**Tallinna Tööstushariduskeskus**

**Noorem tarkvaraarendaja eriala**

**RAKENDUS “WATCHWEATHER APP”**

**Arvestus töö**

**Õpilane: Mihhail Bajandin**

**Rühm: TARgv23**

**Juhendaja: IT kutseõpetaja Marina Oleinik**

**Tallinn 2024**

**Содержание**

[**1.Введение**](#_r2he8vfyhm5m) **3**

[1.1.Обоснование выбора темы](#_4r8rp61b4gx4) 3

[1.2.Цель проекта](#_1mxriowpxn2z) 3

[1.3.Актуальность темы](#_svethojoaklp) 3

[1.4.Целевая аудитория](#_n6a8rr3987sx) 3

[1.5.Технические требования](#_n2a5wkttxlov) 4

[**2.Теоретическая часть**](#_guo4aibfg7fl) **4**

[2.1.Технические требования (язык, фреймворк, SDK)](#_soblokh0p7kp) 4

[2.2.Описание функциональности и дизайна](#_vftsokmekwni) 4

[2.3.Функции приложения](#_56os8uhtsdz9) 4

[2.4.Нефункциональные требования](#_oh1w3d7plu2q) 4

[2.5.Схема проекта](#_4jtfa8oxepoq) 5

[**3.Практическая часть**](#_1yt5lijx7m3e) **5**

[3.1.Разработка дизайна](#_lgyb07y1uvbv) 5

[3.2.Создание базы данных](#_n6e2246sp4el) 5

[3.3.Пример кода](#_ou5wvthea2rk) 5

[3.4.Руководство по использованию](#_cm0cdn4wpb5y) 5

[Заключение](#_hhevrn87ybvz) 6

# **1.Введение**

Проект описывает создание многостраничного мобильного приложения для прогноза погоды. Приложение реализует основные принципы объектно-ориентированного программирования, использует базу данных для хранения избранных городов, а также предоставляет функциональность для получения данных о погоде через API OpenWeatherMap.

Проект направлен на изучение практического использования MAUI Framework, а также демонстрацию возможностей интеграции API и работы с локальными базами данных.

## **1.1.Обоснование выбора темы**

Тема выбрана с целью практического применения полученных знаний в программировании, а также создания полезного мобильного приложения для прогнозирования погоды, которое может быть использовано широким кругом пользователей.

## **1.2.Цель проекта**

Цель проекта - создание многостраничного мобильного приложения для прогноза погоды, которое предоставляет пользователю удобный и современный интерфейс.

## **1.3.Актуальность темы**

Прогноз погоды - это актуальная информация, которая нужна многим людям ежедневно. Приложение предоставляет возможность быстро и удобно узнавать погоду в интересующем городе.

## **1.4.Целевая аудитория**

Целевая аудитория приложения - это пользователи мобильных устройств, которые хотят легко и быстро получать прогноз погоды. Иметь быстрый доступ к своему любимому городу.

## **1.5.Технические требования**

Минимальные технические требования для запуска приложения: мобильное устройство с поддержкой .NET MAUI, подключение к Интернету для получения данных через API.

# **2.Теоретическая часть**

## **2.1.Технические требования (язык, фреймворк, SDK)**

* Язык программирования: C#
* Framework: .NET MAUI
* SDK: .NET 7.0
* База данных: SQLite
* API: OpenWeatherMap

## **2.2.Описание функциональности и дизайна**

Приложение позволяет:

- Вводить название города для получения прогноза погоды.

- Сохранять города в список избранных.

- Переходить между страницами приложения для отображения погоды или управления избранными городами.

## **2.3.Функции приложения**

- Получение погоды через API OpenWeatherMap.

- Хранение избранных городов в локальной базе данных.

- Переход между страницами для отображения информации.

## **2.4.Нефункциональные требования**

Приложение должно быть:

- Удобным и интуитивным в использовании.

- Оптимизированным для работы на мобильных устройствах.

## **2.5.Схема проекта**

Основные страницы приложения:

1. Главная страница с поиском города.
2. Страница избранных городов.
3. Страница с отображением погоды.

# **3.Практическая часть**

## **3.1.Разработка дизайна**

Дизайн приложения выполнен с использованием стандартных компонентов MAUI.

Кнопки были выполнены синем цветом.

## **3.2.Создание базы данных**

Для хранения избранных городов используется база данных SQLite. Она интегрирована в приложение с помощью библиотеки sqlite-net-pcl.

## **3.3.Пример кода**

Пример работы с базой данных:

public class DatabaseService {

private readonly SQLiteConnection \_db;

public DatabaseService() {

var dbPath = Path.Combine(FileSystem.AppDataDirectory, "weather.db");

\_db = new SQLiteConnection(dbPath);

\_db.CreateTable<FavoriteCity>();

}

public IEnumerable<FavoriteCity> GetCities() => \_db.Table<FavoriteCity>().ToList();

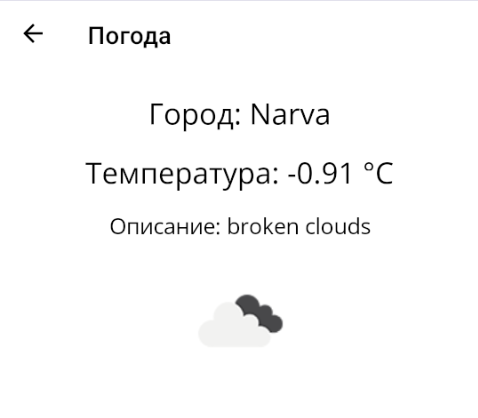
}

## **3.4.Руководство по использованию**

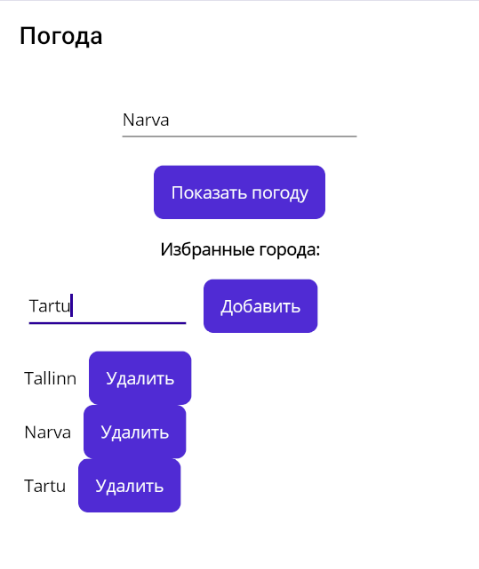
1. Запустите приложение.



1. Введите название города и нажмите "Показать погоду".



1. Сохраните город в избранное, чтобы быстро получать информацию о нём в будущем.



## **Заключение**

Что удалось создать:

- Функциональное приложение для прогноза погоды.

- Поддержка избранных городов и многопоточности.

Что не получилось:

- Реализовать кэширование данных.

Дальнейшая разработка:

- Добавление функции оффлайн-доступа.