Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Средства и технологии анализа и разработки информационных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

СИСТЕМА ОЦЕНКИ КИНОФИЬМОВ

БГУИР КП 1-40 01 02-02 301 ПЗ 107

Студент: гр. 572301 Жилинская К.Ю.

Руководитель: Хомяков П. В.

Минск 2018

СОДЕРЖАНИЕ

[Введение 4](#_Toc513734101)

[1 Описание предметной области оценкии и анализа кинофильма 6](#_Toc513734102)

[2 Описание процесса выставления оценки кинофильму 10](#_Toc513734103)

[3 Спецификация вариантов использования системы 16](#_Toc513734104)

[4 Информационная модель системы и её описание 17](#_Toc513734105)

[5 Обоснование выбора компонентов и технологий 21](#_Toc513734106)

[6 Модели представления системы и их описание 27](#_Toc513734107)

[6.1 Диаграмма состояний 27](#_Toc513734108)

[6.2 Алгоритм удаления данных из БД 27](#_Toc513734109)

[6.3 Алгоритм показа страницы фильмов 27](#_Toc513734110)

[7 Описание применения паттернов проектирования 31](#_Toc513734111)

[8 Руководство по развертыванию системы 35](#_Toc513734112)

[9 Результаты тестирования и оценка выполнения задач 36](#_Toc513734113)

[Заключение 40](#_Toc513734114)

[Список использованных источников 41](#_Toc513734115)

[Приложение А (Диаграмма классов) 42](#_Toc513734126)

[Приложение Б (Скрипт БД) 51](#_Toc513734127)

[Приложение В (Листинг кода) 52](#_Toc513734128)

# **ВВЕДЕНИЕ**

За более чем сто лет своего существования, кинематограф стал одним из наиболее популярных и известных видов искусства, но, тем не менее, кино – на рынке всего лишь продукт, требующий, как и все продукты специализированного и научного подхода по популяризации и продвижению в массы. Как и все продукты, он требует грамотного продвижения, тем более учитывая роль рекламной компании в его кассовых сборах и дальнейшей возможности быть узнаваемым.

В настоящее время использование информационных технологий в сервисе оценки кинофильмов не является редкостью. Спектр их применения широк и варьируется от блогов отдельно взятых людей до полноценного сайта со своей системой оценки каждого из фильмов.

Целью данного курсового проекта является создание сервиса оценки кинофильмов, а так же создание удобного и функционального программного обеспечения с понятным пользователю интерфейсом.

Для достижения поставленной цели необходимо решить следующие задачи:

* изучить предметную область сервиса оценки кинофильмов;
* выполнить функциональное моделирование процесса оценки кинофильмов;
* разработать модели представления сервиса оценки кинофильмов на основе UML;
* проанализировать и создать логическую и физическую модель представления данных;
* создать базу данных, хранящую информацию о кинофильмах, о пользователях и объектах, необходимых для выставления оценки кинофильму;
* реализовать возможности регистрации и авторизации;
* разработать программное обеспечение;
* протестировать программный продукт.

Автоматизация сервиса оценки кинофильмов позволит грамотно организовать подсчет и популярность каждого из фильмов, а так же обеспечивает способ своевременного оставления отзывов о просмотренном кинофильме[1].

# **1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ОЦЕНКИИ И АНАЛИЗА КИНОФИЛЬМА**

Предметной областью данного курсового проекта является сервис оценки кинофильмов.

Фильмы – прекрасный пример искусства и развлечения, а пристальное изучение только усиливает магию кинематографа. Если вы пишете рецензию для газеты или оценку на сайте, то вам потребуется разделить фильм на части и пояснить, что они для вас означают. Чтобы сделать глубокий и всесторонний анализ фильма, нужно внимательно просмотреть картину, исследовать все аспекты и сосредоточиться на темах, который задели струны вашей души.

Часть 1 – посмотрите фильм. Далее необходимо выполнить следующие пункты:

1. Если вы никогда ранее не смотрели фильм, который собираетесь анализировать, то не нужно заранее собирать всю доступную информацию. Пусть именно фильм произведет на вас впечатление, а не чужое мнение о нем. Достаточно знать лишь наиболее общие детали, а все остальное лучше почерпнуть из самого фильма.

2. Смотрите фильм без посторонних (или со спокойным другом). Нужно полностью сосредоточиться на фильме, чтобы сделать хороший анализ, поэтому лучше смотреть без отвлекающих факторов. Некоторым людям неловко ходить в кинотеатр в одиночку, но может оказаться, что это достаточно удобно и позволяет лучше сосредоточиться на важных деталях.

3. Посмотрите фильм за один присест. В отличие от телесериалов, фильмы нужно сразу смотреть целиком и полностью. Если прервать просмотр, чтобы сделать бутерброд или сходить в магазин, то вы рискуете упустить эмоции, которые вкладывают в фильм создатели. Сидите спокойно и старайтесь не нажимать на паузу.

4. Делайте записи. Если вы смотрите фильм не в темном кинотеатре, то можно записать несколько общих наблюдений в процессе просмотра. Важно продолжать следить за фильмом, а не за своими записями, поэтому не нужно браться за глубокий анализ во время сеанса. Это можно сделать позже! Не ставьте фильм на паузу.

5. Запишите все свои мысли после просмотра. Запишите все, что отложилось у вас в голове или же показалось важным в тот момент, когда начались финальные титры и ваше сознание еще остается на волне фильма. Ваши мысли все еще сумбурны и не разложены по полочкам, поэтому достаточно записать интересные моменты или детали, которые намеренно подчеркнули создатели фильма.

6. Через время вернитесь к своим мыслям. Спустя день перечитайте записи, сделанные во время и после завершения фильма. Обдумайте, насколько важными оказались подмеченные вами детали фильма от темы самопожертвования до факта, что шляпы носили только отрицательные персонажи. Если вам удалось определить темы, которые кажутся наиболее важными, то можно приступать к анализу и оценке отдельных аспектов фильма.

Часть 2 – оцените отдельные аспекты. Эта часть так же состоит из нескольких пунктов, таких как:

1. Исследуйте информацию о создании фильма. Каждая картина включает как минимум две истории: сюжет и историю создания фильма. На съемки фильмов уходит много времени, усилий и финансовых средств. Изучите историю создания фильма, чтобы глубже понять его суть и отдельные аспекты.

2. Обдумайте сюжетную линию. Фильмы – это способ повествования, поэтому успех фильма всегда зависит от развития сюжета. Оцените темп повествования. Он может быть прерывистым или ровным. Отмечайте важные повороты сюжета.

3. Рассмотрите сценарий. Сценарий поддерживает развитие сюжета, поэтому фильм с грамотным повествованием обычно имеет хороший сценарий. Оцените, насколько сценарий позволяет узнать всю необходимую информацию. Составьте список значимых цитат или фраз.

4. Актерская игра. Вспомните персонажей фильма. Насколько они правдоподобны? Нужно оценивать не то, насколько вам понравится персонаж, а насколько актерская игра позволила вам поверить, что такой человек мог бы реально существовать. Также обратите внимание на актеров в кадре. Если актер не позволяет вам отвлечься ни на секунду, то он хорошо справляется со своей задачей.

5. Освещение и операторская работа. В ужастиках для аутентичности часто используется эффект трясущейся камеры и приглушенный свет. В блокбастерах нередко отдается предпочтение яркому свету и плавной смене кадров, чтобы актеры выглядели безупречно. Постарайтесь определить настроение каждой сцены и оценить, насколько это является заслугой оператора и осветителей. Большое значение имеет угол съемки: так режиссер показывает, с какой позиции мы должны увидеть конкретную сцену. Камера позволяет смотреть на людей сверху вниз или подглядывать из-за угла?

6. Звуковое сопровождение. Музыка в фильмах позволяет любому человеку познакомиться и понять самые разные, даже оркестровые композиции! Оцените громкость, настроение и влияние звуковой дорожки на восприятие сюжета. Хороший саундтрек помогает глубже погрузиться в настроение фильма и даже способствует развитию сюжета. Музыка не должна отвлекать.

7. Оцените реквизиты. Неодушевленные предметы, которые позволяют задать нужную атмосферу, способны многое рассказать о фильме. Режиссер знаменит фирменным эстетическим вкусом? У вас возникают определенные чувства при взгляде на декорации? Возможно, в данном фильме сюжет отходит на второй план, поскольку основная роль отведена реквизитам и декорациям?

Часть 3 – составьте общее мнение. Далее необходимо выполнить следующие пункты:

1. Систематизируйте доказательную базу. Вам нужны факты, которые подтвердят представление о темах фильм. Это могут быть идеи, цвета или повторяющиеся кадры и строчки из диалогов. Пересмотрите свои мысли о различных элементах фильма и постарайтесь найти доказательства, подтверждающие ваши мысли.

2. Начните с введения. Вы уже составили свое мнение о фильме, а теперь пришло время помочь другим в этом нелегком деле. Укажите общие сведения о фильме, включая создателей, а также не забудьте сообщить о своих предварительных ожиданиях. На этом месте можно поделиться своими предположениями, оставив их в виде намеков, а не развернутых идей, которые могут ошарашить читателей.

3. Кратко изложите сюжет. Опишите завязку сюжета и зарождение главного конфликта в нескольких предложениях. Краткость – ваш верный помощник, ведь сюжет является лишь одним из множества аспектов фильма, поэтому вам предстоит еще много работы.

4. Исследуйте темы, которые вас заинтересовали. Познакомив читателей с остовом фильма, не забудьте рассказать также о верхних слоях картины. Можно выдвинуть предположение о том, что режиссер хотел донести своим фильмом или даже на какие мысли пытался натолкнуть зрителей. Найдите факты из фильма, которые подтверждают вашу точку зрения.

5. Обсудите аспекты, которые вам не понравились. Не бойтесь критиковать. Идеальные фильмы можно пересчитать по пальцам, а детальное обсуждение недостатков ленты лишь укрепит ваш анализ. Не стесняйтесь и прямо скажите, что бы вы изменили в фильме. Возможно, заявленные темы можно было раскрыть глубже и доступнее?

6. Завершите анализ. Фильм оправдал ваши ожидания? Каков ваш общий вердикт? Подкрепите свое мнение анализом и фактами. Очевидно, что ваша рецензия не может быть объективной: сообщите свое мнение о том, насколько успешно фильм справился с поставленными целями и насколько он вам понравился [2].

# **2 ОПИСАНИЕ ПРОЦЕССА ВЫСТАВЛЕНИЯ ОЦЕНКИ КИНОФИЛЬМУ**

Для получения более полного представления о том, как можно автоматизировать процесс оценки кинофильмов, была создана функциональная модель.

Ключевой процесс в данной теме – оценивание кинофильма, ради чего и будет писаться программное обеспечение. Для создания же функциональной модели был выбран стандарт IDEF0. IDEF0 — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временна́я последовательность[3].

На контекстной диаграмме верхнего уровня (рисунок 2.1) представлена функциональная модель «Выставление оценки кинофильму», а также определены потоки входных и выходных данных, механизмы ограничения и управления данными.

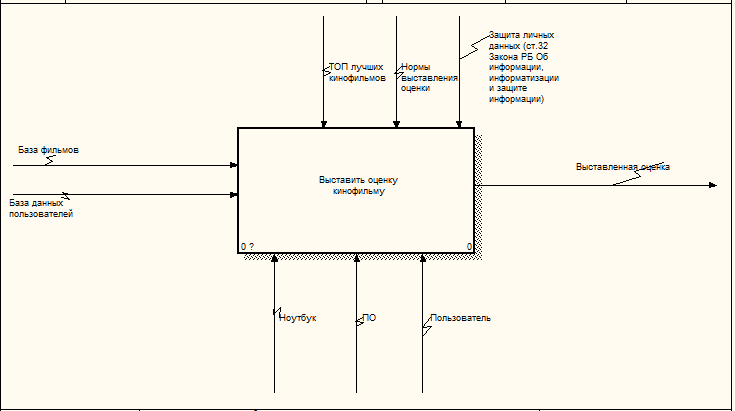


Рисунок 2.1 – Контекстная диаграмма верхнего уровня «Выставить оценку кинофильму»

Входной поток включает в себя базу данных фильмов, а так же базу данных пользователей. После соответствующей обработки в выводном потоке имеем выставленную оценку. В роли управляющих механизмов выступают ТОП лучших кинофильмов, нормы выставления оценок и закон РБ «Об информации, информатизации и защите информации», ст.32. Механизмами являются ноутбук, пользователь и программное обеспечение.

На рисунке 2.2 отображена декомпозиция контекстной диаграммы, состоящая из четырёх блоков: «Найти кинофильм», «Посмотреть кинофильм», «Зайти в аккаунт на сервисе» и «Выставить оценку».

Первый компонент данной декомпозиции («Найти кинофильм») подразумевает нахождение кинофильма с помощью ТОПа лучших кинофильмов.

Декомпозиция данного блока отображена на рисунке 2.3. Она представлена тремя компонентами: «Выбрать жанр», «Выбрать страну производства», «Определиться с фильмом».

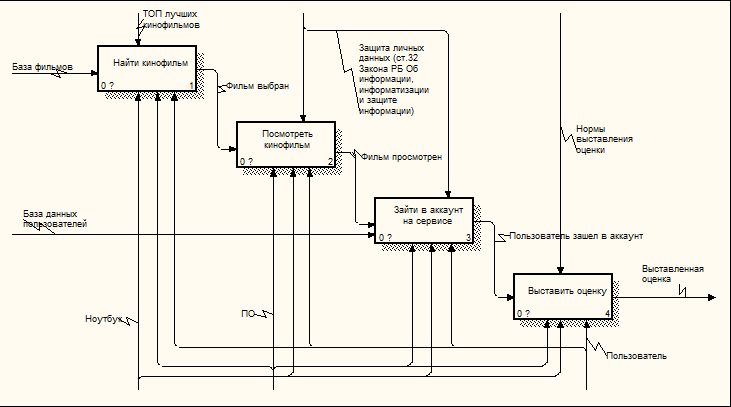


Рисунок 2.2 – Декомпозиция контекстной диаграммы верхнего уровня «Выставить оценку кинофильму»

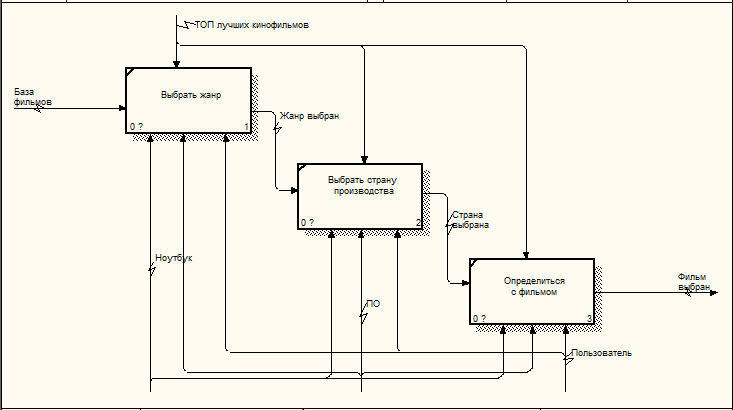


Рисунок 2.3 – Декомпозиция блока «Найти кинофильм»

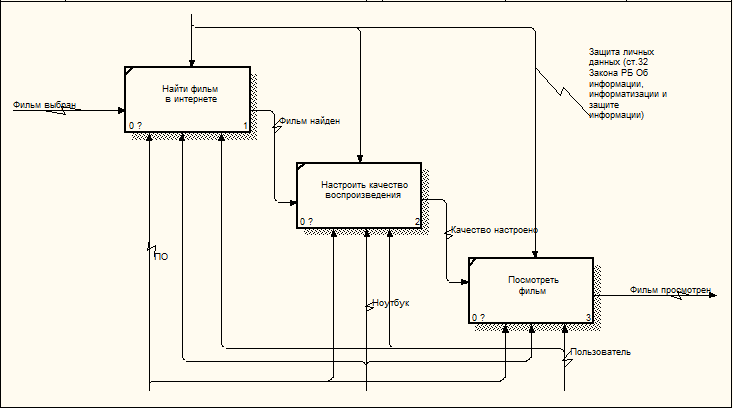


Рисунок 2.4 – Декомпозиция блока «Посмотреть кинофильм»

Второй компонент декомпозиции контекстной диаграммы («Посмотреть кинофильм») включает в себя поиск и просмотр выбранного фильма.

Декомпозиция этого блока отображена на рисунке 2.4. Она представлена тремя компонентами: «Найти фильм в интернете», «Настроить качество воспроизведения» и «Посмотреть фильм».

Третий компонент декомпозиции контекстной диаграммы («Зайти в аккаунт на сервисе») подразумевает собой вход в уже зарегистрированный аккаунт на сервисе. Декомпозиция данного блока отображена на рисунке 2.5. Она представлена тремя компонентами: «Зайти на сайт сервиса», «Ввести логин и пароль от аккаунта» и «Войти в свой аккаунт».

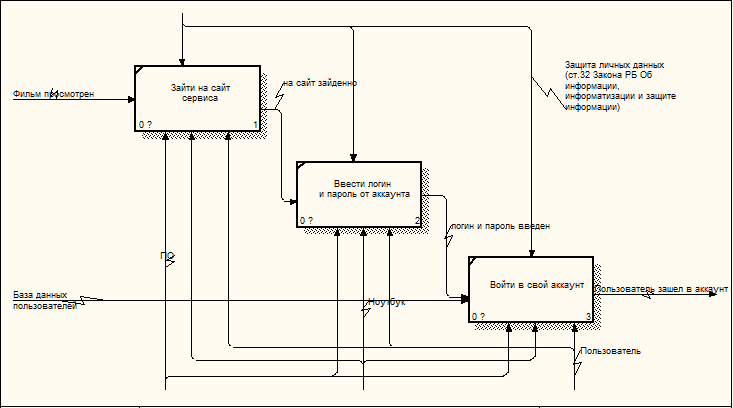


Рисунок 2.5 – Декомпозиция блока «Зайти в аккаунт на сервисе»

Четвертый компонент декомпозиции контекстной диаграммы («Выставить оценку») подразумевает собой выставление оценки на сайте сервиса оценки кинофильмов. Декомпозиция данного блока отображена на рисунке 2.6. Она представлена тремя компонентами: «Найти просмотренный фильм», «Выставить заслуженную оценку» и «Убедиться в верно выставленной оценке».

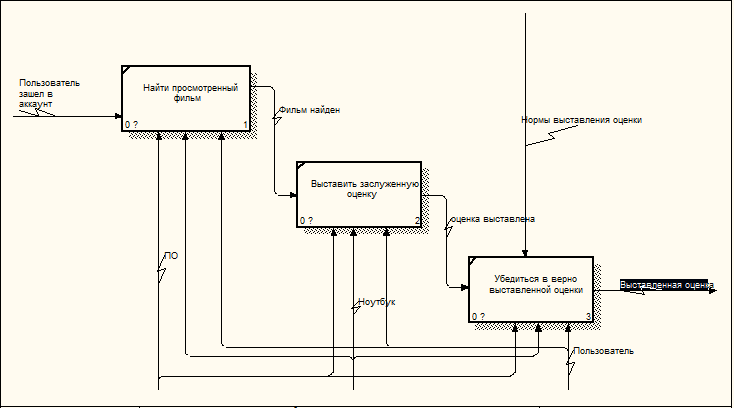


Рисунок 2.5 – Декомпозиция блока «Зайти в аккаунт на сервисе»

Данный продукт открывает некоторые перспективы для упрощения поиска фильма. Пользователи смогут без труда получить информацию о самых интересных фильмах по мнению других пользователей, которые уже посмотрели выбранный фильм.

# **3 СПЕЦИФИКАЦИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ**

Диаграммы вариантов использования описывают взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе.

На данной диаграмме представлены 2 актера: пользователь (студент) и администратор. Актеры данной системы имею общие варианты использования, такие как: авторизация, фильтрация данных о студенте, а так же сортировка. Но при этом каждый актер имеет и уникальные варианты использования. Представление диаграммы показано на рисунке 3.1.

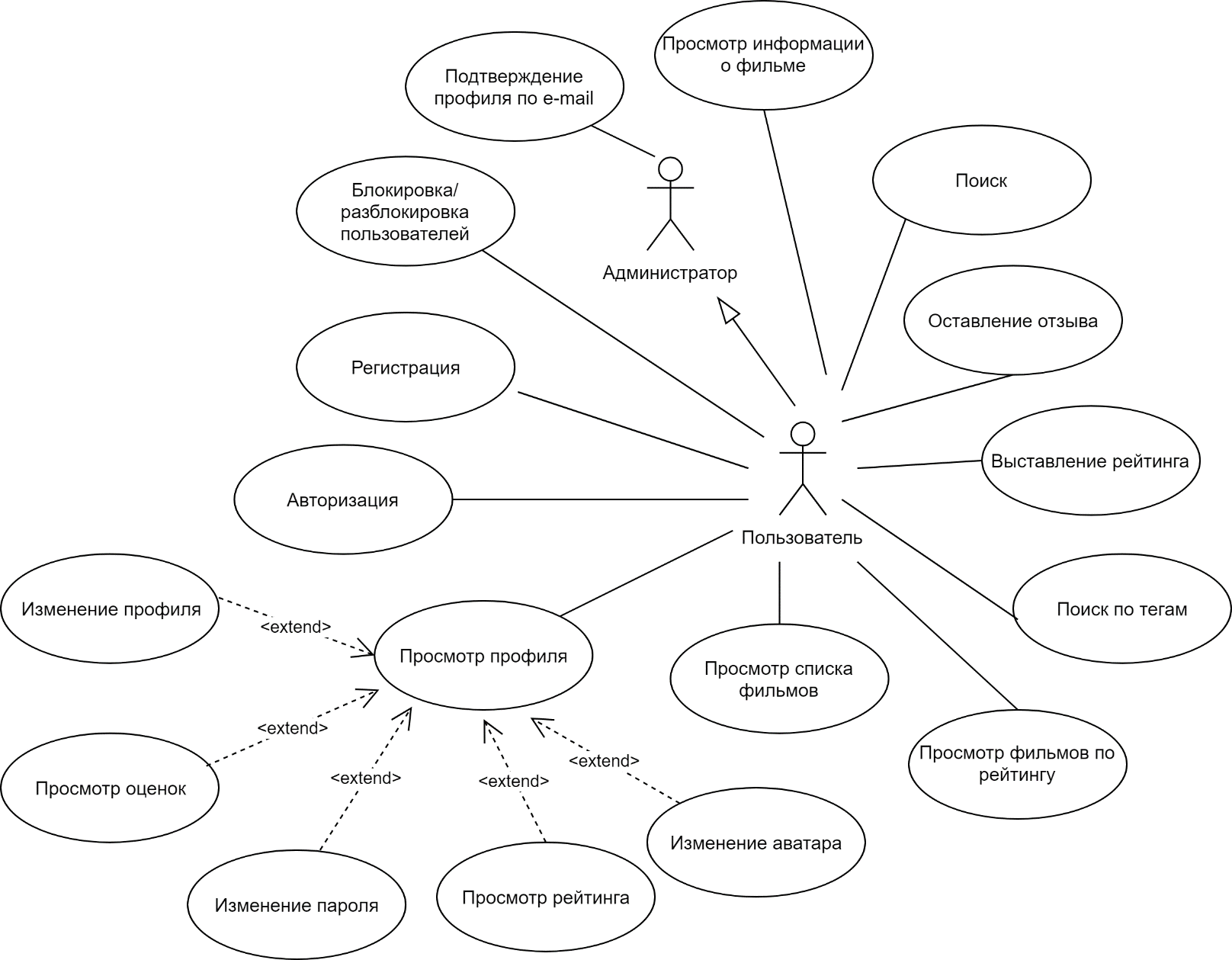


Рисунок 3.1 – Диаграмма вариантов использования

# **4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЁ ОПИСАНИЕ**

Для качественной работы системы оценки кинофильмов потребовалось выделить и создать шесть сущностей: фильм, жанр, рейтинг, отзыв, режиссер и пользователь.

Информационная модель системы в графическом виде представлена ниже на рисунке 4.1.

Сущность «Пользователь» предназначена для хранения информации о пользователях системы. Данные из этой таблицы используются при авторизации в приложении. Сущность содержит в себе следующие атрибуты:

− iduser – содержит идентификационный номер пользователя приложения(уникальный);

− login – содержит имя пользователя;

− password – содержит пароль, используемый для авторизации в системе;

− email – содержит адрес почты пользователя;

− real\_name – содержит настоящее имя пользователя, отображающееся на сайте;

− date\_of\_registration – дата регистрации на сайте;

− date\_of\_birth – дата рождения пользователя;

− status – содержит статус пользователя, например «заблокирован»;

− avatar – содерит ссылку на аватар пользователя;

− level\_points – количество очков уровня пользователя;

− role – хранит роль пользователя. По умолчанию роль соответствует «пользователю».

Сущность «Рейтинг» предназначена для хранения информации о рейтинге фильма. Сущность содержит в себе следующие атрибуты:

− idrating– содержит идентификационный номер записи(уникальный);

− iduser – содержит идентификационный номер пользователя, которому принадлежит оценка(уникальный);

− idfilm – содержит идентификационный номер фильма, которому предназначена оценка(уникальный);

− is\_seen – содержит информацию о том, просмотрен ли фильм;

− rating\_amount – содержит количественное выражение оценки;

Сущность «Фильм» предназначена для хранения информации о фильме. Сущность содержит в себе следующие атрибуты:

− idfilm– содержит идентификационный номер записи (уникальный);

− name – содержит название фильма;

− release\_year – содержит год выпуска фильма;

− duration – содержит длительность фильма в минутах;

− poster – содержит ссылку на постер;

− iddirector – содержитй идентификационный номер режиссера данного фильма (уникальный);

− description – содержит описание фильма;

Сущность «Отзыв» предназначена для хранения информации об отзыве. Сущность содержит в себе следующие атрибуты:

− idreview– содержит идентификационный номер записи (уникальный);

− iduser - содержит идентификационный номер пользователя, которому принадлежит отзыв (уникальный);

− idfilm - содержит идентификационный номер фильма (уникальный);

− text – содержит текст отзыва;

− date – содержит дату, когда был написан отзыв;

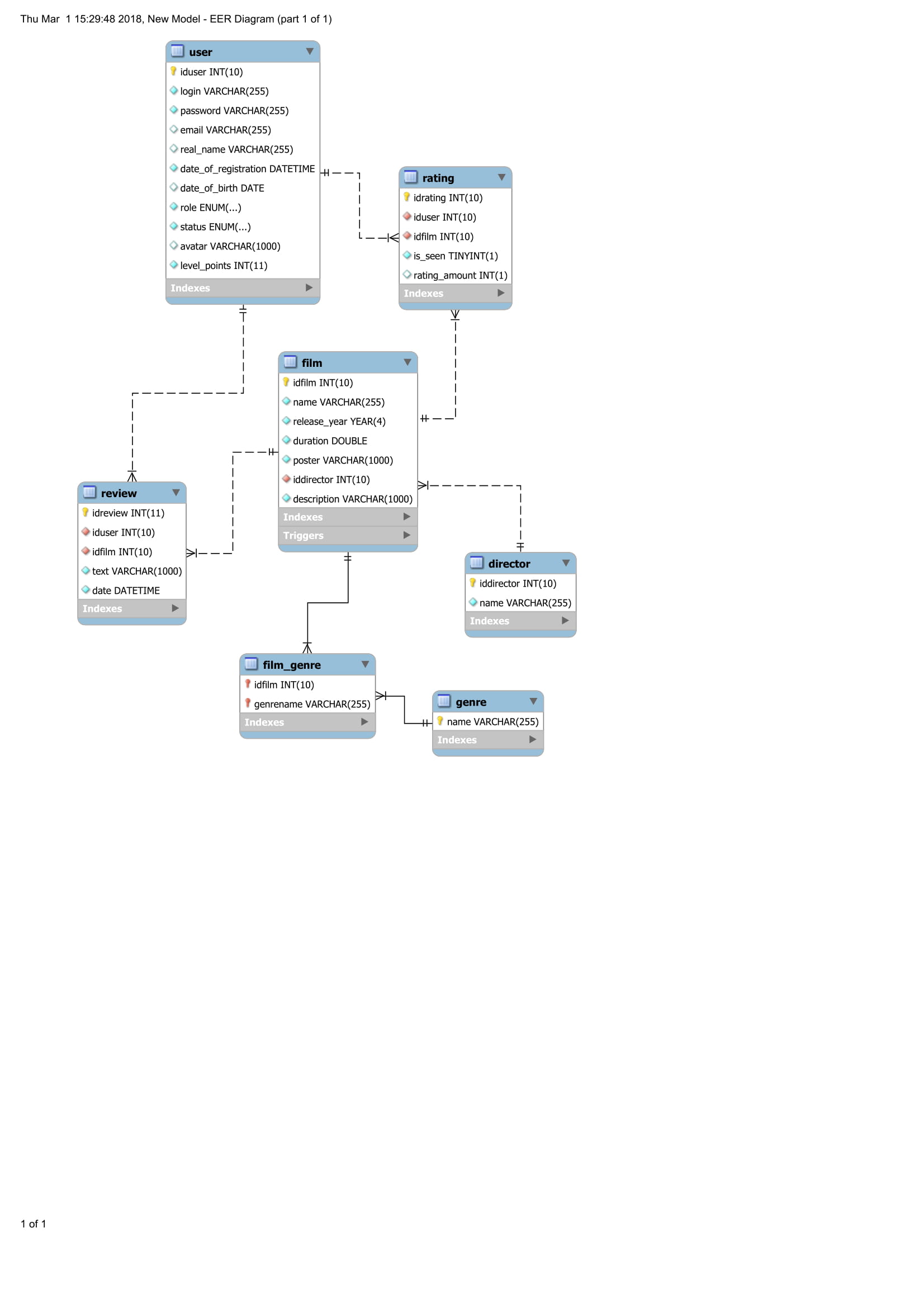


Рисунок 4.1 – Информационная модель системы

Сущность «Жанр» предназначена для хранения информации о жанре. Сущность содержит в себе следующие атрибуты:

− name– содержит название жанра;

Сущность «Режиссер» предназначена для хранения информации о режиссере. Сущность содержит в себе следующие атрибуты:

− iddirector– хранит уникальный идентификационный номер записи;

− name – хранит имя режиссера;

# **5 ОБОСНОВАНИЕ ВЫБОРА КОМПОНЕНТОВ И ТЕХНОЛОГИЙ**

В качестве основной среды разработки в проекте используется IntelliJ IDEA 2017.2.5.

При написании программного кода используется язык Java. Java это объектно-ориентированный язык программирования. Изначально язык программирования назывался Oak (James Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения [4].

У языка Java существует множество преимуществ перед другими языками программирования, что позволяет решать с его помощью практически любые задачи. Так же с помощью Java можно создавать модульные программы, исходный код которых может использоваться многократно.

Одним из основных преимуществ языка Java является возможность переноса программ из одной системы в другую. Поскольку программы на Java не зависят от платформы как на уровне исходного кода, так и на двоичном уровне, их можно запускать в различных системах, что особенно важно для программ, предназначенных для World Wide Web [5].

Работа приложения основывается на том, что клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него, в то время как серверная часть получает запрос от клиента, выполняет вычисления, после чего формирует ответ и отправляет её клиенту. Выполнение всех описанных сетевых функций приложения без проблем реализуется с помощью Java, так как сетевая работа является одной из сильных сторон данного языка программирования.

Огромное количество библиотек классов в Java многократно упрощают работу программиста. Java предоставляет программисту богатый набор классов объектов для ясного абстрагирования многих системных функций, используемых при работе с окнами, сетью и для ввода-вывода. Ключевая черта этих классов заключается в том, что они обеспечивают создание независимых от используемой платформы абстракций для широкого спектра системных интерфейсов.

Так же в курсовом проекте используются СУБД. Система управления базами данных - это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д. В общем СУБД - это система, позволяющая создавать базы данных и манипулировать сведениями из них. А осуществляет этот доступ к данным СУБД посредством специального языка - SQL. SQL (Stuctured Query Language ) - язык структурированных запросов, основной задачей которого является предоставление простого способа считывания и записи информации в базу данных. SQL включает много разных типов операторов, разработанных для взаимодействия с базами данных.

Для реализации хранения, обработки и дальнейшего использования информации в данном приложении используется СУБД MySQL.

MySQL является наиболее приспособленной для применения в среде веб системой управления базами данных.

Основные преимущества MySQL:

− многопоточность, поддержка нескольких одновременных запросов;

− оптимизация связей с присоединением многих данных за один проход;

− записи фиксированной и переменной длины;

− гибкая поддержка форматов чисел, строк переменной длины и меток времени;

− быстрая работа, масштабируемость.

 Таким образом, базы данных имеют достаточное количество явно неоспоримых преимуществ, используемых для решения задач данного курсового проекта.

Так же в курсовом проекте используется ORM.ORM ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Object-Relational Mapping) − технология программирования, которая связывает [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)с концепциями [объектно-ориентированных языков программирования](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), создавая «виртуальную [объектную базу](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) данных». Существуют как [проприетарные](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5" \o "Проприетарное программное обеспечение), так и [свободные](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) реализации этой технологии [6].

Для отображение данных на web-сайте используются jsp страницы. Cервлеты позволяют получать запросы от клиента, совершать некоторую работу и выводить результаты на экран. Технология проектирования Java Server Pages (JSP) - это одна из технологий J2EE, которая представляет собой расширение технологии сервлетов для упрощения работы с Web-содержимым. Страницы JSP позволяет легко разделить Web-содержимое на статическую и динамическую часть, допускающую многократное использование ранее определенных компонентов. Разработчики Java Server Pages могут использовать компоненты JavaBeans и создавать собственные библиотеки нестандартных тегов, которые инкапсулируют сложные динамические функциональные средства.Как и сервлеты, компоненты JSP относятся к компонентам Web и располагаются в Web-контейнере. Страницы JSP не зависят от конкретной реализации Web-контейнера, что обеспечивает возможность их повторного использования [7].

Для подгрузки данных на страницу использутеся AJAX. AJAX (Asynchronous JavaScript and XML) - это способ отправки данных на сервер и получения ответа без перезагрузки страницы. Это называется асинхронными запросами. AJAX не является языком программирования, а концепцией обмена данными с сервером. На определенном этапе работы скрипта на клиентской стороне (JS) вызывается функция, которая отправляет запрос на сервер и указывает какой скрипт запустить на стороне сервера (Java). На сервере запускается скрипт, по результатам работы клиенту приходит ответ и скрипт на клиентской стороне продолжает работу[8].

Для создания UML-диаграмм в работе было использовано средство Enterprise Architect, которое позволяет осуществлять создание диаграмм вариантов использования, состояний, последовательность с помощью просто моделирования. Для выполнения UML-моделей в стандарте IDEF0 использовалось CASE-средство CA AllFusion Process Modeler r7 (BPwin). Для информационного моделирования применялось средство MySQL Workbench 6.3 CE.

Для более подробного изучения и рассмотрения системы программы необходимо рассмотреть такие модели как:

* диаграмма развертывания;
* диаграмма последовательностей;
* диаграмма компонентов.

Система оценки кинофильмов реализована на операционной системе Windows 10 . На диаграмме развертывания (рисунок 5.1), показана конфигурация обрабатывающих узлов, на которых выполняется система, и компонентов, размещенных в этих узлах.

****

Рисунок 5.1 – Диаграмма развёртывания системы оценки кинофильмов

Диаграмма последовательности  — [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) в рамках какого-либо определённого прецедента (отправка запросов и получение ответов)[9]. Представление диаграммы показано на рисунке 5.2.



Рисунок 5.2 – Диаграмма последовательностей входа в систему оценки кинофильмов

Диаграмма компонентов – диаграмма физического уровня, которая служит для представления программных компонентов и зависимостей между ними. Диаграмма компонентов разрабатывается для следующих целей: визуализация общей структуры исходного кода программной системы и спецификация исполнимого варианта программной системы[10] (рисунок 5.3).



Рисунок 5.3 – Диаграмма компонентов системы оценки кинофильмов

Диаграмма классов используется для визуального изображения отношений между классами и интерфейсами в программе. В приложении А представлена диаграмма классов приложения.

# **6 МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ**

## 6.1 Диаграмма состояний

Диаграмма состояний (state diagram) определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате влияния некоторых событий. Диаграммы состояний строятся для единственного класса и описывают поведение единственного объекта[11].

Диаграмма состояний представляет собой граф состояний в которых может находиться объект и связей между ними (рисунок 6.1).

## 6.2 Алгоритм удаления данных из БД

Алгоритм обновления данных о пользователях из БД состоит из нескольких шагов.

1. Соединение с базой данных.
2. Формирование запроса на удаление.
3. Выборка данных из БД по id.
4. Выполнение запроса.

Данный алгоритм представлен на рисунке 6.2.

## 6.3 Алгоритм показа страницы фильмов

Алгоритм показа страницы фильмов состоит из нескольких шагов.

1. Получение со страницы номера страницы.
2. Получение со страницы количества записей на странице.
3. Установка атрибута «Список жанров».
4. Установка атрибута «Список режиссёров».
5. Установка атрибута «Количество фильмов».

Данный алгоритм представлен на рисунке 6.3.



Рисунок 6.1 – Диаграмма состояний при регистрации нового пользователя

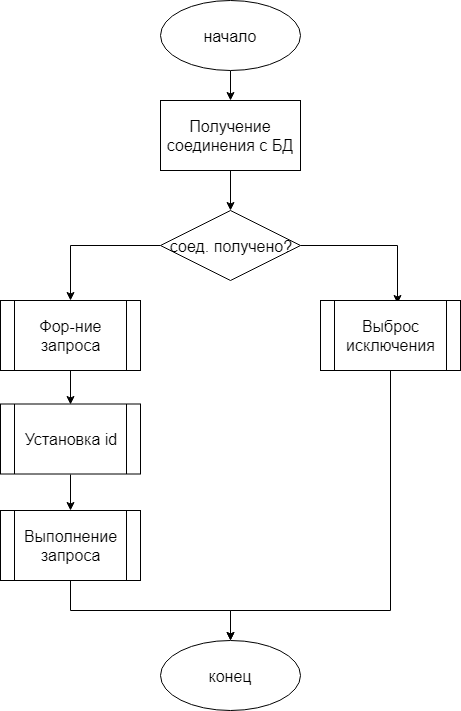


Рисунок 6.2 – Алгоритм удаления пользователя из БД



Рисунок 6.2 – Алгоритм показа страницы фильмов

# **7 ОПИСАНИЕ ПРИМЕНЕНИЯ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ**

Часто в системе могут существовать сущности только в единственном экземпляре, например, система оценки кинофильмов. В таких случаях необходимо уметь создавать единственный экземпляр некоторого типа, предоставлять к нему доступ извне и запрещать создание нескольких экземпляров того же типа.Паттерн Singleton предоставляет такие возможности.

Архитектура паттерна Singleton основана на идее использования глобальной переменной, имеющей следующие важные свойства:

− такая переменная доступна всегда. Время жизни глобальной переменной - от запуска программы до ее завершения;

− предоставляет глобальный доступ, то есть, такая переменная может быть доступна из любой части программы.

Однако, использовать глобальную переменную некоторого типа непосредственно невозможно, так как существует проблема обеспечения единственности экземпляра, а именно, возможно создание нескольких переменных того же самого типа (например, стековых).

Для решения этой проблемы паттерн Singleton возлагает контроль над созданием единственного объекта на сам класс. Доступ к этому объекту осуществляется через статическую функцию-член класса, которая возвращает указатель или ссылку на него. Этот объект будет создан только при первом обращении к методу, а все последующие вызовы просто возвращают его адрес. Для обеспечения уникальности объекта, конструкторы и оператор присваивания объявляются закрытыми [12]. UML-диаграмма классов паттерна Singleton можно увидеть на рисунке 6.1.

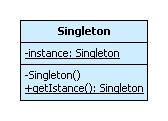


Рисунок 6.1 –Паттерн Синглтон

Достоинства паттерна Singleton:

Класс сам контролирует процесс создания единственного экземпляра, паттерн легко адаптировать для создания нужного числа экземпляров, а также возможность создания объектов классов, производных от Singleton.

Недостатки паттерна Singleton:

В случае использования нескольких взаимозависимых одиночек их реализация может резко усложниться.

Паттерн Command преобразовывает запрос на выполнение действия в отдельный объект-команду. Такая инкапсуляция позволяет передавать эти действия другим функциям и объектам в качестве параметра, приказывая им выполнить запрошенную операцию. Команда – это объект, поэтому над ней допустимы любые операции, что и над объектом.

Интерфейс командного объекта определяется абстрактным базовым классом Command и в самом простом случае имеет единственный метод execute(). Производные классы определяют получателя запроса (указатель на объект-получатель) и необходимую для выполнения операцию (метод этого объекта). Метод execute() подклассов Command просто вызывает нужную операцию получателя [13]. UML-диаграмма классов паттерна Command можно увидеть на рисунке 6.2.

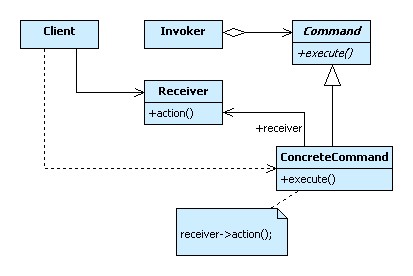


Рисунок 6.2 –Паттерн Команда

Для того, чтобы система оставалась независимой от различных типов объектов, паттерн Factory Method использует механизм полиморфизма - классы всех конечных типов наследуют от одного абстрактного базового класса, предназначенного для полиморфного использования. В этом базовом классе определяется единый интерфейс, через который пользователь будет оперировать объектами конечных типов.

Для обеспечения относительно простого добавления в систему новых типов паттерн Factory Method локализует создание объектов конкретных типов в специальном классе-фабрике. Методы этого класса, посредством которых создаются объекты конкретных классов, называются фабричными. Существуют две разновидности паттерна Factory Method:

Обобщенный конструктор, когда в том же самом полиморфном базовом классе, от которого наследуют производные классы всех создаваемых в системе типов, определяется статический фабричный метод. В качестве параметра в этот метод должен передаваться идентификатор типа создаваемого объекта [14]. UML-диаграмма классов паттерна Factory Method можно увидеть на рисунке 6.3.

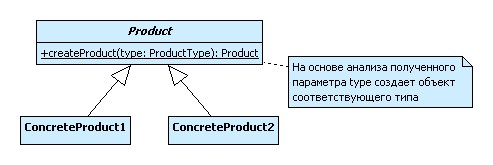


Рисунок 6.3 –Паттерн Фабричный метод

Достоинства паттерна Factory Method:

− создает объекты разных типов, позволяя системе оставаться независимой как от самого процесса создания, так и от типов создаваемых объектов.

Недостатки паттерна Factory Method:

− в случае классического варианта паттерна даже для порождения единственного объекта необходимо создавать соответствующую фабрику.

# **8 РУКОВОДСТВО ПО РАЗВЕРТЫВАНИЮ СИСТЕМЫ**

Для того, чтобы развернуть онлайн-сервис оценки кинофильмов необходимо выполнить нижеперечисленные шаги.

Во-первых необходимо осуществить подключение к базе данных. В данной системе используется СУБД MySQL и для правильности взаимодействия сервиса с базой данных необходимо настроить некоторые параметры.

Для начала необходимо скачать и установить MySQL. Для создания базы данных для разрабатываемой системы необходимо выполнить скрипт, находящийся в приложении под названием db\_script.sql . После этого необходимо проверить настройки для доступа к базе в файле project.properties, находящемся в movierating/src/main/resources. В случае несовпадения, исправить на корректные.

Во-вторых необходимо загрузить сервер на хостинг. Для этого необходимо скачать Apache Maven и Apache Tomcat и запустить проект уже с помощью Tomcat.

Из всего вышеперечисленного следует, что для корректной работы и обеспечения полного функционирования системы необходимы такие инструменты, как СУБД MySQL, Apache Maven и Apache Tomcat.

# **9 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ И ОЦЕНКА ВЫПОЛНЕНИЯ ЗАДАЧ**

В результате выполнения курсового проекта было разработано web-приложение системы оценки кинофильмов.

Для корректной работы программы необходимо проверить систему на обработку исключительных ситуаций.

Если введены неверные логин или пароль, то система выдаст предупреждение об этом. (Рисунок 9.1)

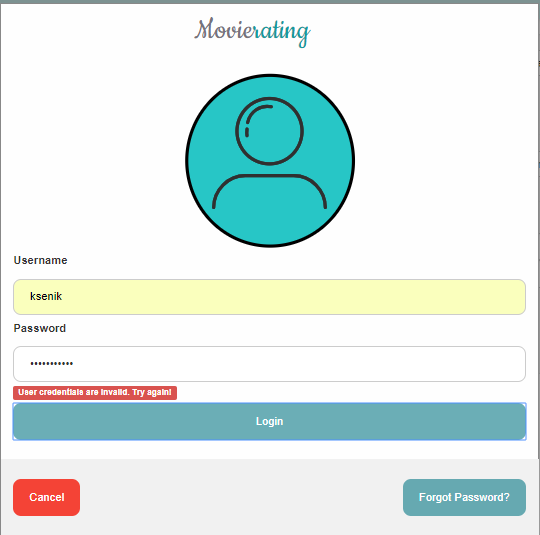


Рисунок 9.1 – Неверный логин или пароль.

При регистрации, необходимо заполнить все поля корректно. Если же они не введены либо введены с ошибкой, то система так же выдает предупреждение. (Рисунок 9.2)

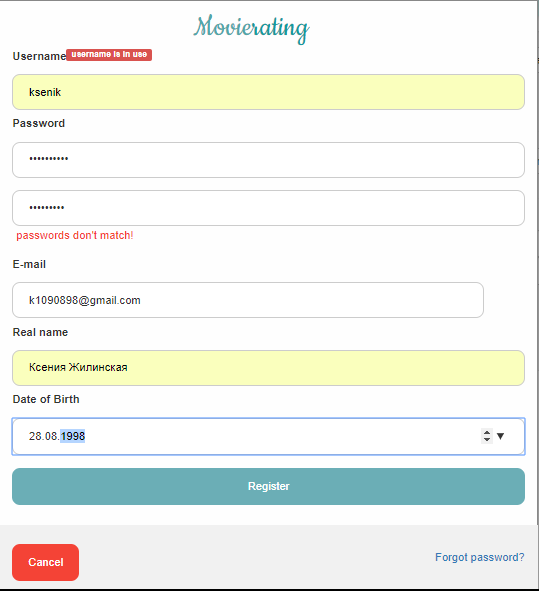


Рисунок 9.2 – Предупреждение о некорректности введенных данных.

При поиска фильма, например, по году, необходимо ввести конкретный промежуток. Если же они не введены либо введены с ошибкой, то система так же выдает предупреждение. (Рисунок 9.3)

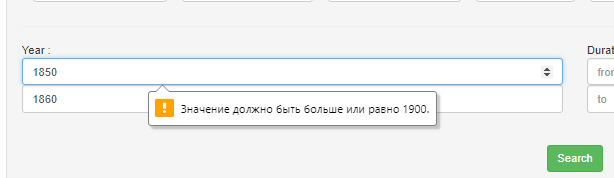


Рисунок 9.3 – Предупреждение о некорректном значении года.

Так же, если пользователь не подтвердил свой email, то в профиле ему об этом будет постоянно напоминать система. (Рисунок 9.4)

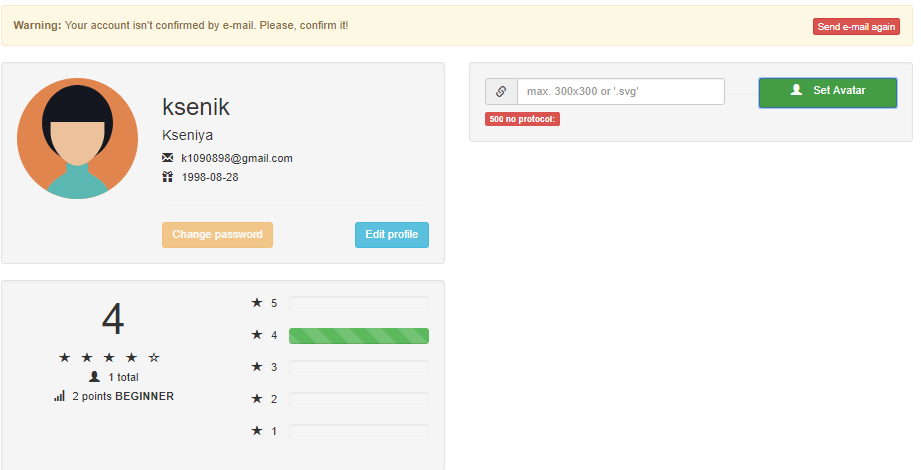


Рисунок 9.4 – Напоминание о подтверждении.

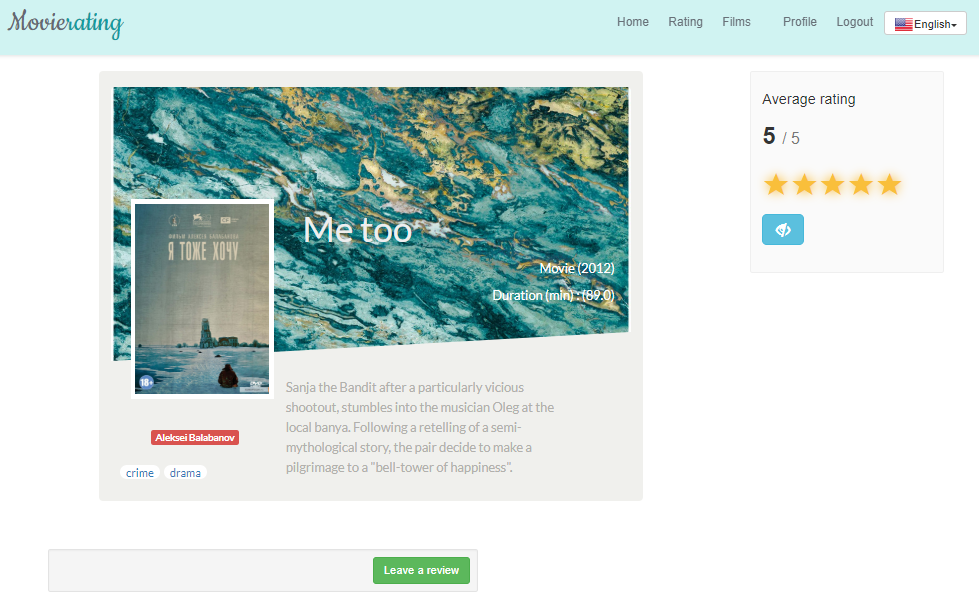
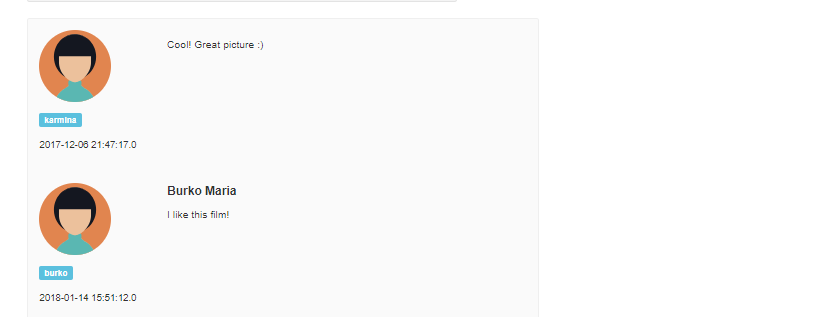
 

Рисунок 9.5 – Пример оставленного комментария.

В нашей системе оценки кинофильмов можно оставлять комментарии. (Рисунок 9.5)

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта нами была разработана программа, которая имеет удобный интерфейс, понятное меню и набор функций для работы с данными о кинофильмах и выставления им оценки. В ней присутствует меню администратора и меню пользователя, в которых реализованы функции просмотра данных о кинофильмах, их оценивании, добавления, удаления и редактирования данных о самом пользователе, а также разного рода фильтрации, сортировки кинофильмов по году и длительности. Вся информация хранится в базе данных.

Нами была изучена предметная область и теоретические аспекты в области оценивания кинофильмов, а так же изучены и рассмотрены различные жанры.

На основе изученной предметной области было разработано Web-приложение, которое заметно упрощает поиск кинофильма на вечер, либо в любое другое время.

Разработанная программа может понадобиться для людей, которые ищут как хорошо провести время, либо же просто не могут определиться с выбором фильма.

В силу того, что программа очень проста в использовании, а так же хорошо спроектирована, в перспективе её можно редактировать, совершенствоватьи перенести на другой язык программирования.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Курсовой проект Жилинской Ксении 5 семестра

[2] [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikihow.com/сделать-анализ-фильма

[3] [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/IDEF0

[4] [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://progopedia.ru/language/java/>

[5] [Электронный ресурс]. – Электронные данные. – Режим доступа https://www.ibm.com/support/knowledgecenter/ru/ssw\_aix\_72/com.ibm.aix.performance/advantages\_java.htm

[6] [Электронный ресурс]. – Электронные данные. – Режим доступа https://ru.wikipedia.org/wiki/ORM

[7] [Электронный ресурс]. – Электронные данные. – Режим доступа http://www.java2ee.ru/jsp/

[8] [Электронный ресурс]. – Электронные данные. – Режим доступа https://devionity.com/ru/courses/js-fundamentals-and-jquery/what-is-ajax

[9] [Электронный ресурс]. – Электронные данные. – Режим доступа <https://ru.wikipedia.org/wiki/>Диаграмма\_последовательности

[10] [Электронный ресурс]. – Электронные данные. – Режим доступа https://studfiles.net/preview/3529703/

[11] [Электронный ресурс]. – Электронные данные. – Режим доступа http://old.exponenta.ru/soft/others/mvs/stud2/26.asp

[12] [Электронный ресурс]. – Электронные данные. – Режим доступа http://cpp-reference.ru/patterns/creational-patterns/singleton/

[13] [Электронный ресурс]. – Электронные данные. – Режим доступа http://cpp-reference.ru/patterns/behavioral-patterns/command/

[14] [Электронный ресурс]. – Электронные данные. – Режим доступа http://cpp-reference.ru/patterns/creational-patterns/factory-method/

# **ПРИЛОЖЕНИЕ А**

**(Диаграмма классов)**



Рисунок А.1 – Диаграмма классов пакета command

**Продолжение приложения А**



Рисунок А.2 – Диаграмма классов пакета config

**Продолжение приложения А**



Рисунок А.3 – Диаграмма классов пакета controller



Рисунок А.4 – Диаграмма классов пакета dao

**Продолжение приложения А**



Рисунок А.5 – Диаграмма классов пакета connection

**Продолжение приложения А**



Рисунок А.6 – Диаграмма классов пакета entity

**Продолжение приложения А**



Рисунок А.7 – Диаграмма классов пакета entity

**Продолжение приложения А**



Рисунок А.8 – Диаграмма классов пакета exception



Рисунок А.9 – Диаграмма классов пакета filter

**Продолжение приложения А**



Рисунок А.10 – Диаграмма классов пакета listener



Рисунок А.11 – Диаграмма классов пакета util

**Продолжение приложения А**



Рисунок А.12 – Диаграмма классов пакета service

# **ПРИЛОЖЕНИЕ Б**

**(Скрипт БД)**

# **ПРИЛОЖЕНИЕ В**

**(Листинг кода)**