

МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)**

Факультет *Компьютерных наук*

Кафедра *программирования и информационных технологий*

Агрегатор аниме “Shiki-Adapter”

Курсовой проект

09.03.04 Программная инженерия

Преподаватель _____ В.С. Тарасов __.__.20__

Обучающийся _____ В.А. Бойченко, 3 курс

Обучающийся _____ Д.С. Арапов, 3 курс

Обучающийся _____ Б.Ю. Есин, 3 курс

Воронеж 2022

Содержание

Содержание	2
Введение	3
1 Постановка задачи	4
2 Глоссарий.....	5
3 Анализ предметной области	7
3.1 Анализ существующих решений	7
3.2 Анализ целевой аудитории.....	8
3.3 Пользователи системы	8
4 Анализ задачи.....	10
4.1 Варианты использования системы	10
4.2 Общая структура системы	11
4.3 Взаимодействие компонентов системы	12
4.4 Варианты состояния системы	16
4.5 Развертывание системы	17
4.6 IDEF0	18
4.7 Диаграмма классов	18
4.8 ER-диаграмма.....	21
5 Анализ средств реализации	22
5.1 Frontend	23
5.2 Backend.....	23
6 Воронки конверсии.....	24
6.1 Поиск.....	26
6.2 Просмотр аниме	26
6.3 Переход на страницу списков.....	27
7 Тестирование	27
7.1 Дымовое тестирование	27
7.2 UI тесты	28
7.3 Юзабилити тесты	30
8 Используемые технологии	32
Заключение	34
Список использованных источников	35

Введение

Приложение является агрегатором тематических видеоматериалов, а именно японской мультипликационной анимации, упрощает процесс поиска, отслеживания и просмотра выхода новых серий. Основная функциональность сайта должна включать в себя:

- возможность просмотра аниме из различных видеоисточников, используя одну страницу с информацией о произведении;
- возможность поиска аниме;
- возможность создания списков “Запланировано”, “Смотрю”, “Просмотрено”, “Брошено” для сортировки произведений;
- возможность мониторинга новых серий в списке “Смотрю”, используя автоматически обновляемый список “Обновления”;
- интеграцию с порталом “Shikimori”.

Целью нашей командной работы является создание проекта, удовлетворяющего всем вышеперечисленным условиям и облегчающим просмотр азиатских мультипликационных произведений.

1 Постановка задачи

Данный проект является полуавтоматическим агрегатором для упрощения просмотра азиатской мультипликации.

Система автоматизирует процесс поиска видеоматериалов для различных произведений азиатской анимации, предоставляет возможность сортировать аниме по нескольким группам интереса, а также предоставляет возможность отслеживания обновлений по недавно вышедшим мультипликационным материалам.

Система предназначена для:

- упрощения процесса отслеживания новых серий;
- агрегации видеоматериалов для произведений в едином пространстве (странице с аниме);
- сортировки мультипликации по группам интереса;
- Упрощения доступа к видеоматериалам произведений, которые пользователь поместил в свою группу интересов на портале “Шикимори”.

Для достижения данной цели были выделены следующие подзадачи:

- разработка дизайна веб-приложения;
- разработка Frontend части сервиса;
- разработка базы данных;
- разработка Backend части сервиса;
- создание связи между Frontend и Backend частями приложения;
- интеграция инструментов веб-аналитики (метрик).

2 Глоссарий

- **Веб-сервис, интернет-сервис, веб-приложение, система** – идентифицируемая уникальным URL-адресом программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя;
- **Сервер, серверная часть** – компьютер, обслуживающий другие компьютеры (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач;
- **Клиент, клиентская сторона** – компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой;
- **Front-end** – клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса;
- **Back-end** – программно-аппаратная часть сервиса;
- **REST API** – это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов;
- **GitHub** – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки;
- **Header** – визуальный элемент, расположенный в верхней части страницы;
- **Footer** – визуальный элемент, расположенный в нижней части страницы;
- **Гость** – неавторизованный на веб-сервисе человек, пользующийся ограниченным функционалом веб-сервиса;
- **Пользователь** – авторизованный на портале человек, пользующийся функционалом веб-сервиса;
- **Модератор** – Административное лицо сервиса, которое действует от лица представителя веб-ресурса и выполняет определенный ряд задач, направленных на поддержание работы сайта;
- **Контент** – наполнение сайта;

- **Аниме** – японская мультипликация;
- **Shikimori (Шикимори)** – одна из крупнейших онлайн энциклопедий аниме;
- **Фреймворк** – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта;
- **Онгоинг** (от англ. on going) – сериал, серии которого выходят в данный момент;
- **Манга** – японские комиксы;
- **Ранобэ** – популярная японская литература различных жанров;
- **Тайтл** – собственно азиатское мультипликационное произведение.

3 Анализ предметной области

3.1 Анализ существующих решений

Шикимори

Является самой популярной русскоязычной энциклопедией аниме, манги, а также ранобэ. Ее преимуществами являются полнота наполнения произведениями азиатской культуры, широкие возможности поиска и сортировки по группам интереса. Однако сервис не предоставляет возможность просмотра аниме, реализует специфичную систему отслеживания онгоингов, а также довольно сильно перегружен информацией, которая не относится к тематике просмотра и агрегации аниме.

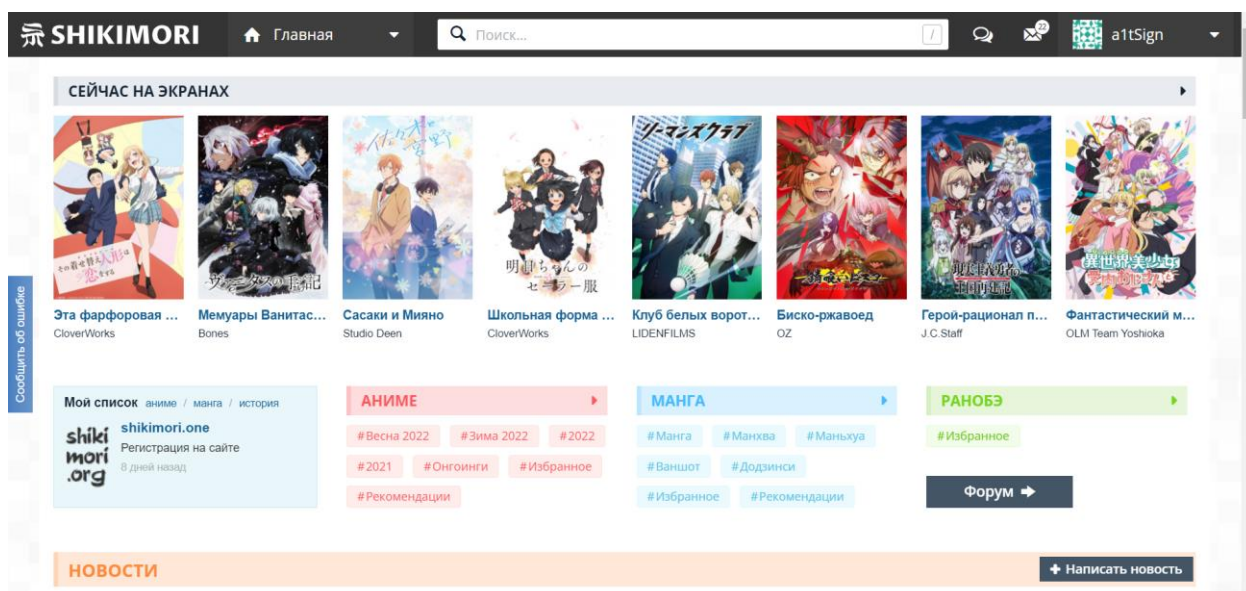


Рисунок 1 – Домашняя страница Шикимори

MyAnimeList

Одна из самых популярных англоязычных энциклопедий аниме и манги. К ее достоинствам можно донести полноту наполнения, интуитивный дизайн страниц с тайтлами, включающий в себя отзывы, информацию о персонажах и актерах озвучки. Однако присутствуют и существенные недостатки в виде недоступности в России, обилия рекламы и перегруженности контентом, который не относится к тематике просмотра и агрегации аниме.

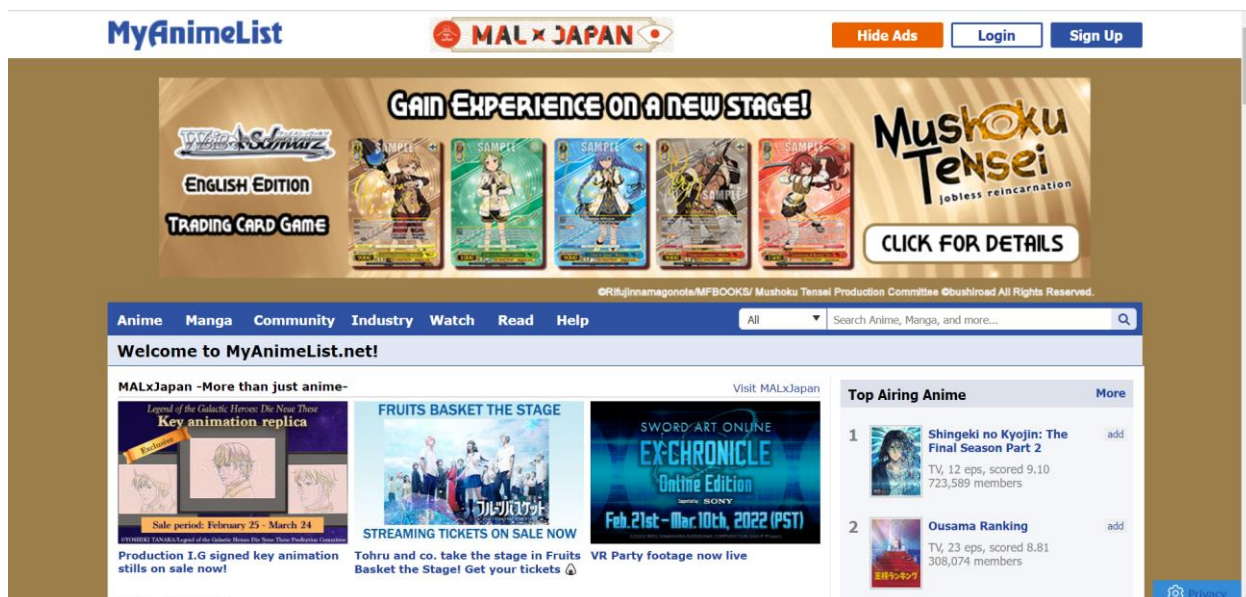


Рисунок 2 – Главная страница MyAnimeList

3.2 Анализ целевой аудитории

Наше приложение позволяет пользователям просматривать аниме без необходимости поиска подходящего плеера на разных сайтах, делая доступным получение видеоматериалов из нескольких источников для повышения вариативности. Возможность создания списков по группам интересов пользователя позволяет ему эффективно распоряжаться своим временем и планировать просмотр видеоматериалов.

Итого:

- просмотр аниме через сервис “Shiki-Adapter”, предназначен для пользователей, в основном входящих в группу молодежи;
- целевая аудитория: женщины и мужчины от 16 до 35 лет;
- в ЦА входят все слои населения;
- семейное положение: замужем/не замужем;
- интересы: путешествия, работа, иностранная культура, мультипликация, отдых и другие;
- ядро ЦА — женщины и мужчины от 16 до 30 лет.

3.3 Пользователи системы

Для взаимодействия с данной системой выделяют следующие виды пользователей:

- гость;
- пользователь;
- модератор;
- администратор.

Гость имеет возможность в процессе взаимодействия с данной системой:

- искать и просматривать аниме;
- производить авторизацию с помощью существующего аккаунта на Shikimori.

Пользователь имеет возможность в процессе взаимодействия с данной системой:

- возможности неавторизованного пользователя;
- добавлять соответствующие аниме в списки “Запланировано”, “Смотрю”, “Просмотрено”, “Брошено”;
- узнавать о появлении новых серий избранных сериалов из списка “Смотрю” путем просмотра информации в автоматически генерируемом списке “Обновления”.

Модератор имеет возможность в процессе взаимодействия с данной системой:

- возможности неавторизованного пользователя;
- редактировать источники видеоматериалов.

Администратор имеет возможность в процессе взаимодействия с данной системой:

- возможности неавторизованного пользователя;
- создавать, редактировать и удалять аккаунты модераторов.

4 Анализ задачи

4.1 Варианты использования системы

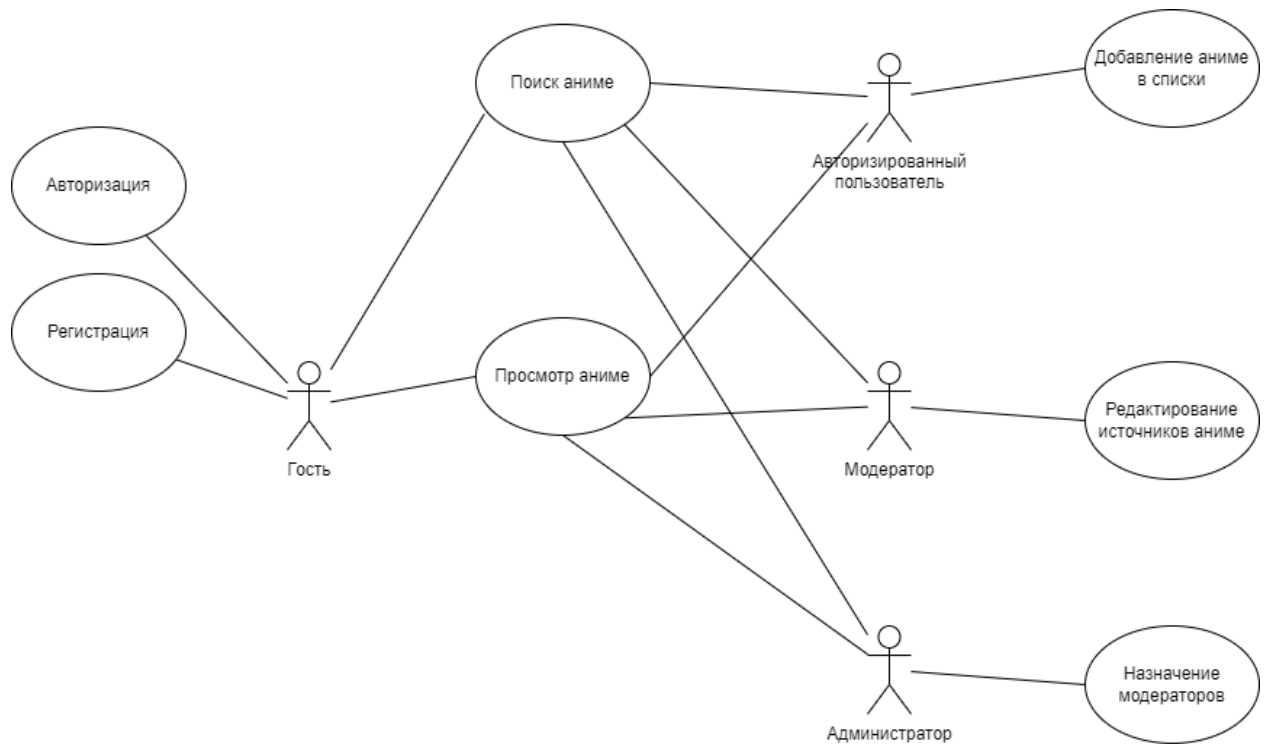


Рисунок 3 – Диаграмма прецедентов

Гость (неавторизированный пользователь) имеет следующие возможности:

- авторизоваться в системе;
- зарегистрироваться в системе;
- найти аниме;
- просмотреть аниме.

Авторизированный пользователь имеет следующие возможности:

- добавление аниме в списки;
- возможности незарегистрированного пользователя, кроме регистрации и авторизации.

Модератор имеет следующие возможности:

- редактирование источников видеоматериалов;

- возможности незарегистрированного пользователя, кроме регистрации и авторизации.

Администратор имеет следующие возможности:

- возможности незарегистрированного пользователя, кроме регистрации и авторизации;
- назначение модераторов.

Опишем конкретно два сценария использования приложения.

Первый сценарий – сценарий незарегистрированного пользователя. Пользователь попадает на сайт может просматривать и искать аниме, но для того, чтобы создавать собственные списки, ему нужно будет авторизоваться или зарегистрироваться в системе. Часть пользователей может никогда не проходить регистрацию или авторизацию, довольствуясь функционалом гостя.

Второй сценарий – сценарий зарегистрированного пользователя. Вдобавок ко всем действиям, описанным в предыдущем сценарии, пользователь получает возможность создавать собственные списки с аниме.

4.2 Общая структура системы

Система является клиент-серверным приложением с панелью администрирования, вследствие чего его поддержка не требует специально обученного персонала. Состоит из трех элементов:

- клиентская часть – графический интерфейс, отображаемый в браузере. Пользователь взаимодействует с приложением через этот самый браузер;
- серверная часть приложения – представляет собой программу, обрабатывающую запросы пользователя (браузера). Браузер в свою

очередь отображает полученную информацию от сервера в графическом интерфейсе;

- система управления базой данных (СУБД) – программное обеспечение, занимающиеся управлением и хранением данных. Серверная часть приложения обращается к базе данных и получает оттуда нужную информацию, которую затем передает графическому интерфейсу.

4.3 Взаимодействие компонентов системы

Варианты последовательностей, доступные только для гостя, отображены на рисунках 4 и 5.

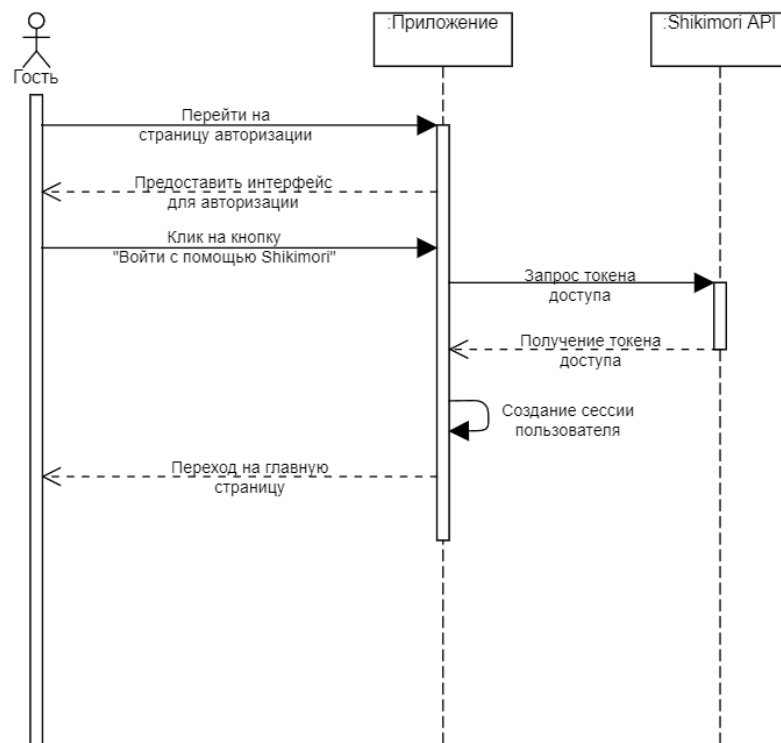


Рисунок 4 – Авторизация

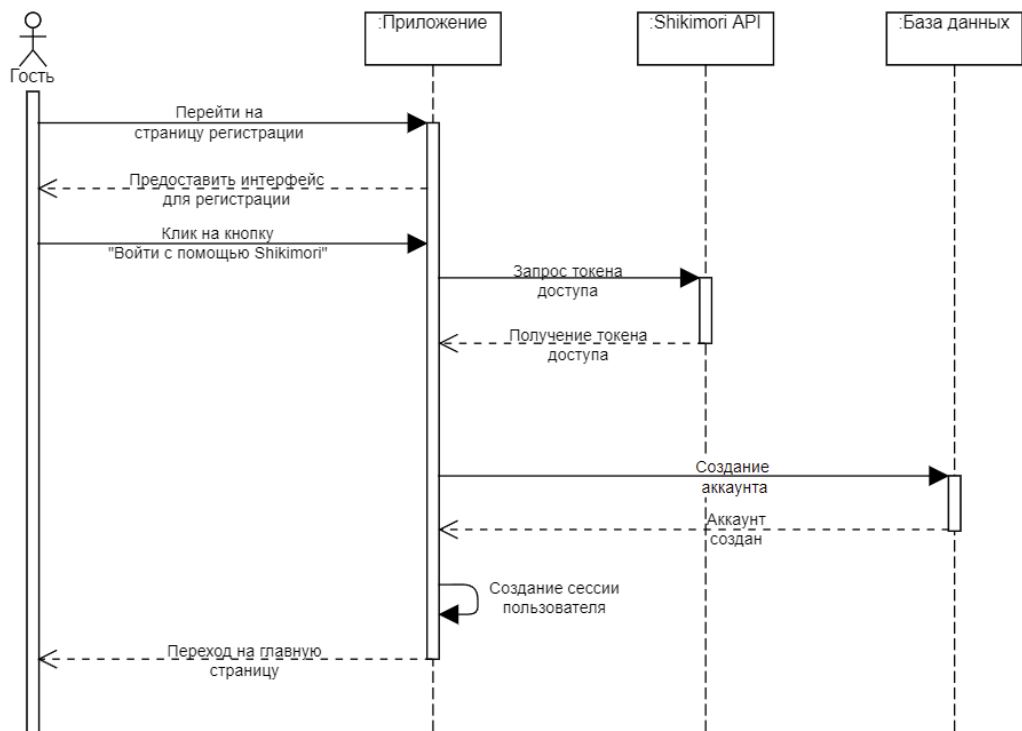


Рисунок 5 – Регистрация

Варианты последовательностей, доступные и гостю, и авторизованному пользователю, отображены на рисунках 6 и 7.

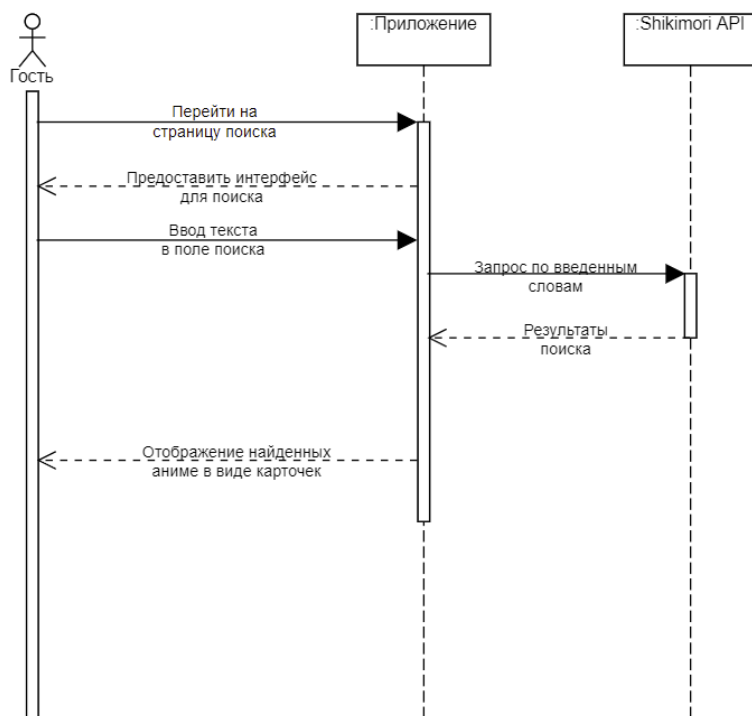


Рисунок 6 – Поиск

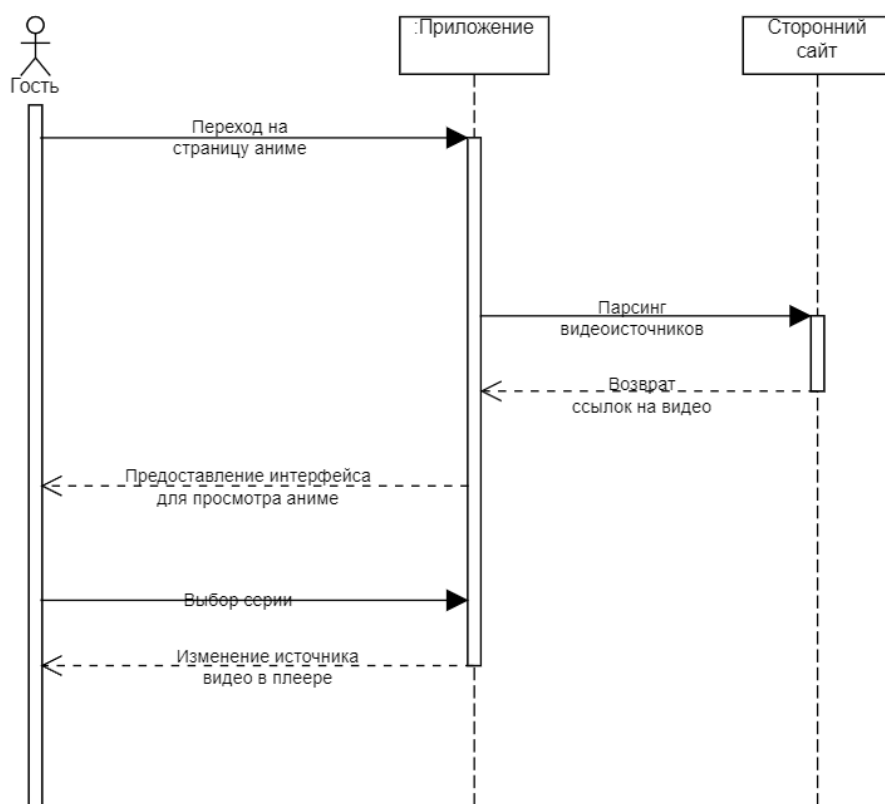


Рисунок 7 – Просмотр аниме

Варианты последовательностей, доступные только для авторизованного пользователя, отображены на рисунках 8 и 9.

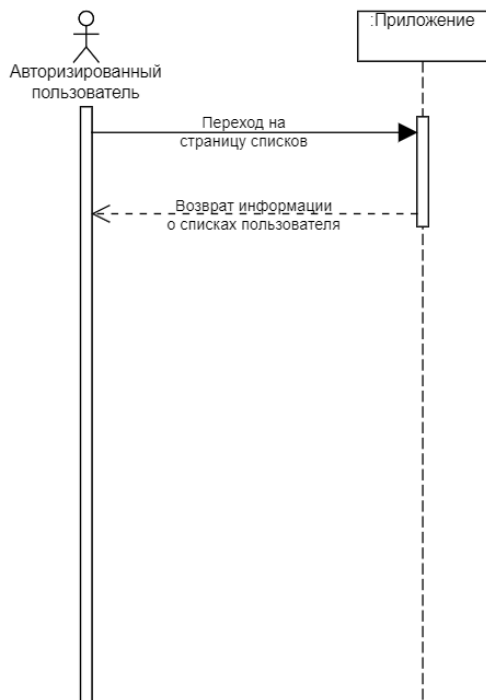


Рисунок 8 – Списки

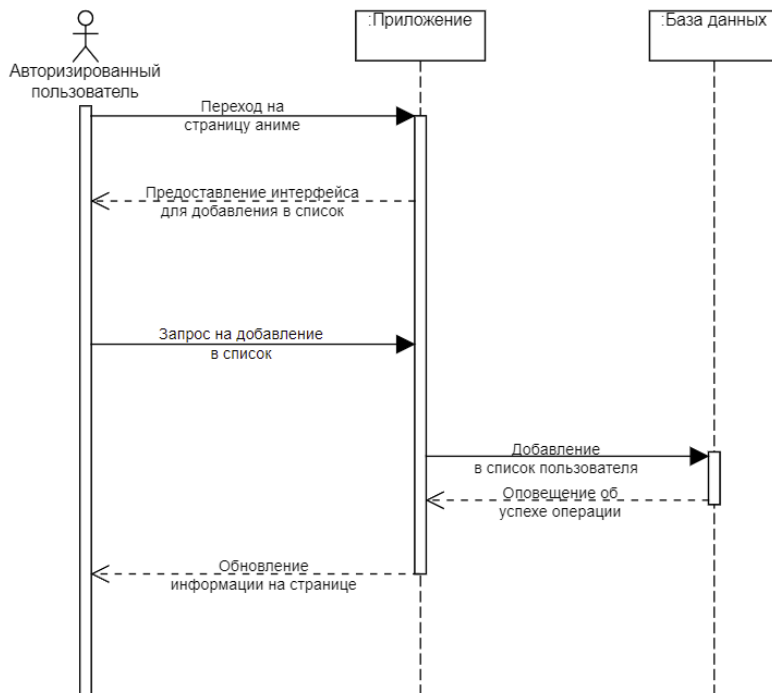


Рисунок 9 – Добавление аниме в список

4.4 Варианты состояния системы

Состояния системы для гостя отображены на Рисунке 10.

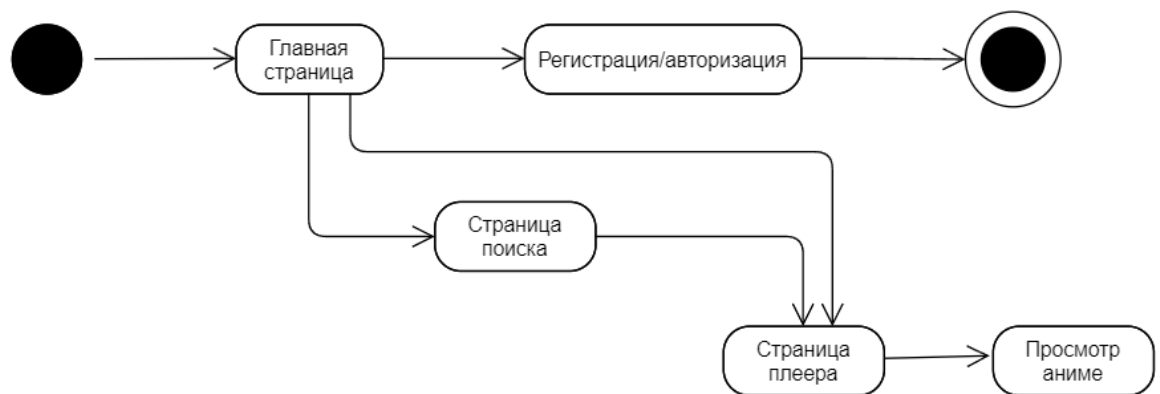


Рисунок 10 – Диаграмма состояний для гостя

Действия авторизованного пользователя расширяются (Рисунок 11).



Рисунок 11 – Диаграмма состояний для авторизованного пользователя

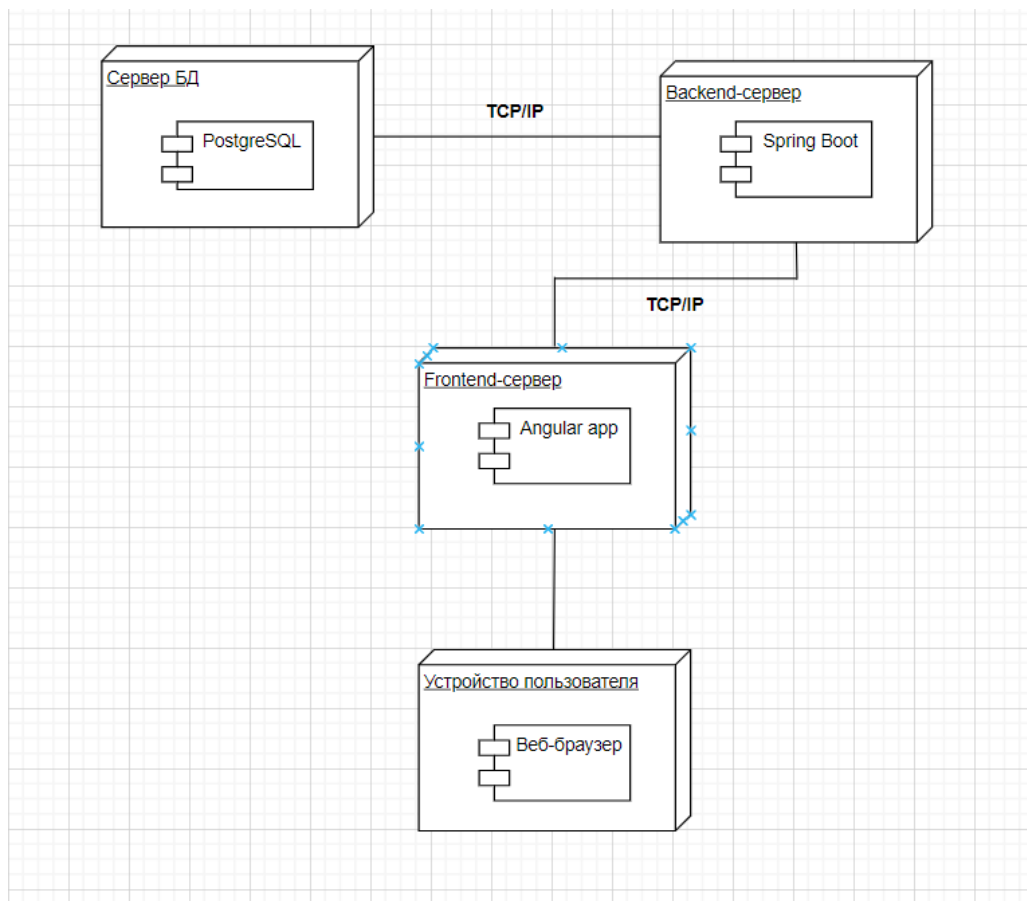


Рисунок 12 – Диаграмма развертывания всего приложения

Данная диаграмма (Рисунок 12) показывает архитектуру развертывания всей системы: программные среды исполнения и промежуточное программное обеспечение, которое описывает эти самые среды

4.6 IDEF0

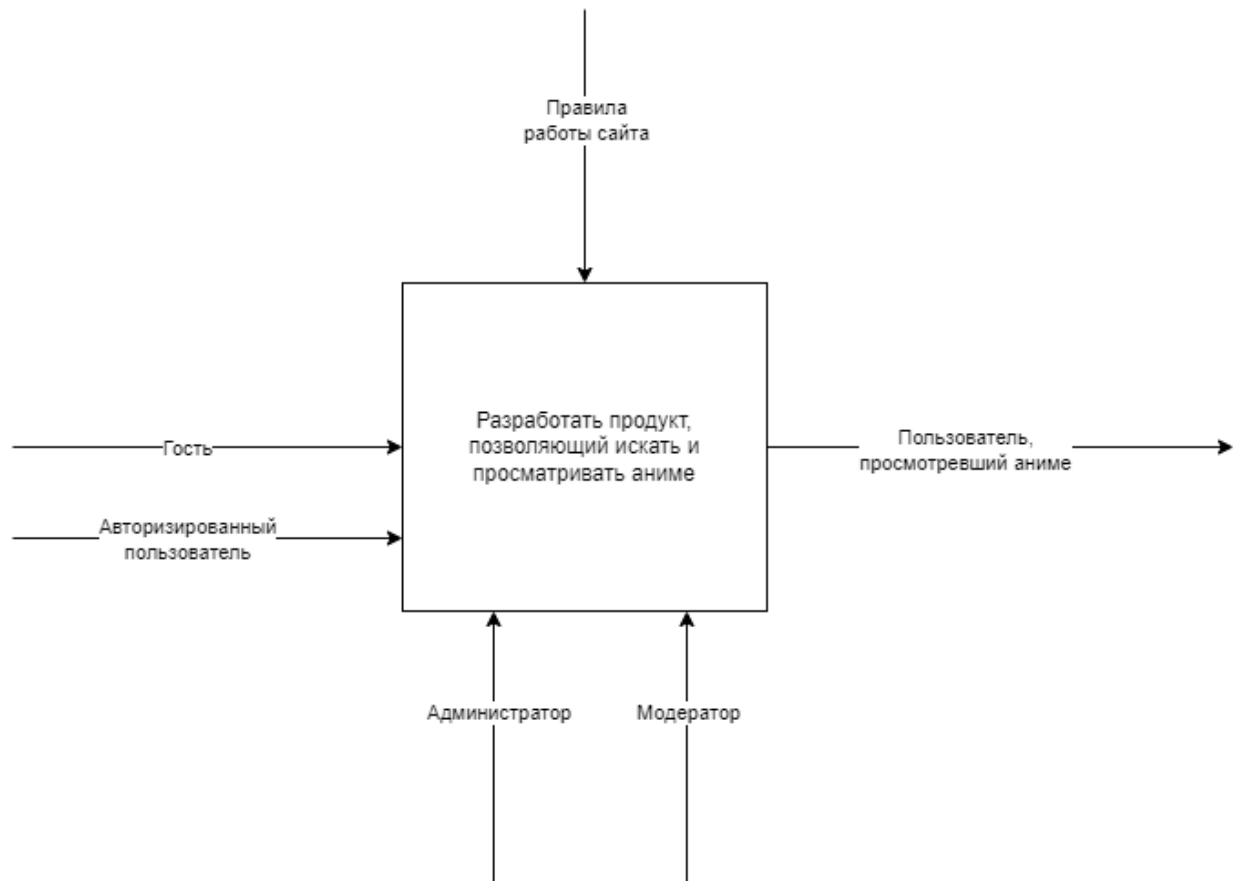


Рисунок 13 – Диаграмма IDEF0

4.7 Диаграмма классов

Диаграмма классов – структурная диаграмма языка UML, показывающая общую структуру иерархии классов системы, их взаимодействие, атрибуты интерфейсы и взаимосвязь между ними.

Ниже приведена диаграмма классов, разбитая на несколько частей, для улучшения ее удобочитаемости.

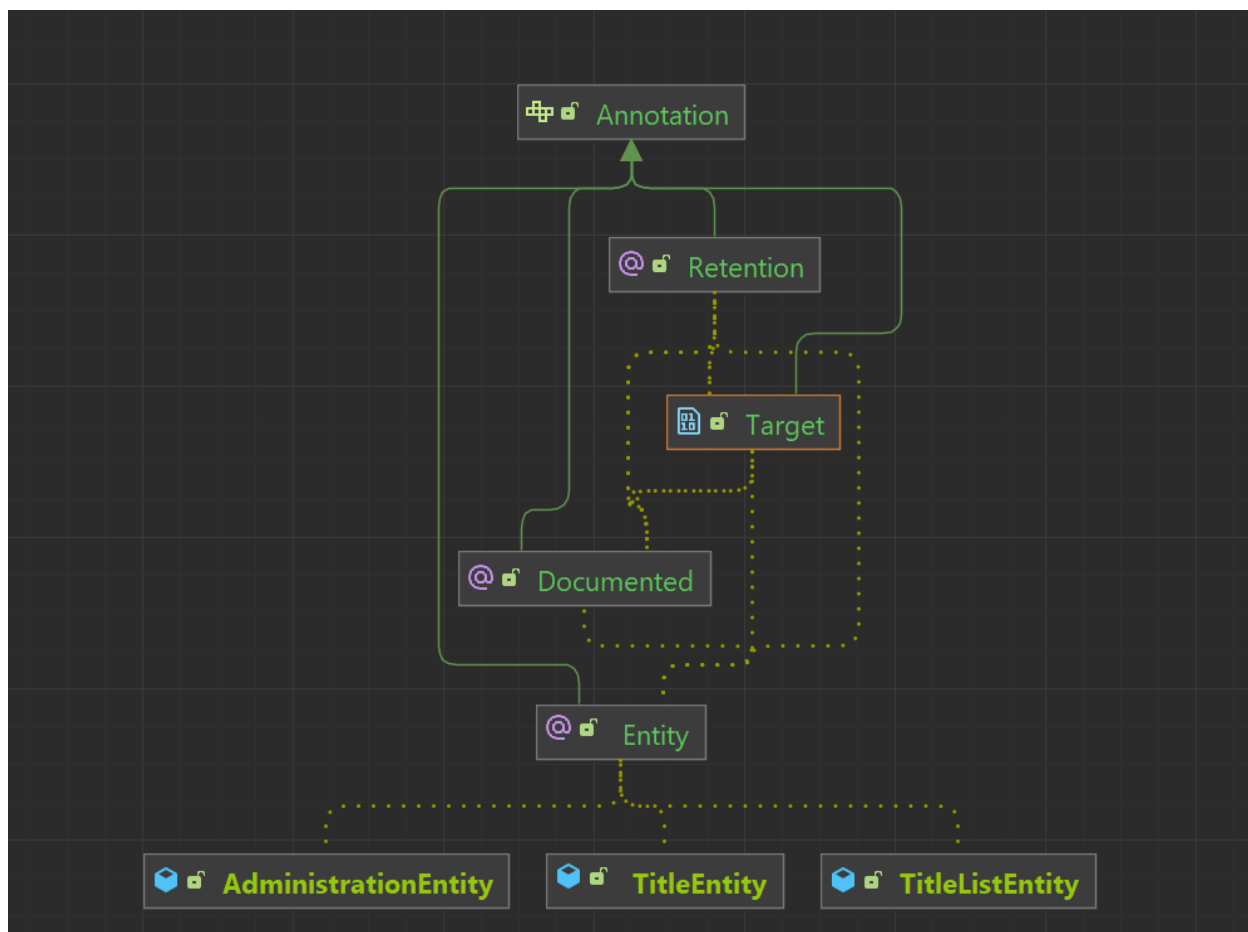


Рисунок 14 – Диаграмма классов для Entity

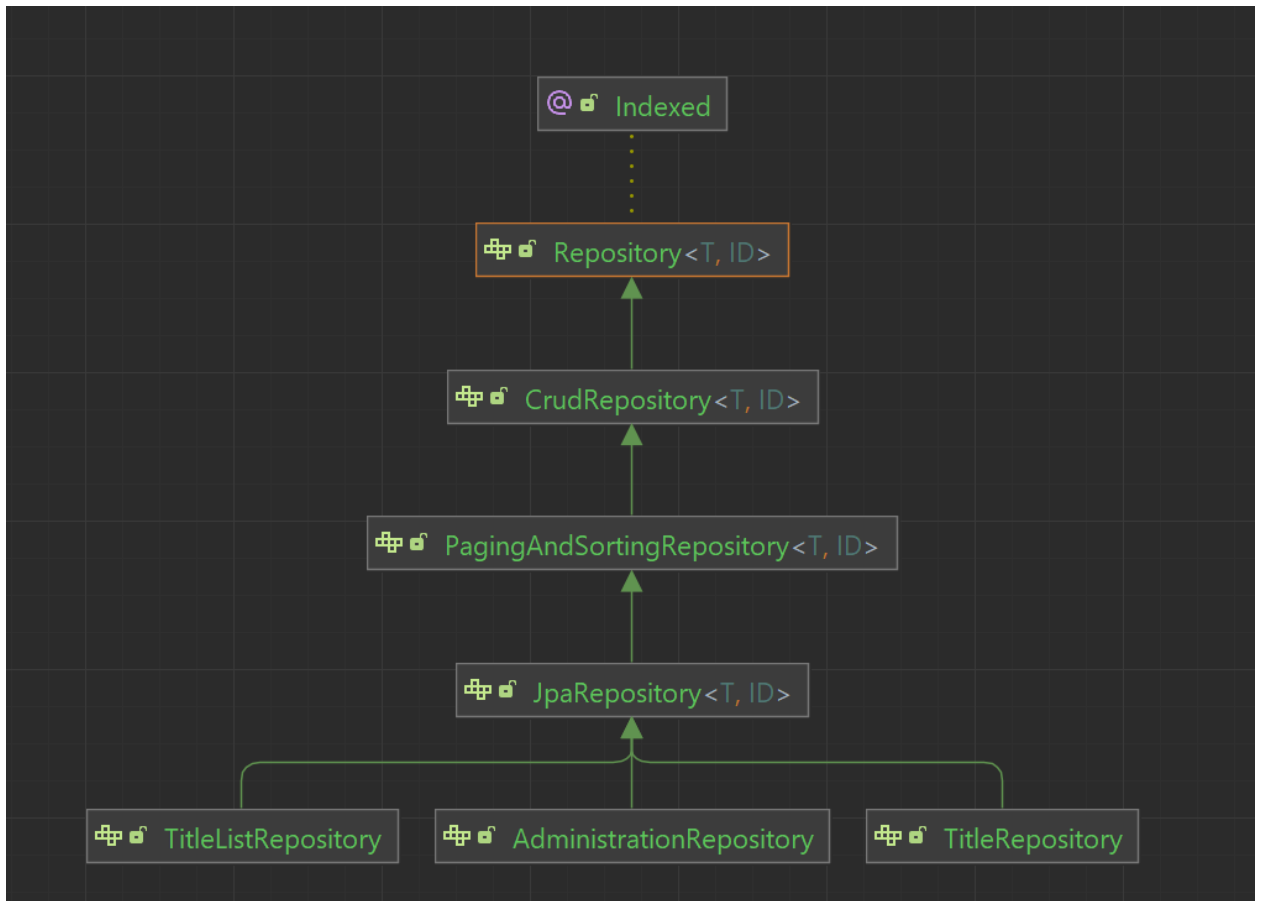


Рисунок 15 – Диаграмма классов для репозиториев

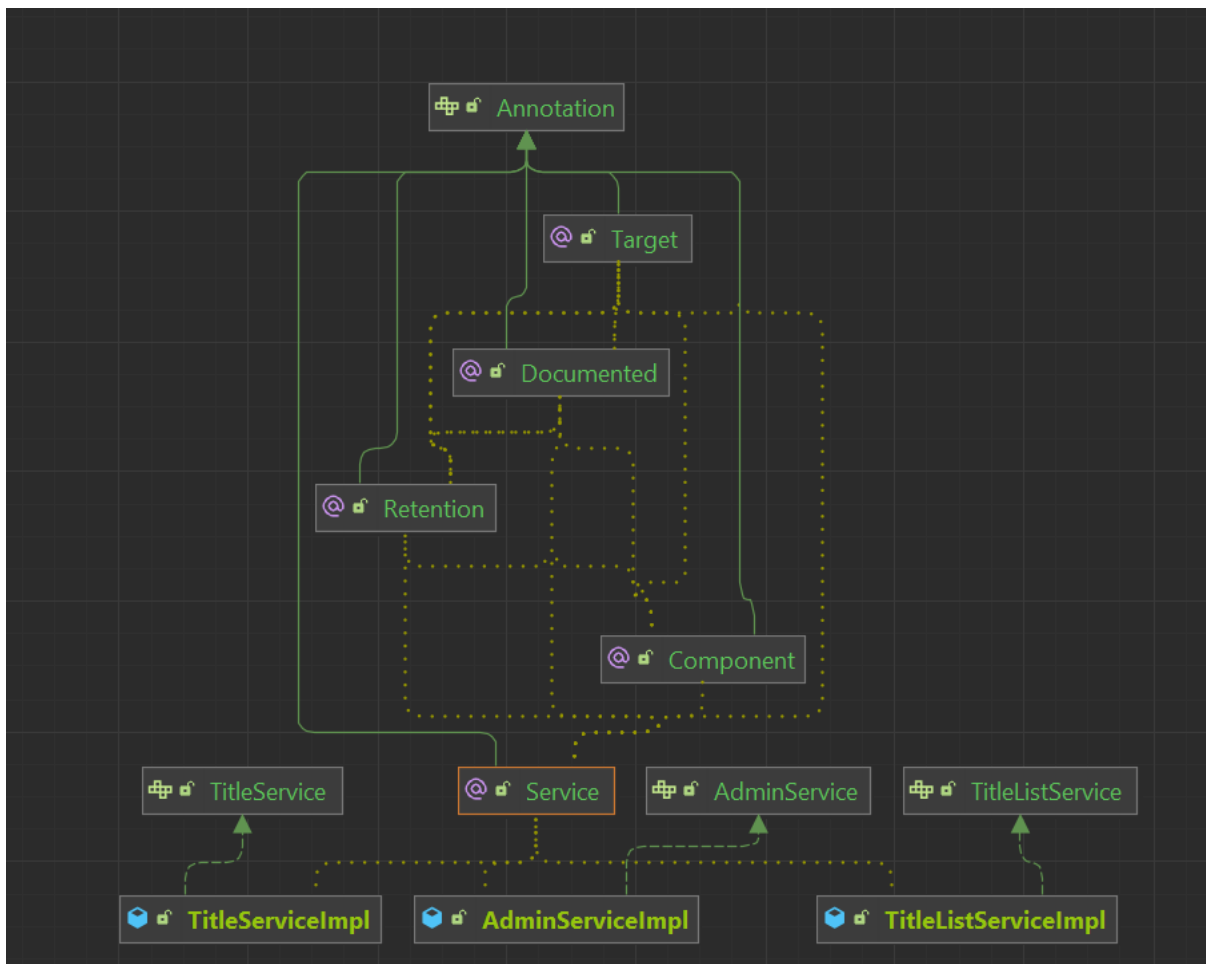


Рисунок 16 – Диаграмма классов для сервисов

4.8 ER-диаграмма

ER-диаграмма базы данных показана на Рисунке 16.

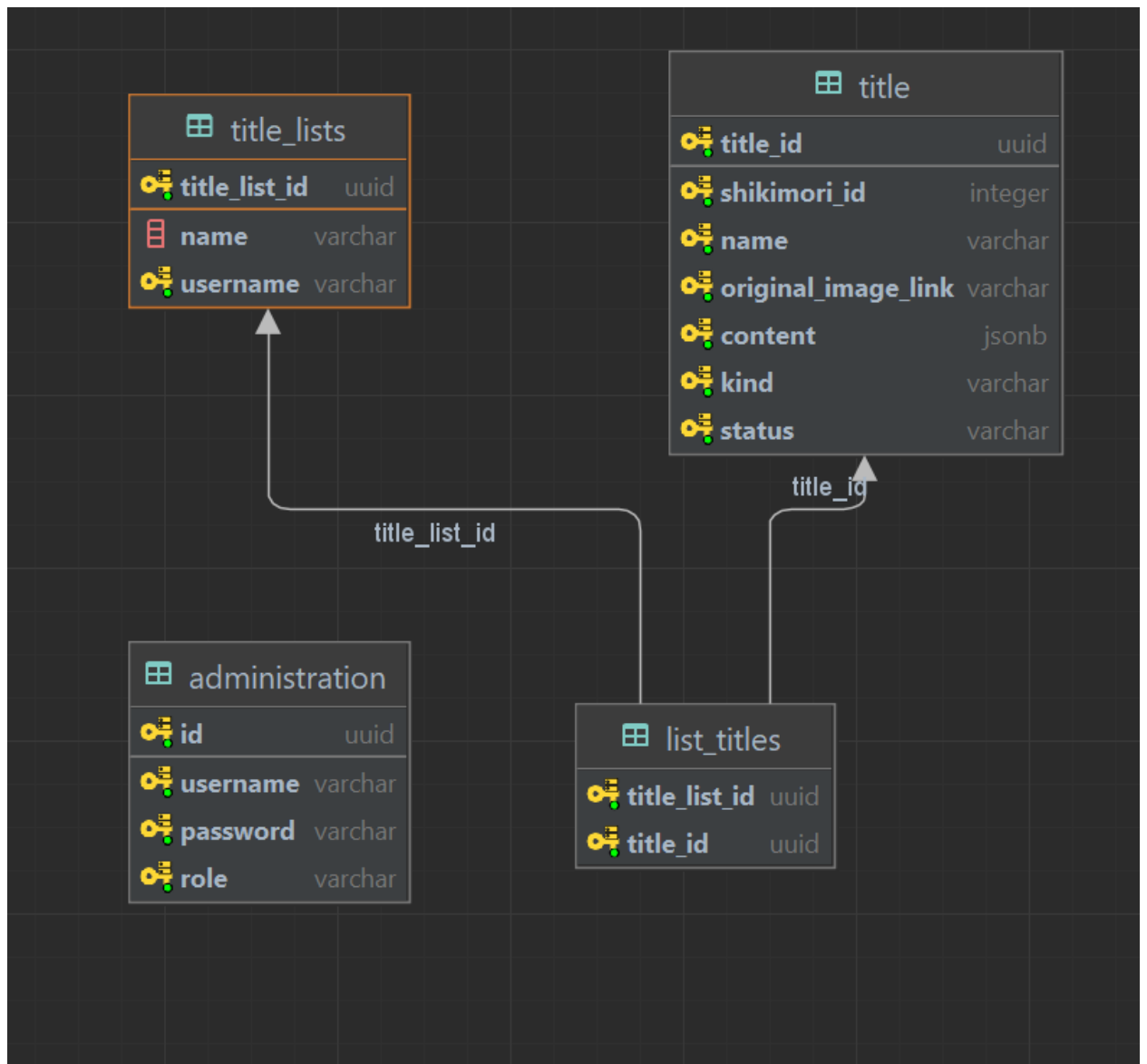


Рисунок 17 – ER-Диаграмма

5 Анализ средств реализации

Для выполнения поставленных задач по Frontend разработке были выбраны следующие средства: NPM, Angular в связке с Typescript. Они обладают следующими преимуществами:

- typescript расширяет язык Javascript, добавляя в него типы. Это помогает при разработке объемных приложений;
- фреймворк Angular предоставляет практически весь необходимый функционал для написания фронтенд-приложения;

- все дополнительные библиотеки можно подключить к приложению с помощью `npm`.

Для выполнения поставленных задач по Backend разработке был выбран язык Java, а в качестве основного фреймворка был выбран Spring Boot. Вместе эта связка технологий дает ряд преимуществ:

- высокая скорость разработки;
- возможность быстрого масштабирования приложения;
- слабая связность приложения;
- автономность приложения;
- дополнительные библиотеки, функциональность которых уменьшает количество кода;
- простота настройки и управления.

В качестве системы управления базой данных была выбрана PostgreSQL. Ее преимуществами являются:

- поддержка базы данных неограниченного размера;
- свободное и открытое программное обеспечение, поддерживаемое широким комьюнити;
- простое взаимодействие с нетипичными типами данных;
- устойчивость и расширяемость.

5.1 Frontend

Коммуникация с серверной частью приложения осуществлена при помощи стандартных возможностей Angular. В нужные места импортирован `HttpClient`, который и позволяет отправлять запросы.

Используется компонентная архитектура, компоненты следуют архитектуре MVC.

5.2 Backend

Серверная часть приложения написана с помощью языка `java`, используя фреймворк Spring Boot.

Архитектура включает в себя следующие компоненты:

- контроллеры;
- сервисы;
- репозитории;
- сущности (Entity).

Контроллеры выполняют роль непосредственного обработчика входящих запросов.

Сервисы в свою очередь выполняют основную бизнес-логику приложения, используя различные утилиты или написанный в них же код.

Репозитории – это слой, который позволяет удобно извлекать из базы данных все данные в виде сущностей, с целью их дальнейшей передаче на обработку сервисам.

Сущности в свою очередь являются объектным представлением таблиц базы данных, с которыми в java коде намного привычнее работать.

Серверная часть была задеплоена на Heroku для полноценной работы. Деплой происходил простым методом `git push` в удаленный репозиторий, расположенный на Heroku. Данный способ удобен тем, что внесенные изменения сразу же можно задеплоить и они применятся на удаленном сервере.

6 Воронки конверсии

Для сбора статистики был использован сервис Яндекс.Метрика. Были выделены основные цели, в рамках которых и отслеживались основные активности. Информация о достижении такой цели передается в Яндекс.Метрику с помощью JavaScript, что позволяет отслеживать практически любые произвольные события. Созданные цели и

последовательности действий для их достижения отображены на Рисунке 18. Результаты на Рисунке 19.















Название цели	Описание	Номер цели
 1  Поиск	 составная цель Переход на страницу поиска: <ul style="list-style-type: none">• событие идентификатор цели: search_page_loaded Выполнен поиск: <ul style="list-style-type: none">• событие идентификатор цели: search_field_used	235642823  
 2  Просмотр аниме	 составная цель Открытие страницы аниме: <ul style="list-style-type: none">• событие идентификатор цели: anime_page_loaded Клик на плеер: <ul style="list-style-type: none">• событие идентификатор цели: anime_player_clicked	235642934  
 3  Просмотр своих списков	идентификатор: lists_page_loaded	235643346  

Рисунок 18 – Созданные метрики

Для сбора отзывов по работоспособности наших воронок конверсии, несколько людей из других команд прошли по сценариям:

Пользователь	Поиск	Просмотр аниме	Просмотр своих списков
Караваев Валентин	Успешно	Успешно	Успешно
Тишанский Данила	Успешно	Успешно	Успешно
Сметанин Иван	Успешно	Успешно	Успешно
Мясоедов Артем	Успешно	Успешно	Успешно
Крикунов Иван	Успешно	Успешно	Успешно

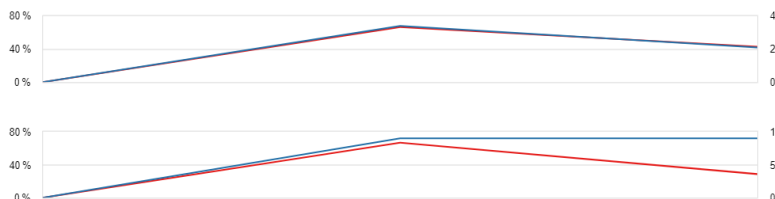
1. Составная цель «Поиск»

Переход на страницу поиска

Конверсия 50 %
Достижения цели 55
Целевые визиты 5

Выполнен поиск

Конверсия 40 %
Достижения цели 18
Целевые визиты 4



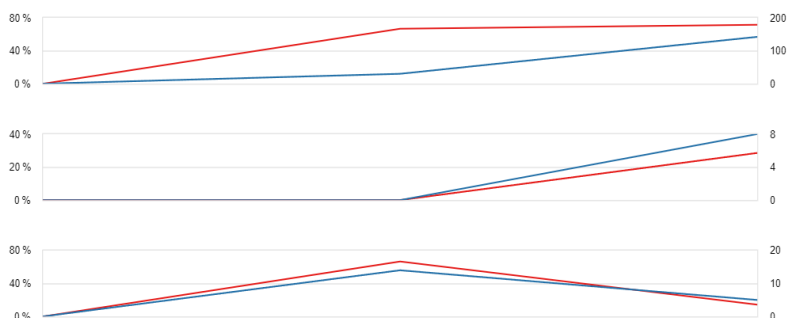
2. Составная цель «Просмотр аниме»

Открытие страницы аниме

Конверсия 70 %
Достижения цели 172
Целевые визиты 7

Клик на плеер

Конверсия 20 %
Достижения цели 8
Целевые визиты 2



3. Просмотр своих списков

Конверсия 30 %
Достижения цели 19
Целевые визиты 3

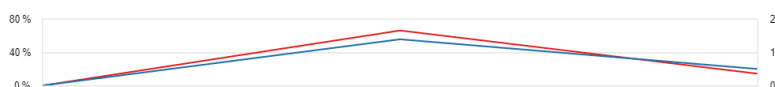


Рисунок 19 – Статистические данные

6.1 Поиск

Использование Яндекс.Метрики для учета количества использований возможностей поиска позволит отследить количество людей, заинтересованных в подборе аниме через наш сайт. Это важно учитывать, поскольку поиск может занимать много времени, если пользователь не уверен, что ищет.

Для выполнения условий данной воронки пользователю необходимо совершить следующие действия:

- открыть страницу поиска;
- ввести данные в текстовое поле.

6.2 Просмотр аниме

Статистика просмотров аниме является еще более важной, потому что основная цель сайта — предоставить пользователям возможность смотреть видео.

Для выполнения условий данной воронки пользователю необходимо совершить следующие действия:

- открыть страницу плеера;
- нажать на плеер.

6.3 Переход на страницу списков

Списки – важная функция нашего сайта, поэтому необходимо отследить количество посещений этой страницы.

Для выполнения условий данной воронки пользователю необходимо совершить следующие действия:

- открыть страницу списка.

7 Тестирование

После реализации всех задач был проведен запланированный набор тестов. Он включает 3 вида тестирования:

- дымовое тестирование;
- UI тесты;
- юзабилити тесты.

7.1 Дымовое тестирование

Для данного тестирования необходимо было проверить работоспособность сайта на следующих основных сценариях:

- авторизация;
- поиск аниме;
- просмотр страницы тайтла;
- воспроизведение видеоматериала;
- редактирование источников видео;

- редактирование, добавление и удаление модераторов.

Дымовое тестирование проводилось ручным образом в следующих браузерах: Google Chrome, Yandex Browser. Результаты, полученные в ходе тестирования, приведены в Таблице 1.

Таблица 1 – Результаты дымового тестирования

Сценарий	Результат
Авторизация	Пройден
Поиск аниме	Пройден
Просмотр страницы тайтла	Пройден
Воспроизведение видеоматериала	Пройден
Редактирование источников видео	Пройден
Редактирование, добавление и удаление модераторов	Пройден

По итогу дымового тестирования было установлено, что сайт проходит все основные утвержденные сценарии.

7.2 UI тесты

Было проведено UI тестирование, охватывающее основные возможности сайта.

Таблица 2 – Результаты UI тестирования

Шаги теста	Ожидаемый результат	Статус
Нажимается кнопка “FAQ” на главной странице	Открылась страница с описанием проекта	Пройден
Нажимается кнопка “GitHub” на главной странице	Открылась страница с исходным кодом проекта	Пройден
Нажимается кнопка “Я модератор” на главной странице	Открылась форма регистрации для модераторов	Пройден
Нажимается кнопка со значком “домик”	Переход на главную страницу приложения	Пройден

Нажимается кнопка со значком “лупа”	Переход на страницу поиска	Пройден
Нажимается кнопка со значком “вход” (в левом верхнем углу)	Переход на страницу авторизации с помощью Шикимори	Пройден
Нажимается элемент тайтла на всех страницах, где он расположен	Переход на страницу просмотра видео	Пройден
1. Нажимается кнопка поиска тайтлов 2. Вводится подстрока поиска	На странице появляются тайтлы, название которых содержит в себе подстроку поиска	Пройден
1. Авторизироваться с помощью Шикимори 2. Нажать на кнопку “списки” (представлена в виде кнопки с тремя линиями) 3. Выбрать любой из списков, расположенных справа под кнопкой авторизации	На странице появляются тайтлы, которые содержатся в выбранном списке пользователя	Пройден
Открыть главную страницу приложения (как неавторизованный пользователь)	На странице сверху доступны две кнопки: домашняя страница и поиск	Пройден
1. Нажать на кнопку “Я модератор” 2. Авторизироваться через аккаунт модератора 3. Перейти на страницу любого тайтла	Доступны все возможности неавторизованного пользователя	Пройден
1. Нажать на кнопку “Я модератор” 2. Авторизироваться через аккаунт администратора	На странице отображены все модераторы	Пройден

3. Перейти на страницу управления модераторами		
1. Нажать на кнопку авторизации через Шикимори 2. Авторизироваться 3. Нажать на любой из тайтлов	Откроется страница для просмотра видео	Пройден
1. Нажать на кнопку авторизации через Шикимори 2. Авторизироваться 3. Нажать на любой из тайтлов, который есть в каком-либо списке	Откроется страница для просмотра видео, а одна из кнопок списков будет подсвечена синим	Пройден
1. Нажать на кнопку авторизации через Шикимори 2. Авторизироваться 3. Нажать на любой из тайтлов, который есть в каком-либо списке 4. Удалить тайтл из списка, нажав на кнопку с названием списка, в котором тайтл находится (подсвечена синим цветом)	Откроется страница для просмотра видео, а одна из кнопок, изначально подсвеченная синим, примет тот же цвет, что и остальные	Пройден

7.3 Юзабилити тесты

Для проведения юзабилити тестирования было отобрано три случайных человека, не пользовавшиеся заранее сайтом. Для данного тестирования необходимо проверить следующие основные сценарии взаимодействия пользователя с сайтом:

- авторизация;
- просмотр аниме без авторизации;

- просмотр аниме с авторизацией;
- просмотр вкладки о нас;
- поиск аниме без авторизации;
- поиск аниме с авторизацией;
- просмотр списков с авторизацией;
- возможность авторизации как модератор;
- возможность редактирования источников видео для модераторов.

Таблица 3 – Результаты юзабилити тестирования

Сценарий	Пользователь 1	Пользователь 2	Пользователь 3
Авторизация	Пройден	Пройден	Пройден
Просмотр аниме без авторизации	Пройден	Пройден	Пройден
Просмотр аниме с авторизацией	Пройден	Пройден	Пройден
Просмотр вкладки о нас	Пройден	Пройден	Пройден
Поиск аниме без авторизации	Пройден	Пройден	Пройден
Поиск аниме с авторизацией	Пройден	Пройден	Пройден
Просмотр списков с авторизацией	Пройден	Пройден	Пройден
Возможность авторизации как модератор	Пройден	Пройден	Пройден
Возможность редактирования источников видео для модераторов	Пройден	Пройден	Пройден

8 Используемые технологии

Разберём технологии используемые в Backend части приложения.

Серверная часть веб-приложения написана на языке Java. Для упрощения конфигурации и взаимодействия с веб-сервером в приложении использовался фреймворк Spring Boot. Для построения системы аутентификации и авторизации, а также для обеспечения безопасности был использован фреймворк Spring Security и JWT. Слой доступа к данным и взаимодействия с базой данной реализован с помощью фреймворка Spring Data JPA. Для управления схемой базы данных используется библиотека Liquibase. Для сокращения количества однотипного кода в Entity и DTO классах была использована библиотека Lombok. В качестве реляционной СУБД использовалась PostgreSQL а в качестве системы контроля .

Теперь перейдём к технологиям, которые были использованы во Frontend части системы.

- фреймворк Angular 13 версии;
- NPM – менеджер пакетов, который входит в состав Node.JS;
- Express – фреймворк, с помощью которого поднимается сервер;
- TypeScript – язык, расширяющий возможности EcmaScript;
- Yandex Metrika – инструмент для отслеживания статистики посещений сайта;
- дополнительные библиотеки.

Заключение

В результате работы был проведен анализ предметной области, разработано веб-приложение, которое помогает пользователю просматривать аниме. Были выполнены следующие задачи:

- разработана Frontend часть сайта, развернутая на удаленном сервере;
- разработана Backend часть сайта, развернутая на удаленном сервере;
- была создана связь между Frontend и Backend частями приложения;
- разработана база данных, развернутая на удаленном сервере;
- подключена Яндекс.Метрика, позволяющая фиксировать активность пользователей.

Приложение отвечает всем заявленным требованиям.

Список использованных источников

1. Using Liquibase with PostgreSQL – URL: <https://docs.liquibase.com/workflows/database-setuptutorials/postgresql.html> (дата обращения: 05.03.2021)
2. Документация ReactJS – URL: <https://ru.reactjs.org/docs> (дата обращения: 15.04.2021)
3. Craig Walls. Spring in Action (5th Edition) / Craig Walls, 2018.
4. Фаулер М. UML. Основы, 3 е издание. / Фаулер М. – Пер. с англ. – СПб: Символ Плюс, 2005. – 192 с.
5. Информационный ресурс Baeldung. – URL: <https://www.baeldung.com> (дата обращения 20.04.2021)