



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования

«Дальневосточный федеральный университет»
(ДВФУ)

Институт математики и компьютерных технологий
Департамент математического и компьютерного
моделирования

«Корпоративная библиотека»

ОТЧЕТ

к проекту по дисциплине
«Программная инженерия»

Выполнили студенты
гр. Б9121-09.03.03пикд

Травка Е.А. _____
(ФИО) (подпись)

Блинов Е.Е. _____
(ФИО) (подпись)

Руководитель
ст. преподаватель Кленина Н.В.

(должность) (ФИО)

(подпись)

г. Владивосток, 2023

Содержание

Аннотация	2
1 Введение	3
1.1 Глоссарий	3
1.2 Описание предметной области	4
1.3 Неформальная постановка задачи	6
1.4 Обзор существующих методов решения	7
1.5 План работ	10
2 Требования к окружению	14
2.1 Требования к аппаратному обеспечению	14
2.2 Требования к программному обеспечению	14
2.3 Требования к пользователям	14
2.4 Организационные требования	15
3 Архитектура системы	16
3.1 Backend	16
3.2 Frontend	17
4 Спецификация данных	18
4.1 Описание сущности	18
5 Функциональные требования	25
6 Требования к интерфейсу	27
7 Прочие требования	28
7.1 Требования к надёжности	28
7.2 Требования к безопасности	28
7.3 Требования к производительности	28
8 Проект	29
8.1 Средства реализации	29
8.1.1 Backend	29
8.1.2 Frontend	30
8.2 Структура данных	32
8.3 Модули и алгоритмы	32
8.3.1 Backend	32
8.3.2 Frontend	33
8.5 Проект интерфейса	34
9 Реализация и тестирование	40
9.1 Backend	40
9.2 Frontend	40

Аннотация

Проект ориентирован на использование внутри компании Farpost, его основной задачей является автоматизация процессов взаимодействия с ресурсами корпоративной библиотеки.

1 Введение

1.1 Глоссарий

Библиотека

учреждение, собирающее и осуществляющее хранение произведений печати и письменности для общественного пользования, а также ведущее справочно-библиографическую работу.

Бронировать издание

добавлять запись, включающую дату и название издания в читательскую карточку.

Информационная система

система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Компания

коммерческое предприятие, торговое или промышленное объединение предпринимателей.

Корпоративная библиотека

систематизированная коллекция печатных и электронных изданий, доступная для использования внутри компании.

Читательская карточка

данные о прочитанных Пользователем изданиях, текущих бронированиях.

1.2 Описание предметной области

В современных компаниях все чаще появляются корпоративные библиотеки с профильной литературой. Объем изданий в таких библиотеках может достигать нескольких сотен, вследствие чего возникает необходимость в ведении учета литературы.

Применяемые подходы к решению задачи по учету ресурсов корпоративной библиотеки:

- Ведение учета на бумажных носителях
- Сервисы онлайн-форм (пр. Google Forms)
- Ведение таблиц (пр. Excel, Google Sheets)

Описанные методы решения поставленной задачи ограничивают использование ресурсов корпоративной библиотеки и не обеспечивают должного учета литературных изданий, поэтому требуется применение специализированных программных решений.

При ведении учета на бумажных носителях не обеспечивается требуемая целостность данных. Также этот подход требует наличия библиотекаря, организующего рабочий процесс. В большинстве случаев должность библиотекаря не предусмотрена в предприятии, поэтому данный подход используется в редких случаях.

Сервисы онлайн-форм: в сравнении с бумажными носителями сохраняется целостность данных, однако нет централизованного хранения и структуризации информации об изданиях и читателях. Данный подход позволяет вести только учет изданий, находящихся на руках.

Ведение электронных таблиц: таблицы позволяют организовать базу данных корпоративной библиотеки и хранить большинство требуемой информации. Однако недостатком этого подхода является закрытость, так как версия с актуальными данными хранится локально. При общем доступе для сохранения целостности данных требуется разграничение прав доступа, что не всегда реализуемо удобным способом.

Существующие информационные системы для библиотек в большинстве случаев не применимы к корпоративным библиотекам. Значительное число решений требуют наличие библиотекаря, обеспечивающего выдачу, поддержку актуальности списка имеющихся изданий и т. д.

В связи с этим появляется необходимость в проектировании и разработке системы, автоматизирующей процессы взаимодействия с имеющимися ресурсами литературы и не требующей наличия библиотекаря.

Ключевые преимущества проектируемой системы:

- Автоматизированный процес выдачи и возврата книг - устраняет необходимость в библиотекаре
- Общедоступный систематизированный электронный каталог - ускоряет процесс поиска необходимого издания
- Централизованный доступ к электронным изданиям

1.3 Неформальная постановка задачи

Требуется разработать веб-сайт для автоматизации работы корпоративных библиотек, предоставляющий следующие возможности для пользователей системы:

- Ведение читательской карточки.
- Получение актуального списка изданий с возможностью поиска
- Скачивание электронных изданий (при наличии).
- Отслеживание забронированных книг с учетом сроков.
- Формирование запросов на добавление книг в библиотеку.
- Получение уведомлений о новых поступлениях и истечении срока брони издания.

Предполагаемая политика распространения программного продукта — продажа на открытом рынке.

1.4 Обзор существующих методов решения

Для анализа выбраны следующие программные продукты:

1. CodeAchi [13]

Одна из самых востребованных платформ на зарубежном рынке. Представляет собой оконное приложение (Windows). Благодаря обширному функционалу и интуитивному интерфейсу популярна среди публичных библиотек, школ, колледжей и университетов.

2. Alexandria [3]

Веб-сайт, ориентированный на людей любых возрастов. Подходит как для небольших, так и для крупных библиотек. В дополнение, существует детский раздел.

3. Mandarin M5 [32]

Представляет собой веб-интерфейс. Дополнительные модули позволяют библиотекам настраивать свою систему в соответствии со своими индивидуальными потребностями. Также платформа поддерживает Unicode, что даёт возможность пользователям искать материалы на разных языках, включая китайский, арабский и т. д.

4. Handy Library Manager [23]

Гибридное локальное и облачное решение системы управления библиотекой с доплатой за последнее. В отличие от других платформ, поддерживает хранение неограниченного количества книг и их каталогизацию.

5. KnowAll Matrix [26]

Представляет собой веб-интерфейс. Поддерживает хранение как печатных, так и электронных изданий.

6. 1С: Библиотека [1]

Продукт позволяет автоматизировать рабочие процессы библиотеки в зависимости от ее назначения, типа, состава фондов. Имеется виртуальный кабинет читателя. Также может быть интегрирован с другими типовыми решениями фирмы "1С".

7. OPAC-Global [35]

Информационная система, основанная на облачных технологиях. Её можно развернуть как для отдельной небольшой библиотеки, так и для

сети библиотек. Систему используют в Президентской библиотеке имени Б. Н. Ельцина, научной библиотеке Государственной Третьяковской галереи, Центральной научной библиотеке ПМГМУ им. И. М. Сеченова и других.

8. Либэр [50]

Имеется два варианта исполнения: оконное приложение (Windows, MacOS, Linux) и браузерное решение. Реализована возможность работы сети библиотек с разными базами или единой базой.

9. АБИС "Марк" [44]

Представляет собой оконное приложение (Windows, MacOS, Linux). У программы имеется облачная версия. Поддерживает различные популярные СУБД, можно гибко настроить и расширить функциональные возможности благодаря использованию встроенных скриптов.

10. Проектируемое решение

Таблица 1: Анализ программных продуктов-аналогов

Критерии	1	2	3	4	5	6	7	8	9	10
Автоматизированный процесс получения и возврата изданий	–	+	–	–	–	–	–	–	–	+
Возможность формировать запросы на добавление книг	–	–	–	–	+	–	–	+	+	+
Заполнение базы данных различными способами (вручную, ISBN)	+	–	–	+	+	+	+	+	+	+
Отслеживание задолженностей	+	+	+	–	+	+	+	+	+	+
Доступ к электронным изданиям	–	–	–	+	+	+	+	+	+	+
Возможность оставлять рецензии на книги	–	+	+	–	+	–	–	–	–	+
Формирование очередей на получение книги	–	–	–	–	–	+	–	+	–	+
Уведомления о новых поступлениях	–	–	–	+	+	–	–	–	–	+
Уведомления об истечении срока бронирования	+	+	+	+	+	–	+	–	+	+
Наличие русской локализации	–	–	–	–	–	+	+	+	+	+

В рамках Зимней проектной школы ДВФУ 2023 была разработана MVP версия описываемого программного продукта. В разработанном решении реализован только базовый функционал: учет книг, основные операции взаимодействия с библиотекой, учет сроков бронирования. Так как мероприятие,

в ходе которого решалась задача, ограничено сроком в 2 недели, времени на реализацию всех функциональных требований, проектирование интерфейса и базы данных было недостаточно. Система требует доработки и расширения существующего функционала.

На основании проведённого анализа перечисленных решений сделан вывод. 5 из 9 решений не имеют русской локализации, так как ориентированы на зарубежный рынок. Продукты, разработанные российскими компаниями, зачастую не соответствуют необходимым функциональным требованиям и давно не обновляются, что затрудняет их использование на современных устройствах. В связи с этим, есть необходимость в разработке продукта, ориентированного на российский рынок и современные требования.

1.5 План работ

Таблица 2: План работ

Дата	Блинов Егор	Травка Евгения
16.03	Написание раздела «Функциональные требования»	Внесение правок в отчет
25.03	Планирование задач по разработке	Разработка архитектуры программного продукта
01.04	Разработка требований к интерфейсу, проект интерфейса	Построение диаграммы вариантов использования
08.04	Разработка требований к пользователям, диаграмма состояний	Разработка требований к ПО, спецификация данных, прочие требования
15.04	Проектирование сущностей БД	Описание спецификации данных
22.04	Рефакторинг кода (frontend), настройка CI/CD	Рефакторинг кода (backend), созвон с "Farpost"
29.04	Отображение информации о нескольких экземплярах	Создание сущности «Экземпляр книги»
06.05	Страница администратора	Реализация электронных изданий
13.05	Фильтрация	Каталогизация

Таблица 2: План работ

20.05	Добавление книг по коду	Изменение связи на «многие-ко-многим» между сущностями «Книга» и «Автор»
27.05	Реализация «случайной книги», подбор цветowych тем	Реализация «очереди на книги», обработка ошибок
03.06	Подготовка презентации продукта	Подготовка доклада
10.06	Тестирование frontend-части, анализ результатов пользовательского тестирования	Тестирование backend-части, проведение пользовательского тестирования
17.06	Внесение правок в продукт, развертывание	Внесение правок в продукт, интеграционные тесты



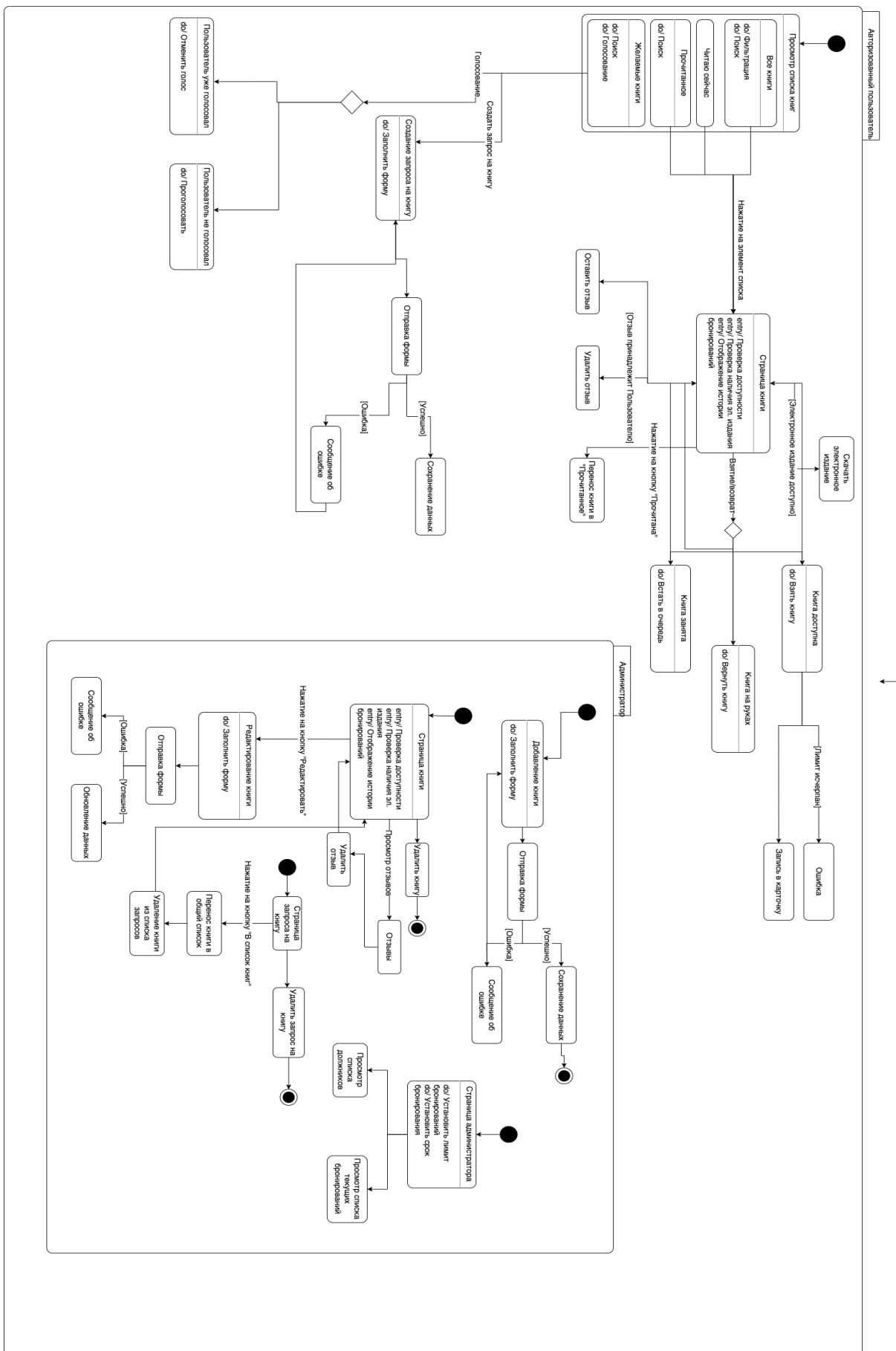


Рис. 2: Диаграмма состояний

2 Требования к окружению

2.1 Требования к аппаратному обеспечению

Для серверной части должны соблюдаться следующие требования:

- Архитектура процессора: x86-64 или amd64.
- ОЗУ не менее 2гб (для корректной работы Docker, при его использовании).
- Подключение к интернету.

Для клиентской части должны соблюдаться следующие требования:

- Подключение к интернету.

2.2 Требования к программному обеспечению

Для серверной части должны соблюдаться следующие требования:

- ОС, поддерживающая работу с Docker (в случае с Windows - требуется WSL).
- Установленный Docker.

Для клиентской части должны соблюдаться следующие требования:

- Поддерживаемый браузер с установленными недавними обновлениями. (Система не поддерживает старые версии Safari и IE11)
- JavaScript в браузере должен быть включен.

2.3 Требования к пользователям

Система рассчитана на сотрудников компаний, в рамках которых интегрируется система.

Существующие роли в системе: Пользователь (Читатель) и Администратор.

Администратор должен иметь постоянный доступ к ресурсам библиотеки, для поддержания актуальности данных в системе.

2.4 Организационные требования

Требуется введение ставки администратора сервиса корпоративной библиотеки. В обязанности администратора входит: ведение базы данных библиотеки, контроль просроченных бронирований, формирование заявок на закупку на основе запросов Пользователей.

3 Архитектура системы

Архитектура системы требуется для упрощения поддержки и модификации продукта в долгосрочной перспективе. Правильное разделение на модули упрощает выделение сущностей, их переиспользование.

3.1 Backend

Для backend части выбран традиционный для фреймворка Laravel паттерн Model-View-Controller, с выделением дополнительных слоев бизнес-логики (Services).

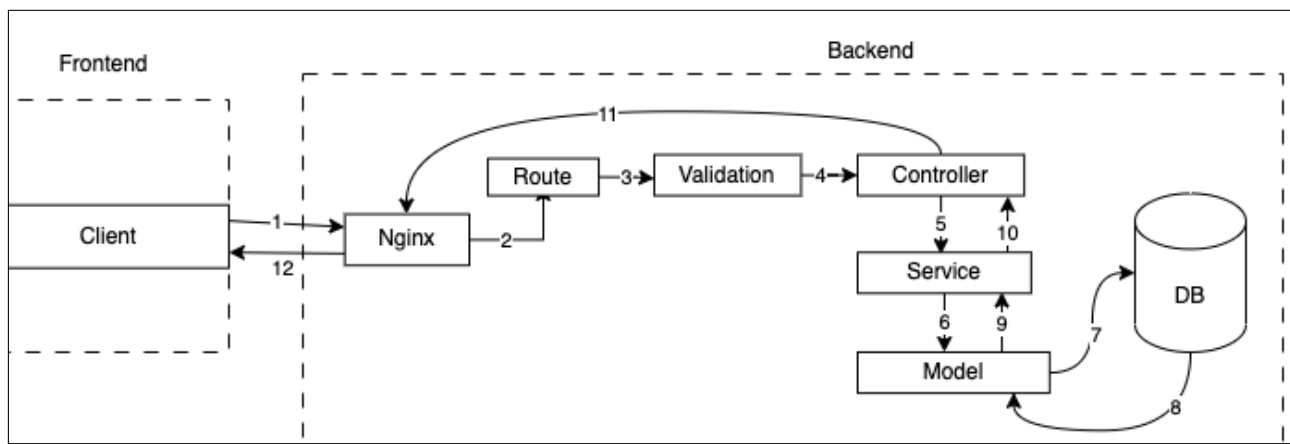


Рис. 3: Схема backend архитектуры

1. Клиент отправляет запрос на сервер.
2. Запрос обрабатывается Nginx (или другим сервером).
3. Проверяется путь запроса с помощью класса Route, вызывается метод связанного с путем Controller.
4. Происходит валидация запроса, объект с информацией о запросе попадает в Controller.
5. Controller вызывает методы связанных с ним Service - классов.
6. Service класс вызывает методы связанных с ним классов Model.
7. Model получает данные из базы данных.
8. Model возвращает данные в Service.
9. Service возвращает результирующий объект с данными.

10. Controller возвращает ответ с полученными данными.

- Route — класс, связывающий URL, допустимые HTTP-методы для него и метод контроллера, вызываемый при запросе по указанному URL.
- Controller — класс, содержащий методы для обработки запросов. Для каждого Controller существует Service и Model.
- Service — класс, содержащий в себе методы для работы с соответствующими Model. Чаще всего это операции чтения, удаления, вставки.
- Model — класс, связанный с конкретной таблицей в базе данных. Позволяет запрашивать данные из этой таблицы. Наследуется от базового класса Model из Eloquent ORM.

3.2 Frontend

Для frontend части достаточно разделения на компоненты (компоненты интерфейса и компоненты взаимодействующие с бизнес-логикой). Из компонентов собираются страницы, с использованием layout компонентов. Так как за хранение состояния приложения и работу с данными отвечает сервер, проектирование хранилища глобального состояния приложения не требуется.

4 Спецификация данных

4.1 Описание сущности

Сущность «Пользователь», предназначенная для хранения информации о пользователях, содержит следующие поля:

- Имя
 - Тип: строка
 - Ограничения: содержит только буквы и тире, не более 50 символов
 - Обязательно
 - Доступно для редактирования
 - Неуникально
- Фамилия
 - Тип: строка
 - Ограничения: содержит только буквы и тире, не более 50 символов
 - Обязательно
 - Доступно для редактирования
 - Неуникально
- Отчество
 - Тип: строка
 - Ограничения: содержит только буквы, не более 50 символов
 - Необязательно
 - Доступно для редактирования
 - Неуникально
- Почта
 - Тип: строка
 - Ограничения: действительный электронный адрес
 - Обязательно

- Доступно для редактирования
- Уникально
- Пароль
 - Тип: строка
 - Ограничения: не меньше 8 символов, не больше 30 символов
 - Обязательно
 - Доступно для редактирования
 - Неуникально
- Отправлять уведомления о новых поступлениях
 - Тип: логический тип
 - Нет ограничений
 - Обязательно
 - Доступно для редактирования
 - Неуникально
 - Значение по умолчанию: true
- Отправлять уведомления об истечении срока бронирования
 - Тип: логический тип
 - Нет ограничений
 - Обязательно
 - Доступно для редактирования
 - Неуникально
 - Значение по умолчанию: true

Сущность «Автор», предназначенная для хранения информации об авторах книг, содержит следующие поля:

- Полное имя
 - Тип: строка

- Ограничения: содержит только буквы, пробелы и тире, не более 65 символов
- Обязательно
- Недоступно для редактирования
- Уникально

Сущность «Книга», предназначенная для хранения информации о книгах библиотеки, содержит следующие поля:

- ISBN

- Тип: целое число
- Ограничения: положительное число, не менее 10, не более 13 цифр в его записи
- Необязательно
- Доступно для редактирования
- Уникально

- Название

- Тип: строка
- Ограничения: не более 100 символов
- Обязательно
- Доступно для редактирования
- Неуникально

- Описание

- Тип: текст
- Ограничения: не более 2500 символов
- Необязательно
- Доступно для редактирования
- Неуникально

- Год публикации

- Тип: год
- Ограничения: не раньше 1900 года, не позже текущего года
- Обязательно
- Доступно для редактирования
- Неуникально
- Количество страниц
 - Тип: целое число
 - Ограничения: положительное число, не более 4 цифр в его записи
 - Необязательно
 - Доступно для редактирования
 - Неуникально
- Путь к обложке
 - Тип: строка
 - Нет ограничений
 - Необязательно
 - Доступно для редактирования
 - Уникально
- Путь к электронному изданию
 - Тип: строка
 - Нет ограничений
 - Необязательно
 - Доступно для редактирования
 - Уникально

Сущность «Экземпляр книги» предназначена для хранения информации о копиях книг, содержит поле:

- Книга
 - Тип: ссылка на *Книга*

- Нет ограничений
- Обязательно
- Недоступно для редактирования
- Неуникально

Сущность «Желаемая книга» предназначена для хранения информации о книгах, которые Пользователи хотели бы видеть в библиотеке. Содержит все те же поля, что и *Книга*. В дополнение, содержит следующее поле:

- Пользователь

- Тип: ссылка на *Пользователь*
- Нет ограничений
- Обязательно
- Недоступно для редактирования
- Неуникально

Сущность «Бронирование», предназначенная для хранения информации о бронированиях экземпляров книг Пользователями, содержит следующие поля:

- Экземпляр книги

- Тип: ссылка на *Экземпляр книги*
- Нет ограничений
- Обязательно
- Недоступно для редактирования
- Неуникально

- Пользователь

- Тип: ссылка на *Пользователь*
- Нет ограничений
- Обязательно
- Недоступно для редактирования

- Неуникально
- Дата бронирования
 - Тип: дата и время
 - Нет ограничений
 - Обязательно
 - Недоступно для редактирования
 - Неуникально
- Дата возврата
 - Тип: дата и время
 - Нет ограничений
 - Необязательно
 - Недоступно для редактирования
 - Неуникально
 - Значение по умолчанию: null
- Дата окончания срока бронирования
 - Тип: дата и время
 - Нет ограничений
 - Обязательно
 - Недоступно для редактирования
 - Неуникально

Сущность «Отзыв», предназначенная для хранения информации об отзывах Пользователей на книги, содержит следующие поля:

- ОТЗЫВ
 - Тип: текст
 - Ограничения: не более 1000 символов
 - Обязательно
 - Недоступно для редактирования

- Неуникально
- Книга
 - Тип: ссылка на *Книга*
 - Нет ограничений
 - Обязательно
 - Недоступно для редактирования
 - Неуникально
- Пользователь
 - Тип: ссылка на *Пользователь*
 - Нет ограничений
 - Обязательно
 - Недоступно для редактирования
 - Неуникально

Сущность «Тег», предназначенная для хранения информации о тегах для книг, содержит следующие поля:

- Название
 - Тип: текст
 - Ограничения: не более 30 символов
 - Обязательно
 - Доступно для редактирования
 - Уникально
- Цвет
 - Тип: текст
 - Нет ограничений
 - Обязательно
 - Доступно для редактирования
 - Неуникально

5 Функциональные требования

Система должна позволять Пользователю:

- Создать учетную запись, используя пару email-пароль.
- Войти в систему, используя пару email-пароль.
- Редактировать личные данные, включая email и пароль.
- На странице «Профиль» включать/отключать уведомления:
 - О новых поступлениях.
 - Об истечении срока бронирования.
- Получать уведомления на email:
 - О новых поступлениях.
 - Об истечении срока бронирования.
 - О книгах, на получение которых Пользователь ранее вставал в очередь.
- Получать актуальный список имеющейся литературы.
- Формировать поисковые запросы по названию, автору, ключевым словам по базе библиотеки.
- Просматривать страницу книги с информацией о ней.
- Брать книгу с добавлением соответствующей записи в читательскую карточку.
- Возвращать книгу с добавлением соответствующей записи в читательскую карточку.
- Видеть, у кого находится книга, если она занята.
- Записываться в очередь на книгу.
- Скачивать электронное издание на странице соответствующей книги (при его наличии).
- Оставлять отзыв на странице книги.

- Удалять оставленные отзывы.
- Просматривать отзывы других пользователей.
- Получать список книг с датой истечения срока бронирования, находящихся у Пользователя на руках.
- Получать список книг, прочитанных Пользователем.
- Формировать запросы на добавление книги в библиотеку.
- Голосовать в запросах на книги других пользователей.
- Менять цветовую схему интерфейса.
- Выйти из системы.

Система должна позволять Администратору:

- Совершать все действия, доступные Пользователю.
- Добавлять/редактировать/удалять информацию о книгах вручную.
- Добавлять книгу, используя ISBN.
- Загружать электронные издания.
- Удалять отзывы Пользователей.
- Принимать запросы на добавление книг.
- Устанавливать ограничение на количество и срок бронирований.
- Получать данные о Пользователях, просрочивших дату возврата книги.

6 Требования к интерфейсу

Требуется разработать Web-интерфейс, ориентированный в первую очередь на использование на устройствах с небольшим размером экрана (смартфоны, планшеты).

Общие требования:

- Шрифт
 - Гарнитура без засечек, так как шрифты без засечек легче воспринимаются при чтении с электронных устройств.
- Цветовая гамма
 - Возможность выбора цветовой схемы интерфейса пользователем из предложенных вариантов. Ключевые элементы интерфейса должны быть контрастными.

Ключевые элементы интерфейса:

- Список изданий с возможностью поиска и фильтрации.
- История бронирований для каждого издания.
- Форма для добавления записи в читательскую карточку при взятии/возврате книги.

7 Прочие требования

7.1 Требования к надёжности

Основные требования к надёжности и целостности пользовательских данных выполняются за счет использования современной и безопасной СУБД — MySQL.

Рекомендуется минимум раз в 3 дня осуществлять резервное копирование базы данных, сохраняя копии во внешнем хранилище.

7.2 Требования к безопасности

Большинство уязвимостей предотвращается за счет возможностей используемого backend-фреймворка "Laravel". Его кодовая база защищена от угроз, например, SQL-инъекций и межсайтовых подделок запросов (CSRF). Это защищает пользователей от потери важных данных.

Также система устойчива к XSS-атакам. Данное требование выполняется за счет возможностей frontend-фреймворка "VueJS", который автоматически выполняет интерполяцию строк для пользовательских данных.

Соединение между пользователем и сервером проходит через протокол HTTPS. Он обеспечивает шифрование данных и делает их недоступными для просмотра посторонними.

7.3 Требования к производительности

Система должна обслуживать одновременно не менее 500 пользователей (каждый пользователь производит около 5 запросов в минуту).

При наличии стабильного интернет-соединения среднее время ответа сервера при осуществлении GET HTTP запроса не должно превышать 300мс.

При наличии стабильного интернет-соединения среднее время ответа сервера при осуществлении POST, DELETE, PATCH HTTP запросов не должно превышать 1с.

8 Проект

8.1 Средства реализации

8.1.1 Backend

1. PHP

Один из самых популярных языков программирования в сфере веб-разработки. Может работать в связке с различными серверами (Apache, Nginx, IIS).

В открытом доступе существует множество фреймворков и библиотек, написанных на PHP, что ускоряет скорость проектирования и разработки продукта.

2. Laravel

PHP-фреймворк с открытым исходным кодом, специально разработанный для создания сложных сайтов и веб-приложений.

Преимущества:

- Обширное сообщество
- Безопасность
- MVC архитектура
- Встроенное ORM для работы с данными

3. Eloquent

ORM-библиотека. Она облегчает взаимодействие с таблицами баз данных, обеспечивая объектно-ориентированный подход к выполнению таких операций, как вставка, обновление и удаление записей. Кроме того, Eloquent предоставляет упрощенный интерфейс для выполнения SQL-запросов.

4. PostgreSQL

Свободная объектно-реляционная система управления базами данных, основанная на языке SQL.

Преимущества:

- Поддержка БД неограниченного размера

- Высокая надёжность и хорошая производительность
- Поддержка транзакций (ACID), репликация реализована встроенными механизмами
- Наследование таблиц

5. Docker

Программное обеспечение, позволяющие автоматизировать развёртывание и управление приложением в средах с поддержкой виртуализации. Docker «упаковывает» приложение со всем окружением и зависимостями в контейнер.

6. PhpStorm

Интегрированная среда разработки на PHP с интеллектуальным редактором.

Преимущества:

- Подсветка синтаксиса, автодополнение кода расширенными параметрами форматирования
- Работа с детектором дублируемого кода, PHP рефакторинг
- Просмотр содержимого баз данных, выполнение SQL кода
- Интеграция с Git
- Интеграция с Docker

8.1.2 Frontend

1. Typescript

Язык программирования на основе JavaScript. Развивается компанией Microsoft как open-source продукт. Благодаря большому сообществу существует огромное количество библиотек, ускоряющих разработку приложений.

Строгая типизация обеспечивает консистентность данных, хранимых клиентским приложением. Реализует многие концепции ООП. Упрощает масштабирование и поддержку клиентских приложений.

2. Vue 3

Open-source фреймворк для разработки SPA приложений. Написан на языке TypeScript.

Преимущества Vue 3:

- Легковесность
- Гибкость
- Скорость и производительность.
- Использование компонентного подхода с использованием HTML - шаблонов.

3. InertiaJS

Библиотека для взаимодействия с backend. Позволяет организовать маршрутизацию, работу с данными и состоянием приложения на стороне сервера. Ускоряет разработку SPA на Vue, так как не требуется организовывать работу с данными на клиентской стороне с помощью хранилищ глобального состояния как Pinia или Vuex.

4. ZiggyJS

Библиотека для использования именованных Laravel-маршрутов. При использовании InertiaJS обеспечивает согласованность именования маршрутов на клиенте и сервере.

5. TailwindCSS

Tailwind CSS — это CSS-фреймворк, предлагающий обширный каталог классов и инструментов для облегчения стилизации сайта или приложения. Tailwind предлагает предварительно разработанные виджеты для создания веб-интерфейса, а не предварительно стилизованные адаптивные компоненты, что делает разработку интерфейса более гибкой.

6. DaisyUI

Плагин для TailwindCSS. Представляет из себя дизайн-систему, организованную в виде CSS-классов с использованием TailwindCSS классов и утилит.

8.2 Структура данных

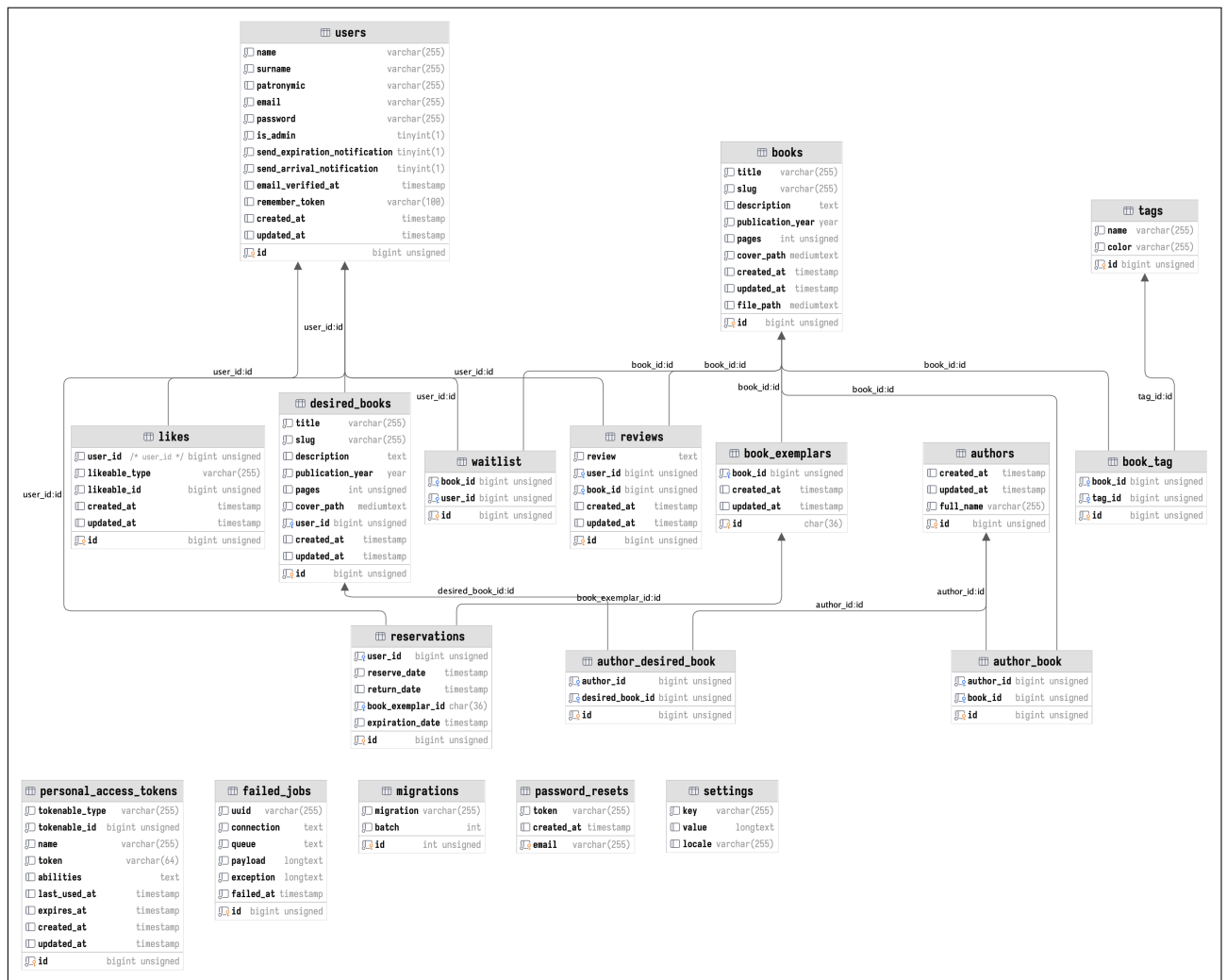


Рис. 4: Схема базы данных

8.3 Модули и алгоритмы

8.3.1 Backend

Серверная часть сервиса состоит из 12 модулей:

- **Controllers** – имеет три подмодуля, состоящих из наборов классов - контроллеров для обработки пользовательских запросов
 - **Auth** – функционал, связанный с авторизацией и аутентификацией.
 - **Books** – функционал, связанный с доменной областью сервиса.
 - **Admin** – функционал, доступный только для Администраторов.
- **Console** – набор классов для создания пользовательских команд (отправка уведомлений).

- Requests – схемы для валидации пользовательских запросов.
- Mail – набор классов для реализации функционала по отправке электронных писем.
- Models – набор классов для взаимодействия с базой данных с помощью EloquentORM.
- Policies – набор классов для разграничения прав доступа для различных сущностей базы данных.
- Rules – кастомные правила валидации.
- Services – набор классов, реализующих бизнес-логику сервиса. Каждой крупной модели соответствует свой класс-сервис.
- config – модуль, содержащий конфигурационные файлы.
- database – имеет три подмодуля:
 - factories – набор классов, определяющих фабрики для классов-моделей. Экземпляры классов заполняются случайными данными, описанными в схеме фабрики.
 - migrations – содержит файлы-миграции базы данных.
 - seeders – набор классов для наполнения базы данных тестовыми данными. Данный подмодуль использует подмодуль factories.
- lang – содержит файлы для локализации сервиса.
- routes – каталог, содержащий web и api-маршруты.

8.3.2 Frontend

Клиентская часть сервиса состоит из 6 модулей:

- components – включает 7 подмодулей:
 - Admin – компоненты для интерфейсов администратора.
 - Auth – формы аутентификации, регистрации.
 - Base – переиспользуемые внутри проекта UI компоненты.

- Books – компоненты, связанные с ключевой доменной областью сервиса.
 - Profile – компоненты, используемые на странице «Профиль».
 - Svg – иконки.
- layouts – компоненты, используемые для создания базовой структуры страницы.
 - pages – страницы сервиса.
 - types – объявления типов, интерфейсов, связанных с моделями данных на серверной части сервиса.
 - utils – функции-утилиты, используемые внутри компонентов.

8.5 Проект интерфейса

Цвета:

Цветовая схема интерфейса изменяется пользователем. На выбор представлены как светлые, так и темные варианты оформления.

Шрифт:

Для всего интерфейса выбран шрифт Jost. Это популярный шрифт без засечек, часто использующийся при разработке web-интерфейсов. В интерфейсе использованы начертания Bold (для заголовков), Regular и Light (для второстепенной информации).

На каждой странице находится меню с ссылками на ключевые разделы и переключателем цветовой схемы интерфейса. На смартфонах навигационное меню находится в верхней части экрана в виде «бургер-меню» (с возможностью сворачивания). На устройствах с большей диагональю экрана меню представлено в виде боковой панели в левой части страницы.

Интерфейс состоит из 10 страниц:

1. «Список книг»

Страница включает в себя список книг, представленных в виде карточек, включающих в себя обложку и основную информацию об издании, с указанием наличия электронной версии. Список разделен на страницы, переключение которых осуществляется при помощи кнопок,

расположенных в нижней части страницы. В верхней части страницы находится строка поиска, панель фильтрации, кнопка «Случайная книга».

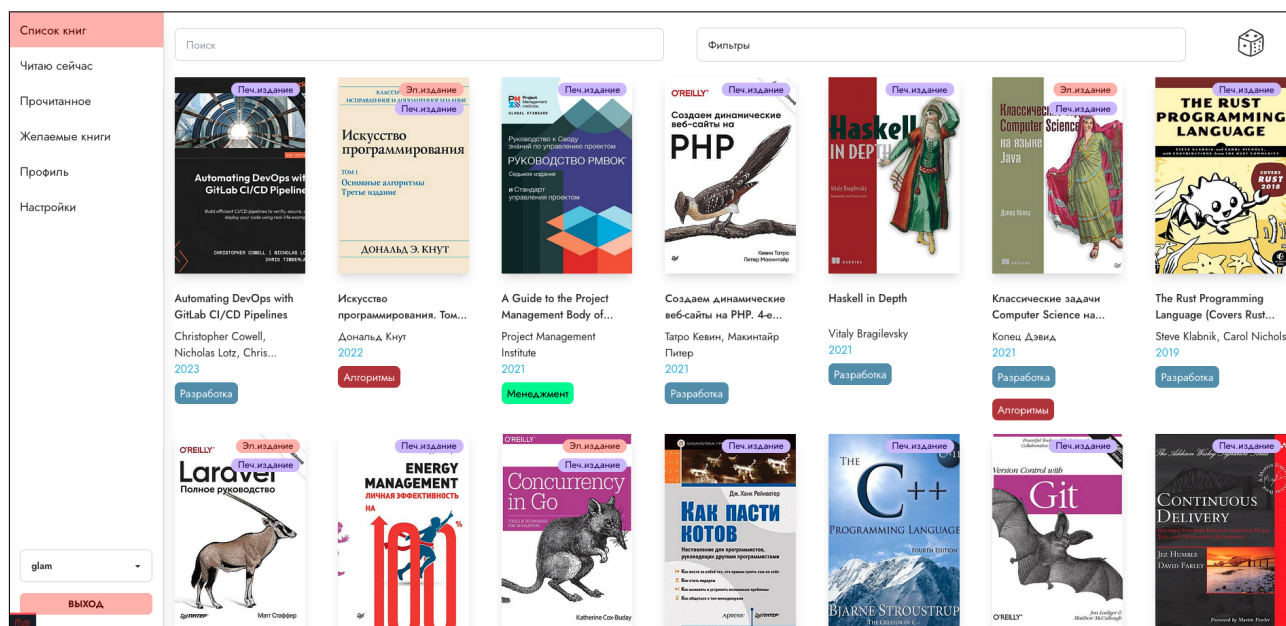


Рис. 5: Страница "Список книг"

2. «Читаю сейчас»

Страница включает в себя список книг, находящихся в текущий момент у Пользователя. Изданиям с истекшим сроком бронирования добавляется метка «Срок истек». Список книг выглядит как на рис. 5.

3. «Прочитанное»

Страница включает в себя список книг, прочитанных Пользователем. При большом количестве элементов списка он разбивается на страницы, в нижней части появляются кнопки для переключения текущей страницы. Список книг выглядит как на рис. 5.

4. Страница книги

На странице книги отображена более детальная информация: обложка, название, автор, описание, теги, год публикации и количество страниц. Пользователь может просмотреть историю бронирований и возвратов каждого из экземпляров книги, с указанием даты бронирования/возврата и ФИО Пользователей.

Если для книги загружено электронное издание, появляется ссылка на его скачивание.

Если выбранный экземпляр доступен для взятия, появляется кнопка «Взять», при нажатии на которую добавится запись в читательскую карточку и историю бронирований.

Если экземпляр книги в текущий момент закреплен за Пользователем, появляется кнопка «Вернуть», при нажатии на которую добавится соответствующая запись в читательскую карточку и историю бронирований. Книга станет доступной для бронирования.

Если экземпляр книги в текущий момент закреплен за другим Пользователем, появляется кнопка «В лист ожидания», при нажатии на которую Пользователь добавляется в очередь на взятие книги.

Если страницу просматривает Администратор, ему доступны кнопки редактирования и удаления книги, добавления и удаления экземпляров.

В нижней части страницы размещен раздел «Отзывы». Он состоит из формы для добавления отзыва и списка отзывов. Пользователь может удалить собственный отзыв, Администратор может удалять все отзывы.

Как пасти котов
Рейнвотер Ханк Дж.
Год публикации: 2015
Кол-во страниц: 256

Менеджмент

Описание

«Как пасти котов» — это книга о лидерстве и руководстве, о том, как первое совмещать со вторым. Это, если хотите, словарь трудных случаев управления IT-проектами. Программист подобен кошке, которая гуляет сама по себе. Так уж исторически сложилось. Именно поэтому так непросто быть руководителем команды программистов. Даже если вы еще месяц назад были блестящим и дисциплинированным программистом и вдруг оказались в роли менеджера, вряд ли вы знаете, с чего надо начать, какой выбрать стиль руководства, как нанимать и увольнять сотрудников, проводить совещания, добиваться своевременного выполнения задач. В таком случае без этой книги вам не обойтись. А может быть, вы — опытный менеджер, желающий пересмотреть свои принципы лидерства? Тогда, опять же, эта книга для вас. Вне зависимости от возраста, пола и социального статуса, она поможет вам укрепить свои позиции в роли лидера программистов. Материал изложен довольно компактно и легко укладывается в голову. Стоя в книжном магазине и раздумывая, что же купить, задайте себе один простой вопрос: «Нужно ли мне совершенствовать свои лидерские навыки?» Полагаю, вы ответите: «Да», — а значит, данная книга окажется для вас небесполезной.

Таймлайн

Экземпляр №1

Доступно

Отзывы

Оставьте свой отзыв

Опубликовать

Рис. 6: Страница книги

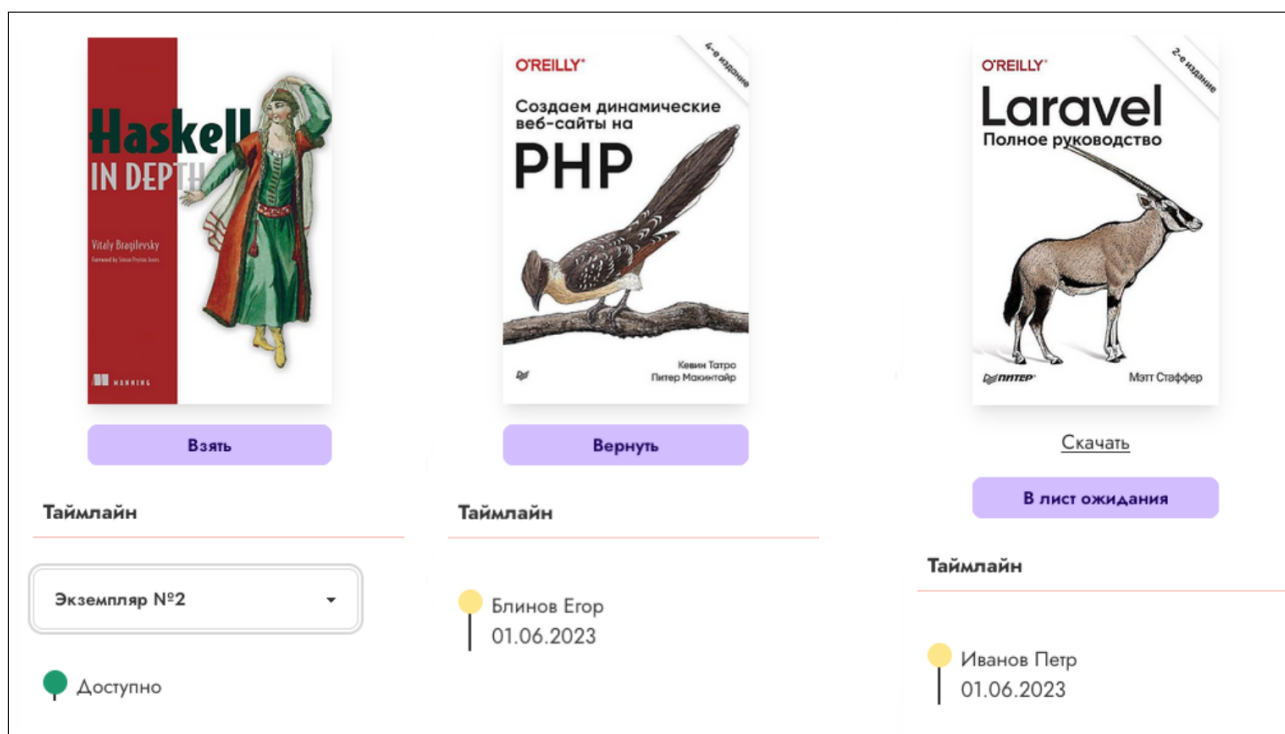


Рис. 7: Состояния книги: доступна, читаю сейчас, занята

5. «Желаемые книги»

На странице находятся книги, которые Пользователи хотели бы видеть в библиотеке. В верхней части экрана расположена поисковая строка и кнопка «Добавить книгу». Также все Пользователи могут голосовать за понравившиеся книги, используя кнопку, находящуюся на карточке издания. Так, ответственный за их закупку сможет отследить интересы и потребности коллег. При большом количестве элементов списка он разбивается на страницы, в нижней части появляются кнопки для переключения текущей страницы.



Рис. 8: Желаемые книги

6. Страница добавления книги

Доступна только для Администратора. Страница состоит из формы, включающей набор полей ввода информации о книге. На мобильных устройствах доступен сканер ISBN, для автоматизированного получения информации о книге с помощью Google Books API. При сохранении книга добавится в основной список книг.

9781449368845

ИСКАТЬ

Обложка

ОБЗОР... Файл не выбран.

Электронное издание

ОБЗОР... Файл не выбран.

Название

Learning Scala

Описание

Why learn Scala? You don't need to be a data scientist or distributed computing expert to appreciate this object-oriented functional programming language. This practical book provides a comprehensive yet approachable introduction to the language, complete with syntax diagrams, examples, and exercises. You'll start with Scala's core types and syntax before diving into higher-order functions and immutable data structures. Author Jason Swartz demonstrates why Scala's concise and expressive syntax make it an ideal language for Ruby or Python developers who want to improve their craft, while its type safety and performance

Год публикации

2014

Количество страниц

255

Количество экземпляров

2

Теги

Разработка X

Автор

Jason Swartz

+

СОХРАНИТЬ

Рис. 9: Страница добавления книги

7. Страница добавления книги в раздел «Желаемые книги»

Страница состоит из формы, включающей набор полей ввода инфор-

мации о книге. После сохранения книга будет отображаться в разделе «Желаемые книги». Интерфейс представлен на рис. 9.

8. Страница книги из раздела «Желаемые книги»

На странице книги отображена более детальная информация о запрашиваемой книге: обложка, название, автор, описание, год публикации и количество страниц. Также отображается ФИО человека, добавившего книгу. Администратору доступна кнопка «В список книг», которую следует нажимать после приобретения издания. Так, книга перестанет отображаться в разделе «Желаемые книги» и станет доступна для бронирования Пользователями в общем списке книг.

9. «Настройки»

Страница доступна только для Администратора. Доступны поля ввода для редактирования максимального срока бронирования и установления лимита бронирований книг Пользователями, интерфейс для редактирования списка тегов.

Также Администратор может просмотреть список всех текущих бронирований, список «должников».

<div>ДОБАВИТЬ КНИГУ</div> <div>ОТКРЫТЬ НАСТРОЙКИ</div> <div>РЕДАКТИРОВАТЬ СПИСОК ТЕГОВ</div>			
Фильтр			
Все			
Читатель	Книга	Дата бронирования	Дата окончания бронирования
Блинов Егор	<u>Создаем динамические веб-сайты на PHP. 4-е междуизд.</u>	01.06.2023	01.08.2023
Иванов Петр	<u>Laravel. Полное руководство. 2-е издание</u>	01.06.2023	01.08.2023

Рис. 10: Страница «Настройки»

10. «Профиль»

На странице доступны чекбоксы для получения уведомлений об истечении срока бронирований и новых книгах.

9 Реализация и тестирование

9.1 Backend

Объем кода, написанного на серверной стороне приложения - файлы с расширением «.php» - 248кб, или 7799 строк кода. Весь код был написан в рамках данной работы. Серверная часть состоит из 12 модулей. Реализованный функционал тестировался вручную.

9.2 Frontend

Объем кода, написанного на клиентской стороне: файлы с расширением «.vue» - 428кб, или 4575 строк кода; файлы с расширением «.ts» - 40кб, или 413 строк кода. Количество разработанных страниц - 12, компонентов - 29. Весь код был написан в рамках данной работы. Реализованный функционал тестировался вручную.

Заключение

В процессе проектной работы были углублены знания в области проектирования веб-приложений и баз данных, разработке интерфейсов. Был разработан и развернут веб-сервис, успешно интегрированный во внутренние сервисы компании Farpost.

Таким образом, результатом работы является полностью работоспособное веб-приложение для корпоративной библиотеки.

Список литературы

- [1] *1C: Бибlioтека*. URL: <https://solutions.1c.ru/catalog/library>.
- [2] *20 Best Library Management Software in 2023*. URL: <https://research.com/software/best-library-management-software>.
- [3] *Alexandria: The Highly-Customizable ILS*. URL: [Alexandria:%20The%20Highly-Customizable%20ILS](#).
- [4] *Alfa Ebooks Manager*. URL: <https://www.alfaebooks.ru/>.
- [5] *All My Books*. URL: https://www.bolidesoft.com/rus/allmybooks/uchet-knig.html?__c=1.
- [6] *Apollo ILS: The Public Library Specialist*. URL: <https://www.biblionix.com/products/apollo-ils-only-for-public-libraries/>.
- [7] Public Library Association. *2020 public library technology survey*. URL: <https://www.ala.org/pla/sites/ala.org.pla/files/content/data/PLA-2020-Technology-Survey-Summary-Report.pdf>.
- [8] *Atrium: Powerful Web-Based Library System*. URL: <https://booksys.com/atrium>.
- [9] *Biblio STOR-M*. URL: <https://bibliostorm.ru/>.
- [10] M Breeding. "Discoverability of library collections". B: Library Technology Guides (2022). URL: <https://librarytechnology.org/document/27289/discoverability-of-library-collections>.
- [11] M Breeding. "Integrated library system statistics". B: (2002). URL: <https://doi.org/10.5860/ltr.38n3>.
- [12] *Calibre*. URL: <https://calibre-ebook.com/>.
- [13] *CodeAchi Library Management System*. URL: <http://codeachi.com/product/library-management-system/>.
- [14] *Drupal SOPAC*. URL: <https://www.drupal.org/project/sopac>.
- [15] *DSpace*. URL: <http://dspace.org/>.
- [16] EBSCO. *2021 trends in library analytics*. 2021. URL: <https://www.ebsco.com/blogs/ebscopost/2021-trends-library-analytics>.
- [17] *ELiS*. URL: <https://lib.elibsystem.ru/>.

- [18] *Evergreen ILS*. URL: <https://evergreen-ils.org/>.
- [19] *Evolve Integrated Library Software*. URL: <https://www.infovisionsoftware.com/products/library>.
- [20] *Follett Destiny Library Manager*. URL: <https://www.follettlearning.com/education-technology/library-management-system>.
- [21] *Genesis G4*. URL: <https://www.lrms.com/>.
- [22] *Greenstone*. URL: <https://www.greenstone.org/>.
- [23] *Handy Library Manager*. URL: <https://www.primasoft.com/librarymanager/handylibrary.htm>.
- [24] *Infiniti Management Software: Custom Library Management Platform*. URL: <https://psp-assetprotection.co.uk/solutions/library-management-software/>.
- [25] K Kishore. “How much does it cost to create a library management system?” B: *Octal bog* (2021). URL: <https://www.octalsoftware.com/blog/library-management-system/>.
- [26] *KnowAll Matrix*. URL: <https://knowallmatrix-librarysoftware.com/>.
- [27] *Koha*. URL: <https://koha-community.org/download-koha/>.
- [28] *Libero: A Platform for Every Library Type*. URL: <https://libero.com.au/>.
- [29] *Libib*. URL: <https://www.libib.com/>.
- [30] *LibraryWorld*. URL: <http://libraryworld.com/>.
- [31] T Mageto. “Design and development of e-library system: COVID-19 pandemic challenges.” B: *Journal of Computer Sciences and Applications* (2021). URL: <http://article.computersciencesapplications.com/pdf/jcsa-9-1-1.pdf>.
- [32] *Mandarin M5: Customizable Library Automation Software*. URL: <https://www.mlasolutions.com/products/m5>.
- [33] *MyLib*. URL: <https://www.bestfree.ru/soft/office/accounting-books.php>.

- [34] L.A. Ogunsola. "The next step in librarianship: Is the traditional library dead." B: *Library Philosophy and Practice*, 1, 42 (2011). URL: <https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1638&context=libphilprac>.
- [35] *OPAC-Global*. URL: <http://www.ditm.ru/opac-global>.
- [36] *Rescarta*. URL: <http://www.rescarta.org/>.
- [37] Adroit Market Research. *Global library management software market analysis*. 2021. URL: <https://www.adroitmarketresearch.com/industry-reports/library-management-software-market>.
- [38] *ResourceMate: Flexible Library Management*. URL: <https://www.resourcemat.com/>.
- [39] *Surpass: An Affordable Library Management System*. URL: <https://surpasssoftware.com/>.
- [40] *SydneyEnterprise: ILS Platforms*. URL: <https://lucidea.com/sydneyenterprise/>.
- [41] *WorldShare Management Services*. URL: <https://www.oclc.org/en/worldshare-management-services.html>.
- [42] *Xerox ПЭБ*. URL: <https://www.xerox.ru/press-centre/release/1158177/>.
- [43] *АБИС «ИРБИС»*. URL: http://www.elnit.org/index.php?option=com_content&view=article&id=35&Itemid=108.
- [44] *АБИС «Марк»*. URL: <http://www.informsystema.ru/ru/content/MarcCloud>.
- [45] *АБИС «Руслан-Нео»*. URL: <https://obs.ruslan.ru/products/abis-ruslan-neo/>.
- [46] *АБИС Лубра*. URL: <http://www.bibliosoft.ru/download.html>.
- [47] *АИС Библиотека*. URL: <https://xn--90abhbolvbbfgb9aje4m.xn--p1ai/avtomatizirovannaya-sistema-biblioteka-uchet-knig/>.
- [48] *Для учёта книг в библиотеке*. URL: <https://www.softportal.com/software-15422-dlya-ucheta-knig-v-biblioteke.html>.
- [49] *ИнфоРост*. URL: <http://elib.shpl.ru/>.

- [50] *Либэр*. URL: <https://www.libermedia.ru/>.
- [51] *МегаПро*. URL: <http://www.data-express.ru/aibc-megapro/>.
- [52] *Учёт книг*. URL: <http://www.simple-soft.ru/BooksCount.htm>.