

Потенциальные сложности разработки и пути их решения

1 Интеграция мобильного и серверного компонентов

Сложность: обеспечение корректного взаимодействия между мобильным приложением, разработанным на React Native, и серверной частью, реализованной с использованием Spring Boot, может потребовать согласования форматов данных, синхронизации сессий и стабильного обмена информацией через REST API.

Пути решения:

- использование стандартизированного REST API для обмена данными между клиентом и сервером;
- применение JWT-токенов для аутентификации, что позволяет обеспечить безопасное и stateless-взаимодействие между компонентами.

2 Обеспечение безопасности системы

Сложность: необходимость защиты пользовательских данных и предотвращения угроз является критически важной.

Пути решения:

- использование Spring Security для реализации аутентификации, авторизации и встроенной защиты;
- применение протокола HTTPS для обеспечения безопасности передачи данных между клиентом и сервером.

3 Масштабируемость и устойчивость архитектуры

Сложность: при росте числа пользователей и объёма данных необходимо обеспечить стабильную работу системы без необходимости кардинальных изменений в архитектуре.

Пути решения:

- проектирование архитектуры с учетом возможности горизонтального масштабирования. Использование PostgreSQL в качестве СУБД для хранения данных;
- развертывание серверной части в контейнерах Docker для упрощения масштабирования и управления инфраструктурой.

4 Адаптивность и кроссплатформенность мобильного приложения

Сложность: обеспечение корректного отображения интерфейса на устройствах с различными разрешениями и характеристиками экранов.

Пути решения:

- применение React Native, который поддерживает адаптивный дизайн и позволяет разрабатывать кроссплатформенные мобильные приложения, оптимизированные для различных устройств под Android.

5 управление версиями и координация работы команды

Сложность: поддержание стабильности кода и координация разработки при работе над различными компонентами проекта.

Пути решения:

- использование GitHub для контроля версий и организации совместной разработки;
- применение системы управления задачами в Jira для планирования, распределения и контроля выполнения работ.

6 Интеграция с внешним AI API

Сложность: при интеграции с AI API для генерации описаний проектов может возникнуть сложность с оптимальным формированием системного промта, который отвечает требованиям качества и структурированности генерируемого текста.

Пути решения:

- проведение предварительного тестирования и экспериментов с различными вариантами системных промтов для определения оптимальных настроек;
- использование A/B-тестирования различных вариантов промтов для выбора наиболее эффективного решения, обеспечивающего корректную и качественную генерацию описания проекта.