

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ
И КОМПЬЮТЕРНОЙ ТЕХНИКИ
МЕГАФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И
УПРАВЛЕНИЯ

**Создание информационной системы для
каталогизации цифровой фильмотеки
«MyFilmList»**
курсовая работа

*по дисциплине «Информационные системы»
направления бакалавриата 09.03.04 «Программная инженерия»,
образовательное направление «Системное и прикладное программное
обеспечение»*

Научный руководитель:

Егошин Алексей Васильевич

Работу выполнили:

Студенты 3 курса

Группы Р3315

Барсуков Максим Андреевич

Горляков Даниил Петрович

Оглавление

Этап 1.....	2
Задание.....	2
1. Согласовать с преподавателем предметную область, для которой будет разрабатываться информационная система.....	2
2. Составить подробное текстовое описание предметной области.....	2
3. Сформулировать, зачем нужна информационная система для представленной предметной области, какие задачи она позволит решить.....	6
4. Составить функциональные/нефункциональные требования к разрабатываемой информационной системе.....	8
5. Построить модели основных прецедентов, составить их описание.....	12
6. Предложить архитектуру будущей системы.....	19
Вывод.....	21
Этап 2.....	22
Задание.....	22
1. ER-модель.....	23
2. Даталогическая модель.....	23
3. Реализация даталогической модели в реляционной СУБД PostgreSQL.....	24
4. Триггеры.....	26
5. Скрипты для создания, удаления БД, заполнения базы тестовыми данными.....	27
6. PL/pgSQL-функции и процедуры для выполнения критически важных запросов.	29
7. Индексы.....	30
Вывод.....	31
Этап 3.....	32
Задание.....	32
1. Диаграммы классов.....	33
2. Реализация.....	49
Вывод.....	49
Этап 4.....	50
Вывод.....	50
Приложение А.....	52

Этап 1

Задание

1. Согласовать с преподавателем предметную область, для которой будет разрабатываться информационная система.

Согласовано посредством кроссплатформенного мессенджера [Telegram](#).

2. Составить подробное текстовое описание предметной области.

MyFilmList — это информационная система для управления и организации коллекции фильмов, сериалов и TV-шоу. Она позволяет пользователям вести собственные списки просмотренного, запланированного к просмотру контента, оценивать фильмы, оставлять отзывы и находить новые произведения для просмотра по рекомендациям, жанрам и рейтингам. Основные элементы системы включают в себя информацию о фильмах (название, жанры, режиссеры, актеры, описание сюжета, оценки, рецензии), профили пользователей, инструменты для поиска и фильтрации контента, а также инструменты участия в социальном взаимодействии с другими пользователями через рецензии и комментарии.

Ключевые аспекты предметной области:

1. Фильмы и сериалы:

- **Фильм** — это произведение аудиовизуального искусства с определенной длительностью и структурой. Каждое произведение имеет ключевые атрибуты: название, дату выхода, жанры, режиссера, актеров, описание сюжета и технические характеристики (например, продолжительность).
- **Сериал** — это телевизионное шоу, которое состоит из нескольких эпизодов, объединенных общим сюжетом или персонажами. Сериал

также имеет ключевые атрибуты: название, количество сезонов и серий, дата начала показа, жанры и основные участники (режиссеры, актеры и сценаристы).

- Фильмы и сериалы имеют метаданные, таких как рейтинги пользователей, отзывы, трейлеры, постеры, которые помогают пользователям ориентироваться в большом объеме контента.

2. Пользователи системы:

- **Зарегистрированные пользователи** — это ключевые участники системы, которые могут создавать учетную запись, добавлять фильмы в свои списки, оставлять оценки и рецензии, участвовать в обсуждениях и получать рекомендации. Каждый пользователь имеет свой профиль, где отображаются его оценки и предпочтения.
- **Модераторы** — это пользователи с расширенными правами, которые контролируют содержание системы. Они могут управлять отзывами, комментариями и следить за соблюдением правил сообщества.
- **Администраторы** — пользователи, которые имеют полный доступ к управлению системой, включая добавление новых фильмов и сериалов, управление модераторами.

3. Списки фильмов:

- Каждый пользователь может составлять несколько списков для организации фильмов и сериалов по категориям: "Просмотрено", "Запланировано", "Избранное" и др. Списки помогают пользователям отслеживать, что они уже посмотрели, что собираются смотреть, и фильтровать контент по личным предпочтениям.
- Списки могут быть публичными или приватными. Публичные списки позволяют другим пользователям видеть вкусы и предпочтения человека, а приватные дают возможность вести личные записи.

4. Рейтинги и рецензии:

- **Рейтинги** — это числовое выражение мнения пользователя о фильме или сериале, обычно по шкале от 1 до 10. Средний рейтинг фильма или сериала формируется на основе всех оценок,

оставленных пользователями, и помогает другим участникам оценить качество произведения перед его просмотром.

- **Рецензии** — это текстовые обзоры, в которых пользователи выражают свои мысли и впечатления от просмотра. Рецензии помогают другим пользователям понять, стоит ли им смотреть конкретный фильм или сериал.

5. Жанры:

- Каждый фильм и сериал может быть отнесен к нескольким жанрам, таким как "Драма", "Комедия", "Фантастика", "Ужасы" и т.д. (жанры могут быть изменены и расширены администратором) Жанры помогают пользователям фильтровать контент и находить фильмы по своему вкусу.
- Пользователи могут искать фильмы по жанрам, создавая списки лучших произведений в каждом из них и делясь своими предпочтениями с другими.

6. Поиск и фильтрация:

- В системе реализованы инструменты поиска и фильтрации, которые позволяют пользователям находить фильмы и сериалы по различным параметрам, включая жанры, год выхода, оценки, популярность и рекомендации.
- Поиск по актеру или режиссеру также помогает находить фильмы, в которых снимались любимые звезды.

7. Рекомендации:

- Система предоставляет рекомендации на основе ранее просмотренных фильмов и оценок пользователя.
- Рекомендации помогают пользователям открывать для себя новые произведения, которые они могли бы упустить.

8. Комментарии:

- Зарегистрированные пользователи могут оставлять комментарии к рецензиям других пользователей, обсуждать фильмы и сериалы, делиться своими мнениями о сюжетах, актерах и т.д. (структура комментария определяется пользователем, который пишет комментарий)

- Комментарии обеспечивают интерактивное взаимодействие между пользователями, создавая более живое и активное сообщество.

9. Администрация и модерирование:

- Для обеспечения порядка в системе необходимо наличие модераторов, которые следят за качеством публикуемых рецензий и комментариев. Они контролируют соблюдение правил, удаляют оскорбительные материалы и блокируют нарушителей.
- Администраторы системы управляют глобальными аспектами платформы, добавляют и редактируют информацию о фильмах и сериалах, следят за обновлениями базы данных.

Основные сущности в базе данных:

Пользователь (User), Профиль пользователя (UserProfile), Фильм/Сериал (Movie/TVShow), Жанр (Genre), Оценка (Rating), Рецензия (Review), Комментарий (Comment), Список фильмов (Watchlist), Категория фильма (MovieCategory), Тег фильма (MovieTag), Просмотренный фильм (WatchedMovie), Жалоба (Report), Роль пользователя (UserRole).

3. Сформулировать, зачем нужна информационная система для представленной предметной области, какие задачи она позволит решить.

Информационная система **MyFilmList** предназначена для решения нескольких важных задач, связанных с управлением и организацией информации о фильмах, сериалах и видеоконтенте. Она помогает пользователям отслеживать свой просмотренный контент, вести списки фильмов, делиться мнениями с другими участниками сообщества и находить новые произведения для просмотра. Система позволяет создавать единое место для взаимодействия пользователей, любящих кино, и предлагает множество функций для повышения удобства использования платформы.

Основные задачи, которые решает MyFilmList:

1. Упорядочение и управление личными предпочтениями:

- Пользователи могут вести учет просмотренных фильмов и сериалов, создавать списки того, что они планируют посмотреть, что уже оценено, и что им понравилось. Это помогает лучше управлять своим контентом и не забывать о произведениях, которые пользователи хотели бы посмотреть.

2. Легкий доступ к информации о фильмах и сериалах:

- Система предоставляет подробные данные о фильмах и сериалах, включая описание, жанры, информацию о режиссерах и актерах, а также оценки и рецензии других пользователей. Это позволяет быстро получить все необходимые сведения для принятия решения о просмотре.

3. Получение персонализированных рекомендаций:

- Система анализирует предпочтения пользователей и предлагает рекомендации на основе ранее просмотренных фильмов и оценок. Это помогает находить фильмы, которые могут соответствовать вкусам пользователя, облегчая выбор контента для просмотра.

4. Социальное взаимодействие:

- Пользователи могут делиться своим мнением о фильмах через рецензии и комментарии, а также обсуждать произведения с другими любителями кино. Это создает сообщество людей, которые могут обмениваться идеями и рекомендациями, что делает процесс выбора фильмов и сериалов более увлекательным.

5. Облегчение поиска нового контента:

- Инструменты поиска и фильтрации контента позволяют пользователям быстро находить фильмы и сериалы по жанрам, оценкам, годам выпуска и другим параметрам. Это экономит время и упрощает процесс поиска подходящих произведений для просмотра.

6. Каталогизация и хранение информации:

- MyFilmList сохраняет и структурирует информацию о большом объеме контента. Это важно для пользователей, которые хотят вести учет того, что они посмотрели, а также для разработчиков и модераторов, чтобы поддерживать систему в актуальном состоянии.

7. Повышение вовлеченности пользователей через отзывы и рейтинги:

- Оценки и рецензии помогают другим пользователям принимать решения о просмотре и в целом повышают вовлеченность участников платформы, создавая ценность как для самих пользователей, так и для системы в целом.

8. Поддержка и модерирование:

- Система позволяет модераторам контролировать качество контента, удалять неподобающие отзывы или комментарии и поддерживать платформу в хорошем состоянии, чтобы она оставалась полезной и безопасной для всех пользователей.

4. Составить функциональные/нефункциональные требования к разрабатываемой информационной системе.

Функциональные требования

Требования: Пользователи должны иметь возможность...

- **U000** Создать новую учетную запись, указав электронную почту и пароль.
- **U001** Войти в систему, используя свои учетные данные (электронную почту и пароль).
- **U002** Просматривать доступную информацию о фильмах[A7,A28], такую как название, описание, рейтинг, отзывы пользователей, на платформе без необходимости регистрации.
- **U003** Оценивать[A9] фильмы (только для зарегистрированных пользователей).
- **U004** Оставлять отзыв[A10] к фильму (только для зарегистрированных пользователей).
- **U005** Добавлять (а также удалять) фильм в список избранного (только для зарегистрированных пользователей).
- **U006** Создавать собственные списки (а также удалять и изменять их (менять описание и название списка)) и добавлять в них фильмы (только для зарегистрированных пользователей).
- **U007** Использовать поисковую строку для поиска фильмов по ключевым словам, названиям, категориям, тегам и жанрам.
- **U008** Фильтровать фильмы по критериям, таким как дата публикации, рейтинг, теги и информации, определенной на странице фильма [A28].
- **U009** Использовать расширенные[A16] параметры поиска для точного нахождения интересующего произведения.
- **U010** Просматривать персонализированную ленту на основе ранее просмотренных фильмов.
- **U011** Редактировать свой профиль: фотографию профиля, описание (bio), имя.
- **U012** Восстанавливать доступ к своей учетной записи, если забыли пароль, используя процедуру восстановления пароля.
- **U013** Просматривать истории просмотров и оценок, чтобы отслеживать свои предпочтения в фильмах.

- **U014** Делать списки фильмов общедоступными (доступны для просмотра любыми пользователями, включая неавторизованных) или приватными (доступные только пользователю, создавшему список, модератору и администратору).

Требования: Модератор сайта должен иметь возможность...

- **M000** Принимать решение о том, соответствует ли указанный отзыв или рецензия правилам и политике сайта, и исходя из этого определять дальнейшую реакцию на нарушение (удаление аккаунта пользователя и внесение email в черный список / ограничение на написание отзывов).
- **M001** Редактировать и/или удалять нежелательный и/или незаконный контент с платформы (отзывы, рецензии).
- **M002** Управлять профилями пользователей, включая удаление учетных записей, установку ограничений доступа (ограничение на написание отзывов).
- **M003** Отслеживать жалобы пользователей и реагировать на них с помощью инструментов для управления нарушениями (см. **M002**).

Требования: Администратор сайта должен иметь возможность...

- **A000** Управлять пользователями: Администратор должен иметь возможность просматривать, изменять или удалять учетные записи пользователей, включая возможность назначать или снимать роли (например, назначать модераторов).
- **A001** Управлять контентом: Администратор должен иметь доступ к редактированию и удалению любых данных, связанных с фильмами, сериалами, рецензиями, комментариями и другими материалами, чтобы поддерживать актуальность контента на платформе.
- **A002** Управлять модераторами: Администратор должен иметь возможность управлять правами модераторов (назначать и снимать полномочия, предоставлять доступ к определенным функциям).

Нефункциональные требования

Usability requirements

Требования к удобству использования сайта

- **N000** Компьютерная и мобильная версия сайта должны корректно работать на современных разрешениях экранов, таких как: 1280x720, 1920x1080.
- **N001** Отображение сайта с полностью работающим функционалом и без нарушения дизайна в следующих браузерах: Chrome 121+, Safari 16+, Mozilla Firefox 122+, Яндекс Браузер 23+.
- **N002** Должна быть реализована навигация, позволяющая пользователю найти нужный ему визуальный элемент в пределах 30 секунд.

Performance Requirements

Требования к производительности

- **N003** Время загрузки главной страницы не должно превышать 15 секунд; Время загрузки страницы фильма не должно превышать 10 секунд при скорости интернета не менее 15 Мбит/с.
- **N004** Платформа должна поддерживать одновременное обслуживание не менее 10 активных пользователей.
- **N005** Система должна поддерживать многопоточную обработку запросов для повышения производительности при большом объеме данных.

Safety Requirements

Требования к сохранности

- **N006** Резервное копирование данных должно выполняться ежемесячно с возможностью быстрого восстановления.
- **N007** Данные пользователей должны быть зашифрованы с использованием алгоритма bcrypt и храниться на серверах с сертификатом безопасности SSL.
- **N008** Система должна модерировать контент для предотвращения размещения

- запрещенного материала или нарушения авторских прав.

Software Quality Attributes

Атрибуты качества ПО

- **N009** Система должна вести логи ошибок и событий, в том числе для каждого пользователя.
- **N010** Версия сайта должна регулярно обновляться в зависимости от используемых библиотек и фреймворков..

Security Requirements

Требования к безопасности

- **N011** Система должна предоставлять безопасную передачу данных пользователей (использование протокола HTTPS)
- **N012** Контроль доступа к административным функциям сайта при помощи JWT.
- **N013** Система должна иметь механизм защиты от SQL-инъекций..

Other Requirements

Остальные требования

- **N014** Время ответа службы поддержки на запросы пользователей не должно превышать 240 часа в рабочее время, а процент решенных запросов - не менее 90%.
- **N015** Время обработки запросов модерации контента не должно превышать 360 часов в рабочее время.

5. Построить модели основных прецедентов, составить их описание.

Прецедент	U000
ID	1
Краткое описание	Пользователь создает новую учетную запись на платформе
Главный актер	Пользователь
Второстепенные актеры	Система
Предусловия	Пользователь не имеет учетной записи на платформе.
Основной поток	<p>1. Пользователь открывает страницу регистрации на платформе.</p> <p>2. Пользователь вводит свою электронную почту и пароль.</p> <p>3. Пользователь нажимает кнопку "Зарегистрироваться".</p> <p>4. Система проверяет введенные данные на корректность.</p> <p>5. Система создает новую учетную запись для пользователя и перенаправляет его на главную страницу.</p>
Альтернативный поток	Введенные данные не прошли проверку на корректность. Система отображает сообщение об ошибке и просит пользователя проверить данные и ввести их заново. Пользователь вводит данные заново и продолжает процесс регистрации.
Постусловия	Создана новая учетная запись для пользователя.

Таблица 1 — Прецедент U000.

Прецедент	U001
ID	2
Краткое описание	Пользователь входит в систему, используя свои учетные данные.
Главный актер	Неавторизованный пользователь.
Второстепенные актеры	Система
Предусловия	Пользователь имеет учетную запись на платформе.
Основной поток	<p>1. Пользователь открывает страницу входа на платформе.</p> <p>2. Пользователь вводит свою электронную почту и пароль.</p> <p>3. Пользователь нажимает кнопку "Войти".</p> <p>4. Система проверяет введенные данные на корректность.</p> <p>5. Система аутентифицирует пользователя и перенаправляет его на главную страницу.</p>
Альтернативный поток	Если данные введены некорректно, система информирует пользователя и предлагает ввести данные заново. Если пользователь не помнит свой пароль, он может воспользоваться функцией восстановления пароля, следуя инструкциям системы
Постусловия	Пользователь успешно авторизован на платформе или начал процесс восстановления пароля.

Таблица 2 — Прецедент U001.

Прецедент	U007
ID	3
Краткое описание	Пользователь осуществляет поиск видео по ключевым словам, названиям, категориям, тегам, жанрам и актерам на платформе.
Главный актер	Незарегистрированный пользователь, Неавторизованный пользователь, Авторизованный пользователь
Второстепенные актеры	Система
Предусловия	Карточка фильма расположена на платформе.
Основной поток	<p>1. Система запрашивает ввод критериев поиска: текстовый запрос (обязательно), категория, актеры (дополнительно).</p> <p>2. Пользователь вводит текстовый запрос в поисковую строку или выбирает категорию из предложенных.</p> <p>3. Система обрабатывает запрос и отображает результаты поиска, включая карточку фильма, соответствующие запросу, а также рекомендации.</p> <p>4. Пользователь просматривает результаты поиска и выбирает желаемую карточку фильма.</p> <p>5. Система открывает страницу фильма.</p> <p>6. Пользователь может выполнить дополнительные действия, такие как сохранить в “избранное” и/или оставить отзыв.</p>
Альтернативный поток	Если не найдено точного совпадения по результатам поиска, система выдает наиболее близкие по жанру и теме поиска видеоролики.
Постусловия	Пользователь успешно нашел искомый или похожие фильмы.

Таблица 3 — Прецедент U007.

Прецедент	U004
ID	4
Краткое описание	Пользователь оставляет отзыв о фильме на платформе.
Главный актер	Авторизованный пользователь
Второстепенные актеры	Система
Предусловия	Пользователь авторизован и просматривает карточку фильма.
Основной поток	<ol style="list-style-type: none"> Пользователь просматривает карточку фильма. Система отображает форму для ввода отзыва. Пользователь вводит текст отзыва. Пользователь нажимает кнопку "Отправить". Система сохраняет отзыв и отображает его в блоке отзывов.
Альтернативный поток	Если отзыв содержит запрещенную информацию, модератор скрывает публикацию.
Постусловия	Отзыв пользователя опубликован под видео или пользователь информирован о невозможности оставить комментарий.

Таблица 4 — Прецедент U004.

Прецедент	M000
ID	5
Краткое описание	Модератор проверяет контент на соответствие правилам и политике сайта, а также реагирует на жалобы пользователей.
Главный актер	Модератор
Второстепенные актеры	Пользователи, Система
Предусловия	Модератор авторизован в системе и имеет соответствующие права доступа к

	модерационным функциям.
Основной поток	<ol style="list-style-type: none"> Модератор получает репорт о необходимости модерации контента (например, жалоба от пользователя). Модератор просматривает контент, проверяет его на соответствие правилам и политике сайта. В случае нарушения правил, модератор принимает соответствующие меры: редактирует контент, удаляет его или блокирует аккаунт пользователя, загрузившего контент.
Альтернативный поток	Если модератору требуются дополнительные разъяснения по контенту, он может связаться с автором контента перед принятием решения о его блокировке.
Постусловия	Контент оставлен без изменений, удален или заблокирован.

Таблица 5 — Прецедент U000.

Прецедент	U012
ID	6
Краткое описание	Пользователь, которые имеет аккаунт и забыл пароль, хочет его восстановить.
Главный актер	Неавторизованный пользователь
Второстепенные актеры	Система
Предусловия	Пользователь имеет аккаунт в системе и знает, на какую почту он зарегистрирован. Пользователь имеет доступ к почте.
Основной поток	<ol style="list-style-type: none"> Пользователь нажимает "Забыли пароль?" Пользователь вводит свой email. Система отправляет ссылку для восстановления пароля на почту. Пользователь переходит по ссылке. Пользователь устанавливает новый пароль.

Альтернативный поток	Если пользователь ввел несуществующую в системе почту, система сообщит об этом.
Постусловия	Если пользователь ранее регистрировался на сайте, он восстановил доступ к системе; иначе не восстановил.

Таблица 6 — Прецедент U012.

Прецедент	A001
ID	7
Краткое описание	Администратор управляет фильмами и контентом.
Главный актер	Администратор
Второстепенные актеры	Система, Авторизованные пользователи
Предусловия	Администратор авторизован в системе и хочет обновить или удалить информацию о фильмах.
Основной поток	<ol style="list-style-type: none"> 1. Администратор заходит в админ-панель. 2. Администратор выбирает редактировать базу фильмов. 3. Администратор выбирает фильм 4. Администратор редактирует или удаляет информацию о фильме.
Альтернативный поток	Нет
Постусловия	Администратор изменил базу фильмов требуемым образом.

Таблица 7 — Прецедент A001.

Прецедент	U008
ID	8
Краткое описание	Фильтрация фильмов по критериям
Главный актер	Пользователь
Второстепенные актеры	Система
Предусловия	Пользователь хочет фильтровать фильмы по дате, рейтингу, жанру
Основной поток	<ol style="list-style-type: none"> 1. Пользователь задает критерии фильтрации (например, по жанру или рейтингу) 2. Система отображает фильмы, соответствующие фильтрам
Альтернативный поток	Нет
Постусловия	Пользователь нашел фильм среди отфильтрованных

Таблица 8 — Прецедент U008.

6. Предложить архитектуру будущей системы. Согласовать с преподавателем технологии и фреймворки, которые будут использоваться при реализации системы.

Архитектура:

Для построения информационной системы **MyFilmList** можно использовать многоуровневую (многослойную) архитектуру, которая состоит из нескольких взаимосвязанных компонентов. Это разделение позволяет каждому слою системы выполнять свои собственные задачи и легко масштабироваться при необходимости.

Клиент-серверная архитектура

MyFilmList будет основан на клиент-серверной архитектуре, где frontend и backend отделены друг от друга и взаимодействуют через API. Это позволит легко масштабировать как клиентскую, так и серверную части.

Фронтенд (Client Side): Фронтенд будет представлять собой SSR (Server-Side Rendering). Интерфейс должен быть с поддержкой адаптивного дизайна для работы на мобильных и десктопных устройствах, поддерживать интернационализацию (русский и английский язык). Фронтенд взаимодействует с бэкеном через **RESTful API**.

Бэкенд (Server Side): Реализован как монолит с использованием MVC, разрабатывается по стратегии bottom-up.

Основные модули бэкенда:

- Модуль **авторизации**: JWT-токены для аутентификации.
- Модуль **фильмов и сериалов**: управление данными о фильмах и сериалах, предоставление API для их поиска, фильтрации и детализированной информации.
- Модуль **рекомендаций**: система персональных рекомендаций на основе истории просмотров и оценок пользователя.

- Модуль управления пользователями: регистрация, редактирование профилей, управление ролями (пользователь, модератор, администратор).
- Модуль moderation и отчетов: обработка жалоб, управление контентом.

База данных: База данных будет хранить все данные о фильмах, пользователях, рецензиях, рейтингах и подборках.

Файловое хранилище: Система будет использовать хранилище для хранения медиа файлов, таких как постеры фильмов и аватары пользователей, с целью оптимизации скорости и производительности.

Общая схема архитектуры:

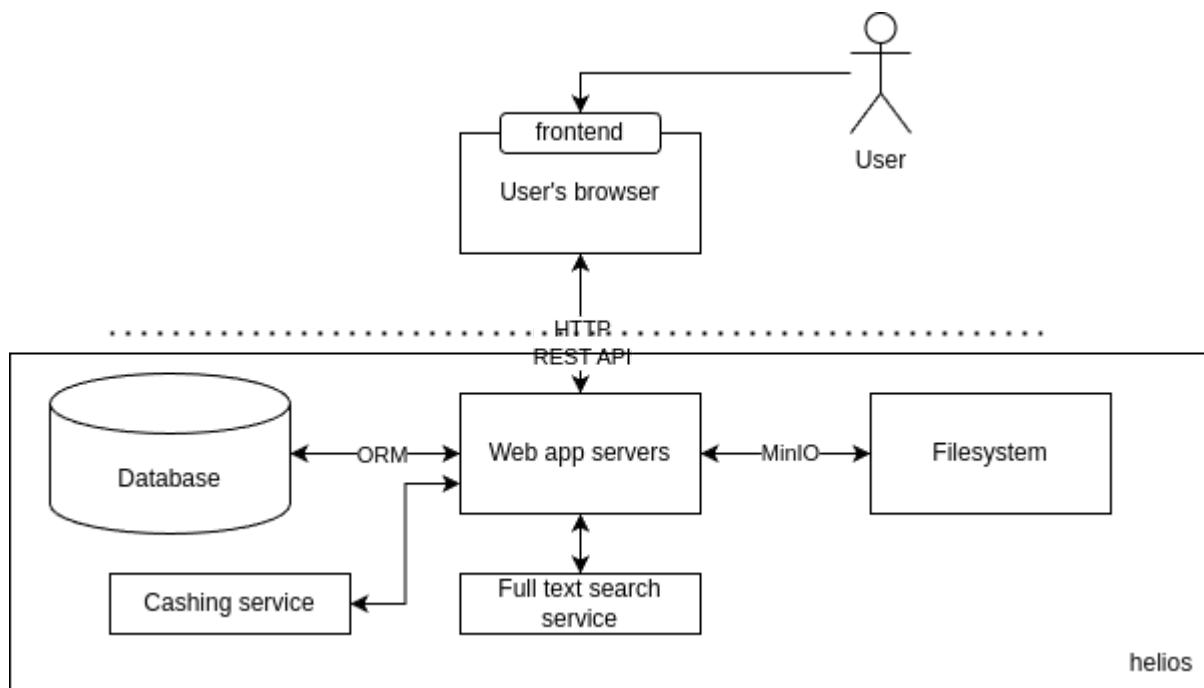


Рисунок 1 — Общая архитектура системы.

Технологии и фреймворки:

Backend	Frontend
<ul style="list-style-type: none">● Java,● Spring Boot,● Hibernate,● PostgreSQL,● MinIO	<ul style="list-style-type: none">● TypeScript,● Next.js,● Redux,● Next UI

Таблица 9 — Технологии и фреймворки.

Вывод

В результате выполнения первого этапа курсовой работы мы верно подобрали и детально изучили предметную область, сформировали подробное описание предметной области с ключевыми сущностями и составили основные требования к информационной системе, а также предложили возможную архитектуру системы и технологии. На этом этапе мы завершили базовое планирование и проектирование системы, и готовы приступить к более детальному анализу и проектированию.

Этап 2

Задание

1. Сформировать ER-модель базы данных (на основе описаний предметной области и прецедентов из предыдущего этапа).
ER-модель должна:
 - a. включать в себя не менее 10 сущностей;
 - b. содержать хотя бы одно отношение вида «многие-ко-многим».
2. Согласовать ER-модель с преподавателем. На основе ER-модели построить даталогическую модель.
3. Реализовать даталогическую модель в реляционной СУБД PostgreSQL.
4. Обеспечить целостность данных при помощи средств языка DDL и триггеров.
5. Реализовать скрипты для создания, удаления базы данных, заполнения базы тестовыми данными.
6. Предложить pl/pgsql-функции и процедуры, для выполнения критически важных запросов (которые потребуются при последующей реализации прецедентов).
7. Создать индексы на основе анализа использования базы данных в контексте описанных на первом этапе прецедентов. Обосновать полезность созданных индексов для реализации представленных на первом этапе бизнес-процессов.
8. Составить отчет.

1. ER-модель

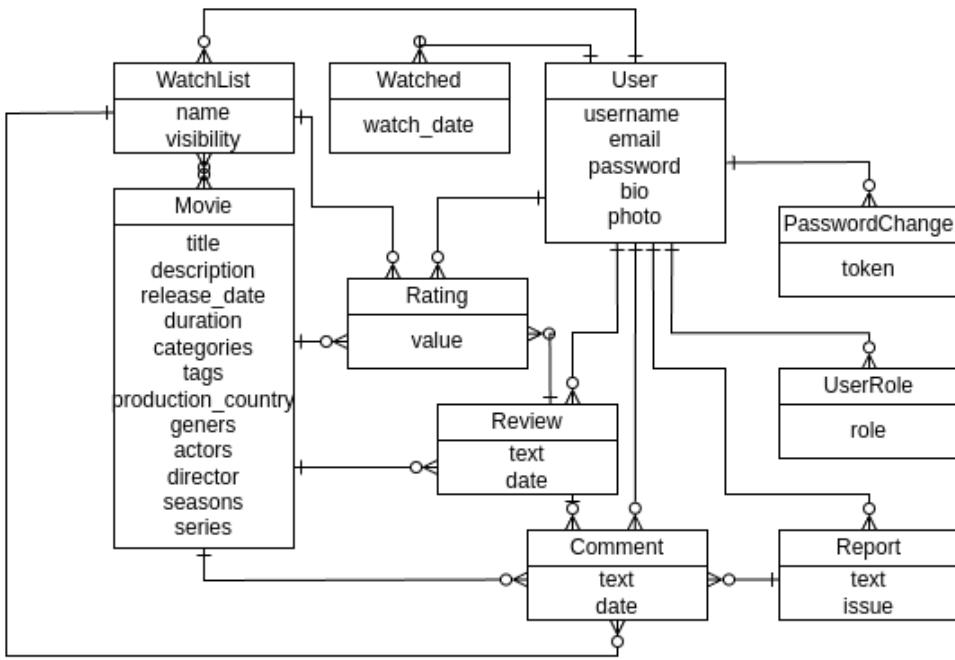


Рисунок 1 — ER-модель

2. Даталогическая модель

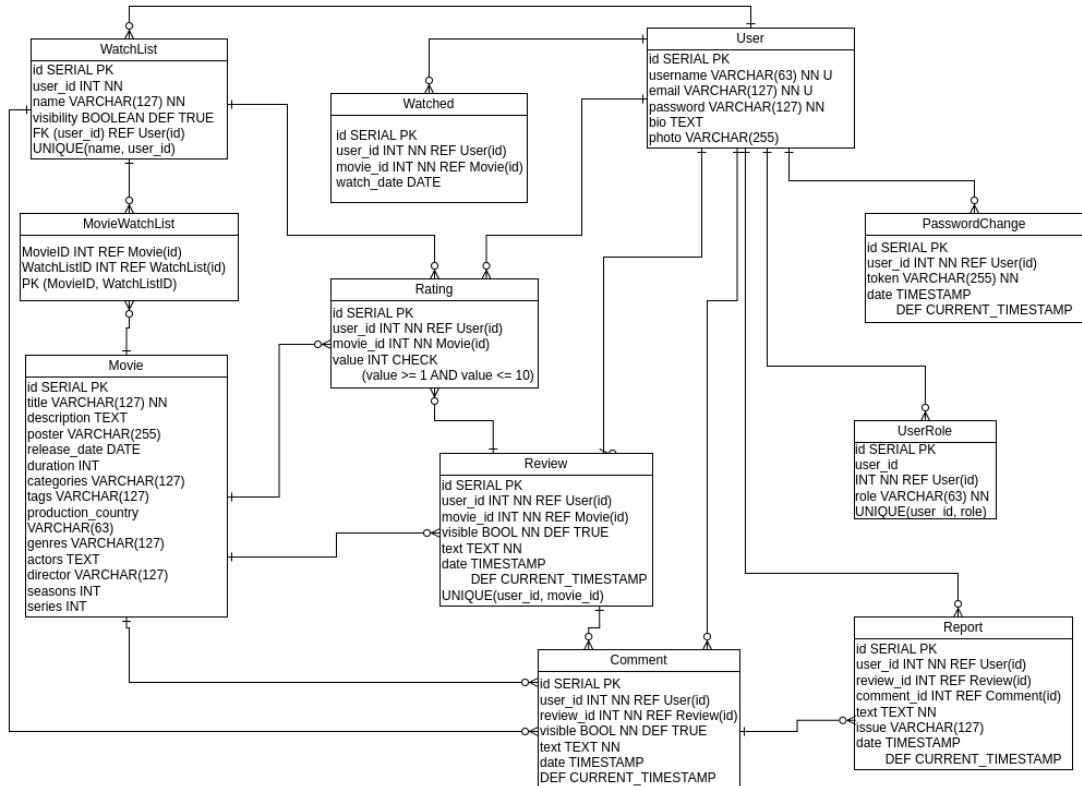


Рисунок 2 — Даталогическая модель

3. Реализация даталогической модели в реляционной СУБД PostgreSQL

```
CREATE TABLE "User" (
    id SERIAL PRIMARY KEY,
    username VARCHAR(63) NOT NULL UNIQUE,
    email VARCHAR(127) NOT NULL UNIQUE,
    password VARCHAR(127) NOT NULL,
    bio TEXT,
    photo VARCHAR(255)
);

CREATE TABLE "WatchList" (
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES "User"(id) ON DELETE SET NULL,
    name VARCHAR(127) NOT NULL,
    visibility BOOLEAN NOT NULL DEFAULT TRUE,
    UNIQUE(name, user_id)
);

CREATE TABLE "Movie" (
    id SERIAL PRIMARY KEY,
    title VARCHAR(127) NOT NULL,
    description TEXT,
    poster VARCHAR(255),
    release_date DATE,
    duration INT,
    rating FLOAT,
    categories VARCHAR(127),
    tags VARCHAR(127),
    production_country VARCHAR(63),
    genres VARCHAR(127),
    actors TEXT,
    director VARCHAR(127),
    seasons INT CHECK (value IS NULL OR value >= 0),
    series INT CHECK (value IS NULL OR value >= 0)
);

CREATE TABLE "MovieWatchList" (
    MovieID INT NOT NULL REFERENCES "Movie"(id) ON DELETE CASCADE,
    WatchListID INT NOT NULL REFERENCES "WatchList"(id) ON DELETE CASCADE,
    PRIMARY KEY (MovieID, WatchListID)
);

CREATE TABLE "Watched" (
    id SERIAL PRIMARY KEY,
    user_id INT NOT NULL REFERENCES "User"(id) ON DELETE CASCADE,
    movie_id INT NOT NULL REFERENCES "Movie"(id) ON DELETE CASCADE,
    watch_date DATE CHECK (value IS NULL OR value > CURRENT_DATE)
);
```

```

CREATE TABLE "Rating" (
    id SERIAL PRIMARY KEY,
    user_id INT NOT NULL REFERENCES "User"(id) ON DELETE SET NULL,
    movie_id INT NOT NULL REFERENCES "Movie"(id) ON DELETE CASCADE,
    value INT CHECK (value >= 1 AND value <= 10)
);

CREATE TABLE "Review" (
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES "User"(id) ON DELETE SET NULL,
    movie_id INT NOT NULL REFERENCES "Movie"(id) ON DELETE CASCADE,
    visible BOOLEAN NOT NULL DEFAULT TRUE,
    text TEXT NOT NULL,
    date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE(user_id, movie_id)
);

CREATE TABLE "Comment" (
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES "User"(id) ON DELETE SET NULL,
    review_id INT NOT NULL REFERENCES "Review"(id) ON DELETE CASCADE,
    visible BOOLEAN NOT NULL DEFAULT TRUE,
    text TEXT NOT NULL,
    date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE "Report" (
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES "User"(id) ON DELETE SET NULL,
    review_id INT REFERENCES "Review"(id) ON DELETE RESTRICT,
    comment_id INT REFERENCES "Comment"(id) ON DELETE RESTRICT,
    text TEXT NOT NULL,
    issue VARCHAR(127),
    date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE "PasswordChange" (
    id SERIAL PRIMARY KEY,
    user_id INT NOT NULL REFERENCES "User"(id) ON DELETE CASCADE,
    token VARCHAR(255) NOT NULL,
    date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE "UserRole" (
    id SERIAL PRIMARY KEY,
    user_id INT NOT NULL REFERENCES "User"(id) ON DELETE CASCADE,
    role VARCHAR(63) NOT NULL,
    UNIQUE(user_id, role)
);

```

4. Триггеры

```
-- delete password's token after 7 days
CREATE OR REPLACE FUNCTION cleanup_password_tokens()
RETURNS TRIGGER AS $$ 
BEGIN
    DELETE FROM "PasswordChange" WHERE date < CURRENT_TIMESTAMP - INTERVAL '7
days';
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER cleanup_tokens
AFTER INSERT OR UPDATE ON "PasswordChange"
FOR EACH ROW
EXECUTE FUNCTION cleanup_password_tokens();

-- update movie rating
CREATE OR REPLACE FUNCTION update_movie_rating()
RETURNS TRIGGER AS $$ 
BEGIN
    UPDATE "Movie"
    SET rating = (
        SELECT AVG(value) ::NUMERIC(3, 2)
        FROM "Rating"
        WHERE movie_id = NEW.movie_id
    )
    WHERE id = NEW.movie_id;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER recalculate_movie_rating
AFTER INSERT OR UPDATE OR DELETE ON "Rating"
FOR EACH ROW
EXECUTE FUNCTION update_movie_rating();
```

5. Скрипты для создания, удаления БД, заполнения базы тестовыми данными

Скрипт для создания БД:

```
CREATE DATABASE movie_db;  
  
\c movie_db;  
  
-- Далее см. "Реализация даталогической модели в реляционной СУБД PostgreSQL"
```

Скрипт для удаления БД:

```
-- Завершение активных соединений с базой  
SELECT pg_terminate_backend(pg_stat_activity.pid)  
FROM pg_stat_activity  
WHERE pg_stat_activity.datname = 'movie_db'  
  AND pid <> pg_backend_pid();  
  
-- Проверка наличия активных сессий после завершения  
SELECT * FROM pg_stat_activity WHERE datname = 'movie_db';  
  
-- Удаление базы данных  
DROP DATABASE movie_db;
```

Скрипт для заполнения базы тестовыми данными:

```
INSERT INTO "User" (username, email, password, bio, photo)  
VALUES  
  ('john_doe', 'john@example.com', 'password123', 'Just a movie lover',  
NULL),  
  ('jane_smith', 'jane@example.com', 'securepass', 'Critic and reviewer',  
NULL),  
  ('admin_user', 'admin@example.com', 'adminpass', 'Admin of the  
platform',NULL);  
  
INSERT INTO "UserRole" (user_id, role)  
VALUES  
  (1, 'USER'),  
  (2, 'USER'),  
  (3, 'ADMIN');  
  
INSERT INTO "Movie" (title, description, release_date, duration, categories,  
tags, production_country, genres, actors, director, seasons, series)  
VALUES  
  ('Inception', 'A mind-bending thriller', '2010-07-16', 148, 'Sci-Fi',  
'dreams, mind', 'USA', 'Science Fiction', 'Leonardo DiCaprio, Ellen Page',  
'Christopher Nolan', NULL, NULL),
```

```

('Breaking Bad', 'A high school teacher turns to cooking meth',
'2008-01-20', 60, 'Drama', 'crime, drugs', 'USA', 'Crime, Drama', 'Bryan
Cranston, Aaron Paul', 'Vince Gilligan', 5, 62);

INSERT INTO "WatchList" (user_id, name, visibility)
VALUES
(1, 'Favorite Movies', TRUE),
(2, 'To Watch', TRUE);

INSERT INTO "MovieWatchList" (MovieID, WatchListID)
VALUES
(1, 1), -- Inception in "Favorite Movies"
(2, 2); -- Breaking Bad in "To Watch"

INSERT INTO "Watched" (user_id, movie_id, watch_date)
VALUES
(1, 1, '2023-11-01'), -- John watched Inception
(2, 2, '2023-10-15'); -- Jane watched Breaking Bad

INSERT INTO "Rating" (user_id, movie_id, value)
VALUES
(1, 1, 9), -- John rate 9/10 Inception
(2, 1, 9), -- Jane rate 7/10 Inception
(2, 2, 10); -- Jane rate 10/10 Breaking Bad

INSERT INTO "Review" (user_id, movie_id, text)
VALUES
(1, 1, 'Amazing movie, highly recommend!'), -- John reviewed on Inception
(2, 2, 'Best TV show ever!'); -- Jane reviewed on Breaking
Bad

INSERT INTO "Comment" (user_id, review_id, text)
VALUES
(2, 1, 'I agree with you!'), -- Jane comment John's review of Inception
(1, 2, 'Thanks!'); -- John comment Jane's review of Breaking
Bad

INSERT INTO "Report" (user_id, review_id, text, issue)
VALUES
(3, 1, 'Spam content', 'Spam');

```

6. PL/pgSQL-функции и процедуры для выполнения критически важных запросов

```
CREATE OR REPLACE FUNCTION is_best_seller(arg_movie_id INT)
RETURNS BOOLEAN AS $$

DECLARE
    avg_rating NUMERIC;          -- Средний рейтинг фильма
    watched_count INT;           -- Количество активных пользователей,
которые посмотрели фильм
    watched_threshold NUMERIC;   -- Пороговое значение (10% от активных
пользователей)

BEGIN
    -- 1. Рассчитываем средний рейтинг фильма
    SELECT AVG(value) INTO avg_rating
    FROM "Rating"
    WHERE movie_id = arg_movie_id;
    -- Если нет рейтингов или средний рейтинг ниже 8, возвращаем FALSE
    IF avg_rating IS NULL OR avg_rating < 8 THEN
        RETURN FALSE;
    END IF;

    -- 2. Определяем количество активных пользователей за последний год
    CREATE TEMPORARY VIEW active_users AS
        SELECT user_id AS id
        FROM "Watched"
        WHERE watch_date >= CURRENT_DATE - INTERVAL '1 year'
        GROUP BY user_id
        HAVING COUNT(movie_id) > 10;

    -- Если активных пользователей нет, возвращаем FALSE
    IF ((SELECT COUNT(DISTINCT id) FROM active_users) = 0) THEN
        RETURN FALSE;
    END IF;

    -- 3. Считаем, сколько активных пользователей посмотрело данный фильм
    SELECT COUNT(DISTINCT user_id) INTO watched_count
    FROM "Watched"
    WHERE movie_id = arg_movie_id AND user_id IN (SELECT id FROM
"active_users");

    -- 4. Определяем, соответствует ли количество просмотров порогу (10% от
активных пользователей)
    watched_threshold := active_users * 0.1;

    IF watched_count >= watched_threshold THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END;
$$ LANGUAGE plpgsql;
```

7. Индексы

```
-- Изначально создаем приватные списки к просмотру
CREATE INDEX idx_watchlist_visibility ON "WatchList"(visibility) WHERE
visibility IS FALSE;

-- Очень часто читаем, редко пишем
CREATE INDEX idx_movie_title ON "Movie" USING GIN(title);
CREATE INDEX idx_movie_production_country ON "Movie"(production_country);
CREATE INDEX idx_movie_genres ON "Movie" USING GIN(genres);
CREATE INDEX idx_movie_tags ON "Movie" USING GIN(tags);
CREATE INDEX idx_movie_director ON "Movie"(director);
CREATE INDEX idx_movie_release_date ON "Movie"(release_date);

-- Показываем отзыв как на странице пользователя, так и на странице фильма
CREATE INDEX idx_review_visible ON "Review"(visible) WHERE visibility IS
FALSE;

-- Показываем отзыв на странице фильма
CREATE INDEX idx_comment_visible ON "Comment"(visible) WHERE visibility IS
FALSE;

-- При смене пароля ищем "PasswordChange" по токену из ссылки
CREATE INDEX idx_passwordchange_token ON "PasswordChange"(token);
CREATE INDEX idx_passwordchange_date ON "PasswordChange"(date);
```

8. Вывод

В результате выполнения второго этапа курсовой работы была разработана полноценная реляционная база данных на основе предварительно сформулированной ER- и даталогической модели базы данных для будущей информационной системы. При реализации функции и создания индексов в реляционной БД PostgreSQL, мы несколько раз пересмотрели даталогическую модель, добавив недостающие поля и ограничения. На основе проделанной работы была создана структура базы данных, которая отвечает поставленным требованиям, гарантирует необходимую целостность хранения данных. Вероятно, реализованная система может быть переработана в дальнейшем с учетом новых требований.

Этап 3

Задание

1. Изобразить диаграмму классов, представляющую общую архитектуру системы.
2. Реализовать уровень хранения информационной системы на основе разработанной на предыдущем этапе базы данных.
3. При реализации уровня хранения должны использоваться функции/процедуры, созданные на втором этапе с помощью pl/pgsql. Нельзя замещать их использование альтернативной реализацией аналогичных запросов на уровне хранения информационной системы.
4. На основе описания бизнес-процессов из первого этапа и построенного уровня хранения реализовать уровень бизнес-логики информационной системы.
5. Составить отчет.

1. Диаграммы классов

Все диаграммы в высоком разрешении могут быть найдены по адресу

<https://github.com/maxbarsukov/MyFilmList/tree/master/docs/part3/uml>.

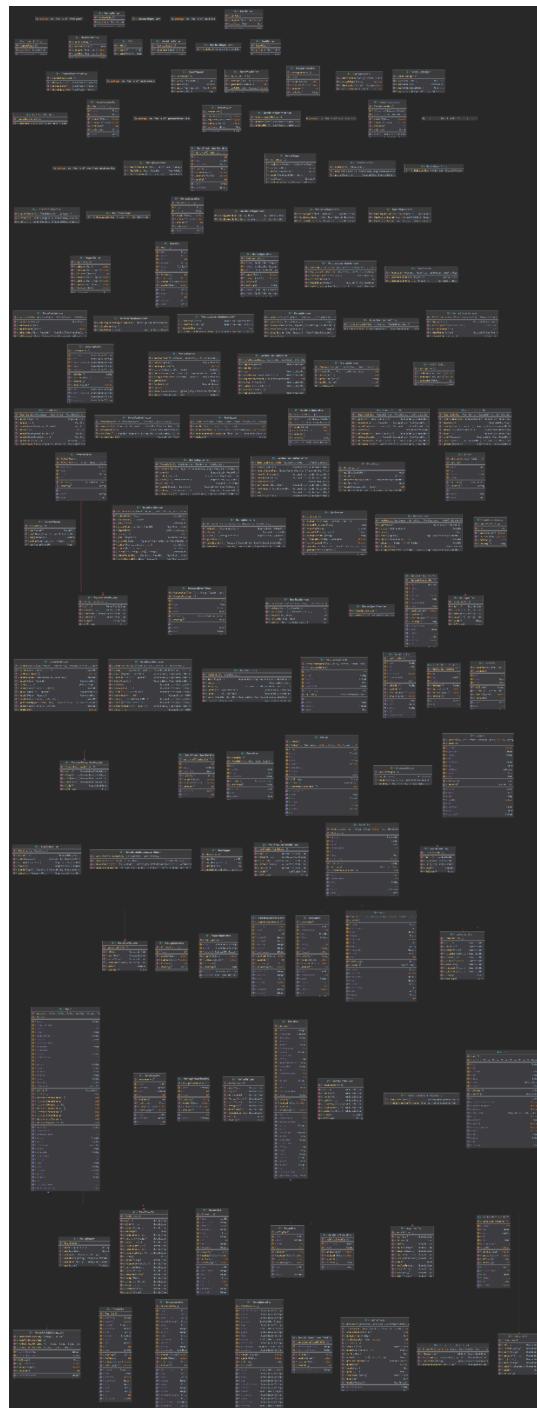


Рисунок 1 — UML-диаграмма классов, представляющая общую архитектуру системы.

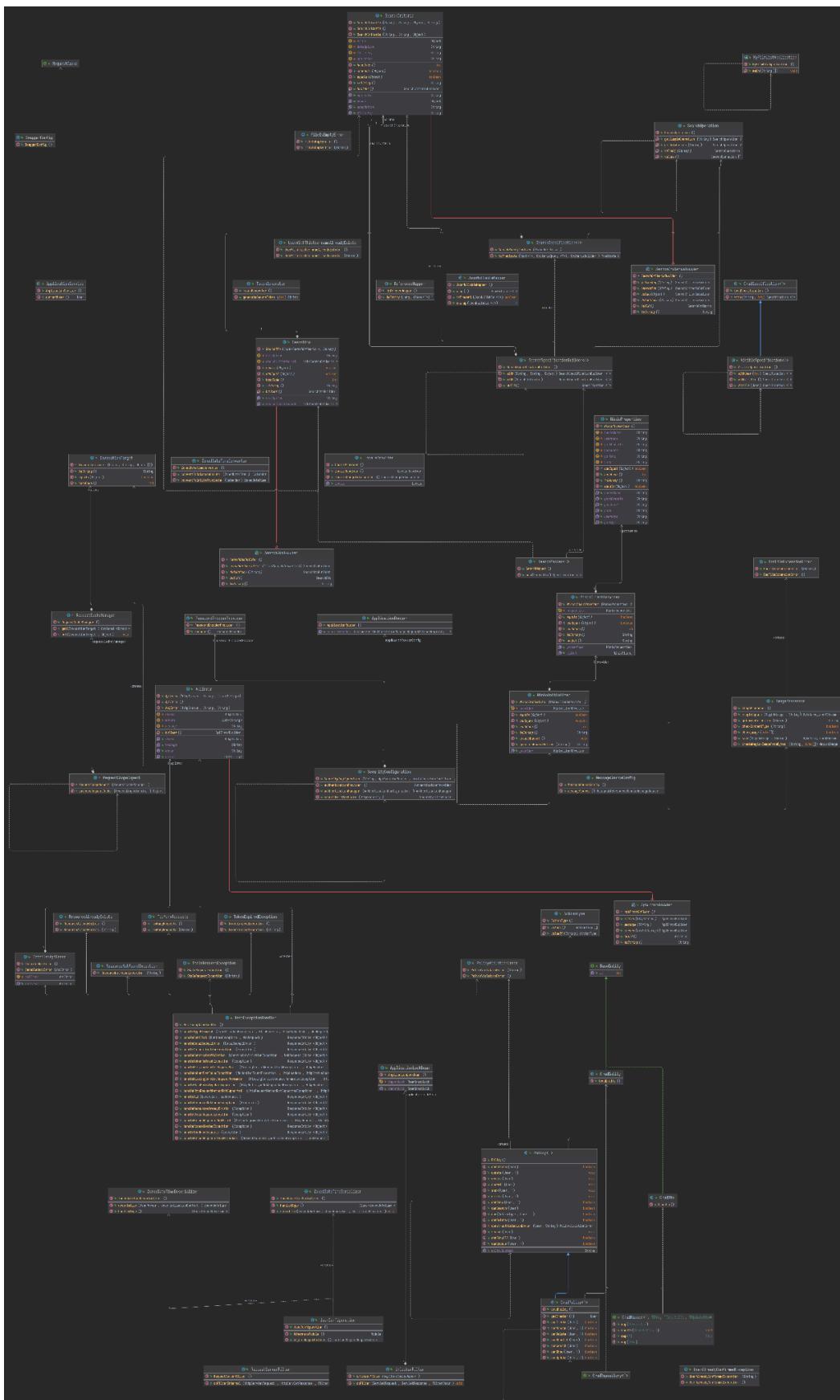


Рисунок 2 — UML-диаграмма классов, представляющая архитектуру общего модуля системы.

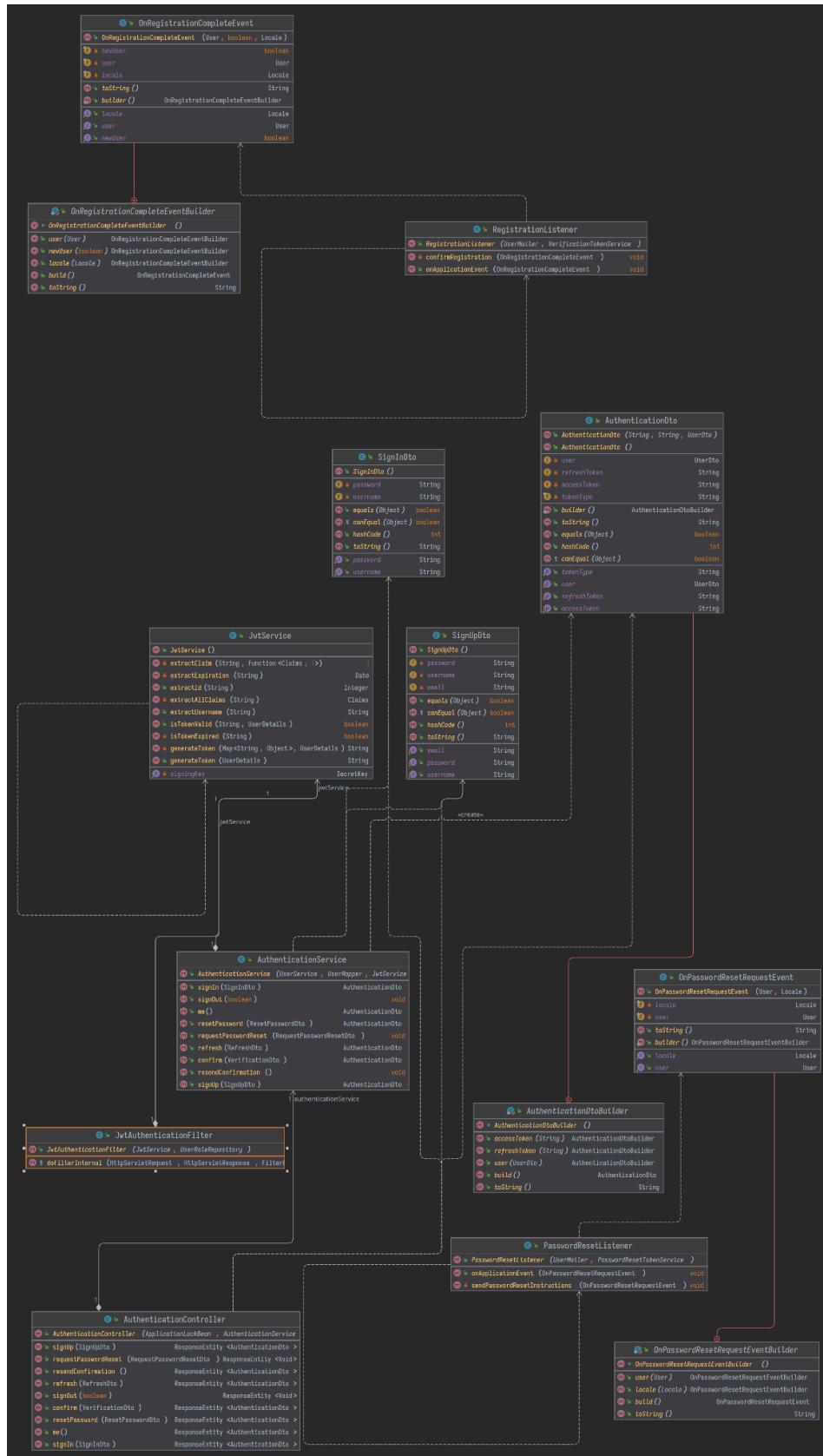


Рисунок 3 — UML-диаграмма классов, представляющая архитектуру модуля аутентификации.

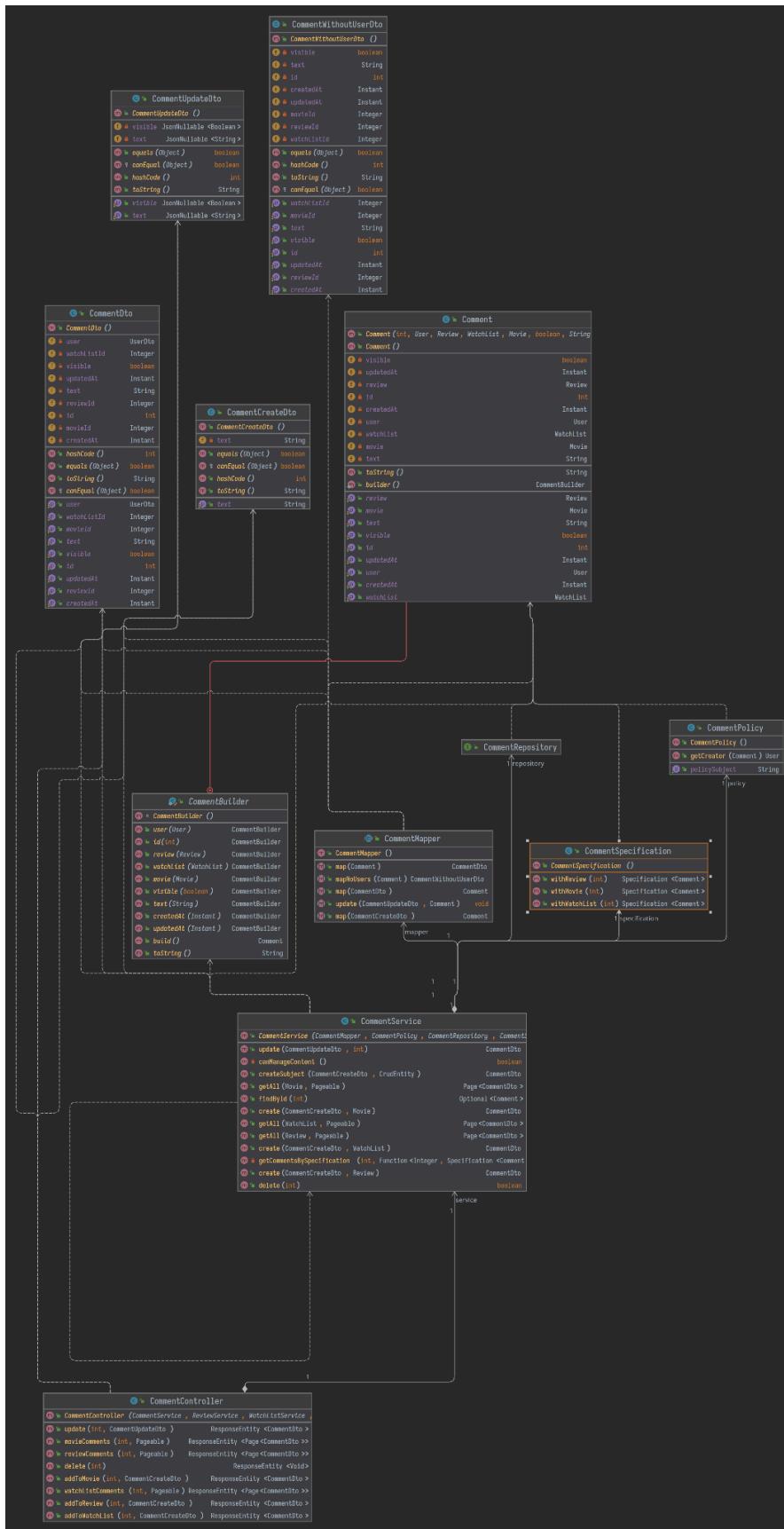


Рисунок 4 — UML-диаграмма классов, представляющая архитектуру модуля комментариев.

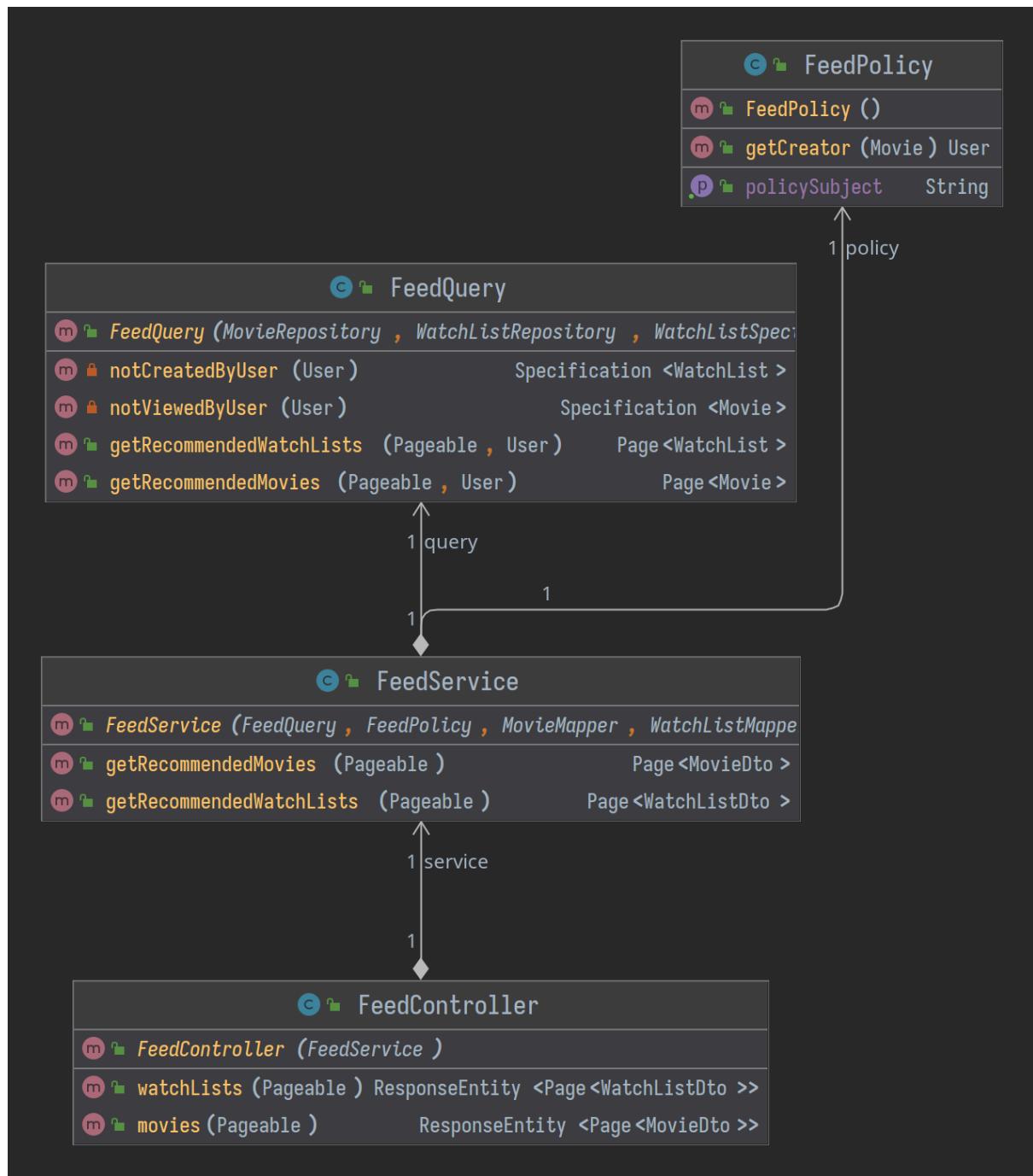


Рисунок 5 — UML-диаграмма классов, представляющая архитектуру модуля рекомендаций.

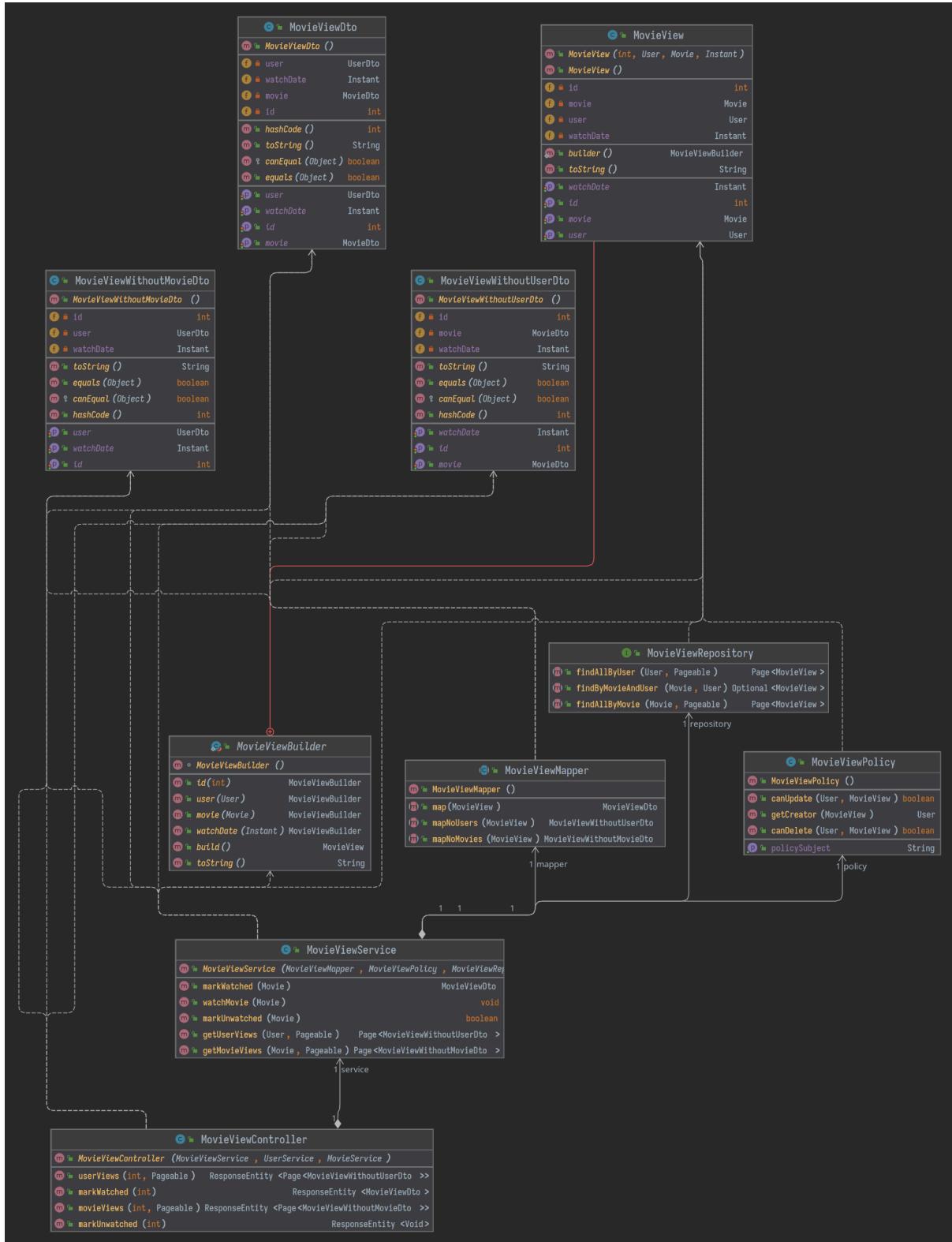


Рисунок 6 — UML-диаграмма классов, представляющая архитектуру модуля просмотров.

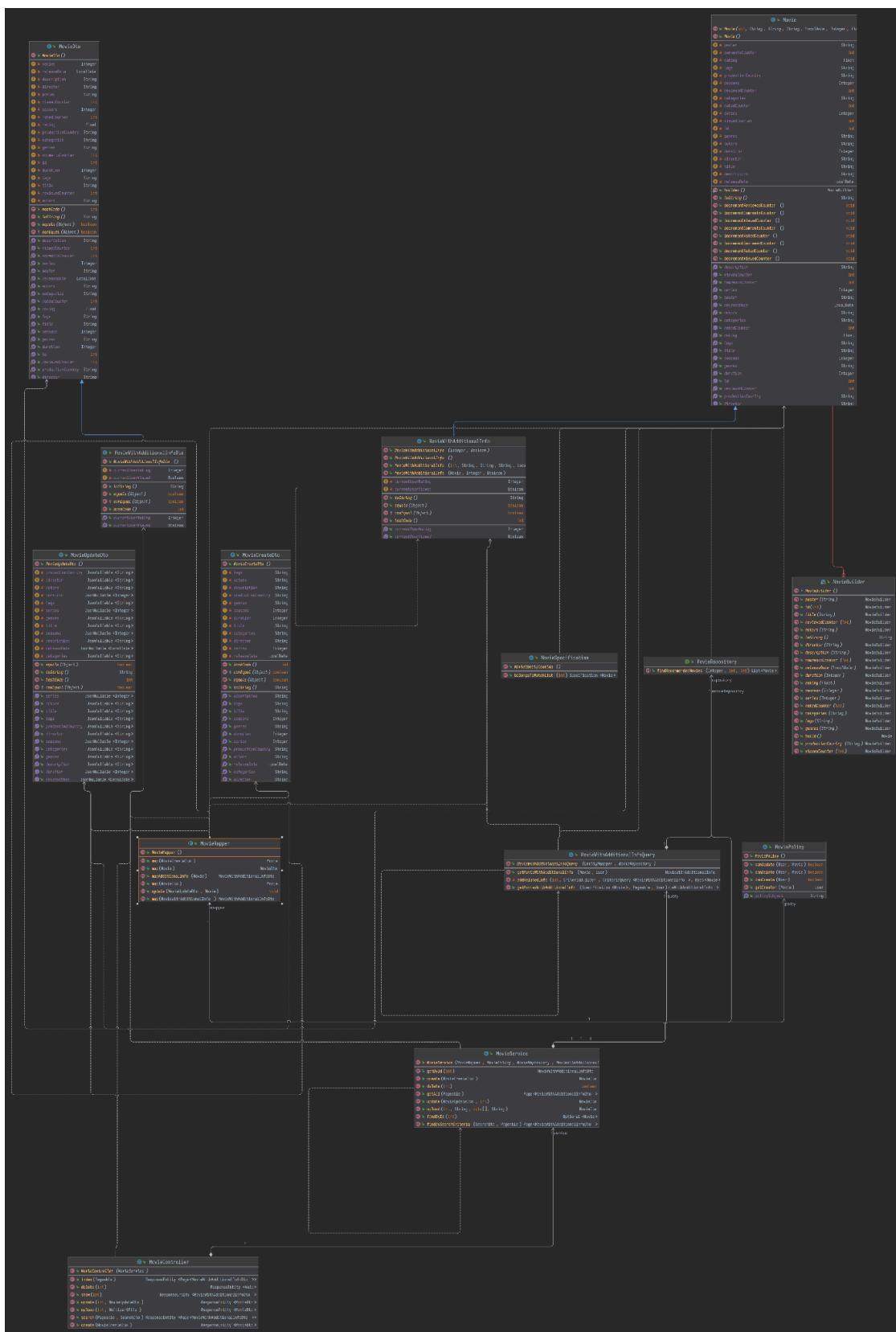


Рисунок 7 — UML-диаграмма классов, представляющая архитектуру модуля фильмов.

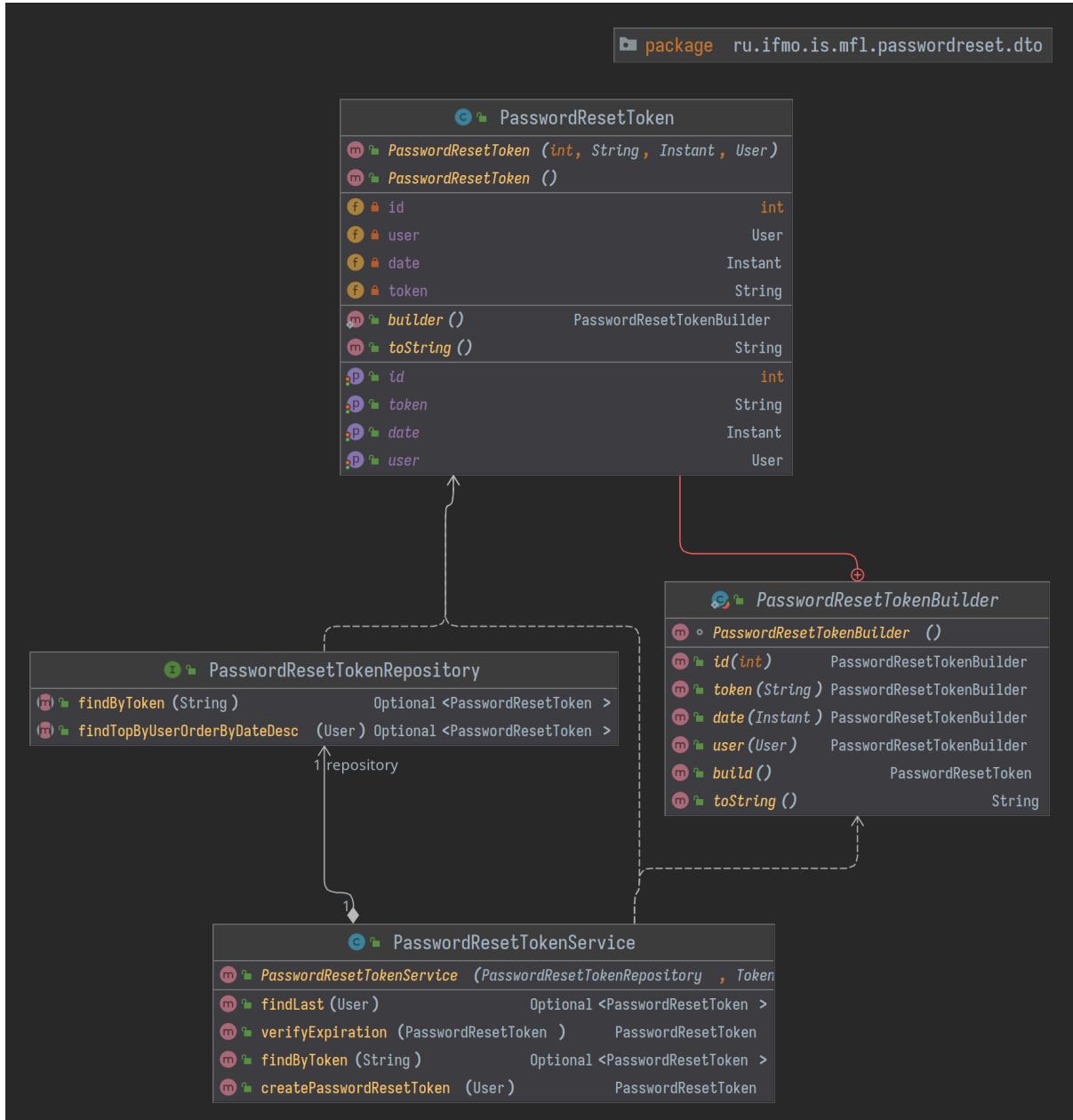


Рисунок 8 — UML-диаграмма классов, представляющая архитектуру модуля сбросов паролей.

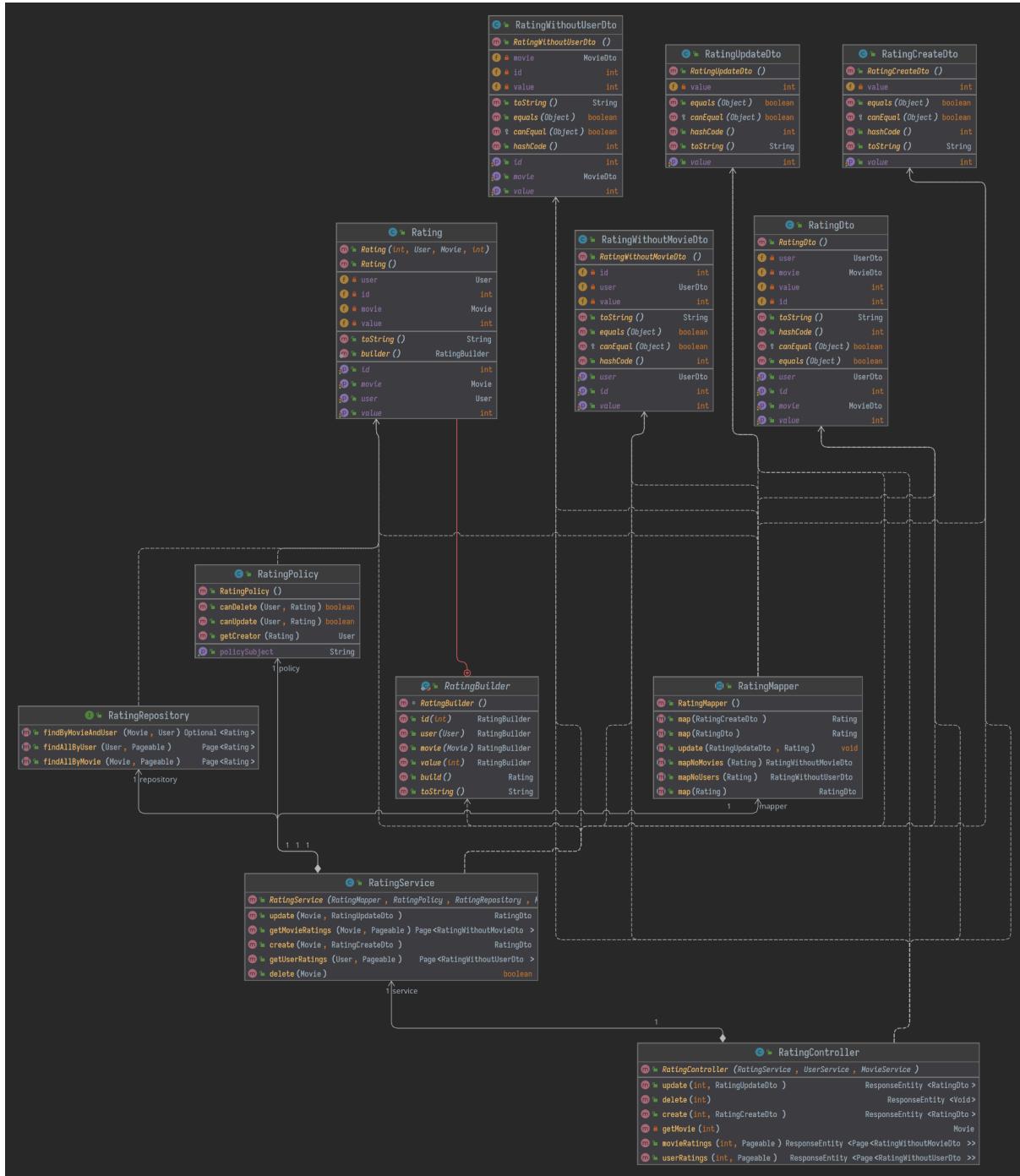


Рисунок 9 — UML-диаграмма классов, представляющая архитектуру модуля рейтингов.

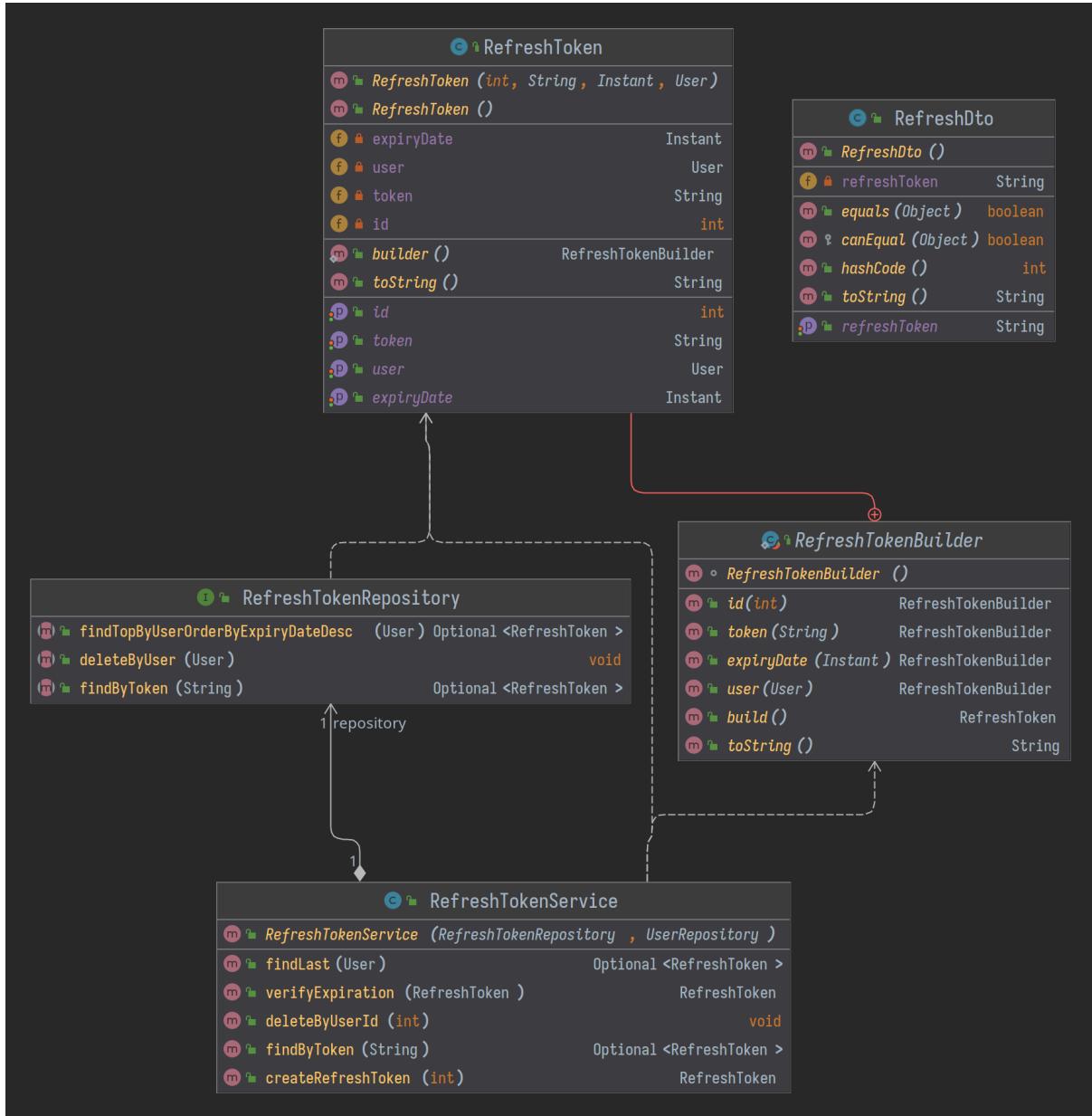


Рисунок 10 — UML-диаграмма классов, представляющая архитектуру модуля refresh-токенов.

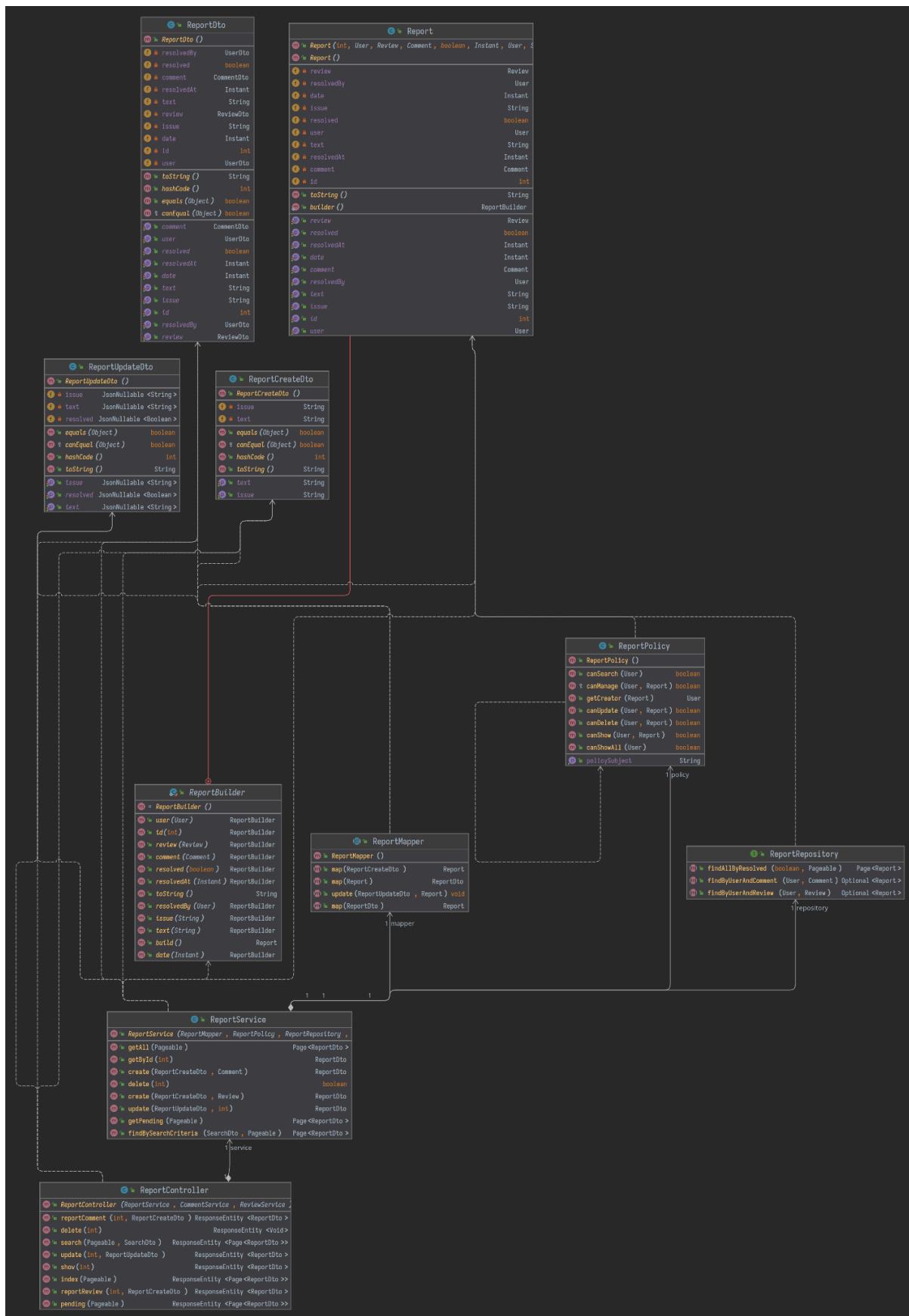


Рисунок 11 — UML-диаграмма классов, представляющая архитектуру модуля жалоб.

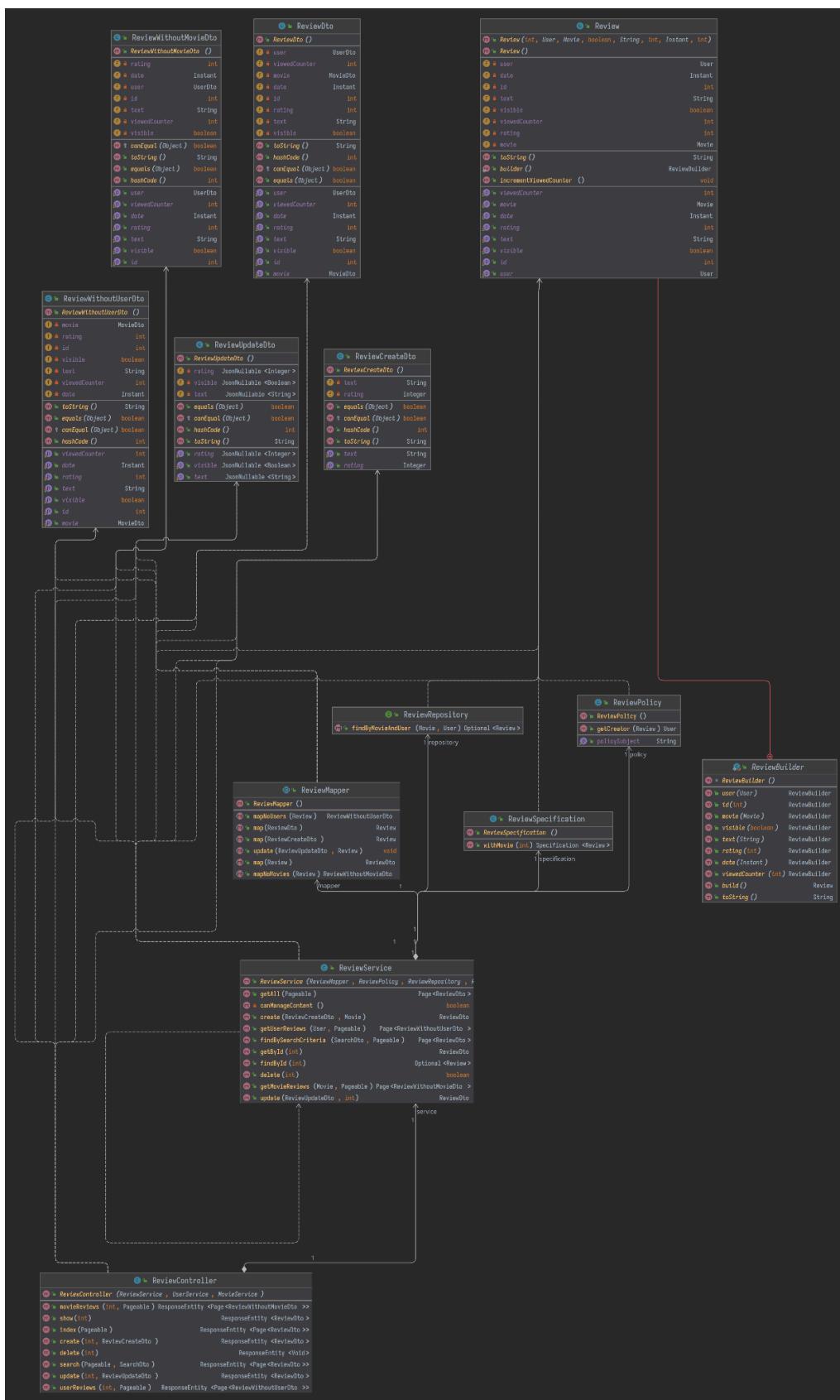


Рисунок 12 — UML-диаграмма классов, представляющая архитектуру модуля рецензий.

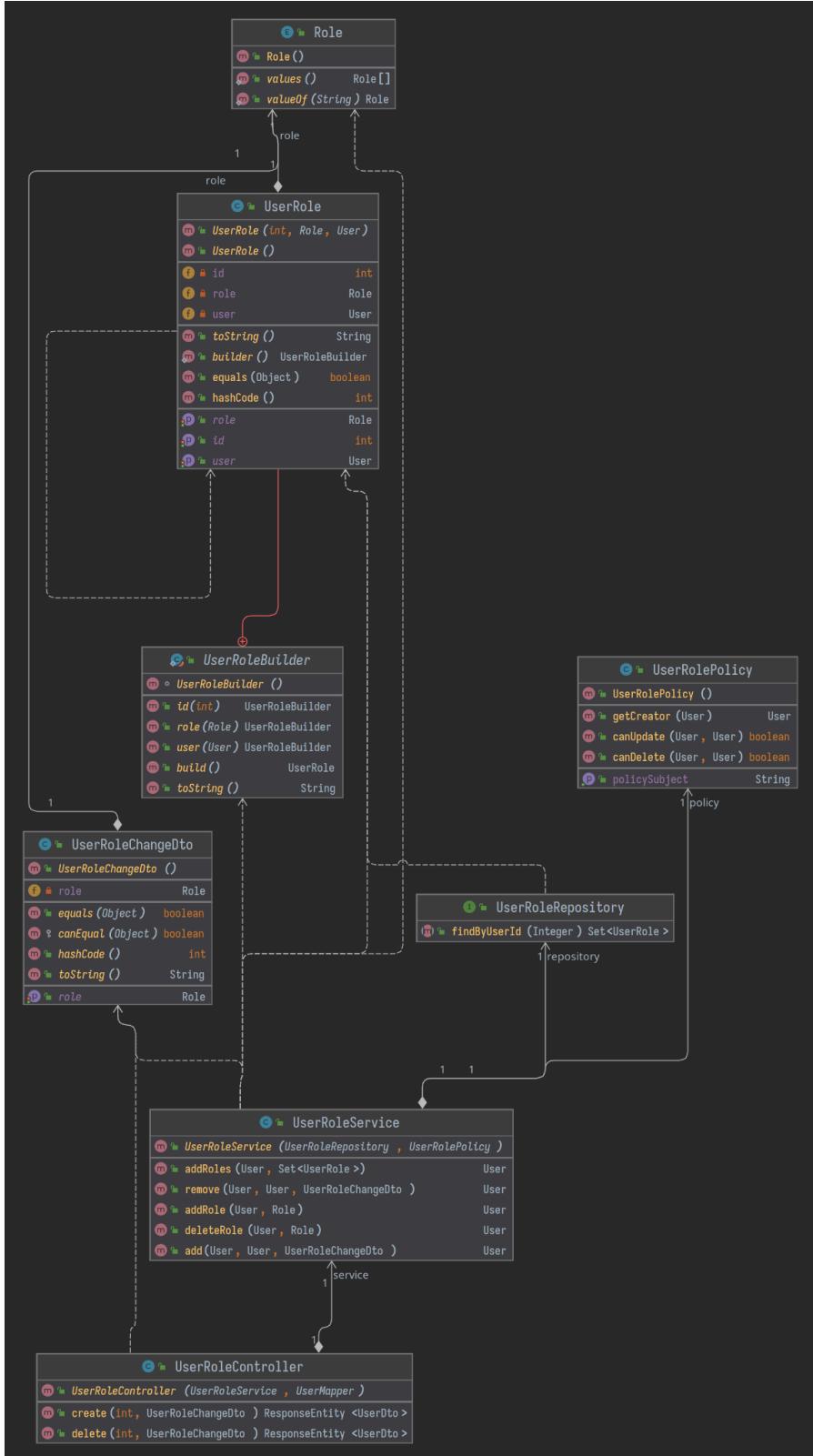


Рисунок 13 — UML-диаграмма классов, представляющую архитектуру модуля пользовательских ролей.

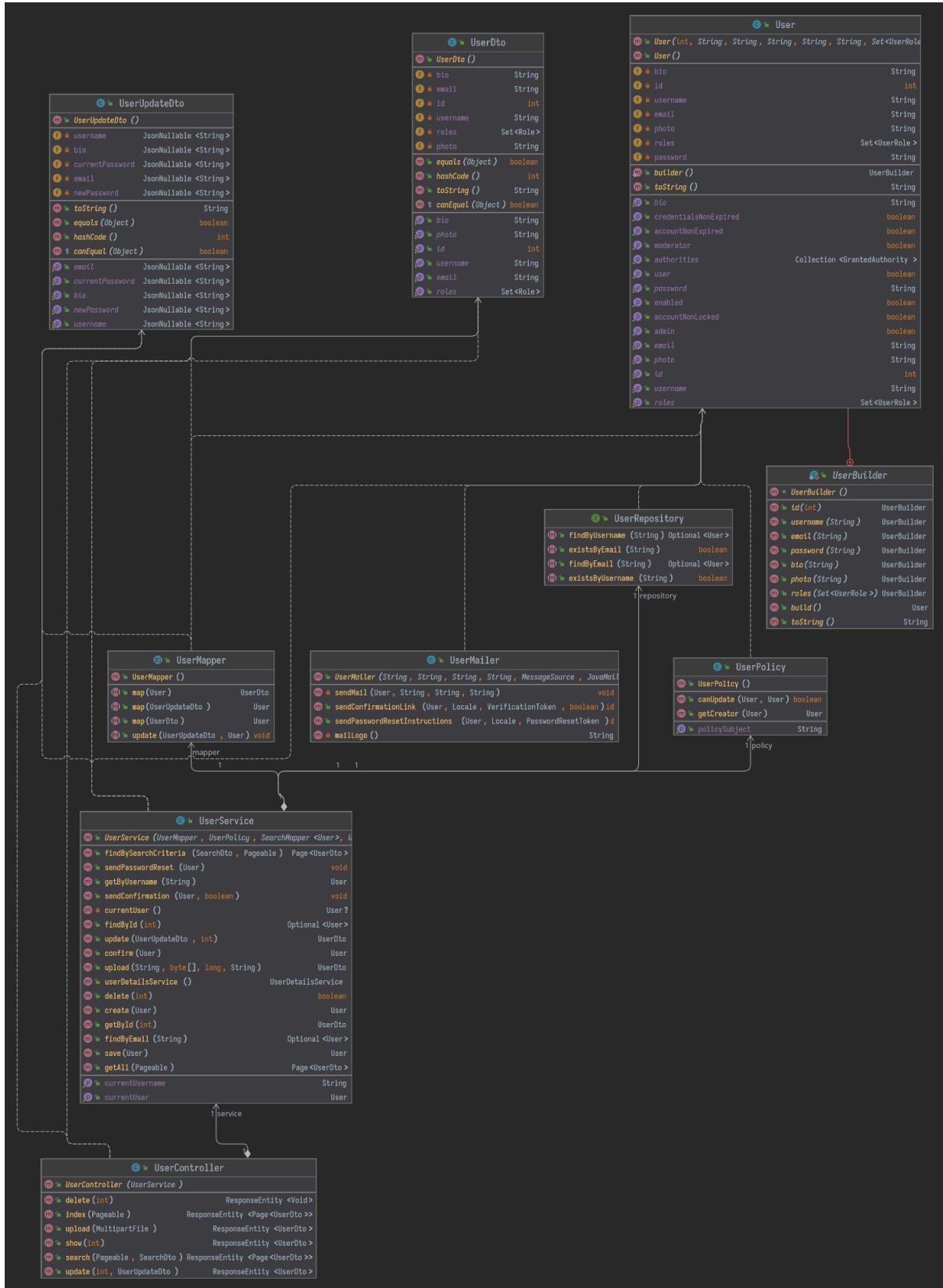


Рисунок 13 — UML-диаграмма классов, представляющую архитектуру модуля пользователей.

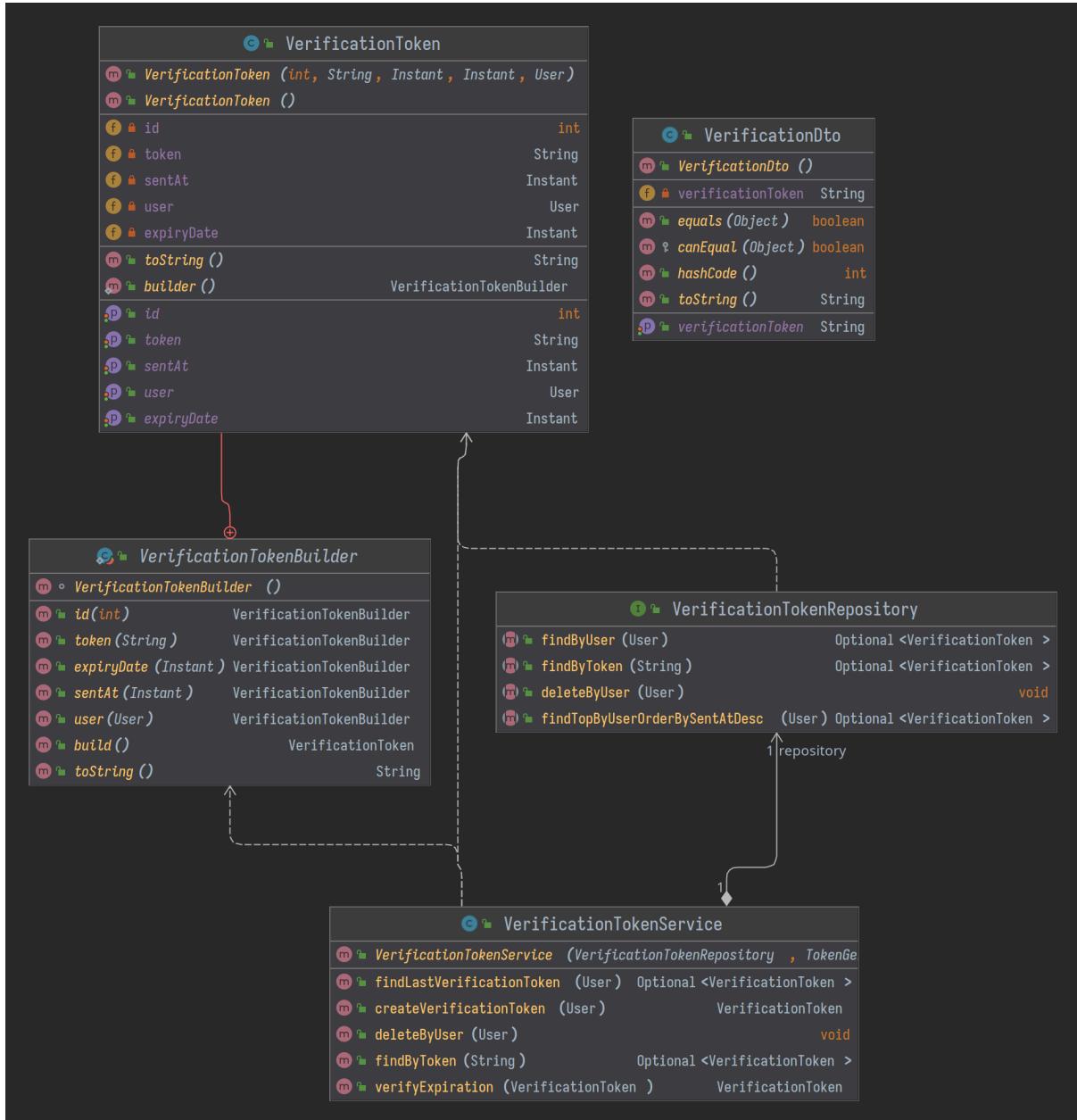


Рисунок 13 — UML-диаграмма классов, представляющую архитектуру модуля подтверждения пользователей.

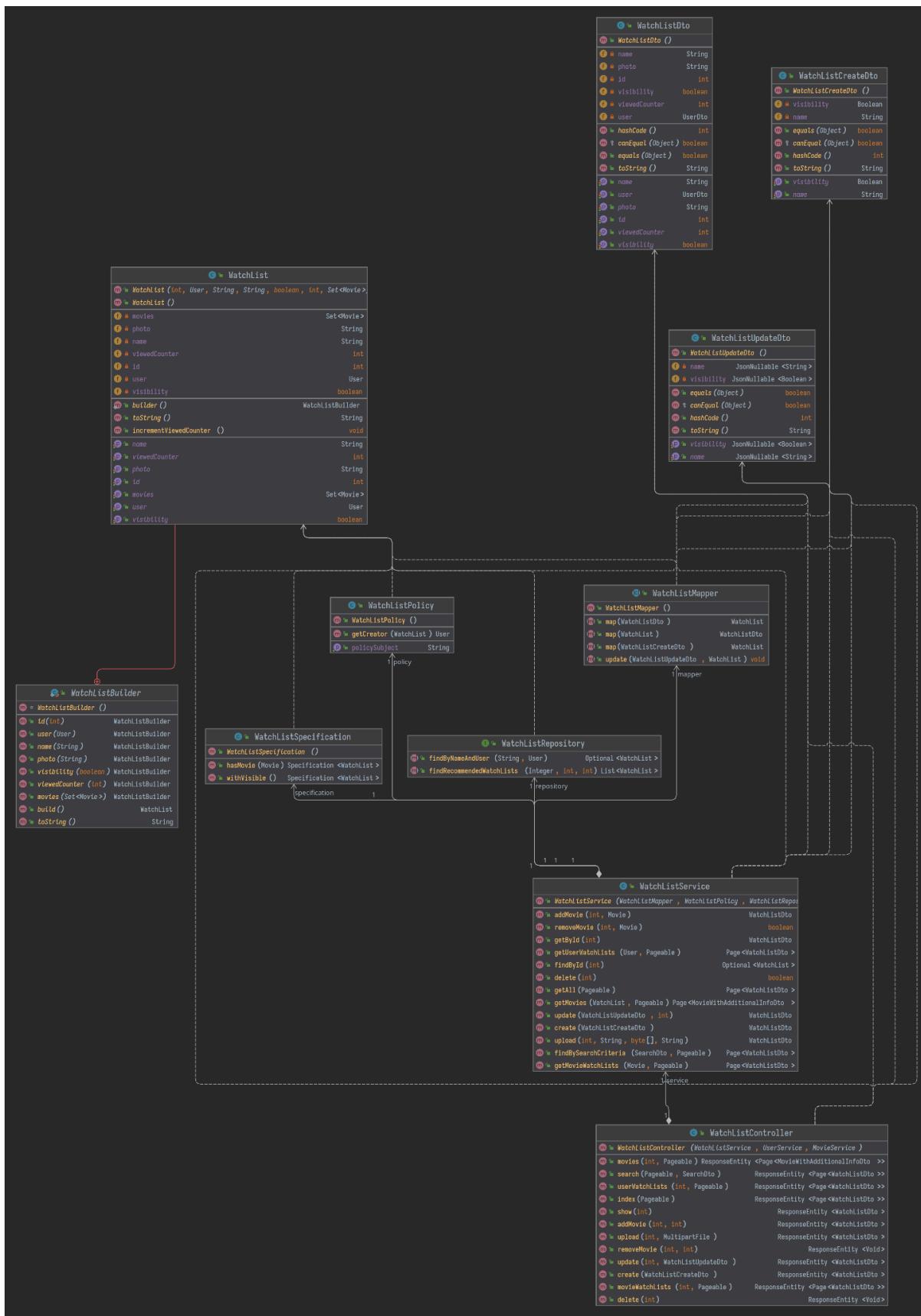


Рисунок 13 — UML-диаграмма классов, представляющую архитектуру модуля коллекций фильмов.

2. Реализация

Уровни хранения и бизнес-логики информационной системы реализованы и опубликованы на веб-сервис для хостинга ИТ-проектов и их совместной разработки «GitHub» и размещены по ссылке <https://github.com/maxbarsukov/MyFilmList>.

3. Вывод

В результате выполнения третьего этапа курсовой работы была успешно разработана и реализована архитектура информационной системы, основанная на ранее созданной диаграмме классов. Мы создали уровень хранения, который использует функции и процедуры, разработанные на втором этапе с помощью PL/pgSQL, что обеспечило соответствие требованиям к использованию единого подхода к взаимодействию с базой данных. На основе описания бизнес-процессов, полученных на первом этапе, был реализован уровень бизнес-логики, который обеспечивает обработку данных в системе в соответствии с моделью. Все компоненты системы были интегрированы, что позволило достичь необходимой функциональности. Таким образом, выполненные задачи обеспечили создание целостной информационной системы, готовой к дальнейшему тестированию и внедрению. Результаты данного этапа послужили основой для последующих улучшений и оптимизаций системы, а также для разработки пользовательского интерфейса и интеграции с внешними сервисами.

Этап 4

Вывод

В ходе выполнения курсовой работы была успешно разработана информационная система, которая включает в себя все ключевые компоненты, необходимые для решения поставленных задач. Мы разработали подробное текстовое описание предметной области, сформулировали функциональные и нефункциональные требования, а также построили модели основных прецедентов, что позволило создать четкую структуру для дальнейшей работы.

На этапе создания ER-модели базы данных мы разработали модель, состоящую из более чем 10 сущностей и реализовали отношения «многие-ко-многим». Это позволило обеспечить целостность данных и создать эффективную архитектуру базы данных в PostgreSQL. Мы реализовали необходимые триггеры и PL/pgSQL-функции, что дало возможность выполнять критически важные запросы с высокой производительностью.

В процессе реализации уровня хранения системы мы использовали разработанные функции и процедуры, что обеспечило согласованность и надежность работы с базой данных. Уровень бизнес-логики был интегрирован на основе описанных бизнес-процессов, что позволило создать эффективное взаимодействие между компонентами системы.

На этапе реализации уровня представления приложения мы создали интуитивно понятный интерфейс, который обеспечивает выполнение всех бизнес-процессов, описанных на первом этапе. Итоговый отчет включает в себя все результаты работы, а презентация проекта продемонстрировала функциональные возможности созданной системы.

В результате работы мы не только углубили свои теоретические знания, но и приобрели практические навыки в области проектирования и разработки информационных систем. Мы научились работать с

различными технологиями, такими как Next.js, Spring MVC и PostgreSQL, а также освоили методы проектирования баз данных и разработки бизнес-логики.

Для более подробного изучения проекта, вы можете ознакомиться с репозиторием по следующей ссылке:
<https://github.com/maxbarsukov/MyFilmList>.

Также доступ к самой системе можно получить по следующему адресу: <https://mfl.maxbarsukov.ru/> .

Приложение А

Глоссарий

1. **Учетная запись** (Account) – совокупность уникальных данных, позволяющих пользователю аутентифицироваться на платформе. Включает уникальную электронную почту и пароль. Учетная запись должна быть дополнительно связана с личной информацией (фото, имя, фамилия, описание (bio), история активности).
2. **Пользователь** (User) – лицо, использующее ИС. Пользователь может быть зарегистрированным или незарегистрированным: Незарегистрированный пользователь имеет доступ к ограниченному функционалу платформы: может просматривать информацию о фильмах (название, описание, рейтинг) без возможности оставлять отзывы, оценивать фильмы или создавать списки, а также имеет возможность зарегистрироваться. Зарегистрированный пользователь имеет учетную запись на платформе и может использовать расширенные функции: оценка фильмов, оставление отзывов, добавление фильмов в избранное, создание собственных списков, редактирование профиля и управление настройками конфиденциальности. Зарегистрированный пользователь также имеет доступ к персонализированной ленте рекомендаций и истории активности. Права и доступ пользователя могут расширяться в зависимости от роли, которую он занимает на платформе (модератор или администратор).
3. **Электронная почта** (Email) – уникальный идентификатор пользователя, используемый для создания учетной записи и восстановления доступа. Это запись, установленная по RFC 5322, однозначно идентифицирующая почтовый ящик, в который следует доставить сообщение электронной почты. Это также средство коммуникации с платформой для получения уведомлений и восстановления пароля.
4. **Пароль** (Password) – набор символов, используемый для аутентификации пользователя. Он должен соответствовать установленным платформой требованиям безопасности (минимальная длина 6 символов, максимальная длина 128 символов, наличие цифр (1234567890) и специальных символов(!@#\$%^&*()_+=)).

5. **Модератор** (Moderator) – пользователь с расширенными правами, ответственными за контроль над контентом и учетными записями пользователей. Модератор занимается проверкой, редактированием и удалением контента, а также обработкой жалоб пользователей.
6. **Администратор** (Administrator) – пользователь с максимальными правами управления на платформе. Администратор может управлять модераторами, пользователями, контентом, системными настройками.
7. **Фильм** (Film) (от англ. film «плёнка»), также [кинó, кинофíльм, телефíльм, кинокартíна] — отдельное произведение киноискусства. В технологическом плане фильм представляет собой совокупность движущихся изображений (монтажных кадров), связанных единым сюжетом. Каждый монтажный кадр состоит из последовательности неподвижных фотографий (кадриков), на которых зафиксированы отдельные фазы движения. Фильм, как правило, имеет звуковое сопровождение. В рамках ИС MyFilmList **фильм** – совокупность метаинформации о фильме, представленной: названием, обложкой (изображение), списком актеров, страной производства, годом выхода, жанром, слоганом, режиссером, сценаристом, продюсером, оператором, композитором, художником, монтажером, бюджетом, сборами в стране производства, сборами в мире, рейтингом МРАА, длительностью в минутах (натуральное число). Фильм в ИС также включает понятия художественного и документального фильма, сериала, TV-шоу, анимацию и другие аудиовизуальные произведения.
8. **Оценка** (Rating) – числовое значение, которое зарегистрированный пользователь может присвоить фильму для выражения своего мнения о его качестве. Шкала оценки варьируется на натуральных числах от 1 до 10, где 1 означает "очень плохо", а 10 – "отлично". Оценки пользователей влияют на средний рейтинг фильма на платформе.
9. **Оценивать** (To Rate) – процесс присвоения фильму числовой оценки зарегистрированным пользователем на основе личного восприятия. Оценка влияет на общий рейтинг фильма, видимый для других пользователей.

- 10. Отзыв** (Comment) – краткий текст, оставленный зарегистрированным пользователем на странице фильма с целью поделиться своими впечатлениями или мнением.
- 11. Рецензия** (Review) – детализированный и аналитический текст (минимум 1000 символов), в котором зарегистрированный пользователь или профессиональный критик (данные об отзывах критиками добавляются в систему администратором) подробно описывает и оценивает фильм. Рецензия включает в себя анализ сюжета, актёрской игры, режиссуры, музыки и других элементов фильма. В отличие от отзыва, рецензия более развёрнута и может предоставлять экспертное мнение.
- 12. Список избранного** (Favorites List) – коллекция фильмов – нумерованный список, – созданная пользователем для быстрого доступа к тем произведениям, которые он планирует посмотреть или которые ему особенно понравились.
- 13. Пользовательский список** (Custom List) – набор фильмов – нумерованный список, созданный и структурированный зарегистрированным пользователем по собственным критериям. Пользователь может создавать различные списки ("Лучшие комедии", "Фильмы на выходные" – название определяет пользователь) и управлять их конфиденциальностью (общедоступные (может видеть любой зарегистрированный и незарегистрированный пользователь) или приватные – могут видеть только создавший список пользователь, администраторы, модераторы).
- 14. Поиск** (Search) – функционал платформы, позволяющий пользователям находить фильмы, вводя ключевые слова, названия фильмов, имена актёров, жанры или другие параметры. Поиск может быть обычным (по ключевым словам) и расширенным (с использованием фильтров).
- 15. Фильтр** (Filter) – инструмент для сужения списка фильмов по различным критериям, таким как дата публикации, популярность, рейтинг и другие параметры (см. **Страница фильма**). Фильтрация позволяет пользователю быстрее находить фильмы, соответствующие его интересам.
- 16. Расширенный поиск** (Advanced Search) – механизм поиска, позволяющий пользователю одновременно использовать несколько

фильтров и критериев для нахождения фильмов с большей точностью. Можно искать фильмы определённого жанра, выпущенные в конкретном году, с высоким рейтингом и т.д. (определенено в “Страница фильма”).

17. **Лента рекомендаций** (Recommendation Feed) – автоматически сгенерированный список фильмов, который предлагается пользователю на основе его прошлых просмотров, оценок и предпочтений. Лента индивидуальна для каждого пользователя.
18. **Профиль пользователя** (User Profile) – персональная страница пользователя, содержащая информацию о нём, такие как фотография, описание, история просмотров и оценок. Профиль может быть отредактирован в любое время зарегистрированным пользователем, которому принадлежит профиль.
19. **История активности** (Activity History) – список фильмов, которые пользователь просматривал или оценивал. Она помогает пользователю отслеживать свои предпочтения и взаимодействия с платформой.
20. **Восстановление пароля** (Password Recovery) – процесс восстановления доступа к учетной записи в случае, если пользователь забыл пароль. Процедура включает отправку электронного письма с инструкциями для сброса пароля.
21. **Жалоба** (Report) – инструмент для пользователей, позволяющий сигнализировать о проблемах с контентом или другим пользователем (например, нарушении правил платформы). Жалобы направляются модераторам или администраторам для дальнейшего рассмотрения.
22. **Конфиденциальность** (Privacy) – набор настроек и политик, которые регулируют доступ к данным и действиям пользователя. Пользователь может выбирать, какие данные (например, списки фильмов, отзывы) будут общедоступными, а какие – приватными.
23. **Обновление системы** (System Update) – процесс внедрения новых версий программного обеспечения, функций или исправлений, направленный на улучшение производительности платформы, её безопасности или функциональности.
24. **Жанр** (Genre) – категория, к которой относится фильм, определяющая его стиль и основную тематику (комедия, драма, триллер). Жанры используются для поиска и фильтрации контента.

25. **Тег (Tag)** – дополнительная метка, которая может быть присвоена фильму для его категоризации и упрощения поиска. Теги могут описывать сюжетные элементы (например, "путешествия во времени") или стилистические особенности (например, "нуар"). Список тегов может быть расширен или изменен администратором.
26. **Ключевые слова** (Keywords) – слова или фразы, которые пользователь может вводить в поисковую строку для нахождения фильмов. Ключевые слова могут быть связаны с названием фильма, актёрами, жанрами и другими аспектами контента.
27. **Страница фильма** (Movie Page) – отдельная веб-страница на платформе, на которой представлена подробная информация о конкретном фильме или сериале. Страница фильма содержит следующие элементы: **Название фильма** (Title) – полное название фильма, отображаемое на странице; **Описание фильма** (Description) – краткий текст, который предоставляет общую информацию о сюжете, основных персонажах и теме фильма. Описание помогает пользователям понять, о чём фильм, и решить, интересен ли он для просмотра; **Рейтинг фильма** (Rating) – средняя оценка фильма, основанная на оценках зарегистрированных пользователей платформы. Рейтинг отображается в числовом виде (например, 8.2/10) и помогает пользователям быстро оценить общий уровень качества фильма; **Отзывы пользователей** (User Comments) – раздел, в котором зарегистрированные пользователи могут оставлять краткие отзывы на фильм. Каждый отзыв включает текст и, при необходимости, может содержать ссылку на более подробную рецензию; **Рецензии** (Reviews) – более детализированные аналитические тексты от пользователей или профессиональных критиков, где рассматриваются различные аспекты фильма (сюжет, актёрская игра, визуальные эффекты и т.д. (структура рецензии определяется пользователем, пишущим рецензию)). Рецензии могут быть сопровождены рейтингом; **Список актёров и съёмочной группы** (Cast and Crew) – информация о ключевых участниках создания фильма: режиссёре, продюсере, сценаристах, композиторах, а также актёрах, исполнивших главные роли. Часто включает фотографии актёров и ссылки на их другие работы; **Трейлер** (Trailer) – видеоматериал, представляющий основные моменты фильма, доступный для просмотра прямо на странице

фильма. Это помогает пользователям быстро оценить визуальный стиль и тематику фильма; **Список жанров** (Genres) – категории, к которым относится фильм, такие как комедия, драма, триллер и т. д., которые помогают пользователю понять основной стиль фильма; **Теги** (Tags) – метки, описывающие элементы фильма, такие как "экшн", "путешествие во времени", "основывается на реальных событиях", которые помогают пользователям искать и находить фильмы по интересующим их темам; **Дата выхода** (Release Date) – информация о том, когда фильм был выпущен в прокат или на платформе; **Продолжительность** (Duration) – длина фильма, обычно указывается в часах и минутах; **Кнопки взаимодействия** (Interaction Buttons) – элементы управления, позволяющие зарегистрированным пользователям взаимодействовать с фильмом. Это может включать: **Оценить фильм** (Rate Movie) – возможность присвоить фильму числовую оценку; **Добавить в список** (Add to List) – возможность добавить фильм в один из пользовательских списков; **Связанные фильмы** (Related Movies) – список фильмов, похожих (на усмотрение администратора или модератора) по жанру, сюжету или другим параметрам, который автоматически генерируется платформой или вручную добавляется администратором и предлагается пользователю для дальнейшего просмотра.

Страница фильма является ключевой точкой взаимодействия пользователей с контентом на платформе, предоставляя всю необходимую информацию для принятия решения о просмотре фильма.