

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Жизненный цикл разработки программного обеспечения

ПРИЛОЖЕНИЕ ДЛЯ ПОИСКА И ЧТЕНИЯ КНИГ «Raise»

Студент гр. 350504

Гончаров М.Д.

Проверил:

Внук О.М.

Минск 2025

## **ВВЕДЕНИЕ**

Продукт Raise – это веб-приложение, представляющее собой онлайн-каталог книг с широкими возможностями для пользователей. Система позволяет пользователям искать, читать, добавлять, изменять и удалять книги, а также взаимодействовать с другими пользователями через комментарии, лайки и статусы чтения.

Границы проекта: Проект охватывает разработку как клиентской (фронтенд), так и серверной (бэкенд) части приложения. Бэкенд будет реализован как распределенная микросервисная архитектура. Проект не включает создание мобильных приложений для iOS или Android, а также не предполагает офлайн-функционала.

## **2 ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ**

### **2.1 Программные интерфейсы**

Продукт будет взаимодействовать со следующими внешними и внутренними системами:

- PostgreSQL: Реляционная база данных для хранения информации.
- Kafka: Система очередей сообщений, используемая для асинхронной связи между микросервисами.
- API Gateway: Единая точка входа для всех запросов с фронтенда, маршрутизирующая их к соответствующим микросервисам.
- Redis: Для кеширования данных.
- Nginx: Веб-сервер, который будет обслуживать статические файлы фронтенда.

Основной стек внутренних сервисов:

- Для серверной части: Java, Spring Boot, Spring Cloud, Hybernate, Python, FastAPI, SQLAlchemy.
- Для клиентской части: TypeScript, React.

### **2.2 Интерфейс пользователя**

Система будет взаимодействовать с пользователем через веб-сайт. Основные элементы интерфейса и их поведение:

- Страница каталога: Отображение списка книг с возможностью поиска, фильтрации и сортировки.
- Страница книги: Детальное представление выбранной книги, включая обложку, описание, автора, год публикации и содержание.
- Читалка: Специальный интерфейс для чтения книги, с возможностью навигации по страницам или главам.
- Пользовательский профиль: Раздел, где пользователь может просмотреть свои статусы чтения ("Читаю", "Бросил читать", "Прочитал"), историю комментариев и лайков, а также управлять своими данными.
- Формы для CRUD операций: Интерфейсы для создания, изменения и удаления книг.

### **2.3 Характеристики пользователей**

Рядовые пользователи: Основная аудитория. Это люди с разным уровнем технической грамотности, которые хотят читать книги онлайн, оставлять комментарии и отмечать прочитанное. Они ожидают интуитивно понятный и простой в использовании интерфейс.

Авторы/Администраторы: Пользователи с расширенными правами, отвечающие за наполнение и модерацию контента. У них есть возможность

добавлять, редактировать и удалять книги. Они могут быть более технически подкованы и нуждаться в более сложных инструментах для управления данными.

## **2.4 Предположения и зависимости**

Предполагается, что пользователи имеют стабильное интернет-соединение для доступа к приложению.

Сервисы, такие как PostgreSQL и Kafka, должны быть доступны и работать без сбоев для корректного функционирования системы.

Обеспечение безопасности пользовательских данных и контента является критически важным.

## **3 СИСТЕМНЫЕ ТРЕБОВАНИЯ**

### **3.1 Функциональные требования**

- Система должна предоставлять пользователю возможность поиска книг по названию, автору и жанру.
- Пользователь должен иметь возможность прочитать книгу онлайн.
- Система должна позволять аутентифицированным пользователям создавать новые книги, редактировать и удалять существующие.
- Пользователь должен иметь возможность оставлять комментарии к книгам и просматривать комментарии других пользователей.
- Пользователь должен иметь возможность ставить "лайки" книгам.
- Пользователь должен иметь возможность отмечать книги статусами: "Читаю", "Прочитал", "Бросил читать".

### **3.2 Нефункциональные требования**

#### **3.2.1 Атрибуты качества**

- Надёжность: Система должна быть устойчива к сбоям. Так как архитектура распределенная, сбой одного микросервиса не должен приводить к полному отказу всей системы. Для этого важна правильная настройка отказоустойчивости и мониторинга, измеряемая через показатель MTBF (Mean Time Between Failures).
- Производительность: Приложение должно обрабатывать запросы пользователей быстро, особенно при поиске и загрузке книг. Это измеряется через время отклика для ключевых операций, таких как поиск и чтение.
- Безопасность: Пользовательские данные (пароли, личная информация) и права доступа должны быть надёжно защищены. Все данные, передаваемые между фронтендом и бэкендом, должны быть зашифрованы. Это измеряется через количество выявленных уязвимостей при тестировании на проникновение.
- Масштабируемость: Архитектура микросервисов должна позволять масштабировать каждый сервис независимо, чтобы справляться с ростом нагрузки и количества пользователей. Это можно измерять по максимальному количеству одновременных пользователей, которое система может поддерживать без деградации производительности.
- Удобство использования (Usability): Интерфейс должен быть интуитивно понятным для пользователей с разным уровнем технической грамотности. Это можно оценить через тестирование пользовательского опыта и сбор обратной связи.

## **4 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ**

### **4.1 Глоссарий предметной области**

В данном разделе представлен глоссарий предметной области системы FitTrack, разработанный на основе анализа требований и модели вариантов использования.

Глоссарий служит основой для формирования объектной модели системы, так как определяет ключевые понятия, их свойства и взаимосвязи. Каждый термин отражает важную сущность системы, которая впоследствии будет реализована в виде класса, объекта или компонента в процессе объектно-ориентированного проектирования.

1. Пользователь (User): Зарегистрированный человек, ведущий учёт питания, тренировок и личных целей в системе.

2. Администратор (Admin/Author): Пользователь с расширенными правами. Управляет каталогом книг, может создавать, удалять, изменять книги.

3. Книга (Book): Основной объект манипуляции в приложении, хранит в себе метаданные книги, хранит также путь к файлу книги.

4. Отзыв (Review): Отзыв представляет собой объект, хранящий цифровую и текстовую оценку книги.

5. Отметка нравится (Like): Объект, характеризующий то, насколько книга нравится пользователям.

6. Статусы (Statuses): Пользователи могут добавлять себе книги в разные свои локальные разделы: читаю, прочитал, бросил читать.

7. Аутентификация (Auth): Процесс входа пользователя в систему с проверкой учётных данных.

Также можно выделить ряд внутрисистемных терминов:

1. Микросервис – отдельный сервис со своей зоной ответственности.

2. Конечная точка (Endpoint)– конкретная конечная точка отдельного микросервиса, которая выполняет какую-то отдельную задачу.

3. Трансфер объект (Dto) – объект, который представляет собой входную/выходную сущность от сервера.

4. Сущность (Entity) – класс, относительно которого можно работать с базой данных.

### **4.2 Актеры системы**

В данном разделе определены основные актеры (участники взаимодействия с системой) в рамках модели прецедентов Raise. Каждый актёр представляет собой роль, которую может выполнять человек или внешняя подсистема при взаимодействии с системой. Описание актёров необходимо для понимания границ системы, распределения

функционала и последующего построения диаграмм вариантов использования (Use Case Diagram).

Guest (Гость) – неавторизованный пользователь, может просматривать публичную информацию, регистрироваться и входить.

User (Пользователь) – основной актёр, взаимодействует с системой после входа: ищет книги, добавляет книги с список для чтения, отмечает понравившиеся книги.

Author (Автор) – актёр, который может создавать, изменять и удалять свои созданные объекты книг.

Admin (Администратор) – актёр с расширенными правами: редактирует каталог книг, управляет справочными данными и модерацией.

System (Система) – выполняет автоматические действия: проверка валидности и консистентности имеющихся данных.