**Техническое задание на проект Mountain Pass Application**

**1. Введение**

**1.1. Описание проекта**

Проект **Mountain Pass Application** представляет собой мобильное приложение для **Android и iOS**, предназначенное для сбора и отправки данных о горных перевалах в Федерацию спортивного туризма России (ФСТР).

**1.2. Проблема**

В настоящее время процесс добавления перевалов в базу данных ФСТР осуществляется вручную через отправку PDF-отчетов по электронной почте. Этот процесс занимает от **3 до 6 месяцев**. Цель проекта — **сократить время обработки информации до 3 дней**, улучшив удобство внесения данных и автоматизировав процесс верификации и хранения.

**1.3. Основные цели проекта**

* Создание мобильного приложения для туристов.
* Разработка **REST API** для взаимодействия мобильного приложения с сервером ФСТР.
* Обеспечение кэширования данных в мобильном приложении для работы **в условиях отсутствия интернета**.
* Автоматизация процесса верификации и обработки данных.
* Разработка удобного пользовательского интерфейса.

**2. Функциональные требования**

**2.1. Возможности пользователя в мобильном приложении**

* Внесение информации о новом перевале.
* Редактирование **неотправленных** данных (при отсутствии интернета).
* Автозаполнение контактных данных пользователя (ФИО, телефон, e-mail).
* Отправка данных на сервер ФСТР.
* Получение **уведомления** о статусе отправки (успешно/неуспешно).
* Согласие с политикой обработки персональных данных перед отправкой.

**2.2. Данные, передаваемые на сервер ФСТР**

* **Координаты** перевала (широта, долгота).
* **Высота** перевала.
* **ФИО** пользователя.
* **Телефон** и **email** пользователя.
* **Название** перевала.
* **Фотографии** перевала (до 3 штук).

**2.3. Возможности модератора ФСТР**

* Просмотр поступивших данных.
* Верификация информации.
* Подтверждение/отклонение перевала.
* Внесение перевала в базу данных.

**3. Архитектура системы**

**3.1. Компоненты системы**

* **Мобильное приложение (Android и iOS)** – отправка данных.
* **REST API** – обработка запросов и взаимодействие с базой данных.
* **База данных** – хранение данных о перевалах и пользователях.
* **Модераторская панель** – веб-интерфейс для верификации данных.

**3.2. REST API**

**Основные эндпоинты API:**

**Регистрация и аутентификация**

* POST /api/auth/register/ – регистрация пользователя.
* POST /api/auth/login/ – аутентификация и получение токена.
* POST /api/auth/logout/ – выход из системы.

**Работа с перевалами**

* POST /api/passes/ – добавление нового перевала.
* GET /api/passes/ – получение списка перевалов.
* GET /api/passes/{id}/ – получение информации о конкретном перевале.
* PUT /api/passes/{id}/ – редактирование перевала (если он не отправлен).
* DELETE /api/passes/{id}/ – удаление неотправленного перевала.

**Работа с фотографиями**

* POST /api/passes/{id}/photos/ – добавление фото.
* DELETE /api/passes/{id}/photos/{photo\_id}/ – удаление фото.

**Работа с модерацией**

* GET /api/moderation/ – список объектов на верификацию.
* PUT /api/moderation/{id}/approve/ – подтверждение перевала.
* PUT /api/moderation/{id}/reject/ – отклонение перевала.

**4. Нефункциональные требования**

**4.1. Производительность**

* Обработка запроса на сервере – не более 500 мс.
* Время ответа API – не более 1 секунды.
* Поддержка **минимум 1000 одновременных пользователей**.

**4.2. Безопасность**

* Авторизация через **JWT**.
* Шифрование паролей (bcrypt/scrypt).
* Защита от SQL-инъекций и XSS.

**4.3. Кеширование**

* Локальное хранение данных о перевалах до их отправки.
* Кеширование списка перевалов для быстрого доступа.

**5. Этапы разработки**

1. **Разработка REST API** (backend).
2. **Разработка мобильного приложения** (Android и iOS).
3. **Тестирование API и мобильного приложения**.
4. **Развертывание и оптимизация**.

**6. Требования к среде разработки**

* **Backend:** Python, Django REST Framework.
* **Frontend:** React (для панели модерации).
* **База данных:** PostgreSQL.
* **Мобильное приложение:** Flutter или Kotlin/Swift.

**7. Тестирование**

* **Unit-тестирование API** (pytest, Django test framework).
* **Интеграционное тестирование API**.
* **Функциональное тестирование мобильного приложения**.
* **Тестирование UI и UX**.

**8. Развёртывание**

* **Backend:** облачный сервер (AWS, DigitalOcean, или аналог).
* **База данных:** PostgreSQL (отдельный сервер).
* **CI/CD:** автоматизированное тестирование и развёртывание через GitHub Actions.

**9. Итог**

Это техническое задание охватывает весь проект **Mountain Pass Application**, включая разработку мобильного приложения и **REST API**. В дальнейшем основной фокус будет направлен на создание API, необходимого для взаимодействия мобильного приложения с сервером ФСТР.

Top of Form

Bottom of Form