

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя/Лабораторная работа

Выполнил:

Захарчук Александр

Группа К3441

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Необходимо спроектировать набор следующих диаграмм:

- общая архитектура решения (сервисы и их взаимосвязи, клиент-серверное взаимодействие);
- диаграмма компонентов;
- диаграммы БД по каждому сервису;
- диаграммы основных пользовательских сценариев (те сценарии, которые позволяют вашим приложением полноценно воспользоваться, пройти весь путь).

Ход работы

Рассматриваемый сервис предназначен для создания полнофункциональной онлайн-платформы кулинарных рецептов, которая позволяет пользователям регистрироваться, публиковать и находить рецепты, взаимодействовать с контентом через лайки и комментарии, а также формировать сообщество через систему подписок на других авторов.

Архитектура системы представляет собой микросервисное приложение для сайта рецептов, состоящее из трёх независимых сервисов. Сервис пользователей (User Service) управляет регистрацией, аутентификацией и профилями пользователей, используя JWT-токены для авторизации. Сервис рецептов (Recipe Service) отвечает за создание, редактирование и хранение рецептов с ингредиентами. Социальный сервис (Social Service) обеспечивает функциональность комментариев и лайков к рецептам. Каждый сервис имеет собственную базу данных PostgreSQL и работает на выделенном порту. Клиенты (веб-браузеры и мобильные приложения) взаимодействуют напрямую со всеми сервисами через HTTP/REST API, при этом Recipe Service и Social Service обращаются к User Service для валидации JWT-токенов, а Social Service дополнительно проверяет существование рецептов через Recipe Service перед созданием комментариев и лайков. Диаграмма архитектуры системы представлена на рисунке 1.

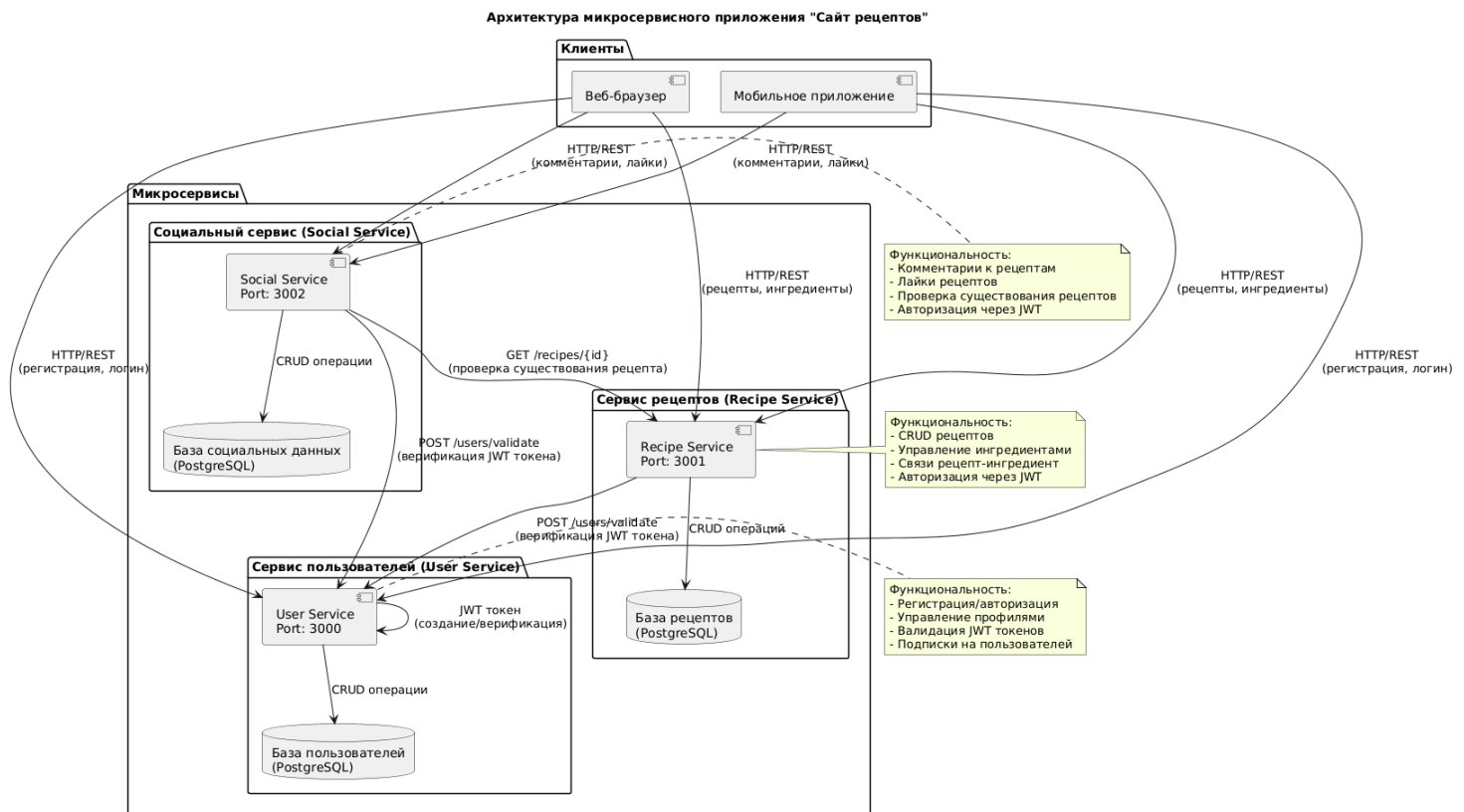


Рисунок 1 - Диаграмма архитектуры системы

User Service включает компоненты для управления пользователями (User Controller, User Service) и подписками (Subscription Controller, Subscription Service), обеспечивая регистрацию, аутентификацию через JWT и социальные связи. Recipe Service содержит компоненты для работы с рецептами (Recipe Controller, Recipe Service) и ингредиентами (Ingredient Controller, Ingredient Service), управляя кулинарной базой данных. Social Service предоставляет компоненты для комментариев (Comment Controller, Comment Service) и лайков (Like Controller, Like Service), обрабатывающие социальные взаимодействия. Диаграмма компонентов представлена на рисунке 2.

Диаграмма компонентов микросервисного приложения "Сайт рецептов"

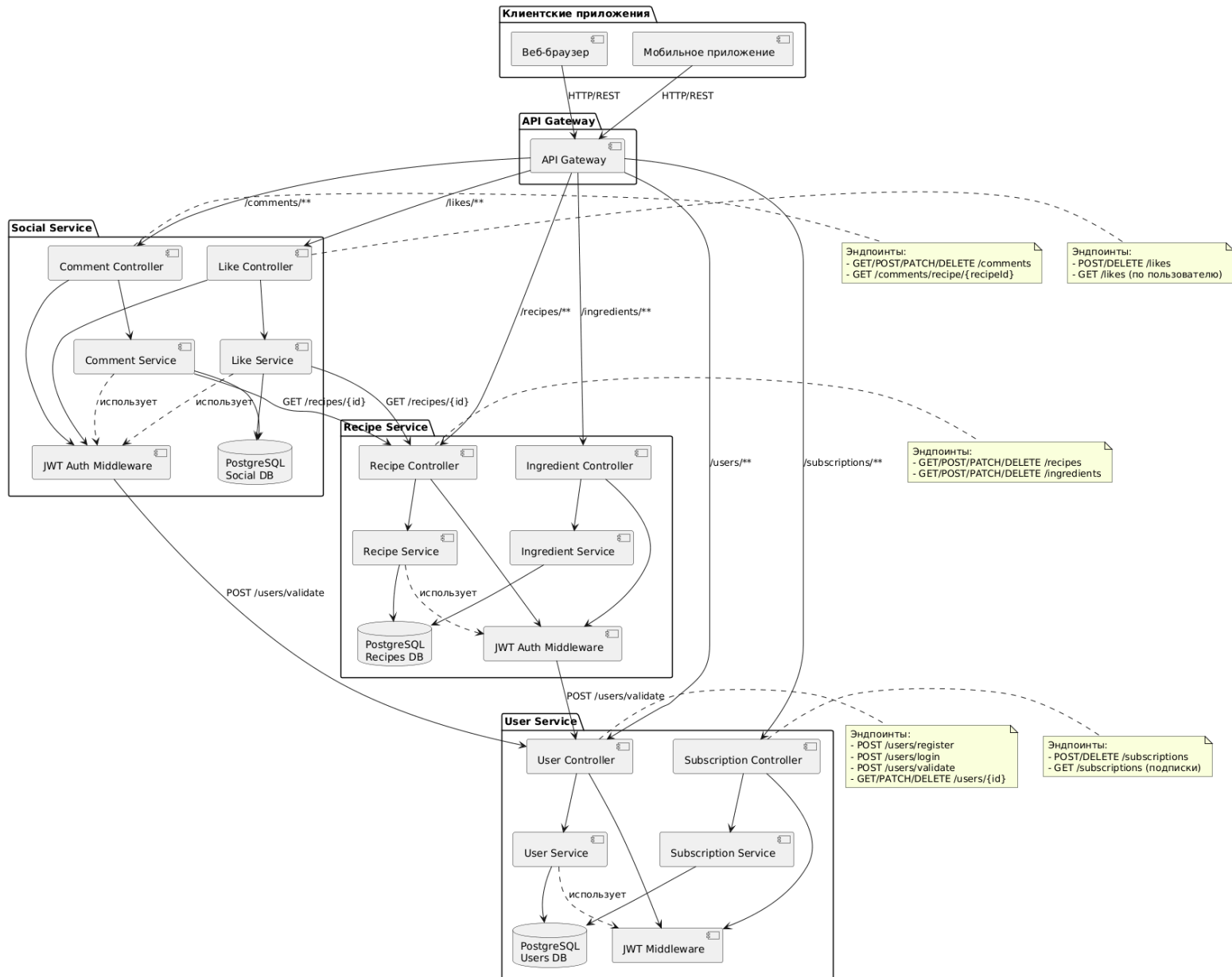


Рисунок 2 - Диаграмма компонентов

User Service DB содержит таблицу *users* для хранения профилей пользователей с уникальными *username* и *email*, и таблицу *subscriptions* для управления подписками между пользователями с ограничением против самоподписки.

Recipe Service DB включает таблицу *recipe* для рецептов с метаданными (тип блюда, сложность, время приготовления), таблицу *ingredient* для справочника ингредиентов и связующую таблицу *recipes_ingredients* с количеством и единицами измерения для связи многие-ко-многим.

Social Service DB состоит из таблиц *comment* для комментариев к рецептам и *like* для лайков, обе содержат ссылки на пользователей и рецепты для отслеживания социальных взаимодействий.

Схема баз данных представлена на рисунке 3.

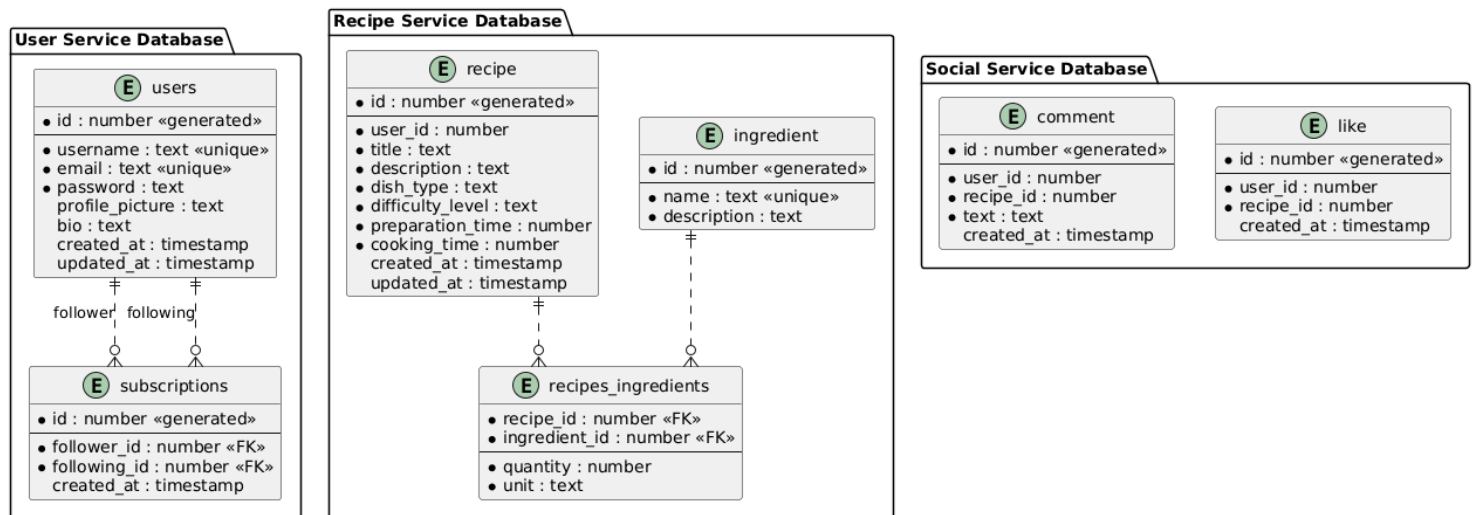


Рисунок 3 - Диаграмма баз данных

Сценарий 1: Регистрация и создание первого рецепта

Пользователь начинает работу с системой с регистрации, создавая учетную запись через User Service. После успешной аутентификации и получения JWT-токена он переходит к созданию своего первого рецепта. Recipe Service проверяет валидность токена через User Service, после чего сохраняет рецепт в базу данных. Этот сценарий охватывает полный путь от нового пользователя до активного автора контента. Диаграмма данного сценария показана на рисунке 4.

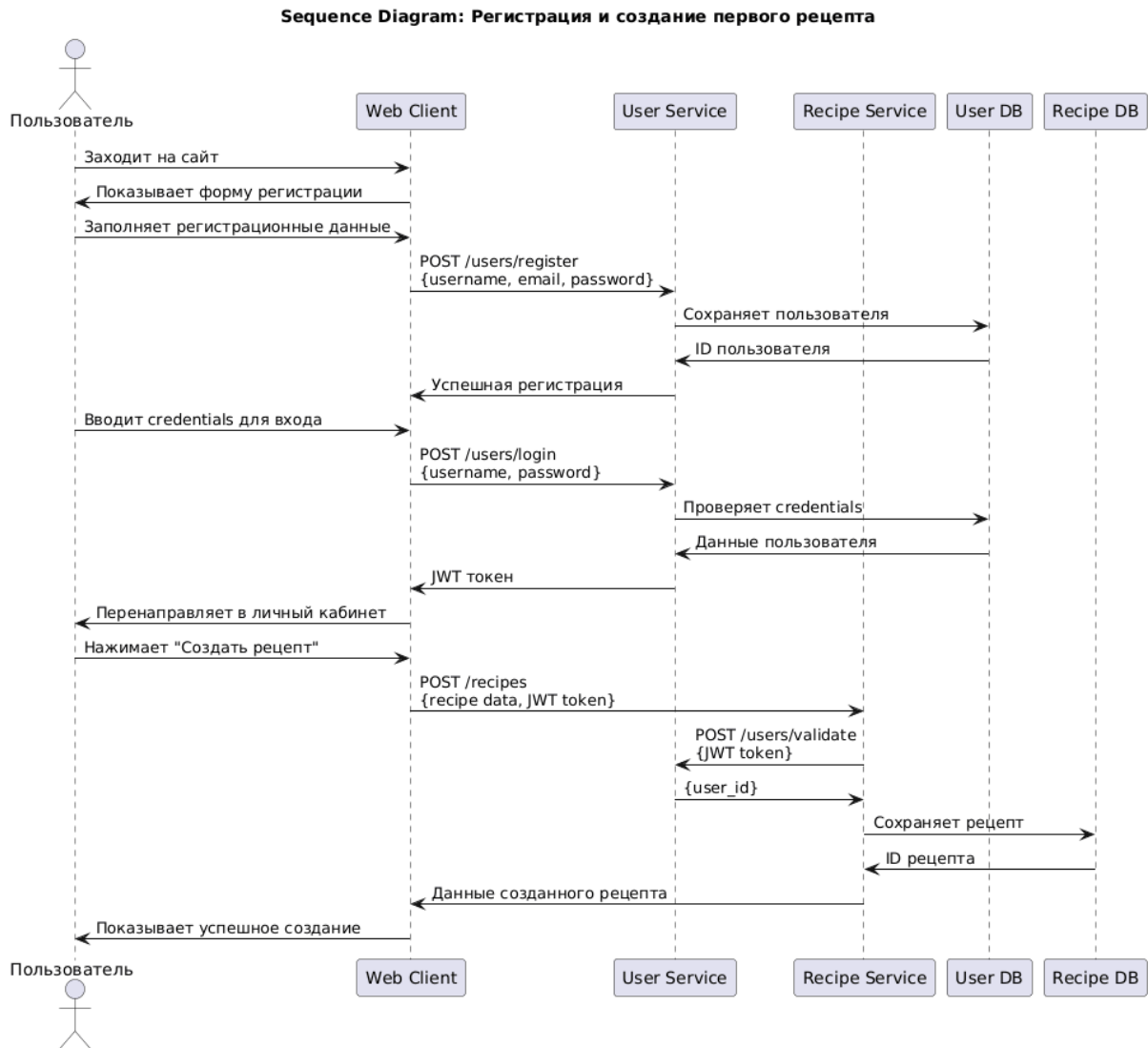


Рисунок 4 - Сценарий регистрации и создание рецепта

Сценарий 2: Просмотр рецепта и социальное взаимодействие

Пользователь просматривает ленту рецептов, выбирает интересующий рецепт и взаимодействует с ним через социальные функции. При попытке поставить лайк или оставить комментарий Social Service последовательно проверяет аутентификацию через User Service и существование рецепта через Recipe Service, обеспечивая целостность данных. Сценарий демонстрирует типичное использование платформы для потребления контента и социальной активности. Диаграмма данного сценария показана на рисунке 5.

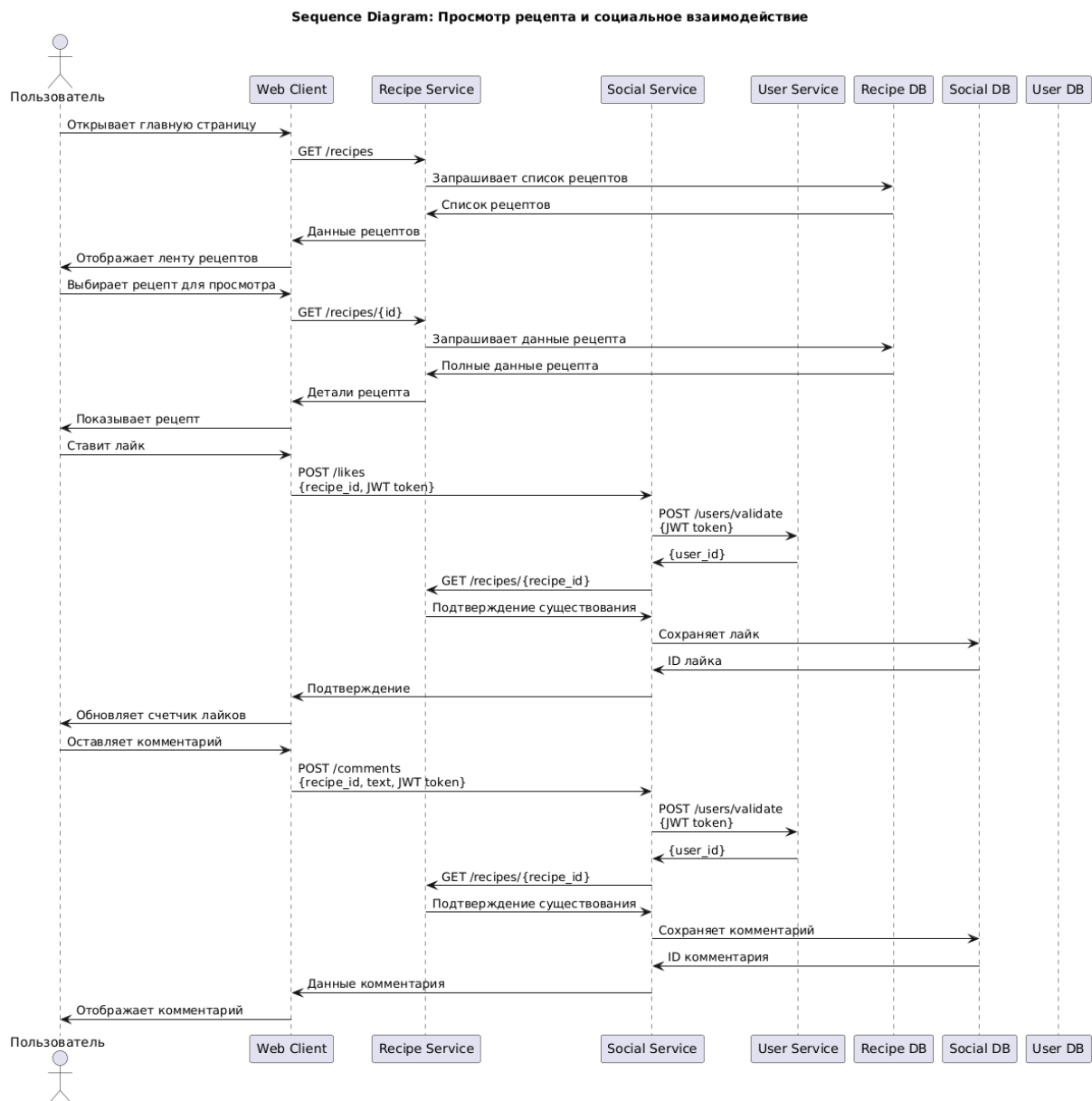


Рисунок 5 - Сценарий просмотра и взаимодействия с рецептом

Сценарий 3: Поиск рецептов и управление подписками

Активный пользователь использует поисковые возможности платформы для нахождения рецептов по фильтрам, изучает профили авторов и управляет своими подписками. Система обеспечивает персонализированный опыт, позволяя пользователю формировать ленту контента от интересующих авторов. Сценарий показывает продвинутые

функции кураторства контента и построения персональной кулинарной ленты. Диаграмма данного сценария показана на рисунке 6.

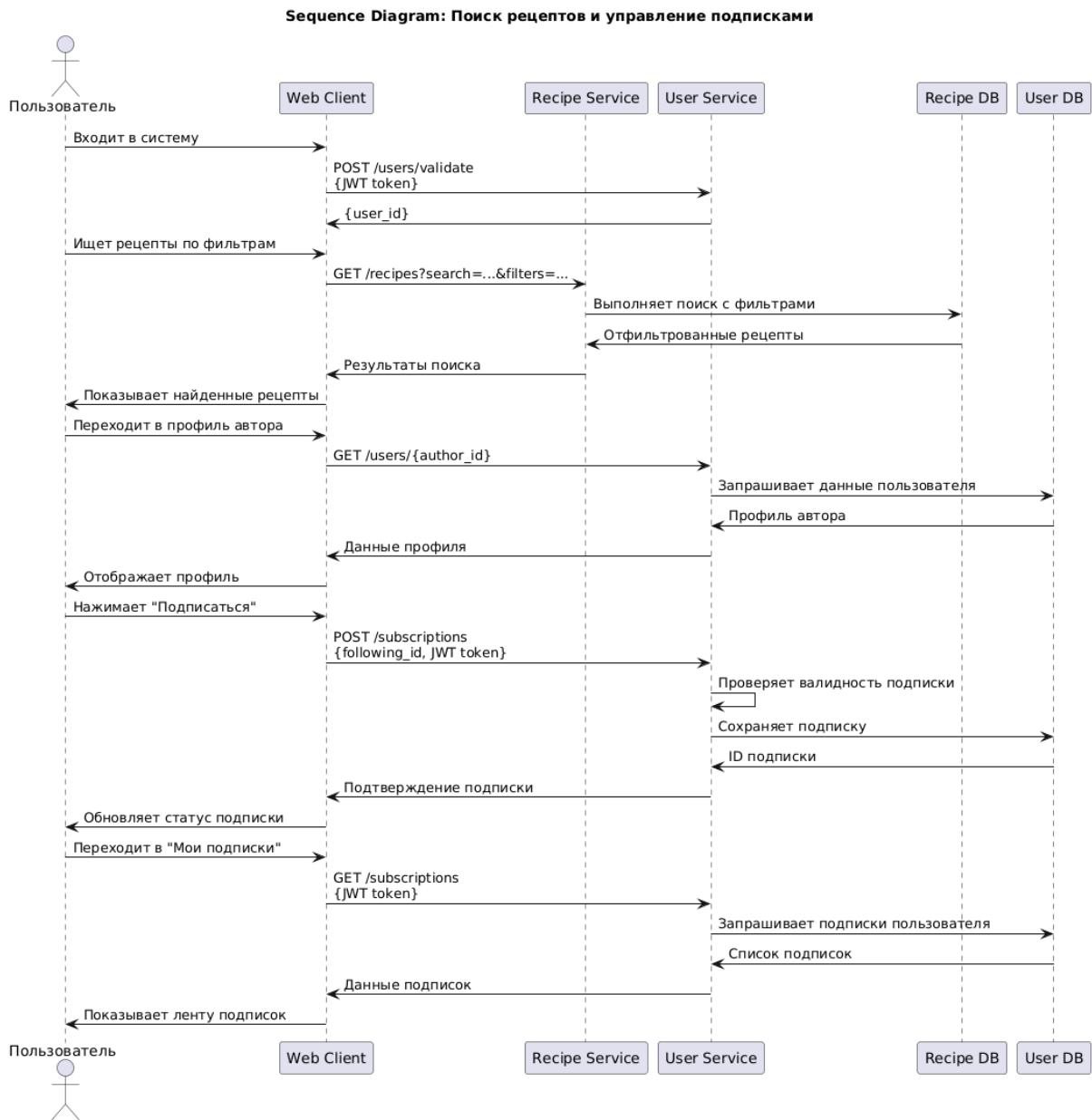


Рисунок 6 - Сценарий поиска рецептов и подписок

Вывод

В ходе работы была спроектирована и детализирована микросервисная архитектура приложения для сайта рецептов, состоящая из трех

независимых сервисов: User Service для управления пользователями и аутентификации, Recipe Service для работы с рецептами и ингредиентами, и Social Service для обработки комментариев и лайков. Были разработаны диаграммы, показывающие общую архитектуру, компоненты системы, структуры баз данных каждого сервиса и ключевые пользовательские сценарии, что демонстрирует слабую связность сервисов, четкое разделение ответственности и эффективное взаимодействие между компонентами через REST API с централизованной JWT-аутентификацией.