|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.04.05-01 51 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.04.05-01 12 01-2 | | **АГРЕГАТОР КУЛИНАРНЫХ РЕЦЕПТОВ НА ANDROID**  **КЛИЕНТ**  **Текст программы**  **RU.17701729.04.05-01 12 01-2**  **Листов 60** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2020**

ОГЛАВЛЕНИЕ

[1. ТЕКСТ ПРОГРАММЫ 4](#_Toc40317077)

[1.1. InfoActivity.cs 4](#_Toc40317078)

[1.2. MainActivity.cs 4](#_Toc40317079)

[1.3. PlatesActivity.cs 10](#_Toc40317080)

[1.4. RecipeActivity.cs 11](#_Toc40317081)

[1.5. SavedIngredientsActivity.cs 15](#_Toc40317082)

[1.6. SavedRecipesActivity.cs 18](#_Toc40317083)

[1.7. SectionRecipesActivity.cs 20](#_Toc40317084)

[1.8. IngredientsAdapter.cs 23](#_Toc40317085)

[1.9. PagerRecipeAdapter.cs 24](#_Toc40317086)

[1.10. PlateAdapter.cs 25](#_Toc40317087)

[1.11. RecipeAdapter.cs 26](#_Toc40317088)

[1.12. SavedIngredientsAdapter.cs 27](#_Toc40317089)

[1.13. StepAdapter.cs 28](#_Toc40317090)

[1.14. RecipeDescriptionFragment.cs 29](#_Toc40317091)

[1.15. RecipeIngredientsFragment.cs 31](#_Toc40317092)

[1.16. RecipeStepsFragment.cs 32](#_Toc40317093)

[1.17. Plate.cs 33](#_Toc40317094)

[1.18. RecipeScrollListener.cs 34](#_Toc40317095)

[1.19. PlateViewHolder.cs 34](#_Toc40317096)

[1.20. RecipeViewHolder.cs 35](#_Toc40317097)

[1.21. SavedIngredientsViewHolder.cs 35](#_Toc40317098)

[1.22. RecipeData.cs 36](#_Toc40317099)

[1.23. IngredientData.cs 38](#_Toc40317100)

[1.24. HttpContext.cs 39](#_Toc40317101)

[1.25. DataContext.cs 40](#_Toc40317102)

[1.26. RecipeTable.cs 40](#_Toc40317103)

[1.27. IngredientTable.cs 41](#_Toc40317104)

[1.28. rectangle\_plate.xml 41](#_Toc40317105)

[1.29. round\_arrow\_back\_ios\_24.xml 41](#_Toc40317106)

[1.30. round\_clear\_24.xml 42](#_Toc40317107)

[1.31. round\_menu\_24.xml 42](#_Toc40317108)

[1.32. round\_share\_24.xml 42](#_Toc40317109)

[1.33. round\_timer\_24.xml\ 43](#_Toc40317110)

[1.34. round\_web\_24.xml 43](#_Toc40317111)

[1.35. spinner.xml 43](#_Toc40317112)

[1.36. activity\_info.xml 44](#_Toc40317113)

[1.37. activity\_main.xml 45](#_Toc40317114)

[1.38. activity\_plates.xml 46](#_Toc40317115)

[1.39. activity\_recipe.xml 47](#_Toc40317116)

[1.40. activity\_savedIngredients.xml 49](#_Toc40317117)

[1.41. activity\_savedRecipes.xml 50](#_Toc40317118)

[1.42. activity\_sections.xml 51](#_Toc40317119)

[1.43. fragment\_recipe\_description.xml 52](#_Toc40317120)

[1.44. fragment\_recipe\_ingredients.xml 53](#_Toc40317121)

[1.45. fragment\_recipe\_steps.xml 54](#_Toc40317122)

[1.46. list\_ingredients.xml 54](#_Toc40317123)

[1.47. list\_ingredients\_saved.xml 55](#_Toc40317124)

[1.48. list\_item.xml 56](#_Toc40317125)

[1.49. list\_plates.xml 57](#_Toc40317126)

[1.50. list\_steps.xml 57](#_Toc40317127)

[1.51. main\_bar.xml 58](#_Toc40317128)

[1.52. nav\_header\_main.xml 59](#_Toc40317129)

[1.53. spinner\_text.xml 60](#_Toc40317130)

[1.54. activity\_main\_drawer.xml 60](#_Toc40317131)

[1.55. menu\_main.xml 61](#_Toc40317132)

[1.56. colors.xml 61](#_Toc40317133)

[1.57. strings.xml 61](#_Toc40317134)

# ТЕКСТ ПРОГРАММЫ

## InfoActivity.cs

using System;

using Android.Widget;

using Android.App;

using Android.OS;

using Android.Runtime;

using Android.Support.V7.App;

using XamarinApp;

namespace AndroidApp

{

[Activity(Label = "На плите!", Theme = "@style/AppTheme.NoActionBar", Icon = "@drawable/icon",

ConfigurationChanges = Android.Content.PM.ConfigChanges.Orientation | Android.Content.PM.ConfigChanges.ScreenSize)]

internal class InfoActivity : AppCompatActivity

{

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

SetContentView(Resource.Layout.activity\_info);

var relativeLayoutBack = FindViewById<RelativeLayout>(Resource.Id.relativeLayoutBack);

var buttonBack = FindViewById<Button>(Resource.Id.buttonBack);

relativeLayoutBack.Click += OnBackPressed;

buttonBack.Click += OnBackPressed;

}

private void OnBackPressed(object sender, EventArgs args) => base.OnBackPressed();

public override void OnRequestPermissionsResult(int requestCode, string[] permissions,

[GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

}

}

## MainActivity.cs

using Android.App;

using Android.Content;

using Android.Graphics;

using Android.OS;

using Android.Runtime;

using Android.Support.Design.Widget;

using Android.Support.V4.View;

using Android.Support.V4.Widget;

using Android.Support.V7.App;

using Android.Support.V7.Widget;

using Android.Views;

using Android.Widget;

using ObjectsLibrary;

using System;

using XamarinApp;

using System.Collections.Generic;

using System.Threading.Tasks;

using AndroidLibrary;

namespace AndroidApp

{

[Activity(Label = "На плите!", Theme = "@style/AppTheme.NoActionBar", Icon = "@drawable/icon", MainLauncher = true,

ConfigurationChanges =

Android.Content.PM.ConfigChanges.Orientation | Android.Content.PM.ConfigChanges.ScreenSize)]

public class MainActivity : AppCompatActivity, NavigationView.IOnNavigationItemSelectedListener

{

private RecyclerView recyclerView;

private RecipeAdapter recipeAdapter;

private DrawerLayout drawer;

private SwipeRefreshLayout swipeRefreshLayout;

private LinearLayoutManager linearLayoutManager;

private NavigationView navigationView;

private Button buttonMenu;

private RecipeScrollListener recipeListener;

private EditText editText;

private Spinner spinner;

private ArrayAdapter spinnerAdapter;

private ActionBarDrawerToggle actionBarDrawerToggle;

private Button findButton;

private List<RecipeShort> recipeShorts;

private int page = 1;

private string lastQuery;

private string messageErrorTitle;

private string messageErrorText;

private string messageErrorTextButton = "Повторить";

private Android.Support.V7.App.AlertDialog.Builder errorDialog;

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

SetContentView(Resource.Layout.activity\_main);

// Определяем компоненты:

recyclerView = FindViewById<RecyclerView>(Resource.Id.listRecipeShorts);

swipeRefreshLayout = FindViewById<SwipeRefreshLayout>(Resource.Id.swipeRefreshLayout);

drawer = FindViewById<DrawerLayout>(Resource.Id.drawer\_layout);

navigationView = FindViewById<NavigationView>(Resource.Id.nav\_view);

buttonMenu = FindViewById<Button>(Resource.Id.menu\_button);

editText = FindViewById<EditText>(Resource.Id.TextFind);

spinner = FindViewById<Spinner>(Resource.Id.spinner);

findButton = FindViewById<Button>(Resource.Id.findButton);

linearLayoutManager = new LinearLayoutManager(this);

recipeListener = new RecipeScrollListener(linearLayoutManager);

// Инициализируем элементы:

swipeRefreshLayout.SetColorSchemeColors(Color.Orange, Color.DarkOrange);

swipeRefreshLayout.Refresh += RefreshLayout;

recyclerView.AddOnScrollListener(recipeListener);

recipeListener.LoadMoreEvent += LoadMoreElements;

findButton.Click += OnClickFind;

editText.KeyPress += FindByRecipeName;

actionBarDrawerToggle = new ActionBarDrawerToggle(this, drawer,

Resource.String.navigation\_drawer\_open, Resource.String.navigation\_drawer\_close);

drawer.AddDrawerListener(actionBarDrawerToggle);

actionBarDrawerToggle.SyncState();

navigationView.SetNavigationItemSelectedListener(this);

buttonMenu.SetBackgroundResource(Resources.GetIdentifier("round\_menu\_24", "drawable", PackageName));

buttonMenu.Click += SetMenuButtonClick;

spinner.ItemSelected += new EventHandler<AdapterView.ItemSelectedEventArgs>(SelectedItemSpinner);

spinnerAdapter =

ArrayAdapter.CreateFromResource(this, Resource.Array.sort\_array, Resource.Layout.spinner\_text);

spinnerAdapter.SetDropDownViewResource(Resource.Layout.spinner\_text);

spinner.Adapter = spinnerAdapter;

string[] messagesError = Resources.GetStringArray(Resource.Array.messagesError);

messageErrorTitle = messagesError[0];

messageErrorText = messagesError[1];

errorDialog = new Android.Support.V7.App.AlertDialog.Builder(this)

.SetTitle(messageErrorTitle)

.SetMessage(messageErrorText)

.SetCancelable(false)

.SetNeutralButton(messageErrorTextButton, RefreshLayout);

}

private void OnClickFind(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(editText.Text))

return;

page = 1;

lastQuery = $"section=recipe&recipeName={editText.Text}&page={page}";

UpdateListView(lastQuery);

Window.SetSoftInputMode(SoftInput.StateAlwaysHidden);

}

private void FindByRecipeName(object sender, View.KeyEventArgs e)

{

e.Handled = false;

if (e.Event.Action != KeyEventActions.Down || e.KeyCode != Keycode.Enter)

return;

if (string.IsNullOrWhiteSpace(editText.Text))

return;

page = 1;

Window.SetSoftInputMode(SoftInput.StateHidden);

lastQuery = $"section=recipe&recipeName={editText.Text}&page={page}";

UpdateListView(lastQuery);

//Toast.MakeText(this, "Загрузка...", ToastLength.Short).Show();

e.Handled = true;

}

private void LoadMoreElements(object sender, EventArgs e)

{

// Получаем строку для нового запроса:

string query = lastQuery.Substring(0, lastQuery.IndexOf("page=") + 5) + (++page);

// Обновляем коллекцию:

UpdateListView(query, recipeShorts);

}

private void RefreshLayout(object sender, EventArgs e)

{

// Обновление странички.

if (lastQuery == null)

UpdateListView();

else

UpdateListView(lastQuery);

}

private void SetMenuButtonClick(object sender, EventArgs args)

{

// Кнопка "меню" на toolbar.

drawer.OpenDrawer(GravityCompat.Start);

}

private async void UpdateListView(string query = "section=new", List<RecipeShort> recipeShorts = null)

{

// Запустить кружочек.

swipeRefreshLayout.Post(() =>

{

swipeRefreshLayout.Refreshing = true;

recyclerView.Clickable = false;

});

recyclerView.SetLayoutManager(linearLayoutManager);

try

{

if (recipeShorts == null)

{

// Тут какой-то костыль, трогать я не буду, простите :(

// Из книги "Техники индуса-программиста-под-андройд-на-ксамарине-с-нуля".

this.recipeShorts = await UpdateCollectionRecipes(query);

recipeAdapter = new RecipeAdapter(this.recipeShorts, this);

recipeAdapter.ItemClick += OnRecipeClick;

recyclerView.SetAdapter(recipeAdapter);

}

else

{

var newRecipes = await UpdateCollectionRecipes(query);

recipeAdapter.AddItems(newRecipes);

recipeShorts.AddRange(newRecipes);

}

}

catch (Exception exp)

{

errorDialog.SetMessage(exp.Message);

errorDialog.Show();

}

finally

{

// Остановить кружочек.

swipeRefreshLayout.Post(() =>

{

swipeRefreshLayout.Refreshing = false;

recyclerView.Clickable = true;

});

}

}

public override void OnBackPressed()

{

if (drawer.IsDrawerOpen(GravityCompat.Start))

drawer.CloseDrawer(GravityCompat.Start);

else

base.OnBackPressed();

}

public override bool OnCreateOptionsMenu(IMenu menu)

{

MenuInflater.Inflate(Resource.Menu.menu\_main, menu);

return true;

}

private void OnRecipeClick(object sender, int position)

{

var intent = new Intent(this, typeof(RecipeActivity));

intent.PutExtra("url", recipeShorts[position].Url);

intent.PutExtra("recipeShort", DataContext.RecipeToByteArray(recipeShorts[position]));

StartActivity(intent);

}

private void SelectedItemSpinner(object sender, AdapterView.ItemSelectedEventArgs e)

{

var item = spinner.GetItemAtPosition(e.Position);

page = 1;

string query = null;

switch (item.ToString())

{

case "Популярные рецепты":

query = $"section=popular&page={page}";

UpdateListView(query);

break;

case "Случайные рецепты":

query = $"section=random&page={page}";

UpdateListView(query);

break;

case "Новые рецепты":

query = $"section=new&page={page}";

UpdateListView(query);

break;

}

lastQuery = query;

}

private async Task<List<RecipeShort>> UpdateCollectionRecipes(string query)

{

return await Task.Run(function: () => HttpContext.GetPages(query));

}

public override void OnRequestPermissionsResult(int requestCode, string[] permissions,

[GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

public bool OnNavigationItemSelected(IMenuItem menuItem)

{

// Закрывает отрисовщик и возвращает, закрыт ли он или нет.

int id = menuItem.ItemId;

switch (id)

{

case Resource.Id.nav\_favorite:

{

Intent intent = new Intent(this, typeof(SavedRecipesActivity));

StartActivity(intent);

break;

}

case Resource.Id.nav\_cart:

{

Intent intent = new Intent(this, typeof(SavedIngredientsActivity));

StartActivity(intent);

break;

}

case Resource.Id.nav\_section:

{

Intent intent = new Intent(this, typeof(PlatesActivity));

StartActivity(intent);

break;

}

case Resource.Id.nav\_app:

{

Intent intent = new Intent(this, typeof(InfoActivity));

StartActivity(intent);

break;

}

}

return CLoseDrawer(drawer).IsDrawerOpen(GravityCompat.Start);

}

private static DrawerLayout CLoseDrawer(DrawerLayout drawerLayout)

{

drawerLayout.CloseDrawer(GravityCompat.Start);

return drawerLayout;

}

}

}

## PlatesActivity.cs

using System;

using System.Collections.Generic;

using Android.Widget;

using AndroidLibrary;

using Android.App;

using Android.OS;

using Android.Runtime;

using Android.Support.V7.App;

using Android.Support.V7.Widget;

using Android.Content;

using AndroidApp.General;

using XamarinApp;

namespace AndroidApp

{

[Activity(Label = "На плите!", Theme = "@style/AppTheme.NoActionBar", Icon = "@drawable/icon",

ConfigurationChanges =

Android.Content.PM.ConfigChanges.Orientation | Android.Content.PM.ConfigChanges.ScreenSize)]

internal class PlatesActivity : AppCompatActivity

{

private RecyclerView recyclerView;

private LinearLayoutManager linearLayoutManager;

private List<Plate> plates;

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

SetContentView(Resource.Layout.activity\_plates);

recyclerView = FindViewById<RecyclerView>(Resource.Id.recyclerViewRecipesPlates);

linearLayoutManager = new LinearLayoutManager(this);

recyclerView.SetLayoutManager(linearLayoutManager);

var relativeLayoutBack = FindViewById<RelativeLayout>(Resource.Id.relativeLayoutBack);

var buttonBack = FindViewById<Button>(Resource.Id.buttonBack);

relativeLayoutBack.Click += OnBackPressed;

buttonBack.Click += OnBackPressed;

plates = new List<Plate>()

{

new Plate("Основные блюда", "osnovnye", "горячее"),

new Plate("Супы", "supy", "супы"),

new Plate("Салаты", "salaty", "салаты"),

new Plate("Выпечка", "vypechka", "выпечка"),

new Plate("Десерты", "deserty", "десерты"),

new Plate("Закуски", "zakuski", "закуски"),

new Plate("Соусы", "sousy", "соусы")

};

var adapter = new PlateAdapter(plates, this);

adapter.ItemClick += OnPlateClick;

recyclerView.SetAdapter(adapter);

}

private void OnPlateClick(object sender, int position)

{

Intent intent = new Intent(this, typeof(SectionRecipesActivity));

intent.PutExtra("key", plates[position].Key);

intent.PutExtra("title", plates[position].Title);

StartActivity(intent);

}

private void OnBackPressed(object sender, EventArgs args) => base.OnBackPressed();

public override void OnRequestPermissionsResult(int requestCode, string[] permissions,

[GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

}

}

## RecipeActivity.cs

using Android.App;

using Android.Content;

using Android.OS;

using Android.Runtime;

using Android.Support.Design.Widget;

using Android.Support.V4.View;

using Android.Support.V7.App;

using Android.Util;

using Android.Widget;

using ObjectsLibrary;

using System;

using System.Threading.Tasks;

using AndroidApp.Adapters;

using AndroidLibrary;

using XamarinApp;

namespace AndroidApp

{

[Activity(Label = "На плите!", Theme = "@style/AppTheme.NoActionBar", Icon = "@drawable/icon",

ConfigurationChanges =

Android.Content.PM.ConfigChanges.Orientation | Android.Content.PM.ConfigChanges.ScreenSize)]

internal class RecipeActivity : AppCompatActivity

{

private string urlRecipe;

private RecipeFull recipeFull;

private RecipeShort recipeShort;

private Button buttonBack;

private FloatingActionButton buttonStar;

private Button buttonShare;

private Button buttonWeb;

private RelativeLayout relativeLayoutBack;

private TextView title;

private ViewPager viewPager;

private string messageErrorTitle;

private string messageErrorText;

private string messageErrorTextButton;

private Android.Support.V7.App.AlertDialog.Builder errorDialog;

private string GetMessageOnButtonShare() =>

$"Мне понравился рецепт под названием '{recipeFull.Title}'. " +

$"Подробнее ты можешь прочитать о нём по ссылке: {recipeFull.Url}." +

$"Не забудь также попробовать приложение 'На плите' (доступно в Play Market)!";

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

SetContentView(Resource.Layout.activity\_recipe);

title = FindViewById<TextView>(Resource.Id.titleRecipe);

buttonShare = FindViewById<Button>(Resource.Id.shareRecipe);

buttonStar = FindViewById<FloatingActionButton>(Resource.Id.starRecipe);

relativeLayoutBack = FindViewById<RelativeLayout>(Resource.Id.relativeLayoutBack);

viewPager = FindViewById<ViewPager>(Resource.Id.viewPager);

buttonBack = FindViewById<Button>(Resource.Id.buttonBack);

buttonWeb = FindViewById<Button>(Resource.Id.webRecipe);

// Получаем необходимые данные:

urlRecipe = Intent.GetStringExtra("url");

recipeShort = DataContext.ByteArrayToObject<RecipeShort>(Intent.GetByteArrayExtra("recipeShort"));

string[] messagesError = Resources.GetStringArray(Resource.Array.messagesError);

messageErrorTitle = messagesError[0];

messageErrorText = messagesError[1];

messageErrorTextButton = messagesError[2];

errorDialog = new Android.Support.V7.App.AlertDialog.Builder(this)

.SetTitle(messageErrorTitle)

.SetMessage(messageErrorText)

.SetCancelable(false)

.SetNeutralButton(messageErrorTextButton, OnBackPressed);

}

protected override async void OnStart()

{

base.OnStart();

// Проверка на избранный рецепт:

if (RecipeData.ExistsRecipe(urlRecipe))

// Если избранный, то сделать звёздочку закрашенной.

buttonStar.SetBackgroundResource(Resources.GetIdentifier("outline\_favorite\_24", "drawable", PackageName));

try

{

recipeFull = await LoadRecipe(urlRecipe);

title.Text = recipeFull.Title;

viewPager.Adapter = (new PagerRecipeAdapter(SupportFragmentManager, recipeFull));

}

catch (AndroidException exp)

{

errorDialog.SetMessage(exp.Message);

}

// Обработка случая, когда рецепт не загружается:

catch

{

errorDialog.SetMessage(messageErrorText);

errorDialog.Show();

}

buttonShare.Click += OnShareButton;

buttonStar.Click += OnAddFavRecipe;

relativeLayoutBack.Click += OnBackPressed;

buttonBack.Click += OnBackPressed;

buttonWeb.Click += OnStartWeb;

}

private void OnBackPressed(object sender, EventArgs args) => base.OnBackPressed();

private void OnStartWeb(object sender, EventArgs args)

{

Android.Net.Uri address = Android.Net.Uri.Parse(recipeFull.Url);

Intent openLinkIntent = new Intent(Intent.ActionView, address);

if (openLinkIntent.ResolveActivity(PackageManager) != null)

{

StartActivity(openLinkIntent);

}

}

private void OnAddFavRecipe(object sender, EventArgs args)

{

if (!RecipeData.ExistsRecipe(urlRecipe))

{

Toast.MakeText(this, "Рецепт добавлен в избранное", ToastLength.Short).Show();

RecipeData.SaveRecipe(recipeShort);

buttonStar.SetImageResource(Resources.GetIdentifier("outline\_favorite\_24", "drawable", PackageName));

return;

}

// else:

RecipeData.DeleteRecipe(urlRecipe);

buttonStar.SetImageResource(Resources.GetIdentifier("outline\_favorite\_border\_24", "drawable",

PackageName));

}

private void OnShareButton(object sender, EventArgs args)

{

Intent intent = new Intent(Intent.ActionSend);

intent.SetType("text/plain");

String textToSend = GetMessageOnButtonShare();

intent.PutExtra(Intent.ExtraText, textToSend);

try

{

StartActivity(Intent.CreateChooser(intent, "Поделиться рецептом."));

}

catch (Android.Content.ActivityNotFoundException)

{

Toast.MakeText(ApplicationContext, "Some error", ToastLength.Short).Show();

}

}

/// <summary>

/// Загрузка рецепта с сервера.

/// </summary>

/// <param name="url">Интернет адрес рецепта.</param>

/// <returns><see cref="RecipeFull"/></returns>

private static async Task<RecipeFull> LoadRecipe(string url)

{

return await Task.Run(() => HttpContext.GetRecipe(url));

}

public override void OnRequestPermissionsResult(int requestCode, string[] permissions,

[GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

}

}

## SavedIngredientsActivity.cs

using Android.App;

using Android.Content;

using Android.OS;

using Android.Runtime;

using Android.Support.Design.Widget;

using Android.Support.V4.View;

using Android.Support.V7.App;

using Android.Util;

using Android.Widget;

using ObjectsLibrary;

using System;

using System.Threading.Tasks;

using AndroidApp.Adapters;

using AndroidLibrary;

using XamarinApp;

namespace AndroidApp

{

[Activity(Label = "На плите!", Theme = "@style/AppTheme.NoActionBar", Icon = "@drawable/icon",

ConfigurationChanges =

Android.Content.PM.ConfigChanges.Orientation | Android.Content.PM.ConfigChanges.ScreenSize)]

internal class RecipeActivity : AppCompatActivity

{

private string urlRecipe;

private RecipeFull recipeFull;

private RecipeShort recipeShort;

private Button buttonBack;

private FloatingActionButton buttonStar;

private Button buttonShare;

private Button buttonWeb;

private RelativeLayout relativeLayoutBack;

private TextView title;

private ViewPager viewPager;

private string messageErrorTitle;

private string messageErrorText;

private string messageErrorTextButton;

private Android.Support.V7.App.AlertDialog.Builder errorDialog;

private string GetMessageOnButtonShare() =>

$"Мне понравился рецепт под названием '{recipeFull.Title}'. " +

$"Подробнее ты можешь прочитать о нём по ссылке: {recipeFull.Url}." +

$"Не забудь также попробовать приложение 'На плите' (доступно в Play Market)!";

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

SetContentView(Resource.Layout.activity\_recipe);

title = FindViewById<TextView>(Resource.Id.titleRecipe);

buttonShare = FindViewById<Button>(Resource.Id.shareRecipe);

buttonStar = FindViewById<FloatingActionButton>(Resource.Id.starRecipe);

relativeLayoutBack = FindViewById<RelativeLayout>(Resource.Id.relativeLayoutBack);

viewPager = FindViewById<ViewPager>(Resource.Id.viewPager);

buttonBack = FindViewById<Button>(Resource.Id.buttonBack);

buttonWeb = FindViewById<Button>(Resource.Id.webRecipe);

// Получаем необходимые данные:

urlRecipe = Intent.GetStringExtra("url");

recipeShort = DataContext.ByteArrayToObject<RecipeShort>(Intent.GetByteArrayExtra("recipeShort"));

string[] messagesError = Resources.GetStringArray(Resource.Array.messagesError);

messageErrorTitle = messagesError[0];

messageErrorText = messagesError[1];

messageErrorTextButton = messagesError[2];

errorDialog = new Android.Support.V7.App.AlertDialog.Builder(this)

.SetTitle(messageErrorTitle)

.SetMessage(messageErrorText)

.SetCancelable(false)

.SetNeutralButton(messageErrorTextButton, OnBackPressed);

}

protected override async void OnStart()

{

base.OnStart();

// Проверка на избранный рецепт:

if (RecipeData.ExistsRecipe(urlRecipe))

// Если избранный, то сделать звёздочку закрашенной.

buttonStar.SetBackgroundResource(Resources.GetIdentifier("outline\_favorite\_24", "drawable", PackageName));

try

{

recipeFull = await LoadRecipe(urlRecipe);

title.Text = recipeFull.Title;

viewPager.Adapter = (new PagerRecipeAdapter(SupportFragmentManager, recipeFull));

}

catch (AndroidException exp)

{

errorDialog.SetMessage(exp.Message);

}

// Обработка случая, когда рецепт не загружается:

catch

{

errorDialog.SetMessage(messageErrorText);

errorDialog.Show();

}

buttonShare.Click += OnShareButton;

buttonStar.Click += OnAddFavRecipe;

relativeLayoutBack.Click += OnBackPressed;

buttonBack.Click += OnBackPressed;

buttonWeb.Click += OnStartWeb;

}

private void OnBackPressed(object sender, EventArgs args) => base.OnBackPressed();

private void OnStartWeb(object sender, EventArgs args)

{

Android.Net.Uri address = Android.Net.Uri.Parse(recipeFull.Url);

Intent openLinkIntent = new Intent(Intent.ActionView, address);

if (openLinkIntent.ResolveActivity(PackageManager) != null)

{

StartActivity(openLinkIntent);

}

}

private void OnAddFavRecipe(object sender, EventArgs args)

{

if (!RecipeData.ExistsRecipe(urlRecipe))

{

Toast.MakeText(this, "Рецепт добавлен в избранное", ToastLength.Short).Show();

RecipeData.SaveRecipe(recipeShort);

buttonStar.SetImageResource(Resources.GetIdentifier("outline\_favorite\_24", "drawable", PackageName));

return;

}

// else:

RecipeData.DeleteRecipe(urlRecipe);

buttonStar.SetImageResource(Resources.GetIdentifier("outline\_favorite\_border\_24", "drawable",

PackageName));

}

private void OnShareButton(object sender, EventArgs args)

{

Intent intent = new Intent(Intent.ActionSend);

intent.SetType("text/plain");

String textToSend = GetMessageOnButtonShare();

intent.PutExtra(Intent.ExtraText, textToSend);

try

{

StartActivity(Intent.CreateChooser(intent, "Поделиться рецептом."));

}

catch (Android.Content.ActivityNotFoundException)

{

Toast.MakeText(ApplicationContext, "Some error", ToastLength.Short).Show();

}

}

/// <summary>

/// Загрузка рецепта с сервера.

/// </summary>

/// <param name="url">Интернет адрес рецепта.</param>

/// <returns><see cref="RecipeFull"/></returns>

private static async Task<RecipeFull> LoadRecipe(string url)

{

return await Task.Run(() => HttpContext.GetRecipe(url));

}

public override void OnRequestPermissionsResult(int requestCode, string[] permissions,

[GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

}

}

## SavedRecipesActivity.cs

using System;

using System.Collections.Generic;

using System.Linq;

using Android.Widget;

using ObjectsLibrary;

using AndroidLibrary;

using Android.App;

using Android.Content;

using Android.OS;

using Android.Runtime;

using Android.Support.V7.App;

using Android.Support.V7.Widget;

using static Android.Support.V7.Widget.SearchView;

using XamarinApp;

namespace AndroidApp

{

[Activity(Label = "На плите!", Theme = "@style/AppTheme.NoActionBar", Icon = "@drawable/icon",

ConfigurationChanges =

Android.Content.PM.ConfigChanges.Orientation | Android.Content.PM.ConfigChanges.ScreenSize)]

internal class SavedRecipesActivity : AppCompatActivity

{

private RecipeAdapter \_adapter;

private RecyclerView recyclerView;

private Android.Support.V7.Widget.SearchView \_searchView;

private LinearLayoutManager linearLayoutManager;

private List<RecipeShort> recipeShorts;

private List<RecipeShort> dataRecipes;

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

SetContentView(Resource.Layout.activity\_savedRecipes);

var relativeLayoutBack = FindViewById<RelativeLayout>(Resource.Id.relativeLayoutBack);

var buttonBack = FindViewById<Button>(Resource.Id.buttonBack);

buttonBack.SetBackgroundResource(

Resources.GetIdentifier("round\_arrow\_back\_ios\_24", "drawable", PackageName));

relativeLayoutBack.Click += new EventHandler((sender, args) => { base.OnBackPressed(); });

buttonBack.Click += new EventHandler((sender, args) => { base.OnBackPressed(); });

\_searchView = FindViewById<Android.Support.V7.Widget.SearchView>(Resource.Id.searchViewFind);

\_searchView.QueryTextChange += new EventHandler<QueryTextChangeEventArgs>((sender, args) =>

{

var adapter =

new RecipeAdapter(

dataRecipes = this.recipeShorts

.Where(x => x != null && x.Title.ToLower().Contains(args.NewText.ToLower())).ToList(),

this);

adapter.ItemClick += OnItemClick;

\_adapter = adapter;

recyclerView.SetAdapter(adapter);

});

recyclerView = FindViewById<RecyclerView>(Resource.Id.recyclerViewRecipes);

linearLayoutManager = new LinearLayoutManager(this);

recyclerView.SetLayoutManager(linearLayoutManager);

recipeShorts = RecipeData.GetArrayRecipes().ToList();

if (recipeShorts != null)

{

var adapter = new RecipeAdapter(this.recipeShorts, this);

adapter.ItemClick += OnItemClick;

\_adapter = adapter;

recyclerView.SetAdapter(adapter);

dataRecipes = recipeShorts;

}

}

public override void OnRequestPermissionsResult(int requestCode, string[] permissions,

[GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

private void OnItemClick(object sender, int position)

{

Intent intent = new Intent(this, typeof(RecipeActivity));

intent.PutExtra("url", dataRecipes[position].Url);

intent.PutExtra("recipeShort", DataContext.RecipeToByteArray(dataRecipes[position]));

StartActivity(intent);

}

}

}

## SectionRecipesActivity.cs

using System;

using System.Collections.Generic;

using Android.Widget;

using AndroidLibrary;

using Android.App;

using Android.OS;

using Android.Runtime;

using Android.Support.V7.App;

using Android.Support.V7.Widget;

using Android.Content;

using Android.Support.V4.Widget;

using ObjectsLibrary;

using Android.Graphics;

using System.Threading.Tasks;

using XamarinApp;

using System.Linq;

namespace AndroidApp

{

[Activity(Label = "На плите!", Theme = "@style/AppTheme.NoActionBar", Icon = "@drawable/icon",

ConfigurationChanges =

Android.Content.PM.ConfigChanges.Orientation | Android.Content.PM.ConfigChanges.ScreenSize)]

internal class SectionRecipesActivity : AppCompatActivity

{

private List<RecipeShort> recipeShorts;

private SwipeRefreshLayout swipeRefreshLayout;

private RecyclerView recyclerView;

private RecipeAdapter recipeAdapter;

private LinearLayoutManager linearLayoutManager;

private RecipeScrollListener recipeListener;

private RelativeLayout relativeLayoutBack;

private Button buttonBack;

private TextView textViewTitle;

private string title;

private string lastQuery;

private int page = 1;

private string section;

private string messageErrorTitle;

private string messageErrorText;

private string messageErrorTextButton = "Назад";

private Android.Support.V7.App.AlertDialog.Builder errorDialog;

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

SetContentView(Resource.Layout.activity\_sections);

relativeLayoutBack = FindViewById<RelativeLayout>(Resource.Id.relativeLayoutBack);

buttonBack = FindViewById<Button>(Resource.Id.buttonBack);

textViewTitle = FindViewById<TextView>(Resource.Id.textView\_title);

recyclerView = FindViewById<RecyclerView>(Resource.Id.recyclerViewRecipesSections);

swipeRefreshLayout = FindViewById<SwipeRefreshLayout>(Resource.Id.swipeRefreshLayout);

section = Intent.GetStringExtra("key");

title = Intent.GetStringExtra("title");

lastQuery = $"section={section}&page=1";

textViewTitle.Text = "Раздел: " + title;

relativeLayoutBack.Click += OnBackPressed;

buttonBack.Click += OnBackPressed;

linearLayoutManager = new LinearLayoutManager(this);

recipeListener = new RecipeScrollListener(linearLayoutManager);

swipeRefreshLayout.SetColorSchemeColors(Color.Orange, Color.DarkOrange);

swipeRefreshLayout.Refresh += RefreshLayout;

recyclerView.AddOnScrollListener(recipeListener);

recipeListener.LoadMoreEvent += LoadMoreElements;

string[] messagesError = Resources.GetStringArray(Resource.Array.messagesError);

messageErrorTitle = messagesError[0];

messageErrorText = messagesError[1];

errorDialog = new Android.Support.V7.App.AlertDialog.Builder(this)

.SetTitle(messageErrorTitle)

.SetMessage(messageErrorText)

.SetCancelable(false)

.SetNeutralButton(messageErrorTextButton, OnBackPressed);

UpdateListView();

}

private void LoadMoreElements(object sender, EventArgs e)

{

// Получаем строку для нового запроса:

string query = lastQuery.Substring(0, lastQuery.IndexOf("page=") + 5) + (++page);

// Обновляем коллекцию:

UpdateListView(query, recipeShorts);

}

private void RefreshLayout(object sender, EventArgs e)

{

UpdateListView($"section={section}&page=1");

}

private async void UpdateListView(string query = "section={section}", List<RecipeShort> recipeShorts = null)

{

query = query.Replace("{section}", $"{section}");

// Запустить кружочек.

swipeRefreshLayout.Post(() =>

{

swipeRefreshLayout.Refreshing = true;

recyclerView.Clickable = false;

});

recyclerView.SetLayoutManager(linearLayoutManager);

try

{

if (recipeShorts == null)

{

this.recipeShorts = await UpdateCollectionRecipes(query);

recipeAdapter = new RecipeAdapter(this.recipeShorts, this);

recipeAdapter.ItemClick += OnRecipeClick;

recyclerView.SetAdapter(recipeAdapter);

}

else

{

var newRecipes = await UpdateCollectionRecipes(query);

recipeAdapter.AddItems(newRecipes);

recipeShorts.AddRange(newRecipes);

}

}

catch (Exception exp)

{

errorDialog.SetMessage(exp.Message);

errorDialog.Show();

}

finally

{

// Остановить кружочек.

swipeRefreshLayout.Post(() =>

{

swipeRefreshLayout.Refreshing = false;

recyclerView.Clickable = true;

});

}

}

private async Task<List<RecipeShort>> UpdateCollectionRecipes(string query)

{

return (await Task.Run(function: () => HttpContext.GetPages(query))).ToList();

}

private void OnRecipeClick(object sender, int position)

{

Intent intent = new Intent(this, typeof(RecipeActivity));

intent.PutExtra("url", recipeShorts[position].Url);

intent.PutExtra("recipeShort", DataContext.RecipeToByteArray(recipeShorts[position]));

StartActivity(intent);

}

private void OnBackPressed(object sender, EventArgs args) => base.OnBackPressed();

public override void OnRequestPermissionsResult(int requestCode, string[] permissions,

[GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

}

}

## IngredientsAdapter.cs

using Android.Content;

using Android.Views;

using Android.Widget;

using ObjectsLibrary;

using ObjectsLibrary.Components;

using AndroidLibrary;

using XamarinApp;

namespace AndroidApp

{

public class IngredientsAdapter : BaseAdapter<RecipeFull>

{

private readonly RecipeFull \_recipeFull;

private readonly Context \_context;

private readonly Ingredient[] \_ingredients;

public IngredientsAdapter(Context context, RecipeFull recipeFull)

{

\_recipeFull = recipeFull;

\_context = context;

if (recipeFull.Ingredients != null)

\_ingredients = recipeFull.Ingredients;

}

public override long GetItemId(int position)

{

return position;

}

public override View GetView(int position, View convertView, ViewGroup parent)

{

convertView = LayoutInflater.From(\_context).Inflate(Resource.Layout.list\_ingredients, null, false);

var ingredientName = convertView.FindViewById<TextView>(Resource.Id.ingredientName);

var ingredientUnit = convertView.FindViewById<TextView>(Resource.Id.ingredientUnit);

var checkBox = convertView.FindViewById<CheckBox>(Resource.Id.ingredientCheck);

ingredientName.Text = \_ingredients[position].Name;

ingredientUnit.Text = \_ingredients[position].Unit;

checkBox.CheckedChange += (sender, e) =>

{

if (e.IsChecked)

{

Toast.MakeText(\_context, "Ингредиент добавлен в корзину продуктов!", ToastLength.Short).Show();

IngredientData.SaveIngredient(\_ingredients[position]);

}

else

{

Toast.MakeText(\_context, "Ингредиент убран из корзины продуктов!", ToastLength.Short).Show();

IngredientData.DeleteIngredient(\_ingredients[position]);

}

};

if (IngredientData.ExistsIngredient(\_ingredients[position]))

checkBox.Checked = true;

else

checkBox.Checked = false;

return convertView;

}

public override int Count => \_ingredients.Length;

public override RecipeFull this[int position] => \_recipeFull;

}

}

## PagerRecipeAdapter.cs

using System;

using Android.OS;

using Android.Runtime;

using Android.Support.V4.App;

using AndroidApp.Fragments;

using Java.Lang;

using ObjectsLibrary;

using AndroidLibrary;

namespace AndroidApp.Adapters

{

public class PagerRecipeAdapter : FragmentPagerAdapter

{

private readonly RecipeFull \_recipeFull;

private readonly string[] pagesName = {"Описание", "Ингредиенты", "Пошаговый рецепт"};

public PagerRecipeAdapter(FragmentManager fm, RecipeFull recipeFull) : base(fm)

{

\_recipeFull = recipeFull;

}

public override int Count { get; } = 3;

public override Fragment GetItem(int position)

{

var arguments = new Bundle();

arguments.PutByteArray("recipeFull", DataContext.RecipeToByteArray(\_recipeFull));

var fragment = position switch

{

0 => (Fragment) new RecipeDescriptionFragment(),

1 => new RecipeIngredientsFragment(),

2 => new RecipeStepsFragment(),

\_ => new RecipeDescriptionFragment()

};

fragment.Arguments = arguments;

return fragment;

}

public override ICharSequence GetPageTitleFormatted(int position)

{

return new Java.Lang.String(pagesName[position]);

}

}

}

## PlateAdapter.cs

using Android.App;

using Android.Support.V7.Widget;

using Android.Views;

using XamarinApp.ViewHolders;

using System;

using System.Collections.Generic;

using XamarinApp;

using AndroidApp.General;

namespace AndroidApp

{

public class PlateAdapter : RecyclerView.Adapter

{

private readonly List<Plate> \_items;

private readonly Activity \_activity;

public event EventHandler<int> ItemClick;

public override int ItemCount => \_items.Count;

public PlateAdapter(List<Plate> plates, Activity activity)

{

\_activity = activity;

\_items = plates;

}

public override void OnBindViewHolder(RecyclerView.ViewHolder holder, int position)

{

PlateViewHolder vh = holder as PlateViewHolder;

vh.Title.Text = \_items[position].Title;

vh.Background.SetBackgroundResource(\_activity.Resources.GetIdentifier(\_items[position].Background,

"drawable", \_activity.PackageName));

}

public override RecyclerView.ViewHolder OnCreateViewHolder(ViewGroup parent, int viewType)

{

View itemView = LayoutInflater.From(parent.Context).Inflate(Resource.Layout.list\_plates, parent, false);

PlateViewHolder vh = new PlateViewHolder(itemView, OnClick);

return vh;

}

private void OnClick(int position)

{

ItemClick?.Invoke(this, position);

}

}

}

## RecipeAdapter.cs

using Android.Support.V7.Widget;

using Android.Views;

using ObjectsLibrary;

using Square.Picasso;

using System;

using System.Collections.Generic;

using Android.App;

using XamarinApp;

using XamarinApp.ViewHolders;

namespace AndroidApp

{

public class RecipeAdapter : RecyclerView.Adapter

{

private readonly List<RecipeShort> \_items;

private readonly Activity \_activity;

public event EventHandler<int> ItemClick;

public override int ItemCount => \_items.Count;

public RecipeAdapter(List<RecipeShort> recipeShorts, Activity activity)

{

\_activity = activity;

\_items = recipeShorts;

}

public void AddItems(List<RecipeShort> recipes)

{

\_items.AddRange(recipes);

NotifyDataSetChanged();

}

public override void OnBindViewHolder(RecyclerView.ViewHolder holder, int position)

{

RecipeViewHolder vh = holder as RecipeViewHolder;

vh.Title.Text = \_items[position]?.Title;

var urlArray = \_items[position]?.Url?.Split('/');

if (urlArray != null && urlArray.Length >= 2)

vh.Link.Text = urlArray[2];

var url = \_items[position]?.Image?.ImageUrl;

if (!string.IsNullOrEmpty(url))

Picasso.With(\_activity)

.Load(url)

.Into(vh.Image);

}

public override RecyclerView.ViewHolder OnCreateViewHolder(ViewGroup parent, int viewType)

{

View itemView = LayoutInflater.From(parent.Context).Inflate(Resource.Layout.list\_item, parent, false);

RecipeViewHolder vh = new RecipeViewHolder(itemView, OnClick);

return vh;

}

void OnClick(int position)

{

ItemClick?.Invoke(this, position);

}

}

}

## SavedIngredientsAdapter.cs

using Android.Support.V7.Widget;

using Android.Views;

using ObjectsLibrary.Components;

using XamarinApp;

using System.Collections.Generic;

using AndroidLibrary;

using XamarinApp.ViewHolders;

namespace AndroidApp

{

public class SavedIngredientsAdapter : RecyclerView.Adapter

{

private readonly List<Ingredient> \_items;

public override int ItemCount => \_items.Count;

public SavedIngredientsAdapter(List<Ingredient> ingredients)

{

\_items = ingredients;

}

public override void OnBindViewHolder(RecyclerView.ViewHolder holder, int position)

{

SavedIngredientsViewHolder vh = holder as SavedIngredientsViewHolder;

vh.Title.Text = \_items[position].Name;

vh.Unit.Text = \_items[position].Unit;

vh.RecipeName.Text = \_items[position].RecipeName;

}

private void RemoveItem(int position)

{

NotifyItemRemoved(position);

IngredientData.DeleteIngredient(\_items[position]);

\_items.RemoveAt(position);

}

public override RecyclerView.ViewHolder OnCreateViewHolder(ViewGroup parent, int viewType)

{

var itemView = LayoutInflater.From(parent.Context)

.Inflate(Resource.Layout.list\_ingredients\_saved, parent, false);

var vh = new SavedIngredientsViewHolder(itemView, RemoveItem);

return vh;

}

}

}

## StepAdapter.cs

using Android.Content;

using Android.Views;

using Android.Widget;

using ObjectsLibrary;

using ObjectsLibrary.Components;

using Square.Picasso;

using System.Collections.Generic;

using XamarinApp;

namespace AndroidApp

{

public class StepAdapter : BaseAdapter<RecipeFull>

{

private readonly RecipeFull \_recipeFull;

private readonly Context \_context;

private readonly List<StepRecipe> \_stepRecipeBoxes;

public StepAdapter(Context context, RecipeFull recipeFull)

{

\_context = context;

\_recipeFull = recipeFull;

\_stepRecipeBoxes = new List<StepRecipe>();

if (recipeFull.StepsRecipe != null)

\_stepRecipeBoxes.AddRange(recipeFull.StepsRecipe);

}

public override long GetItemId(int position)

{

return position;

}

public override View GetView(int position, View convertView, ViewGroup parent)

{

View view = LayoutInflater.From(\_context).Inflate(Resource.Layout.list\_steps, null, false);

var stepDescription = view.FindViewById<TextView>(Resource.Id.stepDescription);

stepDescription.Text = \_stepRecipeBoxes[position].Description;

var imageView = view.FindViewById<ImageView>(Resource.Id.stepImage);

var url = \_stepRecipeBoxes[position].Image.ImageUrl;

if (!string.IsNullOrEmpty(url))

Picasso.With(\_context)

.Load(url)

.Into(imageView);

return view;

}

public override int Count => \_stepRecipeBoxes.Count;

public override RecipeFull this[int position] => \_recipeFull;

}

}

## RecipeDescriptionFragment.cs

using System;

using Android.OS;

using XamarinApp;

using Android.Views;

using Android.Webkit;

using Android.Widget;

using ObjectsLibrary;

using Square.Picasso;

using AndroidLibrary;

namespace AndroidApp.Fragments

{

public class RecipeDescriptionFragment : Android.Support.V4.App.Fragment

{

private RecipeFull \_recipeFull;

private ImageView \_image;

private TextView \_description;

private TextView \_cpfcRecipe;

private TextView \_authorNameRecipe;

private TextView \_additionalInfoRecipe;

private TextView \_urlRecipe;

private WebView \_videoView;

public override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

var arguments = Arguments;

if (arguments != null)

{

\_recipeFull = DataContext.ByteArrayToObject<RecipeFull>(arguments.GetByteArray("recipeFull"));

}

}

public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

{

var view = inflater.Inflate(Resource.Layout.fragment\_recipe\_description, container, false);

\_image = view.FindViewById<ImageView>(Resource.Id.imageMainRecipe);

\_description = view.FindViewById<TextView>(Resource.Id.titleMainDescription);

\_cpfcRecipe = view.FindViewById<TextView>(Resource.Id.CPFCRecipe);

\_authorNameRecipe = view.FindViewById<TextView>(Resource.Id.authorNameRecipe);

\_additionalInfoRecipe = view.FindViewById<TextView>(Resource.Id.additionalInfoRecipe);

\_urlRecipe = view.FindViewById<TextView>(Resource.Id.urlRecipe);

\_videoView = view.FindViewById<WebView>(Resource.Id.videoRecipe);

if (string.IsNullOrEmpty(\_recipeFull.Additional?.VideoUrl))

{

// Если видео нет:

\_videoView.LayoutParameters.Height = 0;

\_videoView.RefreshDrawableState();

\_videoView.RequestLayout();

}

return view;

}

public override void OnActivityCreated(Bundle savedInstanceState)

{

if (!string.IsNullOrEmpty(\_recipeFull.TitleImage?.ImageUrl))

Picasso.With(View.Context)

.Load(\_recipeFull.TitleImage.ImageUrl)

.Into(\_image);

\_description.Text += \_recipeFull.Description;

if (\_recipeFull.Additional?.CPFC != null)

{

if (\_recipeFull.Additional.CPFC.Calories != 0)

\_cpfcRecipe.Text +=

$"Калории: {\_recipeFull.Additional.CPFC.Calories} Ккал.{System.Environment.NewLine}";

if (\_recipeFull.Additional.CPFC.Proteins != 0)

\_cpfcRecipe.Text += $"Белки: {\_recipeFull.Additional.CPFC.Proteins} г.{System.Environment.NewLine}";

if (\_recipeFull.Additional.CPFC.Fats != 0)

\_cpfcRecipe.Text += $"Жиры: {\_recipeFull.Additional.CPFC.Fats} г.{System.Environment.NewLine}";

if (\_recipeFull.Additional.CPFC.Carbohydrates != 0)

\_cpfcRecipe.Text +=

$"Углеводы: {\_recipeFull.Additional.CPFC.Carbohydrates} г.{System.Environment.NewLine}";

}

if (\_recipeFull.Additional != null && !string.IsNullOrEmpty(\_recipeFull.Additional.AuthorName))

\_authorNameRecipe.Text = $"Рецепт от: {\_recipeFull.Additional.AuthorName}";

if (\_recipeFull.Additional?.CountPortions != 0)

\_additionalInfoRecipe.Text +=

$"Количество порций: {\_recipeFull.Additional.CountPortions}.{System.Environment.NewLine}";

if (\_recipeFull.Additional?.PrepMinutes != 0)

\_additionalInfoRecipe.Text += $"Количество минут на готовку: {\_recipeFull.Additional.PrepMinutes} мин.";

\_urlRecipe.Text = $"Ссылка на рецепт: {\_recipeFull.Url}";

if (!string.IsNullOrEmpty(\_recipeFull.Additional?.VideoUrl))

{

String videoSource = \_recipeFull.Additional.VideoUrl;

WebSettings webSettings = \_videoView.Settings;

webSettings.JavaScriptEnabled = true;

\_videoView.LoadUrl(videoSource);

}

base.OnActivityCreated(savedInstanceState);

}

}

}

## RecipeIngredientsFragment.cs

using XamarinApp;

using Android.OS;

using Android.Views;

using Android.Widget;

using ObjectsLibrary;

using AndroidLibrary;

namespace AndroidApp.Fragments

{

public class RecipeIngredientsFragment : Android.Support.V4.App.Fragment

{

private RecipeFull \_recipeFull;

private ListView \_listIngredients;

public override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

var arguments = Arguments;

if (arguments != null)

{

\_recipeFull = DataContext.ByteArrayToObject<RecipeFull>(arguments.GetByteArray("recipeFull"));

}

}

public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

{

var view = inflater.Inflate(Resource.Layout.fragment\_recipe\_ingredients, container, false);

\_listIngredients = view.FindViewById<ListView>(Resource.Id.listIngredients);

return view;

}

public override void OnActivityCreated(Bundle savedInstanceState)

{

var listIngredientsAdapter = new IngredientsAdapter(View.Context, \_recipeFull);

if (listIngredientsAdapter.Count > 0)

\_listIngredients.Adapter = listIngredientsAdapter;

base.OnActivityCreated(savedInstanceState);

}

}

}

## RecipeStepsFragment.cs

using XamarinApp;

using Android.OS;

using Android.Views;

using Android.Widget;

using ObjectsLibrary;

using AndroidLibrary;

namespace AndroidApp.Fragments

{

public class RecipeStepsFragment : Android.Support.V4.App.Fragment

{

private RecipeFull \_recipeFull;

private ListView \_listSteps;

public override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

var arguments = Arguments;

if (arguments != null)

{

\_recipeFull = DataContext.ByteArrayToObject<RecipeFull>(arguments.GetByteArray("recipeFull"));

}

}

public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

{

var view = inflater.Inflate(Resource.Layout.fragment\_recipe\_steps, container, false);

\_listSteps = view.FindViewById<ListView>(Resource.Id.listSteps);

return view;

}

public override void OnActivityCreated(Bundle savedInstanceState)

{

var adapterStep = new StepAdapter(View.Context, \_recipeFull);

\_listSteps.Adapter = adapterStep;

base.OnActivityCreated(savedInstanceState);

}

}

}

## Plate.cs

namespace AndroidApp.General

{

public class Plate

{

public string Title { get; }

public string Background { get; }

public string Key { get; }

public Plate(string title, string background, string key)

{

Title = title;

Background = background;

Key = key;

}

}

}

## RecipeScrollListener.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Android.App;

using Android.Content;

using Android.OS;

using Android.Runtime;

using Android.Support.V7.Widget;

using Android.Views;

using Android.Widget;

namespace AndroidApp

{

public class RecipeScrollListener : RecyclerView.OnScrollListener

{

private LinearLayoutManager layoutManager;

public event EventHandler LoadMoreEvent;

public RecipeScrollListener(LinearLayoutManager layoutManager)

{

this.layoutManager = layoutManager;

}

public override void OnScrolled(RecyclerView recyclerView, int dx, int dy)

{

bool isLoading = false;

base.OnScrolled(recyclerView, dx, dy);

var visibleItemCount = recyclerView.ChildCount;

var totalItemCount = recyclerView.GetAdapter().ItemCount;

var pastVisiblesItems = layoutManager.FindFirstVisibleItemPosition();

if ((visibleItemCount + pastVisiblesItems) >= totalItemCount - 4 && !isLoading && recyclerView.Clickable)

{

isLoading = true;

LoadMoreEvent?.Invoke(this, null);

// Loading code.

isLoading = false;

}

}

}

}

## PlateViewHolder.cs

using System;

using Android.Support.V7.Widget;

using Android.Views;

using Android.Widget;

namespace XamarinApp.ViewHolders

{

public class PlateViewHolder : RecyclerView.ViewHolder

{

public RelativeLayout Background { get; }

public TextView Title { get; }

public PlateViewHolder(View itemView, Action<int> listener) : base(itemView)

{

Title = itemView.FindViewById<TextView>(Resource.Id.textView\_title);

Background = itemView.FindViewById<RelativeLayout>(Resource.Id.relativeLayout\_background);

itemView.Click += (sender, e) => listener(base.LayoutPosition);

}

}

}

## RecipeViewHolder.cs

using System;

using Android.Support.V7.Widget;

using Android.Views;

using Android.Widget;

namespace XamarinApp.ViewHolders

{

public class RecipeViewHolder : RecyclerView.ViewHolder

{

public ImageView Image { get; }

public TextView Title { get; }

public TextView Link { get; }

public RecipeViewHolder(View itemView, Action<int> listener) : base(itemView)

{

Image = itemView.FindViewById<ImageView>(Resource.Id.imageTitle);

Title = itemView.FindViewById<TextView>(Resource.Id.title);

Link = itemView.FindViewById<TextView>(Resource.Id.textLink);

itemView.Click += (sender, e) => listener(base.LayoutPosition);

}

}

}

## SavedIngredientsViewHolder.cs

using System;

using Android.Support.V7.Widget;

using Android.Views;

using Android.Widget;

namespace XamarinApp.ViewHolders

{

public class SavedIngredientsViewHolder : RecyclerView.ViewHolder

{

public TextView Title { get; }

public TextView Unit { get; }

public TextView RecipeName { get; }

public SavedIngredientsViewHolder(View itemView, Action<int> listener) : base(itemView)

{

var CartButton = itemView.FindViewById<Button>(Resource.Id.cart\_button\_saved\_ingredient);

Title = itemView.FindViewById<TextView>(Resource.Id.title\_saved\_ingredient);

Unit = itemView.FindViewById<TextView>(Resource.Id.unit\_saved\_ingredient);

RecipeName = itemView.FindViewById<TextView>(Resource.Id.recipe\_saved\_ingredient);

CartButton.Click += (sender, e) => listener(base.LayoutPosition);

}

}

}

## RecipeData.cs

using System;

using System.IO;

using ObjectsLibrary;

using SQLite;

namespace AndroidLibrary

{

public static class RecipeData

{

private static string dbPath =

Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), "recipes.db3");

private static SQLiteConnection dbConnection = new SQLiteConnection(dbPath);

static RecipeData()

{

dbConnection.CreateTable<RecipeTable>();

}

private static string GetFileRecipeName(string url)

{

// Пример: https://www.povarenok.ru/recipes/show/163893/

if (url.Contains("https://www.povarenok.ru"))

{

return "pk" + url[38..^1];

}

// Пример: https://povar.ru/recipes/postnyi\_apelsinovyi\_keks-80038.html

if (url.Contains("https://povar.ru"))

{

return "pr" + url.Split('-')[^1].Split('.')[0];

}

// Пример: https://www.edimdoma.ru/retsepty/137347-syrnik-s-izyumom-i-tsukatami

if (url.Contains("https://www.edimdoma.ru"))

{

return "edm" + url[33..^1].Split('-')[0];

}

// Пример: https://eda.ru/recepty/zavtraki/amerikanskie-bliny-30600

if (url.Contains("https://eda.ru/"))

{

return "eda" + url.Split('-')[^1];

}

return url[8..^1];

}

public static RecipeShort[] GetArrayRecipes()

{

var recipes = dbConnection.Table<RecipeTable>().ToArray();

RecipeShort[] recipeShorts = new RecipeShort[recipes.Length];

for (int i = 0; i < recipes.Length; i++)

{

if (recipes[i].Recipe == null || recipes[i].Name == null)

continue;

recipeShorts[i] = DataContext.ByteArrayToObject<RecipeShort>(recipes[i].Recipe);

}

return recipeShorts;

}

public static bool ExistsRecipe(string url)

{

string fileName = GetFileRecipeName(url);

return dbConnection.Table<RecipeTable>().FirstOrDefault(x => x.Name == fileName) != null;

}

public static void DeleteRecipe(string url)

{

string fileName = GetFileRecipeName(url);

int id = dbConnection.Table<RecipeTable>().First(x => x.Name == fileName).Id;

dbConnection.Delete<RecipeTable>(id);

}

public static void SaveRecipe(RecipeShort recipeShort)

{

string fileName = GetFileRecipeName(recipeShort.Url);

RecipeTable recipeTable = new RecipeTable(fileName, DataContext.RecipeToByteArray(recipeShort));

dbConnection.Insert(recipeTable);

}

}

}

## IngredientData.cs

using ObjectsLibrary.Components;

using SQLite;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace AndroidLibrary

{

public static class IngredientData

{

private static string dbPath =

Path.Combine(System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal),

"ingredients.db3");

private static SQLiteConnection dbConnection = new SQLiteConnection(dbPath);

static IngredientData()

{

dbConnection.CreateTable<IngredientTable>();

}

public static List<Ingredient> GetArrayIngredients()

{

var ingredientsDb = dbConnection.Table<IngredientTable>().ToArray();

List<Ingredient> ingredients = new List<Ingredient>(ingredientsDb.Length);

ingredients.AddRange(from t in ingredientsDb

where t.Name != null && t.Unit != null && t.RecipeName != null

select new Ingredient(t.Name, t.Unit, t.RecipeName));

return ingredients;

}

public static bool ExistsIngredient(Ingredient ingredient)

{

return dbConnection.Table<IngredientTable>().FirstOrDefault(x =>

x.Name == ingredient.Name &&

x.Unit == ingredient.Unit &&

x.RecipeName == ingredient.RecipeName) != null;

}

public static void DeleteIngredient(Ingredient ingredient)

{

var ingredientDb = dbConnection.Table<IngredientTable>().FirstOrDefault(x =>

x.Name == ingredient.Name &&

x.RecipeName == ingredient.RecipeName);

if (ingredientDb == null)

return;

dbConnection.Delete<IngredientTable>(ingredientDb.Id);

}

public static void SaveIngredient(Ingredient ingredient)

{

IngredientTable ingredientTable =

new IngredientTable(ingredient.Name, ingredient.Unit, ingredient.RecipeName);

dbConnection.Insert(ingredientTable);

}

}

}

## HttpContext.cs

using Newtonsoft.Json;

using ObjectsLibrary;

using System.Collections.Generic;

using System.Net;

using System.Net.Http;

using System.Threading.Tasks;

namespace AndroidLibrary

{

public static class HttpContext

{

//http://51.77.58.18/

private const string ipAdress = "http://51.77.58.18/";

private static async Task<string> GetSource(string query)

{

string currentUrl = ipAdress + query;

var client = new HttpClient(new Xamarin.Android.Net.AndroidClientHandler());

var response = await client.GetAsync(currentUrl);

var source = string.Empty;

if (response != null && response.StatusCode == HttpStatusCode.OK)

source = await response.Content.ReadAsStringAsync();

if (response != null && response.StatusCode == HttpStatusCode.NotFound)

throw new HttpRequestException("Не удалось загрузить рецепт.");

if (source == string.Empty)

throw new HttpRequestException("Нет подключения к интернету, попробуйте позже :(");

return source;

}

public static List<RecipeShort> GetPages(string query)

{

string json = GetSource("page/get?" + query).Result;

var obj = JsonConvert.DeserializeObject<List<RecipeShort>>(json);

return obj;

}

public static RecipeFull GetRecipe(string url)

{

string json = GetSource("recipe/get?url=" + url).Result;

var obj = JsonConvert.DeserializeObject<RecipeFull>(json);

return obj;

}

}

}

## DataContext.cs

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace AndroidLibrary

{

public static class DataContext

{

public static T ByteArrayToObject<T>(byte[] arrBytes)

{

var binaryFormatter = new BinaryFormatter();

using var memoryStream = new MemoryStream(arrBytes);

return (T) binaryFormatter.Deserialize(memoryStream);

}

public static byte[] RecipeToByteArray(object obj)

{

if (obj == null)

return null;

using MemoryStream memoryStream = new MemoryStream();

BinaryFormatter binaryFormatter = new BinaryFormatter();

binaryFormatter.Serialize(memoryStream, obj);

return memoryStream.ToArray();

}

}

}

## RecipeTable.cs

using SQLite;

namespace AndroidLibrary

{

[Table("Recipe")]

public class RecipeTable

{

[PrimaryKey, AutoIncrement]

public int Id { get; set; }

public string Name { get; set; }

public byte[] Recipe { get; set; }

public RecipeTable()

{

//

}

public RecipeTable(string name, byte[] recipe) : this()

{

Name = name;

Recipe = recipe;

}

}

}

## IngredientTable.cs

using SQLite;

namespace AndroidLibrary

{

[Table("Recipe")]

public class IngredientTable

{

[PrimaryKey, AutoIncrement]

public int Id { get; set; }

public string Name { get; set;}

public string Unit { get; set; }

public string RecipeName { get; set; }

public IngredientTable()

{

//

}

public IngredientTable(string name, string unit, string recipeName) : this()

{

Name = name;

Unit = unit;

RecipeName = recipeName;

}

}

}

## rectangle\_plate.xml

<selector

xmlns:android="http://schemas.android.com/apk/res/android">

<item>

<shape>

<solid android:color="#ffffff"/>

<corners android:radius="20dp" />

</shape>

</item>

</selector>

## round\_arrow\_back\_ios\_24.xml

<vector xmlns:android="http://schemas.android.com/apk/res/android"

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="@android:color/white">

<path

android:fillColor="@android:color/white"

android:pathData="M16.62,2.99c-0.49,-0.49 -1.28,-0.49 -1.77,0L6.54,11.3c-0.39,0.39 -0.39,1.02 0,1.41l8.31,8.31c0.49,0.49 1.28,0.49 1.77,0s0.49,-1.28 0,-1.77L9.38,12l7.25,-7.25c0.48,-0.48 0.48,-1.28 -0.01,-1.76z"/>

</vector>

## round\_clear\_24.xml

<vector xmlns:android="http://schemas.android.com/apk/res/android"

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="?attr/colorControlNormal">

<path

android:fillColor="@android:color/white"

android:pathData="M18.3,5.71c-0.39,-0.39 -1.02,-0.39 -1.41,0L12,10.59 7.11,5.7c-0.39,-0.39 -1.02,-0.39 -1.41,0 -0.39,0.39 -0.39,1.02 0,1.41L10.59,12 5.7,16.89c-0.39,0.39 -0.39,1.02 0,1.41 0.39,0.39 1.02,0.39 1.41,0L12,13.41l4.89,4.89c0.39,0.39 1.02,0.39 1.41,0 0.39,-0.39 0.39,-1.02 0,-1.41L13.41,12l4.89,-4.89c0.38,-0.38 0.38,-1.02 0,-1.4z"/>

</vector>

## round\_menu\_24.xml

<vector xmlns:android="http://schemas.android.com/apk/res/android"

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="@android:color/white">

<path

android:fillColor="@android:color/white"

android:pathData="M4,18h16c0.55,0 1,-0.45 1,-1s-0.45,-1 -1,-1L4,16c-0.55,0 -1,0.45 -1,1s0.45,1 1,1zM4,13h16c0.55,0 1,-0.45 1,-1s-0.45,-1 -1,-1L4,11c-0.55,0 -1,0.45 -1,1s0.45,1 1,1zM3,7c0,0.55 0.45,1 1,1h16c0.55,0 1,-0.45 1,-1s-0.45,-1 -1,-1L4,6c-0.55,0 -1,0.45 -1,1z"/>

</vector>

## round\_share\_24.xml

<vector xmlns:android="http://schemas.android.com/apk/res/android"

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="@android:color/white">

<path

android:fillColor="@android:color/white"

android:pathData="M18,16.08c-0.76,0 -1.44,0.3 -1.96,0.77L8.91,12.7c0.05,-0.23 0.09,-0.46 0.09,-0.7s-0.04,-0.47 -0.09,-0.7l7.05,-4.11c0.54,0.5 1.25,0.81 2.04,0.81 1.66,0 3,-1.34 3,-3s-1.34,-3 -3,-3 -3,1.34 -3,3c0,0.24 0.04,0.47 0.09,0.7L8.04,9.81C7.5,9.31 6.79,9 6,9c-1.66,0 -3,1.34 -3,3s1.34,3 3,3c0.79,0 1.5,-0.31 2.04,-0.81l7.12,4.16c-0.05,0.21 -0.08,0.43 -0.08,0.65 0,1.61 1.31,2.92 2.92,2.92s2.92,-1.31 2.92,-2.92 -1.31,-2.92 -2.92,-2.92z"/>

</vector>

## round\_timer\_24.xml\

<vector xmlns:android="http://schemas.android.com/apk/res/android"

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="@android:color/white">

<path

android:fillColor="@android:color/white"

android:pathData="M14,1h-4c-0.55,0 -1,0.45 -1,1s0.45,1 1,1h4c0.55,0 1,-0.45 1,-1s-0.45,-1 -1,-1zM12,14c0.55,0 1,-0.45 1,-1L13,9c0,-0.55 -0.45,-1 -1,-1s-1,0.45 -1,1v4c0,0.55 0.45,1 1,1zM19.03,7.39l0.75,-0.75c0.38,-0.38 0.39,-1.01 0,-1.4l-0.01,-0.01c-0.39,-0.39 -1.01,-0.38 -1.4,0l-0.75,0.75C16.07,4.74 14.12,4 12,4c-4.8,0 -8.88,3.96 -9,8.76C2.87,17.84 6.94,22 12,22c4.98,0 9,-4.03 9,-9 0,-2.12 -0.74,-4.07 -1.97,-5.61zM12,20c-3.87,0 -7,-3.13 -7,-7s3.13,-7 7,-7 7,3.13 7,7 -3.13,7 -7,7z"/>

</vector>

## round\_web\_24.xml

<vector xmlns:android="http://schemas.android.com/apk/res/android"

android:width="24dp"

android:height="24dp"

android:viewportWidth="24"

android:viewportHeight="24"

android:tint="@android:color/white">

<path

android:fillColor="@android:color/white"

android:pathData="M20,4L4,4c-1.1,0 -1.99,0.9 -1.99,2L2,18c0,1.1 0.9,2 2,2h16c1.1,0 2,-0.9 2,-2L22,6c0,-1.1 -0.9,-2 -2,-2zM4,9h10.5v3.5L4,12.5L4,9zM4,14.5h10.5L14.5,18L5,18c-0.55,0 -1,-0.45 -1,-1v-2.5zM19,18h-2.5L16.5,9L20,9v8c0,0.55 -0.45,1 -1,1z"/>

</vector>

## spinner.xml

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

<shape xmlns:android="http://schemas.android.com/apk/res/android">

<solid android:color="@android:color/transparent" />

<corners android:radius="5dp" />

<stroke

android:width="1dp"

android:color="@android:color/darker\_gray" />

</shape>

## activity\_info.xml

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:fitsSystemWindows="true">

<LinearLayout

android:id="@+id/rectangle\_2"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:padding="10dp"

android:orientation="horizontal"

android:background="@drawable/rectangle\_2">

<RelativeLayout

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_gravity="left|center\_vertical"

android:id="@+id/relativeLayoutBack">

<Button

android:layout\_width="28dp"

android:layout\_height="28dp"

android:gravity="left|center\_vertical"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

android:background="@drawable/round\_arrow\_back\_ios\_24"

android:id="@+id/buttonBack"/>

</RelativeLayout>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="@android:color/white"

android:textSize="17dp"

android:paddingLeft="20dp"

android:fontFamily="@font/actorregular"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:text="Информация о приложении"

/>

</LinearLayout>

<android.support.v7.widget.CardView

android:layout\_width="match\_parent"

android:layout\_margin="10dp"

android:padding="10dp"

app:cardCornerRadius="4dp"

android:elevation="10dp"

android:layout\_height="match\_parent">

<ScrollView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<LinearLayout

android:layout\_width="match\_parent"

android:orientation="vertical"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/textViewInfo"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="17dp"

android:paddingLeft="20dp"

android:text="О приложении: \nКурсовая работа студента 1 курса БПИ-194 2019-2020 года Романюк Андрея 'Агрегатор кулинарных рецептов на Android.'.\n\nПриложение поддерживает следующие сайты:\npovarenok.ru, povar.ru, eda.ru, edimdoma.ru.\n\n\nЗадиизайнено: Полина Круглик."

android:fontFamily="@font/actorregular"/>

<ImageView

android:layout\_width="300dp"

android:layout\_height="300dp"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="10dp"

android:background="@drawable/cat"/>

</LinearLayout>

</ScrollView>

</android.support.v7.widget.CardView>

</LinearLayout>

## activity\_main.xml

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

<android.support.v4.widget.DrawerLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/drawer\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:fitsSystemWindows="true"

tools:openDrawer="start">

<LinearLayout

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<include

layout="@layout/main\_bar"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="left|center\_vertical"

android:background="@color/colorMain"

android:padding="2dp"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:textSize="16dp"

android:gravity="left|center"

android:paddingBottom="7dp"

android:paddingLeft="3dp"

android:textColor="@android:color/white"

android:text="Фильтр:"/>

<Spinner

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/spinner"

android:layout\_gravity="left"

android:gravity="left"

android:padding="3dp"

android:popupBackground="@color/colorMain"

android:background="@color/colorMain"

android:fontFamily="@font/actorregular"

android:spinnerMode="dropdown"/>

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/keyboard\_arrow\_down\_black\_24dp"/>

</LinearLayout>

<android.support.v4.widget.SwipeRefreshLayout

android:id="@+id/swipeRefreshLayout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<android.support.v7.widget.RecyclerView

android:id="@+id/listRecipeShorts"

android:orientation="vertical"

android:padding="5dp"

android:layout\_margin="5dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</android.support.v4.widget.SwipeRefreshLayout>

</LinearLayout>

<android.support.design.widget.NavigationView

android:id="@+id/nav\_view"

app:itemTextColor="#fff9f0"

app:theme="@style/NavigationViewStyle"

android:background="@color/colorMain"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_gravity="start"

android:fitsSystemWindows="true"

app:headerLayout="@layout/nav\_header\_main"

app:menu="@menu/activity\_main\_drawer"/>

</android.support.v4.widget.DrawerLayout>

## activity\_plates.xml

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:fitsSystemWindows="true">

<LinearLayout

android:id="@+id/rectangle\_2"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:padding="10dp"

android:orientation="horizontal"

android:background="@drawable/rectangle\_2">

<RelativeLayout

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_gravity="left|center\_vertical"

android:id="@+id/relativeLayoutBack">

<Button

android:layout\_width="28dp"

android:layout\_height="28dp"

android:gravity="left|center\_vertical"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

android:background="@drawable/round\_arrow\_back\_ios\_24"

android:id="@+id/buttonBack"/>

</RelativeLayout>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="@android:color/white"

android:textSize="17dp"

android:paddingLeft="20dp"

android:fontFamily="@font/actorregular"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:text="Выберите желаемый раздел"

/>

</LinearLayout>

<android.support.v7.widget.RecyclerView

android:id="@+id/recyclerViewRecipesPlates"

android:orientation="vertical"

android:padding="5dp"

android:layout\_margin="5dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</LinearLayout>

## activity\_recipe.xml

<android.support.design.widget.CoordinatorLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<LinearLayout

android:orientation="horizontal"

android:id="@+id/rectangle\_2"

android:layout\_width="match\_parent"

android:padding="10dp"

android:layout\_height="70dp"

android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true"

android:gravity="left|center"

android:background="@drawable/rectangle\_2">

<RelativeLayout

android:layout\_width="40dp"

android:layout\_height="40dp"

android:id="@+id/relativeLayoutBack">

<Button

android:layout\_width="28dp"

android:layout\_height="28dp"

android:layout\_gravity="right"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

android:background="@drawable/round\_arrow\_back\_ios\_24"

android:id="@+id/buttonBack"/>

</RelativeLayout>

<TextView

android:id="@+id/titleRecipe"

android:gravity="left|center\_vertical"

android:fontFamily="@font/actorregular"

android:layout\_gravity="left|center\_vertical"

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:textColor="#FCFCFC"

android:layout\_weight="1"

android:layout\_margin="2dp"

android:textSize="18dp"

/>

<Button

android:layout\_gravity="right|center\_vertical"

android:layout\_width="26dp"

android:layout\_height="26dp"

android:layout\_margin="4dp"

android:background="@drawable/round\_web\_24"

android:id="@+id/webRecipe"

/>

<Button

android:layout\_gravity="right|center\_vertical"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_margin="5dp"

android:background="@drawable/round\_share\_24"

android:id="@+id/shareRecipe"

/>

</LinearLayout>

<android.support.v4.view.ViewPager

android:layout\_width="match\_parent"

android:id="@+id/viewPager"

android:layout\_height="wrap\_content">

<android.support.v4.view.PagerTabStrip

android:id="@+id/pager\_title\_strip"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="18dp"

android:textColor="@color/colorMain"

android:fontFamily="@font/actorregular"/>

</android.support.v4.view.ViewPager>

</LinearLayout>

<android.support.design.widget.FloatingActionButton

android:id="@+id/starRecipe"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="end|bottom"

app:backgroundTint="@color/colorMain"

android:src="@drawable/outline\_favorite\_border\_24"

android:layout\_margin="16dp"/>

</android.support.design.widget.CoordinatorLayout>

## activity\_savedIngredients.xml

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:fitsSystemWindows="true">

<LinearLayout

android:id="@+id/rectangle\_2"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:padding="10dp"

android:orientation="horizontal"

android:background="@drawable/rectangle\_2">

<RelativeLayout

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_gravity="left|center\_vertical"

android:id="@+id/relativeLayoutBack">

<Button

android:layout\_width="28dp"

android:layout\_height="28dp"

android:gravity="left|center\_vertical"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

android:background="@drawable/round\_arrow\_back\_ios\_24"

android:id="@+id/buttonBack"/>

</RelativeLayout>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="@android:color/white"

android:textSize="17dp"

android:paddingLeft="20dp"

android:fontFamily="@font/actorregular"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:text="Корзина ингредиентов"

/>

</LinearLayout>

<android.support.v7.widget.RecyclerView

android:id="@+id/recyclerViewIngredients"

android:orientation="vertical"

android:padding="5dp"

android:layout\_margin="5dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</LinearLayout>

## activity\_savedRecipes.xml

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:fitsSystemWindows="true">

<LinearLayout

android:id="@+id/rectangle\_2"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:padding="10dp"

android:orientation="horizontal"

android:background="@drawable/rectangle\_2">

<RelativeLayout

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_gravity="left|center\_vertical"

android:id="@+id/relativeLayoutBack">

<Button

android:layout\_width="28dp"

android:layout\_height="28dp"

android:gravity="left|center\_vertical"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

android:background="@drawable/round\_arrow\_back\_ios\_24"

android:id="@+id/buttonBack"/>

</RelativeLayout>

<android.support.v7.widget.SearchView

android:textSize="14dp"

android:layout\_gravity="left|center\_vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:textColor="#ffc4c4c4"

android:layout\_marginRight="10dp"

android:layout\_marginLeft="10dp"

android:background="@drawable/rectangle\_3"

android:id="@+id/searchViewFind"/>

</LinearLayout>

<android.support.v7.widget.RecyclerView

android:id="@+id/recyclerViewRecipes"

android:orientation="vertical"

android:padding="5dp"

android:layout\_margin="5dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</LinearLayout>

## activity\_sections.xml

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:fitsSystemWindows="true">

<LinearLayout

android:id="@+id/rectangle\_2"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:padding="10dp"

android:orientation="horizontal"

android:background="@drawable/rectangle\_2">

<RelativeLayout

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_gravity="left|center\_vertical"

android:id="@+id/relativeLayoutBack">

<Button

android:layout\_width="28dp"

android:layout\_height="28dp"

android:gravity="left|center\_vertical"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

android:background="@drawable/round\_arrow\_back\_ios\_24"

android:id="@+id/buttonBack"/>

</RelativeLayout>

<TextView

android:id="@+id/textView\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="@android:color/white"

android:textSize="17dp"

android:paddingLeft="20dp"

android:fontFamily="@font/actorregular"

android:layout\_gravity="center\_horizontal|center\_vertical"

/>

</LinearLayout>

<android.support.v4.widget.SwipeRefreshLayout

android:id="@+id/swipeRefreshLayout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<android.support.v7.widget.RecyclerView

android:id="@+id/recyclerViewRecipesSections"

android:orientation="vertical"

android:padding="5dp"

android:layout\_margin="5dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</android.support.v4.widget.SwipeRefreshLayout>

</LinearLayout>

## fragment\_recipe\_description.xml

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

tools:context=".RecipeDescriptionFragment"

android:padding="10dp">

<android.support.v7.widget.CardView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:cardCornerRadius="4dp">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_gravity="center\_horizontal"

android:orientation="vertical"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:padding="5dp"

android:layout\_marginBottom="20dp">

<ImageView

android:layout\_height="250dp"

android:layout\_width="match\_parent"

android:scaleType="centerCrop"

android:fontFamily="@font/actorregular"

android:id="@+id/imageMainRecipe"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="10dp"

android:paddingRight="10dp"

android:gravity="left"

android:textSize="16dp"

android:fontFamily="@font/actorregular"

android:id="@+id/CPFCRecipe"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="10dp"

android:paddingRight="10dp"

android:gravity="left"

android:fontFamily="@font/actorregular"

android:textSize="16dp"

android:id="@+id/authorNameRecipe"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="10dp"

android:fontFamily="@font/actorregular"

android:paddingRight="10dp"

android:gravity="left"

android:textSize="16dp"

android:id="@+id/additionalInfoRecipe"/>

<TextView

android:textStyle="bold"

android:padding="10dp"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:singleLine="false"

android:gravity="left"

android:textSize="18dp"

android:textColor="#000000"

android:fontFamily="@font/abhayalibrebold"

android:id="@+id/titleMainDescription"/>

<android.webkit.WebView

android:id="@+id/videoRecipe"

android:layout\_width="match\_parent"

android:layout\_height="240dp"

android:layout\_gravity="center\_horizontal"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:gravity="left"

android:textSize="12dp"

android:fontFamily="@font/actorregular"

android:textColor="@color/colorMain"

android:id="@+id/urlRecipe"/>

</LinearLayout>

</ScrollView>

</android.support.v7.widget.CardView>

</RelativeLayout>

## fragment\_recipe\_ingredients.xml

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

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".RecipeIngredientsFragment"

android:orientation="vertical">

<ListView

android:id="@+id/listIngredients"

android:orientation="vertical"

android:padding="5dp"

android:layout\_margin="5dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</ListView>

</LinearLayout>

## fragment\_recipe\_steps.xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".RecipeStepsFragment"

android:orientation="vertical">

<ListView

android:id="@+id/listSteps"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</ListView>

</LinearLayout>

## list\_ingredients.xml

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

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:cardCornerRadius="4dp"

android:layout\_margin="5dp"

android:elevation="10dp"

android:background="@color/colorMain">

<CheckBox

android:id="@+id/ingredientCheck"

android:layout\_gravity="right|center\_vertical"

android:layout\_width="35dp"

android:layout\_height="35dp"

android:layout\_marginRight="5dp"/>

<LinearLayout

android:layout\_width="360dp"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="10dp">

<TextView

android:id="@+id/ingredientName"

android:singleLine="false"

android:layout\_weight="1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:fontFamily="@font/actorregular"

android:textSize="20dp"/>

<TextView

android:id="@+id/ingredientUnit"

android:singleLine="false"

android:layout\_weight="1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:fontFamily="@font/actorregular"

android:textSize="14dp"

/>

</LinearLayout>

</android.support.v7.widget.CardView>

## list\_ingredients\_saved.xml

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:cardCornerRadius="4dp"

android:layout\_margin="5dp"

android:elevation="10dp">

<RelativeLayout

android:padding="10dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<Button

android:id="@+id/cart\_button\_saved\_ingredient"

android:layout\_width="32dp"

android:layout\_height="32dp"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:background="@drawable/round\_clear\_24"/>

<LinearLayout

android:layout\_width="302dp"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<TextView

android:singleLine="false"

android:id="@+id/title\_saved\_ingredient"

android:layout\_width="302.5dp"

android:fontFamily="@font/actorregular"

android:textColor="#000000"

android:layout\_height="wrap\_content"/>

<TextView

android:singleLine="false"

android:id="@+id/unit\_saved\_ingredient"

android:fontFamily="@font/actorregular"

android:textColor="#000000"

android:layout\_width="302.5dp"

android:layout\_height="wrap\_content"/>

<TextView

android:singleLine="false"

android:id="@+id/recipe\_saved\_ingredient"

android:layout\_width="302.5dp"

android:layout\_height="wrap\_content"/>

</LinearLayout>

</RelativeLayout>

</android.support.v7.widget.CardView>

## list\_item.xml

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

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:cardCornerRadius="8dp"

android:layout\_margin="5dp"

android:padding="10dp"

android:elevation="10dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<ImageView

android:layout\_width="140dp"

android:layout\_gravity="left|center\_vertical"

android:layout\_height="100dp"

android:layout\_alignParentLeft="true"

tools:src="@tools:sample/backgrounds/scenic"

android:id="@+id/imageTitle"

android:scaleType="centerCrop"

/>

<LinearLayout

android:orientation="vertical"

android:layout\_width="match\_parent"

android:padding="8dp"

android:layout\_height="wrap\_content">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:gravity="left"

android:fontFamily="@font/actorregular"

android:ems="10"

android:singleLine="false"

android:textSize="18dp"

android:textStyle="normal"

android:id="@+id/title"

android:textColor="@android:color/black"

/>

<TextView

android:id="@+id/textLink"

android:fontFamily="@font/actorregular"

android:singleLine="false"

android:layout\_gravity="left|bottom"

android:gravity="left|bottom"

android:textColor="@color/colorMain"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="11dp"

/>

</LinearLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

## list\_plates.xml

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="210dp"

app:cardCornerRadius="4dp"

android:layout\_margin="5dp"

android:elevation="10dp">

<RelativeLayout

android:id="@+id/relativeLayout\_background"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="3dp">

<RelativeLayout

android:padding="5dp"

android:layout\_width="wrap\_content"

android:layout\_alignParentLeft="true"

android:background="@drawable/rectangle\_plate"

android:scaleType="centerCrop"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/textView\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="14dp"

android:textColor="#eb9502"

android:textStyle="bold"

android:fontFamily="@font/actorregular"/>

</RelativeLayout>

</RelativeLayout>

</android.support.v7.widget.CardView>

## list\_steps.xml

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp">

<android.support.v7.widget.CardView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:cardCornerRadius="8dp"

android:layout\_margin="5dp"

android:elevation="10dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:id="@+id/stepImage"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:ems="10"

android:textSize="18dp"

android:gravity="left"

android:fontFamily="@font/actorregular"

android:id="@+id/stepDescription"

android:textColor="@android:color/black"

/>

</LinearLayout>

</android.support.v7.widget.CardView>

</RelativeLayout>

## main\_bar.xml

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

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/rectangle\_2"

android:layout\_width="match\_parent"

android:layout\_height="80dp"

android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true"

android:padding="10dp"

android:orientation="horizontal"

android:background="@drawable/rectangle\_2">

<Button

android:layout\_width="32dp"

android:layout\_height="32dp"

android:layout\_gravity="center\_vertical"

android:background="@drawable/round\_menu\_24"

android:id="@+id/menu\_button"

android:layout\_marginRight="15.6dp"

android:layout\_marginLeft="3.8dp"

/>

<RelativeLayout

android:outlineSpotShadowColor="#000000"

android:layout\_height="match\_parent"

android:layout\_width="match\_parent"

android:layout\_gravity="center\_vertical"

android:layout\_marginRight="10dp"

android:background="@drawable/rectangle\_3"

android:elevation="3dp">

<EditText

android:imeOptions="actionSearch"

android:layout\_gravity="center\_vertical|left"

android:inputType="text"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:maxLength="100"

android:textSize="17dp"

android:layout\_marginRight="10dp"

android:padding="3dp"

android:layout\_marginLeft="10dp"

android:fontFamily="@font/actorregular"

android:hint="Поиск рецепта..."

android:background="@drawable/rectangle\_3"

android:textColor="#eb9502"

android:layout\_centerInParent="true"

android:id="@+id/TextFind"/>

<Button

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_marginRight="10dp"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:layout\_gravity="center\_vertical|right"

android:background="@drawable/vector"

android:id="@+id/findButton"/>

</RelativeLayout>

</LinearLayout>

## nav\_header\_main.xml

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

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="100dp"

android:background="#f09927"

android:orientation="horizontal"

android:outlineSpotShadowColor="@android:color/black"

android:elevation="10dp"

android:theme="@style/ThemeOverlay.AppCompat.Dark">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:background="@drawable/pan"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center\_vertical"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="20dp"

android:layout\_gravity="center\_vertical"

android:text="На плите!"

android:textColor="#fff9f0"

android:fontFamily="@font/actorregular"

android:textAppearance="@style/TextAppearance.AppCompat.Body1"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Агрегатор кулинарных рецептов"

android:textColor="#fff9f0"

android:fontFamily="@font/actorregular"

android:textSize="15dp"

/>

</LinearLayout>

</LinearLayout>

## spinner\_text.xml

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

<TextView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:textSize="16dp"

android:paddingTop="2dp"

android:paddingLeft="15dp"

android:paddingBottom="10dp"

android:fontFamily="@font/actorregular"

android:textColor="#FFFFFF"/>

## activity\_main\_drawer.xml

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

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

tools:showIn="navigation\_view">

<group android:checkableBehavior="single">

<item

android:id="@+id/nav\_section"

android:title="Поиск по разделам"/>

<item

android:id="@+id/nav\_cart"

android:title="Корзина продуктов"/>

<item

android:id="@+id/nav\_favorite"

android:title="Избранное"/>

<item

android:id="@+id/nav\_app"

android:title="О приложении"/>

</group>

</menu>

## menu\_main.xml

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

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto">

<item android:id="@+id/action\_search"

android:title="Search"

android:icon="@android:drawable/ic\_menu\_search"

app:showAsAction="always|collapseActionView"

app:actionViewClass="android.support.v7.widget.SearchView" />

</menu>

## colors.xml

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

<resources>

<color name="colorPrimary">#2c3e50</color>

<color name="colorPrimaryDark">#1B3147</color>

<color name="colorAccent">#3498db</color>

<color name="colorMain">#ffa000</color>

</resources>

## strings.xml

<resources>

<string-array name="sort\_array">

<item>Новые рецепты</item>

<item>Популярные рецепты</item>

<item>Случайные рецепты</item>

</string-array>

<string-array name="recipeTitles">

<item>Описание</item>

<item>Ингредиенты</item>

<item>Рецепт</item>

</string-array>

<string-array name="messagesError">

<item>Произошла ошибка!</item>

<item>К сожалению, данный рецепт загрузить пока нельзя, однако эту проблему мы скоро решим. Извините, что так получилось :(</item>

<item>Закрыть рецепт</item>

</string-array>

<string name="app\_name">На плите</string>

<string name="navigation\_drawer\_open">Open navigation drawer</string>

<string name="navigation\_drawer\_close">Close navigation drawer</string>

<string name="action\_settings">Settings</string>

</resources>

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