

## **Требования к проекту**

## **1 ВВЕДЕНИЕ**

WealthFlow – веб-приложение, предоставляющее пользователям единую панель для отслеживания акций и криптовалютных активов. Продукт собирает актуальные и исторические данные из публичных финансовых API, позволяя пользователям настраивать список интересующих их активов, просматривать динамику цен на графиках и получать уведомления о достижении пороговых значений цен.

Границы проекта: продукт является платформой для отслеживания и анализа, но не предоставляет функционал для покупки/продажи. Продукт лишь отображает данные, а не прогнозирует изменения цен, используя алгоритмы машинного обучения.

## **2 ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ**

### **2.1 Программные интерфейсы**

Приложение будет взаимодействовать со следующими внешними сервисами:

1 Alpha Vantage API – источник данных по котировкам акций и индексов.

2 CoinGecko API – источник данных по котировкам криптовалют.

3 SendGrid – для реализации системы уведомлений.

Основной стек разработки:

1 Клиентская часть: TypeScript, React.

2 Серверная часть: Java, Spring Boot, PostgreSQL.

### **2.2 Интерфейс пользователя**

Интерфейс будет реализован как одностраничное приложение с использованием React. Будет содержать следующие основные экраны:

1 Главный дашборд: список избранных активов пользователя с текущей ценой, изменением за 24ч и краткосрочным графиком.

2 Страница поиска: поисковая строка и вывод результатов поиска по акциям и криптовалютам.

3 Детальная страница актива: подробная информация об активе, включая расширенный график за различные периоды времени, ключевые метрики.

4 Пользовательский профиль: личные данные пользователя, настройки аккаунта.

### **2.3 Характеристики пользователей**

**2.3.1** Студенты, изучающие финансы. Эта группа представляет студентов, обучающихся в области финансов, которые будут использовать приложение в учебных целях. Характеристика данной группы:

1 Уровень образования: обучение в вузе.

2 Опыт: теоретические знания в области финансов.

3 Техническая грамотность: высокая, быстрое осваивание новых приложений.

**2.3.2** Инвесторы-любители. Эта группа представляет инвесторов-любителей, которые будут использовать приложения для совершения некрупных сделок, мониторинга и анализа рынков.

1 Уровень образования: высшее образование в сфере финансов

2 Опыт: не многолетний опыт в инвестировании.

3 Техническая грамотность: высокая, способность анализировать свой виртуальный портфель.

## 2.4 Предположения и зависимости

Для данного продукта можно выделить следующие предположения и зависимости:

1 Доступ к выбранным финансовым API (Alpha Vantage, CoinGecko) является стабильным и предоставляет данные в заявленном формате.

2 Функционал уведомлений зависит от доступности и возможностей внешних сервисов (SendGrid, Email).

3 Пользователь обладает современным браузером с включенным JavaScript.

4 Проект зависит от стека технологий: Java 17+, Spring Boot, React, PostgreSQL.

### **3 СИСТЕМНЫЕ ТРЕБОВАНИЯ**

#### **3.1 Функциональные требования**

Для данного продукта можно выделить следующие функциональные требования:

- 1 Поиск и просмотр активов
  - 1.1 Система должна предоставлять возможность поиска акций и криптовалют по названию.
  - 1.2 Система должна отображать результат поиска вместе с основными данными.
- 2 Управление списком избранного
  - 2.1 Авторизованный пользователь должен обладать возможностью добавлять активы в список «Избранное» и удалять их из него.
- 3 Отображение данных и графиков
  - 3.1 Система должна отображать информацию о выбранном активе.
  - 3.2 Система должна отображать график цен выбранного актива за различные периоды времени(день, неделя, месяц, год).
- 4 Управление уведомлениями
  - 4.1 Авторизованный пользователь должен иметь возможность установить пороговое значение цены для актива из своего «Избранного»
  - 4.2 Система должна отслеживать изменения цен и отправлять пользователю уведомление, если цена достигла установленного порога.
- 5 Авторизация и аутентификация
  - 5.1 Система должна предоставлять возможность регистрации нового пользователя (логин, пароль).
  - 5.2 Система должна предоставлять возможность аутентификации зарегистрированного пользователя.

#### **3.2 Нефункциональные требования**

**3.2.1 Надежность.** Обоснование важности: приложение должно стабильно работать и отображать актуальные финансовые данные, так как пользователи принимают на их основе решения, связанные с личными финансами. Метрика: время бесперебойной работы  $\geq 99.5\%$ . Обработка ошибок при недоступности внешних API с адекватным сообщением пользователю.

**3.2.2 Производительность.** Обоснование важности: пользовательский интерфейс должен быть отзывчивым, а данные должны загружаться быстро, чтобы не тормозить процесс анализа. Метрика: время загрузки главной

страницы (для авторизованного пользователя) должно составлять не более 2 секунд. Данные графиков должны подгружаться за время  $\leq 1$  секунды.

**3.2.3 Безопасность.** Обоснование важности: необходимо защищать личные данные и настройки пользователей. Метрика: пароли должны храниться в базе данных в хэшированном виде. Все передаваемые данные должны быть защищены с помощью HTTPS. Доступ к API должен контролироваться с помощью JWT-токенов.

**3.2.4 Масштабируемость.** Обоснование важности: архитектура продукта должна позволять добавлять новые источники данных и функции в будущем. Метрика: четкое разделение на backend (Spring Boot) и frontend (React). Использование jOOQ для типобезопасных SQL-запросов к PostgreSQL, что облегчает поддержку и изменение схемы БД.