|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |

ТЕХНИЧЕСКИЙ ПРОЕКТ

Веб-приложения «Букшеринг»

2024

Содержание

[1 Уточнение структур данных 2](#_Toc182795763)

[2 Формы и их пердставления 7](#_Toc182795764)

[3 Разработка алгоритма решения задачи 9](#_Toc182795765)

[4 Определения языка 20](#_Toc182795766)

[5 Структура программы и требования к техническим средствам 21](#_Toc182795767)

# Уточнение структур данных

У веб-приложения «Букшеринг» должна быть база данных. База данных для веб-приложения «Букшеринг» изображена в логической ER диаграмме и физической ER диаграмме они в свою очередь изображены на рисунках 1 и 2.

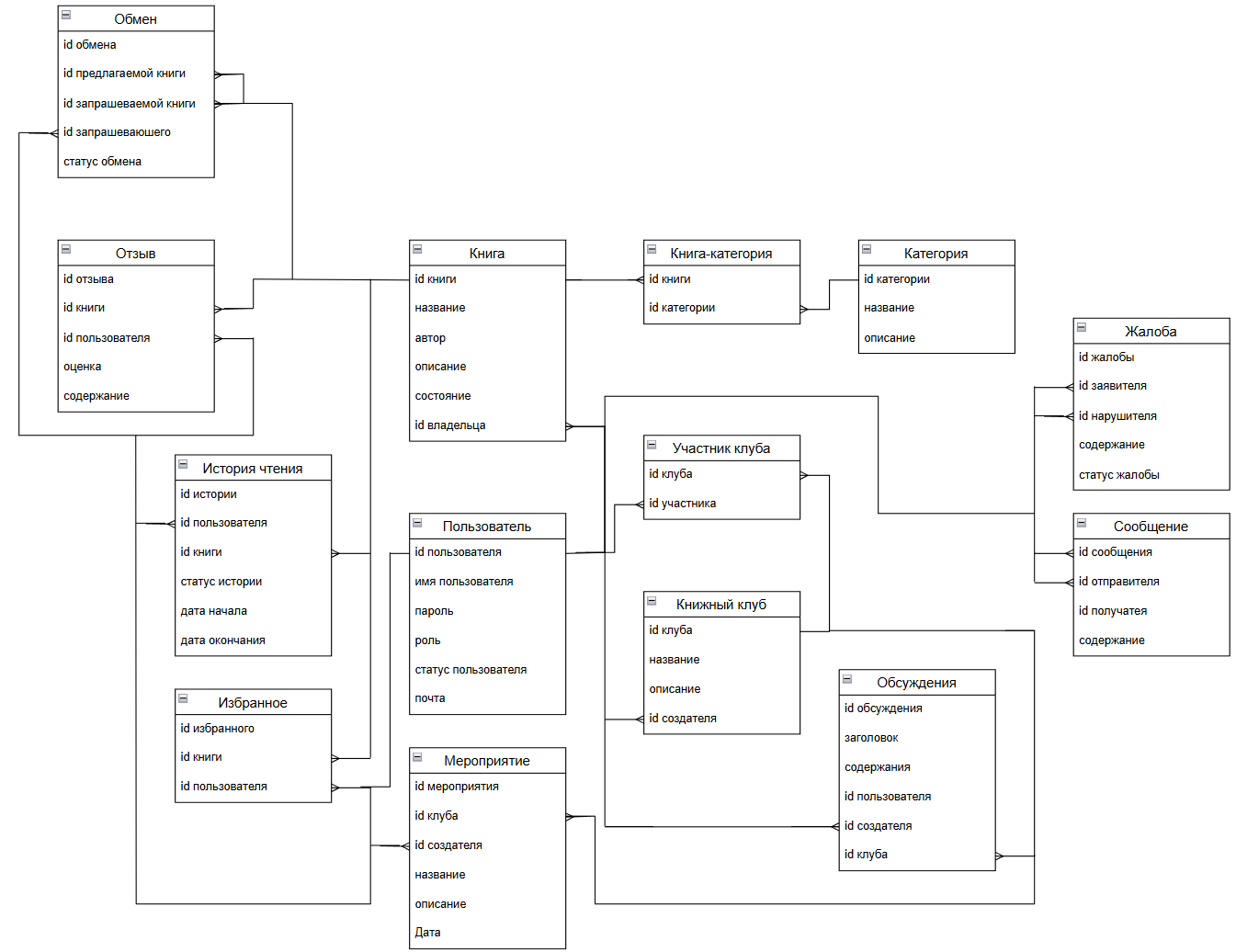


Рисунок 1 – Логическая ER диаграмма

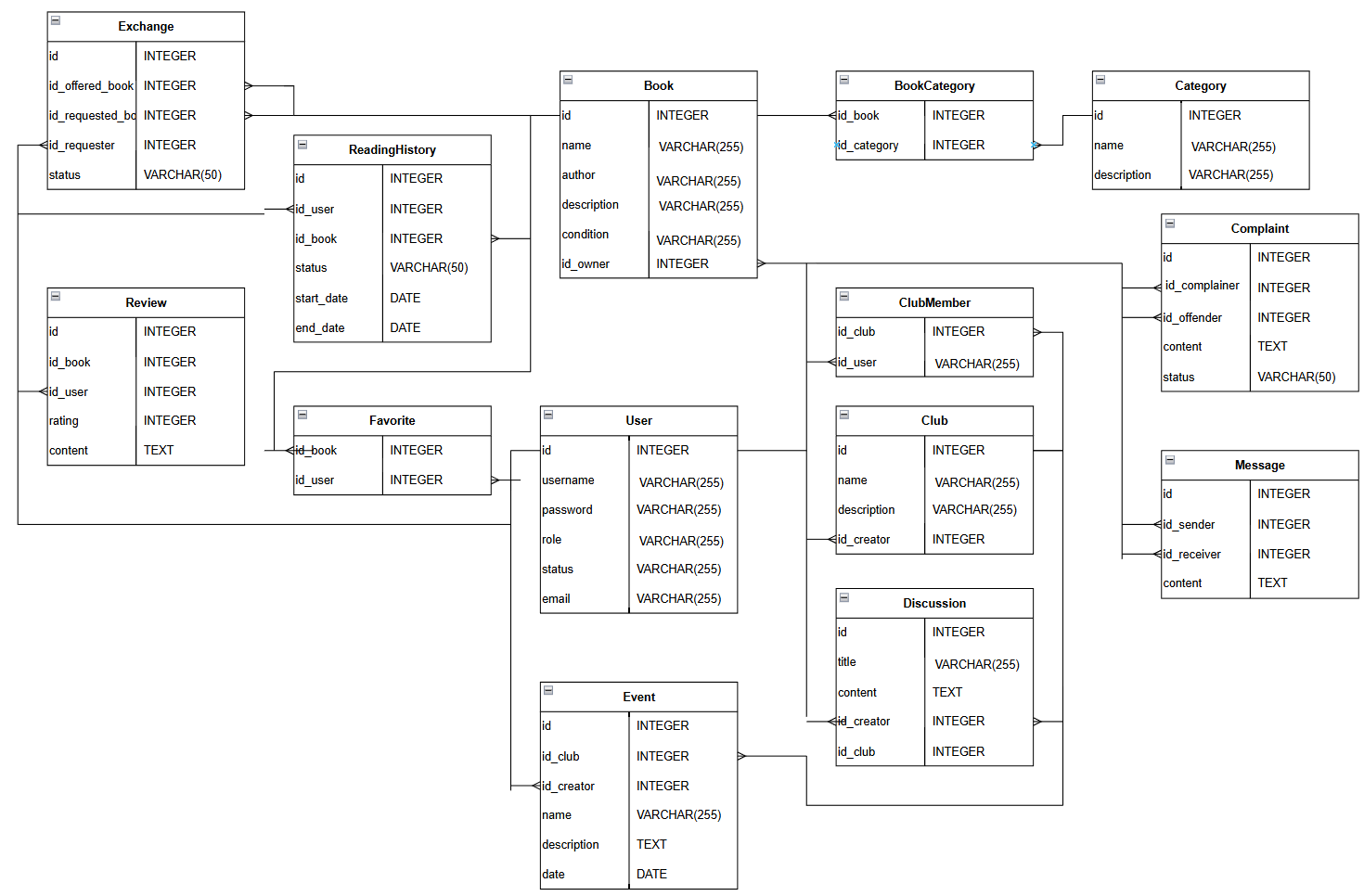


Рисунок 2 – Физическая ER диаграмма

На ER диаграммах показано, что есть 14 таблиц: «Обмен», «Отзыв», «Книга», «Книга-категория», «Категория», «Жалоба», «История чтения», «Пользователь», «Участник клуба», «Сообщение», «Избранное», «Мероприятие», «Книжный клуб», и «Обсуждения».

Таблица «Книги»

* id книги: Уникальный идентификатор книги.
* Название книги: Название книги.
* Автор: Автор книги.
* Описание: Краткое описание книги.
* Состояние: Состояние книги (например, новая, хорошая, повреждённая).
* id владельца: Ссылка на пользователя, которому принадлежит книга (внешний ключ к таблице «Пользователи»).

Таблица «Пользователи»

* id пользователя: Уникальный идентификатор пользователя.
* Имя пользователя: Логин или псевдоним пользователя.
* Пароль: Хэшированный пароль пользователя.
* Роль: Роль пользователя (например, администратор, пользователь).
* Статус пользователя: Статус учетной записи пользователя (например, активный, заблокированный).
* Почта: Электронная почта пользователя.

Таблица «Категории»

* id категории: Уникальный идентификатор категории.
* Название категории: Название категории (например, «Фантастика», «Детективы»).
* Описание категории: Краткое описание категории.

Таблица «Книга-Категория»

(Связь между книгами и категориями, представляющая связь "многие ко многим")

* id книги: Ссылка на таблицу «Книги», указывающая на книгу.
* id категории: Ссылка на таблицу «Категории», указывающая на категорию.

Таблица «Книжные клубы»

* id клуба: Уникальный идентификатор книжного клуба.
* Название клуба: Название книжного клуба.
* Описание клуба: Краткое описание клуба.
* id создателя: Ссылка на пользователя, создавшего клуб (внешний ключ к таблице «Пользователи»).

Таблица «Участники клуба»

* id клуба: Ссылка на таблицу «Книжные клубы», указывающая на клуб.
* id участника: Ссылка на таблицу «Пользователи», указывающая на участника клуба.

Таблица «Обсуждения»

* id обсуждения: Уникальный идентификатор обсуждения.
* Заголовок обсуждения: Заголовок темы обсуждения.
* Содержание обсуждения: Текстовое содержание обсуждения.
* id создателя: Ссылка на пользователя, который создал обсуждение (внешний ключ к таблице «Пользователи»).
* id клуба: Ссылка на таблицу «Книжные клубы», указывающая на клуб, в рамках которого создано обсуждение.

Таблица «Жалобы»

* id жалобы: Уникальный идентификатор жалобы.
* id заявителя: Ссылка на таблицу «Пользователи», указывающая на пользователя, подавшего жалобу.
* id нарушителя: Ссылка на таблицу «Пользователи», указывающая на пользователя, на которого подана жалоба.
* Содержание жалобы: Текстовое описание проблемы или жалобы.
* Статус жалобы: Текущий статус жалобы (например, в рассмотрении, решена, отклонена).

Таблица «Отзывы»

* id отзыва: Уникальный идентификатор отзыва.
* id книги: Ссылка на таблицу «Книги», указывающая на книгу, на которую оставлен отзыв.
* id пользователя: Ссылка на таблицу «Пользователи», указывающая на автора отзыва.
* Оценка: Числовая оценка книги (например, от 1 до 5).
* Содержание отзыва: Текстовое содержание отзыва.

Таблица «Сообщения»

* id сообщения: Уникальный идентификатор сообщения.
* id отправителя: Ссылка на таблицу «Пользователи», указывающая на отправителя сообщения.
* id получателя: Ссылка на таблицу «Пользователи», указывающая на получателя сообщения.
* Содержание сообщения: Текст сообщения.

Таблица «Избранное»

* id избранного: Уникальный идентификатор записи в списке избранного.
* id книги: Ссылка на таблицу «Книги», указывающая на книгу, добавленную в избранное.
* id пользователя: Ссылка на таблицу «Пользователи», указывающая на владельца избранного.

Таблица «История чтения»

* id истории: Уникальный идентификатор записи истории чтения.
* id пользователя: Ссылка на таблицу «Пользователи», указывающая на пользователя, который читал книгу.
* id книги: Ссылка на таблицу «Книги», указывающая на книгу, которую читал пользователь.
* Статус истории: Статус чтения (например, "читаю", "завершено").
* Дата начала: Дата начала чтения книги.
* Дата окончания: Дата завершения чтения книги.

Таблица «Обмен»

* id обмена: Уникальный идентификатор обмена книгами.
* id предлагаемой книги: Ссылка на таблицу «Книги», указывающая на книгу, предложенную для обмена.
* id запрашиваемой книги: Ссылка на таблицу «Книги», указывающая на книгу, которую хочет получить пользователь.
* id запрашивающего: Ссылка на таблицу «Пользователи», указывающая на пользователя, инициировавшего обмен.
* Статус обмена: Текущий статус обмена (например, "ожидание", "принят", "отклонён").

Таблица «Мероприятия»

* id мероприятия: Уникальный идентификатор мероприятия.
* id клуба: Ссылка на таблицу «Книжные клубы», указывающая на клуб, организовавший мероприятие.
* id создателя: Ссылка на таблицу «Пользователи», указывающая на создателя мероприятия.
* Название мероприятия: Название мероприятия.
* Описание мероприятия: Краткое описание мероприятия.
* Дата: Дата проведения мероприятия.

# Формы и их пердставления

Прототип главной страницы изображена на рисунках 3 и 4.

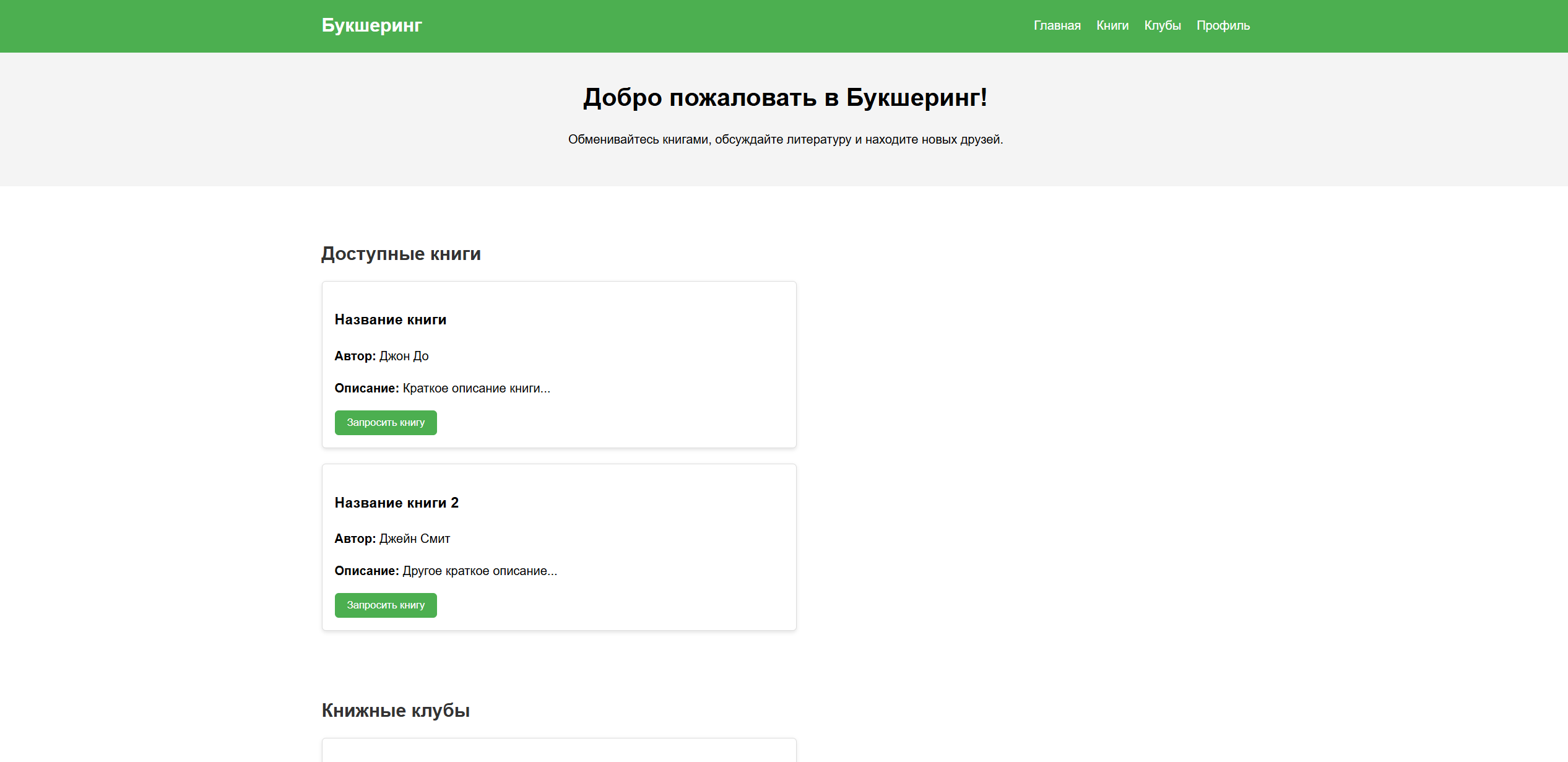


Рисунок 3 – Прототип главной страницы

На этом прототипе показано, что пользователь может запросить книги и кнопками сверху перемещаться по главной странице на нужные ему объекты веб-приложения.

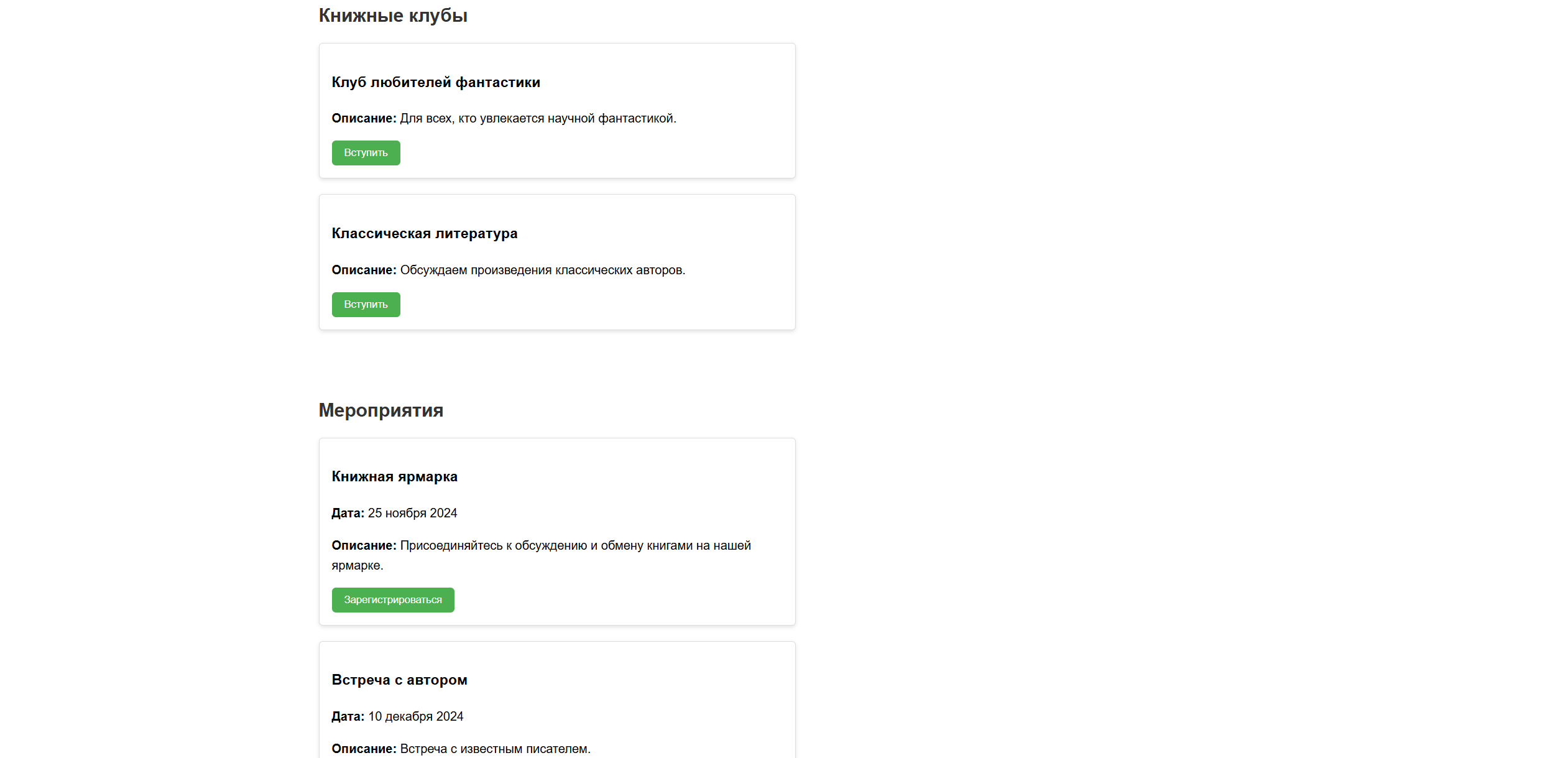


Рисунок 4 – Прототип главной страницы

На этом прототипе показано, что пользователь может вступить в клубы и зарегистрироваться на мероприятия 5.

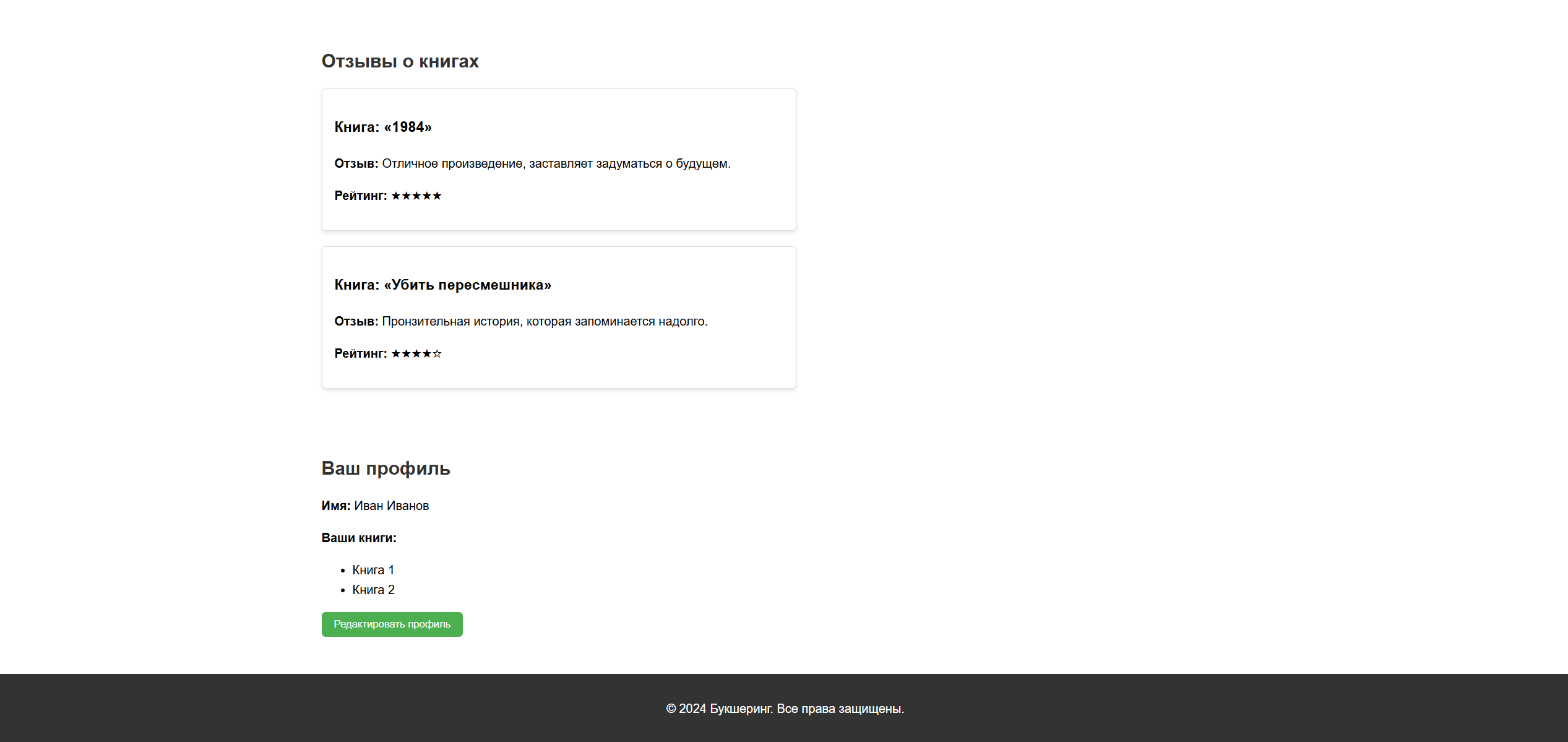


Рисунок 5 – Прототип главной страницы

На это прототипе показано, что пользователь может увидеть популярные отзывы о книгах и перейти на страницу редактирования профиля 6.

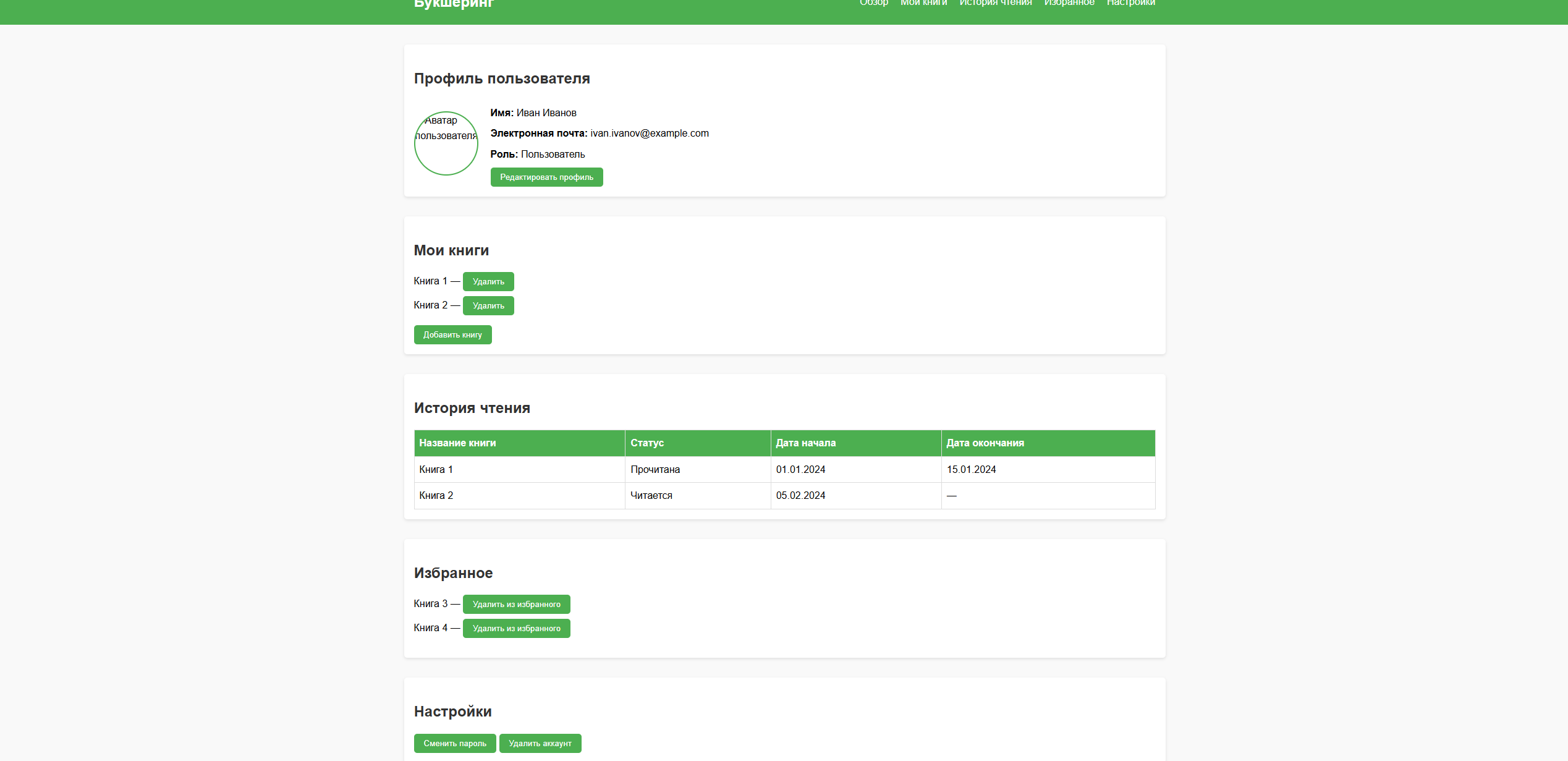


Рисунок 6 – Прототип профиля

На этом прототипе показано, как пользователь может редактировать свой профиль 6.

# Разработка алгоритма решения задачи

IDEF0 диаграмма веб-приложения «Букшеринг» изображена на рисунке 7.

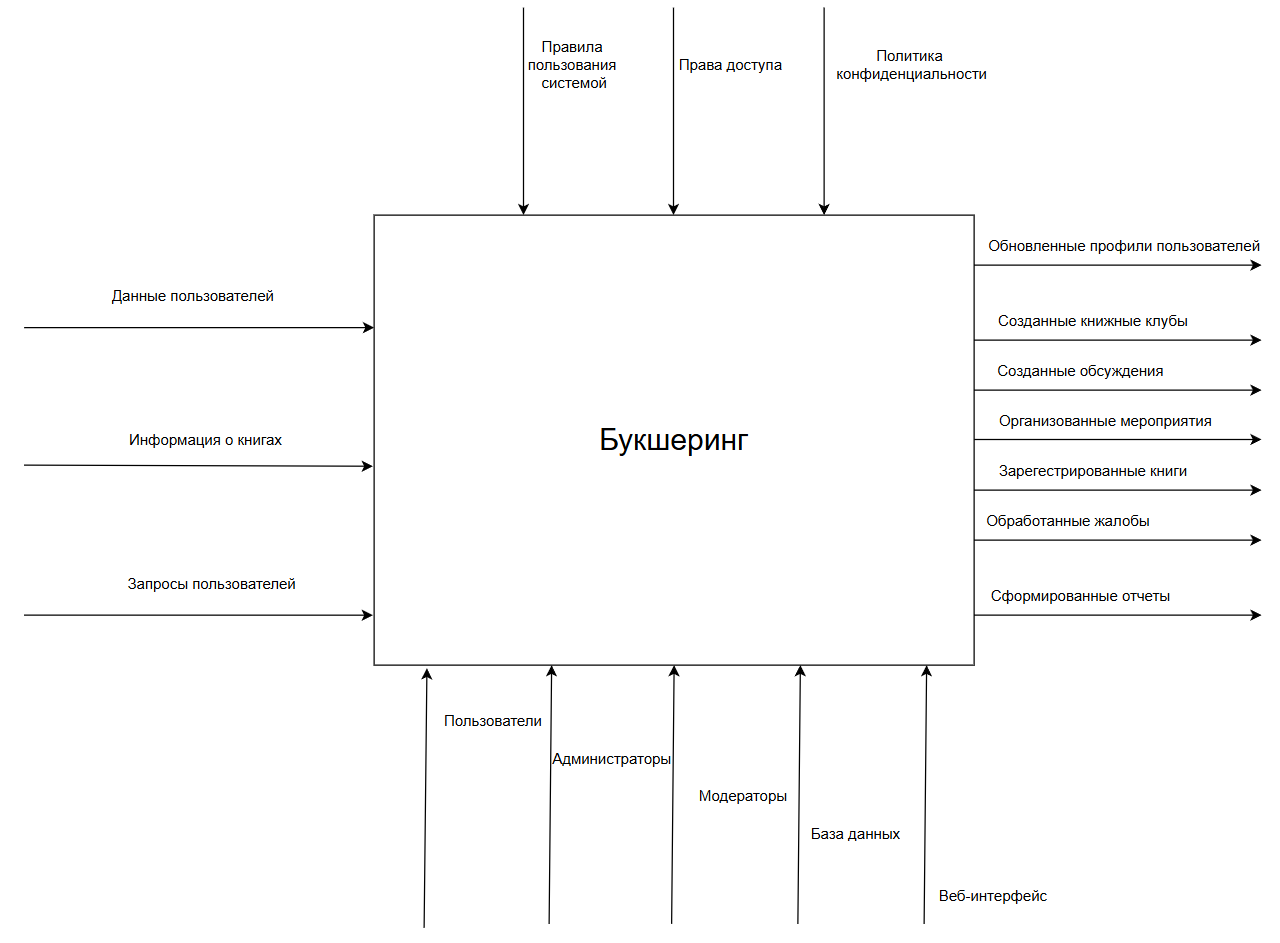


Рисунок 7 – IDEF0 диаграмма веб-приложения

Для понимания как работает система создания книжного клуба была создана схема алгоритма, которая изображена на рисунке 8.

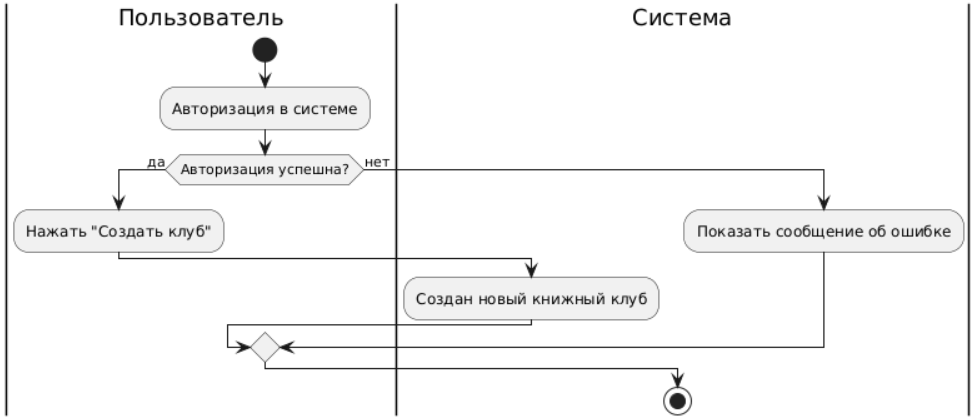


Рисунок 8 – Схема алгоритма «Книжный клуб»

Алгоритм «**Создание нового книжного к»**

Для лучшего понимания алгоритма «Избранное» была создана схема алгоритма, которая изображена на рисунке 8.

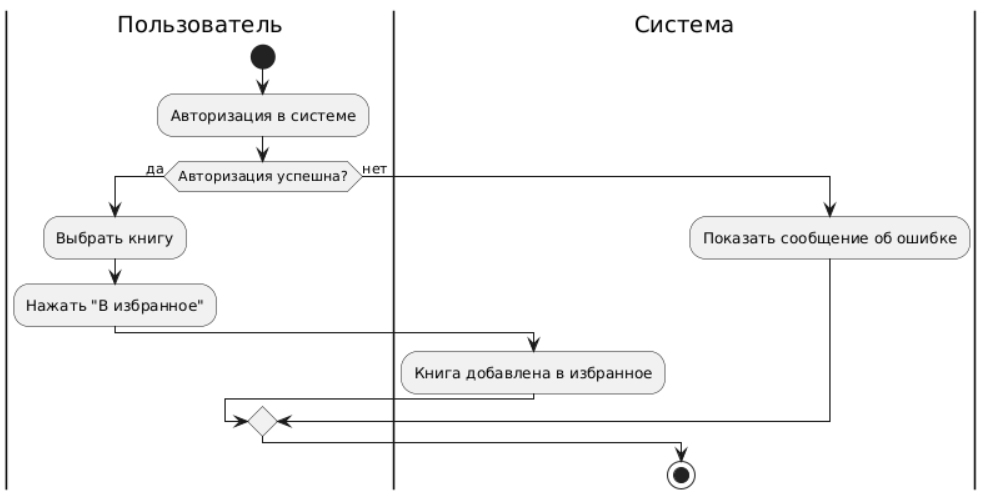


Рисунок 8 – Схема алгоритма «Избранное»

Алгоритм «**Обсуждение»**

Для лучшего понимания алгоритма «Обсуждение» была создана схема алгоритма, которая изображена на рисунке 9.

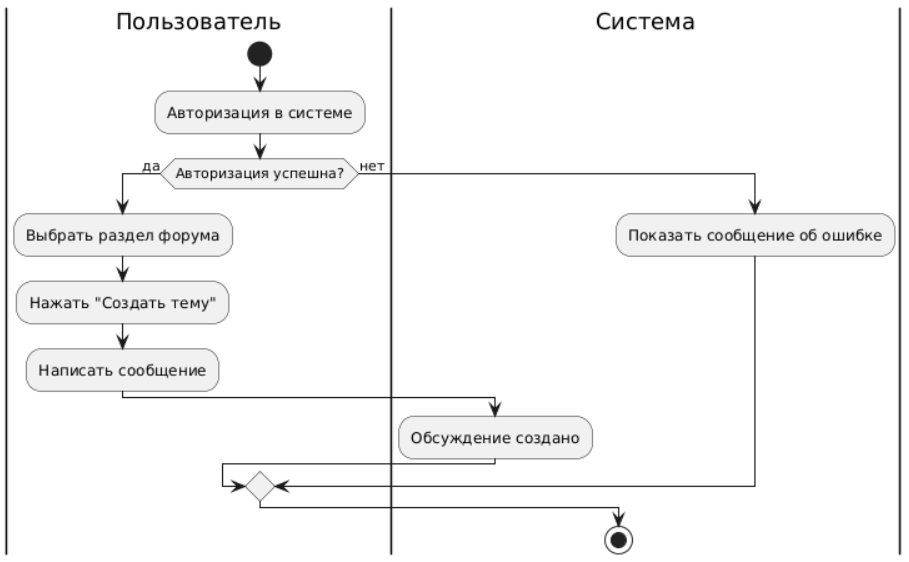


Рисунок 9 – Схема алгоритма «Обсуждение»

Алгоритм «**Блокировка/остранение»**

Для лучшего понимания алгоритма «Блокировка/остранение» была создана схема алгоритма, которая изображена на рисунке 10.

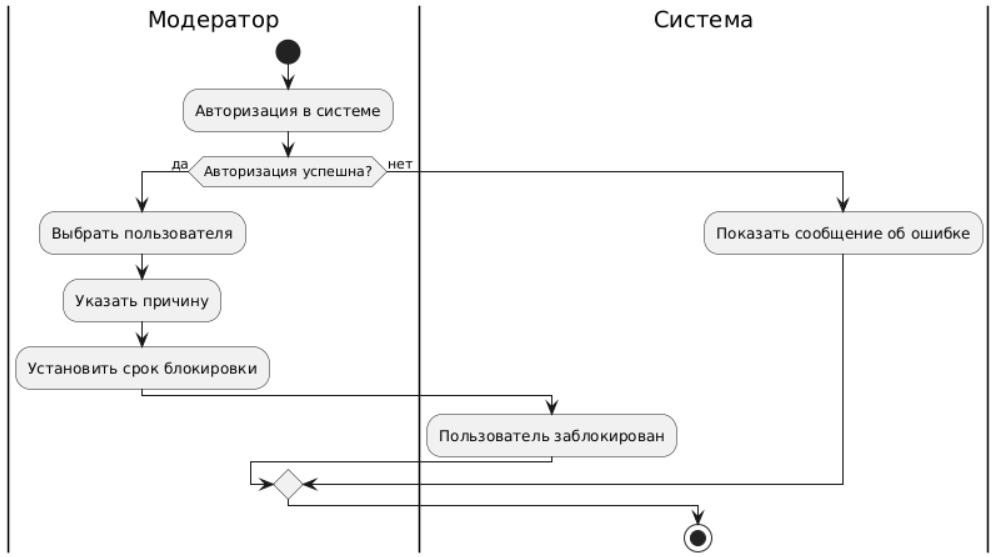


Рисунок 10 – Схема алгоритма «Блокировка/остранение»

Алгоритм «**Редактирование профиля»**

Для лучшего понимания алгоритма «Редактирование профиля» была создана схема, которая изображена на рисунке 11.

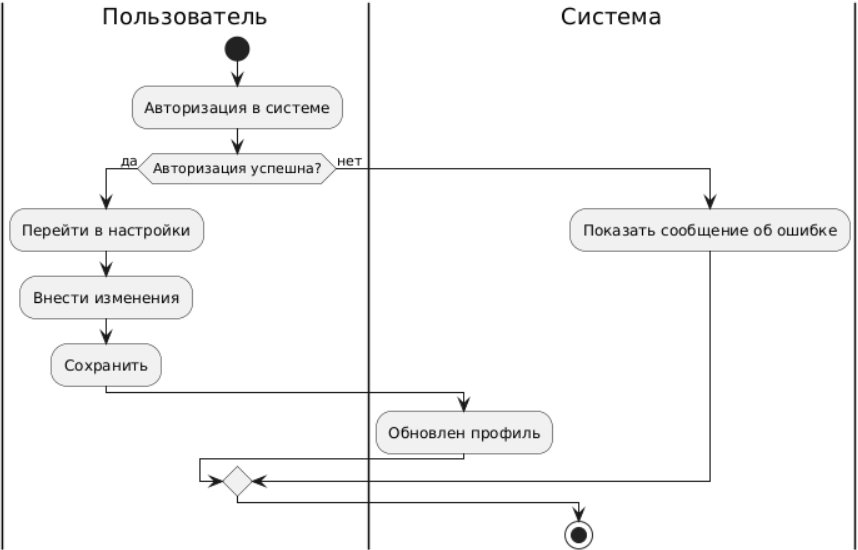


Рисунок 11 – Схема алгоритма «Редактирование профиля»

Алгоритм «**Мероприятие»**

Для лучшего понимания алгоритма «Мероприятие» была создана схема, которая изображена на рисунке 12.

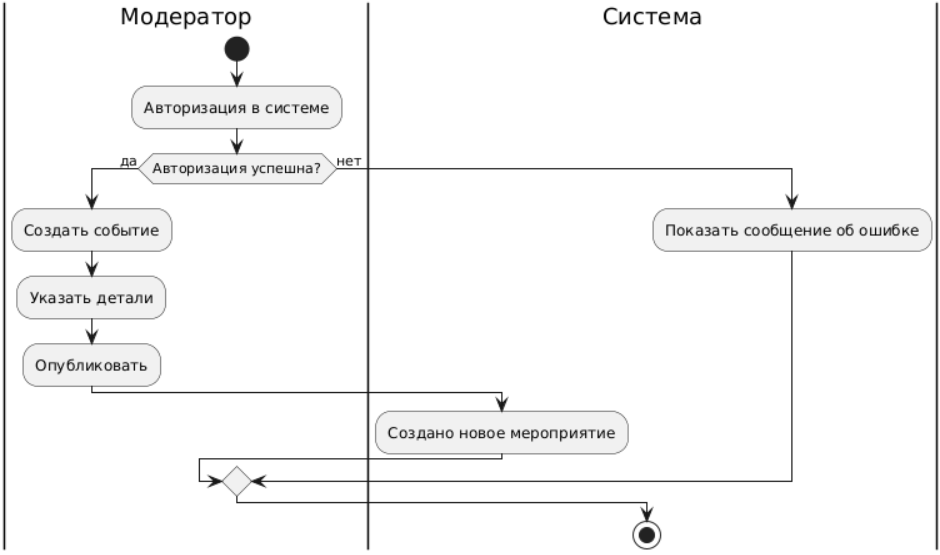


Рисунок 12 – Схема алгоритма «Мероприятие»

Алгоритм «**Жалобы»**

Для лучшего понимания алгоритма «Жалоб» была создана схема, которая изображена на рисунке 13.

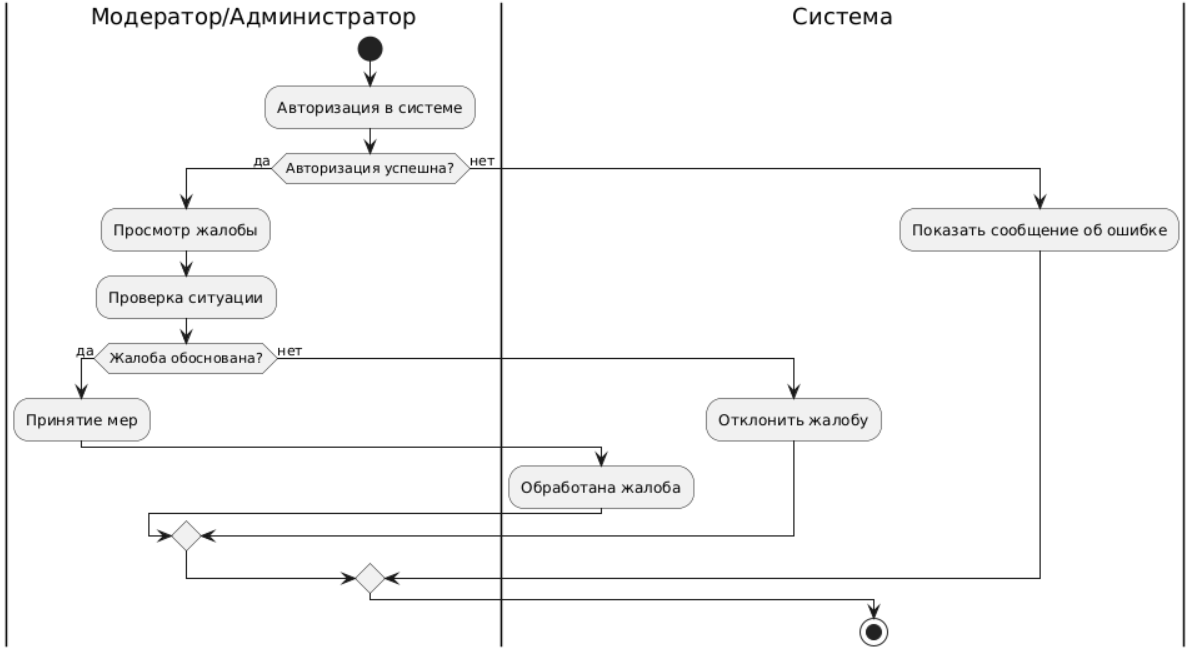


Рисунок 13 – Схема алгоритма «Жалобы»

Алгоритм «**Управление категориями**»

Для лучшего понимания алгоритма «Управление категориями» была создана схема, которая изображена на рисунке 14.

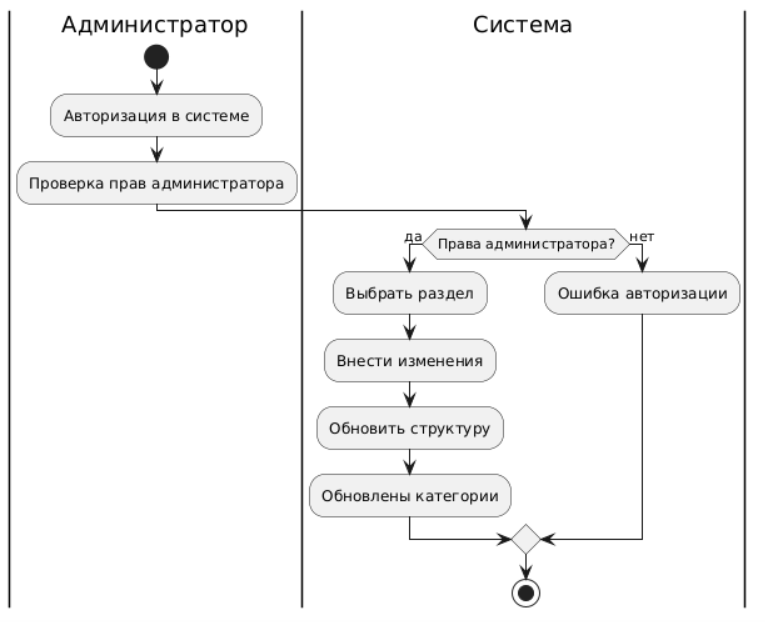


Рисунок 14 – Схема алгоритма «Управление категориями»

Алгоритм «**Отчет**»

Для лучшего понимания алгоритма «Отчет» была создана схема, которая изображена на рисунке 15.

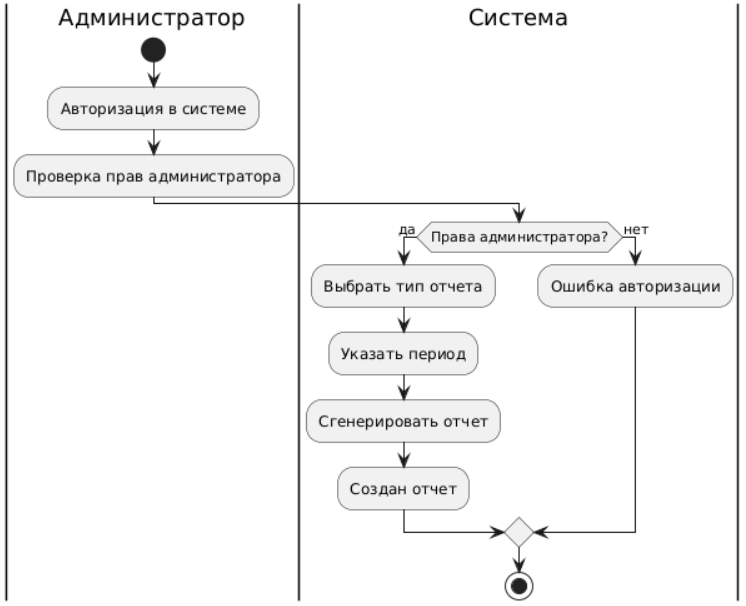


Рисунок 15 – Схема алгоритма «Отчет»

Алгоритм «**Сообщения**»

Для лучшего понимания алгоритма «Сообщения» была создана схема, которая изображена на рисунке 16.

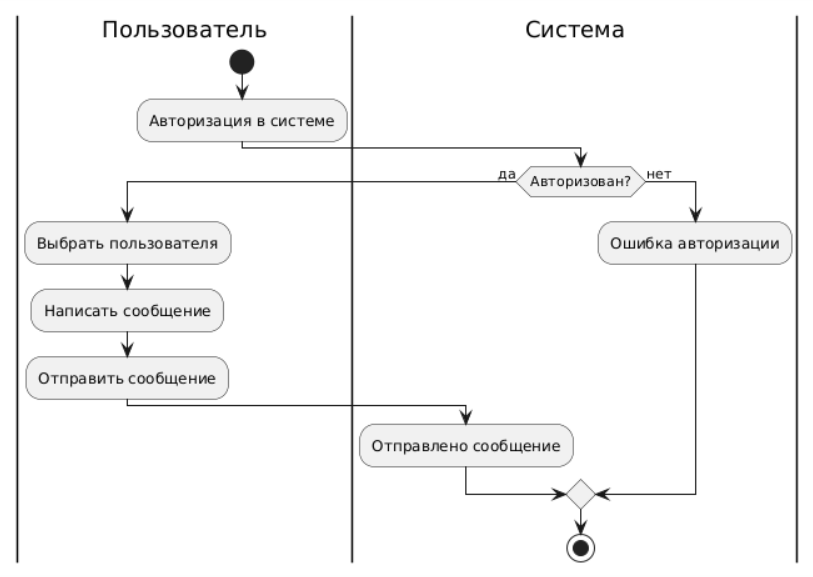


Рисунок 16 – Схема алгоритма «Сообщения»

Алгоритм «**Оценка**»

Для лучшего понимания алгоритма «Оценка» была создана схема, которая изображена на рисунке 17.

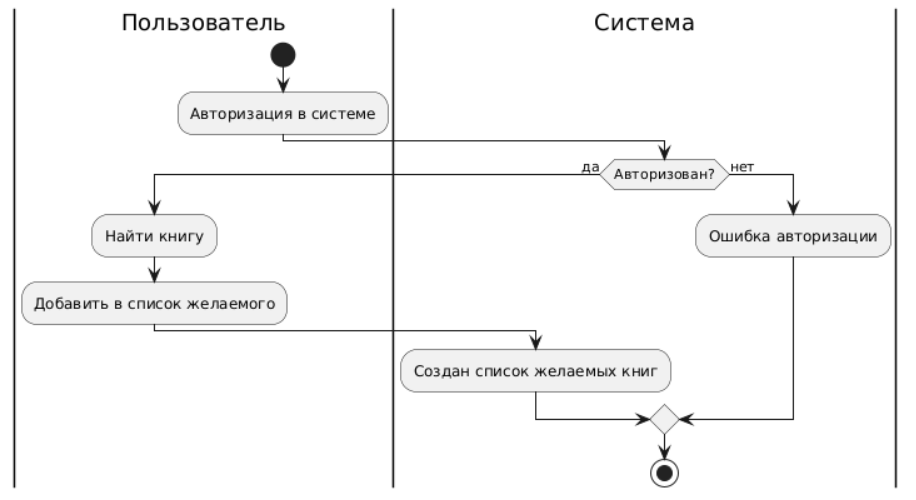


Рисунок 17 – Схема алгоритма «Оценка»

Алгоритм «**Создание списка желаемого**»

Для лучшего понимания алгоритма «Создание списка желаемого» была создана схема, которая изображена на рисунке 18.

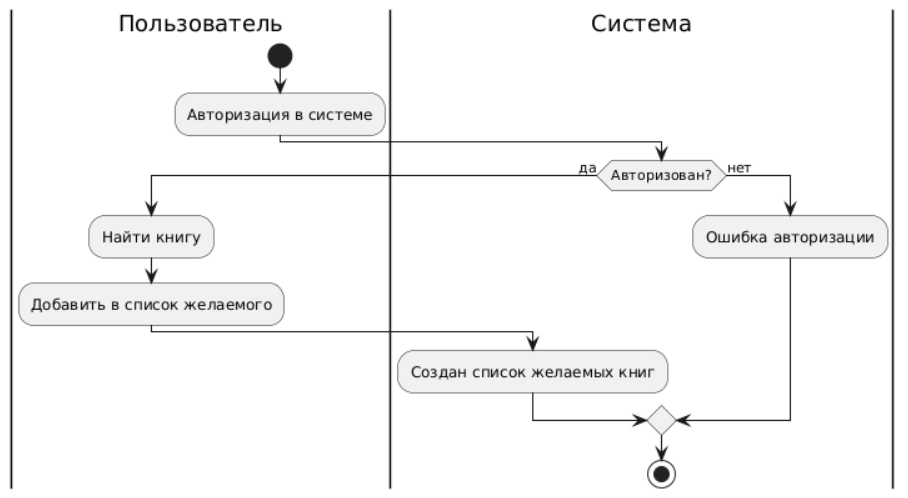


Рисунок 18 – Схема алгоритма «Создание списка желаемого»

Так же для понимания как реагирует система был составлен пользовательский сценарий, который указан в таблице 1.

Таблица 1 – Пользовательские сценарий

|  |  |  |
| --- | --- | --- |
| № функции | Пользователь | Система |
| 1 | Создание книжного клуба | 1. Отображает форму для создания клуба.  2. После ввода данных и нажатия кнопки "Создать", создает новый клуб с указанными параметрами.  3.Перенаправляет пользователя на страницу созданного клуба. |
| 2 | Добавление в избранное | 1. Отображает книгу  2. После нажатия на кнопку для добавления в избранное, создается список избранных книг. |
| 3 | Создание обсуждения | 1. Отображает форму для создания обсуждения.  2. После ввода данных тематики обсуждения, создается обсуждение с указанными параметрами.  3. Перенаправляет пользователя на страницу созданного обсуждения. |
| 4 | Редактирование профиля | 1. Отображение страницы профиля.  2. После ввода изменений обновляет страницу профиля. |
| 5 | Организация мероприятия | 1. Отображение страницы организации мероприятия.  2. После ввода нужных данных мероприятия создает страницу мероприятия. |
| 6 | Обработка жалоб | 1. Отображение жалобы  2. После принятия решения либо жалоба отклоняется, либо нарушитель блокируется на сайте. |
| 7 | Управление категориями | 1. Отображения списка категорий.  2. После редактирования списка сохраняет изменения. |
| 8 | Формирование отчетов | 1. Отображение видов отчетов.  2. После выбора отчета формирует и отображает его. |
| 9 | Обмен сообщениями | 1. Отображает страницу сообщений. 2. Перенаправляет полученные и отправленные сообщения от пользователя к пользователю. |
| 10 | Оценка книги | 1. Отображение страницы книги.  2. После выбора оценки обновляет общий рейтинг книги. |
| 11 | Настройка уведомлений | 1. Отображает страницу настройки уведомлений.  2. После изменений нужных данных обновляет правила уведомлений для пользователей. |
| 12 | Отметка о прочтении | 1. Отображение страницы книги.  2. После нажатия кнопки о прочтение помечает книгу как прочитанная. |
| 13 | Создание списка желаемого | 1. Отображение страницы книги.  2. После нажатия кнопки добавления в список желаемого создает или обновляет список желаемых книг пользователя. |

# Определения языка

Для реализации проекта "Вебшеринг", будут использоваться следующие языки программирования и технологии:

* HTML (HyperText Markup Language). Язык разметки, используемый для структуры и содержания веб-страниц. В проекте HTML будет применяться для создания каркаса пользовательского интерфейса, включая элементы управления, такие как кнопки, панели инструментов и формы.
* CSS (Cascading Style Sheets). Таблицы стилей, используемые для оформления и визуальной презентации веб-страниц. CSS обеспечит адаптивный и современный дизайн интерфейса сайта, а также поддержит кросс-браузерную совместимость.
* JavaScript (JS). Основной язык программирования, ответственный за интерактивные элементы на стороне клиента. JavaScript будет управлять функционалом, управления доступом пользователей (например, запрет или разрешение на редактирование), и обновления данных в реальном времени без перезагрузки страницы.
* Node.js. Среда выполнения JavaScript на серверной стороне. Node.js будет использоваться для обработки запросов от клиентов, управления сеансами работы, обеспечения безопасности и синхронизации действий между пользователями в режиме реального времени.
* Библиотеки и фреймворки. В проекте будут использоваться специализированные библиотеки и фреймворки, которые обеспечат создание функционала сайта. В том числе: Socket.IO для реализации взаимодействия пользователей в режиме реального времени. Express.js для управления серверной логикой.

# Структура программы и требования к техническим средствам

Функциональные требования:

* Авторизация и регистрация;
* Возможность пользоваться поиском книг и сообществ;
* Регистрация и размещение своей книги на платформе;
* Участие в обсуждениях и книжных клубах;
* Оставление отзывов и комментариев;
* Просмотр каталога книг;
* Управление профилем;
* Отправка сообщений;

Нефункциональные требования:

* Удобство использования: интуитивно понятный и простой интерфейс, адаптивный дизайн для различных устройств;
* Надежность: система должна быть стабильной и доступной большую часть времени;

Взаимодействие с внешними системами:

Интеграция с системами аутентификации:

* Возможность входа через VK ID;