**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра технологий программирования**

Лазовская Татьяна Андреевна

Дроздова Юлия Витальевна

Рыжков Ярослав Александрович

Сорока Александр Владимирович

Проектирование и разработка веб-приложения и мобильного приложения. Построение окружения на основе docker-контейнеров

Отчет по лабораторной работе № 6

 «Проектирование человеко-машинных интерфейсов»

студентов 4 курса 13 группы

**Преподаватель**

**Давидовская М. И.**

**Минск 2023**

**Цели лабораторной работы**: разработка технического задания для внутреннего использования (стратегия дизайна, создание диаграммы бизнес-процессов и диаграммы вариантов использования, диаграммы деятельности, диаграммы классов и объектов, диаграммы компонентов, диаграммы развертывания, схему базы данных, используя диаграммы "сущность-связь" (Entity RelationShip Diagram — ERD), диаграммы объектно-реляционного отображения, создание структуры проекта на github, использование github projects для управления проектом, составление плана работы по бекенду (разработка API), добавление автоматических тестов, установка докера и контейнеров, необходимых для разработки, встроить генерацию необходимых файлов для Swagger одним из этапов сборки, опубликовать проект на Google Cloud, настройка Better Code Hub.

**Стратегия дизайна**:

• **заинтересованные стороны — определить, кто является потенциальной заинтересованной стороной, которая будет использовать приложение;**

Целевыми пользователями являются подростки и молодые люди, которые любят читать книги, делиться своим мнением, обсуждать книги. Более того приложение могут использовать различные книжные блогеры для роста своей аудитории, а также для получения информации о книжных новинках.

**• видение продукта заинтересованными лицами (задачи продукта) — какие задачи**

**пользователей должно решать разрабатываемое приложение;**

Приложение должно поддерживать возможность поиска книг, добавление книг в свои подборки, чтение и написание рецензий, отслеживание прочитанных книг, добавление книг в список “Хочу прочитать”, отслеживание профилей друзей (их прочитанные книги, отзывы, рецензии).

**• конфликты и противоречия — наличие конфликтов, противоречий по функционалу**

**приложения между различными заинтересованными лицами;**

Явных конфликтов и противоречий не обнаружено.

**• измеримые критерии успешности — по каким критериям будет оцениваться успешность приложения;**

Количество активных пользователей, отзывы на приложение, обращения в техподдержку приложения.

**• технические возможности и ограничения — в данном разделе необходимо указать используемые технологии и платформы для разработки, а также перечень систем, с которыми интегрируется разрабатываемое приложение;**

Будет разработана Веб-версия приложения forWORDS, а также мобильная версия для ОС Андроид. Для развертывания приложения будет использован Docker.

**• представления заинтересованных лиц о пользователях (целевая аудитория) —кто целевая аудитория приложения;**

Целевыми пользователями являются подростки и молодые люди, которые любят читать книги, делиться своим мнением, обсуждать книги. Также приложение могут использовать различные книжные блогеры для роста своей аудитории, а также для получения информации о книжных новинках.

**• бюджет и график проекта — бюджет и примерный график реализации проекта.**

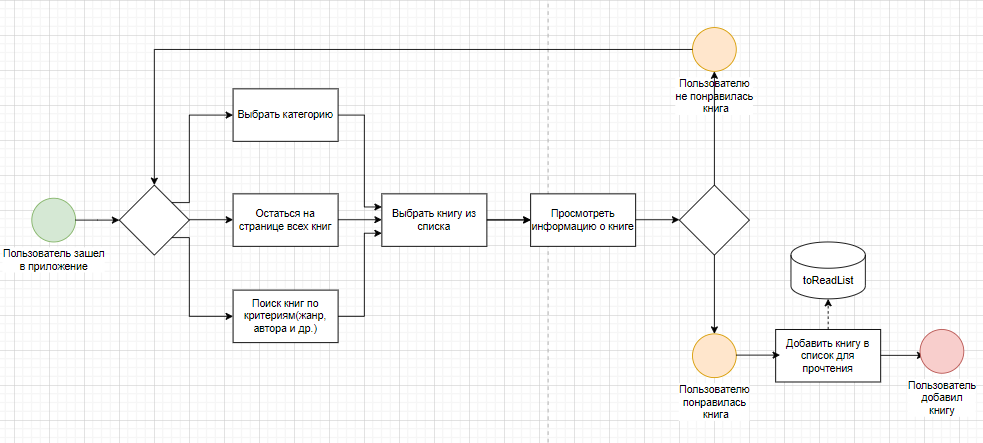
Бюджет проекта с учетом выплаты заработной платы 4 разработчикам, работающим удаленно, а также с учетом трат на маркетинг и рекламу приложения будет составлять около 25000$.

Примерный график реализации проекта:

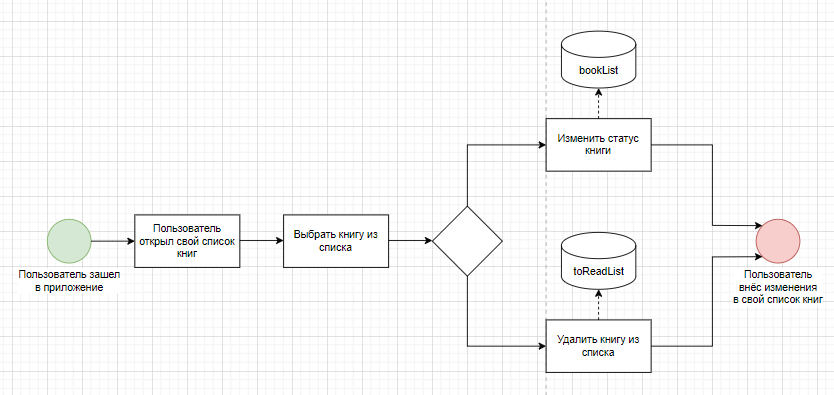
1. Проектирование - 1 месяц
2. Разработка - 1 месяц
3. Тестирование, релиз - 1 месяц

**Диаграммы бизнес-процессов**

БП “Добавление книги в список для прочтения”



БП “Внесение изменений в список книг пользователя”



**Диаграммы вариантов использования**

Диаграмма варианта использования “Вход в систему. Регистрация”:

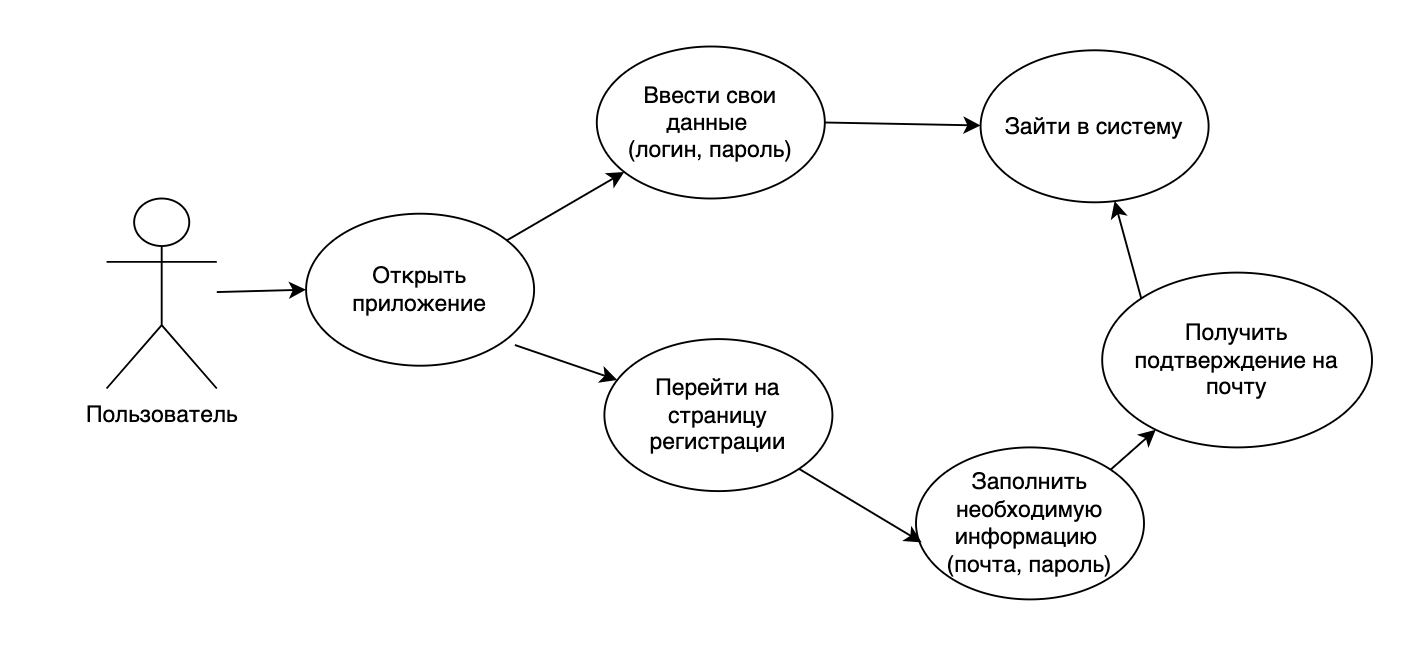
****

Диаграмма варианта использования “Добавить книгу в “Хочу прочитать”:

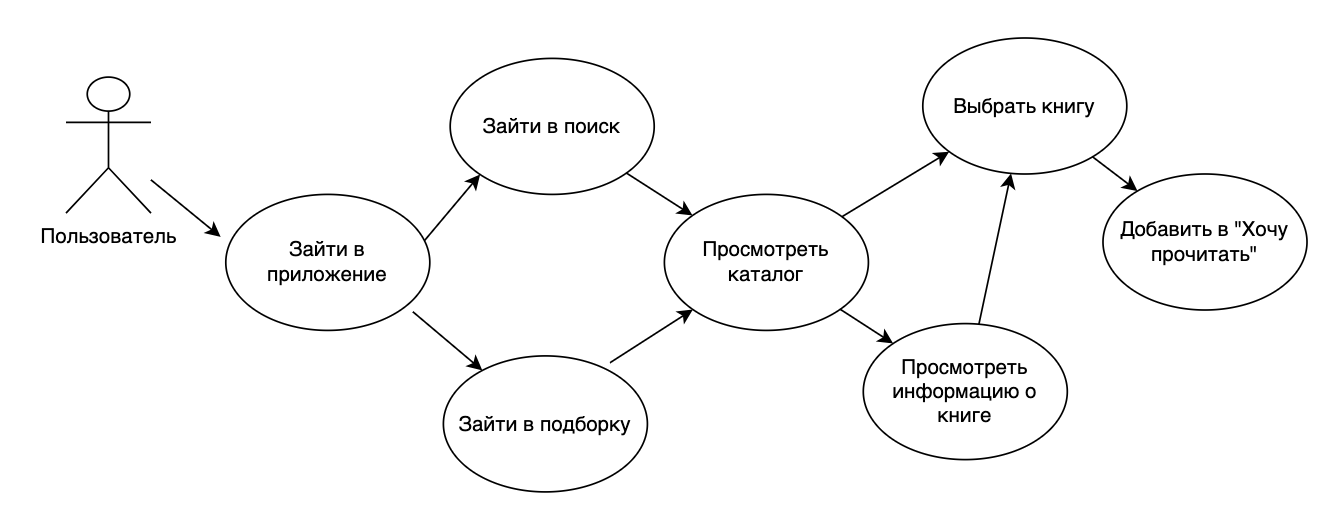
****

Диаграмма варианта использования “Написать рецензию на книгу”:

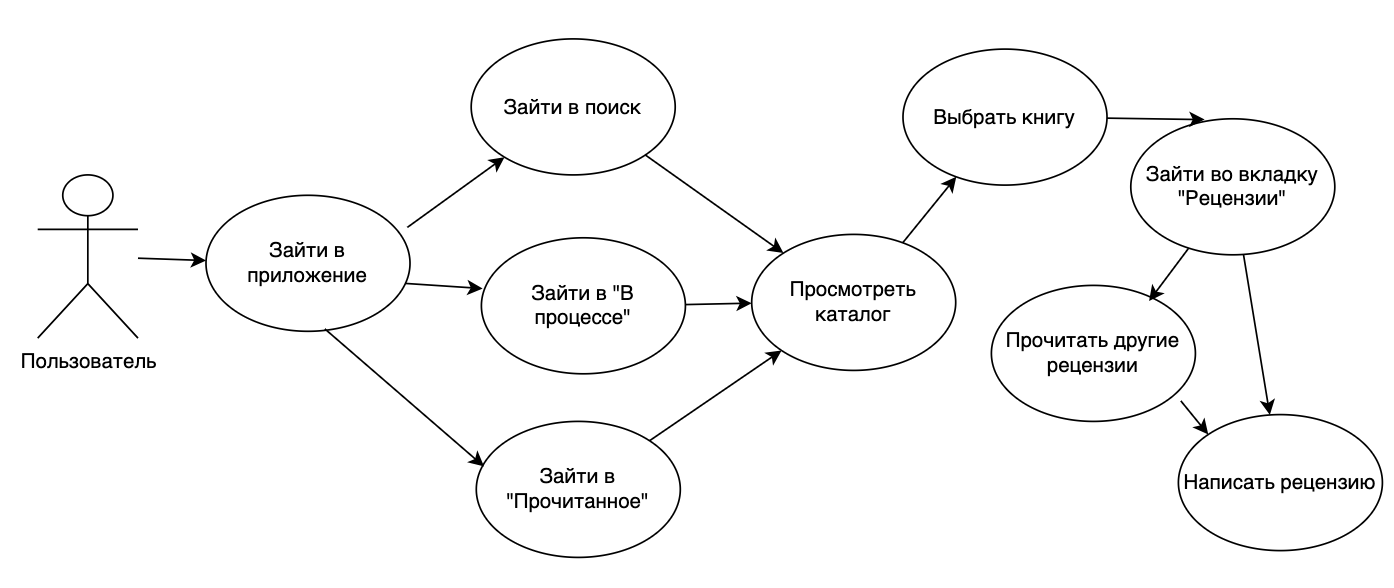
****

Диаграмма вариантов использования “Добавить книгу в подборку”:

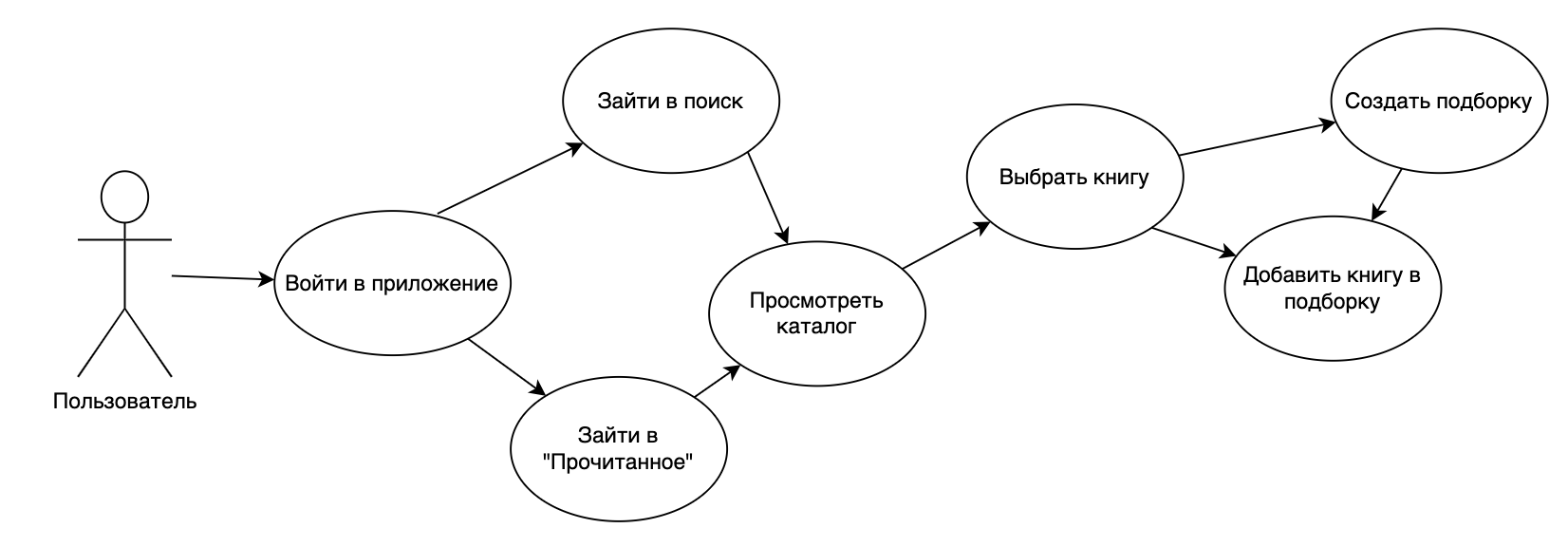
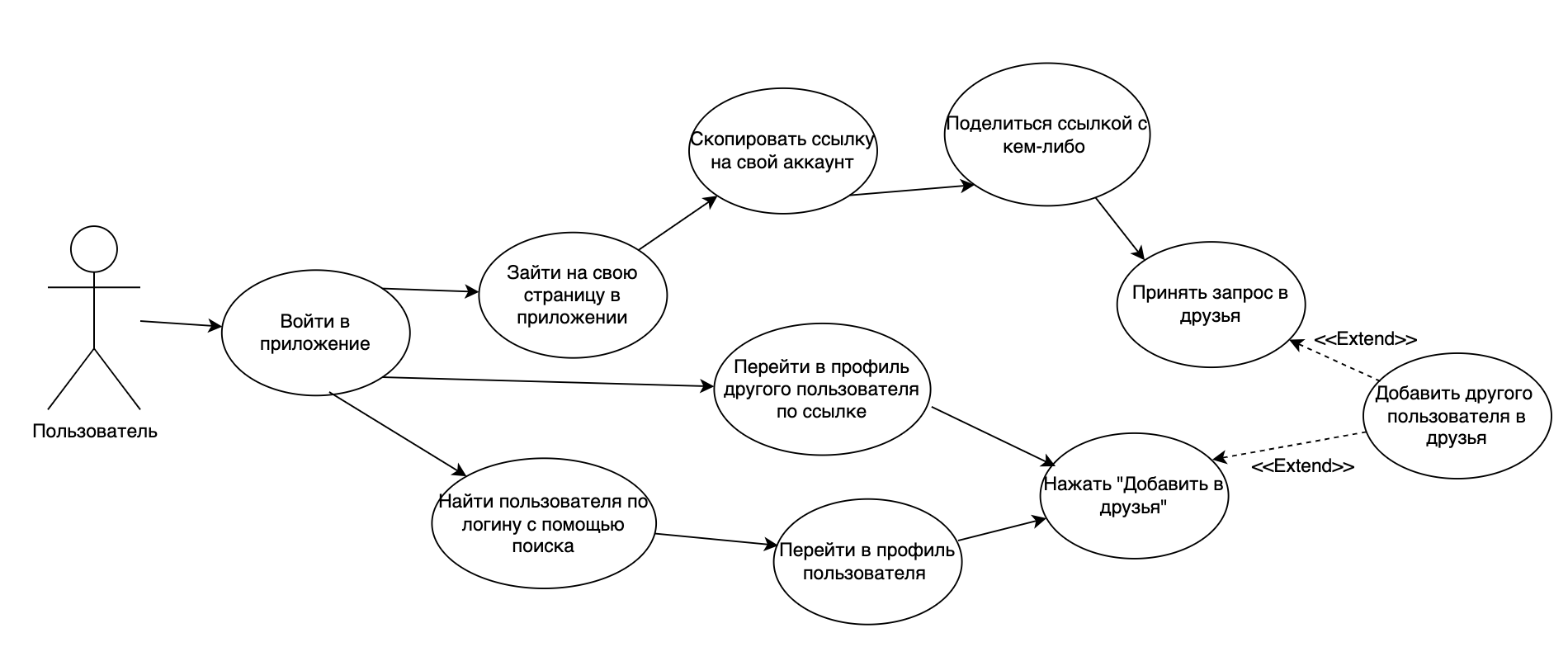
****

Диаграмма вариантов использования “Трекинг прогресса чтения”:



Диаграмма вариантов использования “Добавить другого пользователя в друзья”:



**Диаграммы деятельности**

Диаграмма деятельности “Вход в систему”:

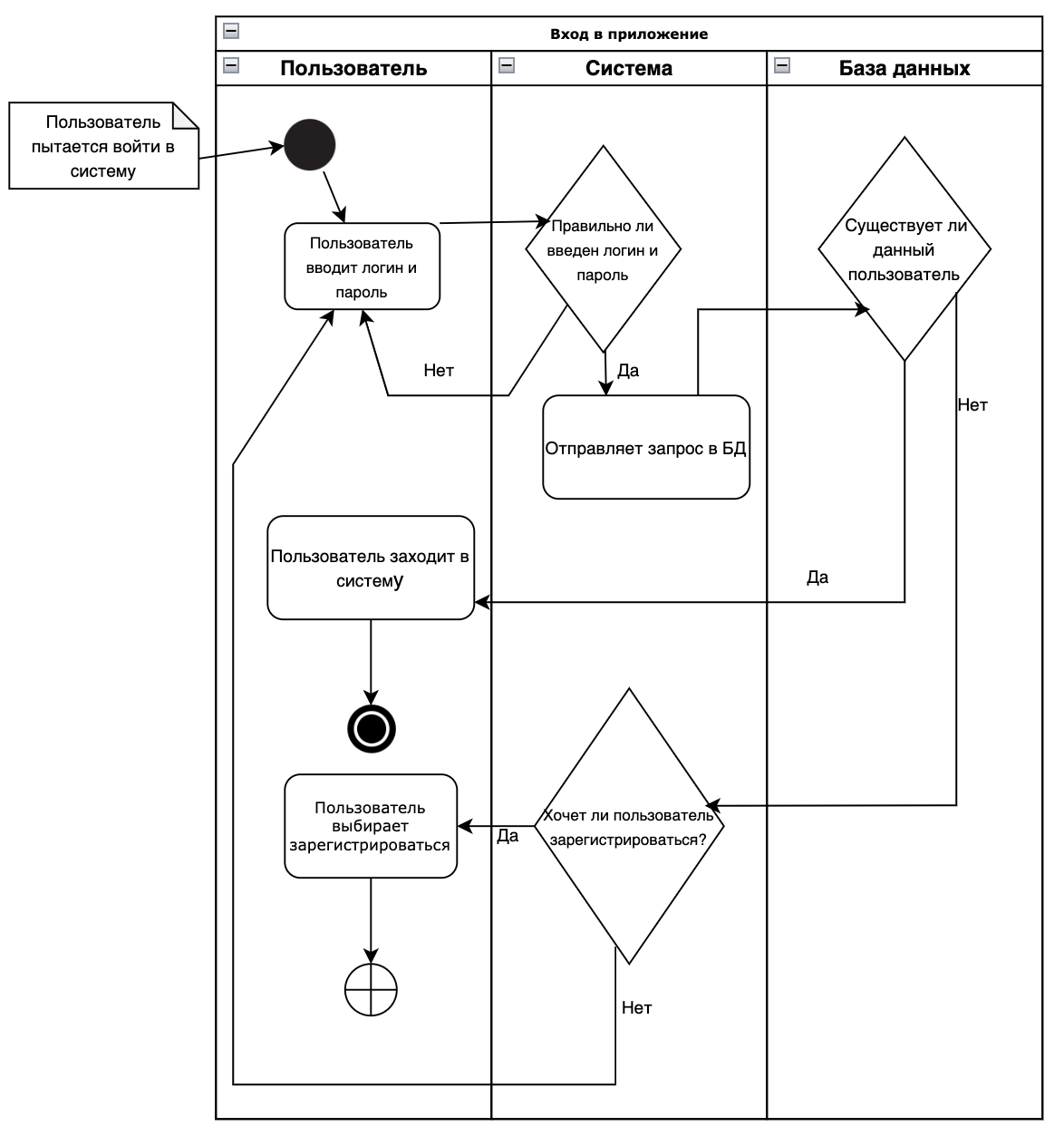
****

Диаграмма деятельности “Регистрация пользователя”:

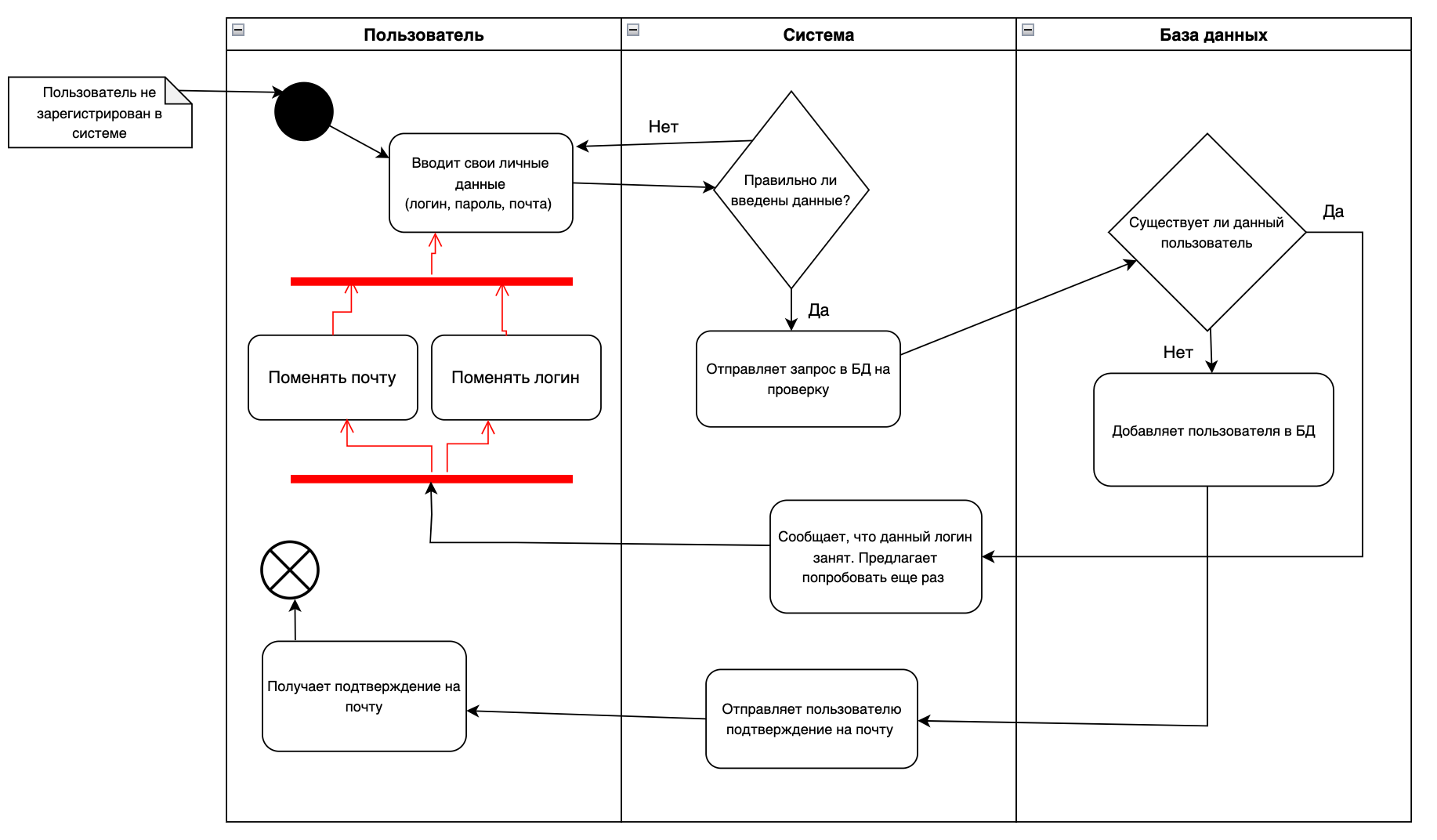
****

Диаграмма деятельности “Добавить книгу в “Хочу прочитать”:

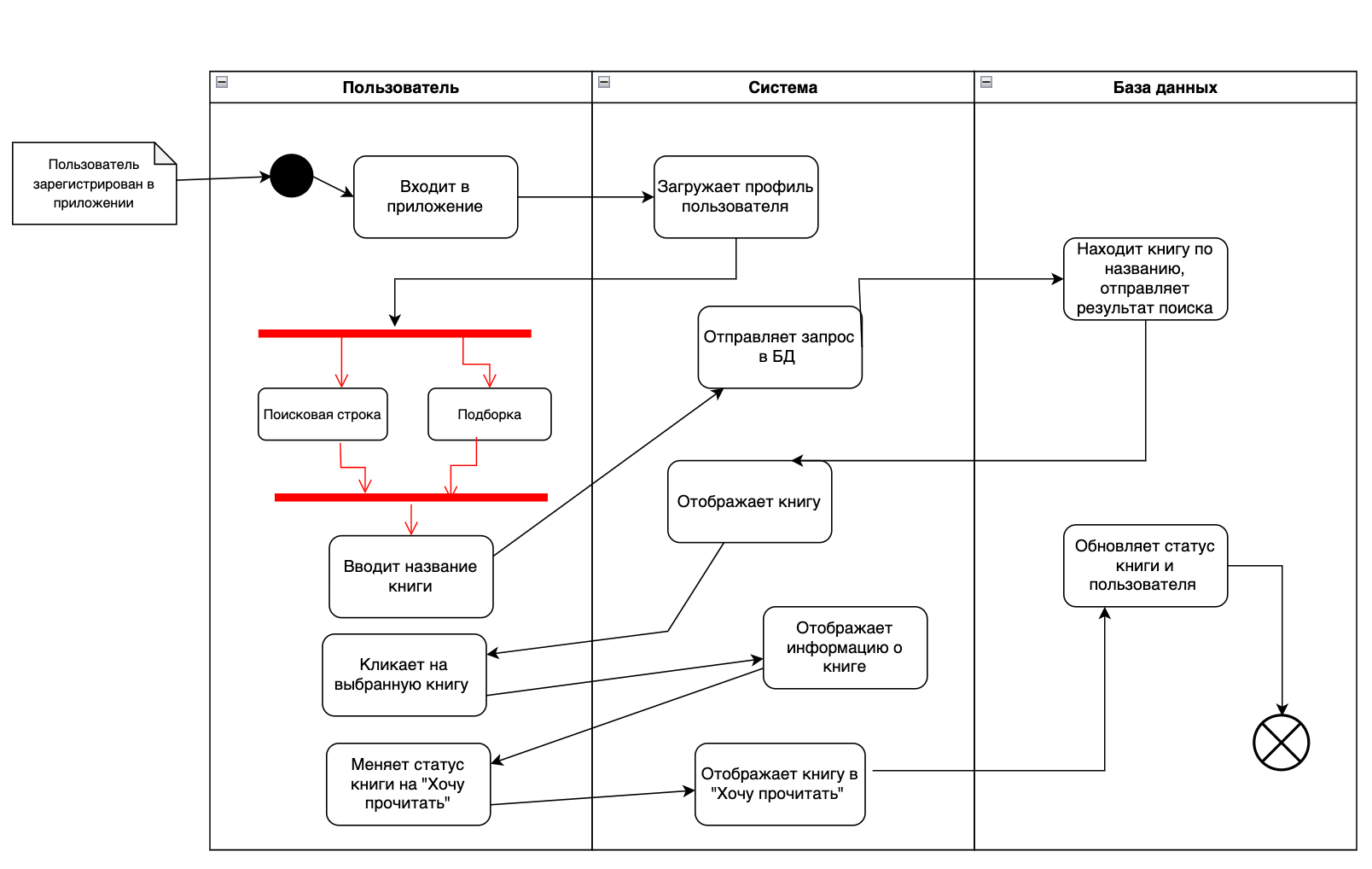


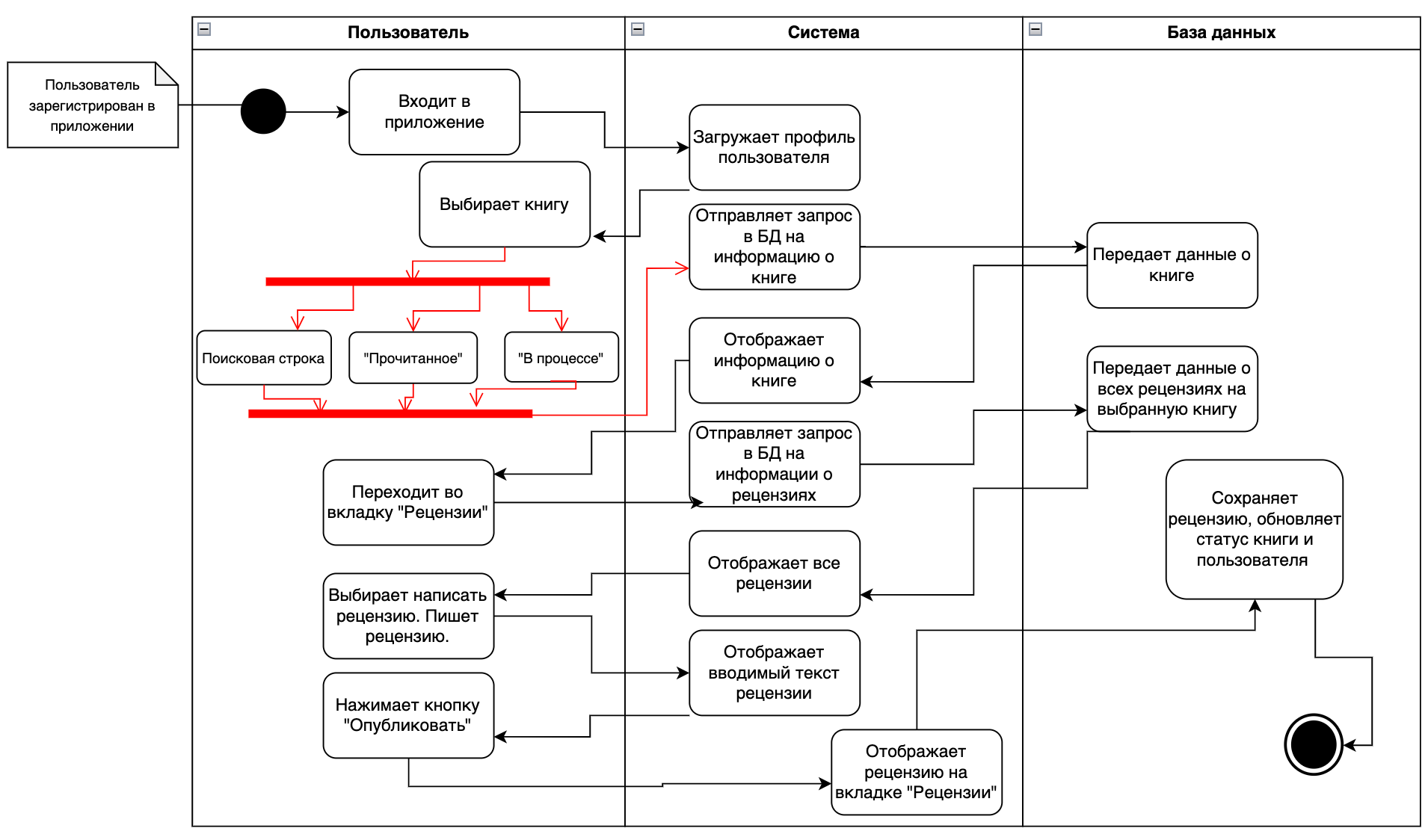
Диаграмма деятельности “Добавить книгу в “Написать рецензию”:

Диаграмма деятельности “Книжный челлендж”:

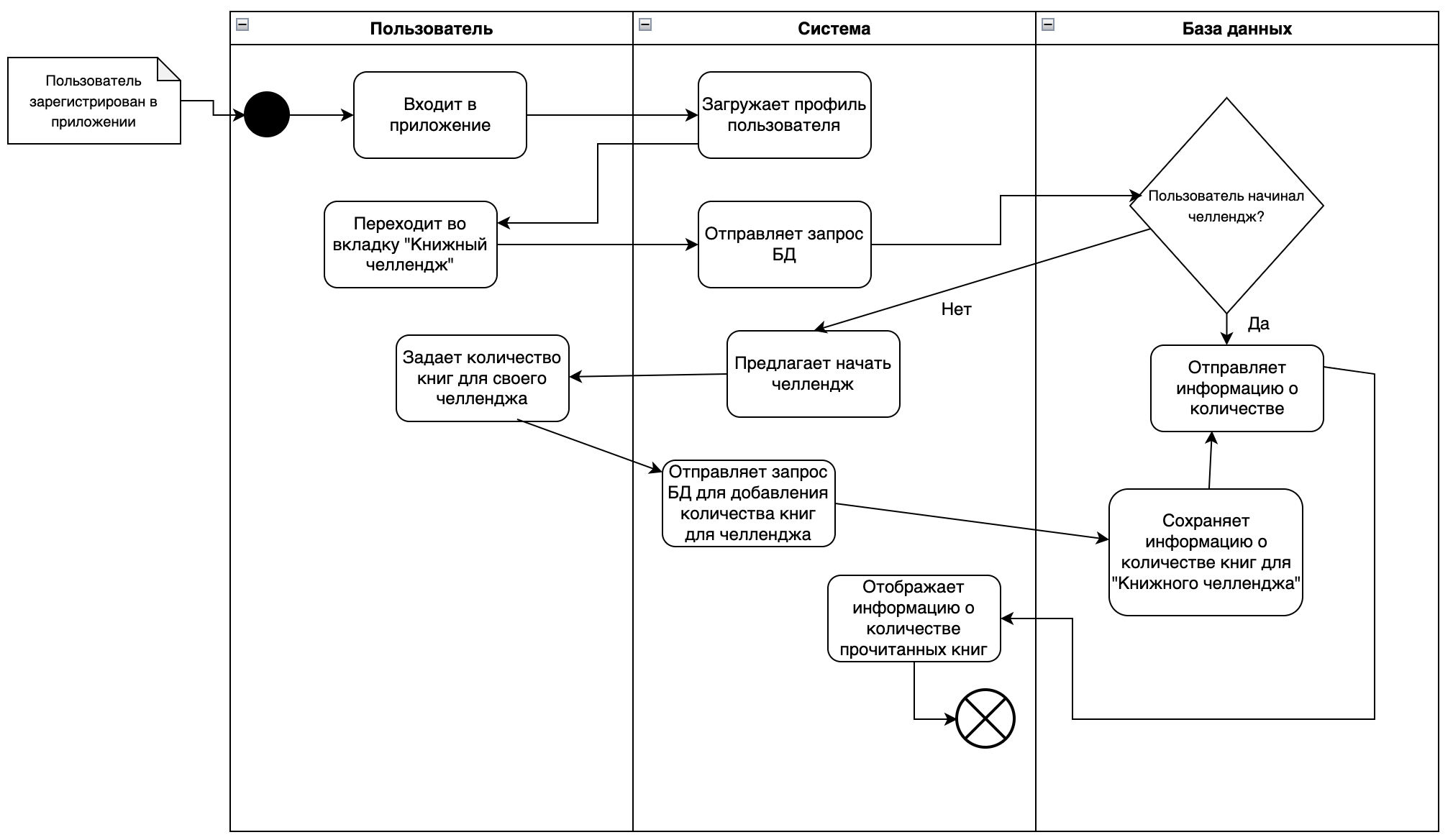
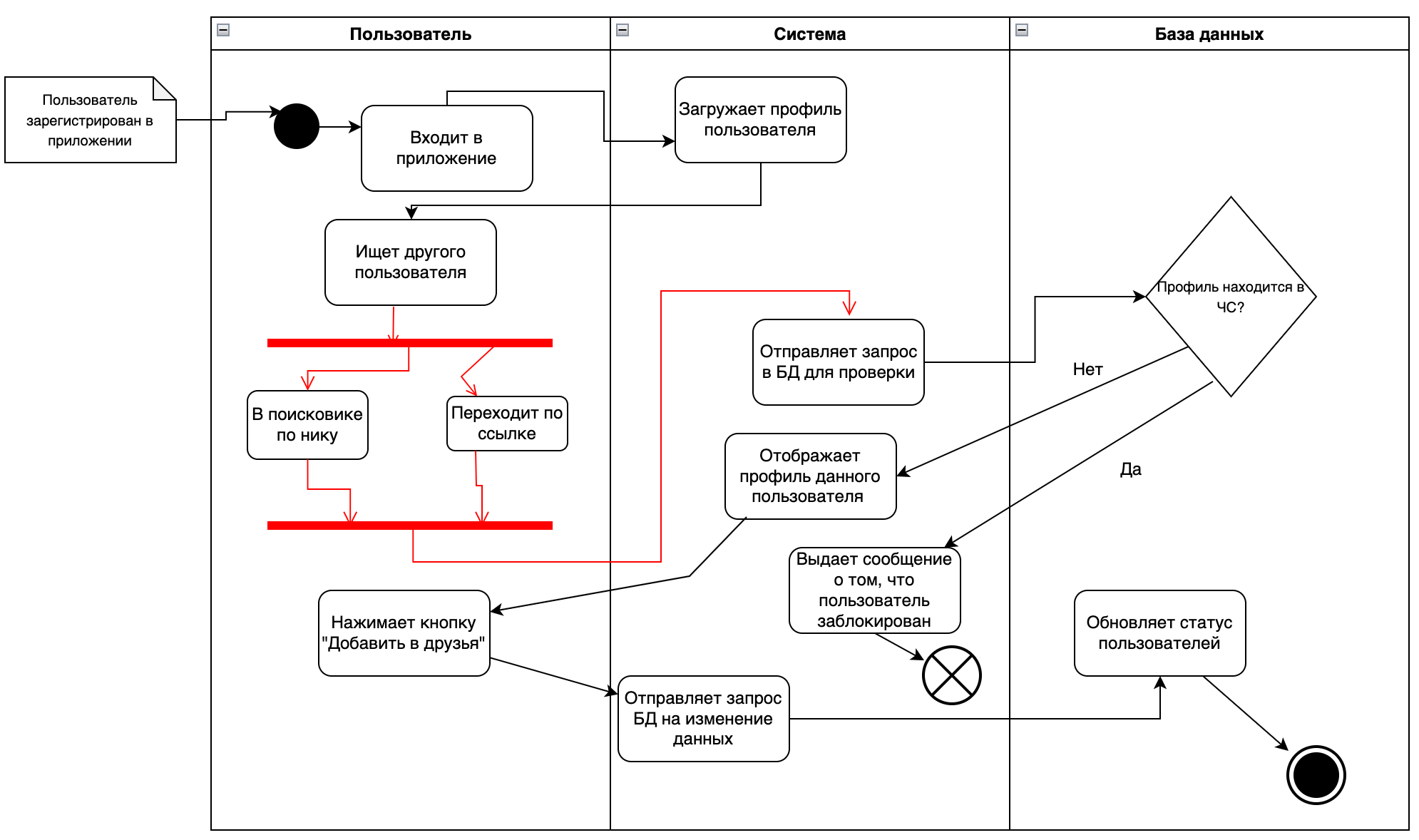
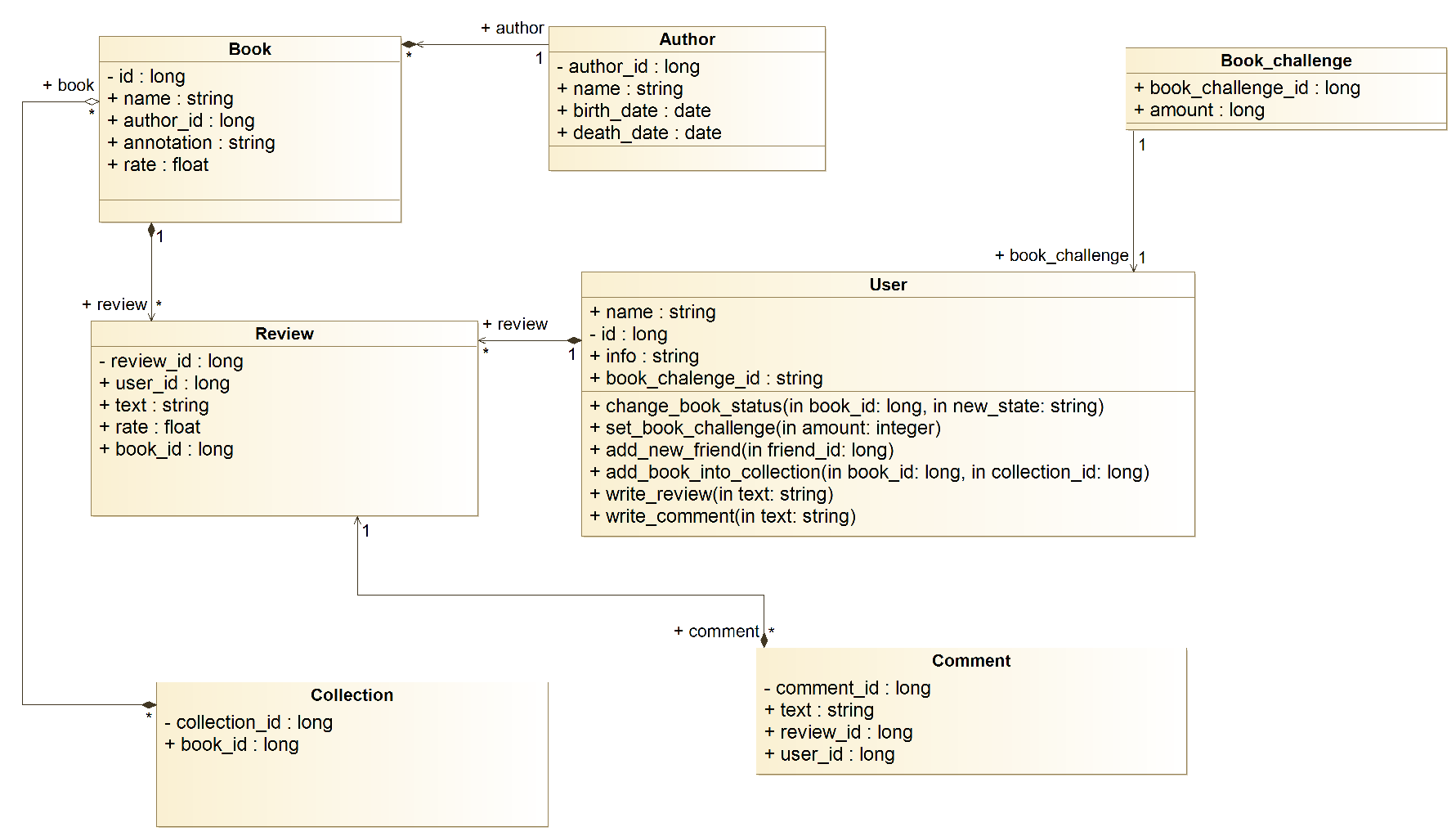
****

Диаграмма деятельности “Добавление в друзья”:

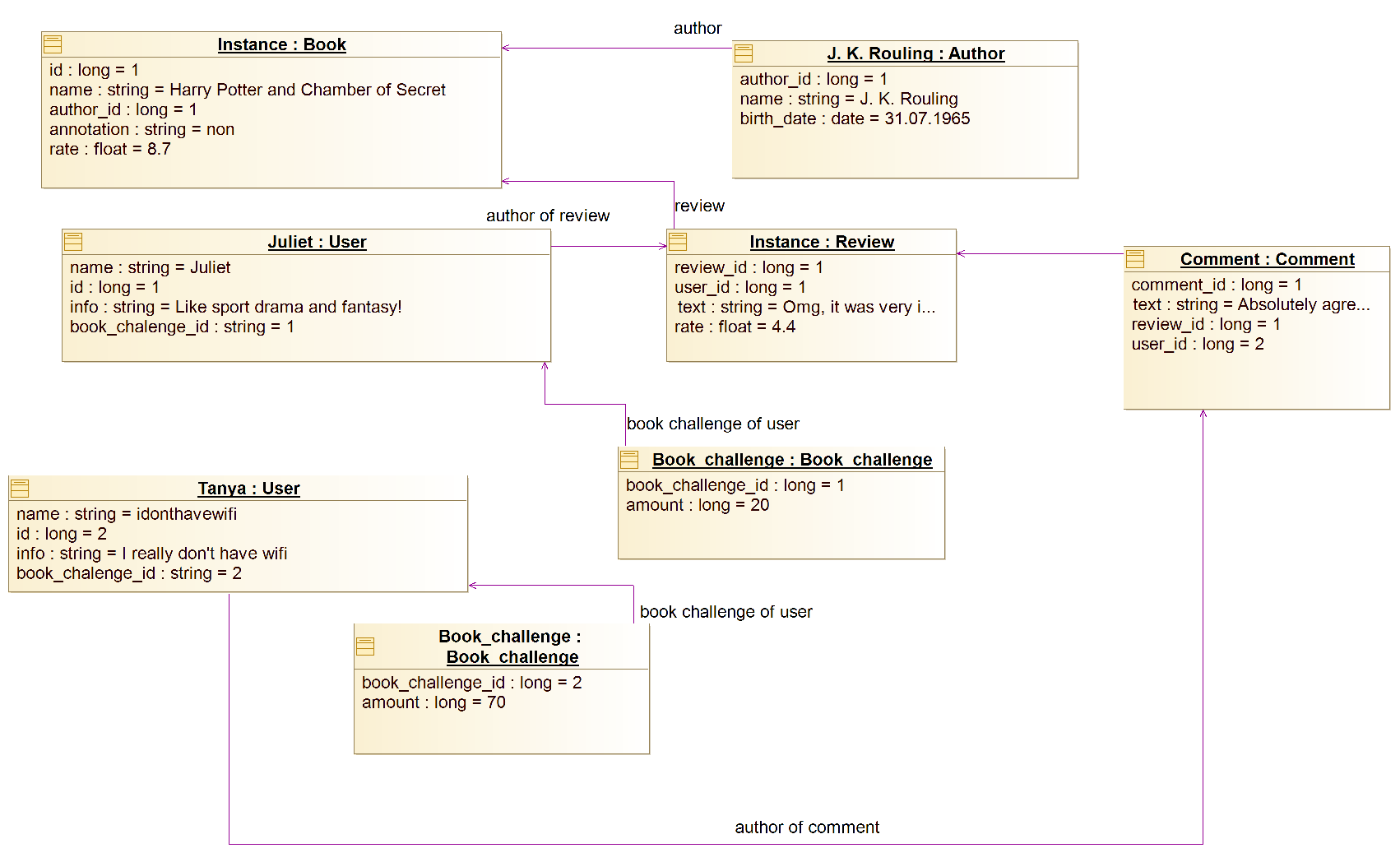


**Диаграммы классов и объектов**

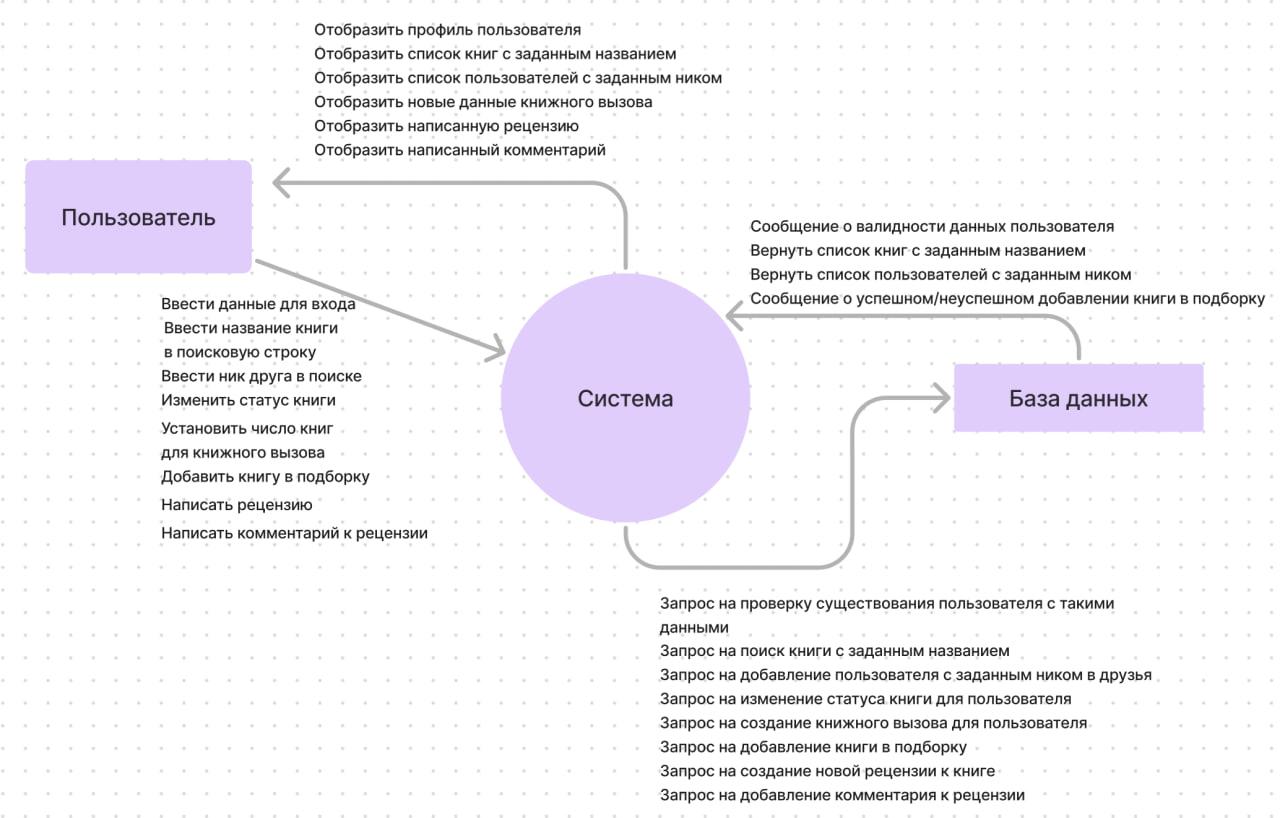
*Диаграмма классов:*

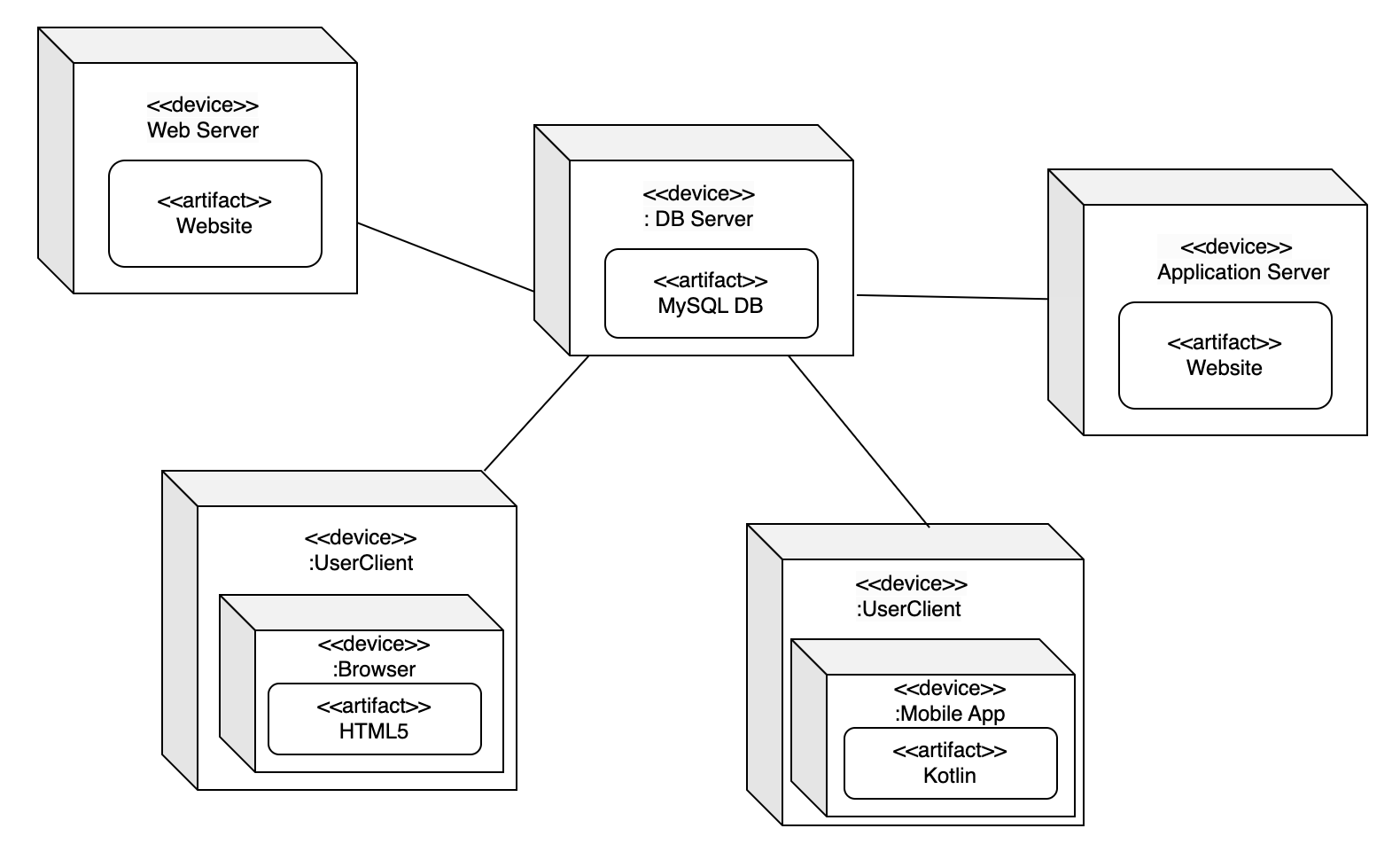
****

*Диаграмма объектов:*

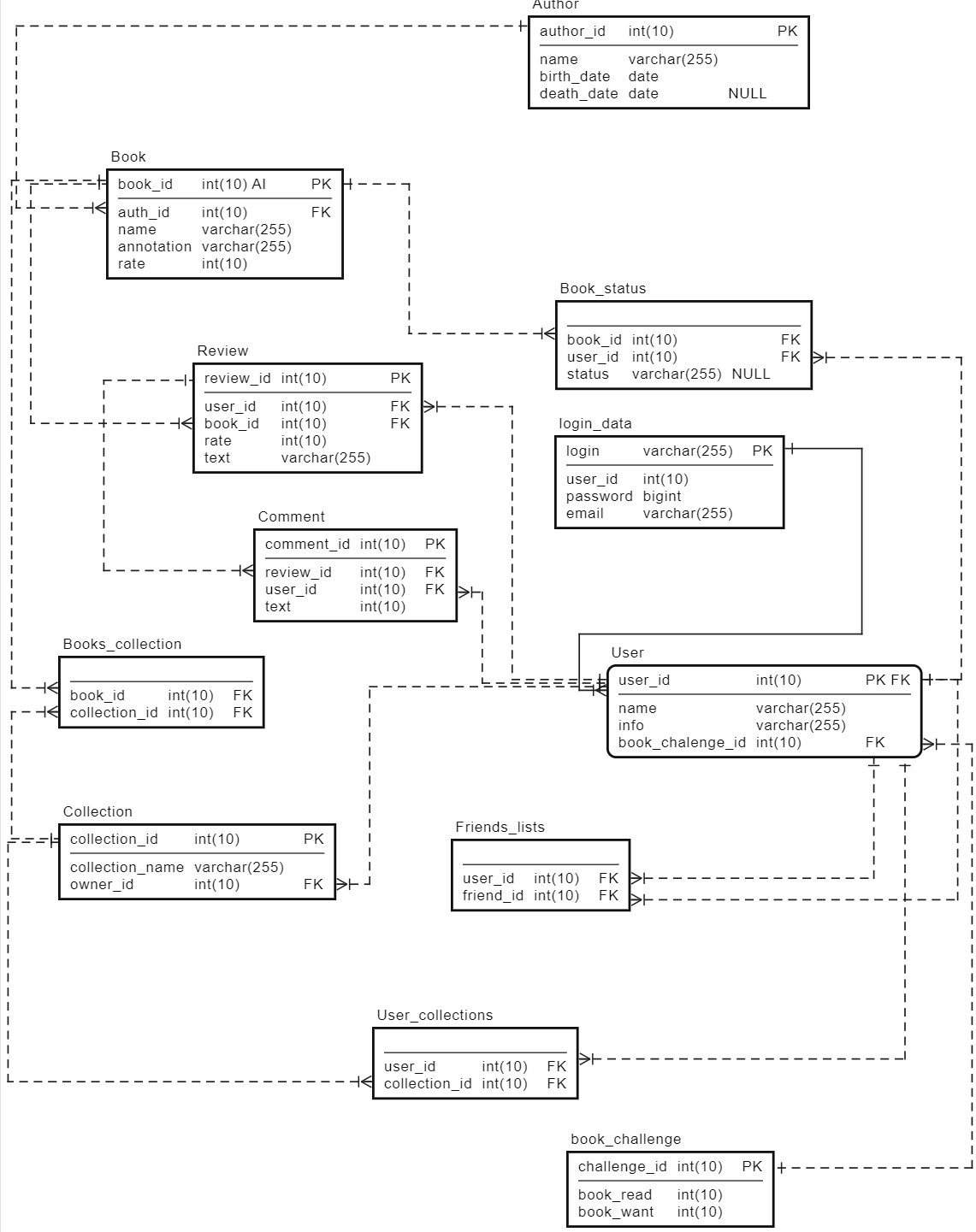
**

**Диаграммы компонентов**

****

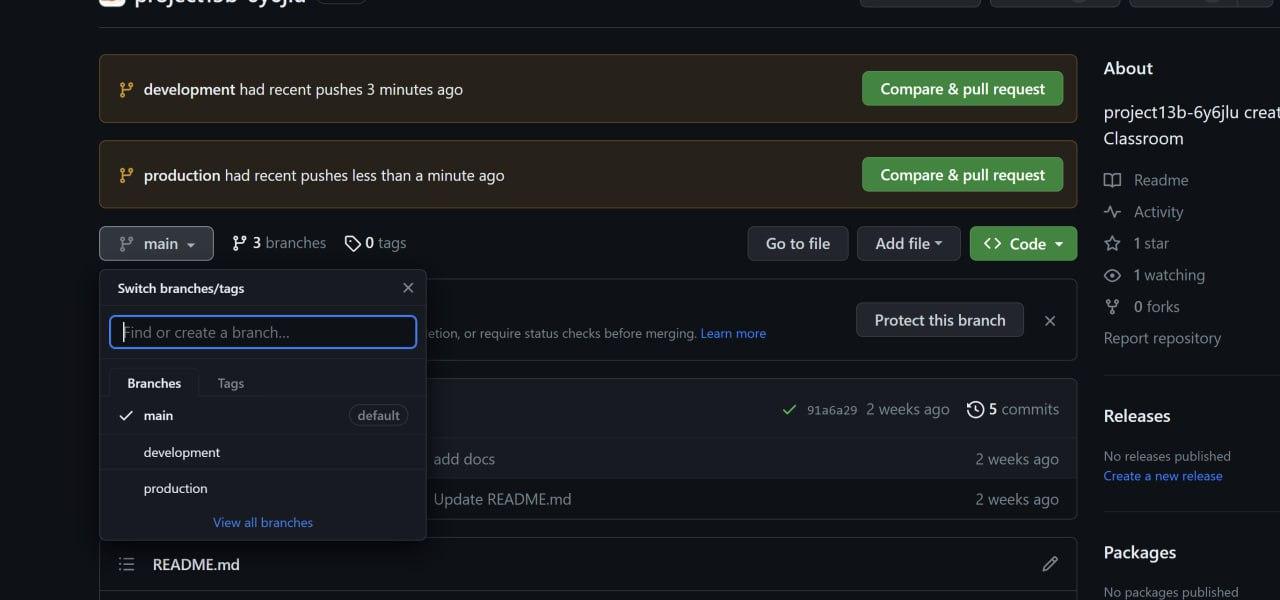
**Диаграммы развертывания**

**Схема базы данных**

****

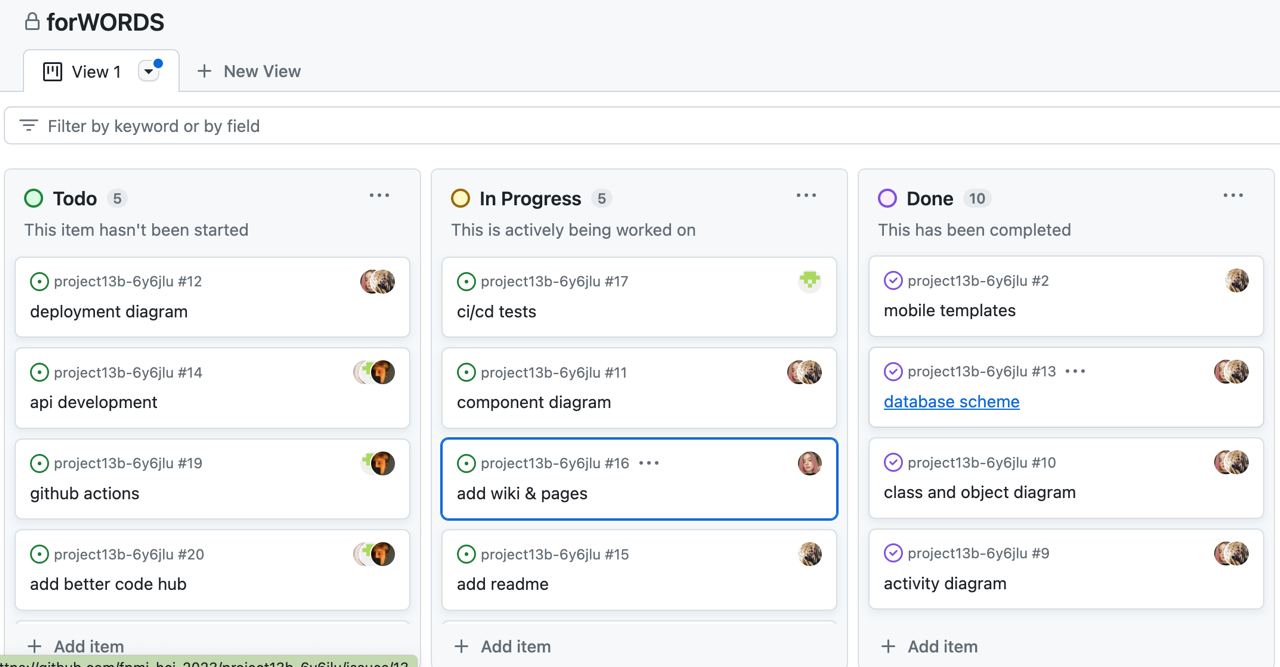
**Создание структуры проекта на github**

Были созданы необходимые ветки для разработки (ветки для релиз-версии (production) и тестовой версии (development)).

****

**Управление проектом с использованием Project в github.**

Задачи распределялись и добавлялись в Projects, были созданы issues, по отслеживанию которых смотрели, что нужно сделать, что находится в процессе, что нужно доработать и что уже сделано. Можно также увидеть, что выполнял каждый участник команды:



**План работы по бекенду**

Для взаимодействия приложения и сервера будет использоваться REST API.

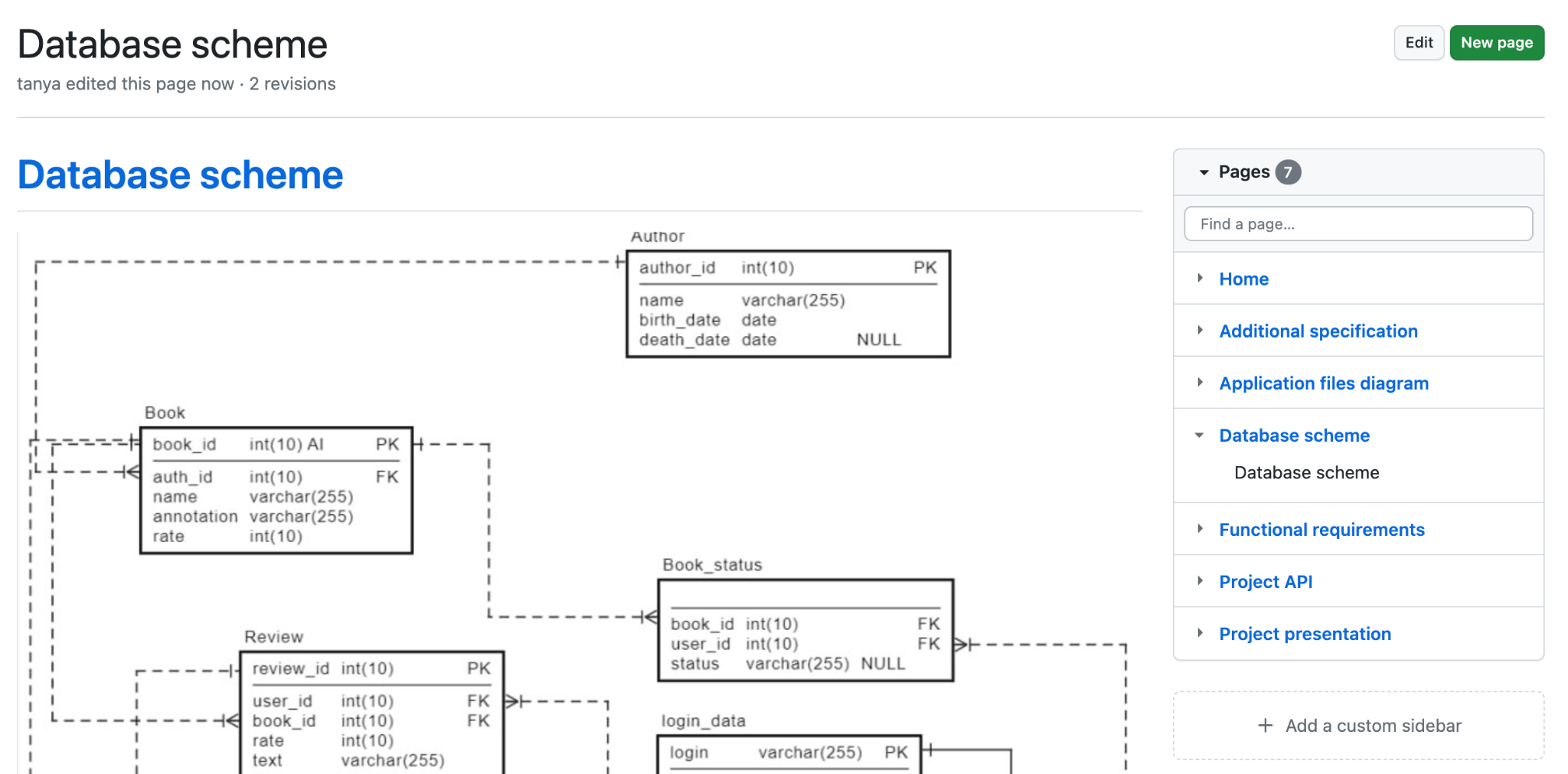
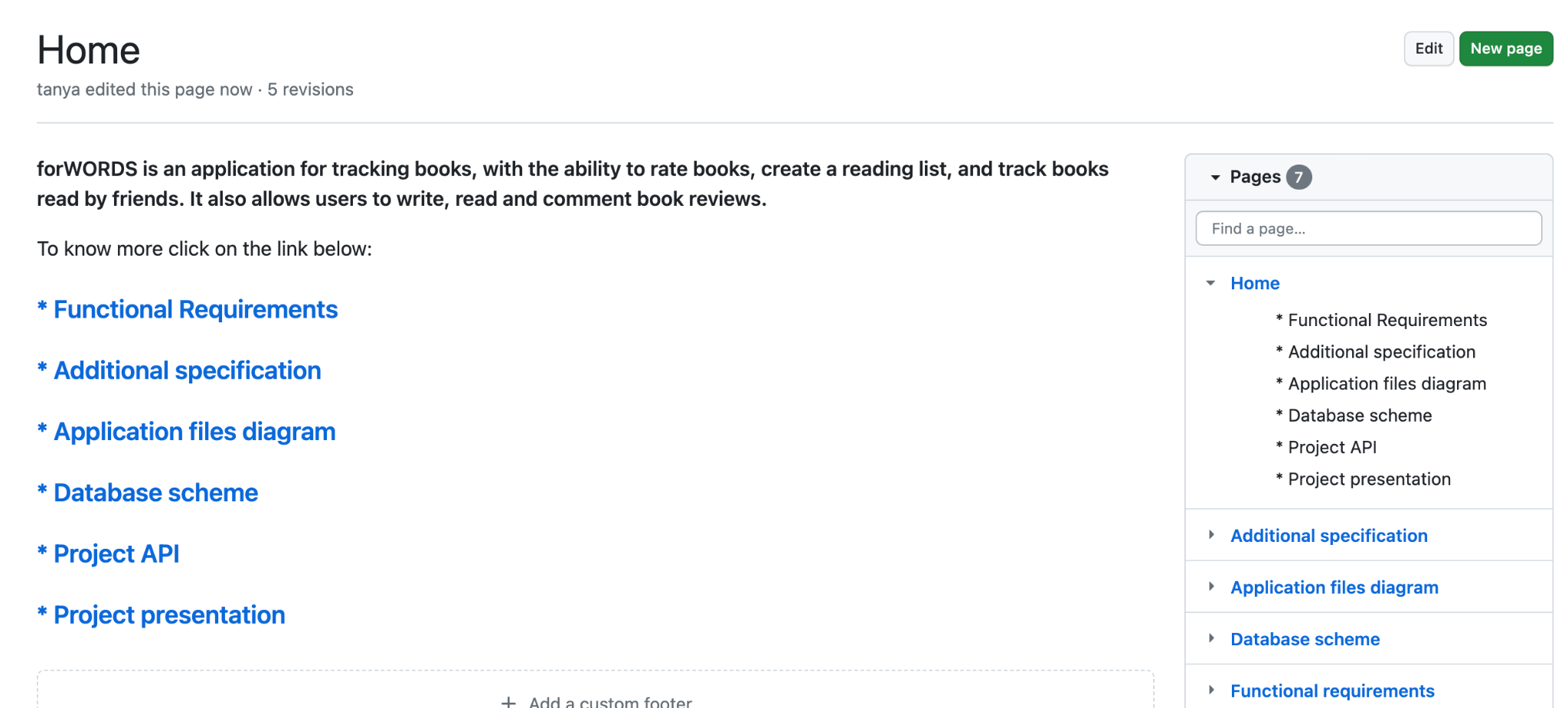
Основные пункты работы по бекенду (необходимый функционал, этапы работы):

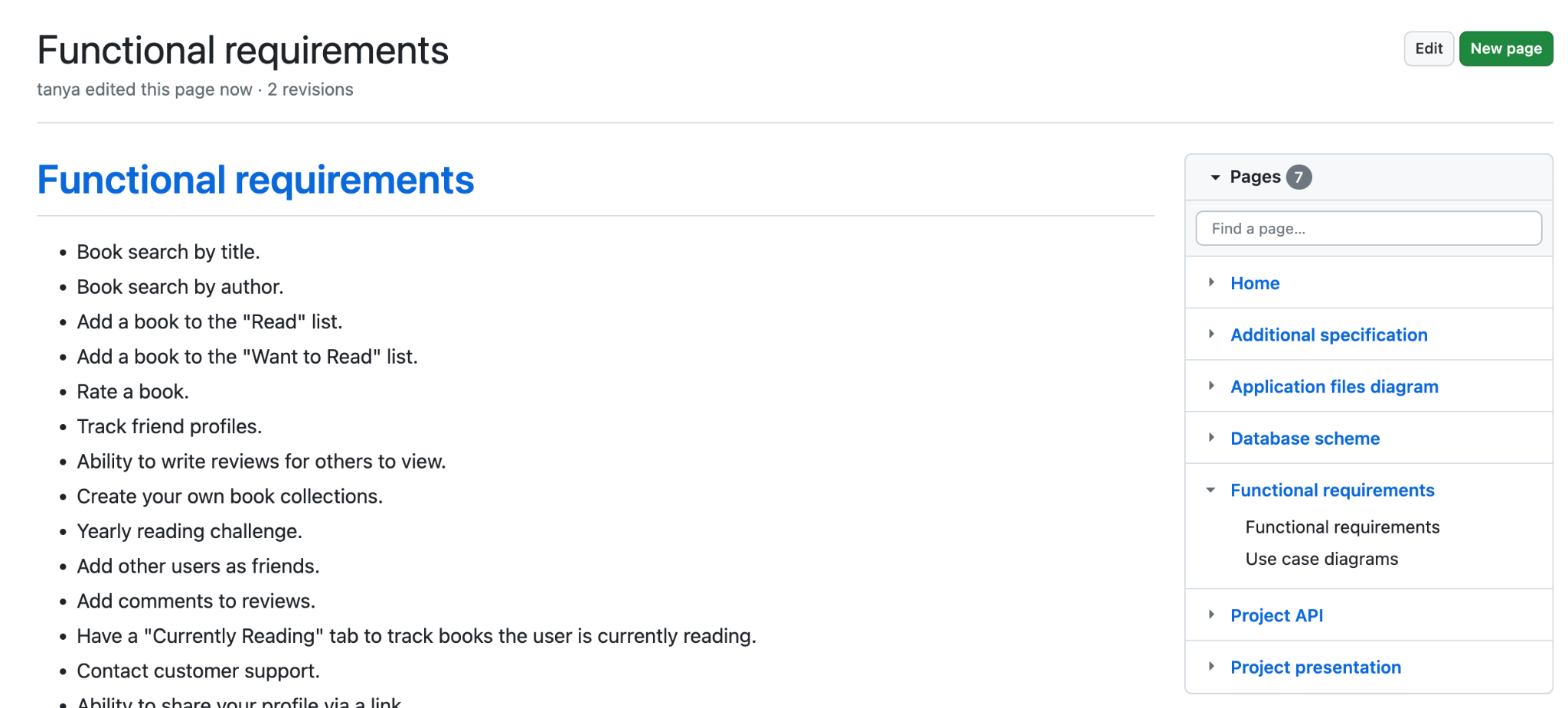
1. Создание базовых сущностей Книг, полей, взаимосвязей.
2. Разработка сущностей комментариев
3. Инициализация бд
4. Добавление данных в бд
5. Разработка API эндпоинтов для:

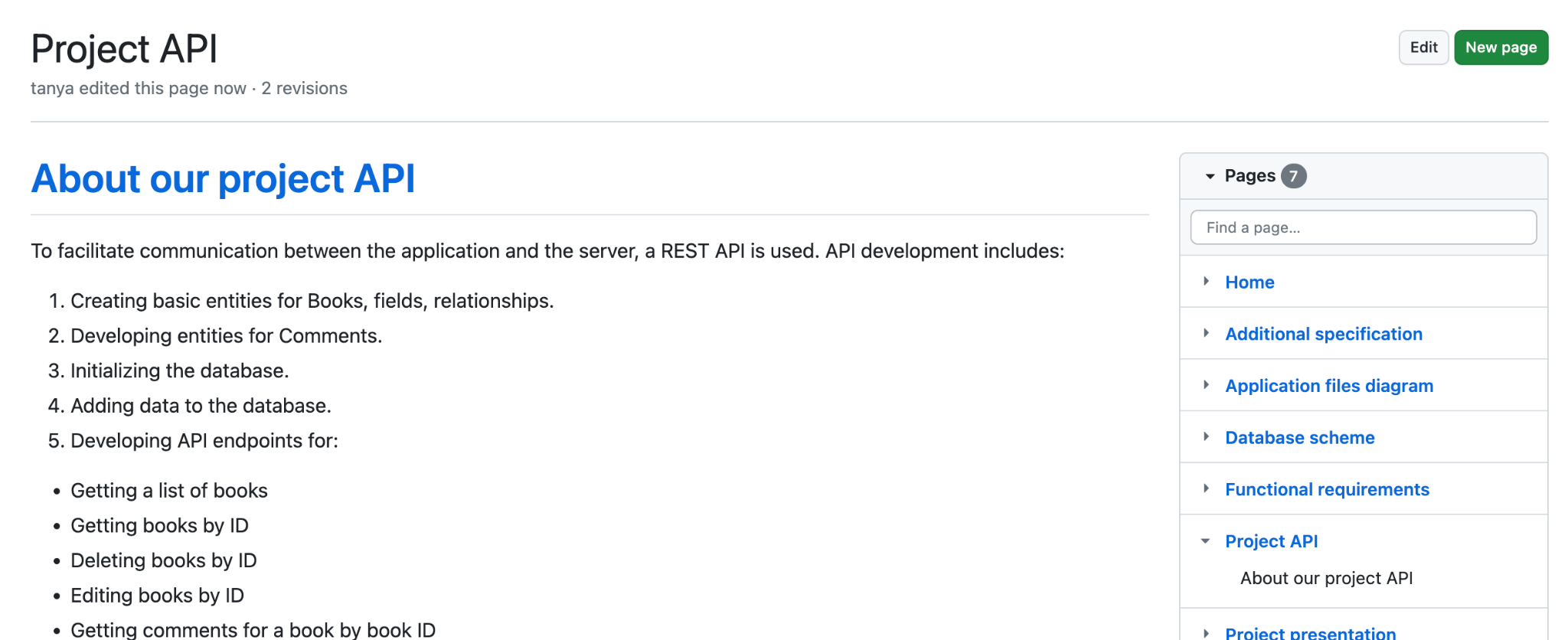
* получения списка книг
* получения книг по id
* удаление книг по id
* редактирование книг по id
* получение комментариев книги по id книги
* добавление комментариев к книге по id книги
* получение данных юзера
* редактирование данных юзера
* список любимых книг пользователя
* добавить/удалить в список любимых книг

**Документация**

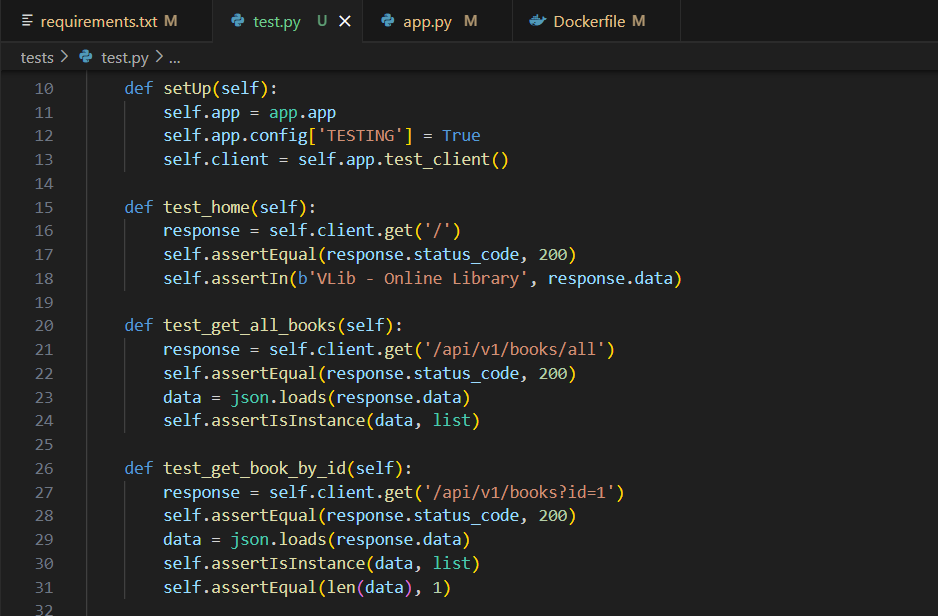
wiki заполняется по ходу разработки, готовые части документации поэтапно добавляются на страницы (некоторые примеры представлены ниже):

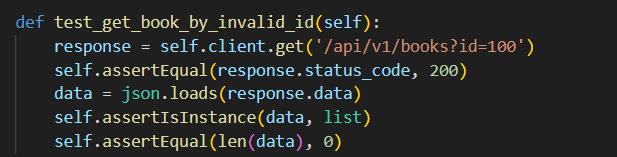
****

****

****

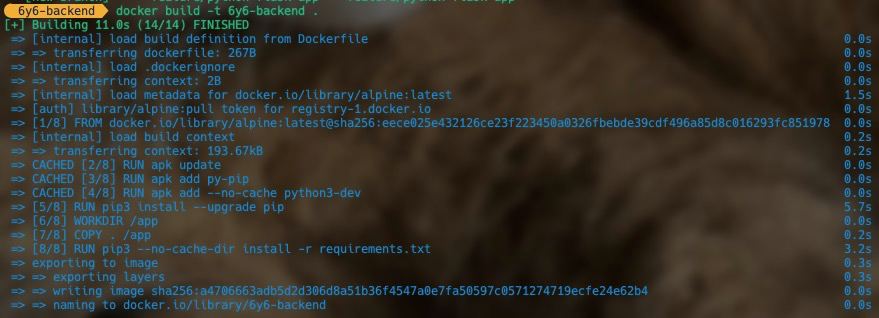
**Автоматические тесты**

****

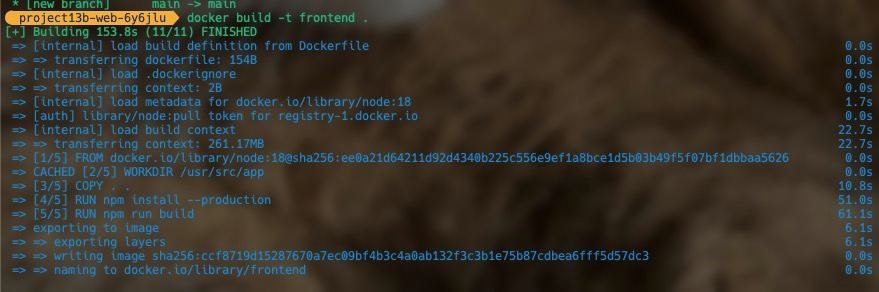
****

**Установка докера и контейнеров**

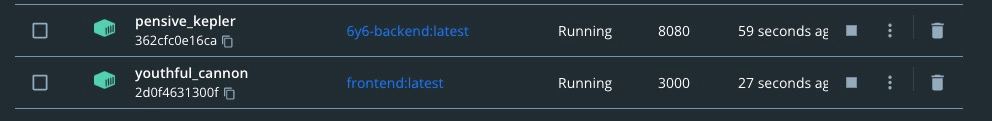
Сборка докер образов для бекенда:

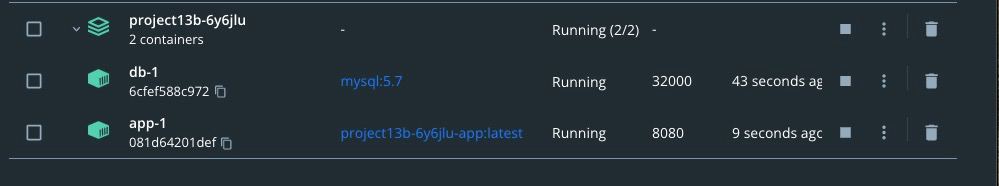
****

Сборка докер образов для фронтенда:

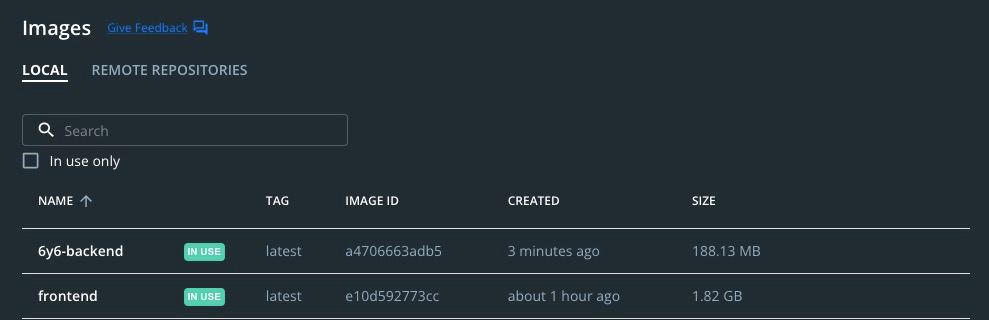


Запустили докер контейнеры:

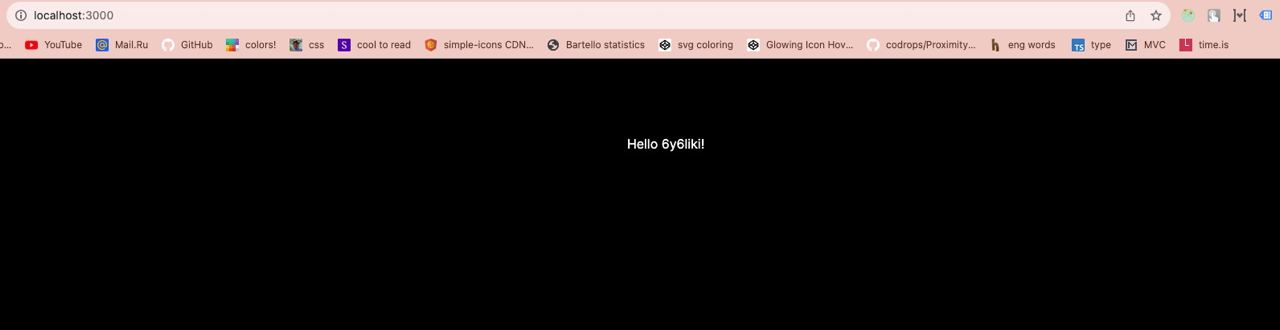
Контейнер с базой данных и бекендом:



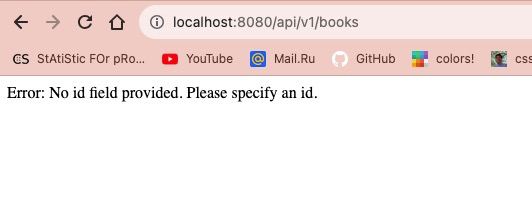
Запустили докер образы:



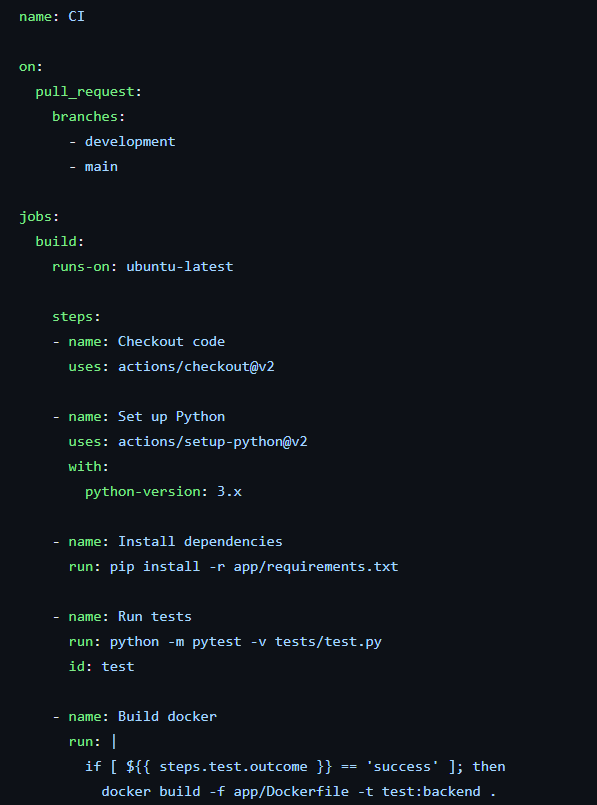
Запущенный фронт приложения из образа докера

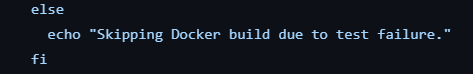


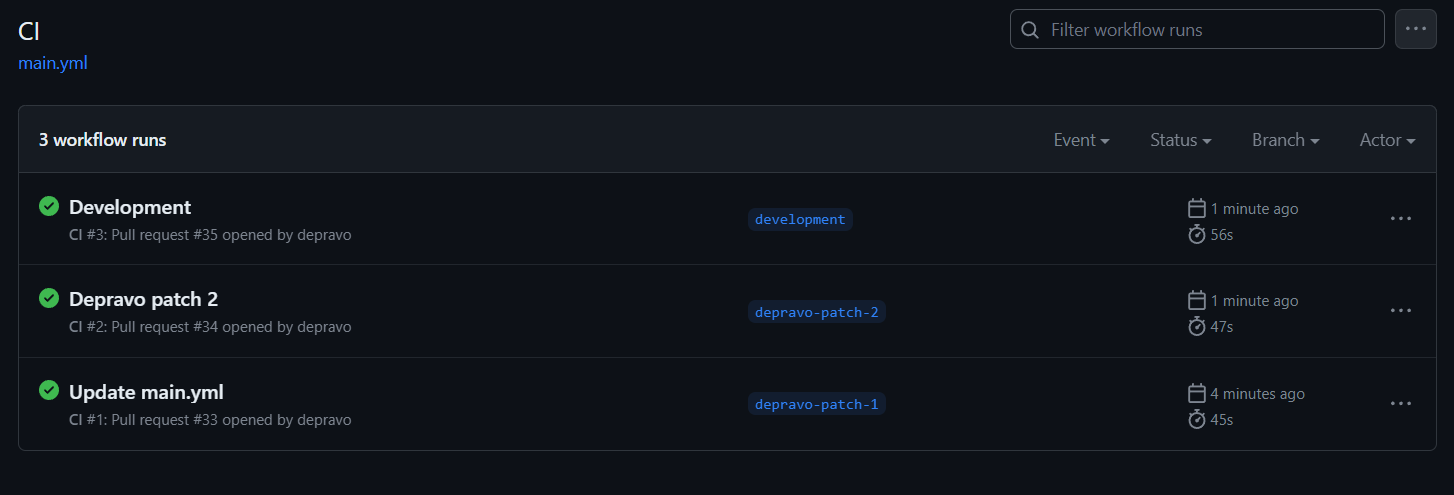
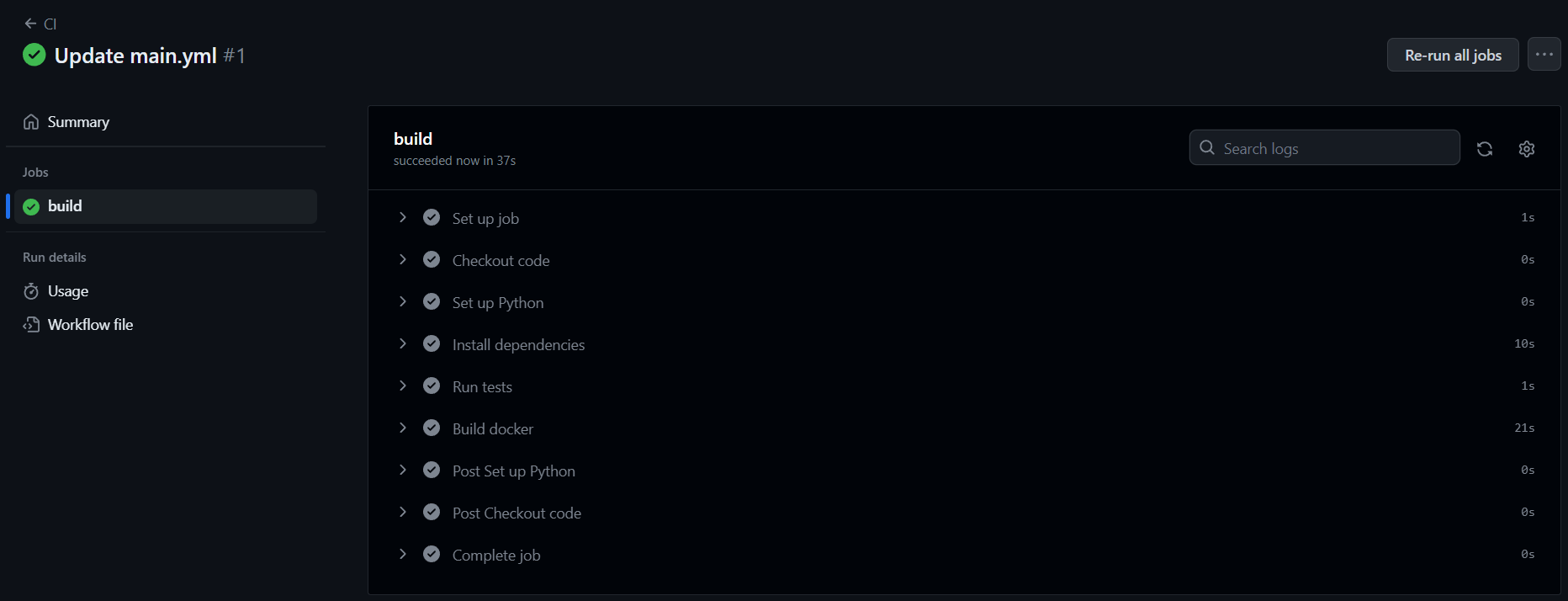
Запущенный фронт приложения из образа докера

****

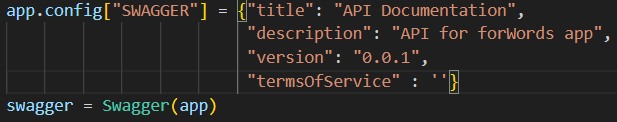
**Сервис непрерывной интеграции (Github Actions)**

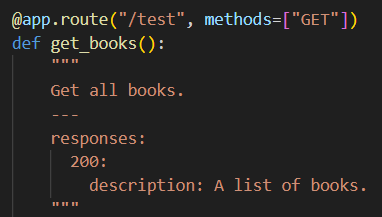
****

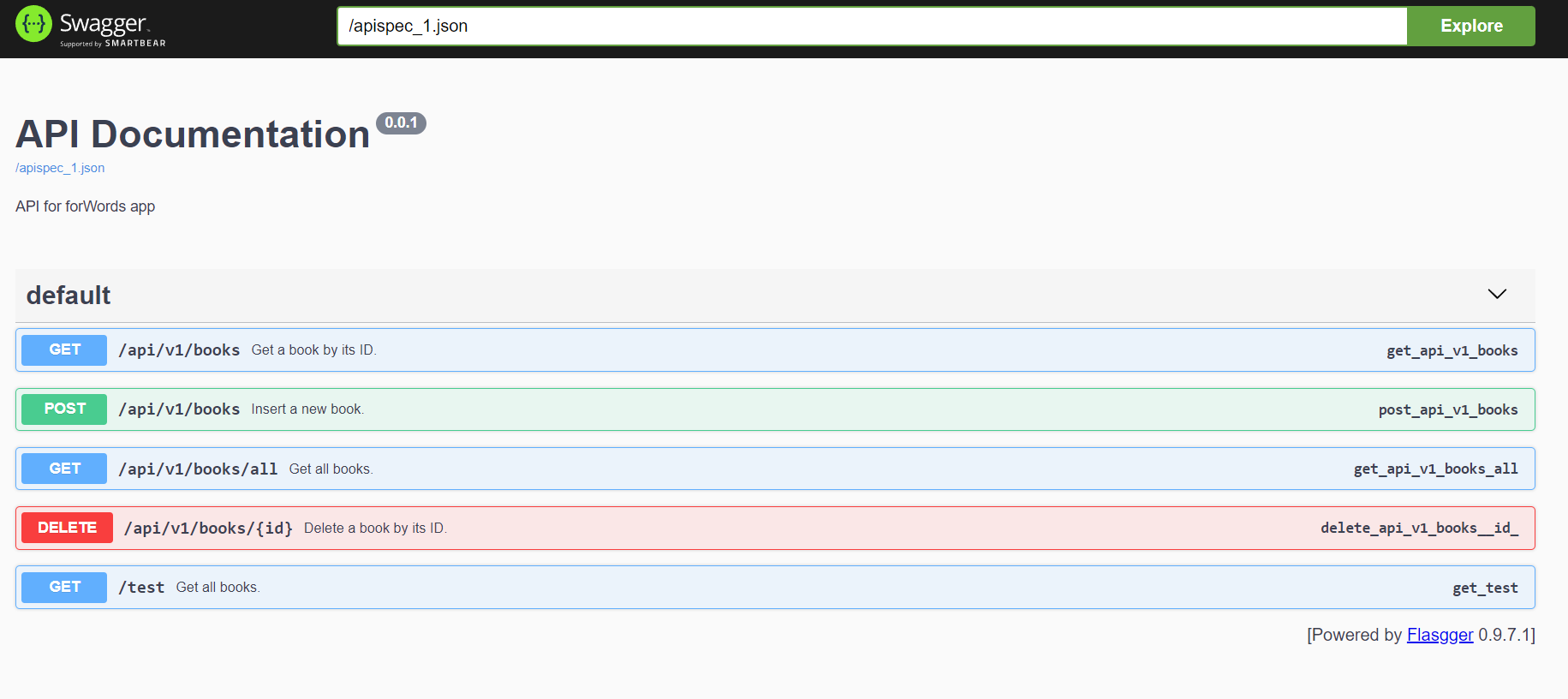
****

****

**Генерация необходимых файлов для Swagger**

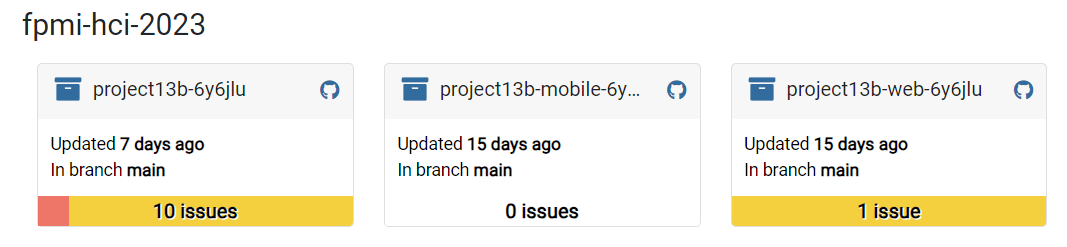
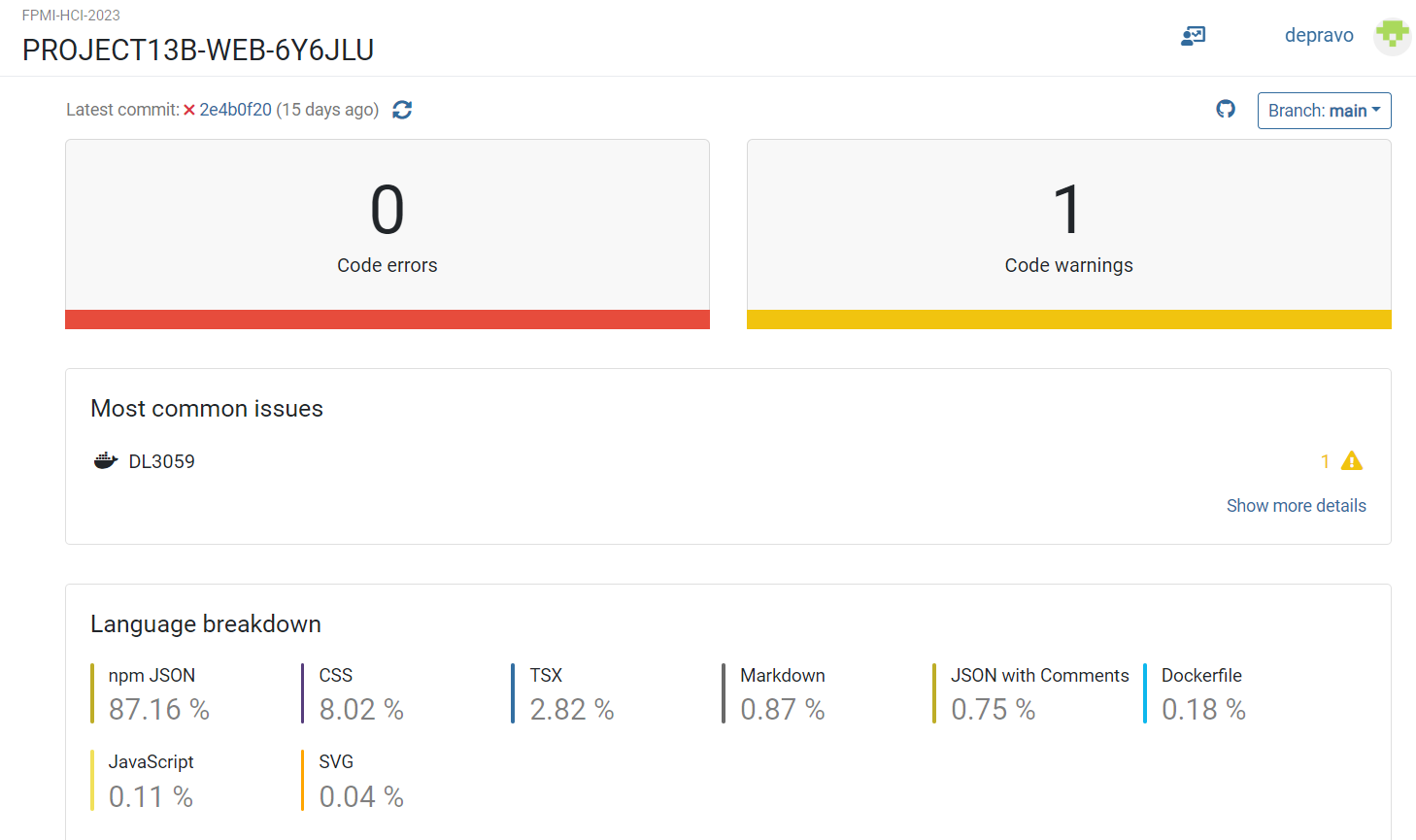
****

****

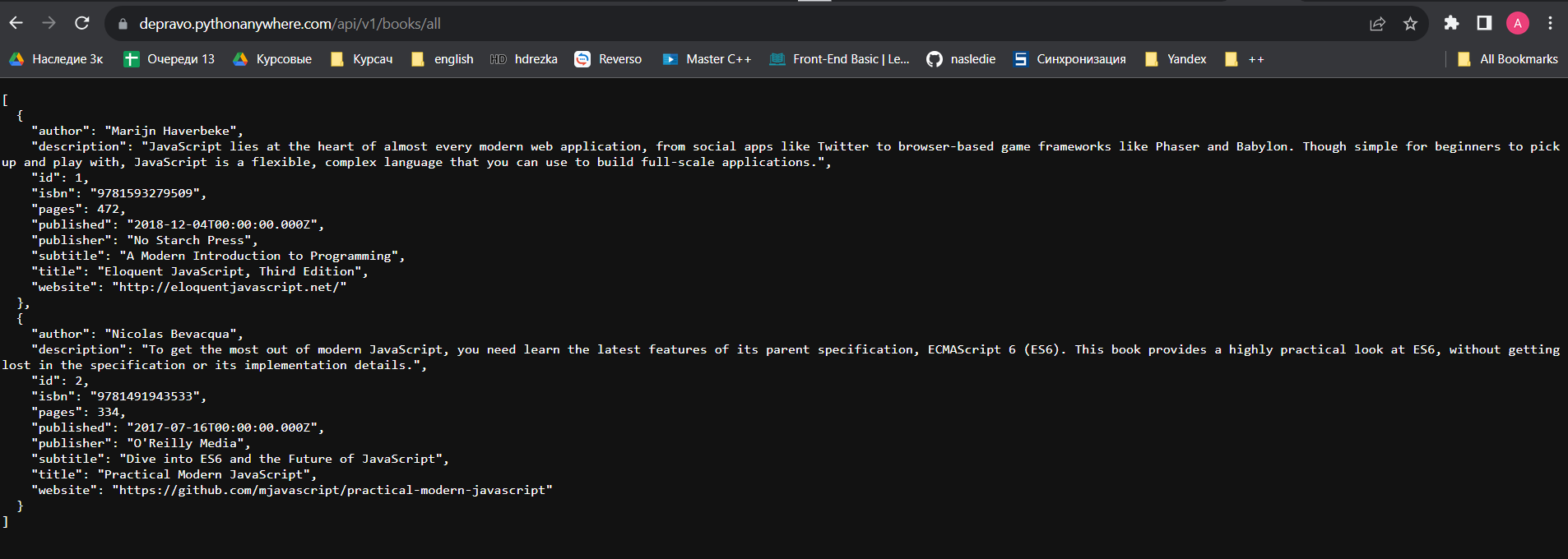
****

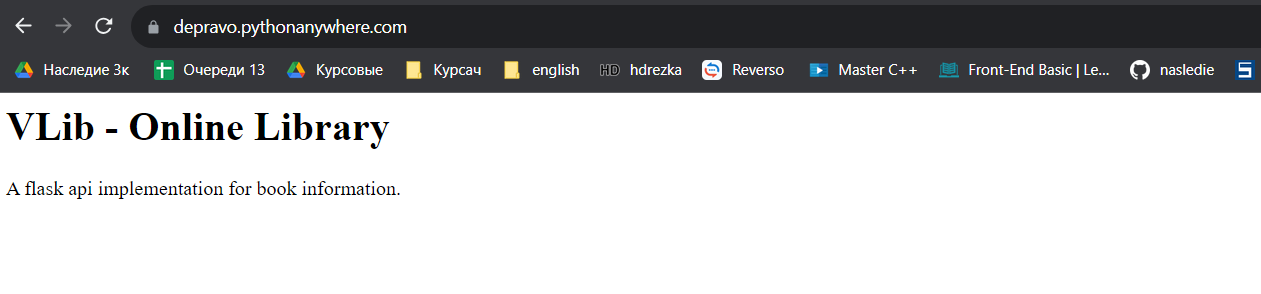
**Настройка анализатора кода**

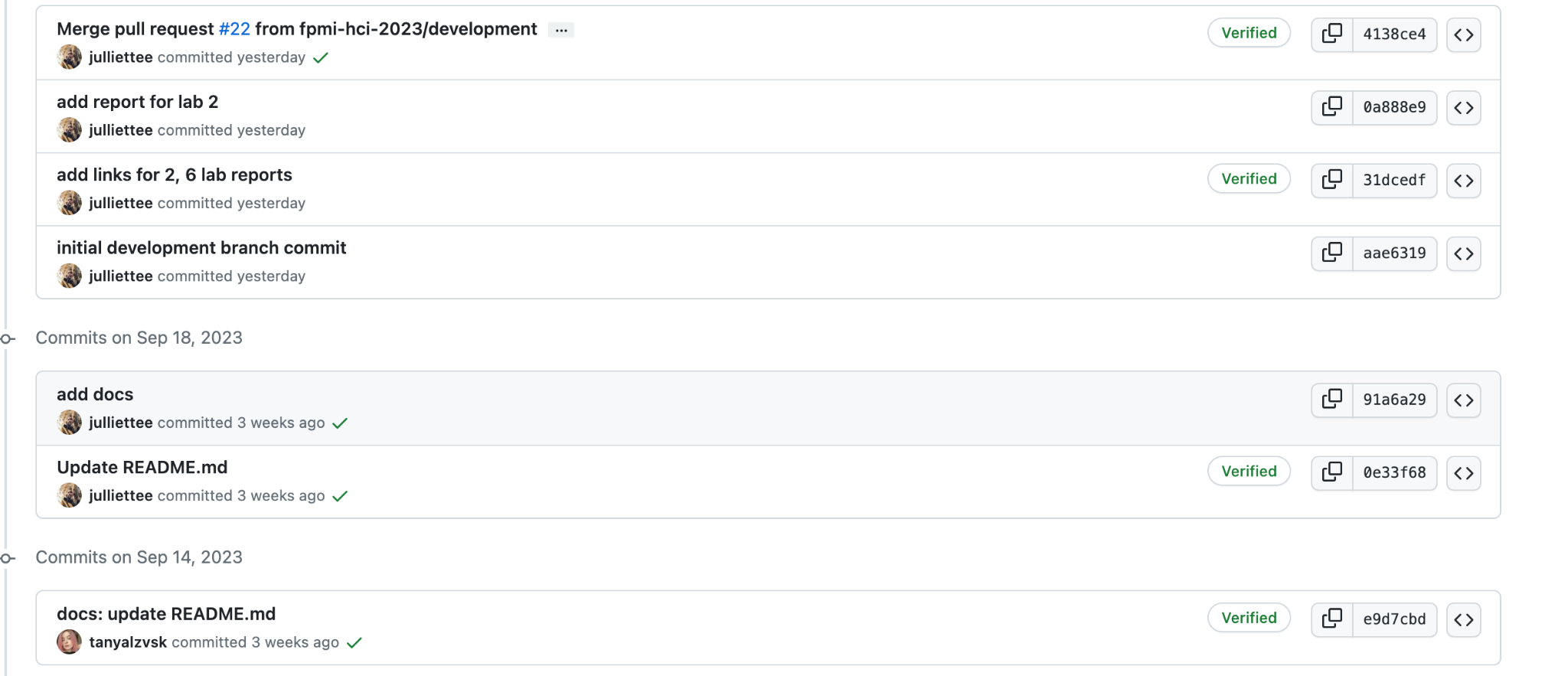
Автоматический анализатор кода Codeac подключен ко всем репозиториям.

**Пример анализа кода для репозитория с веб-приложением:**

**Публикация проекта**

****

****

**Коммиты**

**Выводы**

Разработано техническое задание для внутреннего использования, определяющее стратегию дизайна и требования к приложению "forWORDS".

Созданы различные диаграммы, включая диаграммы бизнес-процессов, диаграммы вариантов использования, диаграммы деятельности, диаграммы классов и объектов, диаграммы компонентов и диаграммы развертывания.

Разработана схема базы данных с использованием диаграммы "сущность-связь" (ERD) и диаграммы объектно-реляционного отображения.

Создана структура проекта на GitHub и использованы GitHub Projects для управления проектом.

Составлен план работы по разработке бекенд-части приложения "книжный трекер", включая разработку API, добавление автоматических тестов и использование Docker для упрощения разработки.

Встроена генерация файлов для Swagger в процесс сборки проекта и проект успешно опубликован на Google Cloud.

Настроена интеграция с Better Code Hub для контроля качества кода.