**Техническое задание на разработку мобильного приложения "WeatherMood"**

**1. Краткое описание проекта**

**Название:** WeatherMood

**Описание:** Мобильное приложение для Android, которое предоставляет пользователю актуальную информацию о погоде, а также персонализированные рекомендации по одежде, улучшению настроения и экологичным привычкам. Приложение не требует регистрации и работает полностью в офлайн-режиме (за исключением загрузки данных о погоде).

**Цель:** создать простое, но функциональное приложение, которое поможет пользователям быстро получать полезные советы на основе текущих погодных условий без необходимости создания аккаунта.

**Аналоги:**

* Погодные приложения (Yandex.Pogoda, AccuWeather) — не предлагают персонализированных советов.
* Mood-трекеры (Daylio, Moodnotes) — требуют регистрации и не учитывают погоду.
* Эко-приложения (JouleBug) — не интегрированы с погодными данными.

**Дата сдачи:** **17.11.2025** (1-й день зачётной недели для группы 413ИС).

**2. Основные требования**

**2.1. Функциональные требования**

1. **Отображение текущей погоды:**
   * Показ температуры, погодных условий (солнечно, дождь, снег и т.д.), иконки погоды.
   * Возможность выбора города вручную или автоматическое определение по геолокации.
2. **Панели советов:**
   * **"Что одеть?"** — рекомендации по одежде на основе текущей погоды (температура, осадки, ветер).
   * **"Твоё настроение"** — советы по улучшению настроения в зависимости от погоды (например, "пасмурно → послушай весёлую музыку").
   * **"Эко-совет"** — рекомендации по сокращению углеродного следа (например, "при хорошей погоде используй велосипед").
3. **Оценка настроения:**
   * Возможность оценить своё настроение по шкале от 1 до 5 (без сохранения истории).
4. **Геолокация:**
   * Автоматическое определение текущего города пользователя (с запросом разрешения).
   * Возможность вручную выбрать другой город.
5. **Офлайн-режим:**
   * Кеширование последних данных о погоде для отображения без интернета.

**2.1. Нефункциональные требования**

* **Платформа:** Android 8.0+ (Oreo).
* **Производительность:**
  + Быстрая загрузка данных о погоде (не более 2 секунд при наличии интернета).
  + Плавная анимация и отзывчивость интерфейса.
* **Безопасность:**
  + Запрос разрешения на геолокацию с пояснением, для чего это нужно.
* **Дизайн:**
  + Интуитивно понятный интерфейс с минималистичным дизайном.
  + Использование иконок и цветовых схем для улучшения восприятия.
* **Локализация:**
  + Поддержка русского языка (основной), английский опционально.

**3. Технологический стек**

* **Язык программирования:** Kotlin.
* **UI:** Jetpack Compose (для современного и гибкого интерфейса).
* **Сетевые запросы:** Retrofit (для работы с OpenWeatherMap API).
* **Локальное хранилище:** Room (для кеширования данных о погоде).
* **Геолокация:** Fused Location Provider (для определения местоположения).
* **API погоды:** OpenWeatherMap (для получения актуальных данных о погоде).

**4. Рекомендуемые инструменты разработки**

* **IDE: Android Studio (Kotlin + Jetpack Compose).**
* **Дизайн: Figma (прототипы интерфейса).**
* **Версии/CI: GitHub + GitHub Actions.**
* **Тестирование API: Postman.**
* **Отладка: Android Profiler, LeakCanary.**

**5. Архитектура и данные**

**5.1. Архитектурная схема (вкратце)**

Приложение построено по архитектуре **MVVM (Model-View-ViewModel)** с использованием локального хранилища и внешнего API:

* **Мобильное приложение (Kotlin + Jetpack Compose)** взаимодействует с **OpenWeatherMap API** для получения данных о погоде.
* **Локальная база данных (Room)** используется для кеширования данных о погоде и избранных городах, обеспечивая работу в офлайн-режиме.
* **Геолокация (Fused Location Provider)** позволяет автоматически определять текущий город пользователя.
* **Логика формирования советов** реализована на стороне клиента без необходимости в серверной части.

**5.2. Структура локальной базы данных**

Локальное хранилище включает две основные сущности:

* **Кеш погоды**: хранит последние данные о погоде для каждого города (температура, условия, иконка, скорость ветра, влажность).
* **Избранные города**: список городов, добавленных пользователем, с указанием координат и статуса "город по умолчанию".

**6. API / Endpoints**

Для получения данных о погоде используется **OpenWeatherMap API** (One Call API 3.0):

* **Эндпоинт для текущей погоды**: GET https://api.openweathermap.org/data/3.0/onecall
  + Параметры запроса:
    - lat, lon: координаты города.
    - units=metric: температура в градусах Цельсия.
    - lang=ru: ответ на русском языке.
  + Ответ содержит:
    - Температуру.
    - Погодные условия (солнечно, дождь, снег и т.д.).
    - Иконку погоды.
    - Скорость ветра и влажность.

**7. UI/UX: Основные экраны и требования**

**7.1. Основные экраны**

1. **Экран загрузки (Splash Screen)**
   * Отображается при запуске приложения.
   * Логотип и краткая анимация загрузки.
2. **Главный экран**
   * **Верхний блок**: текущая погода (город, температура, иконка, краткое описание).
   * **Три панели советов**:
     + **"Что одеть?"**: рекомендации по одежде на основе погоды.
     + **"Твоё настроение"**: советы по улучшению настроения и возможность его оценки по шкале от 1 до 5.
     + **"Эко-совет"**: рекомендации по сокращению углеродного следа.
   * **Кнопка "Обновить"**: ручное обновление данных о погоде.
   * **Кнопка "Сменить город"**: переход на экран выбора города.
3. **Экран выбора города**
   * Поисковая строка для ручного ввода города.
   * Кнопка **"Определить моё местоположение"** для автоматического определения города по геолокации.
   * Список избранных городов (если добавлены).
4. **Экран настроек (опционально)**
   * Переключатель единиц измерения температуры (°C/°F).
   * Настройки уведомлений (если будут реализованы).

**7.2. Требования к дизайну**

* **Цветовая схема**:
  + Основной цвет: синий (ассоциируется с погодой).
  + Фон: светлый для дневной темы.
* **Шрифты**:
  + Основной текст: Roboto (читабельный).
  + Советы: Comic Sans (дружелюбный тон).
* **Иконки**:
  + Погода: иконки из OpenWeatherMap.
  + Одежда и настроение: кастомные иконки в стиле Material Design.
* **Анимации**:
  + Плавные переходы между экранами.
  + Анимация загрузки погоды (спиннер или прогресс-бар).

**8. Тестирование**

**8.1. Виды тестирования**

* **Unit-тесты**:
  + Проверка логики формирования советов (например, корректность рекомендаций по одежде при разных температурах).
  + Тестирование парсинга ответа от OpenWeatherMap API.
* **UI-тесты (Espresso)**:
  + Проверка отображения погоды и советов.
  + Тестирование навигации между экранами.
* **Интеграционное тестирование**:
  + Сценарий: "Загрузка погоды → отображение советов → смена города".
* **Ручное тестирование**:
  + Проверка на реальных устройствах (Android 8.0+).
  + Тестирование геолокации и офлайн-режима.

**8.2. Критические сценарии**

* Отсутствие интернета: отображение кешированных данных с пометкой "Данные могут быть устаревшими".
* Ошибка геолокации: предложение ввести город вручную.
* Некорректный ответ от API: сообщение об ошибке с кнопкой "Повторить".

**9. Релиз**

**9.1. Подготовка к релизу**

* **Репозиторий**:
  + Хранилище: GitHub/GitLab.
  + Ветки: main (стабильная версия), develop (разработка), feature/\* (фичи).
  + Pull Request с обязательным код-ревью.
* **CI/CD (GitHub Actions)**:
  + Автоматическая сборка APK при пуше в develop.
  + Запуск unit-тестов и проверок кода.
* **Сборка**:
  + Debug-версия для тестирования.
  + Release-версия (оптимизированная, без логов).

**9.2. Артефакты для сдачи**

* APK-файл (отлаженная версия).
* Исходный код (с документацией в README).
* Инструкция по установке (как запустить проект локально).
* Демонстрационное видео (3–5 минут: основной функционал).
* Отчёт (описание, скриншоты, сложности и решения).

**10. Безопасность и приватность**

**10.1. Меры безопасности**

* **Геолокация**:
  + Запрос разрешения с пояснением ("Нужен доступ к местоположению для определения погоды в вашем городе").
  + Данные о местоположении не сохраняются и не передаются третьим лицам.
* **Сетевые запросы**:
  + Все запросы к OpenWeatherMap по HTTPS.
  + API-ключ хранится в local.properties (не в репозитории).
* **Локальные данные**:
  + Кеш погоды и избранные города хранятся только на устройстве (Room DB).
  + Нет сбора личной информации пользователя.

**10.2. Конфиденциальность**

* Приложение не требует регистрации и не собирает данные о пользователе.
* В AndroidManifest.xml объявлены все необходимые разрешения:
  + Доступ к интернету.
  + Доступ к геолокации (грубое и точное местоположение).

**11. Роадмап (план разработки)**

**Сроки:** 16.09.2025 – 17.11.2025 (9 недель).

1. **Неделя 1 (16–21.09.2025):**
   * Сбор требований, создание макетов в Figma.
   * Настройка проекта в Android Studio, подключение необходимых библиотек.
2. **Неделя 2–3 (22.09–05.10.2025):**
   * Реализация главного экрана с отображением погоды.
   * Интеграция OpenWeatherMap API для получения данных о погоде.
3. **Неделя 4 (06–12.10.2025):**
   * Реализация панелей советов ("Что одеть?", "Твоё настроение", "Эко-совет").
   * Логика формирования рекомендаций на основе погоды.
4. **Неделя 5 (13–19.10.2025):**
   * Добавление функции оценки настроения.
   * Реализация геолокации и выбора города.
5. **Неделя 6 (20–26.10.2025):**
   * Кеширование данных о погоде для офлайн-режима.
   * Тестирование и исправление багов.
6. **Неделя 7–8 (27.10–09.11.2025):**
   * Оптимизация интерфейса и производительности.
   * Подготовка к финальному тестированию.
7. **Неделя 9 (10–17.11.2025):**
   * Финальное тестирование, сборка APK.
   * Подготовка документации и презентации проекта.

**12. Уникальные особенности проекта**

* **Отсутствие регистрации:** Пользователь может сразу начать пользоваться приложением.
* **Три панели советов:** Удобное отображение рекомендаций на одном экране.
* **Персонализированные рекомендации:** Советы формируются на основе текущей погоды.
* **Простота использования:** Минималистичный интерфейс без лишних функций.