САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №3

Выполнил:

Захарчук Александр

К3341

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- Выделить самостоятельные модули в вашем приложении;
- Провести разделение своего API на микросервисы (минимум, их должно быть 3);
- Настроить сетевое взаимодействие между микросервисами.

Ход работы

Монолитный сервис для управления рецептами из прошлой лабораторной работы был разделен на 3 микросервиса: user-service, recipe-service, social-service.

User Service

Отвечает за регистрацию, аутентификацию, авторизацию и управление пользователями.

Основные функции:

- Регистрация и вход пользователя.
- Обновление профиля и удаление аккаунта.
- Валидация JWT токенов по запросу других сервисов.

Recipe Service

Управляет рецептами и связанными с ними ингредиентами.

Основные функции:

- CRUD-операции для рецептов.
- Поддержка категорий (тип блюда, уровень сложности).
- Ассоциация рецептов с ингредиентами.

Social Service

Обрабатывает социальные действия, связанные с рецептами: лайки, комментарии, а также подписки пользователей.

Основные функции:

- Добавление, удаление и получение лайков.
- Комментирование рецептов (создание, чтение, обновление, удаление).

- Подписка и отписка от пользователей.
- Получение списка подписок и подписчиков.

В результате разделения сервисы стали независимыми, они имеют изолированный код, а также индивидуальную базу данных для каждого из них.

Рисунок 1 - Swagger для user-service

Рисунок 2 - Swagger для recipe-service

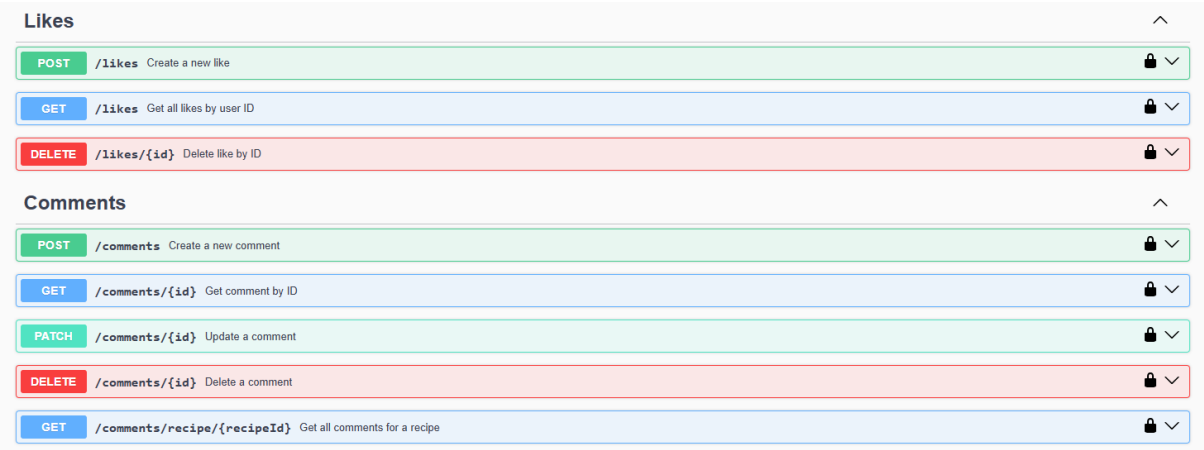

Рисунок 3 - Swagger для social-service

Вывод

В ходе работы был произведён переход от монолитной архитектуры к микросервисной. Приложение было разделено на отдельные сервисы: управление пользователями, рецептами, подписками и социальными взаимодействиями (лайки, комментарии, сохранения). Каждый микросервис стал независимым, с собственной базой данных и четко определёнными API. Это повысило масштабируемость, упростило поддержку и тестирование, а также обеспечило гибкость в дальнейшем развитии проекта.