);

isrus = a.isrus;

IsThemeWhite = a.IsThemeWhite;

Links = a.Links;

}

catch (Exception e){}

}

/// <summary>

/// Write to file

/// </summary>

static void Load()

{

try

{

FileStruct a = new FileStruct() {isrus=isrus,

IsThemeWhite= IsThemeWhite,

Links = Links};

string json = JsonConvert.SerializeObject(a);

string path = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

string filePath = Path.Combine(path, "AppData.txt");

using (var file = File.Open(filePath, FileMode.Create, FileAccess.Write))

using (var strm = new StreamWriter(file))

{

strm.Write(json);

}

}

catch (Exception e){}

}

}

}

## Models\Item.cs

using Xamarin.Forms;

namespace App5.Models

{

public class Item

{

public string Type { get; set; }

public string Day { get; set; } = "";

public string Header { get; set; } = "";

public string Description { get => string.Join(", ", Tags); }

public string ExtraString { get => string.Join(", ", Extra); }

public string Link { get; set; } = "";

public string Place { get; set; } = "";

public string Time { get; set; } = "";

public string Date { get => Time == "" ? Day : Day + ", " + Time; }

public string[] Tags { set; get; } = { };

public string[] Extra { set; get; } = { };

public bool DayVisible { get => Type == "Day"; }

public bool ExtraVisible { get => ExtraString != "" && Type == "Event"; }

public bool DescriptionVisible { get => Description != "" && Type == "Event"; }

public bool TimeVisible { get => Time != "" && Type == "Event"; }

public bool PlaceVisible { get => Place != "" && Type == "Event"; }

public bool TagsVisible { get => Tags.Length > 0 && Type == "Event"; }

public bool HeaderVisible { get => Type == "Event"; }

public bool Clickable { get => Type == "Event"; }

public string FavSource { get => AppData.Links.Contains(Link) ? "Added.png" : "fplus.png"; }

public Color FrontColor { get => AppData.FrontColor;}

public Color BackgroundColor { get => AppData.BackgroundColor; }

}

}

## Servises\IDataStore.cs

using System.Collections.Generic;

using System.Threading.Tasks;

namespace App5.Services

{

public interface IDataStore<T>

{

Task<IEnumerable<T>> GetItemsAsync(bool forceRefresh = false);

}

}

## Servises\MockDataStore.cs

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Threading.Tasks;

using App5.Models;

using Newtonsoft.Json;

using App5.Views;

namespace App5.Services

{

public class MockDataStore : IDataStore<Item>

{

/// <summary>

/// List of Items

/// </summary>

List<Item> items;

/// <summary>

/// Request to server

/// </summary>

/// <param name="url">Link</param>

/// <returns></returns>

static public string Get(string url)

{

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;

try

{

using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())

using (Stream stream = response.GetResponseStream())

using (StreamReader reader = new StreamReader(stream))

return reader.ReadToEnd();

}

catch

{

return null;

}

}

/// <summary>

/// Reloadl Data from server

/// </summary>

static public void RealoadData()

{

AppData.ru = Get("https://shakura.dev/hseapi");

AppData.en = Get("https://shakura.dev/hseapien");

AboutPage.f();

}

/// <summary>

/// Local relaod data after swiching language

/// </summary>

public MockDataStore()

{

Item[] data = JsonConvert.DeserializeObject<Item[]>(AppData.isrus ? AppData.ru : AppData.en);

items = data.OfType<Item>().ToList();

}

public async Task<IEnumerable<Item>> GetItemsAsync(bool forceRefresh = false)

{

return await Task.FromResult(items);

}

}

}

## ViewModels\AboutViewModel.cs

## ViewModels\BaseViewModel.cs

## ViewModels\ItemDetailViewModel.cs

## ViewModels\ItemsViewModel.cs

## Views\AboutPage.xaml.cs

## Views\Favorite.xaml.cs

## Views\ItemDetailPage.xaml.cs

## Views\ItemsPage.xaml.cs

## Views\MainPage.xaml.cs

## App.xaml.cs

## Views\AboutPage.xaml

## Views\Favorite.xaml

## Views\ItemDetailPage.xaml

## Views\ItemsPage.xaml

## Views\MainPage.xaml

## App.xaml

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |