Техническое предложение

по разработке сервиса-агрегатора текущей погоды "Погода"

1.Введение.

Цель проекта — создать максимально простой и удобный веб-сервис и мобильное приложение для получения усредненной информации о текущей погоде по выбранному городу. Сервис будет опрашивать несколько сторонних погодных API, объединять их данные и предоставлять пользователю единый, понятный интерфейс.

2.Основные функциональные требования.

* Взаимодействие с пользователем: запрос названия города или автоматическое определение местоположения.
* Агрегация данных: опрос нескольких погодных сервисов (Яндекс.Погода, AccuWeather, ГИС-Метео, OpenWeatherMap).
* Усреднение данных: расчет средней температуры, влажности, скорости ветра и других параметров.
* Аутентификация: хранение настроек через cookie; автоматическая установка cookie при первом посещении или при отсутствии cookie.
* Настройки пользователя: возможность изменить список источников погоды и местоположение в любой момент.
* Доступность: как веб-сайт (браузер), так и мобильное приложение для Android.

3. Архитектура системы

3.1 Клиентская часть (Frontend)

Веб-интерфейс — HTML/CSS/JavaScript (использование фреймворков не обязательно, можно на чистом JS).

Мобильное приложение — простое приложение на Android (использование WebView или нативных технологий).

3.2 Серверная часть (Backend)

API для обработки запросов пользователя, получения данных от сторонних сервисов, их агрегации и возврата результата.

Реализация на Node.js или Python (например, Flask/Django).

3.3 Хранение настроек пользователя

Использование cookies для хранения выбранных источников погоды и местоположения.

4.Техническое решение по интеграции с погодными API

4.1 Источники данных (пример)

* Яндекс.Погода — API закрыт или ограничен; возможна интеграция через парсинг сайта или использование альтернативных решений.
* AccuWeather — требует API ключа; регистрация возможна бесплатно с лимитами.
* ГИС-Метео — предоставляет API или данные через XML/JSON интерфейсы.
* OpenWeatherMap — бесплатный API с лимитами; легко интегрируется.

4.2 Получение данных

Для каждого сервиса реализовать отдельный модуль-запрос с учетом их API спецификаций, обработку ошибок и таймаутов.

4.3 Усреднение данных

На сервере — после получения данных от всех источников — вычислять средние значения по каждому параметру.

5. Механизм аутентификации и работы с cookie

5.1 Изначальная установка cookie при первом посещении или отсутствии cookie:

Запросить у пользователя город или определить его автоматически по IP (используя геолокацию).

Установить cookie с настройками (город, список источников).

5.2 В дальнейшем — при повторных обращениях — извлекать настройки из cookie и показывать усредненную погоду без повторного ввода.

5.3 Возможность редактировать настройки в интерфейсе в любой момент.

* Безопасность и защита данных;
* Использовать HTTPS для всех соединений;
* Обеспечить безопасность cookie (HttpOnly, Secure);
* Разработка мобильного приложения для Android:

Можно реализовать как WebView внутри нативного приложения или создать нативное приложение на Kotlin/Java с использованием REST API сервера;

* Этапы реализации:

|  |  |  |
| --- | --- | --- |
| Этап | Описание | Сроки |
| Анализ требований | Детализация функционала | 1 неделя |
| Проектирование архитектуры | Структура системы, схемы взаимодействия | 1 неделя |
| Разработка backend | API, интеграция с погодными сервисами | 3 недели |
| Разработка фронтенда | Веб-интерфейс | 2 недели |
| Разработка мобильного приложения | Android-приложение | 3 недели |
| Тестирование и отладка | Проверка работоспособности | 2 недели |
| Итоговая сдача | Передача заказчику | По завершении |

Используемые технологии

Backend: Node.js (Express) или Python (Flask/Django)

Frontend: HTML/CSS/JavaScript без сложных фреймворков для простоты использования

Мобильное приложение: Android Studio (Kotlin/Java) либо WebView-based приложение

Итоги: Данный проект обеспечит пользователю быстрый доступ к актуальной погоде в удобной форме, минимизируя необходимость повторного ввода данных за счет использования cookies и автоматического определения местоположения.