**4. # ADR - [Выбор языка программирования для клиентской части]**

Участники: Геворгян Арсен, Афанасьев Никита, Сахабутдинов Тимур

Дата: 25.09.2024

Статус: принят

**## Контекст**

Для клиентской части приложения нам нужно выбрать язык программирования, который подойдет под ОС android и далее интегрировать с web-сервисом.

Команда: Сахабутдинов, Геворгян, Афанасьев (3 человека)

**## Рассматриваемые варианты**

1. Java - ЯП с поддерживающий ОС android с большим количеством библиотек и фреймворков, но имеет сложную интеграцию с web-сервисом.

2. C++ - ЯП с высокой производительностью, поддержка низкоуровневого программирования, но имеет маленькое количество библиотек для работы с клиентской частью.

3. JavaScript (TypeScript) - ЯП для веб-разработки и для гибридных мобильных приложений, что ограничивает возможности для android разработки. Также более низкая производительность в сравнении с другими вариантами.

4. Rust - ЯП с сложным синтаксисом и с маленьким количеством библиотек для клиентской части. К плюсам относится высокая производительность, безопасность памяти и поддержка WASM. Более подходит для высокопроизводительных приложений

5. Kotlin - ЯП совместимо с Java, поддерживает мобильную платформу (Android), современный синтаксис. Имеет меньше библиотек с Java, подходит для большинства приложений на телефон(но не такие как игры).

6. Python - простой лаконичный язык разработки с большим количеством библиотек и фреймворков. Подходит для небольших проектов из-за низкой производительности.

7. WASM (WebAssembly) - самый производительный ЯП с компиляцией современных ЯП, но с ограничением библиотек и фреймворков. Больше подходит для игр, или приложений где есть большая нагрузка на производительность.

**## Решение**

Kotlin

**## Обоснование**

Kotlin совместим с Java, что позволяет использовать существующие библиотеки и инструменты. Современный синтаксис, что ускоряет разработку и уменьшает количество ошибок. Официальная поддержка Google для разработки Android-приложений. Хорошая производительность для нашего приложения + есть опыт у разработчиков в работе с этим языком.

Другие были отклонены по причинам: сложность интегрирования с web-приложением, меньшее количество библиотек и фреймворков, сложность синтаксиса и обучения персонала, либо меньшая производительность.

**## Последствия**

Плюсы

·        Присутствие опыта работы с Kotlin у персонала.

·        Ускорение разработки благодаря современному синтаксису.

·        Возможность использования существующих Java-библиотек.

. Привлечение доп. персонала за счет современного ЯП.

Минусы

·        Разработчики с++ недовольны.

. Зарплата разработчиков огромная в сравнении с другими языками.

**## Риски**

Возможна проблема в будущем при дополнительных обновлениях, где будет требоваться большая производительность. - Изучать персоналу алгоритмы, которые будут оптимизировать весь процесс.

**## Затронутые области**

Клиентская часть приложения, development

**## История**

25.09.24: Первоначальное обсуждение вариантов ЯП для клиентской части.