Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизации обработки информации (АОИ)

ВЕБ-САЙТ «ЭЛЕКТРОННАЯ БИБЛИОТЕКА»

Отчёт к проекту по дисциплине  
Методология и технология программирования

|  |  |  |
| --- | --- | --- |
|  |  | Выпонили  Студент гр. 429-1    В. А. Хиль  (подпись)    А. В. Зарипов  (подпись)  « » 2021 г.  (дата) |
|  | (оценка) | Руководитель:  старший преподаватель АОИ  С. С. Пекарская  (подпись)  « » 2021 г.  (дата) |

Томск 2021

# Оглавление

[1 Постановка задачи 3](#_Toc73529476)

[2 Анализ идеи веб-сайта 4](#_Toc73529477)

[3 Описание требуемых функциональных возможностей 5](#_Toc73529478)

[4 Модель предметный области 6](#_Toc73529479)

[5 Эскиз интерфейса 7](#_Toc73529480)

[6 Варианты использования 10](#_Toc73529481)

[7 Описание требуемых функциональных возможностей веб-сайта 15](#_Toc73529482)

[8 Используемые frontend-решения 16](#_Toc73529483)

[9 Используемые backend-решения 17](#_Toc73529484)

[10 Описание работы созданного веб-сайта 18](#_Toc73529485)

[11 Заключение 21](#_Toc73529486)

# Постановка задачи

Разработать веб-сайт библиотеки электронных книг, с возможностью добавлять и удалять книги, редактировать описание хранимых книг.

# Анализ идеи веб-сайта

Для начала проведём анализ идеи веб-сайта. Для этого воспользуемся Lean Model Canvas, изображённой на рисунке 1.



Рисунок 1 – Анализ идеи приложения

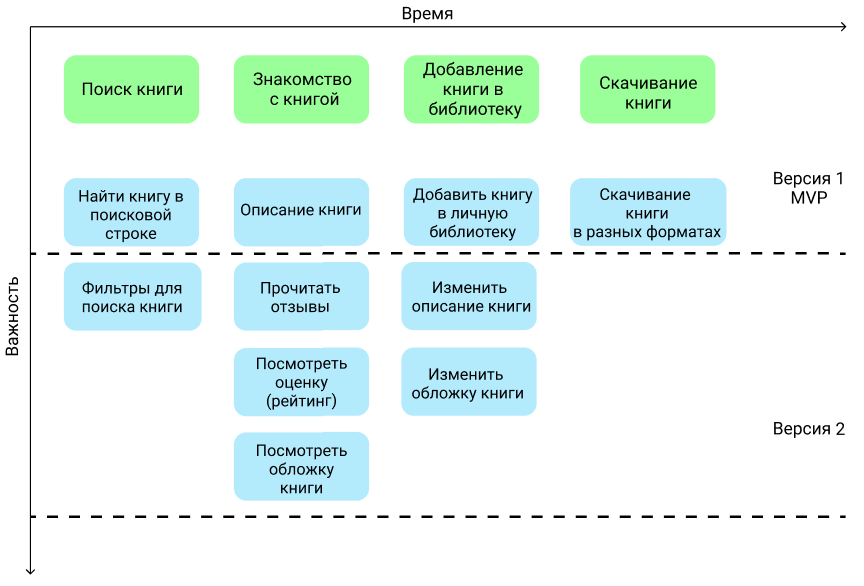
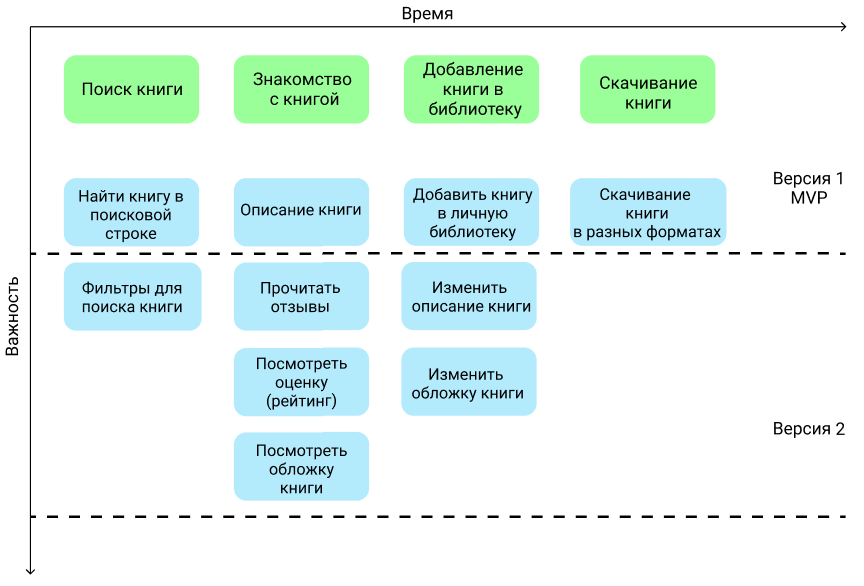
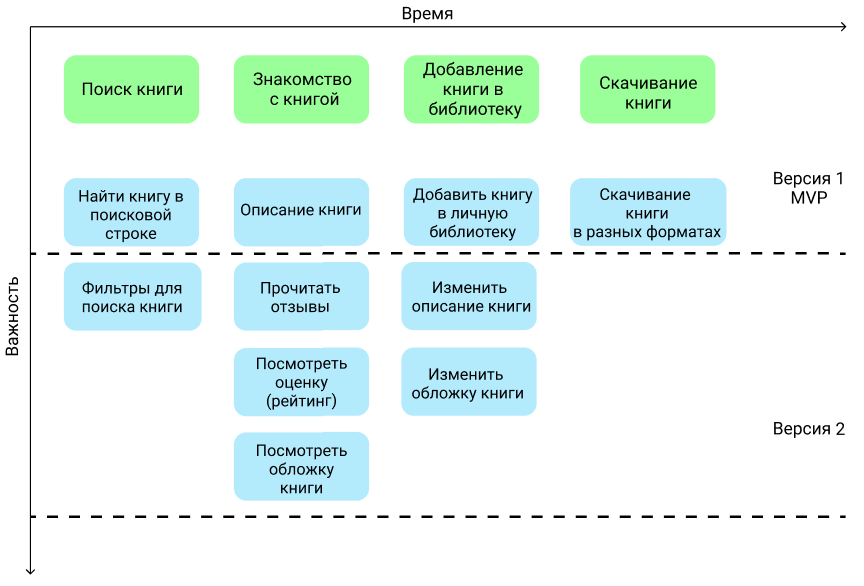
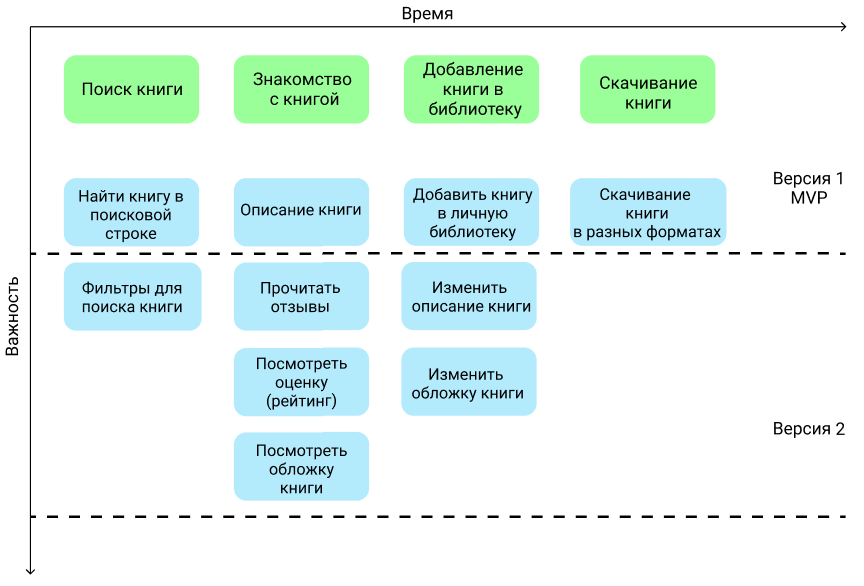
Для начала нужно решить, кому необходима электронная библиотека, то есть определить сегменты заказчиков. Электронной библиотекой пользуются школьники и студенты, учителя и преподаватели, а также просто любители книг. Проблемы, с которыми могут столкнуться наши заказчики, следующие:

* скачивание электронных книг в разных форматах;
* добавление электронной книги в личную библиотеку;
* помощь подбора книг.

Для того чтобы решить данные проблемы, на нашем сайте мы будем хранить книги в разных форматах (DOCX, FB2, PDF), введём оценку книг, то есть рейтинг, и добавим возможность пользователям оставлять отзывы на книги, а также при скачивании рекомендовать книги. Особенностью нашего сайта будет система достижений, которые пользователи будут получать за скачивание книг и качество отзывов. Пользователи могут найти наш сайт через поисковые системы и с помощью реферальных ссылок. Также мы будем вести статистику перешедших к нам пользователей через поисковые запросы и реферальных ссылок.

# Описание требуемых функциональных возможностей

Для описания требуемых функциональных возможностей веб-сайта и порядка их реализации составим User Story Map (USM), представленной на рисунке 2.



Авторизация и регистрация

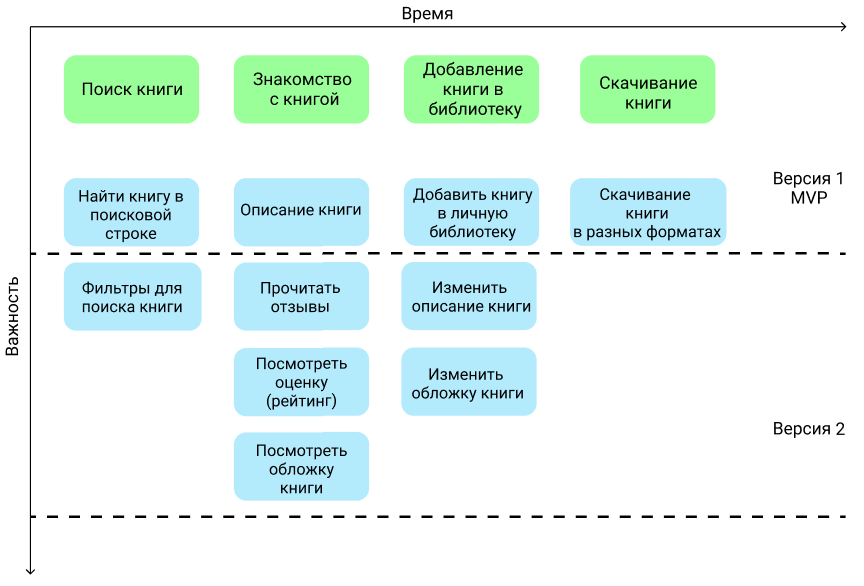


Рисунок 2 – User Story Map

На USM представлено две версии веб-сайта. Первая версия MVP – минимальный жизнеспособный продукт и вторая версия с добавлением новых функций. Мы разработаем первую версию. В неё входит следующие функции:

* поиск книги в поисковой строке;
* описание книги
* добавление книги в общую библиотеку (функция администратора);
* скачивание книги в разных форматах;
* авторизация и регистрация пользователей.

# Модель предметный области

Также составим модель предметной области, то есть создадим UML диаграмму, которые отражают различные взаимосвязи между сущностями предметной области, такими как объекты и подсистемы, а также описывают их внутреннюю структуру и типы отношений. Модель предметной области представлена на рисунке 3.

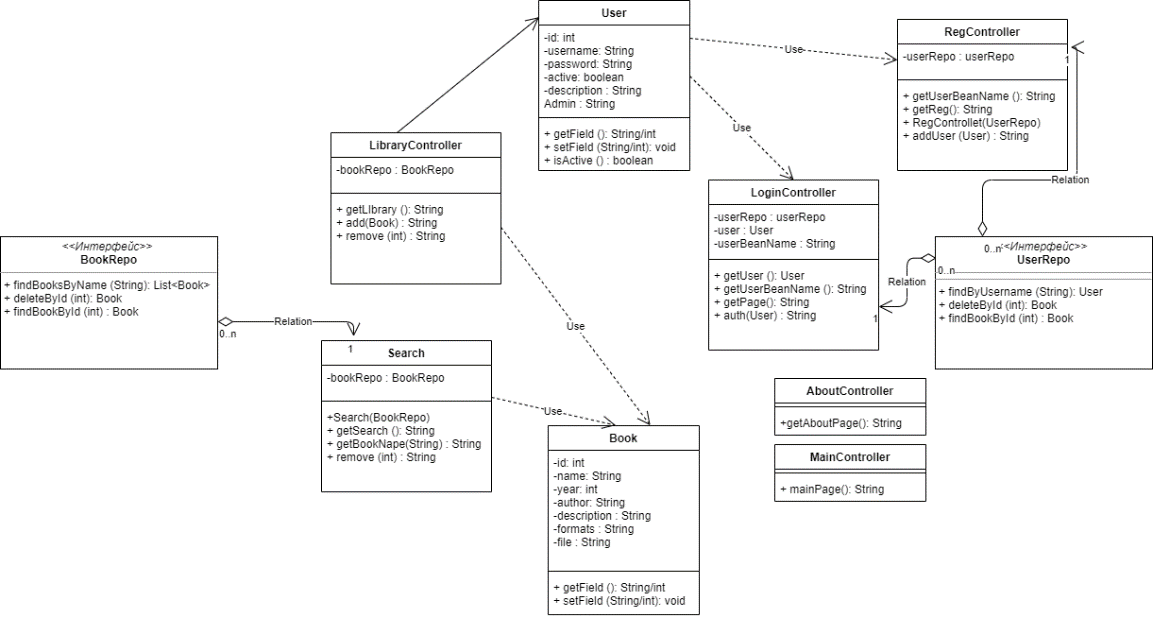


Рисунок 3 – UML-диаграмма предметной области

UserRepo и BookRepo – это контейнеры данных, из них берутся данные другими классами. Класс Book описывает сущность книги. С ним связаны с такими классами, как Search (отвечает за поиск книги и отображение страниц в поиске) и LibraryController (отвечает за отображение библиотеки и скачивание книг). UserRepo связан с классами LoginController и RegController, которые отвечают за авторизацию и регистрацию пользователей соответственно. MainController и AboutController нужны для отображения гланой страницы и страницы с информацией о разработчиках веб-сайта.

# Эскиз интерфейса

Переходим к разработке эскиза интерфейса. Интерфейс состоит из нескольких страниц с одинаковой структурой, представленная на рисунке 4.

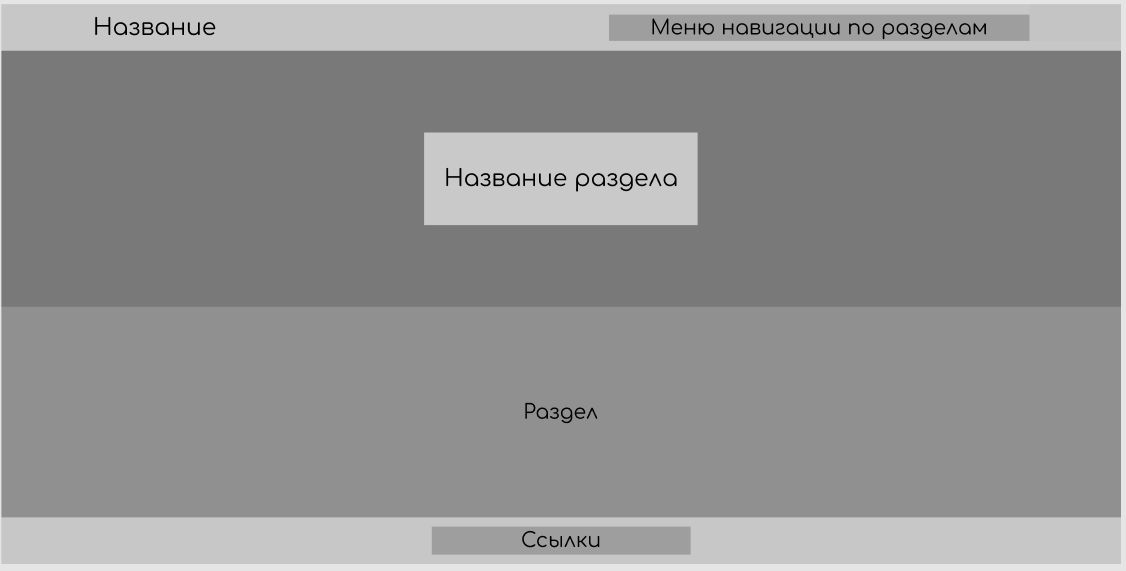


Рисунок 4 – Эскиз сайта

В каждой странице есть шапка, содержащая в себе панель навигации, футер, содержащая в себе ссылки на страницы разработчиков в социальных сетях, и две зоны. Одна зона содержит в себе название страницы, а вторая зона отвечает за содержание страницы. Страницы различаются только этими двумя зонами. Рассмотрим вторую зону для каждой страницы отдельно.

Главная страница содержит в себе только зону с названием сайта.

Страница «Библиотека» содержит ячейки, состоящие из названия, автора, года издания, описания, издания и ссылки для скачивания книги. Изобразим одну ячейку для наглядного примера (Рисунок 5).



Рисунок 5 – Часть раздела “Библиотека”

Страница «Поиск» похожа на предыдущую страницу, только с добавлением панели для ввода и кнопки (Рисунок 6).

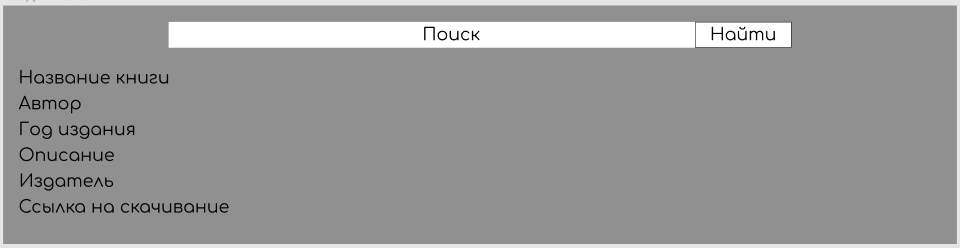


Рисунок 6 – Часть раздела “Поиск”

Страница «О нас» содержит ячейки с информацией о разработчиках (Рисунок 7).



Рисунок 7 – Часть раздел “О нас”

Страница «Авторизация» состоит только из одной зоны и содержит в себе два поля для ввода логина и пароля и двух кнопок «Войти» и «Регистрация» (Рисунок 8). Страница «Регистрация» содержит точно такие же поля и кнопки.

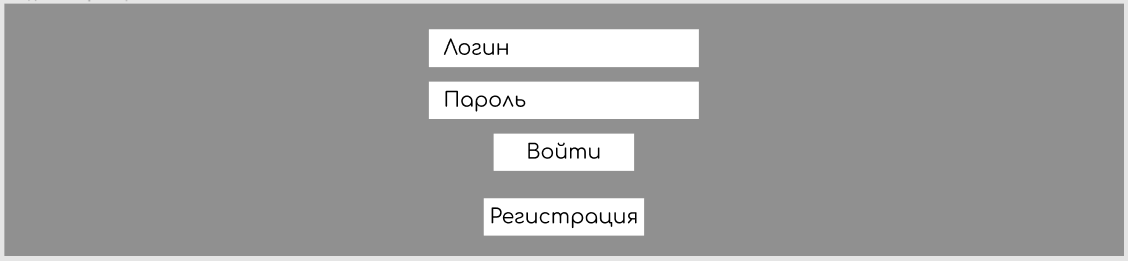


Рисунок 8 – Раздел «Авторизация»

У администратора различия есть только в странице «Библиотека». Там добавятся поля для ввода информации о книге, а также поле для добавления ссылки на скачивание электронной книги (Рисунок 9).

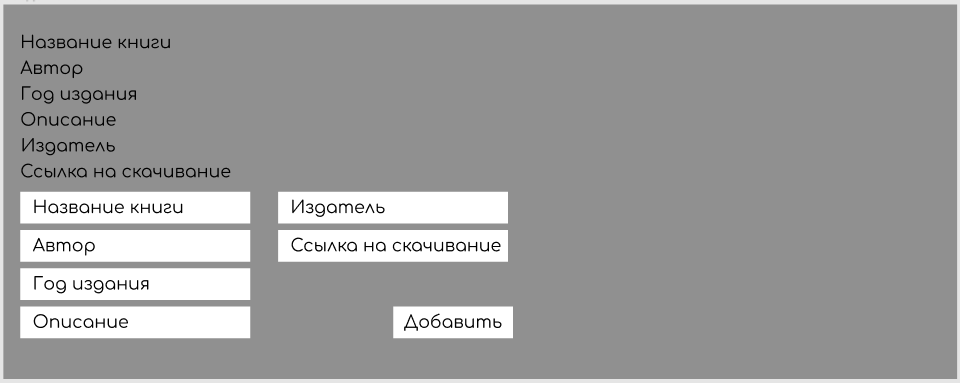


Рисунок 9 – Раздел «Библиотека» для администратора

# Варианты использования

Составим диаграмму вариантов использования, представленной на рисунке 10.

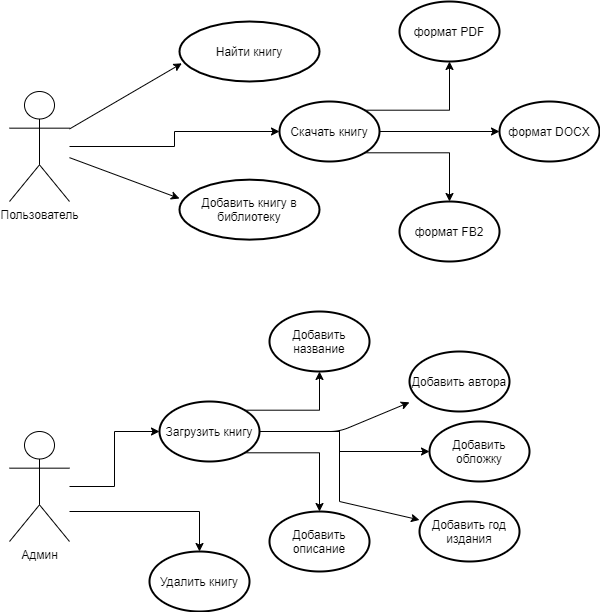


Рисунок 10 – Диаграмма вариантов использования

Опишем варианты использования через Use Cases:

* Авторизация

1. Имя: авторизоваться на сайте.

2. Контекст использования: не авторизованный пользователь авторизуется на сайте, чтобы сайт его узнал и показывал персональную для него информацию: для пользователя библиотеку книг, для администратора библиотеку и форму для добавления книг.

3. Область действия: взаимодействие клиента с приложением.

4. Уровень: цель пользователя.

5. Основное действующее лицо: пользователь.

6. Предусловие: посетитель должен быть не авторизован.

7. Минимальные гарантии: посетитель узнает факт успешной или неуспешной попытки авторизации.

8. Постусловие: посетитель авторизовался.

9. Триггер: пользователь нажимает кнопку «Авторизоваться».

10. Основной сценарий:

1) Пользователь запускает авторизацию.

2) Система отображает форму с авторизацией.

3) Клиент вводит логин и пароль.

4) Система ищет и подтверждает пользователя с таким логином и совпадение пароля и не превышение количества попыток входа в аккаунт.

5) Система авторизует клиента главную страницу сайта.

11. Расширения:

5.а. Аккаунта с данным логином не обнаружено:

5.а.1. Переход к шагу 2, либо переход к сценарию «Регистрация пользователя».

5.б. Пароль от аккаунта с данным логином не совпадает с введенным.

5.б.1. Переход к шагу 2.

* Регистрация

1. Имя: зарегистрироваться на сайте

2. Контекст использования: пользователь создает новый аккаунт для авторизации на сайте.

3. Область действия: взаимодействие клиента с приложением.

4. Уровень: цель пользователя.

5. Основное действующее лицо: пользователь.

6. Предусловие: посетитель должен быть не авторизован.

7. Минимальные гарантии: пользователь узнает факт успешной или неуспешной регистрации.

8. Постусловие: посетитель зарегистрировался.

9. Триггер: пользователь нажимает кнопку «Зарегистрироваться».

10. Основной сценарий:

1) Пользователь запускает регистрацию.

2) Система отображает форму с регистрацией.

3) Клиент вводит логин и пароль.

4) Система ищет и подтверждает пользователя с таким логином не существует.

5) Система регистрирует и авторизует пользователя и перемещает на главную страницу сайта.

6) Система выводит сообщение об успешной авторизации.

Расширения:

5.а. Аккаунт с данным логином уже существует:

5.а.1. Переход к шагу 2, либо переход к сценарию «Регистрация пользователя».

* Скачивание книг

1. Имя : скачивание книги

2. Контекст использования: пользователь скачивает книгу для дальнейшего использования.

3. Область действия: взаимодействие клиента с приложением.

4. Уровень: цель пользователя.

5. Основное действующее лицо: пользователь.

6. Предусловие: посетитель должен находиться на сайте.

7. Минимальные гарантии: пользователь переходит на гугл диск с разными форматами нужной книги.

8. Постусловие: посетитель скачивает книгу.

9. Триггер: пользователь нажимает кнопку «Скачать».

10. Основной сценарий:

1) Пользователь запускает окно библиотеки.

2) Система отображает книги из базы данных.

3) Клиент нажимает кнопку «Скачать».

4) Система переводит клиента на гугл диск с нужной его книгой.

5) Пользователь скачивает книгу.

* Добавление книг

1. Имя : добавление книги

2. Контекст использования: администратор добавляет книгу .

3. Область действия: взаимодействие клиента с приложением.

4. Уровень: цель пользователя.

5. Основное действующее лицо: администратор

6. Предусловие: посетитель быть авторизован, как администратор.

7. Минимальные гарантии: пользователь заходит на форму добавления книги.

8. Постусловие: пользователь добавляет книгу.

9. Триггер: пользователь нажимает кнопку «Добавить книгу».

10. Основной сценарий:

1) Пользователь запускает окно библиотеки.

2) Система проверяет пользователя на наличие прав администратора.

3) Система отображает форму добавления книги в библиотеку.

4) Пользователь заполняет форму данными о книге.

5) Пользователь нажимает кнопку «Добавить книгу».

6) Система добавляет книгу в библиотеку.

* Поиск книги

1. Имя : поиск книги

2. Контекст использования: пользователь ищет книгу в библиотеке по названию.

3. Область действия: взаимодействие клиента с приложением.

4. Уровень: цель пользователя.

5. Основное действующее лицо: пользователь

6. Предусловие: пользователь узнал наличие нужной книги.

7. Минимальные гарантии: пользователь заходит на форму добавления книги.

8. Постусловие: пользователь находит книгу.

9. Триггер: пользователь нажимает кнопку «Поиск».

10. Основной сценарий:

1) Пользователь запускает окно поиска.

2) Система отображает форму поиска книги.

3) Пользователь вводит название книги.

4) Пользователь нажимает кнопку «Найти книгу».

5) Система проверяет наличие книги в базе данных.

6) Система высвечивает найденную книгу на экране.

Расширения:

6.а Такой книги нет

6.а.1 переход к шагу 2.

# Описание требуемых функциональных возможностей веб-сайта

Опишем функциональные возможности нашего веб-сайта. Для того чтобы зарегистрироваться на сайте, пользователю нужно войти на форму регистрацию, ввести логин и пароль, и тогда его логин и пароль внесутся в базу данных. Для того чтобы войти в аккаунт, пользователю нужно войти на форму авторизации и ввести свой логин и пароль. После пройдёт проверка соответствие введённых логина и пароль с тем, что находится в базе данных. После успешной проверки пользователь зайдёт в аккаунт. Для поиска книги пользователь заходит на страницу «Поиск», вводит в форме название книги. После нажатии кнопки происходит поиск книги с таким же названием в базе данных. Если такая книга найдена, то эта книга появляется на странице. Для добавления электронной книги администратор заходит в свой аккаунт и переходит на страницу «Библиотека». Только у администратора высвечиваются поля для ввода информации о добавляемой книге. После внесения всех данных, эти данные отправляются на базу данных, и книга появляется на странице «Библиотека». Удаление книги происходит посредством нажатия на кнопку «Удалить».

# Используемые frontend-решения

Для реализации фронтенда сайта мы использовали следующий инструментарий:

* HTML как язык разметки;
* CSS как язык описания внешнего вида документа;
* JavaScript как язык программирования для обеспечения функциональности на стороне клиента.

Сайт содержит только одну функцию JavaScript для скрытия панели меню навигации за кнопкой, если размер окна браузера уменьшить до определённых размеров или для входа на сайт через мобильные устройства.

# Используемые backend-решения

Для реализации бэкенда сайта мы использовали следующий инструментарий:

* Java как основной язык программирования;
* Spring как фреймворк для работы с бэкендом;
* Thymeleaf как шаблонизатор;
* PostgreSQL как СУБД для работы с базой данных.

Для реализации веб-сайта был выбран JavaSpring, потому что у него присутствует удобный шаблонизатор и есть возможность вынесения конфигурации приложения в отдельном файле отдельно от программного кода. PostgreSQL удобен для работы с базой данных, у него есть поддержка неограниченного размера базы данных.

# Описание работы созданного веб-сайта

При входе на сайт пользователь попадает на главную страницу. Из главной страницы пользователь может авторизоваться, нажав на кнопку «Авторизация», находящейся на панели меню навигации (Рисунок 11).

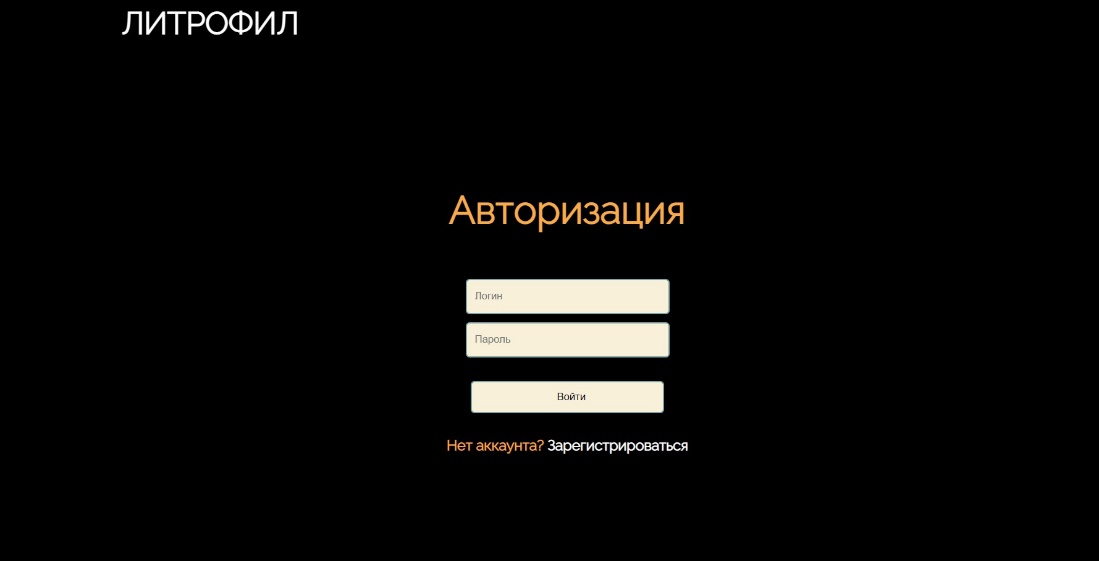


Рисунок 11 – Страница «Авторизация»

Зайдя на страницу авторизации, пользователь может перейти на страницу регистрации, нажав на соответствующую кнопку (рисунок 12).

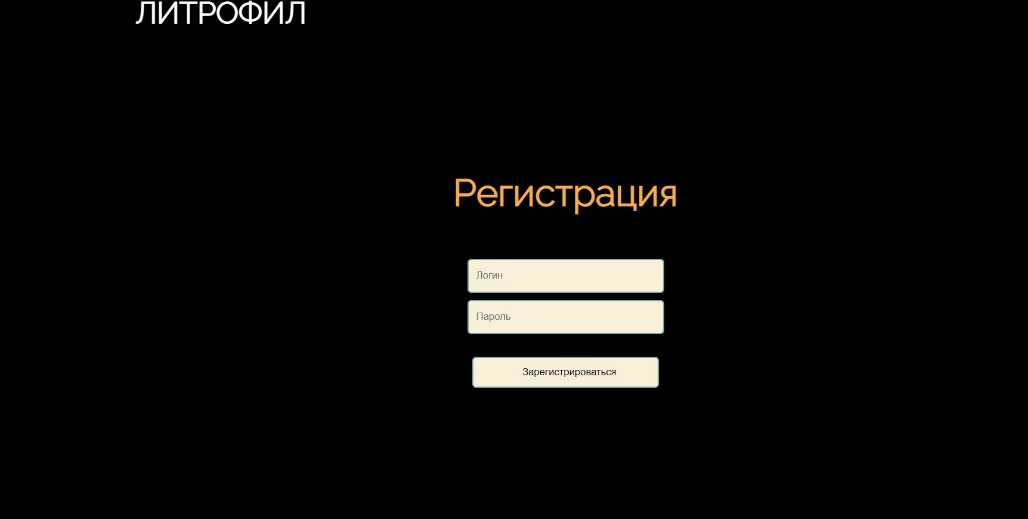


Рисунок 12 – Страница «Регистрация»

После успешной регистрации и авторизации пользователь возвращается на главную страницу (Рисунок 13).

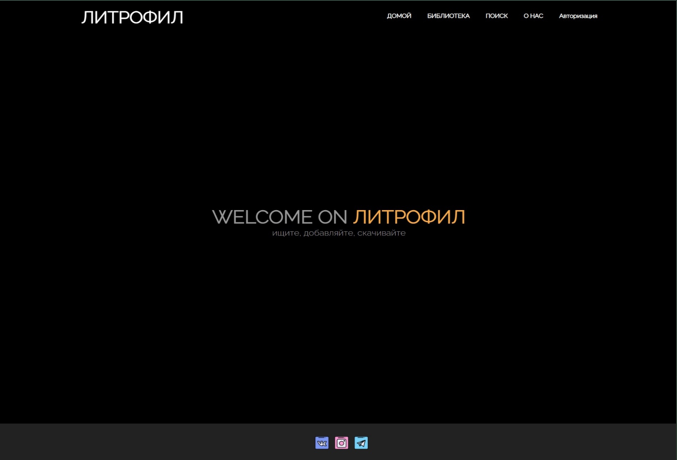


Рисунок 13 – Главная страница

С главной страницы пользователь может перейти на страницу «Библиотека», с которой он может скачать книгу из всей базы книг (рисунок 14). После нажатии на кнопку «Скачать» пользователь перейдёт на Google диск, где хранятся файлы со всеми форматами нужной книги.

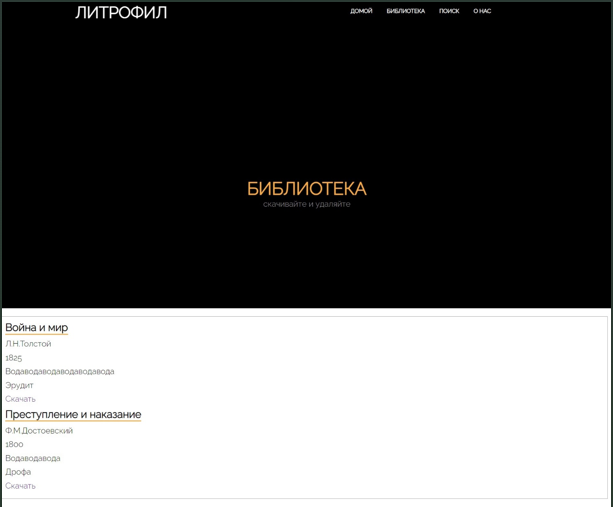


Рисунок 14 – Страница «Библиотека»

Перейдя на страницу «Поиск», пользователь может ввести название книги и при успешном поиске найти определённую книгу (Рисунок 15).

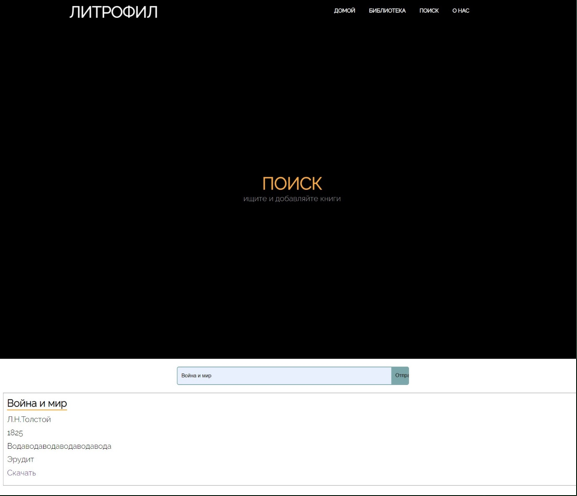


Рисунок 15 – Страница «Поиск»

При открытии страницы «О нас» пользователь сможет прочитать информацию о разработчиках (Рисунок 16).

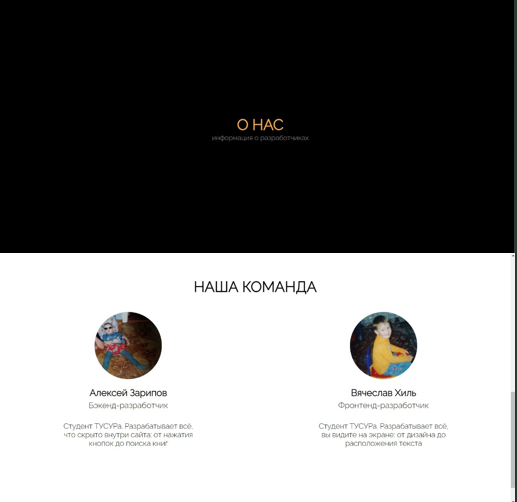


Рисунок 16 – Страница «О нас»

# Заключение

В результате проекта был реализован веб-сайт «Электронная библиотека».

В ходе выполнения проекта были изучены основы работы с фронтедом, а именно с языком разметки HTML, языком стилей CSS и языком программирования Javascript; с бэкендом, а именно с языком программирования Java, фреймворком Spring, шаблонизатором Thymeleaf и базой данных PostgreSQL. Был проведён анализ идеи, а также разработаны варианты использования и модель предметной области.