МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ “ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Веб-приложение для поиска фильмов «MovieGenie»

Курсовой проект

09.03.04 Программная инженерия

Информационные системы и сетевые технологии

Зав. кафедрой _____С.Д. Махортов, д.ф.- м.н., доцент __.__.20__

Обучающийся _____Д.Р. Утянский, 4 курс, д/о

Обучающийся _____Д.С. Зюбан, 4 курс, д/о

Обучающийся _____Н.В. Потапов, 4 курс, д/о

Руководитель _____В.С. Тарасов, ст. преподаватель

Воронеж 2023

Содержание

Введение

В современном мире люди все больше и больше используют интернет для поиска информации и развлечения. Один из самых популярных способов провести свободное время - это просмотр кинофильмов. Однако, с таким большим количеством фильмов, доступных для просмотра, найти что-то подходящее может быть непросто. Для упрощения этого процесса, все больше людей обращаются к сайтам для поиска фильмов.

Один из основных аспектов, почему такие сайты так популярны, заключается в том, что они предоставляют обширные базы данных о фильмах, содержащие информацию о релизах, актерах, съемочной группе, сюжете, рейтингах и многое другое. Благодаря этим сайтам можно получить подробную информацию о фильмах, просмотреть трейлеры, узнать о предстоящих премьерах и планах выпуска новых фильмов.

Кроме того, они упрощают процесс выбора конкретного фильма для просмотра, Предоставляя функции поиска и фильтрации, позволяющие найти фильмы по определенным критериям, таким как жанр, год выпуска, режиссер, актерский состав и так далее. Это помогает нам не только сэкономить время и получить рекомендации, исходя из наших предпочтений, но и углубиться в мир кино, найти новые фильмы для просмотра и поделиться своими впечатлениями с другими людьми.

В этой курсовой работе будет рассмотрена тема создания сайта для поиска фильмов. Будут рассмотрены различные технологии, которые необходимы для разработки такого сайта, а также анализ существующих решений на рынке. Особое внимание будет уделено вопросам, связанным с функциональностью сайта, его дизайном и удобством использования, чтобы у пользователей был легкий доступ к необходимой информации.

1 Постановка задачи

Целью данного курсового проекта является разработка веб-приложения для поиска фильмов и создания подборок с ними.

Данное приложение предназначено для:

— Поиска фильма по названию или содержанию;

— Поиска и возможности сохранения подборок с фильмами;

— Создания собственных подборок с фильмами.

Для достижения поставленных целей необходимо:

— Иметь представление о разрабатываемой системе, представленное необходимыми UML-диаграммами и разработанным дизайном веб-приложения, как в целом, так и в отдельных сценариях;

— Провести обзор аналогов с целью определения достоинств и недостатков систем, реализующих схожий функционал;

— Спроектировать приложение с учётом данных, полученных в результате выполнения анализа;

— Реализовать приложение, соответствующее требованиям, представленным выше;

— Описать результат разработки.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

Разрабатываемый сервис должен иметь следующие функциональные возможности:

— Поиск фильма по названию;

— Поиск фильма по содержанию;

— Просмотр информации о фильме;

— Добавление фильма в список «Буду смотреть»;

— Просмотр пользовательских подборок фильмов;

— Сохранение подборок других пользователей;

— Создание своих подборок;

— Редактирование своих подборок;

— Добавление фильмов в созданную подборку;

— Удаление фильма из созданной подборки;

— Удаление своих подборок.

1.1.2 Требования к приложению и программному обеспечению

Для успешной работы веб-приложения для поиска фильмов необходимо выполнение следующих требований к приложению и программному обеспечению:

— Кроссплатформенность. Приложение должно быть доступно пользователям на разных операционных системах, таких как Windows, MacOS, Linux, а также на разных браузерах, таких как Google Chrome, Mozilla Firefox, Safari, Opera и других;

— Отзывчивость. Приложение должно быстро отображать результаты поиска и информацию о фильмах, а также реагировать на действия пользователя без задержек;

— Безопасность. Приложение должно обеспечивать безопасность пользовательских данных, включая защиту от взлома, хакерских атак и других угроз;

— Надежность. Приложение должно быть стабильным и не иметь ошибок, которые могут привести к потере данных пользователя или некорректному отображению информации;

— Масштабируемость. Приложение должно быть способно обрабатывать большое количество запросов и поддерживать большое количество пользователей без значительного снижения производительности;

— Совместимость. Приложение должно быть совместимо с различными сервисами, такими как API сторонних сервисов, базы данных и другие;

— Легкость использования. Приложение должно иметь интуитивно понятный интерфейс;

— Масштабируемость базы данных. База данных приложения должна быть масштабируемой и способной обрабатывать большой объем данных без значительного снижения производительности[4].

2 Анализ предметной области

2.1 Терминология (глоссарий) предметной области

— **Сервер, серверная часть** – компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач;

— **Клиент, клиентская сторона** – в данном проекте, устройство с установленным на него приложением для доступа в интернет, предоставляет возможности пользователю взаимодействовать со всей системой;

— **Front-end** – клиентская часть приложения. Отвечает за получение информации с программно-аппаратной части и отображение ее на устройстве пользователя;

— **Back-end** – программно-аппаратная часть приложения. Отвечает за функционирование внутренней (серверной) части приложения;

— **Пользователь, клиент** – человек, пользующийся функционалом приложения;

— **База данных** – это упорядоченный набор структурированной информации, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД);

— **Аутентификация** – процедура проверки подлинности, например, проверка подлинности пользователя путем сравнения введённого им пароля с паролем, сохраненным в базе данных;

— **Авторизация** – предоставление определённому лицу или группе лиц прав на выполнение определенных действий;

— **Фреймворк** – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта;

— **Аккаунт (учетная запись)** – это персональная страница пользователя или личный кабинет, который создается после регистрации в приложении;

— **Java** – строго типизированный объектно-ориентированный язык программирования[1];

— **Spring Framework** – универсальный фреймворк с открытым исходным кодом для Java-платформы[2];

— **Liquibase** – продукт с открытым исходным кодом для обеспечения миграций баз данных[6];

— **Docker** – программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации;

— **PostgreSQL** – объектно-реляционная система управления базами данных;

— **Flutter** – это платформа с открытым исходным кодом, который разработан и поддерживается Google;

— **Dart** – язык программирования общего назначения от компании Google, который предназначен прежде всего для разработки прикладных приложений[7];

— **REST API** – стиль архитектуры программного обеспечения для построения масштабируемых веб–приложений.

2.2 Обзор аналогов

2.2.1 IMDb

IMDb (The Internet Movie Database) - один из самых популярных и узнаваемых сайтов о кино, который предоставляет информацию о фильмах, телепередачах, сериалах, актерах, режиссерах и многое другое. На сайте имеется возможность смотреть трейлеры, читать критики и оставлять свои отзывы. Рассмотрим его плюсы и минусы:

Плюсы:

— Обширная база данных. На сайте содержится информация о многих фильмах, телесериалах, актерах, режиссерах и многое другое;

— Разнообразные оценки. На сайте можно найти оценки критиков и зрителей на фильмы, что помогает пользователям сделать более обоснованный выбор при выборе фильма для просмотра;

— Разделы с новостями и интервью. На сайте имеются разделы, в которых можно найти новости из мира кино и интервью с актерами и режиссерами;

— Возможность оставлять отзывы. Пользователи могут оставлять свои отзывы о просмотренных фильмах и смотреть отзывы других пользователей.

Минусы:

— Большое количество рекламы;

— Не всегда актуальная информация;

— Не всегда точные оценки (некоторые оценки на сайте могут быть завышенными или заниженными, так как они основаны на мнении обычных пользователей, которые могут быть необъективны);

— Ограниченная возможность поиска. Несмотря на обширную базу данных, поиск на сайте не всегда дает полный список результатов;

— Нет возможности смотреть фильмы на сайте. IMDb не предоставляет возможности просмотра фильмов на своем сайте. Пользователи должны искать другие ресурсы для просмотра фильмов, о которых они узнали на IMDb.

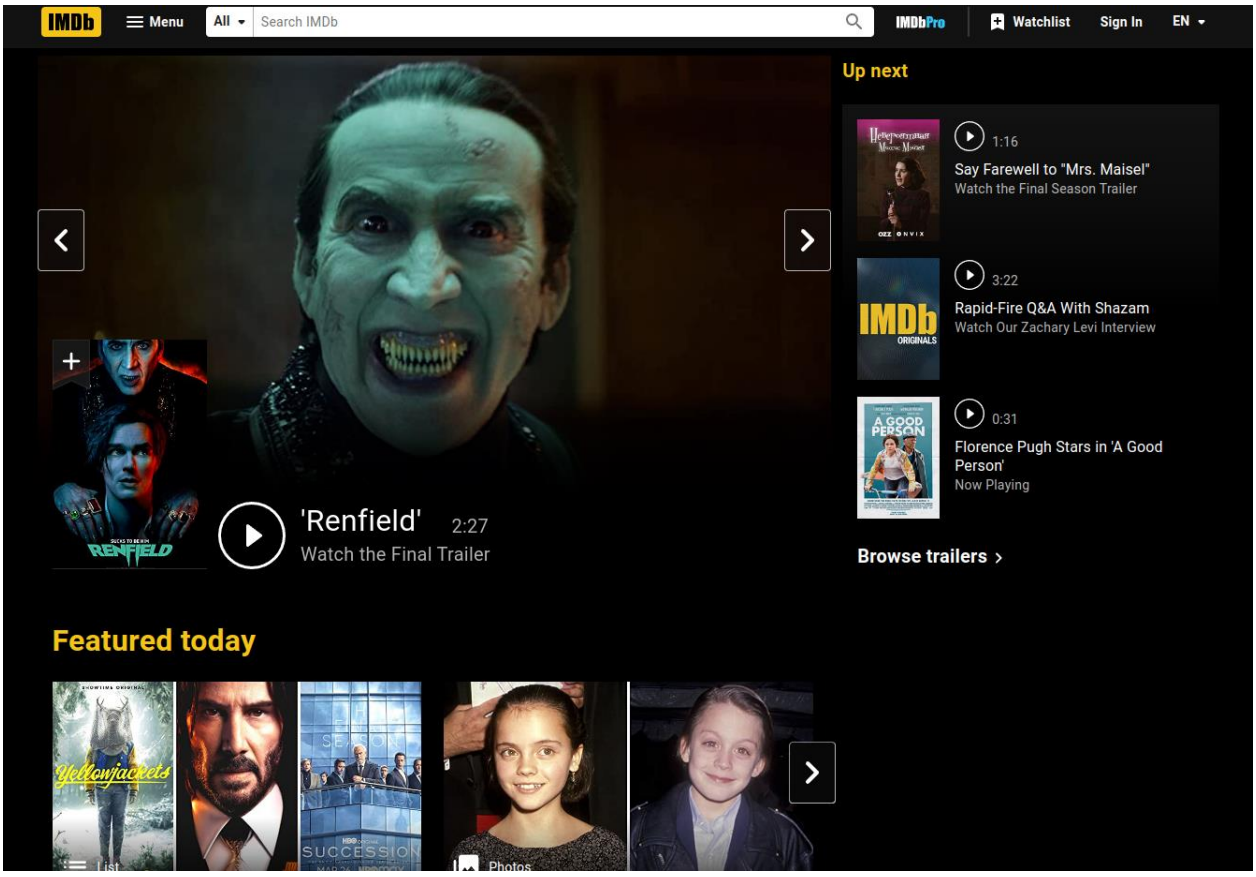

Рисунок 1 - Интерфейс приложения IMDb

2.2.2 Kinopoisk

Kinopoisk - это популярный российский онлайн-кинотеатр и база данных о фильмах, сериалах и актерах. Рассмотрим его плюсы и минусы:

Плюсы:

— Обширная база данных. На сайте можно найти информацию о многих фильмах, сериалах, актерах, режиссерах и т.д. База данных насчитывает более 400 тысяч фильмов и сериалов;

— Полезные рекомендации. Сайт использует информацию об истории поиска и просмотра пользователей для предоставления персональных рекомендаций на основе их предпочтений;

— Надежные оценки. Кинопоиск использует два типа оценок для каждого фильма: средняя оценка зрителей и средняя оценка кинокритиков;

— Разнообразные функции поиска. Сайт предоставляет возможность искать фильмы по жанру, году выпуска, стране производства и другим параметрам;

— Широкий выбор языков интерфейса. КиноПоиск имеет интерфейс на нескольких языках, что удобно для пользователей, которые не говорят на русском языке.

Минусы:

— Некоторые оценки могут быть необъективными. Некоторые зрители могут давать завышенные или заниженные оценки, что может повлиять на общую среднюю оценку фильма;

— Ограничения на просмотр фильмов. Некоторые фильмы могут быть доступны только на платформе определенного партнера, а не на самом Кинопоиске. Также некоторые фильмы могут быть доступны только с определенной географической локации;

— Реклама. Сайт содержит много рекламы, которая может отвлекать внимание пользователей от поиска нужной информации;

— Ограниченная доступность на мобильных устройствах. В отличие от IMDb, приложение КиноПоиск не так хорошо оптимизировано для мобильных устройств, что может вызывать некоторые неудобства при использовании сайта на смартфонах и планшетах.

Рисунок 2 - Интерфейс приложения Kinopoisk

2.2.3 Rotten Tomatoes

Rotten Tomatoes - это американский сайт, специализирующийся на сборе отзывов о фильмах, телевизионных шоу, видеоиграх и т.д. Рассмотрим его плюсы и минусы:

Плюсы:

— Агрегатор отзывов. Rotten Tomatoes собирает отзывы от кинокритиков и зрителей со всего мира, и представляет среднюю оценку на основе полученных результатов;

— Оценки кинокритиков и зрителей. Кроме общей средней оценки, сайт предоставляет информацию о средней оценке кинокритиков и зрителей отдельно;

— Обзоры кинокритиков. Сайт позволяет читать отзывы кинокритиков о фильмах и сериалах, что помогает пользователям сделать более информированный выбор;

— Возможность добавлять свои отзывы. Пользователи могут оставлять свои отзывы о фильмах и сериалах на сайте, что может помочь другим пользователям в выборе фильма;

— Разнообразные функции поиска. Сайт позволяет искать фильмы по жанру, году выпуска, режиссеру и другим параметрам.

Минусы:

— Ограниченность отзывов. Некоторые фильмы и сериалы могут не иметь достаточного количества отзывов для расчета средней оценки;

— Небольшое количество критиков на сайте;

— Система рейтингов не всегда справедлива. Некоторые пользователи жалуются на то, что оценки фильмов и сериалов могут быть недостаточно объективными, и что некоторые фильмы получают завышенные или заниженные оценки;

— Наличие рекламы на сайте. Как и IMDb и КиноПоиск, Rotten Tomatoes содержит много рекламы, что может отвлекать внимание пользователей от поиска нужной информации;

— Сайт ориентирован на американскую аудиторию. Rotten Tomatoes в основном ориентирован на американскую аудиторию, что может привести к недостаточному количеству информации о фильмах, выпущенных в других странах.

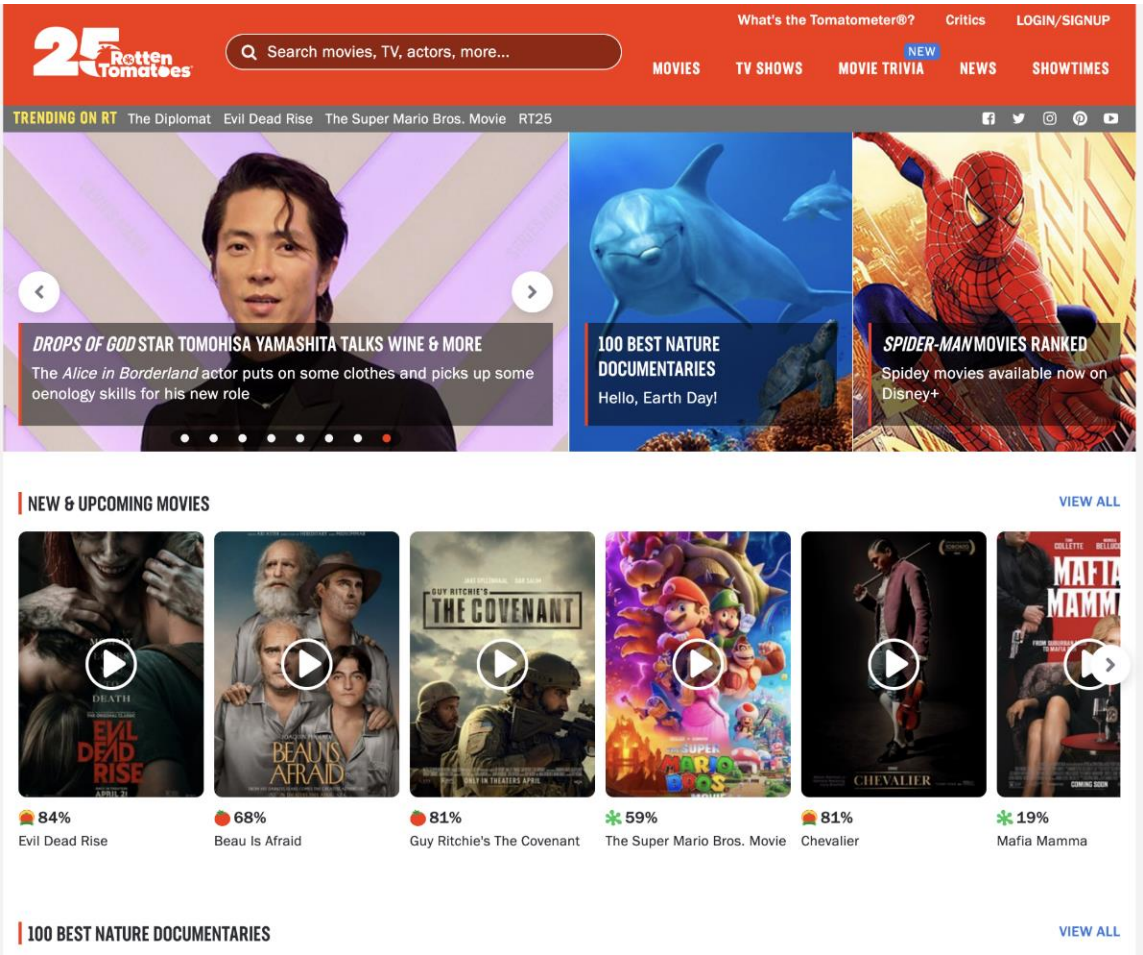

Рисунок 3 - Интерфейс приложения Rotten Tomatoes

3 Графическое описание работы системы

3.1 Диаграмма IDEF0

Благодаря диаграммам в стиле методологии IDEF0 можно увидеть работу системы, когда пользователь взаимодействует с приложением на этапах процесса работы с подборками.

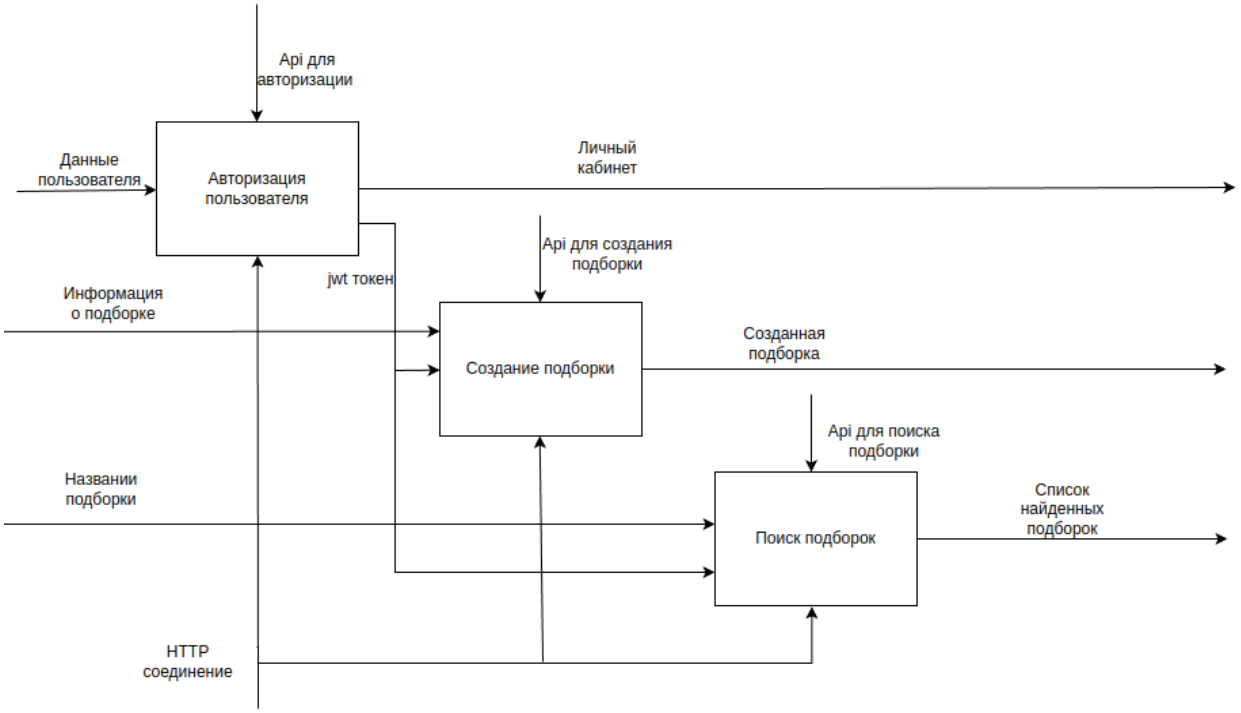

Рисунок 4 - Диаграмма IDEF0

3.2 Диаграмма вариантов использования

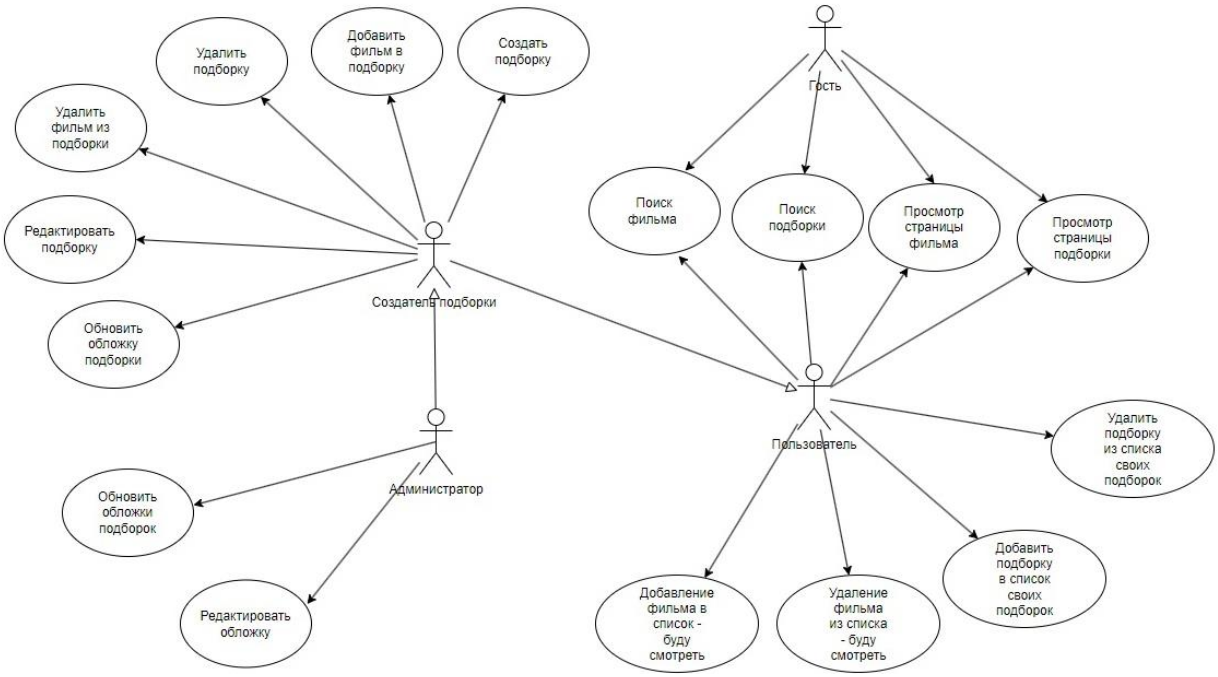

Рисунок 5 - Диаграмма вариантов использования

Диаграмма вариантов использования (Use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. Она помогает понять требования к системе и определить её функциональные возможности.

3.3 Диаграмма состояний

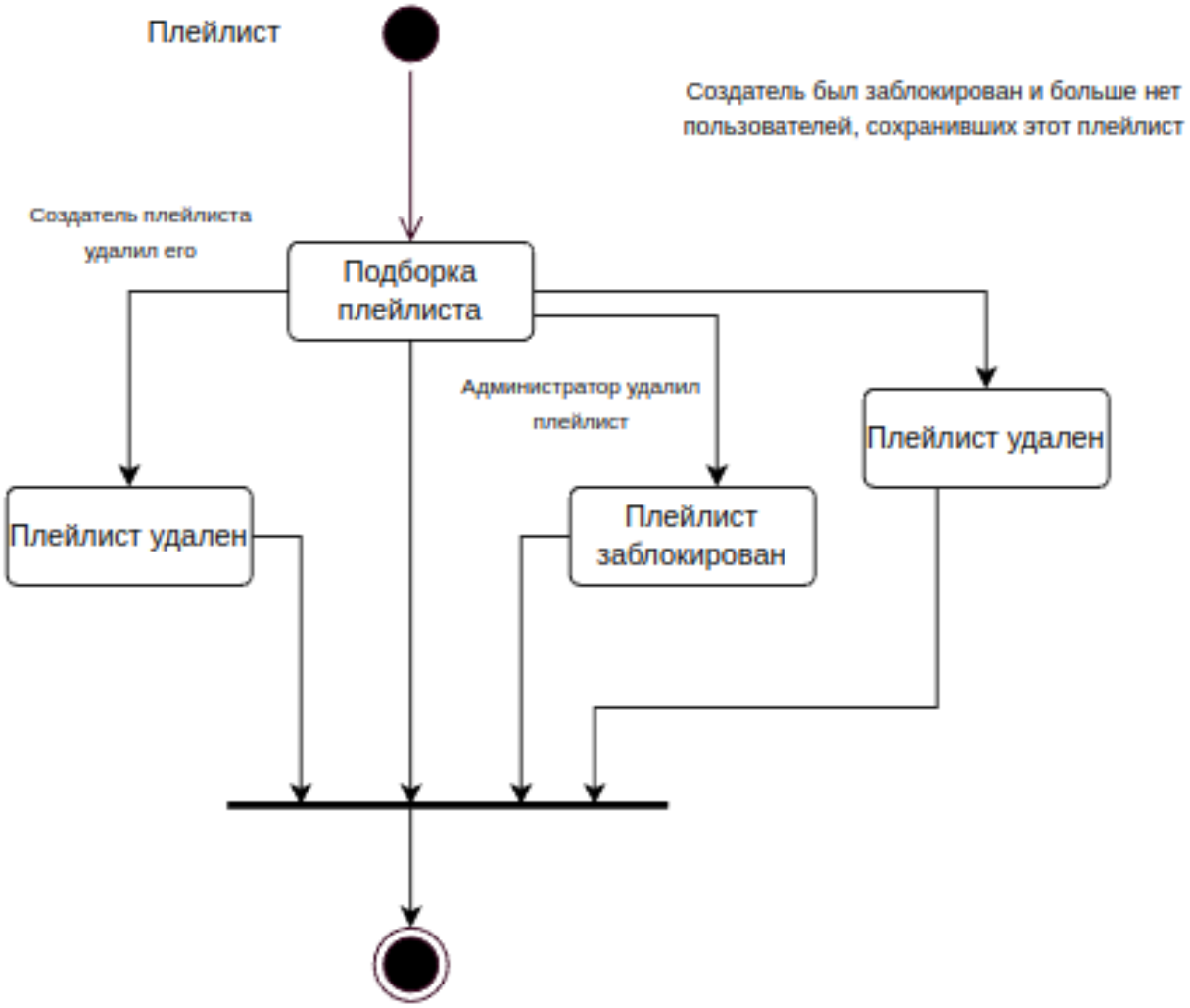

Рисунок 6 - Диаграмма состояний подборки

Диаграмма состояний позволяет визуализировать и моделировать поведение объекта или системы в различных состояниях. Она позволяет описать все возможные состояния объекта, а также переходы между ними в ответ на определенные события.

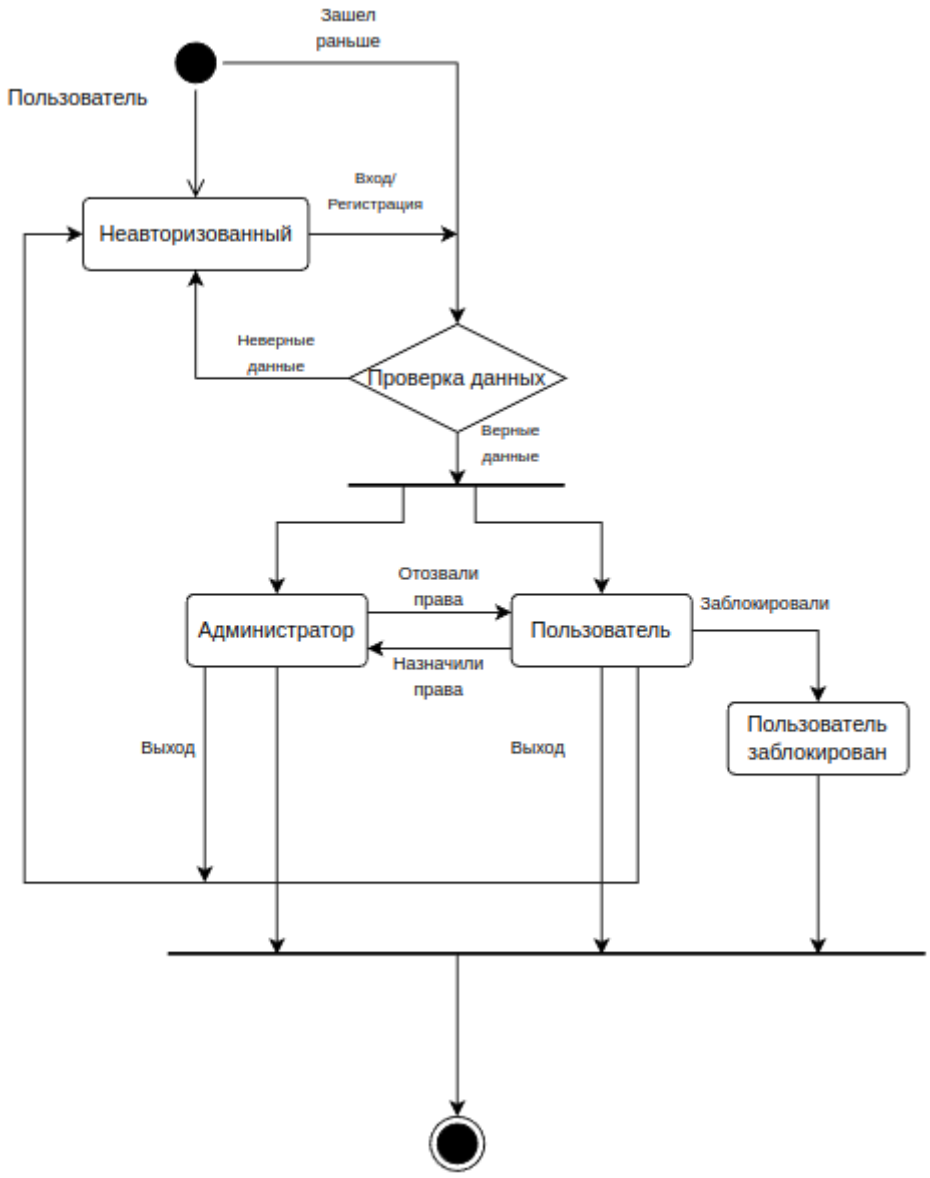

Рисунок 7 - Диаграмма состояний пользователя

3.4 Диаграмма деятельности

Благодаря диаграмме деятельности можно увидеть действия, состояния которых описаны на диаграмме состояний во время процесса взаимодействия пользователя с приложением на этапах создания новой подборки с фильмами.

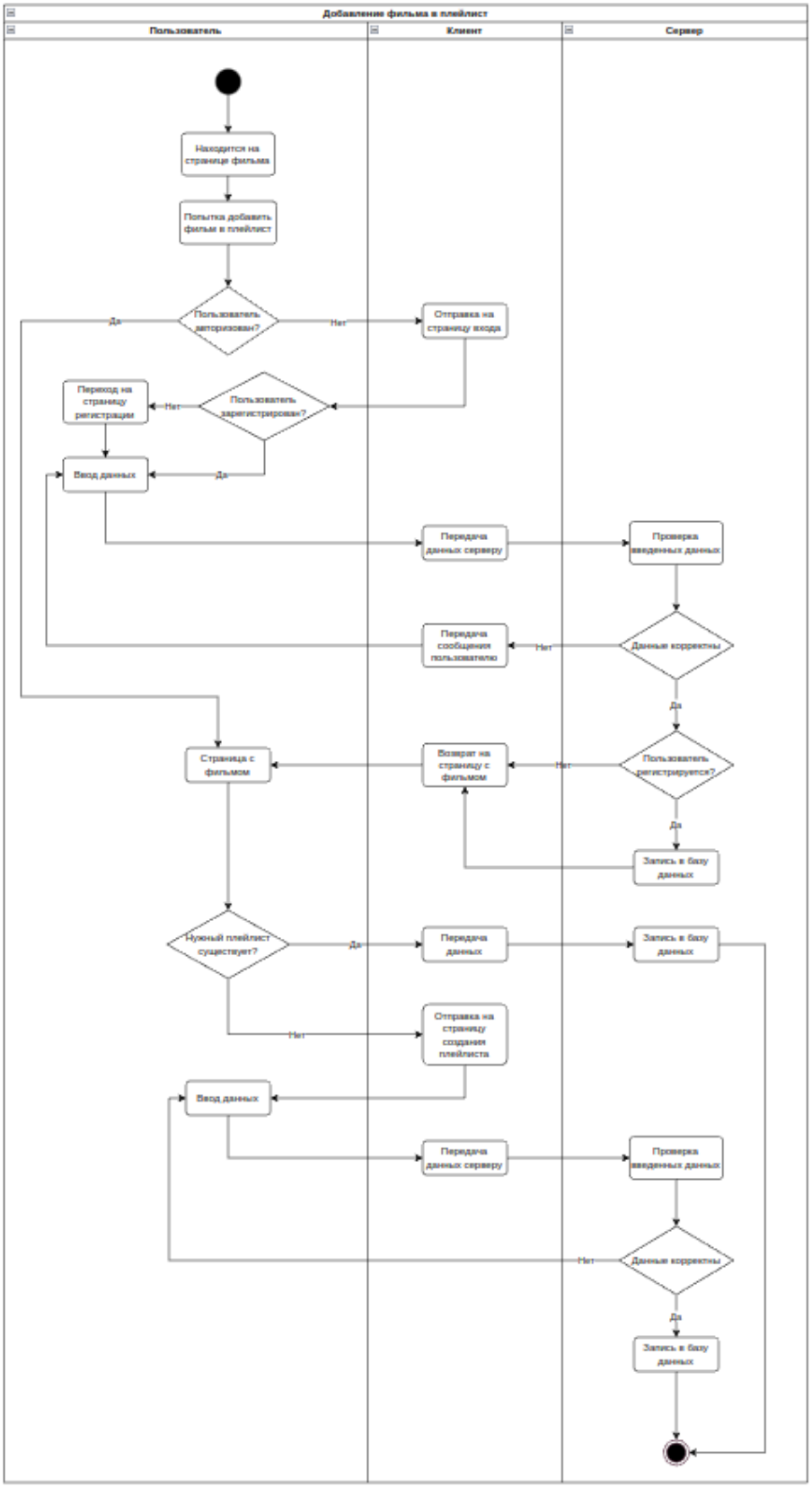

Рисунок 8 - Диаграмма деятельности

3.5 Диаграмма классов сущностей

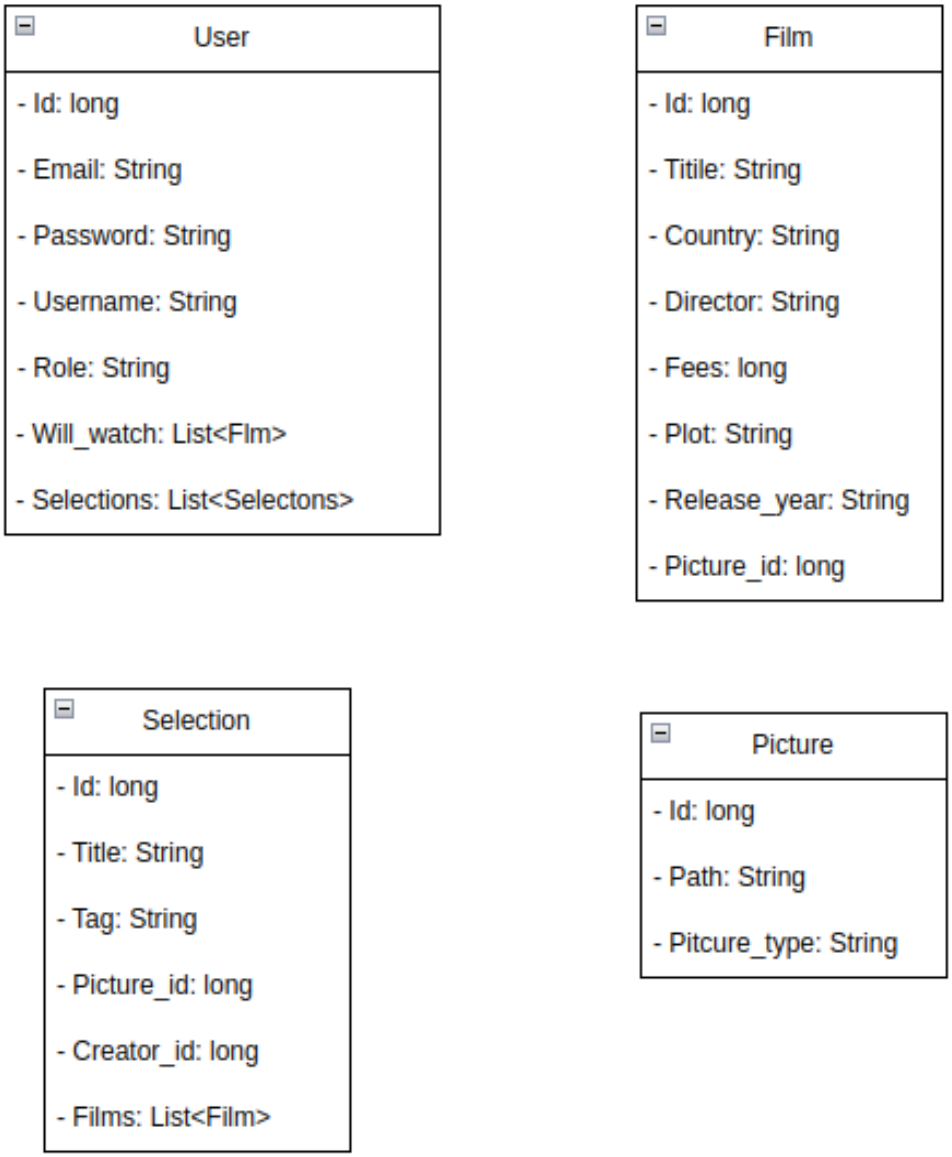

Рисунок 9 - Диаграмма классов сущностей

3.6 Диаграмма объектов

Диаграмма классов представляет описание структуры классов в системе и их взаимосвязи. Она отображает как статические аспекты системы, включая классы, атрибуты и методы, а также динамические аспекты, такие как связи между объектами и выполнение методов во время выполнения программы

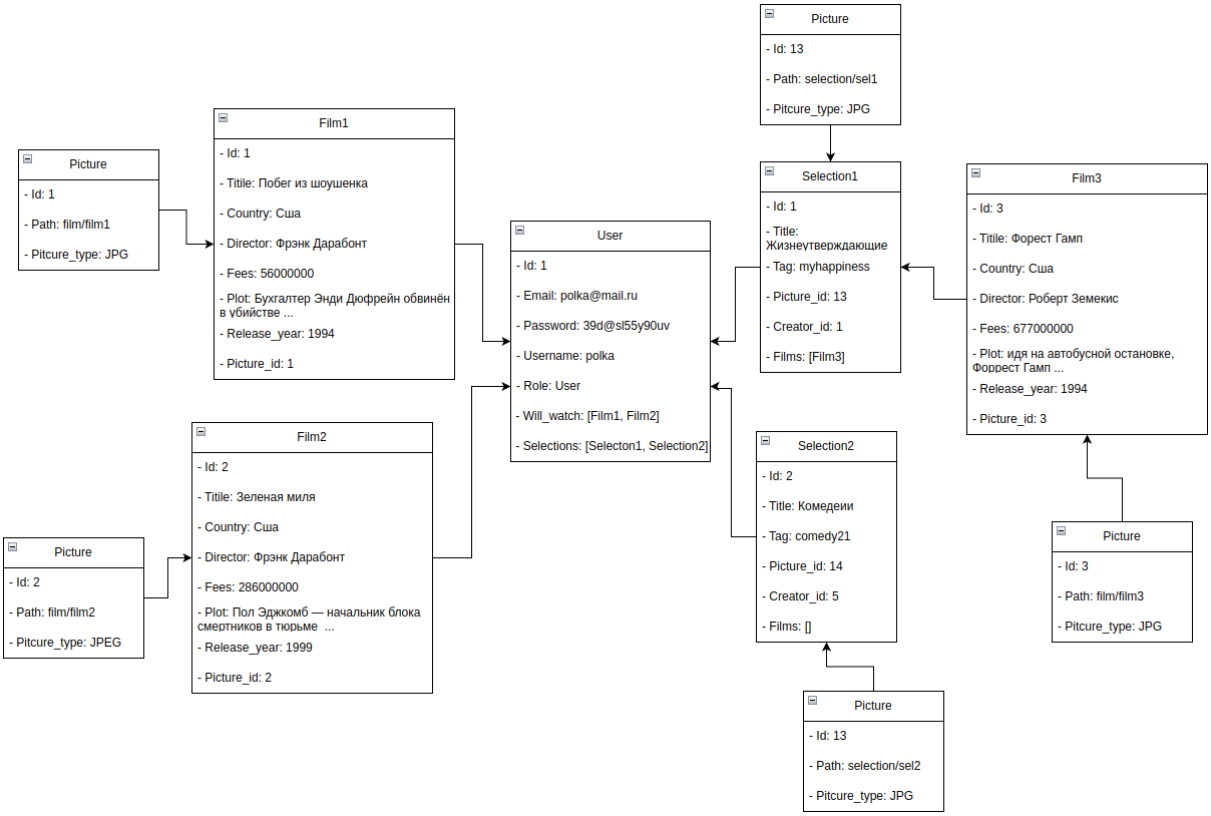

Рисунок 10 - Диаграмма объектов

4 Реализация

4.1 Средства реализации

В качестве средств реализации мобильного приложения были выбраны:

Для сервера:

— Язык программирования Java 17 версия;

— Фреймворк Spring Boot;

— СУБД PostgreSQL;

— Система автоматизации сборки проектов Apache Maven;

— Инструмент для создания документации API Swagger;

— Yandex Object Storage;

— Docker, для развёртывания приложения.

Для реализации клиентской части были выбраны:

— Язык программирования Dart;

— Фреймворк Flutter.

Сочетание языка программирования Java и Spring Boot было выбрано из-за большого выбора дополнительных библиотек, возможности быстро разворачивать разрабатываемое приложение.

СУБД PostgreSQL обладает рядом плюсов, среди которых можно особенно выделить открытый исходный код, свободную лицензию и широкую функциональность.

Загруженные пользователем фотографии хранятся в облачном хранилище от Yandex, взаимодействие с которым происходит при помощи Amazon S3 SDK для Java. Выбор данного сервиса для сохранения фото обусловлен возможностью его автоматического масштабирования и удобством взаимодействия.

4.2 Языковые версии приложения

Приложение реализовано с поддержкой русского языка.

4.3 Реализация серверной части приложения

4.3.1 Архитектура серверной части приложения

Для исполнения серверной части используем Spring Boot фреймворк для разработки веб-приложений, который облегчает создание приложений на основе архитектуры MVC (Model-View-Controller). MVC — это популярный шаблон проектирования, который разделяет приложение на три основных компонента: модель (Model), представление (View) и контроллер (Controller).

Модель представляет собой данные приложения и логику их обработки.

Представление отвечает за отображение данных пользователю.

Контроллер обрабатывает запросы от пользователя, связывает модель и представление, и определяет логику обработки запросов.

К плюсам MVC архитектуры можно отнести:

— Разделение ответственности между моделью, представлением и контроллером;

— Улучшенная читаемость и поддержка кода;

— Легкая замена или модификация компонентов без изменения других частей приложения;

— Улучшенная масштабируемость и возможность повторного использования компонентов.

Таким образом, Spring Boot и MVC архитектура предоставляют удобный и эффективный способ разработки веб-приложений, позволяя разделить логику, данные и представление, и обеспечивая удобство использования и поддержки кода.

4.3.2 Диаграмма классов сущностей

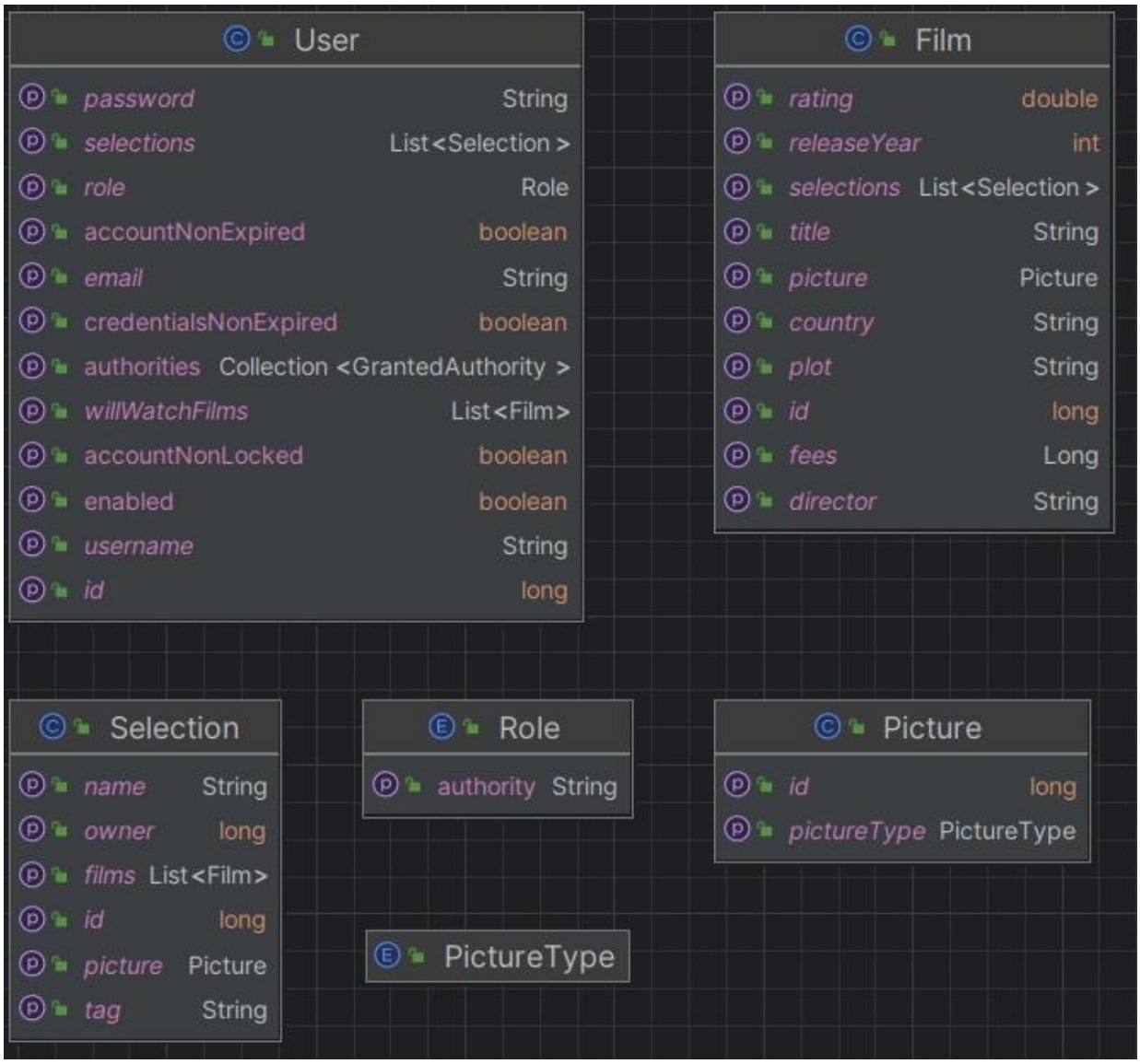

Рисунок 11 - Диаграмма классов сущностей

Слой моделей является ключевой частью трехслойной архитектуры веб-приложения на основе фреймворка Spring Boot и паттерна MVC. В этом слое определены модели данных, которые представляют бизнес-объекты или сущности, с которыми работает приложение. Слой моделей отвечает за работу с данными. На рисунке ниже можно увидеть реализацию моделей.

4.3.3 Слой доступа к данным

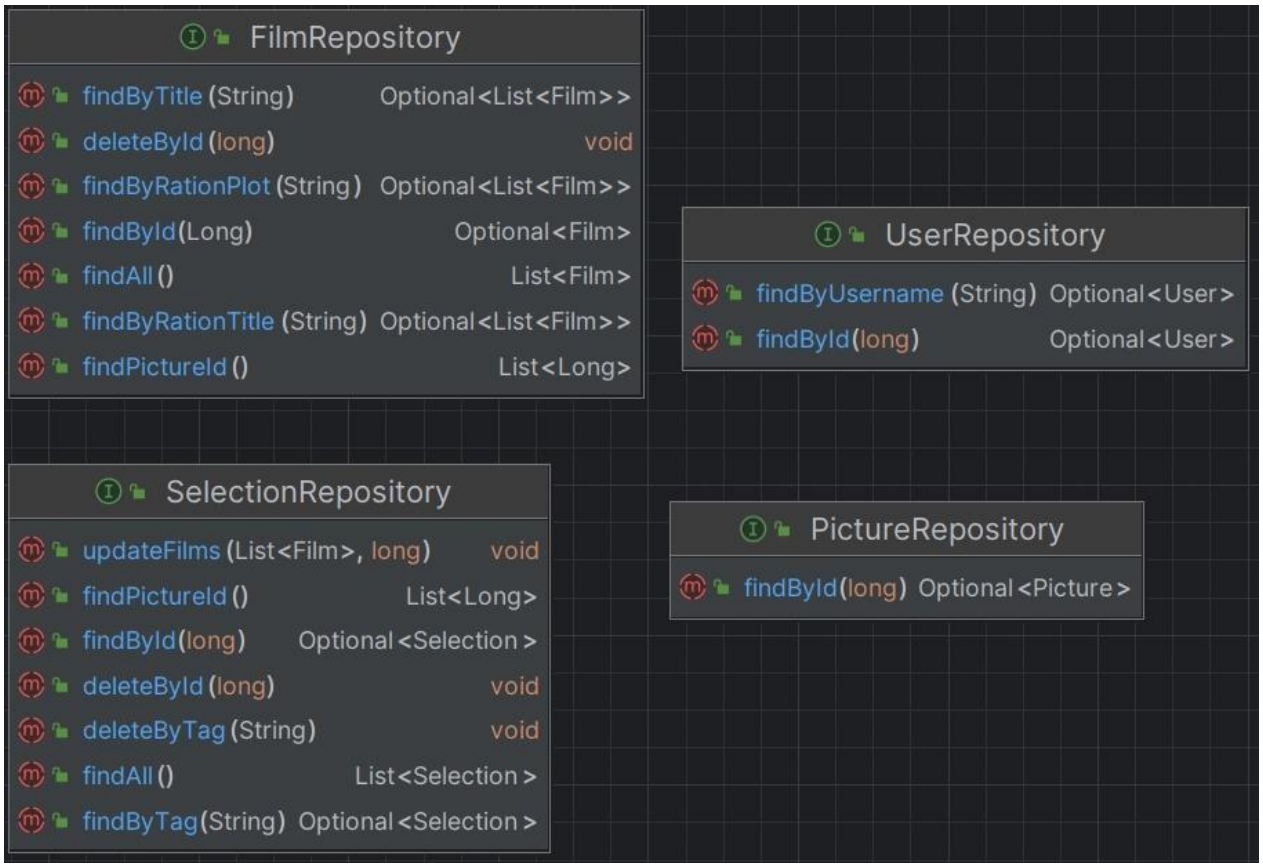

Рисунок 12 - Реализация интерфейсов

Слой доступа к данным отвечает за взаимодействие с базой данных или другими источниками данных. В приложении используется Spring Data JPA, что позволяет нам упростить доступ к данным и уменьшить объем написания повторяющегося кода. Для каждой сущности в приложении был создан соответствующий репозиторий, который наследуется от интерфейса JpaRepository.

JpaRepository предоставляет широкий набор методов для основных операций доступа к данным, включая сохранение, поиск по id и т.д.

4.3.4 Слой контроллеров

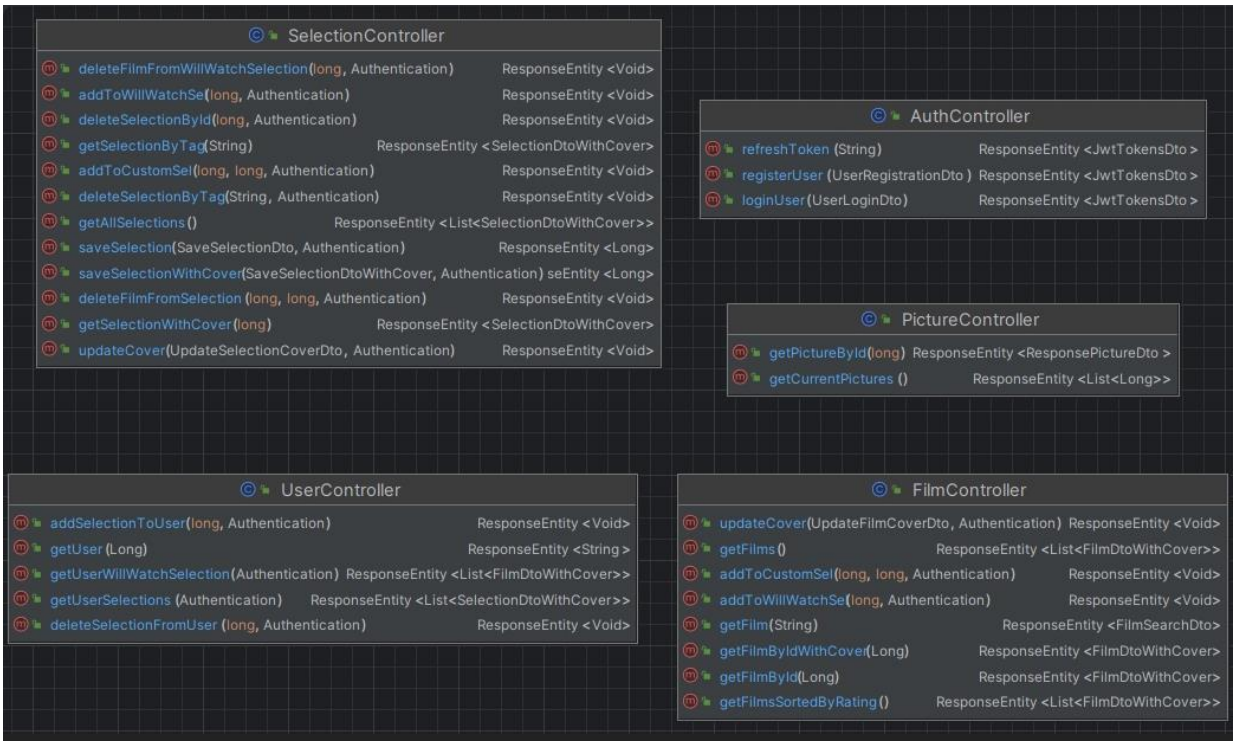

Рисунок 13 - Реализация контроллеров

Слой контроллеров отвечает за обработку входящих запросов от клиентов на определенный маппинг и возвращает ответ в виде ResponceEntityRestAPI.

Для каждой из сущностей были написаны контроллеры, методы которых отвечают за необходимые приложению действия с этими сущностями.

4.3.5 Слой бизнес-логики

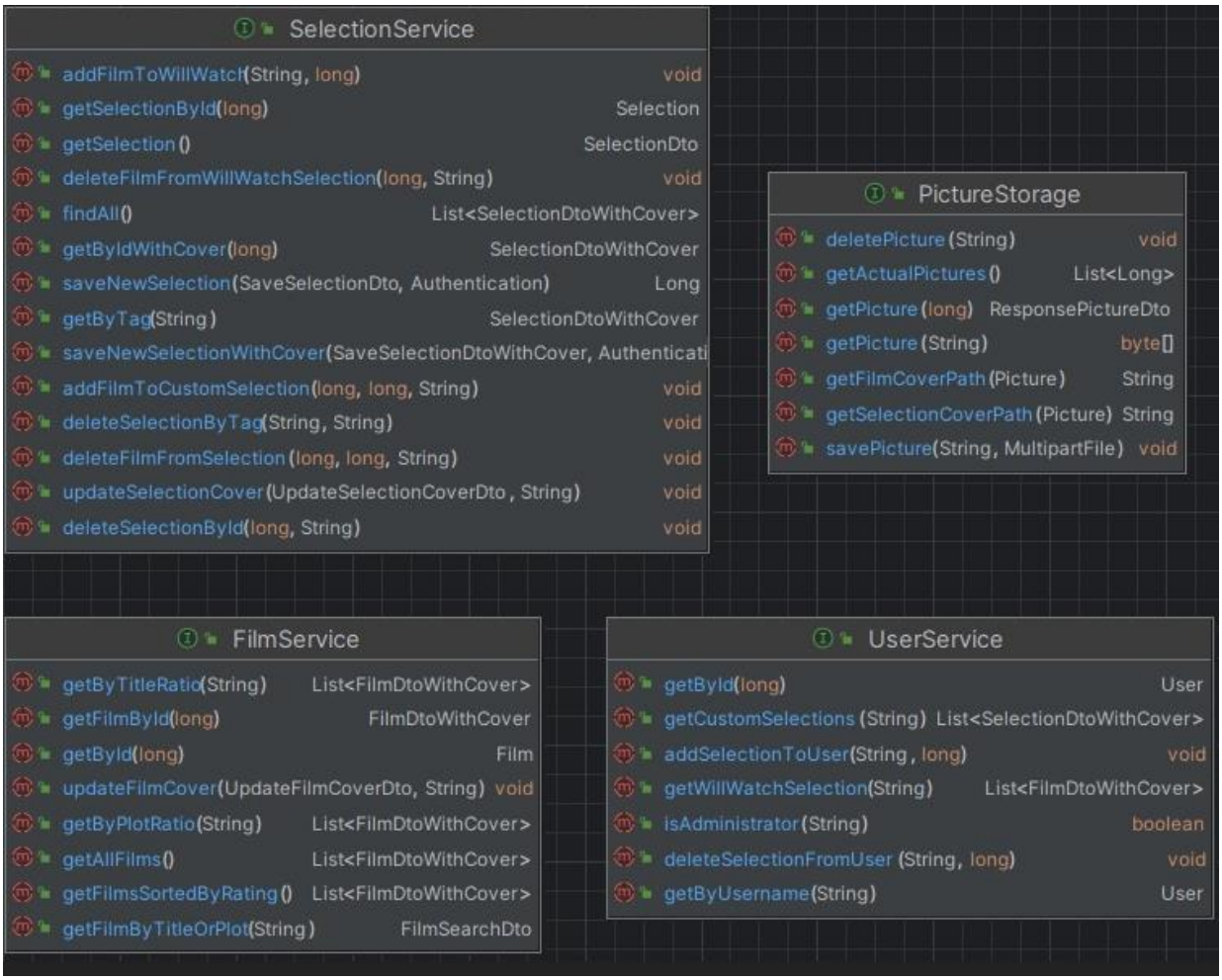

Рисунок 14 - Реализация бизнес-логики

Вся бизнес-логика реализована в service слое. Слой сервисов абстрагирует бизнес-логику от слоя контроллеров и доступа к данным в базе данных, поэтому чаще всего методы просто передают аргумент, полученный из контроллера, вызвавшего его, в соответствующий метод в репозитории, затем возвращают ответ в контроллер, или собирают объект из полученных аргументов и также передают его в репозиторий, а полученный ответ затем возвращают контроллеру для дальнейшего использования данных.

4.4 Реализация клиентской части приложения

4.4.1 Описание главной страницы

При загрузке главной страницы сайта появляются:

— Слева располагается основное навигационное меню;

— В верхней части располагается строка поиска фильмов;

— Чуть ниже строки поиска находится каталог фильмов;

— Справа вверху находится кнопка «Войти», которая переместит пользователя на страницу авторизации.

У авторизованного пользователя и администратора вместо кнопки «Войти» будет «Выйти».

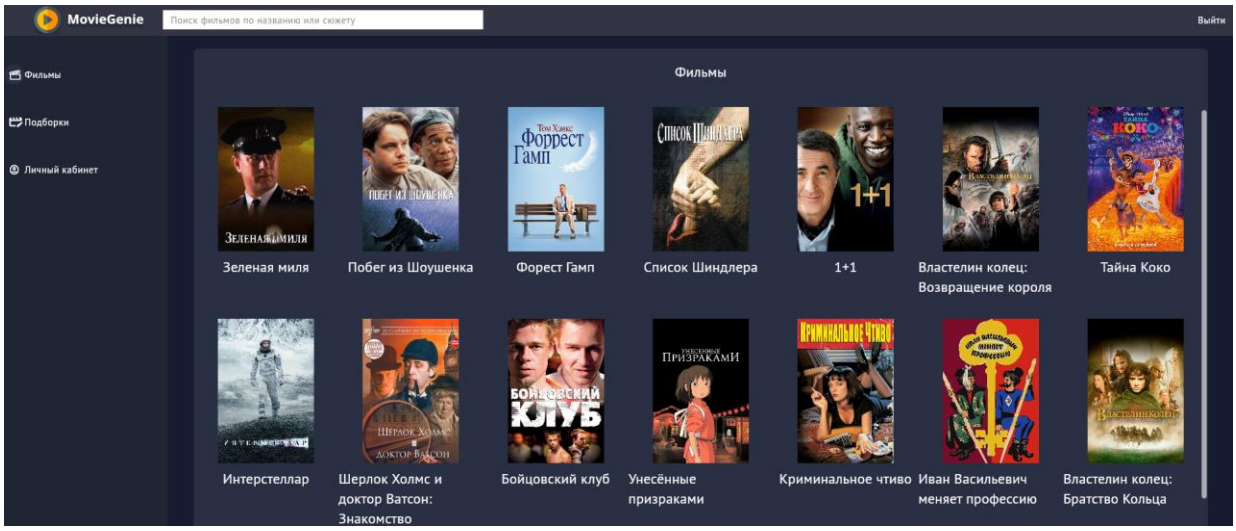

Рисунок 15 - Главная страница

4.4.2 Описание главной страницы при поиске

При поиске фильма на главной странице появится фильм, который ищет пользователь.

При нажатии на заставку или название фильма, пользователь перейдёт на страницу фильма.

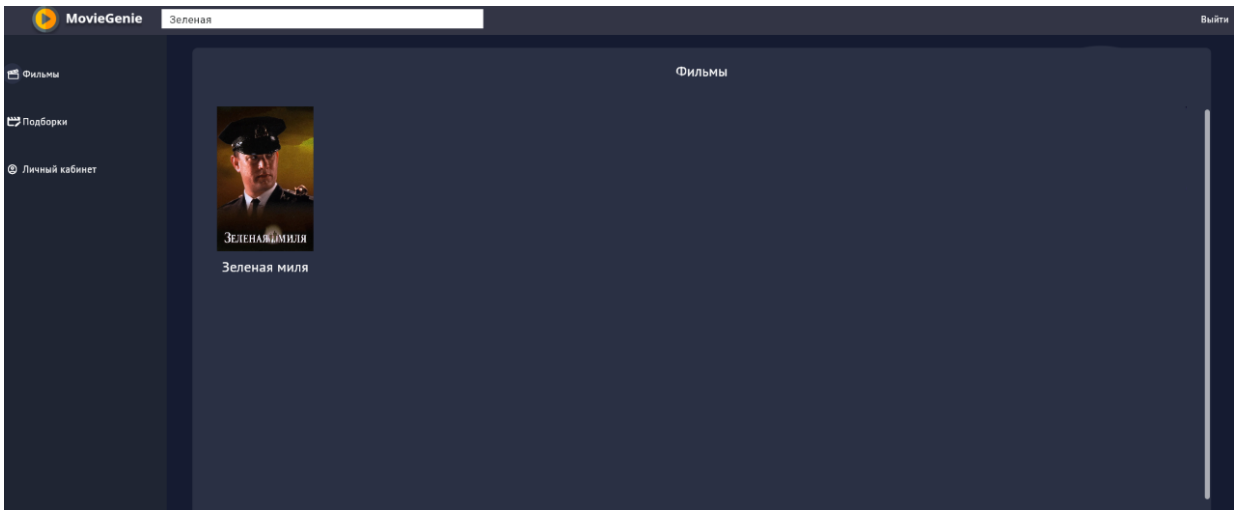

Рисунок 16 - Главная страница при поиске

4.4.3 Описание страницы выбранного фильма

При переходе на страницу фильма появится выбранный фильм с различными характеристиками:

— Название фильма;

— Год;

— Страна производства;

— Режиссер;

— Сборы в мире;

— Сюжет;

— Заставка фильма.

У авторизованного и неавторизованного пользователя на странице выбранного фильма будет кнопка «Буду смотреть», при нажатии на которую авторизованный пользователь сможет добавить выбранный фильм в личный кабинет.

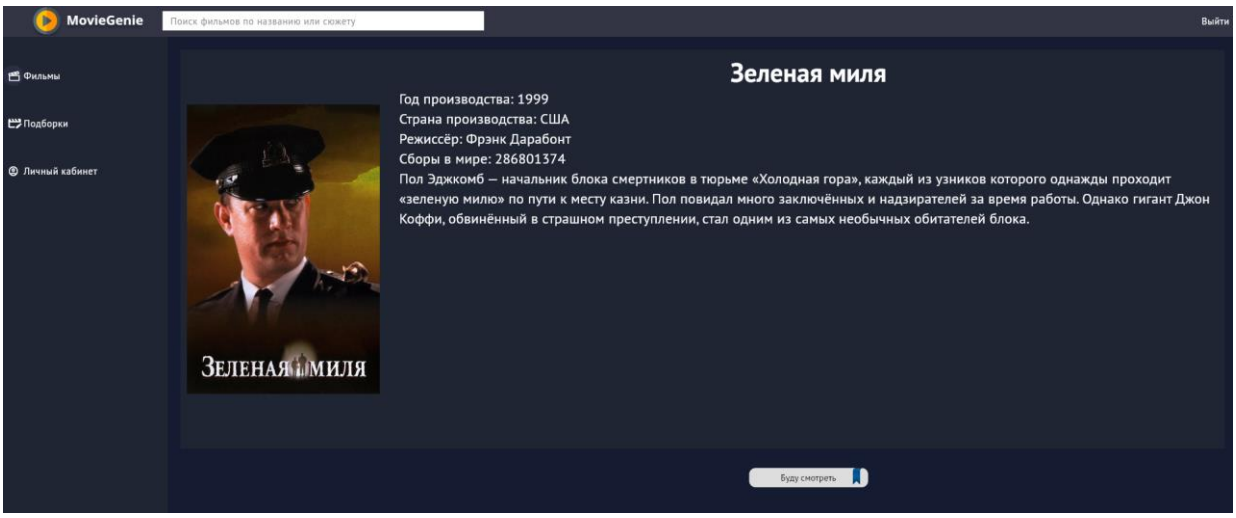

Рисунок 17 - Страница выбранного фильма

У неавторизованного пользователя данный функционал не будет работать, выдавая сообщение с просьбой авторизоваться.

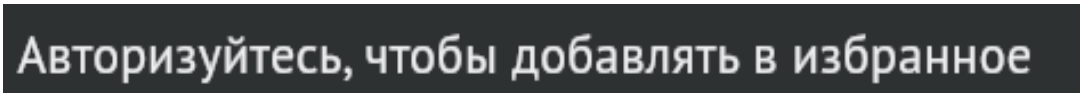

Рисунок 18 - Сообщение с просьбой авторизоваться

4.4.4 Описание страниц авторизации и регистрации

Страница авторизации располагается по центру сверху и имеет следующие элементы:

— Поле ввода «Логин»;

— Поле ввода «Пароль»;

— Кнопка «Войти»;

— Кнопка «Зарегистрироваться»;

— Кнопка «Назад».

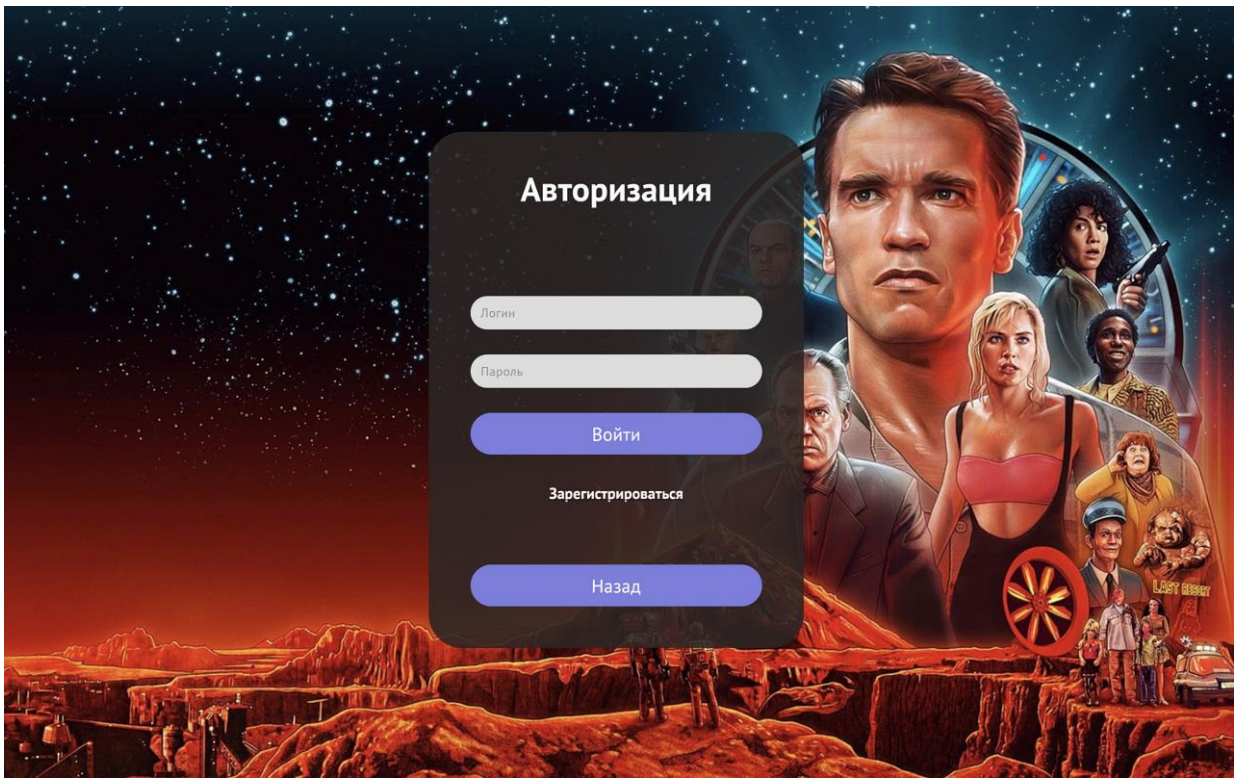

Рисунок 19 - Страница авторизации

Страница регистрации располагается по центру сверху и имеет следующий функционал:

— Поле ввода «Email»;

— Поле ввода «Логин»;

— Поле ввода «Пароль»;

— Поле ввода «Повторить пароль»;

— Кнопка «Зарегистрироваться»;

— Кнопка «Войти» с ссылкой на страницу авторизации.

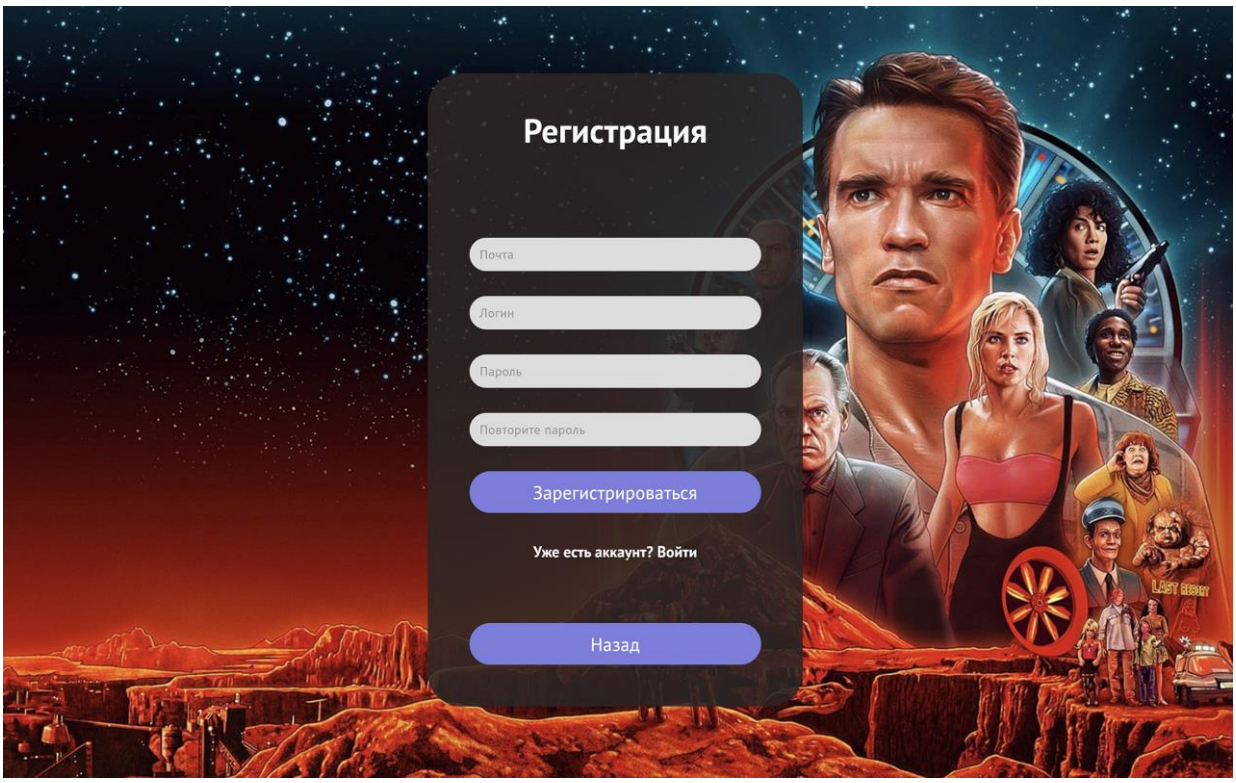

Рисунок 20 - Страница регистрации

4.4.5 Описание страницы с подборками фильмов

У неавторизованного пользователя есть возможность только просматривать подборки.

У авторизованного пользователя функционал разрешает добавлять подборки в личный кабинет.

Администратор может использовать вышеперечисленный функционал, а также удалять подборки, созданные пользователями.

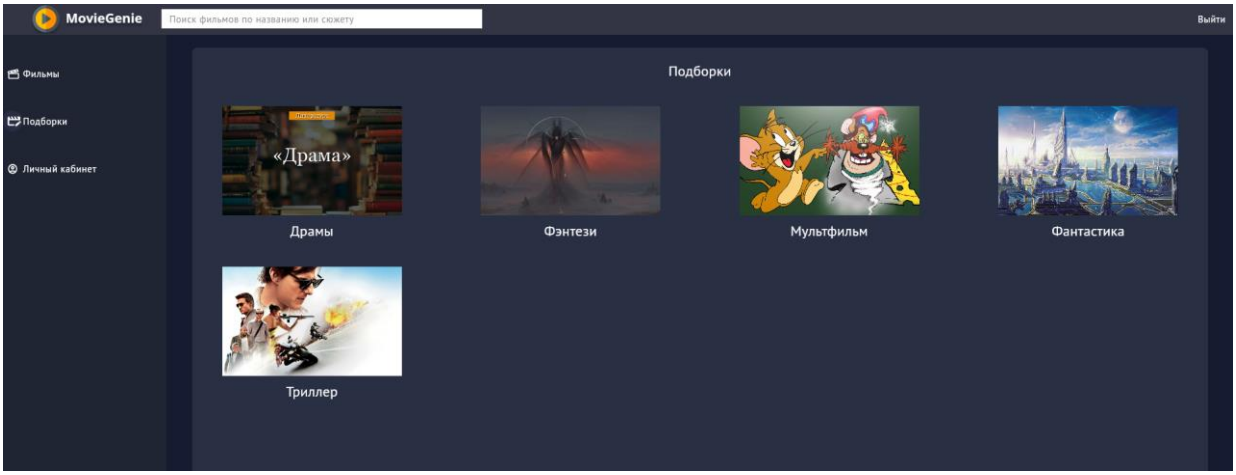

Рисунок 21 - Страница с подборками фильмов

4.4.6 Описание страницы конкретной подборки фильмов

При загрузке страницы с конкретной подборкой появляются фильмы, входящие в неё. Также снизу появляются кнопка «Добавить подборку», которая активна у авторизованного пользователя и администратора.

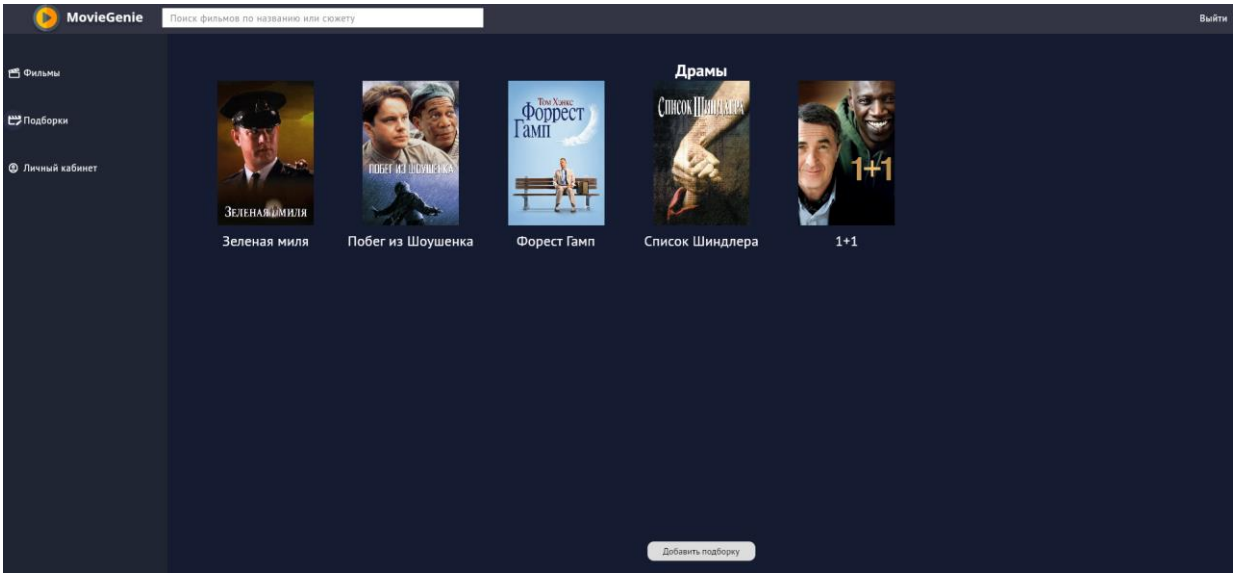

Рисунок 22 - Страница конкретной подборки фильмов

При попытке добавить подборку неавторизованным пользователем, будет выведено сообщение с просьбой авторизоваться.

4.4.7 Описание страницы личного кабинета

У авторизованного пользователя в личном кабинете есть возможность просмотра добавленных и созданных подборок фильмов, а также просмотр фильмов, добавленных в «Буду смотреть».

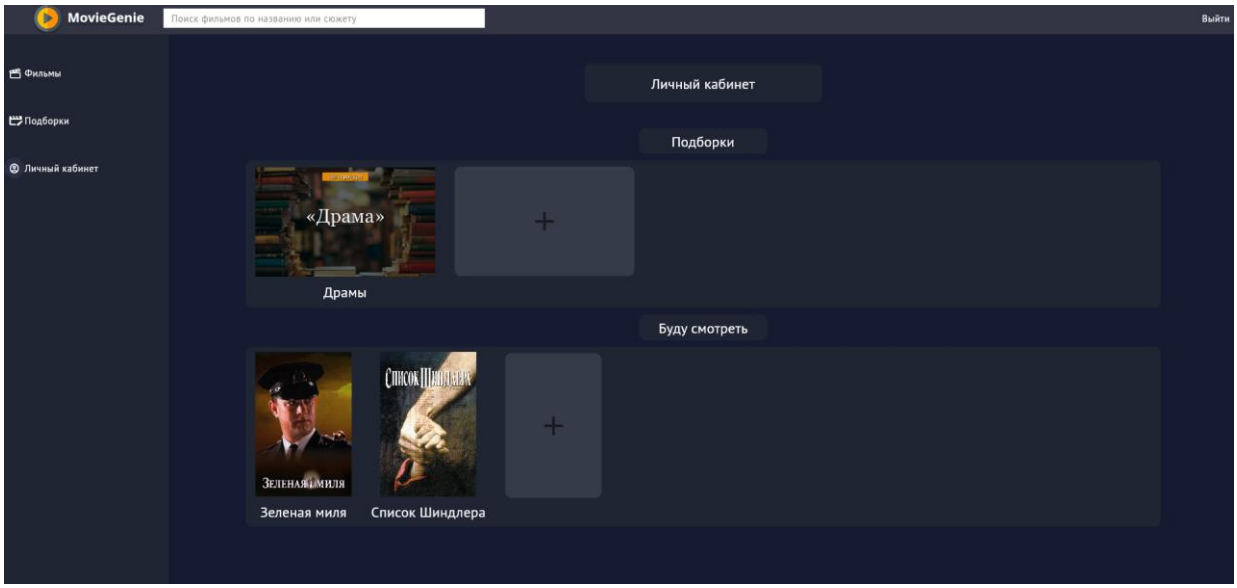

Рисунок 23 - Страница личного кабинета

Неавторизованному пользователю доступ в личный кабинет будет закрыт. Для перехода необходимо быть авторизованным.

Заключение

В рамках данной курсовой работы было разработано приложение, которое позволяет пользователям искать интересующие их фильмы, сохранять понравившиеся подборки, а также создавать свои.

В начале разработки был проведен анализ предметной области, определены основные требования к разрабатываемой системе, определены основные сценарии приложения и пользовательские истории.

В процессе работы были реализованы следующие задачи проекта:

— Поиск фильма по названию;

— Поиск фильма по содержанию;

— Просмотр информации о фильме;

— Добавление фильма в список “Буду смотреть”;

— Просмотр пользовательских подборок фильмов;

— Сохранение подборок других пользователей;

— Создание своих подборок;

— Редактирование своих подборок;

— Добавление фильмов в созданную подборку;

— Удаление фильма из созданной подборки;

— Удаление своих подборок.

Список используемых источников

1. Документация языка программирования Java [Электронный ресурс]. – Режим доступа: https://docs.oracle.com/javame/8.3/index.html
2. Документация SpringBoot [Электронный ресурс]. – Режим доступа: URL: https://docs.spring.io/spring-boot/docs/current/reference/html/ – Заглавие с экрана. – (Дата обращения 22.04.2023).
3. Документация Swager [Электронный ресурс]. – Режим доступа: URL: https://swagger.io/docs/ - (Дата обращения 10.05.2023).
4. Вигерс К. Разработка требований к программному обеспечению/ Карл Вигерс, Джой Бити. — М.: Изд-во Русская редакция, 2014. — 736 с.
5. The Prosand Cons of Online Booking Systems [Электронный ресурс]. – Режим доступа: URL: https://miyn.app/the-pros-and-cons-of-online-booking-system/ – Заглавие с экрана. – (Дата обращения 21.05.2023).
6. Документация библиотеки Liquibase [Электронный ресурс]. – Режим доступа: https://docs.liquibase.com/home.html
7. Google Flutter Mobile Development Quick Start Guide by Prajyot Mainkar Salvatore Giordano – 2019.
8. Buckett C. Dart in action. – Simon and Schuster, 2013.