Министерство образования Республики Беларусь

Учреждение образования «Институт информационных технологий Белорусского государственного университета информатики и

радиоэлектроники»

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Дисциплина: Разработка программного обеспечения для мобильных платформ.

ОТЧЁТ

по лабораторной работе №1

Вариант № 6

Выполнил: студент гр. 981064 Ефименко П. В.

Проверил: Калитеня И. Л.

Минск 2021

**Цель:** изучение и создание прототипов мобильных приложений.

**Задание:** Разработать мобильное приложение согласно прототипа, представленного ниже на рисунке 1. Реализовать следующий функционал:

1. По нажатию на кнопку «FEED» происходит обновление счетчика в поле Satiety, каждые 15 нажатий, реализовать анимацию кота(по своему усмотрению)
2. Предусмотреть согласно Guidelines платформы возможность поделиться результатом в соц сетях
3. Реализовать пункт О Разработчике, где указать номер группы, фамилию и номер лабораторной
4. Вести запись результатов и отобразить в виде таблице (Дата и результат)

Код программы:

namespace Laba\_1.Services

{

public class MemoryRepository : IRepository

{

private readonly IList<object> \_dbList;

public MemoryRepository()

{

\_dbList = new List<object>();

}

public Task AddAsync<T>(T entity) where T : class, IBaseEntity

{

return Task.Run(() =>

{

entity.Id = \_dbList.Count;

\_dbList.Add(entity);

return true;

});

}

public Task AddAllAsync<T>(IEnumerable<T> entities) where T : class, IBaseEntity

{

RemoveAllAsync<T>();

return Task.Run(() =>

{

entities = AddIndexToEntities<T>(entities);

foreach (var entity in entities)

{

\_dbList.Add(entity);

}

return true;

});

}

public async Task<IList<T>> GetAllAsync<T>() where T : class, IBaseEntity

{

return await Task.Run(() => \_dbList.OfType<T>().ToList());

}

public Task RemoveAllAsync<T>() where T : class, IBaseEntity

{

return Task.Run(() =>

{

\_dbList.Clear();

return true;

});

}

private IEnumerable<T> AddIndexToEntities<T>(IEnumerable<T> entities) where T : class, IBaseEntity

{

var list = entities.ToList();

for (int i = 0; i < list.Count; i++)

{

var item = list[i];

item.Id = i;

}

return list;

}

}

}

{

public class LoginViewModel : BaseViewModel

{

public Command LoginCommand { get; }

public LoginViewModel()

{

LoginCommand = new Command(OnLoginClicked);

}

private async void OnLoginClicked(object obj)

{

// Prefixing with `//` switches to a different navigation stack instead of pushing to the active one

await Shell.Current.GoToAsync($"//{nameof(AboutPage)}");

}

}

}

class CatClickerViewModel : BaseViewModel

{

private readonly Image \_catImage;

public CatClickerViewModel(Image catImage)

{

Title = "Задание на 1-4";

FeedClickCommand = new Command(() => OnFeedClickCommand());

ShareResultCommand = new Command(async () => await OnShareResultCommand());

Satiety = new ObservableObject<int> { Property = 0 };

\_catImage = catImage ?? throw new ArgumentNullException();

}

public ObservableObject<int> Satiety { get; }

public ICommand FeedClickCommand { get; }

public ICommand ShareResultCommand { get; }

private async void OnFeedClickCommand()

{

if (IsBusy) return;

IsBusy = true;

try

{

this.Satiety.Property++;

await \_catImage.RotateTo(360, 200, Easing.CubicInOut);

await \_catImage.ScaleTo(2, 100, Easing.CubicInOut);

await \_catImage.ScaleTo(1, 100, Easing.CubicInOut);

\_catImage.Rotation = 0;

}

catch (Exception ex)

{

Debug.WriteLine(ex);

}

finally

{

IsBusy = false;

}

}

private async Task OnShareResultCommand()

{

await Share.RequestAsync(new ShareTextRequest

{

Text = $"Мой результат: {this.Satiety.Property}!",

Title = "Поделиться результатом"

});

}

}

}

public class AboutViewModel : BaseViewModel

{

public AboutViewModel()

{

Title = "О Разработчике";

OpenWebCommand = new Command(async () => await Browser.OpenAsync("https://aka.ms/xamarin-quickstart"));

}

public ICommand OpenWebCommand { get; }

}

}

public class BaseViewModel : INotifyPropertyChanged

{

bool isBusy = false;

public bool IsBusy

{

get { return isBusy; }

set { SetProperty(ref isBusy, value); }

}

string title = string.Empty;

public string Title

{

get { return title; }

set { SetProperty(ref title, value); }

}

protected bool SetProperty<T>(ref T backingStore, T value,

[CallerMemberName] string propertyName = "",

Action onChanged = null)

{

if (EqualityComparer<T>.Default.Equals(backingStore, value))

return false;

backingStore = value;

onChanged?.Invoke();

OnPropertyChanged(propertyName);

return true;

}

#region INotifyPropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

protected void OnPropertyChanged([CallerMemberName] string propertyName = "")

{

var changed = PropertyChanged;

if (changed == null)

return;

changed.Invoke(this, new PropertyChangedEventArgs(propertyName));

}

#endregion

}

namespace Laba\_1.Droid

{

[Activity(Label = "Laba\_1", Icon = "@mipmap/icon", Theme = "@style/MainTheme", MainLauncher = true, ConfigurationChanges = ConfigChanges.ScreenSize | ConfigChanges.Orientation | ConfigChanges.UiMode | ConfigChanges.ScreenLayout | ConfigChanges.SmallestScreenSize )]

public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity

{

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

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

global::Xamarin.Forms.Forms.Init(this, savedInstanceState);

LoadApplication(new App());

}

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);

}

}

}

Результат выполнения программы представлен на рисунке 1,2,3.

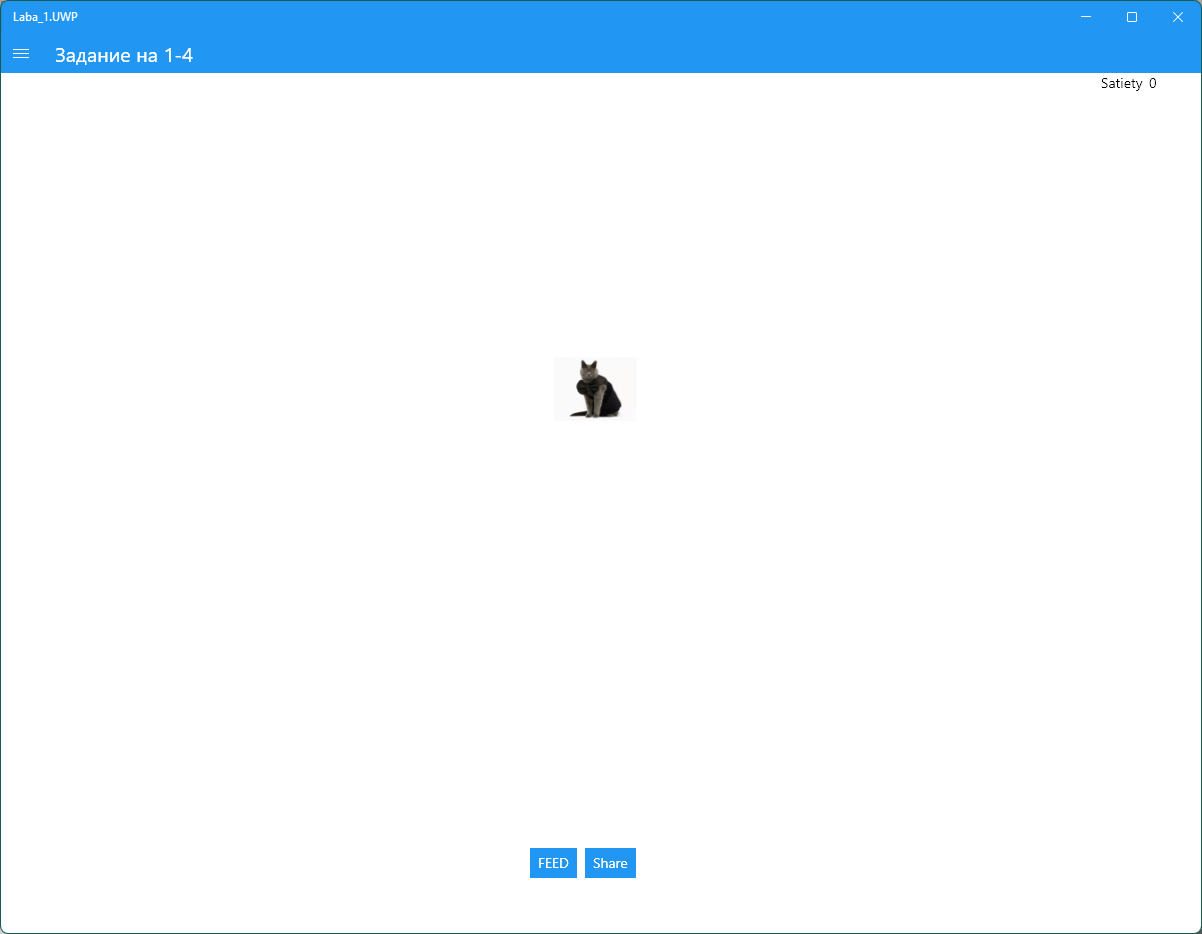


Рисунок 1 – Главное меню

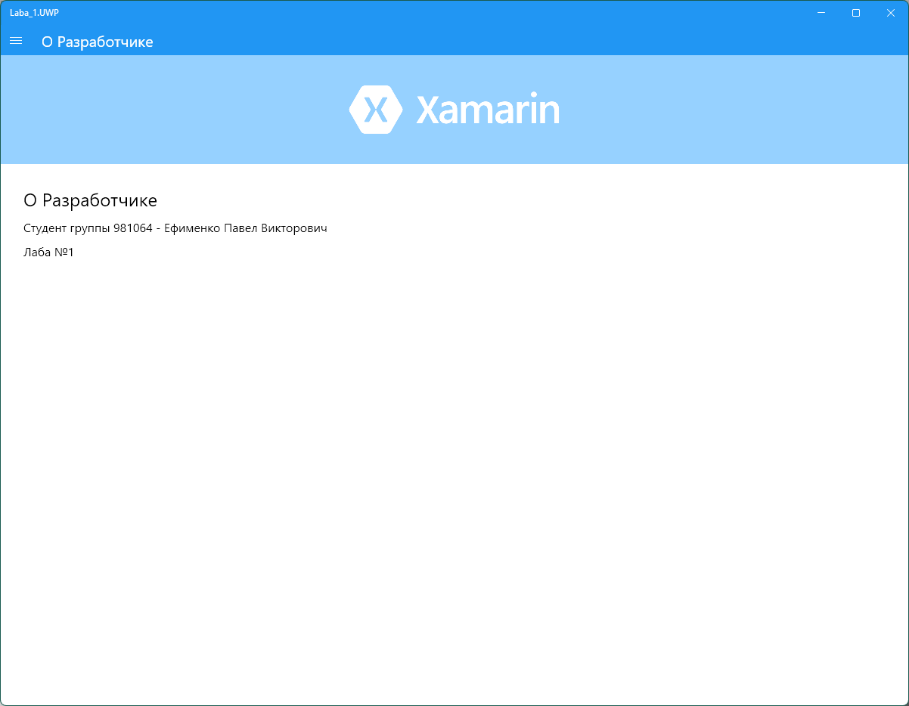


Рисунок 2 – О разработчике

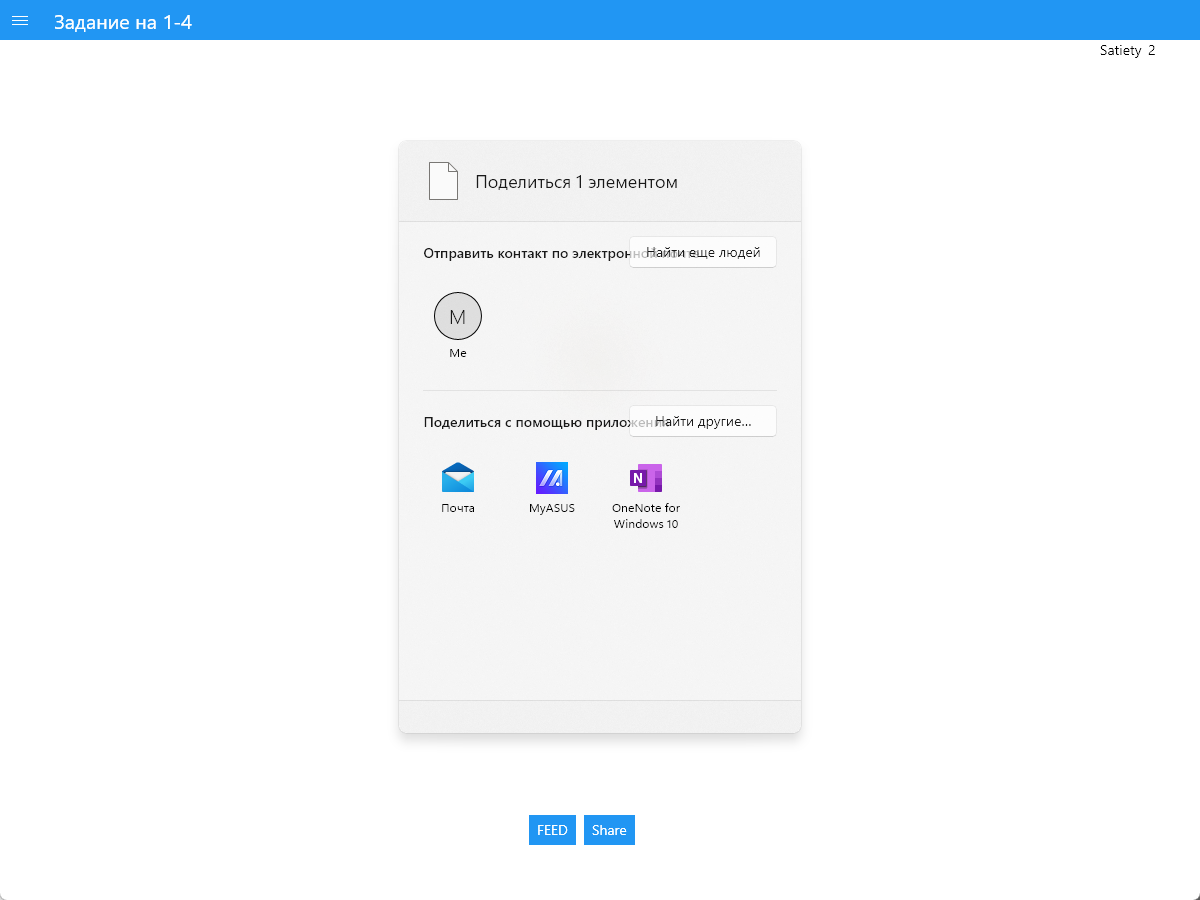


Рисунок 3 – Окно «Поделиться»

**Вывод:** в ходе выполнения лабораторной работы мною было изучено мобильное приложения для операционной системы Android написанное в среде разработки «Xamarin», на языке программирования C#.