# **ВВЕДЕНИЕ**

История существования кинематографа насчитывает более ста лет. За эти годы 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.

Использование информационных технологий в сервисе оценки и подбора кинофильмов является неотъемлемой частью разувающегося информационного пространства. 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 удобного, полезного и функционального сайта для оценки кинофильмов 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.

После решения поставленных задач, приложение, разработанное в данном курсовом проекте, позволит оптимизировать процесс автоматизации сервиса оценки фильмов представляется возможность удобной организации рейтинга популярности представленных кинофильмов, а также помогает обеспечить фильмы релевантными отзывами.

**1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

Предметная область данной курсовой работы – сервис оценки кинофильмов.

Задачей курсового проекта является создание приложения, которое позволило бы автоматизировать процессы оценки кинофильмов.

При написании программного кода необходимо использовать язык Java. Java — объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Изначально новый 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 [3].

У языка Java есть много преимуществ перед другими языками программирования, что позволяет решать с его помощью практически любые задачи. Java - это объектно-ориентированный язык. Это позволяет 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 [4].

Широкие возможности Java, простота применения, независимость от платформы и встроенные функции защиты делают этот язык программирования одним из лучших для создания приложений для Internet.

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.

В Java имеется изящный и вместе с тем сложный механизм реализации взаимодействия потоков, он позволяет сосредоточиться непосредственно на задаче - без раздумий о том, как бы организовать взаимодействие потоков самому [5].

Многочисленные библиотеки классов в Java значительно упрощают работу программиста и помогают сделать красивый и понятный код. 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 [3].

При выполнении предыдущих курсовых работ мы использовали файлы для хранения данных, помещая в них данные, а затем извлекая для последующей работы. Данный способ накладывает дополнительные ограничения на работу программы, значительно увеличивает количество и сложность кода. Вместе с этим возрастает вероятность совершения ошибки при написании программы.

Для решения вышеназванный проблем придумано отличное средство – база данных. База данных - набор сведений, хранящихся некоторым упорядоченным способом. Базы данных сами заботятся о безопасности информации и её сортировке и позволяют извлекать и размещать информацию при помощи небольшого количества специализированного кода. При использовании базы данных код получается более компактным, понятным и легче тестируется и отлаживается.

В противоположность файловой организации хранения информации, использование баз данных дает неоспроримые преимущества. Например, легко организовывать поиск, сортировать записи по дате и времени, проводить различные отборы записей, так как в БД предусмотрена эффективная организация хранения информации, минимизирующая время доступа и поиска. Можно быстро найти среди многих тысяч записей одну определенную (по заданному идентификатору).

Базы данных лишены большого недостатка файлов: с ними нет проблем с совместным доступом к данным. Сценарий, который изменяет файл в процессе своей работы, могут запустить одновременно два человека, и, если если не принять мер по блокировке файла, то могут возникнуть проблемы. С базами данных таких проблем не существует, потому что предусмотрено решение проблем совместного доступа на низком уровне с максимальной эффективностью.

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 – это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. MySQL – свободно распространяемая система, т.е. платить за ее применение не нужно. Кроме того, это достаточно быстрая, надежная и, главное, простая в использовании СУБД, вполне подходящая для не слишком глобальных проектов [6].

MySQL Workbench представляет собой единый визуальный инструмент для разработчиков баз данных, разработчиков и администраторов баз данных. MySQL Workbench обеспечивает моделирование данных, разработку SQL и комплексные средства администрирования для настройки сервера, управления пользователями, резервного копирования [7].

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

**2 ОПИСАНИЕ ПРОЦЕССА ВЫСТАВЛЕНИЯ ОЦЕНКИ КИНОФИЛЬМУ**

Чтобы получить полное представление о том, как можно автоматизировать процесс оценки кинофильмов, 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 временная 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.

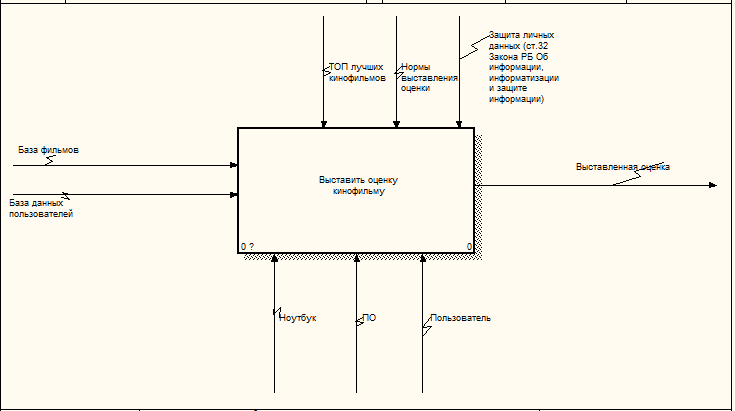


Рисунок 2.1 – Контекстная диаграмма верхнего уровня «Выставить оценку кинофильму»

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 режиссера», «Определиться с фильмом».

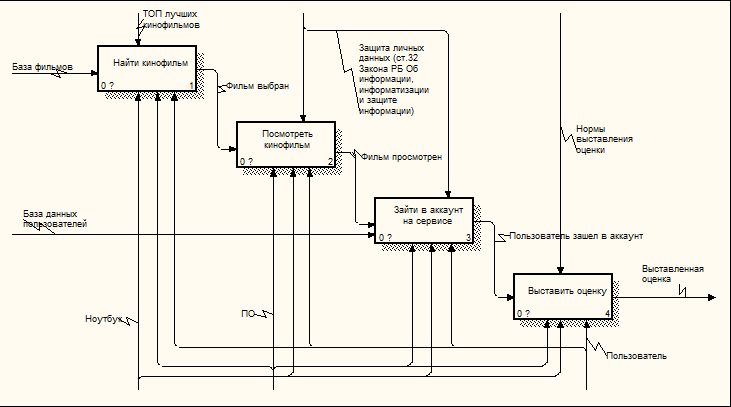


Рисунок 2.2 – Декомпозиция контекстной диаграммы верхнего уровня «Выставить оценку кинофильму»

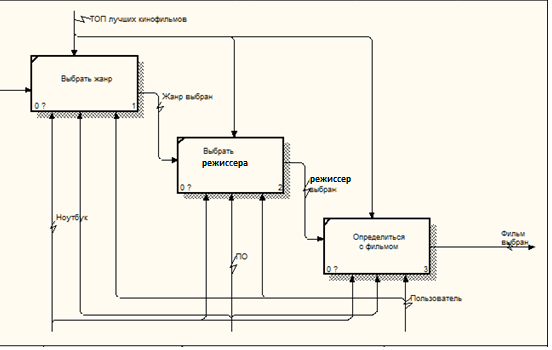


Рисунок 2.3 – Декомпозиция блока «Найти кинофильм»

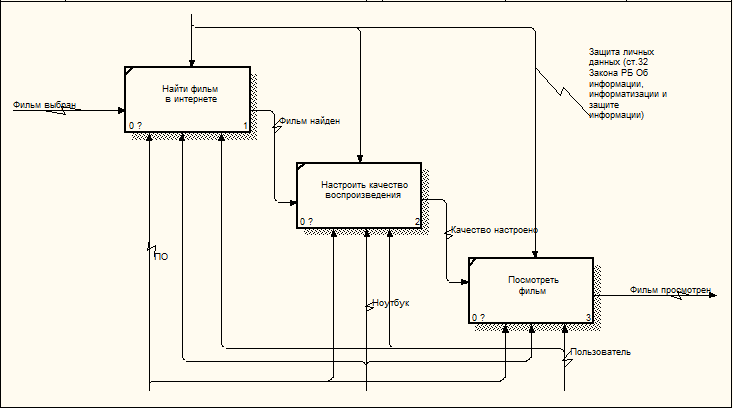


Рисунок 2.4 – Декомпозиция блока «Посмотреть кинофильм»

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».

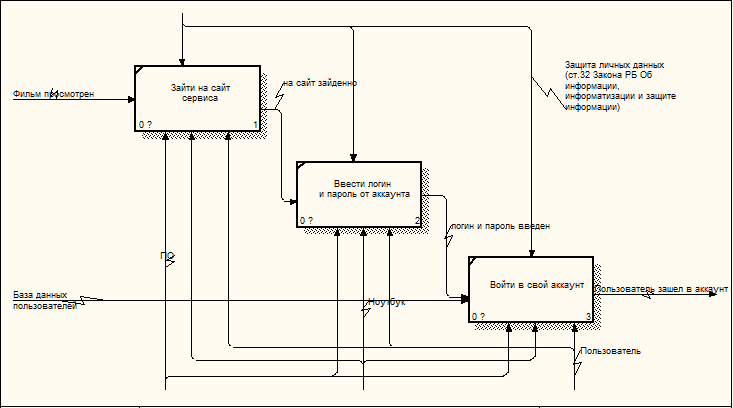


Рисунок 2.5 – Декомпозиция блока «Зайти в аккаунт на сервисе»

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».

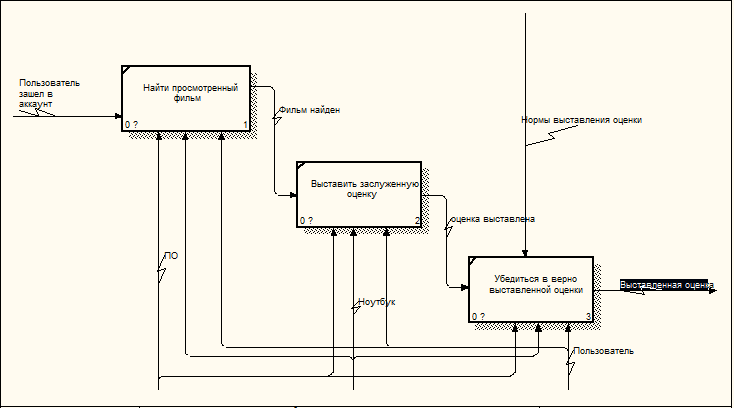


Рисунок 2.5 – Декомпозиция блока «Зайти в аккаунт на сервисе»

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**

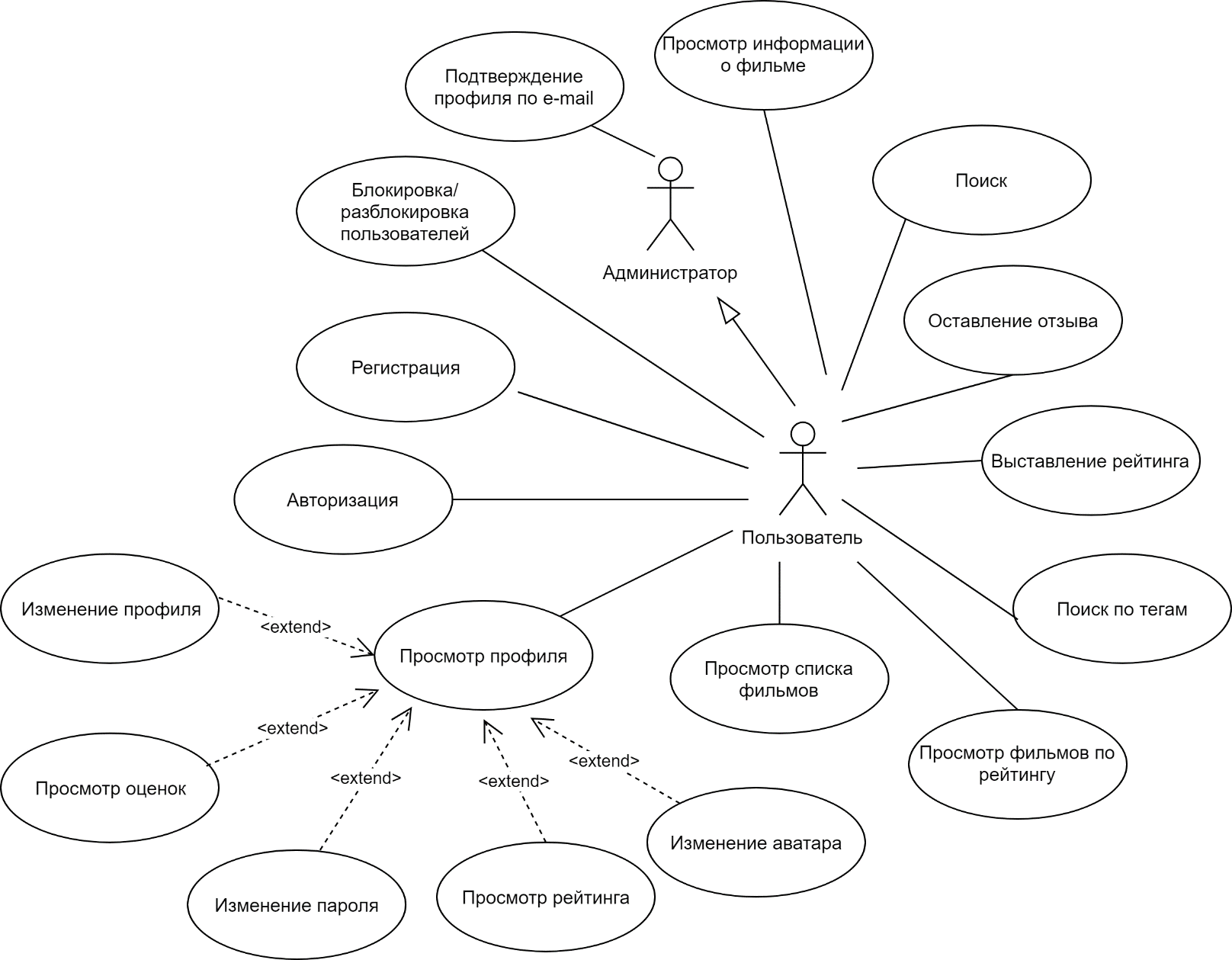


Рисунок 3.1 – Диаграмма вариантов использования

**4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЁ ОПИСАНИЕ**

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

Информационная модель 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 хранит в себе следующие атрибуты:

1. iduser – хранит уникальный идентификационный номер пользователя приложения;
2. login – хранит имя пользователя, которое указывается при входе в приложение;
3. password – хранит пароль, используемы для авторизации в системе;
4. email – хранит адрес почты пользователя;
5. real\_name – хранит настоящее имя пользователя, отображающееся на сайте;
6. date\_of\_registration – дата регистрации на сайте;
7. date\_of\_birth – дата рождения пользователя;
8. status – статус пользователя, например «заблокирован»;
9. avatar – хранит ссылку на аватар пользователя;
10. level\_points – количество очков уровня пользователя;
11. role – хранит роль пользователя. По умолчанию роль соответствует «пользователю».

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

1. idrating– хранит уникальный идентификационный номер записи;
2. iduser – хранит уникальный идентификационный номер пользователя, которому принадлежит оценка;
3. idfilm – хранит уникальный идентификационный номер фильма, которому предназначена оценка;
4. is\_seen – хранит информацию о том, просмотрен ли фильм;
5. rating\_amount – хранит количественное выражение оценки;

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

1. idfilm– хранит уникальный идентификационный номер записи;
2. name – хранит название фильма;
3. release\_year – хранит год выпуска фильма;
4. duration – хранит длительность в минутах;
5. poster – хранит ссылку на постер;
6. iddirector – хранит уникальный идентификационный номер режиссера данного фильма;
7. description – хранит описание фильма;

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

1. idreview– хранит уникальный идентификационный номер записи;
2. iduser - хранит уникальный идентификационный номер пользователя, которому принадлежит отзыв;
3. idfilm - хранит уникальный идентификационный номер фильма;
4. text – хранит текст отзыва;
5. date – хранит дату, когда был написан отзыв;

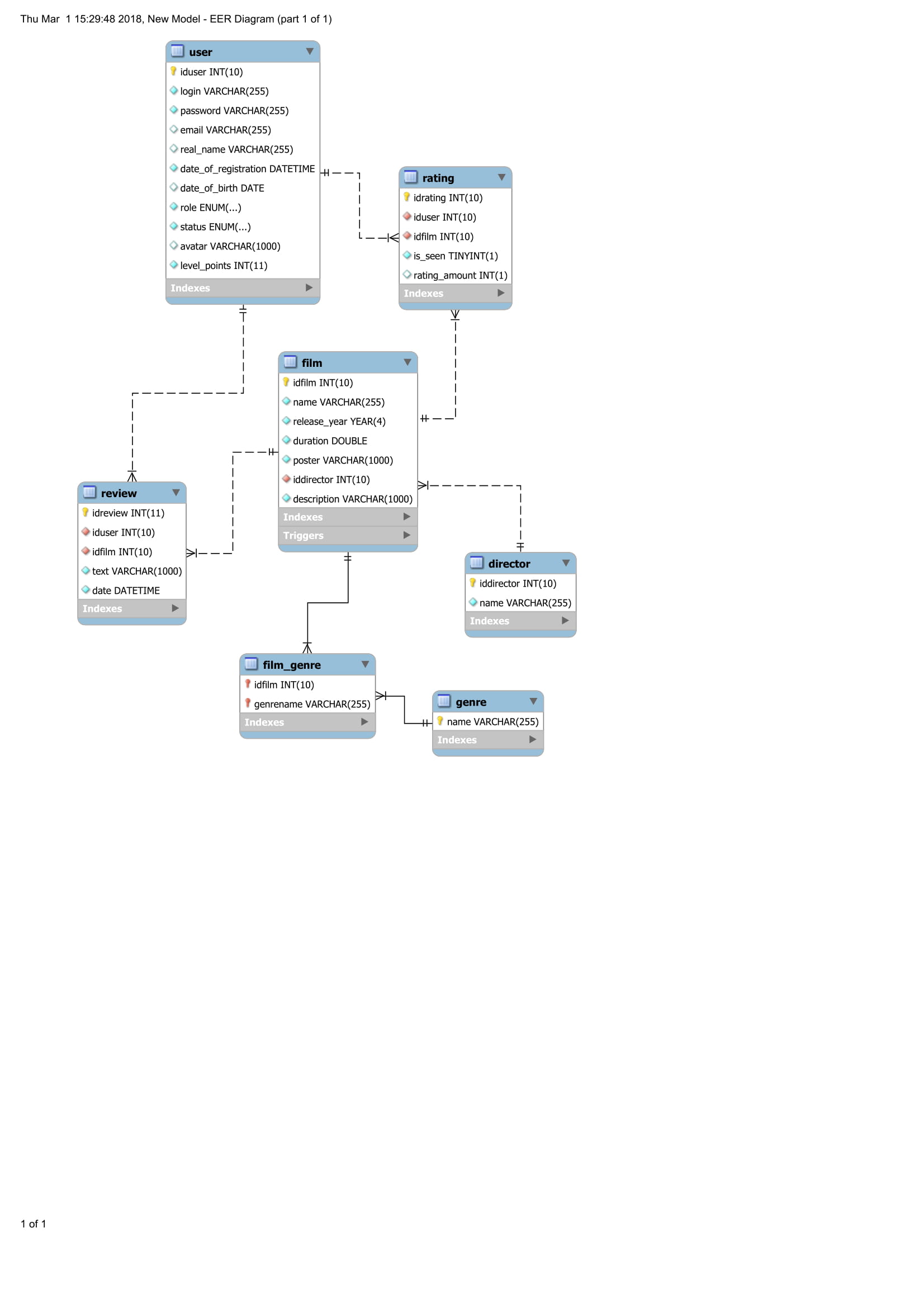


Рисунок 4.1 – Информационная модель системы

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

1. name– хранит название жанра;

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

1. iddirector– хранит уникальный идентификационный номер записи;
2. name – хранит имя режиссера;