**Веб-приложение «Букшеринг»**

**Руководство пользователя**

**Содержание**

[1 Введение 2](#_Toc182784809)

[1.1 Наименование системы 2](#_Toc182784810)

[1.2 Область применения 2](#_Toc182784811)

[1.3 Условные обозначения системы 2](#_Toc182784812)

[2 Установка системы 3](#_Toc182784813)

[2.1 Требование для установки 3](#_Toc182784814)

[2.2 Установка зависимостей 3](#_Toc182784815)

[2.3 Настройка базы данных 3](#_Toc182784816)

[2.4 Запуск сервера 3](#_Toc182784817)

[3 Внутреняя структура системы 4](#_Toc182784818)

[3.1 Используемый технологический стек 4](#_Toc182784819)

[3.2 Функции и их описание 4](#_Toc182784820)

[3.3 Структура данных 5](#_Toc182784821)

[3.4 Описание API 9](#_Toc182784822)

# Введение

## Наименование системы

Сайт «Букшеринг».

## Область применения

«Букшеринг» представляет собой веб-приложение, создано для объединения любителей чтения на единой цифровой платформе, где можно обмениваться книгами и общаться. Система реализует гибкое распределение прав доступа - от базовых возможностей для гостей до расширенного функционала для администраторов.

## Условные обозначения системы

Условное обозначение сайта «Букшеринг» - Сайт.

# Установка системы

## Требование для установки

Для развертывания Сайта нужен сервер, который поддерживает Node.js и так же PostgreSQL. И так же на сервер должны установлены следующие библиотеки: Canvas API и т.д.

Так же Сайт должен поддерживать все современные браузеры, в числе которых есть: Chrome, Firefox, Safari и т.д.

## Установка зависимостей

Фронтенд: установите зависимости для фронтенда (если есть), выполнив следующую команду:

npm install

Бэкенд: установите необходимые библиотеки node.js (если применимо), например, через Composer:

npm install expres

## Настройка базы данных

Чтоб установить базу данных на сервер нужно создать новую базу данных в PostgreSQL и импортировать готовую структуру из уже существующей базы данных.

## Запуск сервера

Чтоб запустить сервер, в папке проекта нужно открыть терминал и вписать следующую команду:

npm start

# Внутреняя структура системы

## Используемый технологический стек

Сайт использует для фронтенда HTML и CSS, для бекэнда используется Node.js с библиотекой Canvas API. В качестве базы данных используется PostgreSQL

## Функции и их описание

У Сайта имеются следующие функции:

* Создание книжного клуба - создает новую запись в базе данных с информацией о клубе (название, описание, правила), генерирует уникальный ID клуба и связывает его с создателем;
* Добавление в избранное - добавляет запись в таблицу избранного, связывая ID пользователя с ID книги;
* Создание обсуждения - создает новую запись в базе данных форума с темой, содержанием и метаданными, связывает с автором и категорией;
* Блокировка нарушителя - обновляет статус пользователя в базе данных, устанавливая флаг блокировки и срок действия;
* Редактирование профиля - обновляет информацию о пользователе в базе данных, включая личные данные и настройки;
* Организация мероприятия - создает запись в базе данных мероприятий с датой, описанием и участниками, генерирует уведомления;
* Обработка жалоб - создает тикет в системе модерации, обновляет статусы жалоб в базе данных;
* Управление категориями - позволяет создавать, редактировать и удалять категории в базе данных, обновляя связи с книгами;
* Формирование отчетов - использует SQL-запросы для сбора статистики из базы данных и формирует отчеты через библиотеку генерации PDF;
* Обмен сообщениями - реализует чат через WebSocket, сохраняя историю сообщений в базе данных;
* Оценка книги - обновляет рейтинг книги в базе данных, пересчитывает средний балл;
* Восстановление пароля - генерирует временный токен, отправляет email через SMTP, обновляет пароль в базе данных;
* Настройка уведомлений - сохраняет предпочтения пользователя по уведомлениям в базе данных;
* Отметка о прочтении - добавляет запись в историю чтения пользователя, обновляет статистику;
* Создание списка желаемого - создает связи между пользователем и желаемыми книгами в базе данных.

## Структура данных

Сайт имеет база данных, структура этой базы данных изображена в виде логической ER на рисунке 1 и физической ER которая изображена на рисунке 2 соответственно.

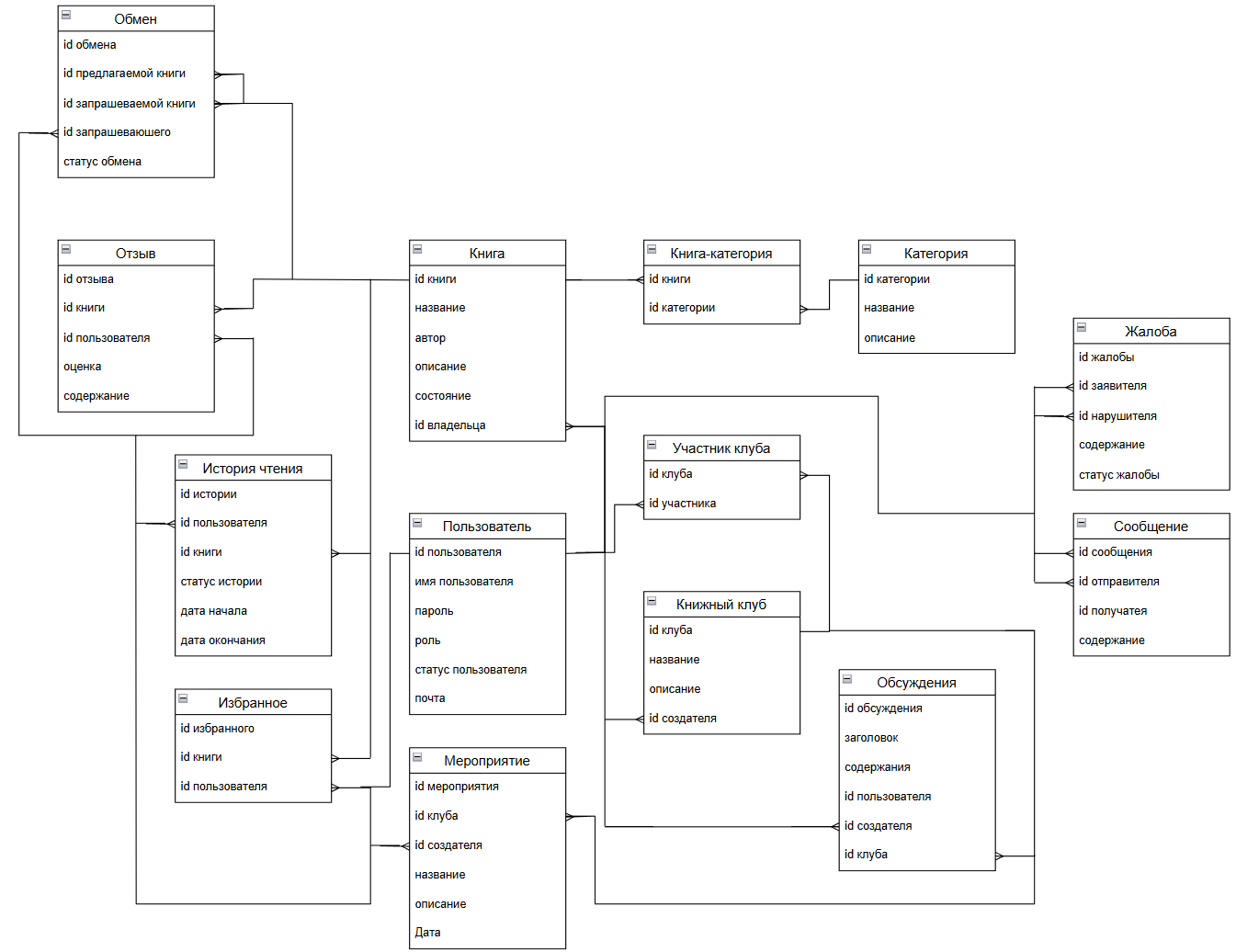


Рисунок 1 – Логическая ER диаграмма

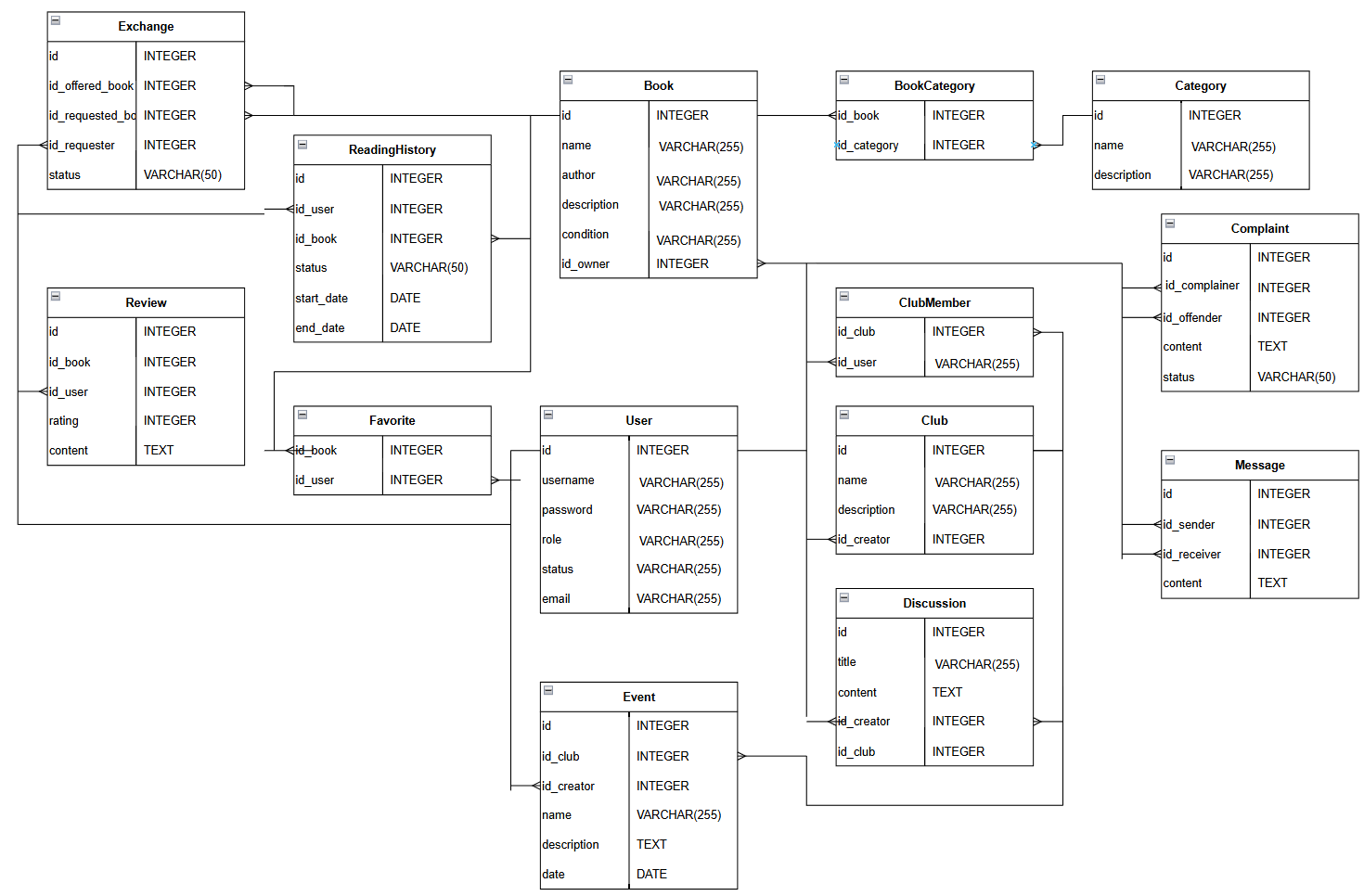


Рисунок 2 – Физическая ER диаграмма

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.

## Описание API

На Сайте используются следующие API запросы:

* POST - запрос – используется для функции «добавления книг»;
* GET - запрос – используется для получения данных о книге;
* PATCH - запрос – используется для редактирования информации о книге
* DELETE - запрос – используется для функции «удаление книги»