Техническое задание на разработку мобильного приложения  
«Справка о предоставляемых услугах»

**1. Общие сведения**

**Наименование модуля:**  
Мобильное приложение «Справка о предоставляемых услугах».

**Основание для разработки:**  
Производственная практика, проведение учебной разработки в рамках образовательной программы.

**Разработчик:**  
Силкин Антон Андреевич  
АНО ПОО «ОТК»

**Дата начала разработки:**  
26.05.2025

**Дата окончания:**

20.06.2025

**2. Назначение разработки**

Разрабатываемое программное обеспечение предназначено для предоставления пользователям справочной информации о государственных услугах в удобной, структурированной форме. Приложение позволяет выбирать категорию услуг, просматривать список доступных услуг в данной категории и получать подробную информацию о каждой услуге.

**3. Требования к программе**

**3.1 Требования к функциональности**

Программа должна предоставлять следующий функционал:

* Отображение списка категорий услуг;
* Отображение списка услуг, относящихся к выбранной категории;
* Отображение подробной информации о выбранной услуге (название, документы, способы подачи, стоимость и др.);
* Навигация между экранами приложения;
* Работа на мобильных устройствах (Android);
* Возможность масштабирования в будущем под iOS и Windows.

**3.2 Требования к интерфейсу**

* Удобный и интуитивно понятный интерфейс;
* Использование цветовой схемы, подходящей для длительного просмотра (тёмная тема);
* Адаптивная верстка под разные размеры экранов;
* Полный вывод названия услуги (без обрезки текста);
* Реализация с использованием XAML и элементов управления .NET MAUI.

**3.3 Требования к надёжности**

* Приложение не должно аварийно завершать работу при переключении между экранами;
* Данные должны отображаться корректно и соответствовать категории.

**3.4 Требования к техническим средствам**

* Устройство под управлением Android 8.0 или выше;
* Минимальный объём оперативной памяти: 2 ГБ;
* Разрешение экрана не ниже 720×1280.

**4. Стадии и этапы разработки**

* Анализ предметной области и формирование требований
* Проектирование архитектуры приложения и интерфейсов
* Разработка прототипа интерфейса
* Реализация бизнес-логики приложения
* Отладка и тестирование
* Оформление итогового отчёта и документации

**5. Источники данных**

Информация о государственных услугах добавляется вручную, при необходимости возможна интеграция с внешними API в будущем.

**6. Порядок контроля и приемки**

Контроль правильности работы осуществляется методом ручного тестирования:

* Проверка работы всех экранов и переходов;
* Проверка отображения информации;
* Проверка на отсутствие критических ошибок;
* Проверка соответствия требованиям раздела 3.

**7. Прочие требования**

* Код должен быть оформлен в соответствии с общепринятыми стандартами C#;
* Использовать систему контроля версий Git (локально или с GitHub);
* Возможность расширения функционала в будущем (например, добавление поиска, фильтров, сохранения в избранное и т. д.).

**ПРИЛОЖЕНИЯ**

ПРИЛОЖЕНИЕ А «СКРИНШОТЫ РАБОТЫ ПРИЛОЖЕНИЯ»

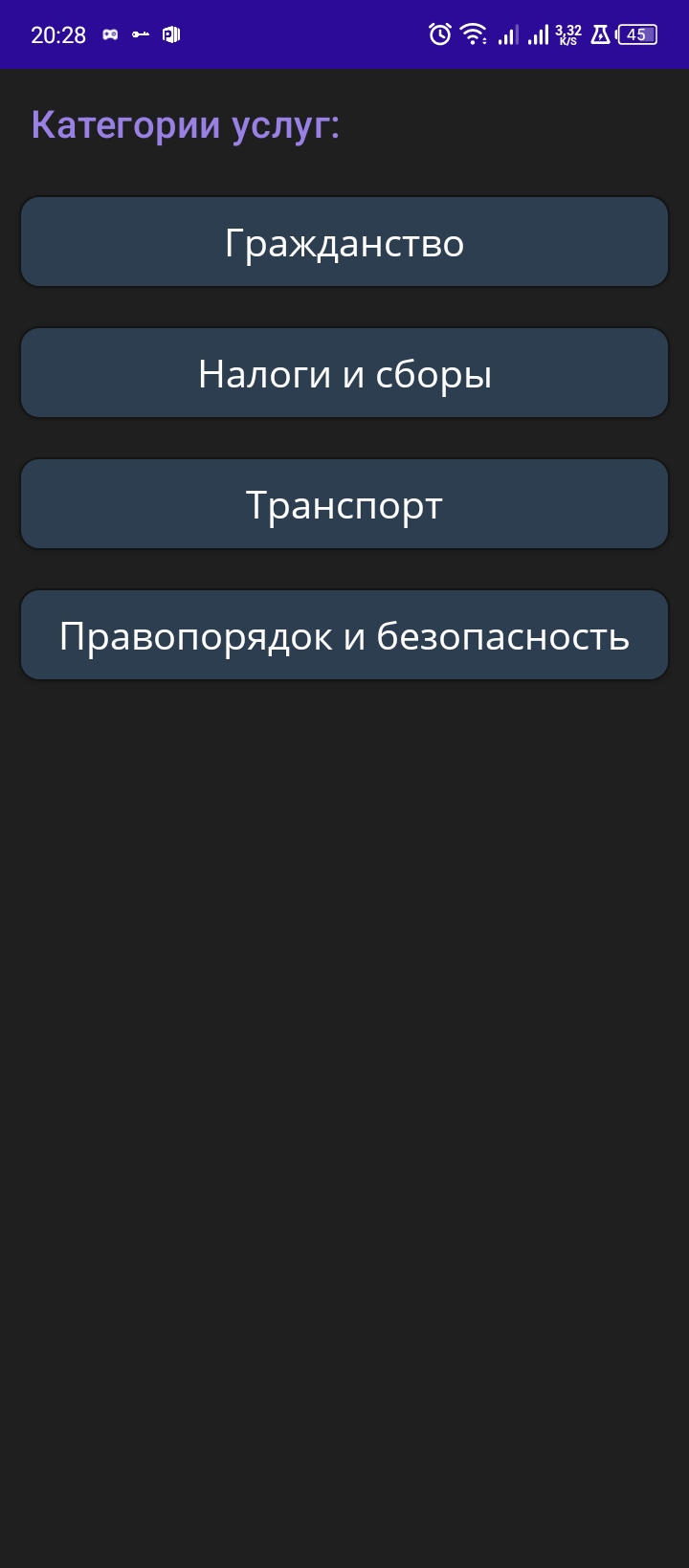


Рисунок 1 «Категории услуг»

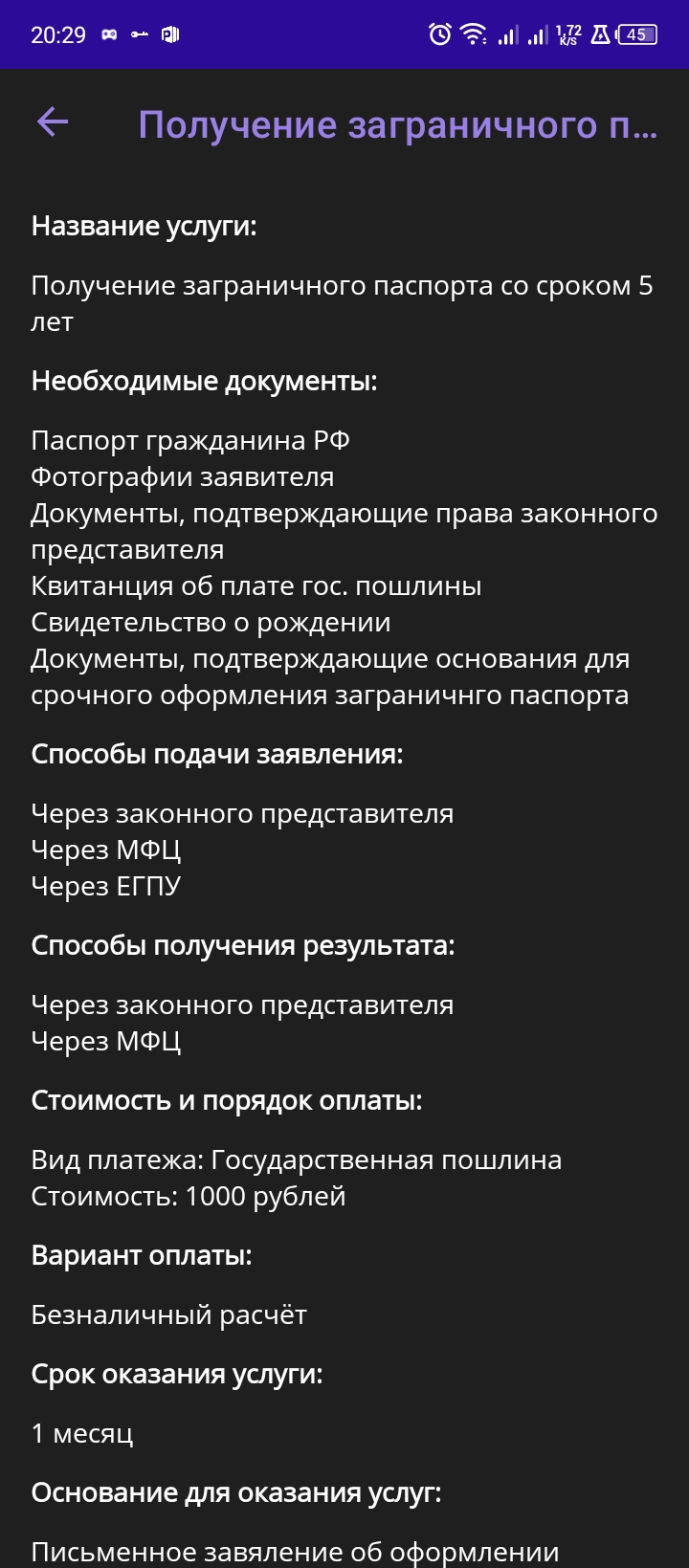


Рисунок 2 «Подробная информация о услуге»

ПРИЛОЖЕНИЕ Б «ЛИСТИНГ ОСНОВНЫХ ФРАГМЕНТОВ КОДА»

# ПРИЛОЖЕНИЕ 1: ФРАГМЕНТ КОДА ГЛАВНОЙ СТРАНИЦЫ

using MauiApp4.Models;

namespace MauiApp4;

public partial class MainPage : ContentPage

{

public List<string> Categories { get; set; } = new List<string>

{

"Гражданство",

"Налоги и сборы",

"Транспорт",

"Правопорядок и безопасность",

};

public MainPage()

{

InitializeComponent();

CategoriesView.ItemsSource = Categories;

}

private async void OnCategoryClicked(object sender, EventArgs e)

{

if (sender is Button button && button.CommandParameter is string category)

{

await Shell.Current.GoToAsync(nameof(ServicesPage), true, new Dictionary<string, object>

{

["Category"] = category

});

}

}

}

# ПРИЛОЖЕНИЕ 2: ФРАГМЕНТ КОДА СТРАНИЦЫ ПОДРОБНОЙ ИНФОРМАЦИИ ОБ УСЛУГЕ

using MauiApp4.Models;

namespace MauiApp4;

public partial class ServicesPage : ContentPage, IQueryAttributable

{

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

public List<Service> Services { get; set; } = new();

private List<Service> AllServices = new()

{

//категория "Гражданство"

new Service {

Category = "Гражданство",

Name = "Получение внутреннего паспорта",

RequiredDocuments = "Две личные фотографии\nДокументы, необходимые для проставления обязательных отметок в паспорте\nКвитанция об уплате государственной пошлины\nЗаявление о выдаче(замене) паспорта\nПаспорт гражданина РФ, подлежащий замене\nСвидетельство о рождении\nСвидетельство о регистрации брака\nСвидетельство о расторжении брака",

ApplicationMethods = "Лично\nЧерез законного представителя\nЧерез МФЦ\nЧерез ЕГПУ",

ResultMethods = "Лично\nЧерез законного представителя\nЧерез МФЦ",

Cost = "Вид платежа: Государственная пошлина\nСтоимость: 300 рублей",

PaymentOptions = "Безналичный расчёт",

ProcessingTime = "10 дней",

LegalBasis = "Подача лично гражданином заявления о выдаче(замене) паспорта на бумажном носителе в подразделение по вопросам миграции территориального органа МВД России, либо через многофункциональный центр с предоставлением всех необходимых документов",

RefusalBasis = "Наличие у гражданина ранее выданного действительного паспорта\nОтсутствие у лица гражданства РФ\nНедостоверные сведения, указанные в заявлении о выдаче(замене) паспорта",

Result = "Выдача оформленного паспорта гражданину"

},

public ServicesPage()

{

InitializeComponent();

}

public void ApplyQueryAttributes(IDictionary<string, object> query)

{

if (query.TryGetValue("Category", out var categoryObj))

{

Category = categoryObj as string ?? "";

Title = Category;

Services = AllServices

.Where(s => s.Category == Category)

.ToList();

ServicesView.ItemsSource = Services;

}

}

private async void OnServiceTapped(object sender, TappedEventArgs e)

{

if (e.Parameter is Service service)

{

await Shell.Current.GoToAsync(nameof(ServiceDetailPage), true, new Dictionary<string, object>

{

["Service"] = service

});

}

}

}

ПРИЛОЖЕНИЕ В «ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ»



Рисунок 2 «Диаграмма вариантов использования»