

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Южно-Уральский государственный университет»  
(национальный исследовательский университет)  
Высшая школа экономики и управления  
Кафедра «Информационные технологии в экономике»

**РАБОТА ПРОВЕРЕНА**

Рецензент, доцент. каф. «Менеджмент» ЮУрГУ, к.т.н, доцент

\_\_\_\_\_/ А.И. Демченко /  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

**ДОПУСТИТЬ К ЗАЩИТЕ**

Заведующий кафедрой «ИТЭ»,  
д.т.н., с.н.с.

\_\_\_\_\_/ Б.М. Суховилов /  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

**РАЗРАБОТКА СОЦИАЛЬНОЙ ПЛАТФОРМЫ ДЛЯ ОРГАНИЗАЦИИ  
ОБРАТНОЙ СВЯЗИ И ОБМЕНА МНЕНИЯМИ О ВЫПУСКАЕМОЙ  
ПРОДУКЦИИ**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 09.04.03.2018.884 ВКР**

Руководитель, к.т.н., доцент

\_\_\_\_\_/ Е.М. Саргасов /  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Автор

студент группы ЭУ–221

\_\_\_\_\_/ Н.А. Микуров /  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Нормоконтролер, доцент

\_\_\_\_\_/ Е.Н. Конова /  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Челябинск 2018

## АННОТАЦИЯ

Микуров Н.А. «Разработка социальной платформы для организации обратной связи и обмена мнениями о выпускаемой продукции». – Челябинск: ЮУрГУ, ЭУ-221, 7 с., 30 ил., 49 табл., библиогр. список – 10 наим.

Целью дипломного проекта является изучение и разработка социальной платформы для организации обратной связи с пользователями для получения мнения о выпускаемой продукции.

Поставлены следующие задачи: анализ предметной области, разработка структуры приложения и схемы базы данных, реализация функционала социальной платформы, разработка соответствующего взаимодействия пользователя с системой для организации обратной связи о выпускаемой продукции.

В ходе выполнения работы были выявлены и проанализированы существующие проблемы и разработано решение. Выполнен анализ схожих решений, продумана логика приложения, разобран функционал, позволяющий конкретизировать и ранжировать оценки, оставлять отзывы и мнения пользователей, вести статистику, решить поставленную задачу.

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
1 ПОСТАНОВКА ЗАДАЧИ.....	6
1.1 Предметная область.....	6
1.2 Анализ существующих решений.....	6
1.3 Решение проблемы.....	11
1.4 Предлагаемый функционал.....	12
Вывод по разделу один.....	16
2 ПРОЕКТИРОВАНИЕ .....	17
2.1 Разработка инфологической модели.....	17
2.2 Выбор программных средств.....	20
2.3 Проектирование базы данных.....	22
2.4 Описание структуры данных .....	24
3.9 Описание структуры социальной платформы .....	39
Вывод по разделу два .....	41
3 РАЗРАБОТКА .....	42
Вывод по разделу три .....	68
ЗАКЛЮЧЕНИЕ .....	69
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	70

## ВВЕДЕНИЕ

На известном портале «KINODATA» был проведен опрос: «Какой Вы видите государственную поддержку кинематографа через Фонд кино?». Победил ответ: «Пересмотреть систему финансирования в сторону стимулирования не только кинопроизводства, но и системы кинопроката», – за него проголосовали 43% респондентов.

В обществе, функционирующем в жестких рыночных условиях, мнение конечного пользователя играет важную роль, но не всегда на это обращают должное внимание. На сегодняшний момент, не существует такого инструмента, который бы помог собрать оценки пользователей и позволил бы ранжировать оценки по необходимым характеристикам. Решения задачи с конкретизацией мнения пользователей до сих пор не существует в массовых условиях.

Необходим инструмент, который помог бы объединить мнения всех конечных пользователей в одном решении и замотивировать их на помощь в улучшении конечной продукции. Такое решение возможно организовать с помощью социальной платформы, которая поможет, как объединить всех пользователей в одном месте, так и заинтересовать их.

Актуальность данного проекта заключается в том, чтобы создать инструмент, позволяющий пользователям не только оставлять свои мнения о выпускаемой продукции, но и ранжировать по оценкам, с помощью которых, в дальнейшем, можно вести статистику. Так же участники данного проекта могут участвовать в рейтинге пользователей за звание лучшего киномана.

Цель дипломного проекта – разработка социальной платформы для эффективного способа накопления, обработки и целостного хранения мнений пользователей о выпускаемой продукции.

Из поставленной цели вытекают следующие задачи:

- рассмотреть предметную область;
- представить решение проблемы;

- проанализировать уже существующие решения;
- разработать логику будущего приложения;
- разработать базу данных для хранения и обработки информации;
- разработать свой функционал;
- разработать интерфейс;
- протестировать приложение.

Выпускная квалификационная работа представляет интерес для пользователей, которые хотят улучшить качество потребляемой продукции, а так же для инвесторов, желающие вложить денежные средства в проекты.

# 1 ПОСТАНОВКА ЗАДАЧИ

## 1.1 Предметная область

У отечественного кинематографа существует много проблем. Начиная от недоверия российских зрителей, заканчивая провалом кинокартины.

Главной проблемой является недоверие зрителя к отечественным фильмам. Данная проблема может возникнуть из-за того, что зрителю предлагают не то, что он хочет или ожидает увидеть. Может не нравиться предлагаемый жанр, задумка, актерский или режиссёрский состав. Так же впечатление может портиться от уже просмотренных отечественных фильмов, от непродуманности сюжета, от неясности или от незаконченности идей, от плохой актерской и режиссёрской работы, от чрезмерной загруженности фильма рекламой.

Посещаемость отечественных фильмов сильно снижается. На многих картинах в кинотеатрах сидит по несколько зрителей. И что самое тревожное, уже не имеет значения, хороший фильм или плохой. Отечественное кино абсолютно потеряло доверие. Оно дискредитировано в глазах зрителя. Существует ли тут проблема зрителя? Или виной всему то, что отечественное кино снимается не для зрителя?

## 1.2 Анализ существующих решений

Проанализирован функционал сайтов, на которых можно увидеть пользовательские мнения и оценки.

Сайты, дающие обратную связь:

- КиноПоиск;
- Отзовик;
- Megacritic.ru.

### **КиноПоиск**

Сайт, где пользователи могут просмотреть всю информацию о фильме и оставить рецензию. Главная страница представлена на рисунке 1.1.

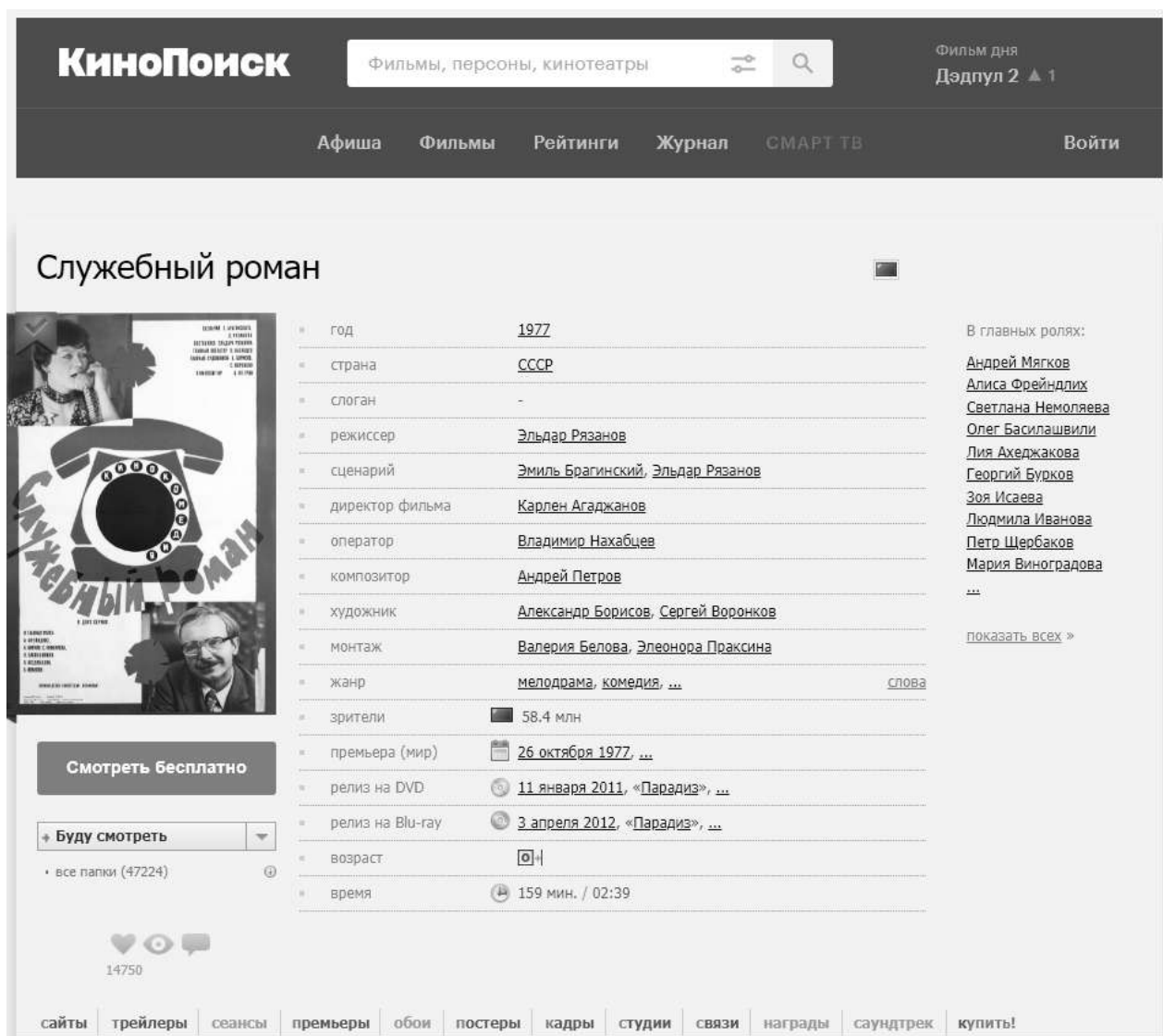


Рисунок 1.1 – Сайт «КиноПоиск»

На сайте есть подробная информация о фильме. Начиная от актеров и сценаристов, заканчивая информацией о сборах в кино. Так же можно посмотреть трейлер фильма, постеры, кадры из фильма и присужденные награды.

На сайте присутствует общий рейтинг, в котором участвует каждый фильм и рейтинг кинокритиков. Пользователь может оставить рецензию на фильм и посмотреть рецензии, написанные другими пользователями.

### Отзовик

Сайт, где пользователь может оставить своё мнение или отзыв о любом виде продукции. Главная страница представлена на рисунке 1.2.



Рисунок 2.2 – Сайт «Отзовник»

Сайт используется не только для оставления отзывов о фильмах, но и о любой другой продукции.

На сайте имеется краткое описание данного продукта и небольшая фотогалерея, где можно оставить фото данной продукции. Так же пользователь может написать отзыв и поставить свой рейтинг данной продукции.

### **Megacritic.ru**

Сайт, где пользователь может оставить зрительские комментарии и оценить фильм одной общей оценкой. Главная страница представлена на рисунке 1.3.



**MEGACRITIC.RU**  
мы работаем с критикой

Премьеры: Дзюба 2 | Колючая | За бортом | Чужеса | Такси 5 | Мстители: Война бесконечности  
Скоро в кино: Хан Соло: Зобатаве Войны, Истории | Миссия | Красный воробей | Суперсемейка 2  
Популярное: Мстители 3 | Собиор | Такси | Трезор | Рамбейдж | Гоголь. Вий | Тихоокеанский рубеж 2

### Время первых

Рейтинг зрителей

**8.2**

из 10

составлен из 108  
отзывов зрителей

составлен по 29  
рецензиям критиков

Рейтинг критиков

**7.6**

из 10

Новый отзыв
Отзывы зрителей
Добавить рецензию
Отзывы критиков



**ИНФОРМАЦИЯ О ФИЛЬМЕ**

Год: 2017

Жанр: Драмы • Исторические

Страна: Россия

Дата выхода в России: 06.04.2017 г.

Режиссер: Дмитрий Киселёв

Сценарий: Ирина Пивоварова • Сергей Калужанов • Юрий Коротков

Продюсер: Тимур Бекмамбетов • Сергей Агеев • Евгений Миронов

Композитор: Юрий Потенко

Киностудия: Базелевс Продакшн • Третий Рим • СBF

Возраст: 6+ лет.

Актёры фильма: Александра Урсуляк • Владимир Ильин • Евгений Миронов • Елена Панова • Константин Хабенский • Нелли Неведина • Артём Быстров • Александр Цема • Анатолий Котенёв • Марта Тимофеева

**Описание фильма Время первых (2017)**

В фильме **Время первых** действие развернётся в шестидесятые, в разгар Холодной Войны между СССР и

**НАВИГАТОР ПО ФИЛЬМАМ:**

Показать по:

дата выхода

Можете уточнить условия:

Все жанры

Все годы

Все страны

Показать

**НОВЫЕ 2017-2018:**

Все жанры	Русские
Комедии	Семейные
Мелодрамы	Триллеры
Мультни	Фантастика

**ФИЛЬМЫ ПО ЖАНРАМ:**

Детские	Сказки
---------	--------

**ЛУЧШИЕ 2017-2018:**

Все жанры	Мультни
Фильмы	Фильмы

Рисунок 3.3 – Сайт «Megacritic.ru»

Пользователь может увидеть информацию по фильму, описание, трейлер.

На сайте есть разделение на критиков и на обычных зрителей, поэтому присутствует два вида рейтинга, но каждый пользователь может поставить свою оценку фильму и написать комментарий.

Проанализировав ряд сайтов по сбору мнений пользователей, можно составить таблицу сравнительного анализа систем, которая приведена в таблице 1.

Таблица 1 – Сравнительный анализ

<p style="text-align: center;">Существующие решения</p> <p style="text-align: center;">Характеристики</p>	<p style="text-align: center;">Разрабатываемое приложение</p>	<p style="text-align: center;">КиноПоиск</p>	<p style="text-align: center;">Отзовик</p>	<p style="text-align: center;">Megacritic.ru</p>
Просмотр информации о фильмах	+	+	-	+
Оставить отзыв пользователя	+	+	+	+
Поставить рейтинг фильму	+	+	+	+
Оставить рецензию фильму	+	+	-	+
Рейтинг по интересующим характеристикам фильма	+	-	+/-	-
Регистрация пользователя	+	+	+	+
Страница пользователя	+	-	-	-
Личная информация о пользователе	+	-	-	-
Табло любимых фильмов	+	-	-	-
Новости и посты пользователя	+	-	-	-
Друзья пользователя	+	-	-	-
Рейтинг киномана	+	-	-	-
Анкеты о фильмах	+	-	-	-
Проверка знания пользователя о фильмах	+	-	-	-
Статистика	+	-	-	-

В результате сравнительного анализа выявлено:

- у всех присутствующих решений есть информация о фильмах;
- отсутствует полностью или частично реализовано бальное голосование по интересующим характеристикам, а не только общее впечатление о фильме;
- у пользователей есть только личный кабинет и, соответственно, пользователя нельзя идентифицировать по его предпочтениям и вкусам;
- отсутствует анкеты для сбора мнений пользователей;
- отсутствует награждение пользователя за активность на сайте;
- отсутствует сбор статистики по характеристикам мнений пользователей.

### **1.3 Решение проблемы**

Решение для сложившейся проблемы – это создать проект, основанный в первую очередь на уже имеющемся функционале, который будет привлекать пользователя и позволит собирать статистику, о мнениях пользователей.

Проект должен содержать в себе информацию о фильмах, отзывы, рецензии и оценки для конкретного фильма. Во вторую очередь информацию развлекательного и социального характера, которая привлечет пользователя в данный проект. Дополняет всё это функционал, который поможет вести статистику по фильмам, позволит идентифицировать пользователя по различным характеристикам, таким как: возраст, регион, любимые жанры, предпочтения в кино, посещаемые картины. Пользовательское мнение, оценки и рецензии характеризованы не одной общей оценкой о фильме, а по конкретным характеристикам. Продюсеры фильмов имеют возможность связаться с модератором и задать свои характеристики оценок, необходимые именно им.

Данный проект нацелен на получение обратной связи от пользователей о его предпочтениях, пожеланиях, оценках, которые помогут улучшить создаваемый продукт в соответствии с заданным уровнем качества, ожидаемым от пользователя.

Данный инструмент сможет дать статистику, об оценках пользователя по предложенным характеристикам и накопить данные для оценивания перспективности выпуска продукции. Такой инструмент позволит мотивировать пользователя делиться своим мнением и оценкой. Производители же продукции получают инструмент, с помощью которого смогут оценивать свою работу и привлечь пользователя к своей продукции.

Такой подход к решению проблемы позволяет собирать детальную статистику, то есть статистику по необходимым характеристикам. А самое главное, предоставит данные, для выявления закономерностей, которые объясняют, почему пользователям понравилась или не понравилась конкретная кинокартина.

Данное решение нацелено на решение следующих задач.

- Возможность изучения статистики, созданной пользователями для принятия решения о финансировании проектов.
- Возможность увидеть все недочеты кинокартины.
- Возможность оценить или выбрать актера, или режиссёра, по мнению пользователей, на данную роль или фильм.
- Возможность пользователям соревноваться за звание самого лучшего киномана.
- Возможность пользователям давать свои рецензии, оценки, отзывы фильмам.
- Возможность пользователям создавать и делиться своими предпочтениями в мире фильмов.

#### **1.4 Предлагаемый функционал**

Сайт содержит страницы фильмов, которые будут создавать модераторы или лица, доверенные продюсерами фильмов и модератором.

На странице фильма представлен постер фильма, информация о фильме, так же будут отображаться все рейтинги, используемые на сайте. На странице сайта можно оставить рецензию. Пользователь сможет оценить фильм, проголосовать за него, увидеть все специальные тесты по этому фильму. Пользователи

смогут вести беседу по фильму, оставляя комментарий на странице фильма. Разработанный интерфейс страницы фильма представлен на рисунке 1.4.

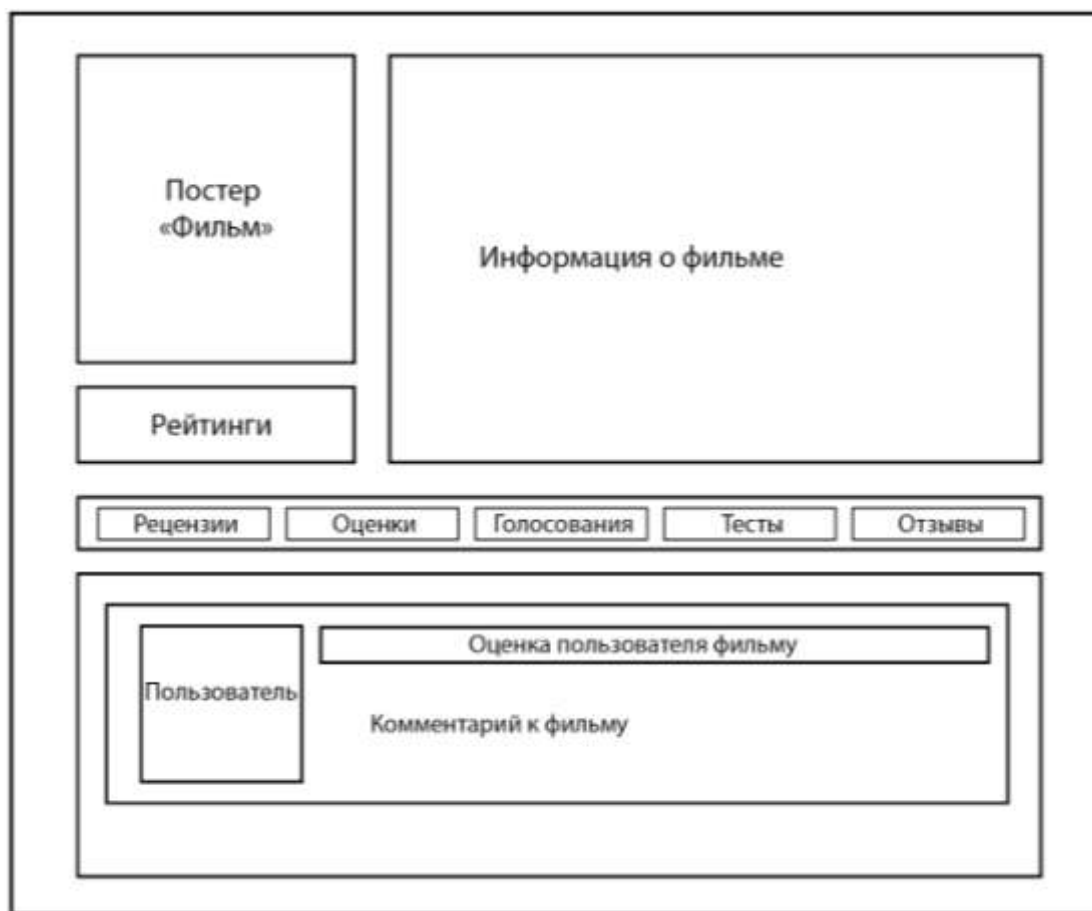


Рисунок 4.4 – Интерфейс страницы фильма

Разработанный сайт содержит страницы пользователей.

У каждого пользователя имеется своя личная страница, где он сможет делиться своими мыслями, идеями, интересами, любимой цитатой и любимыми фильмами. Пользователь сможет создавать свой «Топ» понравившихся ему фильмов, делиться любимыми моментами, на стене. Разработанный интерфейс страницы пользователя представлена на рисунке 1.5.

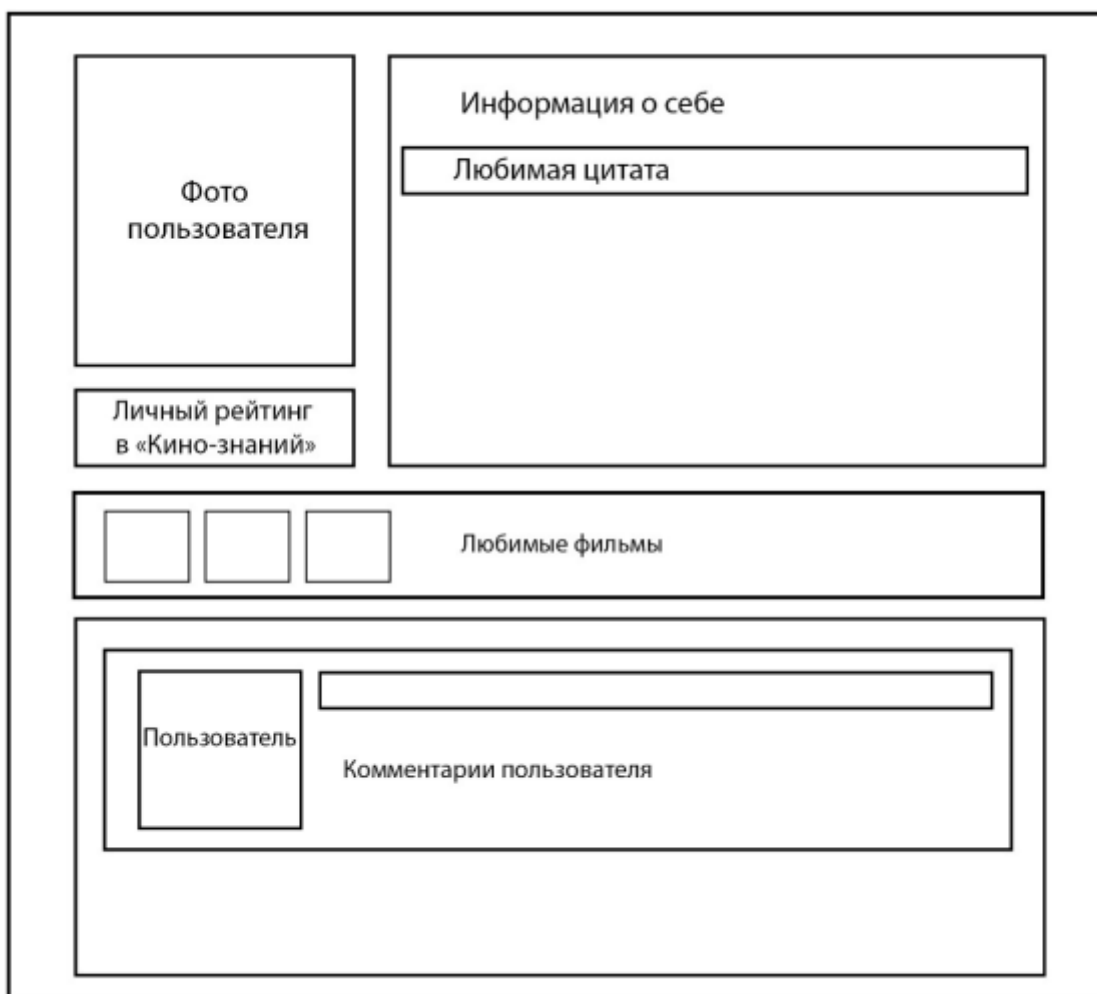


Рисунок 5.5 – Интерфейс страницы пользователя

Разработанный сайт содержит тесты по фильмам за звание лучшего кино-мана. Разработанный интерфейс страницы испытания представлен на рисунке 1.6.

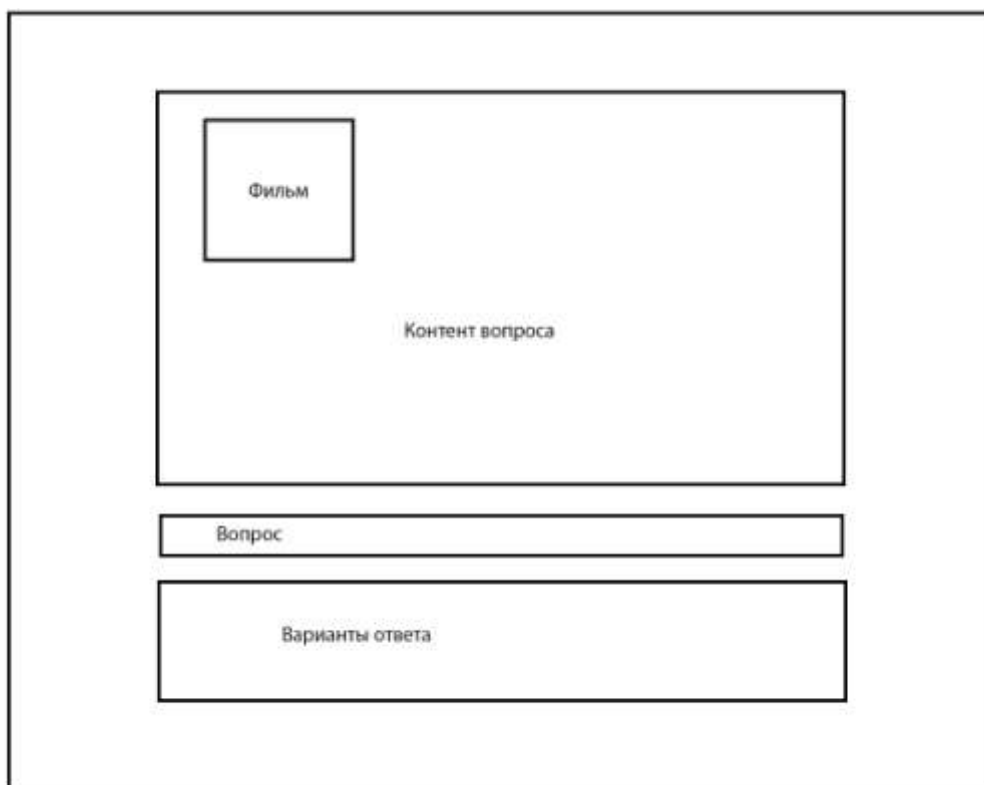


Рисунок 6.6 – Предполагаемая страница испытания

Каждый пользователь сможет участвовать в этих испытаниях, зарабатывая при этом баллы киномана.

Для того чтобы оценить фильм, на сайте разработана страница оценки фильма. На такой странице пользователь сможет оценить фильм по таким критериям как общая оценка и по индивидуальным интересующим характеристикам, которые назначит модератор. Разработанный интерфейс оценки фильма представлен на рисунке 1.7.

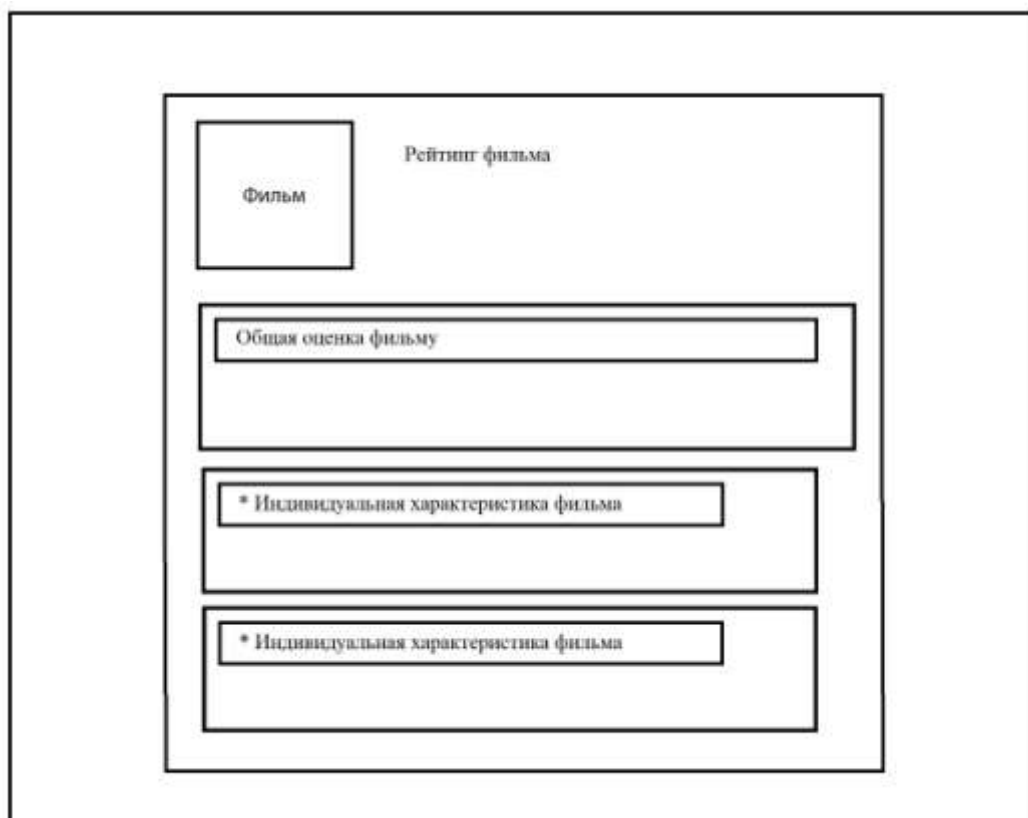


Рисунок 7.7 – Интерфейс страницы оценки фильма

Данная система оценки фильмов позволит модераторам или продюсерам фильмов собирать статистику мнений пользователей по интересующим характеристикам.

### **Вывод по разделу один**

Выполнена анализ и постановка проблемы. Определены задачи в рамках достижения поставленной цели. Проанализированы существующие системы, выявлены их недостатки. Разработан интерфейс функционал социальной платформы.



## **2 ПРОЕКТИРОВАНИЕ**

### **2.1 Разработка инфологической модели**

Основным этапом разработки является моделирование, необходимое для понимания структуры системы в целом. Одних лишь моделей бывает недостаточно. Для того чтобы понять любую систему, необходимо разрабатывать техническую записку, которая детально описывает взаимосвязь моделей и самой системы.

Для разработки таких моделей, применимых к программным системам, необходим язык, с помощью которого можно описать архитектуру самой системы и взаимодействие системы с пользователями с различных точек зрения. Используются правила, слова и модели такого языка, объясняющие, как создавать и понимать модели, но ничего не рассказывается о том, какие модели и в каких случаях нужно создавать. Главная задача разработки программного обеспечения – хорошо организованный процесс, помогающий программисту понять и описать, что требуется сделать, какие данные и ресурсы необходимы для их создания, какие логические структуры будут приложены, какой функционал необходим, для выполнения поставленных задач.

Для решения задачи моделирования полностью подходит язык UML. Данный язык моделирования позволяет графически описать объекты, структуры, классы моделирования разрабатываемого программного продукта, а также моделирования бизнес-процессов, логики выполнения частей программы и понимания общей структуры, организованной в программе.

Диаграмма прецедентов используется для моделирования вида системы и показывает отношения между субъектами. На рисунке 2.1 приведена диаграмма прецедентов, задающая представление будущей концепции создаваемого приложения.

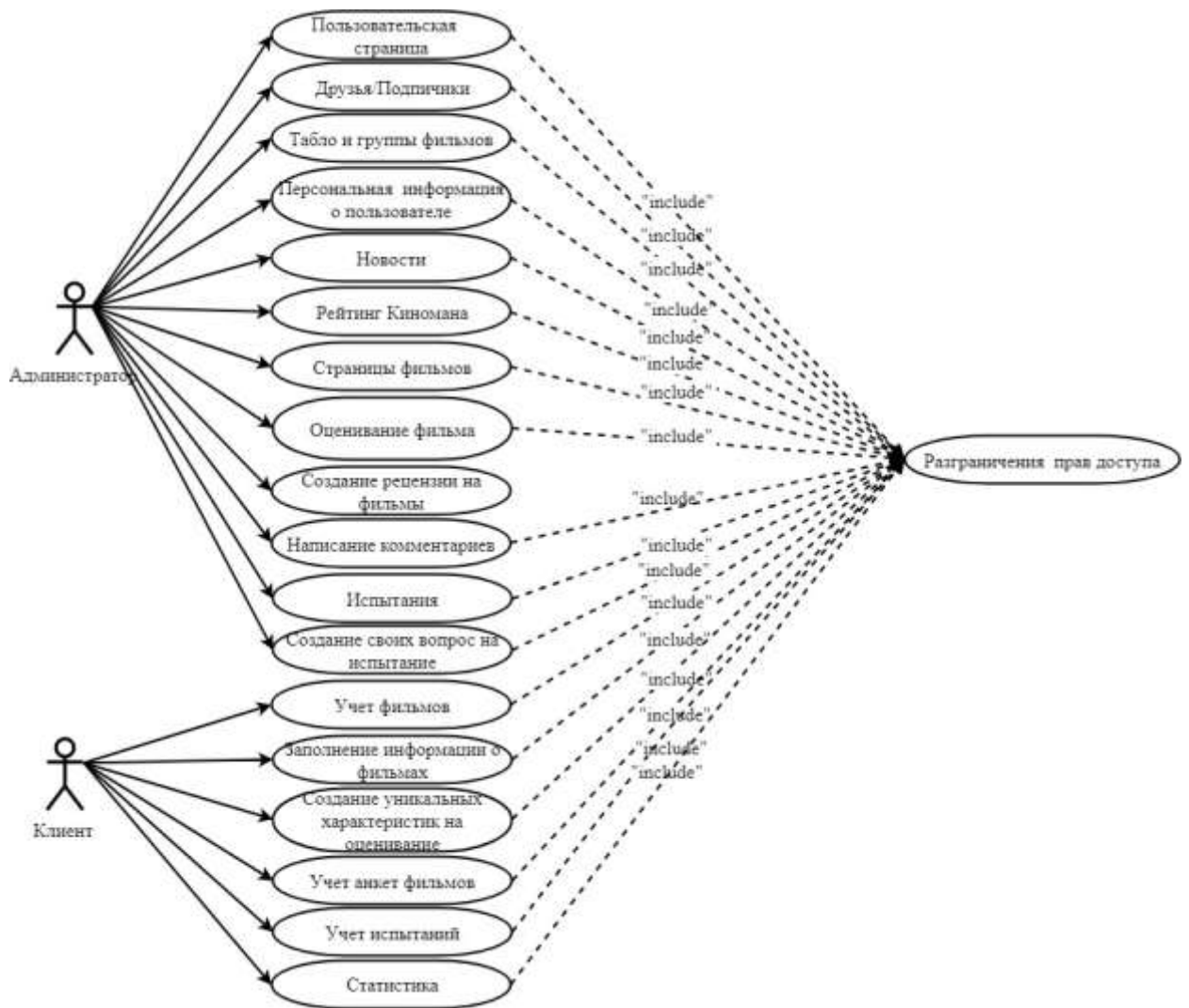


Рисунок 2.1 – Диаграмма прецедентов

На диаграмме представлена разновидность ролей пользователей системы: таких как «Пользователь» и «Модератор», и их взаимодействие.

Диаграмма деятельности, показывающая алгоритм прохождения испытания, приведена на рисунке 2.2.



Рисунок 2.2 – Диаграмма деятельности алгоритма прохождения испытания

Диаграмма показывает алгоритм прохождения испытания.

На рисунке 2.3 приведена диаграмма, показывающая последовательность действий при добавлении в друзья другого пользователя системы.

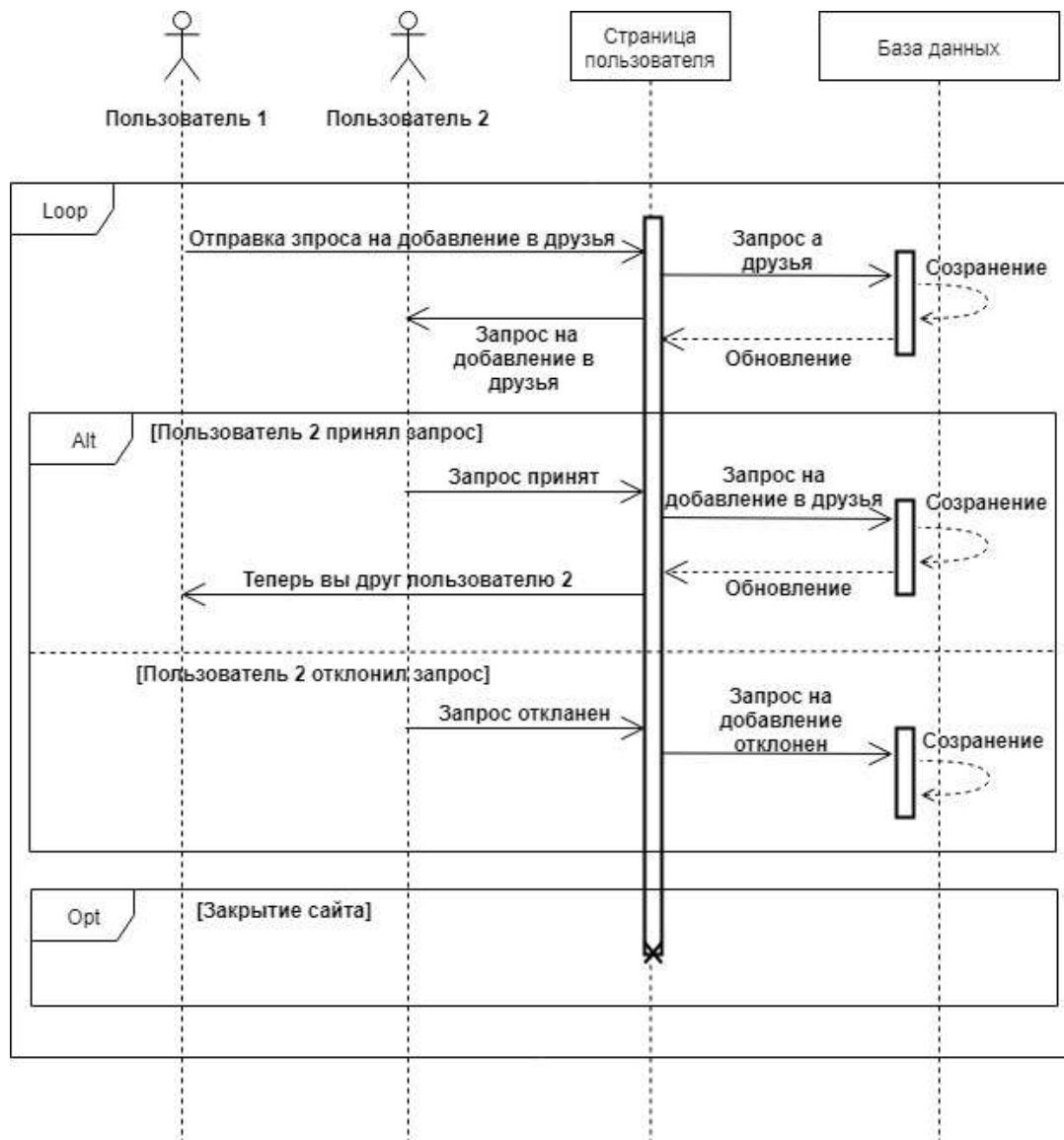


Рисунок 2.3 – Диаграмма последовательности добавления в друзья

Диаграмма показывает процесс того, как один пользователь добавляет в друзья другого пользователя системы.

## 2.2 Выбор программных средств

Для обеспечения безопасности данных в Web-приложении требуется разделение прав доступа между клиентами и сотрудниками.

Разрабатываемое Web-приложение должно осуществлять целостность базы данных, значит, вести контроль вводимой пользователем информации. Помимо

этого, приложение должно предоставлять не только удобный просмотр запрашиваемой информации, но и удобное добавление информации.

В основе Web-приложения должна быть клиент-серверная технология. База данных хранится на сервере, а клиентом выступает браузер. Обмен информацией происходит по сети. Преимущество клиент-серверной технологии заключается в том, что пользователи не зависят от установленной операционной системы.

Основной язык, используемый в Web-приложении, объектно-ориентированный язык программирования C#. Используется платформой Microsoft .NET Framework. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [3].

В Web-приложениях применяется ASP.NET технология. ASP.NET (Active Server Pages для .NET) – технология создания Web-приложений и Web-сервисов от компании Microsoft. Она является составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP [3].

Для решения поставленных задач выбрана реляционная база данных. Реляционная база данных представляет собой множество взаимосвязанных таблиц. Каждая таблица содержит информацию об объектах определенного вида. Каждая строка таблицы содержит данные об одном объекте, а столбцы таблицы содержат различные характеристики этих объектов – атрибуты. Строки таблицы называются записями. Все записи таблицы имеют одинаковую структуру – они состоят из полей, в которых хранятся атрибуты объекта. Каждое поле записи содержит одну характеристику объекта и представляет собой заданный тип данных. Для идентификации записей используется первичный ключ. Первичным ключом называется набор полей таблицы, комбинация значений которых однозначно определяет каждую запись в таблице [4].

СУБД отвечает за поддержку отношений между данными, за согласованность данных и отказоустойчивость.

В качестве СУБД выбрана база данных Microsoft SQL Server.

Microsoft SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft. Основным используемым языком запросов – Transact SQL. Microsoft SQL Server эффективно использует многопроцессорные системы благодаря поддержке симметричной архитектуры сервера и многопроцессорной обработки на уровне потоков [5].

На клиентской части применяются стандартные технологии: HTML, CSS, JavaScript.

HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине. Большинство Web-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML представляет собой текстовый файл, содержащий команды, называемые тегами. Просмотр HTML-документов осуществляется с помощью программ-браузеров и отображается на экране монитора, планшета или мобильного устройства [6].

CSS (англ. Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида Web-страниц, написанных с помощью языков разметки [7].

JavaScript – прототип, но – ориентированный сценарный язык программирования. Является реализацией языка ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах, как язык сценариев для придания интерактивности Web-страницам [8].

### **2.3 Проектирование базы данных**

База данных – это набор данных, хранимых в соответствии со схемой. Манипулирование самими данными выполняют в соответствии с правилами, заданными на этапе моделирования.

Схема базы данных предназначена для описания структуры и организации целостности данных, используемых при создании и поддержке базы данных.

Реляционная база данных представляет собой организованный набор данных, представленных в виде совокупного набора взаимосвязанных объектов, каждый из которых имеет определенный тип, согласно требованиям нормализации базы данных. Каждый объект, то есть таблица, должен иметь уникальный идентификатор, то есть ключ, означающий однозначно определенную запись в таблице.

Нормализация базы данных необходима для организации реляционных баз данных. Она разрешает конфликт дублирования данных, позволяет обеспечить целостность и упростить разработку базы данных.

Спроектированная база данных, позволяет решить поставленные цели и задачи, основана на имеющейся информации о фильмах и дополнена данными для реализации социальной платформы. Нормализованная база данных состоит из 33 взаимосвязанных таблиц. Схема данных приведена на рисунке 2.6.

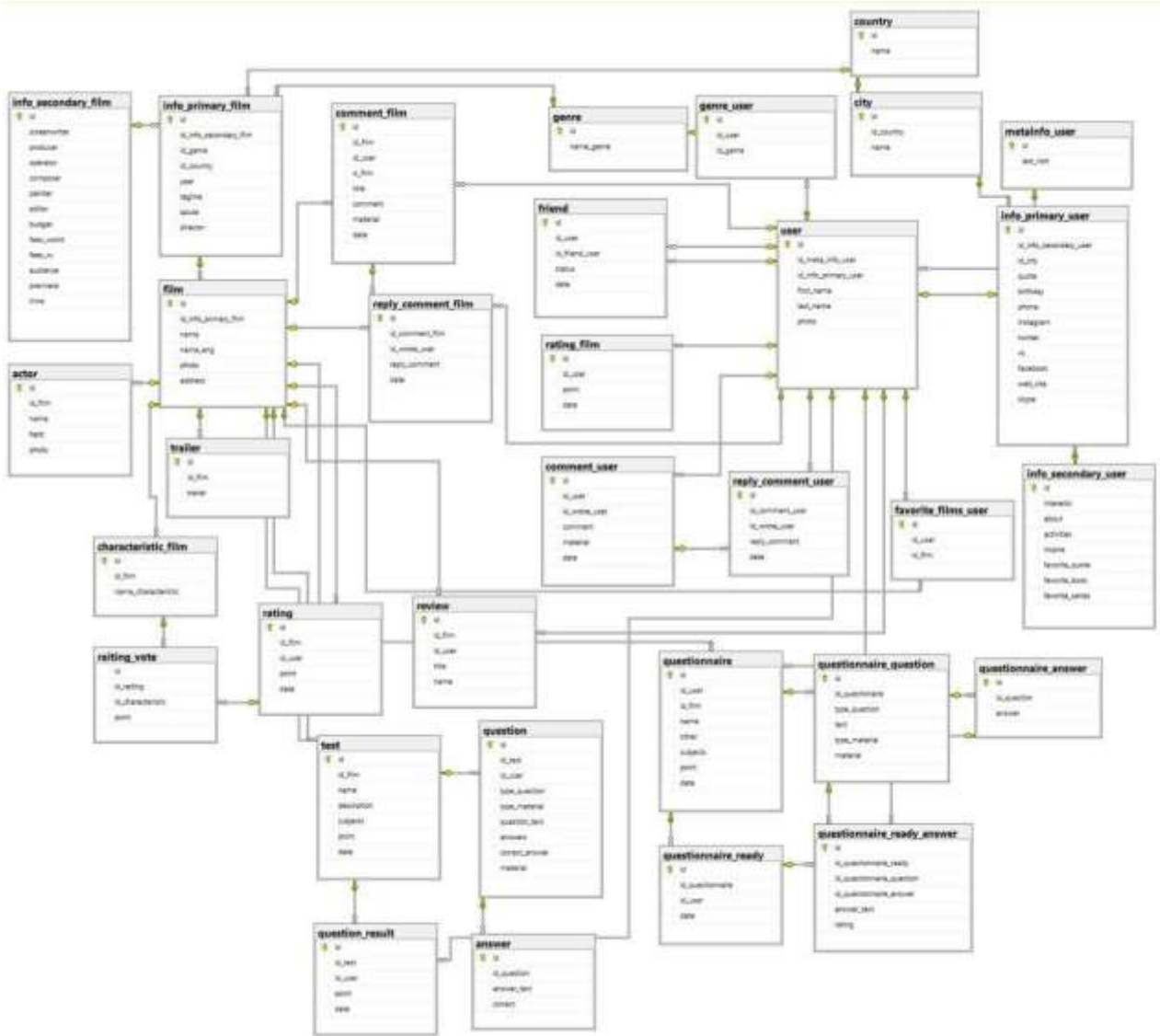


Рисунок 2.6 – Схема базы данных

## 2.4 Описание структуры данных

Рассмотрим структуру и тип хранящихся данных каждой таблицы, созданной для решения поставленной задачи.

База данных, созданная для социальной платформы, содержит следующие таблицы, приведенные в таблице 2.1.



Таблица 2.1 – Описание таблиц базы данных

№	Название таблицы	Назначение
1.	User	Содержит информацию о пользователях сайта
2.	Info_primary_user	Содержит первичные данные о пользователе
3.	Info_secondary_user	Содержит информацию о предпочтениях пользователя
4.	Country	Страны
5.	City	Города
6.	Comment_user	Комментарии на стене пользователя
7.	Reply_comment_user	Ответ на комментарии на стене пользователя
8.	Friend	Друзья пользователя
9.	Genre	Жанры кино
10.	Genre_user	Любимые жанры пользователя
11.	Rating_film	Рейтинг киномана
12.	Film	Страница фильма
13.	Info_primary_film	Содержит первичную информацию о фильме
14.	Info_secondary_film	Содержит информацию полную информацию о фильме
15.	Test	Испытания
16.	Question	Вопросы испытания
17.	Answer	Ответы испытания
18.	Favorite_films_user	Фильмы на табло пользователя
19.	Questionnaire	Опрос
20.	Questionnaire_question	Вопросы опроса
21.	Questionnaire_answer	Ответы опроса
22.	Questionnaire_ready	Прохождение опроса пользователем
23.	Questionnaire_ready_answer	Данных ответах пользователем
24.	Comment_film	Мнения о фильме
25.	Reply_comment_film	Ответ на мнения на стене фильма
26.	Meta_info_user	Системная информация о пользователе
27.	Question_result	Прохождение испытания
28.	Review	Рецензия
29.	Actor	Актеры фильма
30.	Rating	Рейтинг фильма
31.	Rating_vote	Оценки пользователя
32.	Characteristic_film	Характеристики голосования
33.	Trailer	Трейлер фильма

Таблица «User» необходима для идентификации пользователя. Содержит в себе первичную информацию о пользователе, его имя, фамилию и фотографию. Более подробное содержание полей приведено в таблице 2.2.

Таблица 2.2 – Описание «User» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_user	Целый	Идентификатор аутентификации
3.	id_meta_info_user	Целый	Идентификатор информации о пользователе
4.	id_info_primary_user	Целый	Идентификатор первичной информации
5.	first_name	Целый	Имя пользователя
6.	last_name	Текстовый	Фамилия пользователя
7.	photo	Текстовый	Ссылка на фото

Таблица «Info\_secondary\_user» содержит в себе информацию о пользовательских интересах, его мысли, виды деятельности, интересы, любимые цитаты, книги и сериалы. Более подробно содержание полей приведено в таблице 2.3.

Таблица 2.3 – Описание «Info\_secondary\_user» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	about	Целый	О пользователе
3.	activities	Целый	Виды деятельности
4.	inspire	Текстовый	Интересы пользователя
5.	favorite_quote	Текстовый	Любимые цитаты
6.	favorite_book	Текстовый	Любимые книги
7.	favorite_series	Текстовый	Любимые сериалы

Таблица «Info\_primary\_user» содержит в себе основную информацию о пользователе, его место проживания, цитату, дату рождения, номер телефона и его социальные страницы. Более подробно содержание полей приведено в таблице 2.4.

Таблица 2.4 – Описание таблицы базы данных «Info\_primary\_user»

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_info_secondary_user	Целый	Идентификатор вторичной информации о пользователе
3.	id_city	Целый	Идентификатор города
4.	quote	Текстовый	Цитата пользователя
5.	birthday	Дата	День рождения
6.	genres	Текстовый	Любимые жанры
7.	phone	Текстовый	Телефон
8.	instagram	Текстовый	Инстаграм
9.	twitter	Текстовый	Твиттер
10.	vk	Текстовый	Вконтакте
11.	facebook	Текстовый	Фейсбук
12.	web_site	Текстовый	Свой сайт

Таблица «Country» содержит в себе информацию о странах. Более подробно содержание полей приведено в таблице 2.5.

Таблица 2.5 – Описание «Country» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	name	Текстовый	Название страны

Таблица «Comment\_user» содержит в себе информацию о комментариях на странице пользователей, на какой странице был оставлен комментарий, каким

пользователем, его текст и дату публикации. Более подробно содержание полей приведено в таблице 2.6.

Таблица 2.6 – Описание «Comment\_user» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_user	Целый	Идентификатор страницы пользователя
3.	id_wrote_user	Целый	Идентификатор пользователя, оставившего комментарии
4.	comment	Текстовый	Комментарии
5.	date	Дата	Дата создания

Таблица «Comment\_film» содержит в себе информацию о мнениях о фильме, о каком фильме был оставлен комментарий, каким пользователем, его текст и дату публикации. Более подробно содержание полей приведено в таблице 2.7.

Таблица 2.7 – Описание «Comment\_film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	id_user	Целый	Идентификатор пользователя
4.	is_film	Логический	Оставил фильм
5.	title	Текстовый	Заголовок
6.	comment	Текстовый	Комментарии
7.	material	Текстовый	Материал
8.	date	Дата	Дата создания

Таблица «Reply\_comment\_user» содержит в себе информацию о ответах на комментарии, оставленные на странице пользователей, на какой комментарий был оставлен ответ, каким пользователем, его ответ и дату публикации. Более подробно содержание полей приведено в таблице 2.8.

Таблица 2.8 – Описание «Reply\_comment\_user» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_comment_user	Целый	Идентификатор комментария на стене
3.	id_wrote_user	Целый	Идентификатор пользователя, оставившего комментарии
4.	reply_comment	Текстовый	Ответ
5.	date	Дата	Дата создания

Таблица «Friend» содержит в себе информацию о друзьях пользователя, идентификатор друга и дату добавления. Более подробно содержание полей, приведено в таблице 2.9.

Таблица 2.9 – Описание «Friend» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_user	Целый	Идентификатор пользователя
3.	id_friend_user	Целый	Идентификатор пользователя друга
4.	status	Целый	Статус пользователя
5.	visible	Логический	Скрытый друг
6.	date	Дата	Дата добавления

Таблица «Genre» содержит в себе информацию о жанрах фильмов. Более подробно содержание полей приведено в таблице 2.10.

Таблица 2.10 – Описание «Genre» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	name_genre	Текст	Название жанра

Таблица «Rating\_film» содержит в себе информацию о рейтинге киномана, идентификатор пользователя, количество очков и дату. Более подробно содержание полей приведено в таблице 2.11.

Таблица 2.11 – Описание «Rating\_film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_user	Целый	Идентификатор пользователя
3.	point	Целый	Количество очков
4.	date	Дата	Операторы

Таблица «Genre\_user» содержит в себе информацию о любимых жанрах пользователя. Более подробно содержание полей приведено в таблице 2.12.

Таблица 2.12 – Описание «Genre\_user» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное,
2.	id_user	Целый	Идентификатор пользователя
3.	id_genre	Целый	Идентификатор жанра

Таблица «City» содержит в себе информацию о городах. Более подробно содержание полей приведено в таблице 2.13.

Таблица 2.13 – Описание «City» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_country	Целый	Идентификатор страны
3.	name	Текстовый	Название страны

Таблица «Film» содержит первичную информацию о фильме, название фильма и постер. Более подробно содержание полей приведено в таблице 2.14.

Таблица 2.14 – Описание «Film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_info_primary_film	Целый	Идентификатор первичной информации о фильме
3.	name	Текстовый	Название фильма
4.	name_eng	Текстовый	Название фильма на английском
5.	photo	Текстовый	Ссылка на фото

Таблица «Answer» содержит в себе информацию об ответах на вопросы испытания. Более подробно содержание полей приведено в таблице 2.15.

Таблица 2.15 – Описание «Answer» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_test	Целый	Идентификатор вопроса
3.	answer_text	Целый	Ответ

Таблица «Rating\_vote» содержит в себе информацию о том, как пользователь проголосовал за характеристики фильма. Более подробно содержание полей приведено в таблице 2.16.

Таблица 2.16 – Описание «Rating\_vote» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_rating	Целый	Идентификатор общей оценки пользователя
3.	id_characteristic	Целый	Идентификатор характеристики фильма
4.	point	Целый	Количество очков

Таблица «Info\_primary\_film» содержит информацию о фильме, название, стране производства, годе выпуска, продюсере, цитату и слоган. Более подробно содержание полей приведено в таблице 2.17.

Таблица 2.17 – Описание «Info\_primary\_film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_info_secondary_film	Целый	Идентификатор вторичной информации о фильме
3.	id_genre	Целый	Идентификатор жанра
4.	id_country	Целый	Страна производитель
5.	year	Дата	Год выхода фильма
6.	tagline	Текстовый	Слоган фильма
7.	qoute	Текстовый	Знаменитая цитата фильма
8.	director	Текстовый	Продюсер

Таблица «Questionnaire» содержит в себе информацию существующих опросах в системе, к какому фильму принадлежит опрос, название опроса, тематика, количество очков за прохождение и дата создания опроса. Более подробно содержание полей приведено в таблице 2.18.

Таблица 2.18 – Описание «Questionnaire» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	id_user	Целый	Идентификатор пользователя
4.	name	Текстовый	Название опроса
5.	other	Текстовый	Информация о опросе
6.	subjects	Текстовый	Очки киномана
7.	point	Целый	Правильный ответ
8.	date	Дата	Дата



Таблица «Info\_secondary\_film» содержит вторичную информацию о фильме, сценаристах, операторах, композиторах, художниках, бюджете фильма, дате премьеры и время фильма. Более подробно содержание полей приведено в таблице 2.19.

Таблица 2.19 – Описание «Info\_secondary\_film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	screenwriter	Текстовый	Сценаристы
3.	producer	Текстовый	Продюсеры
4.	operator	Текстовый	Операторы
5.	composer	Текстовый	Композиторы
6.	painter	Текстовый	Художники
7.	editor	Текстовый	Монтажеры
8.	budget	Текстовый	Бюджет
9.	fees_world	Текстовый	Премьера мир
10.	fees_ru	Текстовый	Премьера Россия
11.	audience	Текстовый	Аудитория
12.	premiere	Текстовый	Премии
13.	time	Целый	Время фильма

Таблица «Trailer» содержит в себе информацию о трейлерах фильма. Более подробно содержание полей приведено в таблице 2.20.

Таблица 2.20 – Описание «Trailer» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	trailer	Текст	Трейлер

Таблица «Question» содержит в себе информацию о вопросах испытания, тип вопроса, тип материала вопроса, текст вопроса, ответ и сам материал, если он

есть, название испытания, описание и количество очков за прохождение. Более подробно содержание полей приведено в таблице 2.21.

Таблица 2.21 – Описание «Question» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_test	Целый	Идентификатор теста
3.	id_user	Текстовый	Идентификатор пользователя
4.	type_question	Текстовый	Тип вопроса
5.	type_material	Текстовый	Тип материала
6.	question_text	Целый	Вопрос испытания
7.	correct_answer	Целый	Правильный ответ
8.	material	Текстовый	Ссылка на материал

Таблица «Test» содержит в себе информацию о испытаниях в системе, к какому фильму относится испытание, название испытания, описание и количество очков за прохождение. Более подробно содержание полей приведено в таблице 2.22.

Таблица 2.22 – Описание «Test» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	name	Текстовый	Название испытания
4.	description	Текстовый	Описание
5.	subjects	Текстовый	Тематика
6.	point	Целый	Очки киномана
7.	date	Дата	Дата создания

Таблица «Questionnaire\_question» содержит в себе информацию о вопросах опроса, к какому опросу принадлежит вопрос, тип опроса, текст вопроса, тип ма-

териала и сам материал. Более подробно содержание полей приведено в таблице 2.23.

Таблица 2.23 – Описание «Questionnaire\_question» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_questionnaire	Целый	Идентификатор опроса
3.	type_question	Целый	Тип опроса
4.	text	Текстовый	Текст вопроса
5.	type_material	Текстовый	Тип материала
6.	material	Текстовый	Ссылка на материал

Таблица «Favorite\_films\_user» содержит в себе информацию о табло, представленных на стене пользователя. Более подробно содержание полей приведено в таблице 2.24.

Таблица 2.24 – Описание «Favorite\_films\_user» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_user	Целый	Идентификатор пользователя
3.	id_film	Целый	Идентификатор фильма

Таблица «Questionnaire\_answer» содержит в себе информацию о ответах на вопросы опроса, к какому вопросу принадлежит ответ и сам ответ. Более подробно содержание полей приведено в таблице 2.25.

Таблица 2.25 – Описание «Questionnaire\_answer» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_question	Целый	Идентификатор вопроса, опроса
3.	answer	Текстовый	Ответы

Таблица «Questionnaire\_ready» содержит в себе информацию о прохождении опроса пользователем. Более подробно содержание полей приведено в таблице 2.26.

Таблица 2.26 – Описание «Questionnaire\_ready» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_questionnaire	Целый	Идентификатор опроса
3.	id_user	Целый	Идентификатор пользователя
4.	date	Дата	Дата прохождения

Таблица «Review» содержит в себе информацию о рецензиях, о каком фильме была оставлена рецензия, каким пользователем и сама рецензия. Более подробно содержание полей приведено в таблице 2.27.

Таблица 2.27 – Описание «Review» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	id_user	Целый	Идентификатор пользователя
4.	title	Текст	Название рецензии
5.	name	Текст	Рецензия

Таблица «Meta\_info\_user» содержит в себе информацию о системных полях. Более подробно содержание полей приведено в таблице 2.28.

Таблица 2.28 – Описание «Reply\_comment\_film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	last_visit	Дата, время	Последний вход пользователя в систему

Таблица «Questionnaire\_ready\_answer» содержит в себе информацию о данных ответов, пройденных пользователем опросов. Более подробно содержание полей приведено в таблице 2.29.

Таблица 2.29 – Описание «Questionnaire\_ready\_answer» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_questionnaire_ready	Целый	Идентификатор прохождения опроса
3.	id_questionnaire_question	Целый	Идентификатор вопроса
4.	id_questionnaire_answer	Целый	Идентификатор ответа
5.	answer_text	Текстовый	Ответ данный пользователем
6.	rating	Целый	Рейтинг

Таблица «Actor» содержит в себе информацию об актерах, в каком фильме снимался актер, какой у него был герой. Более подробно содержание полей приведено в таблице 2.30.

Таблица 2.30 – Описание «Actor» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	name	Текст	Имя актера
4.	hero	Текст	Имя героя фильма
5.	photo	Текст	Фотография

Таблица «Reply\_comment\_film» содержит в себе информацию о ответах на комментарии, оставленные на мнения о фильмах, на какое мнение был оставлен ответ, каким пользователем, текст ответа и дату публикации. Более подробно содержание полей приведено в таблице 2.31.

Таблица 2.31 – Описание «Reply\_comment\_film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_comment_film	Целый	Идентификатор страницы фильма
3.	id_wrote_user	Целый	Идентификатор пользователя
4.	reply_comment	Текстовый	Ответ
5.	date	Дата	Дата создания

Таблица «Characteristic\_film» содержит в себе информацию о характеристиках конкретного фильма. Более подробно содержание полей приведено в таблице 2.32.

Таблица 2.32 – Описание «Characteristic\_film» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	name_characteristic	Текст	Характеристика

Таблица «Rating» содержит в себе информацию о рейтинге фильма, какой пользователь оценил конкретный фильм и сколько он поставил баллов. Более подробно содержание полей приведено в таблице 2.33.

Таблица 2.33 – Описание «Rating» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_film	Целый	Идентификатор фильма
3.	id_user	Целый	Идентификатор пользователя
4.	point	Целый	Количество очков
5.	date	Дата	Дата

Таблица «Question\_result» содержит в себе информацию о прохождении пользователем испытания, какой тест прошел пользователь и сколько баллов он заработал. Более подробно содержание полей приведено в таблице 2.34.

Таблица 2.34 – Описание «Question\_result» таблицы базы данных

№	Название поля	Тип поля	Назначение
1.	id	Целый	Идентификатор таблицы, поле ключевое, автоинкрементное
2.	id_test	Целый	Идентификатор теста
3.	id_user	Целый	Идентификатор пользователя
4.	point	Целый	Ответ
5.	date	Дата	Дата создания

### 3.9 Описание структуры социальной платформы

Структура сайта представляет собой иерархию ссылок, а также отображение взаимосвязей с пользователем сайта, начиная с главной страницы.

Социальной платформой пользуется как обычные пользователи, так и модераторы разработанной социальной платформы. Для того чтобы подтвердить свой статус в системе необходимо выполнить аутентификацию на сайте, чтобы отобразить модераторское или пользовательское меню.

На рисунке 2.7 приведена структура социальной платформы для организации обратной связи и обмена мнениями о выпускаемой продукции.

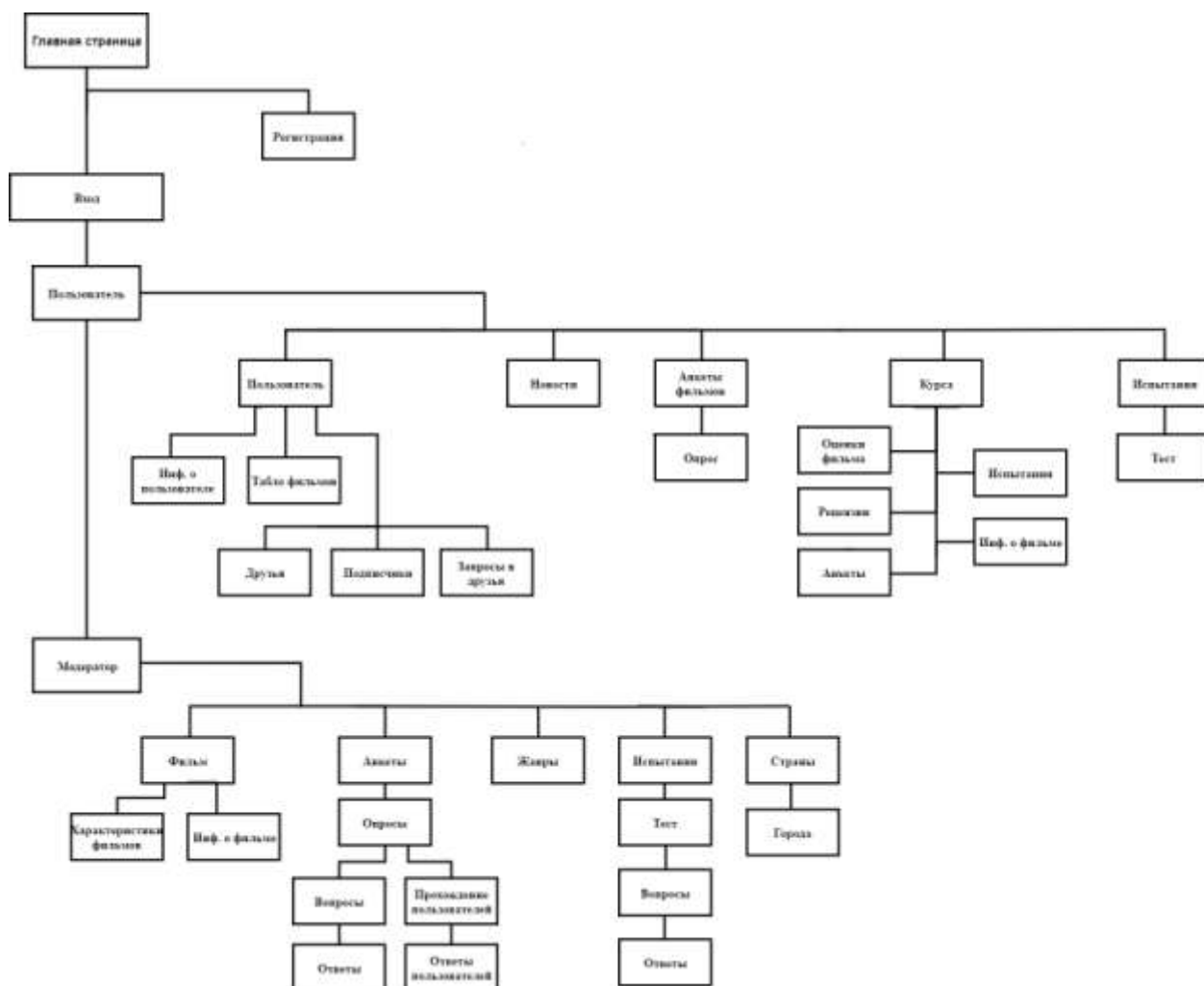


Рисунок 2.7 – Структура социальной платформы

Незарегистрированный пользователь видит следующие разделы сайта:

- главная страница;
- фильмы;
- регистрация пользователя.

Аутентифицированный пользователь видит следующие разделы сайта:

- главная страница;
- пользователь;
- новости;
- анкеты;



- рейтинг киномана;
- испытания;
- фильмы.

Аутентифицированный модератор видит следующие разделы сайта:

- главная страница;
- фильмы;
- характеристики;
- анкеты;
- города;
- жанры;
- испытания;
- пользователей.

### **Вывод по разделу два**

Построена модель приложения, на основе которой подготовлена структура приложения. Разработана база данных и определен состав таблиц и типы полей. Построена структура социальной платформы.

### 3 РАЗРАБОТКА

Сайт содержит страницы фильмов. Страницы фильмов создаются модераторами или лицами, доверенные как продюсерами фильмов, подтверждёнными модераторами.

На странице фильма представлены: постер фильма, информация о фильме, так же отображаются все рейтинги, используемые на сайте. На странице сайта можно оставить рецензию. Пользователь сможет оценить фильм и проголосовать за него в рейтинге. Так же из конкретного фильма пользователь сможет перейти на страницу испытания, где будут отображаться все испытания для этого фильма. Пользователи платформы смогут вести беседу о фильме, оставляя комментарий на странице фильма. Страница фильма представлена на рисунке 3.1.

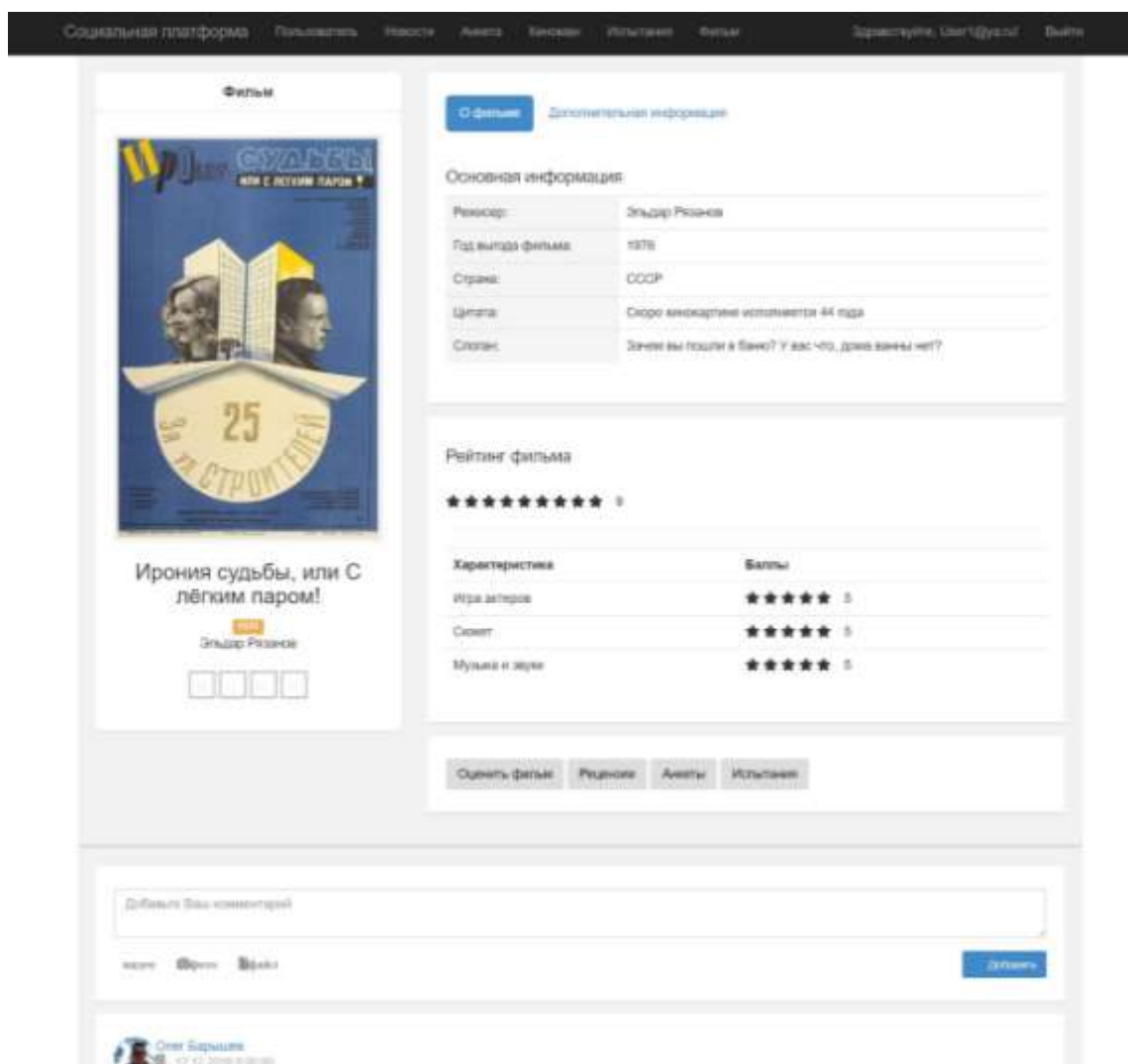


Рисунок 3.1 – Сайт страница фильма

Код контролера фильма приведен в листинге 1.

### Листинг 1 – Код контролера фильма

```
public class FilmController : Controller
{
    private readonly SocialNetContext db = new SocialNetContext();
    // GET: Film
    public ActionResult Index(int? id = 1, int? PageComment = 1 , int?
PageReplyComment = 1)
    {
        FilmView film = new FilmView();
        try
        {
            film.Film = db.Film.Find(id);
            film.InfoPrimaryFilm =
db.InfoPrimaryFilm.Find(film.Film.id_info_primary_film);
            film.InfoSecondaryFilm =
db.InfoSecondaryFilm.Find(film.InfoPrimaryFilm.id_info_secondary_film);

            film.PrSumRating = db.PrSumRating.Find(id);
            film.PrSumRatingVote = db.PrSumRatingVote.Where(x => x.id_film ==
id).ToList();

            //Комменты
            int pageComment = (PageComment ?? 1);
            int pageReplyComment = (PageReplyComment ?? 1);

            var commentFilm = db.CommentFilm.Where(x => x.id_film ==
id).OrderBy(x => x.id);
            film.CommentFilm = commentFilm.ToPagedList(pageComment, 3);
        }
        catch
        {
            return HttpNotFound();
        }
        return View(film);
    }
    // GET: Films
    public ActionResult IndexAllFilm()
    {
        try
        {
            var Film = db.PrFilmAllUser;
            return View(Film.ToList());
        }
        catch
        {
            return HttpNotFound();
        }
    }
}
```

Предложенное решение содержит страницы пользователей.

У каждого пользователя имеется своя личная страница, где он может делиться своими мыслями, идеями, интересами, любимой цитатой и любимыми фильмами. Пользователь имеет возможность создавать свой «Топ» понравившихся ему фильмов. Делиться любимыми моментами и комментариями на стене. Страница пользователя представлена на рисунках 3.2 и 3.3.

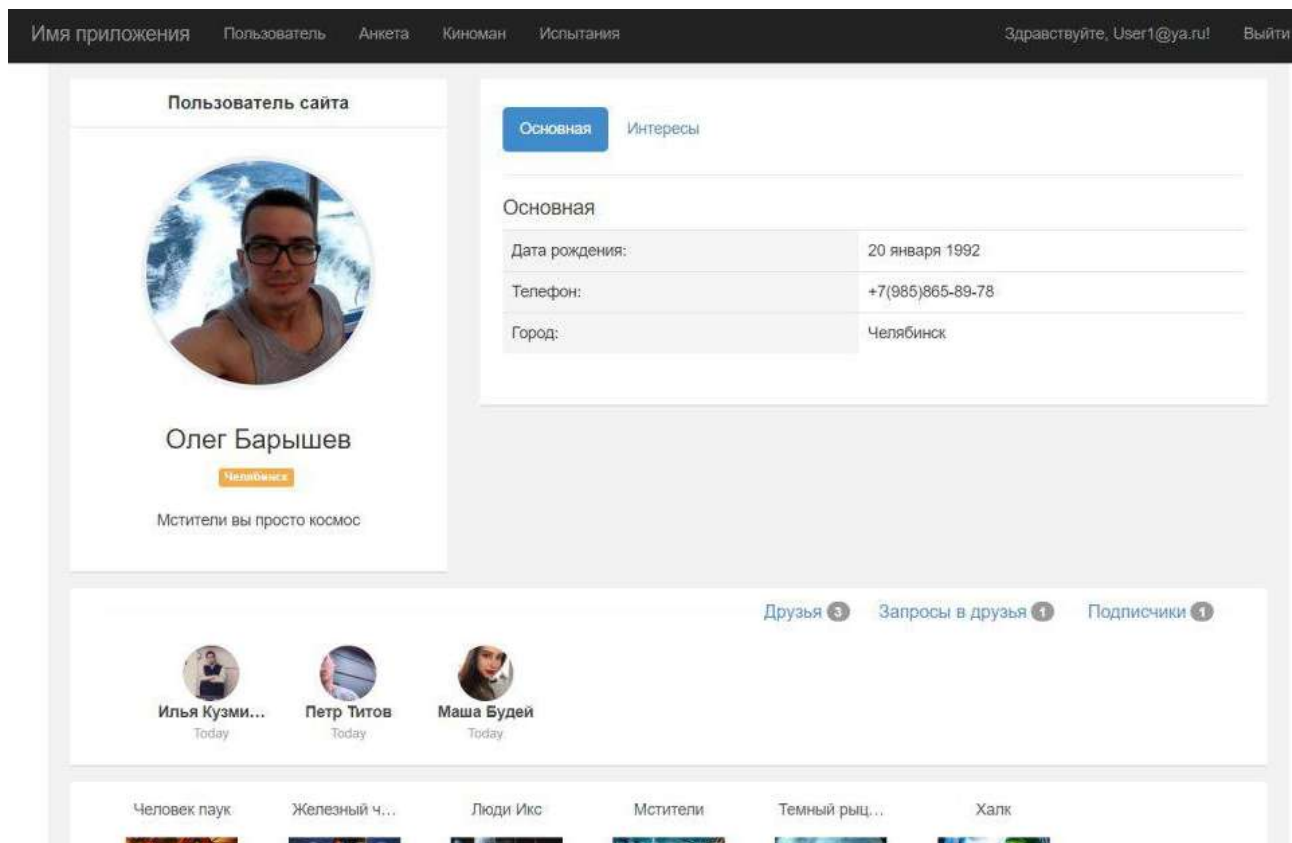


Рисунок 3.2 – Сайт страница пользователя

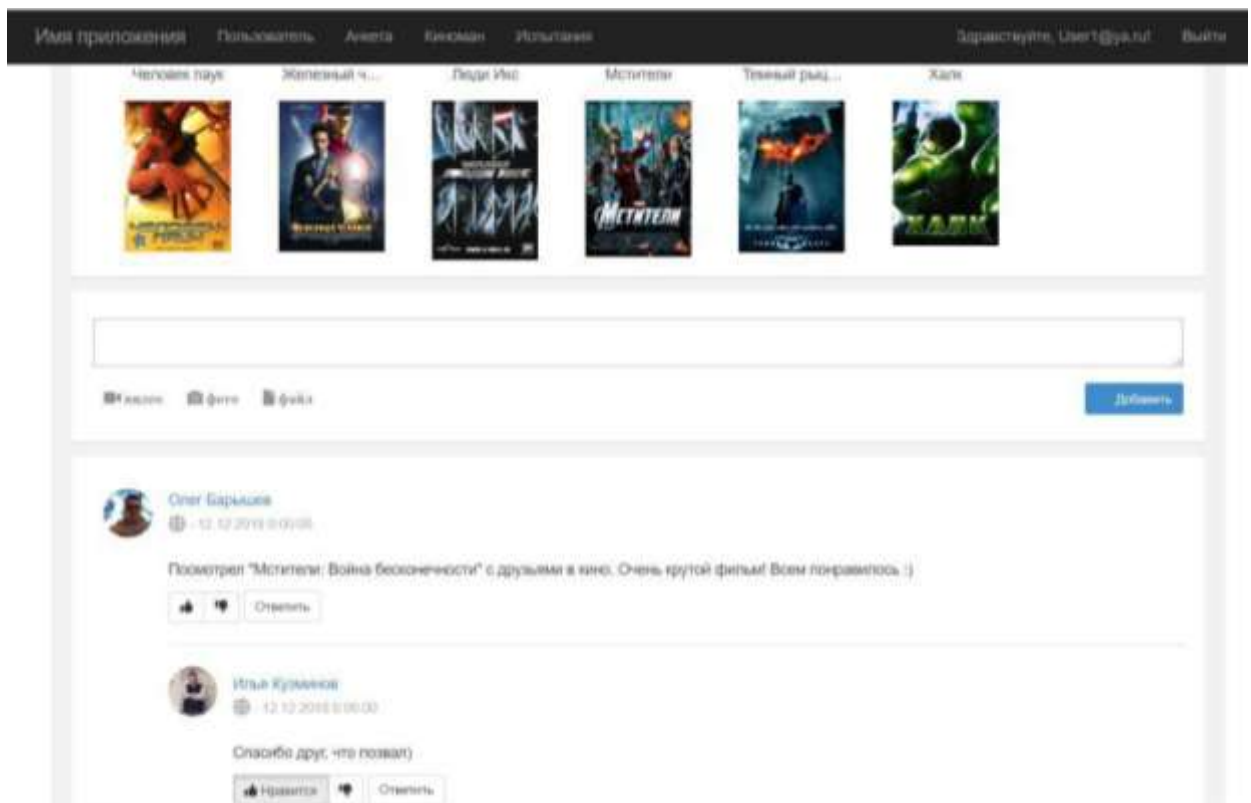


Рисунок 3.3 – Сайт страница пользователя, табло и комментарии

На данной пользовательской странице сайта возможно рассказать о себе, поделиться своими предпочтениями и вкусами. Так же пользователь имеет возможность добавить на табло свой любимый фильм, чтобы все видели его предпочтения. Существует возможность у пользователя на стене делиться с другими пользователями платформы постами, своими мнениями и комментариями. Код контролера страницы пользователя представлен в листинге 2.

Листинг 2 – Код контролера пользовательской страницы

```
namespace SocialNet.Controllers
{
    public class UserController : Controller
    {
        private readonly SocialNetContext db = new SocialNetContext();

        // GET: User
        public ActionResult Index(int? id, int? PageComment, int?
PageReplyComment)
        {
            UserView user = new UserView();
            try
            {
                #region Id User
                int _idUserAccount = 0;
            }
        }
    }
}
```

```

int _idUserPage = 0;
if (User.Identity.IsAuthenticated)
{
    if (Session["id_user_account"] == null)
    {
        _idUserAccount =
db.User.Find(Int32.Parse(User.Identity.GetUserId())).id;
        Session["id_user_account"] = _idUserAccount;
    }
    else
    {
        _idUserAccount = (int)Session["id_user_account"];
    }
}
if (id == null)
{
    _idUserPage = _idUserAccount;
    Session["id_user_page"] = _idUserPage;
}
else
{
    _idUserPage = (int)id;
}
#endregion
//Инфо. о пользователе
user.User = db.User.Find(_idUserPage);
user.InfoPrimaryUser =
db.InfoPrimaryUser.Find(user.User.id_info_primary_user);
user.InfoSecondaryUser =
db.InfoSecondaryUser.Find(user.InfoPrimaryUser.id_info_secondary_user);
user.GenreUser = db.GenreUser.Where(x => x.id_user ==
_ididUserPage).ToList();
//Друзья
var _friendSubscriber = db.FriendSubscriber.Where(x => x.id_user
== _idUserPage);
var _friendDemand = db.FriendDemand.Where(x => x.id_user ==
_ididUserPage);
var _friend = db.Friend.Where(x => x.id_user == _idUserPage);
user.Friend = _friend.ToList();

ViewData["count_subscribers"] = _friendSubscriber.Count();
ViewData["count_friend_demand"] = _friendDemand.Count();
ViewData["count_friend"] = _friend.Count();
#region Key Friend
if (User.Identity.IsAuthenticated)
{
    if (_idUserPage == _idUserAccount)
    {
        ViewData["state_friend"] = 1;
    }
    else
    {
        if (_friend.Where(x => x.id_user ==
_ididUserAccount).Count() == 1)
        {

```

```

        ViewData["state_friend"] = 3;
    }
    else if (_friendDemand.Where(x => x.id_demand_user ==
_idUserAccount).Count() == 0 && _friendSubscriber.Where(x => x.id_friend_user ==
_idUserAccount).Count() == 0)
    {
        ViewData["state_friend"] = 2;
    }
    else
    {
        ViewData["state_friend"] = 4;
    }
    }
}
else
{
    ViewData["state_friend"] = 0;
}
#endregion
//Комменты
int pageComment = (PageComment ?? 1);
int pageReplyComment = (PageReplyComment ?? 1);
var commentUser = db.CommentUser.Where(x => x.id_user ==
_idUserPage).OrderBy(x => x.id);
user.CommentUser = commentUser.ToPagedList(pageComment, 10);

//Фаворит фильм
var favoriteFilm = db.FavoriteFilmsUser.Where(x => x.id_user ==
_idUserPage).OrderBy(x => x.id);
user.FavoriteFilmsUser = favoriteFilm.ToPagedList(1, 10);
}
catch
{
    return HttpNotFound();
}
return View(user);
}

#region Comment
// POST: User/CreateComment
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult CreateComment([Bind(Include = "id_user,comment")]
CommentUser commentUser)
{
    commentUser.id_wrote_user = Int32.Parse(User.Identity.GetUserId());
    commentUser.date = DateTime.Now;

    if (ModelState.IsValid)
    {
        db.CommentUser.Add(commentUser);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}

```

```

        return RedirectToAction("Index");
    }

#region Friend
// GET: ListFriend
public ActionResult ListFriend(int id_user)
{
    var friend = db.Friend.Where(x => x.id_user == id_user);
    return View(friend.ToList());
}

public ActionResult DeleteFriend(int id_user, int id_friend_user, int
flag)
{
    try
    {
        Friend friend = db.Friend.Where(x => x.id_user == id_user).Where(x
=> x.id_friend_user == id_friend_user).First();

        db.Friend.Remove(friend);
        db.SaveChanges();
    }
    catch
    {
    }
    return RedirectToAction("Index");
}

#endregion

#region Edit Info User
// GET: InfoPrimaryUser/EditPrimary/
[Authorize()]
public ActionResult EditPrimary()
{
    try
    {
        int id_user = Int32.Parse(User.Identity.GetUserId());
        var _User = db.User.Find(id_user);
        int id_info_promary = _User.id_info_primary_user;

        InfoPrimaryUser infoPrimaryUser =
db.InfoPrimaryUser.Find(id_info_promary);
        if (infoPrimaryUser == null)
        {
            return HttpNotFound();
        }
        ViewBag.id_city = new SelectList(db.City, "id", "name",
infoPrimaryUser.id_city);
        ViewData["name_user"] = _User.UserName();
        return View(infoPrimaryUser);
    }
    catch
    {
        return HttpNotFound();
    }
}
}

```



```

// POST: InfoPrimaryUser/Edit/
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult EditPrimary([Bind(Include =
"id,id_info_secondary_user,id_city,quote,birthday,genres,phone,instagram,twitter,v
k,facebook,web_site,skype")] InfoPrimaryUser infoPrimaryUser)
{
    if (ModelState.IsValid)
    {
        db.Entry(infoPrimaryUser).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.id_city = new SelectList(db.City, "id", "name",
infoPrimaryUser.id_city);
    return View(infoPrimaryUser);
}
// GET: InfoSecondaryUser/EditSecond/
public ActionResult EditSecond()
{
    int id_user = Int32.Parse(User.Identity.GetUserId());
    var _User = db.User.Find(id_user);
    int id_info_primary = _User.id_info_primary_user;
    int id_info_second =
db.InfoPrimaryUser.Find(id_info_primary).id_info_secondary_user;
    InfoSecondaryUser infoSecondaryUser =
db.InfoSecondaryUser.Find(id_info_second);
    if (infoSecondaryUser == null)
    {
        return HttpNotFound();
    }
    ViewData["name_user"] = _User.UserName();
    return View(infoSecondaryUser);
}
// POST: InfoSecondaryUser/EditSecond/
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult EditSecond([Bind(Include =
"id,interests,about,activities,inspire,favorite_quote,favorite_book,favorite_serie
s")] InfoSecondaryUser infoSecondaryUser)
{
    if (ModelState.IsValid)
    {
        db.Entry(infoSecondaryUser).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(infoSecondaryUser);
}
#endregion

public ActionResult IndexAllUser()
{
    try
    {

```

```

        var User = db.User;
        return View(User.ToList());
    }
    catch
    {
        return HttpNotFound();
    }
}
}

```

Каждый пользователь имеет возможность отправлять заявку на добавление в друзья, либо остаться в подписчиках. Страница запроса пользователей представлена на рисунке 3.4.

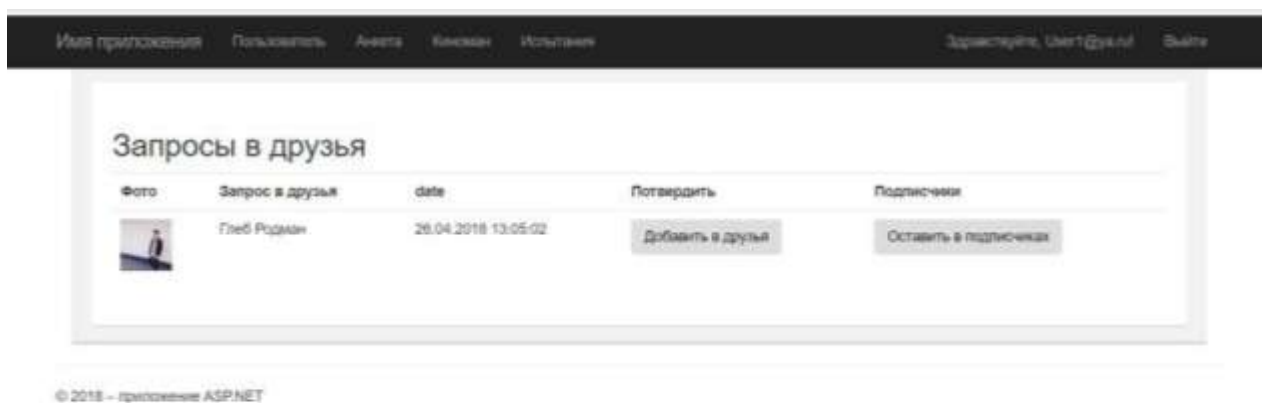


Рисунок 3.4 – Сайт страницы друзей пользователя

Для привлечения аудитории пользователей существует рейтинг киномана. Рейтинг показывает, кто лучше всех знает фильм и все, что с ними связано. Рейтинг киноманов позволяет сайту добавить пользователям соревновательного стимула, чтоб показать своим друзьям или даже всему миру, кто же лучший киноман. Рейтинг киноманов представлен на рисунке 3.5.






Рейтинг	Пользователь	Количество очков	Дата	
№ 1	 Олег Баранов	20	01.05.2018 15:18:02	<a href="#">Details</a>
№ 2	 Илья Курбанов	20	12.05.2018 18:26:14	<a href="#">Details</a>
№ 3	 Петр Титов	15	05.05.2018 10:11:06	<a href="#">Details</a>
	 Маша Буждай	14	08.05.2018 9:11:37	<a href="#">Details</a>
	 Генри Родман	5	03.05.2018 17:01:41	<a href="#">Details</a>

Рисунок 3.5 – Рейтинг киномана

Один из способов зарабатывать баллы киномана – прохождение испытаний. Испытания – это цепочка вопросов по определенной тематике или фильму. Пользователи, дающие правильные ответы, зарабатывают себе баллы киномана. Испытания представлены на рисунке 3.6.

## Вопрос

Какой экзамен сдавали студенты в фильме Приключения Шурика?



Механику Физику Химию Математику

Рисунок 3.6 – Испытания

Код контролера испытания представлен в листинге 3.

### Листинг 3 – Код контролера испытании

```
// GET: Test
public ActionResult Index()
{
    var test = db.Test.Include(t => t.Film);
    return View(test.ToList());
}
// GET: Test/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Test test = db.Test.Find(id);
    if (test == null)
    {
        return HttpNotFound();
    }
    return View(test);
}
// POST: Test/Question
public ActionResult Question(int? id_test, int? id_answer, string
str_answer)
```

```

{
    // FormCollection form
    Question question;
    try
    {
        //Загрузка списка вопросов
        if (Session["question"] == null)
        {
            if (id_test == null)
            {
                return RedirectToAction("Index");
            }
            List<Question> _listQuestion = new List<Question>();
            List<List<Answer>> _listAnswer = new List<List<Answer>>();

            id_test).ToList();
            _listQuestion = db.Question.Where(x => x.id_test ==
            foreach (var item in _listQuestion)
            {
                _listAnswer.Add(item.Answer);
            }

            Session["question"] = new Queue<Question>(_listQuestion);
            Session["answer"] = new Queue<List<Answer>>(_listAnswer);
            Session["id_test"] = id_test;
            Session["point"] = 1;

        }
        var _queueQuestion = (Queue<Question>)Session["question"];
        var _queueAnswer = (Queue<List<Answer>>)Session["answer"];
        //Количество правильных ответов
        if (id_answer != null)
        {
            if ((int)Session["correct"] == id_answer)
            {
                int point = (int)Session["point"];
                point += 1;
                Session["point"] = point;
            }
        }
        //Следующий вопрос
        if (_queueQuestion.Any())
        {
            question = _queueQuestion.Dequeue();
            question.Answer = _queueAnswer.Dequeue();
            Session["correct"] = question.correct_answer;
        }
        else
        {
            var _userManager =
            UserManager.FindById(HttpContext.User.Identity.GetUserId());
            int IdUser = db.User.Where(x => x.IdUser ==
            _userManager.Id).First().id;

            QuestionResult questionResult = new QuestionResult()

```

```

        {
            id_test = (int)Session["id_test"],
            id_user = IdUser,
            point = (int)Session["point"],
            date = DateTime.Now
        };

        RatingFilm ratingFilms = db.RatingFilms.Where(x => x.id_user
== IdUser).First();
        ratingFilms.point += (int)Session["point"];
        ratingFilms.date = DateTime.Now;

        db.Entry(ratingFilms).State = EntityState.Modified;
        db.QuestionResult.Add(questionResult);
        db.SaveChanges();
        return RedirectToAction("Finale");
    }
}
catch
{
    return HttpNotFound();
}

return View(question);
}
// GET: Test/Finale
public ActionResult Finale()
{
    ViewData["point"] = Session["point"];
    return View();
}
}

```

На сайте присутствует функционал для создания опроса, который поможет продюсерам фильмов или их доверенным лицам использовать данный инструмент для осуществления обратной связи по интересующих их вопросам, то есть позволит узнать объективное мнение пользователей.

Пользователи, проходящие опрос, так же зарабатывают баллы в свой общий зачет киномана.


Опрос может быть, как закрытым для конкретного круга пользователя, так и открытым для всех, кто желает его пройти. Так же к каждому опросу можно прикрепить тест, только после этого опрос станет доступным для прохождения. Опрос представлен на рисунке 3.7 и 3.8.

Опрос

Понравился ли вам новый человек паук?

---

Понравился ли вам актер Том Холланд в роли Человека-паука?



★  
 ★★  
 ★★★  
 ★★★★  
 ★★★★★

Рисунок 3.7 – Опрос

Имя приложения Пользователь Анкета Киноман Испытания Здравствуйте, User1@ya.ru! Выйти

Были ли смешные шутки?

- Да, было очень смешно
- Да, сойдет
- Нет, я не понял(а) юмора
- Детский юмор
- Заезженный юмор

Опишите пожалуйста ваши впечатления от просмотра фильма?

Рисунок 3.8 – Опрос, продолжение

У пользователя есть возможность оценить фильм по своим впечатлениям. Для этого надо перейти на страницу фильма и нажать на кнопку «оценить фильм». Пользователю будет предложен список характеристик, по которым необходимо проголосовать, и одна общая оценка – общее впечатление о фильме. Характеристики будут уникальными для каждого фильма. Благодаря этим характеристикам можно узнать мнение пользователей по вопросам, интересующим продюсеров фильмов. Например, был ли хорош данный актер в своей роли. Страница оценки фильма представлена на рисунке 3.9.

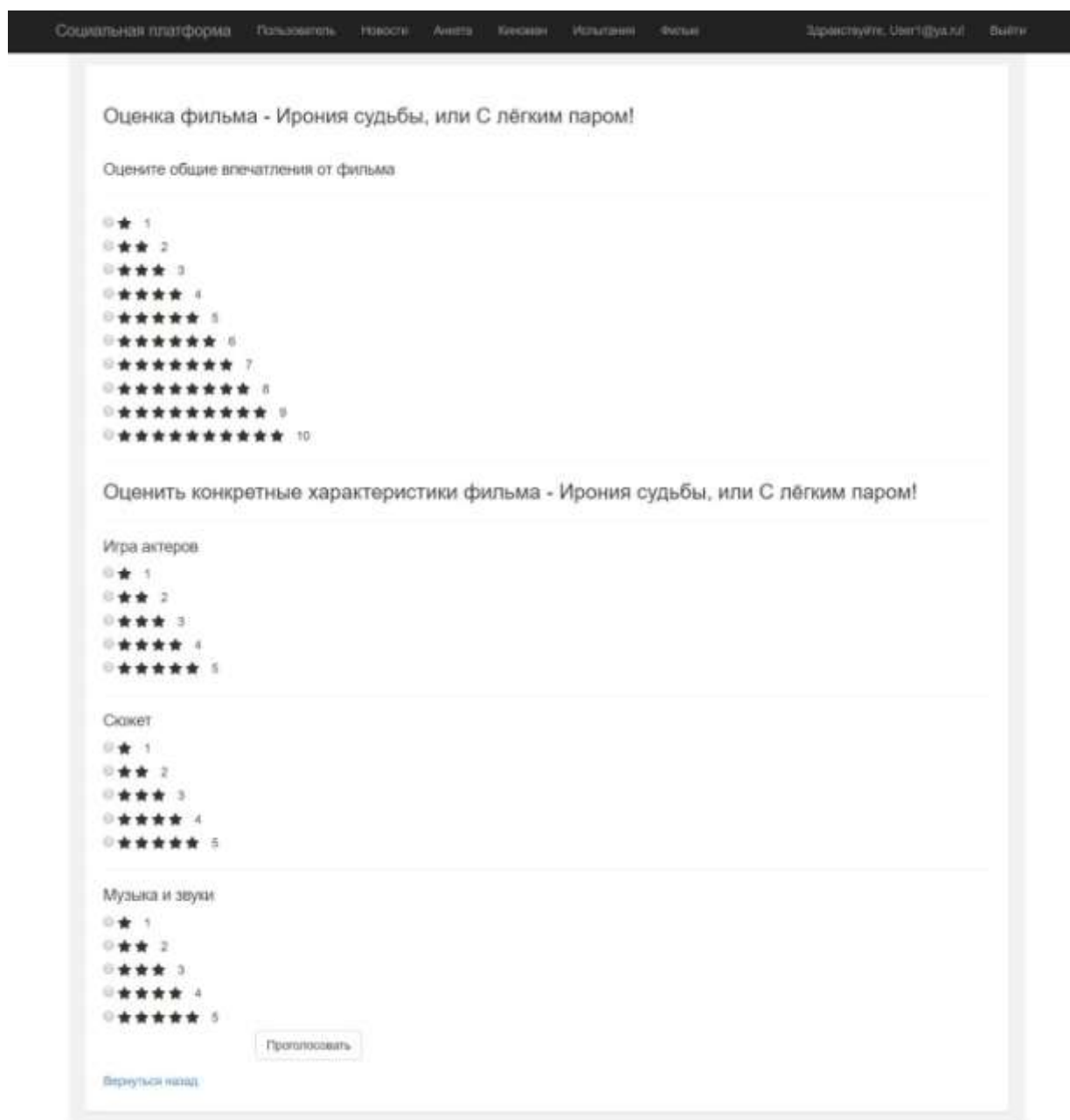


Рисунок 3.9 – Оценить фильм



Код контролера оценить фильм представлен в листинге 4.

#### Листинг 4 – Код контролера оценить фильм

```
public class RatingController : Controller
{
    private SocialNetContext db = new SocialNetContext();
    // GET: Rating
    public ActionResult Index()
    {
        var rating = db.Rating.Include(r => r.Film).Include(r => r.User);
        return View(rating.ToList());
    }
    // GET: Rating/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Rating rating = db.Rating.Find(id);
        if (rating == null)
        {
            return HttpNotFound();
        }
        return View(rating);
    }
    // GET: Rating/Create
    //id_film
    public ActionResult Create(int? id = 1)
    {
        var Film = db.Film.Find(id);
        ViewData["name"] = Film.name;
        ViewBag.id_film = id;
        ViewBag.id_user = new SelectList(db.User, "id", "name");
        RatingView RatingView = new RatingView();
        var Characteristic = db.Characteristic.Where(x => x.id_film ==
id).ToList();
        Session["Characteristic"] = Characteristic;
        RatingView.Characteristic = Characteristic;
        return View(RatingView);
    }
    // POST: Rating/Create
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include = "id,id_film,point")] Rating
rating, FormCollection form)
    {
        //FormCollection form
        try
        {
            int IdUser = Int32.Parse(User.Identity.GetUserId());
            rating.id_user = Int32.Parse(User.Identity.GetUserId());
            rating.date = DateTime.Now;

```

```

        List<Characteristic> Characteristic =
(List<Characteristic>)Session["Characteristic"];
        foreach (var item in Characteristic)
        {
            RatingVote ratingVote = new RatingVote
            {
                id_characteristic = item.id,
                id_user = IdUser,
                point =
Int32.Parse(form.GetValues(item.id.ToString()).First()),
                date = DateTime.Now
            };
            db.RatingVote.Add(ratingVote);
        }
        if (ModelState.IsValid)
        {
            db.Rating.Add(rating);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.id_film = new SelectList(db.Film, "id", "name", rat-
ing.id_film);
        ViewBag.id_user = new SelectList(db.User, "id", "first_name", rat-
ing.id_user);
        return View(rating);
    }
    catch
    {
        return HttpNotFound();
    }
}
}

```

Пользователь может оставить рецензию на фильм. Для этого он должен выбрать фильм и нажать на кнопку «оставить рецензию». После того как пользователь напишет заголовок и текст рецензии, ему необходимо оценить фильм, если он не делал этого раньше. Страница рецензии фильма представлена на рисунке 3.10.

## Рецензии на фильм: Ирония судьбы, или С лёгким паром!

[Добавить рецензию](#)

Олег Барышев

13.05.2016 в 00:30

## Елка, мандарины, «Ирония судьбы»...

Традиционный новогодний фильм целиком и полностью посмотрела сравнительно недавно. Все больше как-то гуляем, под мандарины и новогодние тосты... а юности, причем летом, я его впервые посмотрела полностью. Тогда он мне очень понравился. Тонкостью актерской игры, например. Все это любовно-градусное коммешательство сыграно со вниманием к мельчайшим нюансам. Каждый взгляд, каждая реплика шуток, каждый жест — все это дает зрителю одутить то невольное и невольное впечатление, что почувствовали друг к другу уже вполне устроенные в жизни, казалось бы, люди. И не столь важно, будут ли они вместе. Важно, что оба не совершили ошибку пойти на уступки общественному мнению «пора жениться» «пора закруж». Не так уж они и любяли своих супругов... Писая я рецензию тогда, я поставила бы десятку, но, глядя уже трезвым взглядом... конечно же, есть тут и недостатки. Рязанов бывает несколько акорден, и тут эта акордность, увы, тоже есть. Кое-где действие явно подвисает. Да и финал... история такова, что подходящего финала к ней просто нет, и мы делаемся, так сказать, отфильм. Мне лично еще совершенно не нравится главный герой. Как он с таким истеричным характером умудряется работать хирургом? Или все хладнокровие и жесткость он выстраивает у операционного стола? В кадре он то мучится, то взрывается какой-то выходкой. Непредсказуемое совершенно существо. А ирония... потрясающей смелости, конечно, тема, потому что если бы я пришла домой и увидела на своей кровати спящего незнамого мужчину, я моментально высочила бы на лестницу, заперла бы дверь и позвонила бы в милицию — от соседей. Мне без разницы было бы, как он туда попал, чего хотел... ну, с другой стороны, фильма бы не было, поступил она так, а жанр фильма все же не серьезный. Актерский состав, конечно же, звездный. Яковлев, Аладжакова, Мигнов... Тальзина в роли голоса Нади и подруги Нади. Тут, конечно, только смотреть и радоваться. Очень ложится на душу «то никого не защитила ядали обещанная встреча...» Как и песни про тетю. Пожалуй, песни ключевая, а отпавшие от совершенно для орнамента добавленной «Платочки белые». Собственно, «и если вы не живете, то вам и не умирать», а значит, и наоборот, не терять можно только не жалея по-настоящему. Герои потеряли планы на спокойную жизнь, а приобрели — омерзительное, но такое важное чувство. Влюбленность. Не любовь, нет, откуда она с нескольких часов знакомства. Может быть, и любовь к ним еще придет... Юмор, конечно, Рязановский. Сложный, но всегда добрый, иногда даже неприятный. Какой-то даже мейнстримный... И, пожалуй, фильм слегка проигрывает своему брату — «Служебному роману». Там и действие крепче, и ритм лоще, и история любви более оживленная.

Оценка фильму - 10

Игра актеров - 5; Совет - 5; Музыка и звук - 5

Рисунок 3.10 –Рецензии на фильм

У каждого пользователя есть новостная лента, составленная из записей, оставленных на стене друзей и подписчиков, а также из новостей подписанных фильмов. Новостная лента представлена на рисунке 3.11.

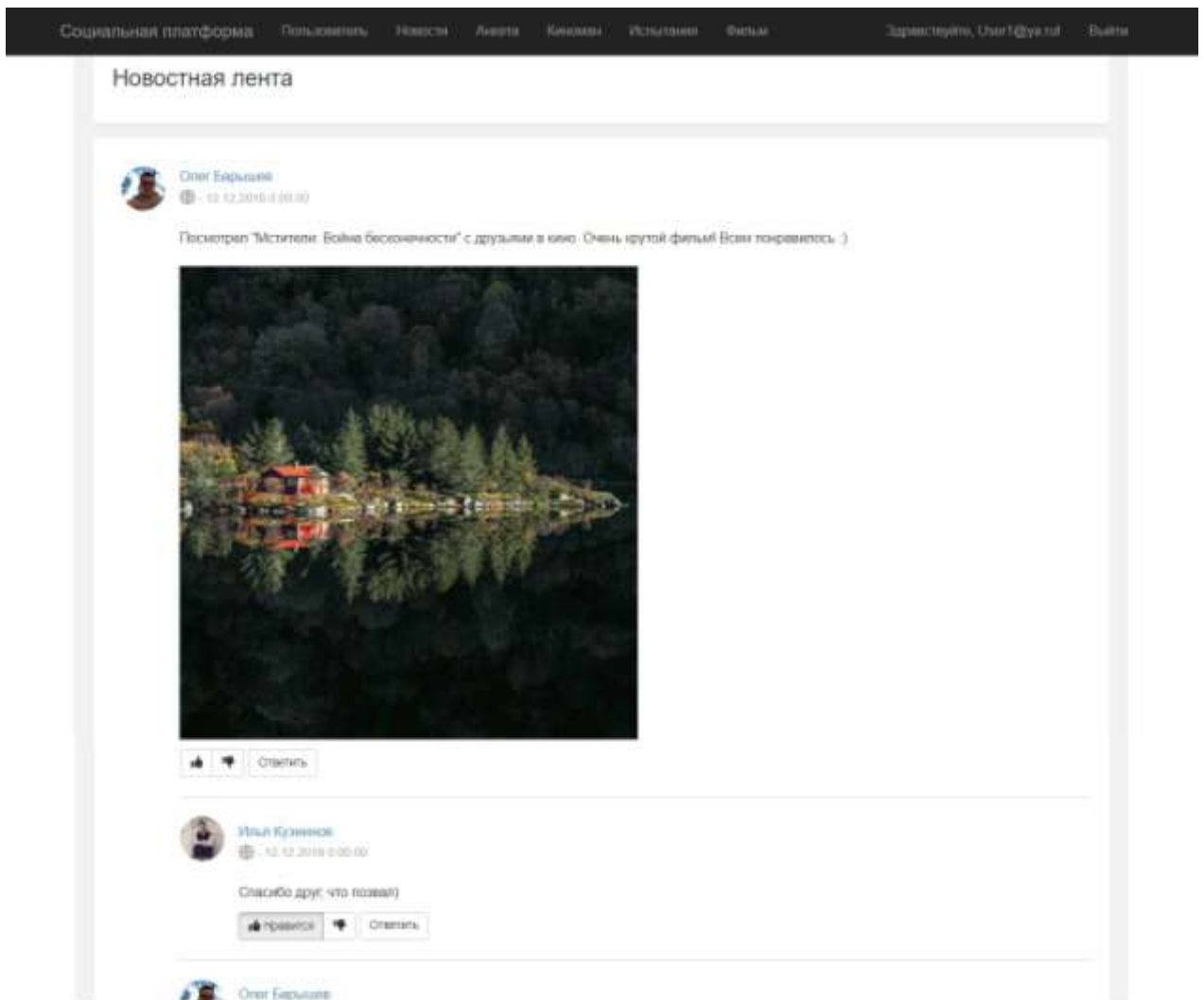


Рисунок 3.11 –Новостная лента пользователей

Код контролера новостной ленты пользователя представлен в листинге 5.

Листинг 5 – Код контролера новостной ленты пользователя

```
public class NewsController : Controller
{
    private readonly SocialNetContext db = new SocialNetContext();
    // GET: News
    public ActionResult Index()
    {
        List<CommentView> ComemntView = new List<CommentView>();
        //Comment User
        var CommentUser = db.CommentUser.Where(x => x.id_user ==
x.id_wrote_user);
        foreach (var item in CommentUser)
        {
            CommentView model = new CommentView()
            {
                id_user = item.id_user,
```

```

        comment = item.comment,
        material = item.material,
        date = item.date,
        User = db.User.Find(item.id_user),
        ReplyCommentUser = db.ReplyCommentUser.Where(x =>
x.id_comment_user == item.id).ToList()
    };
    ComemntView.Add(model);
}
//ComemntFilm
var CommentFilm = db.CommentFilm.Where(x => x.id_film ==
x.id_wrote_film);

foreach (var item in CommentFilm)
{
    CommentView model = new CommentView()
    {
        id_film = item.id_film,
        title = item.title,
        comment = item.comment,
        material = item.material,
        date = item.date,
        Film = db.Film.Find(item.id_film),
        ReplyCommentFilm = db.ReplyCommentFilm.Where(x =>
x.id_comment_film == item.id).ToList()
    };
    ComemntView.Add(model);
}
ComemntView.OrderBy(x => x.date);
return View(ComemntView);
}
}

```

Табло фильмов предназначено для того, чтобы пользователь показал, какие фильмы являются для него избранными или любимыми. Каждый пользователь, заходя на страницы других пользователей, будет видеть избранные фильмы данного пользователя. Табло фильмов представлено на рисунке 3.12.

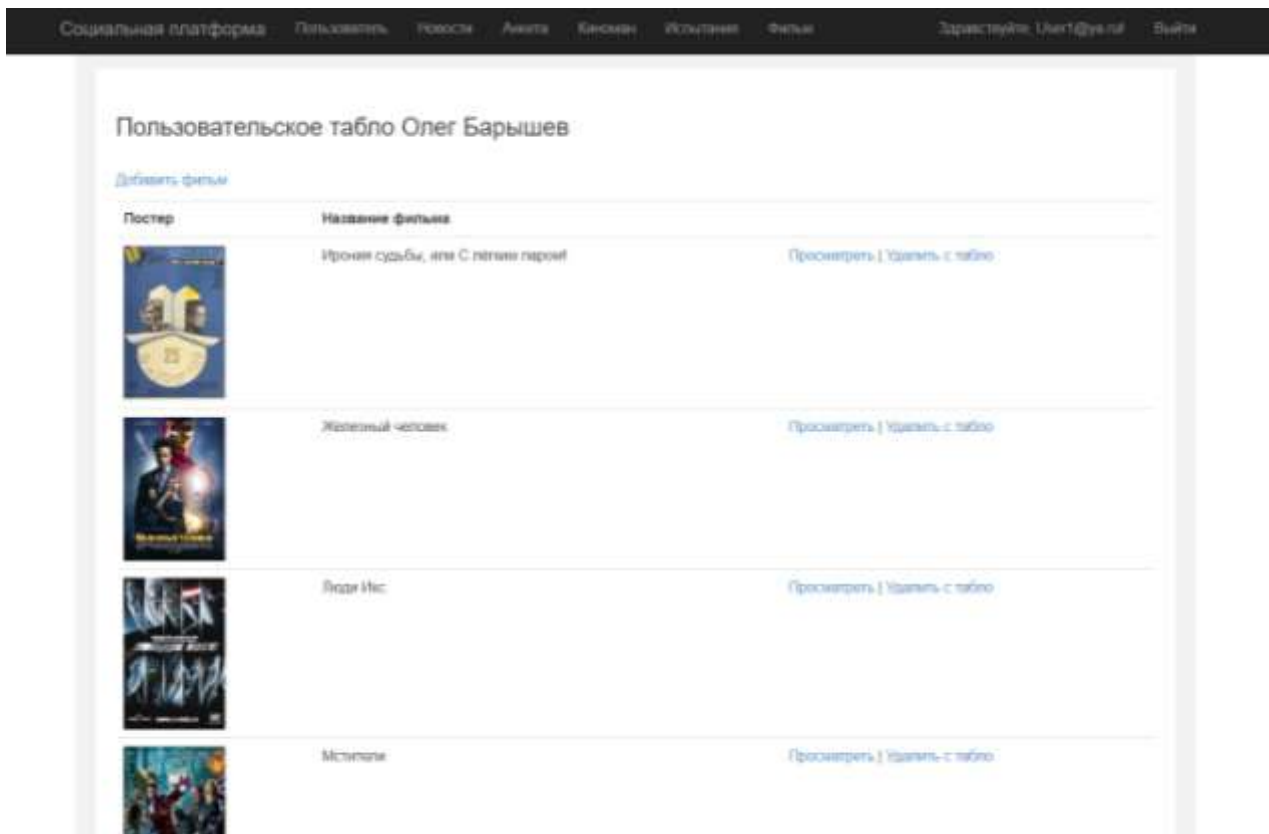


Рисунок 3.12 – Редактирование пользовательского табло

Модератор сайта ведет учет опросов и испытаний в системе. Страница опросы и создание опроса модератором представлена на рисунке 3.13 и 3.14.

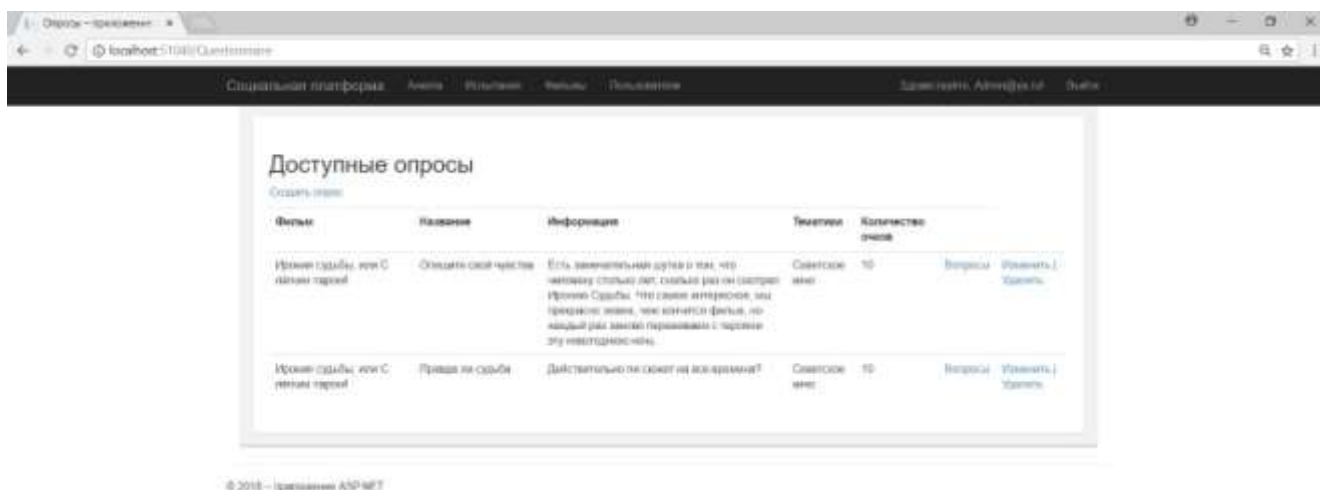


Рисунок 3.13 – Страница опросы в системе

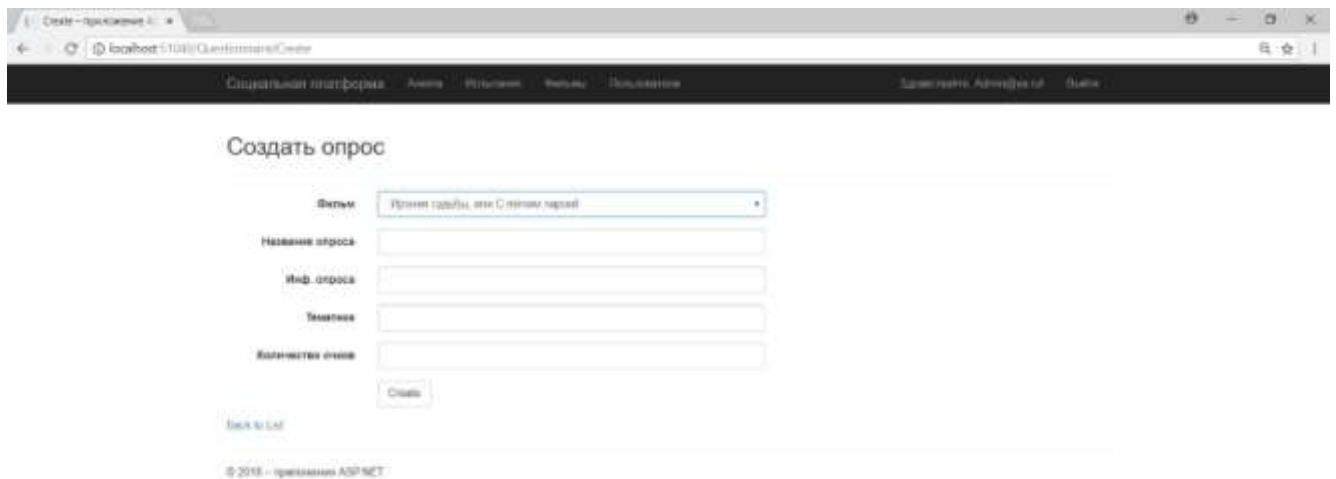


Рисунок 3.14 – Страница создание опроса

Код контролера опроса приведен в листинге 6.

Листинг 6 – Код контролера опроса

```
public class QuestionnaireController : Controller
{
    private SocialNetContext db = new SocialNetContext();
    // GET: Questionnaire
    public ActionResult Index()
    {
        var questionnaire = db.Questionnaire.ToList();
        try
        {
            if (User.IsInRole("Admin"))
            {
                return View("IndexAdmin", questionnaire);
            }
            return View(questionnaire);
        }
        catch
        {
            return View(questionnaire);
        }
    }
    // GET: Questionnaire/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Questionnaire questionnaire = db.Questionnaire.Find(id);
        if (questionnaire == null)
        {
            return HttpNotFound();
        }
        return View(questionnaire);
    }
}
```

```

    }
    public ActionResult Question(int? id_question, int? id_answer)
    {
        List<QuestionnaireQuestion> listQuest;
        try
        {
            var _get_id = HttpContext.User.Identity.GetUserId();
            Session["id_user_account"] = db.User.Where(x => x.IdUser ==
_get_id).First().id;
            ViewData["Name"] = db.Questionnaire.Find(id_question).name;
            Session["id_questionnaire"] = id_question;
            listQuest = db.QuestionnaireQuestion.Where(x => x.id_questionnaire
== id_question).ToList();
        }
        catch
        {
            return HttpNotFound();
        }
        return View(listQuest);
    }
    public ActionResult Finale(FormCollection form, List<string> str_answer)
    {
        try
        {
            int _id_questionnaire = (int)Session["id_questionnaire"];
            int _idUser = (int)Session["id_user_account"];
            QuestionnaireReady ready = new QuestionnaireReady
            {
                id_questionnaire = _id_questionnaire,
                id_user = _idUser,
                date = DateTime.Now
            };
            db.QuestionnaireReady.Add(ready);
            db.SaveChanges();
            int id_questionnaire_ready = ready.id;
            foreach (string item in form)
            {
                if (item != "str_answer")
                {
                    var answer = form[item];
                    QuestionnaireAnswer q_answer =
db.QuestionnaireAnswer.Find(Int32.Parse(answer));
                    QuestionnaireReadyAnswer quest_answer = new
QuestionnaireReadyAnswer
                    {
                        id_questionnaire_ready = id_questionnaire_ready,
                        id_questionnaire_question = q_answer.id_question,
                        id_questionnaire_answer = q_answer.id,
                    };
                    db.QuestionnaireReadyAnswer.Add(quest_answer);
                }
            }
        }
    }
}

```



```

        RatingFilm ratingFilms = db.RatingFilms.Where(x => x.id_user ==
_idUser).First();
        IQueryable<Questionnaire> point_add = db.Questionnaire.Where(x =>
x.id == _id_questionnaire);
        ratingFilms.point = ratingFilms.point + point_add.First().point;
        ratingFilms.date = DateTime.Now;
        db.Entry(ratingFilms).State = EntityState.Modified;
        db.SaveChanges();
        ViewData["point"] = ratingFilms.point;
    }
    catch
    {
    }
    return View();
}
// GET: Questionnaire/Create
public ActionResult Create()
{
    ViewData["id_user"] = int.Parse(User.Identity.GetUserId());
    ViewBag.id_film = new SelectList(db.Film, "id", "name");
    return View();
}
// POST: Questionnaire/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"id,id_user,id_film,name,other,subjects,point")] Questionnaire questionnaire)
{
    questionnaire.date = DateTime.Now;
    if (ModelState.IsValid)
    {
        db.Questionnaire.Add(questionnaire);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.id_film = new SelectList(db.Film, "id", "name");
    return View(questionnaire);
}
// GET: Questionnaire/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }

    ViewBag.id_film = new SelectList(db.Film, "id", "name");
    ViewData["id_user"] = int.Parse(User.Identity.GetUserId());
    Questionnaire questionnaire = db.Questionnaire.Find(id);
    if (questionnaire == null)
    {
        return HttpNotFound();
    }
    return View(questionnaire);
}
}

```

```

// POST: Questionnaire/Edit/5
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
"\"id,id_user,id_film,name,other,subjects,point\"")] Questionnaire questionnaire)
    {
        questionnaire.date = DateTime.Now;
        if (ModelState.IsValid)
        {
            db.Entry(questionnaire).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.id_film = new SelectList(db.Film, "id", "name");
        return View(questionnaire);
    }
// GET: Questionnaire/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Questionnaire questionnaire = db.Questionnaire.Find(id);
    if (questionnaire == null)
    {
        return HttpNotFound();
    }
    return View(questionnaire);
}
// POST: Questionnaire/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Questionnaire questionnaire = db.Questionnaire.Find(id);
    db.Questionnaire.Remove(questionnaire);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}

```

Модератор сайта так же ведет учет самих вопросов и ответов на каждый опрос. Страница вопросов и ответов приведены на рисунках 2.13, 2.14.

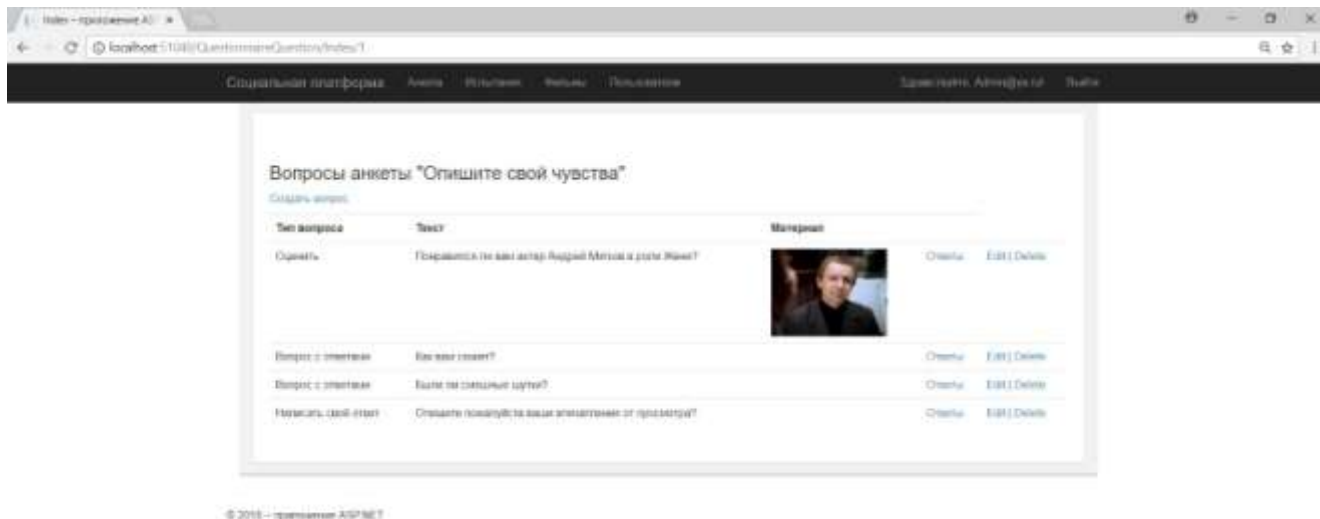


Рисунок 3.15 – Страница вопросов опроса

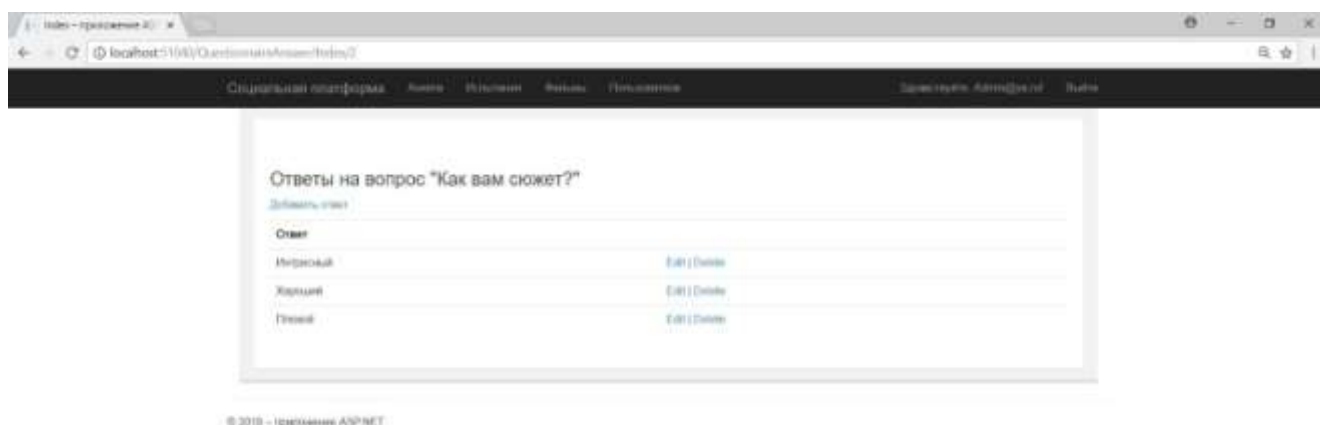


Рисунок 3.15 – Страница ответов на вопросы опроса

Модератор создает и ведет учет индивидуальных характеристик, по которым оценивается фильм. Страница характеристики представлена на рисунке 3.16.

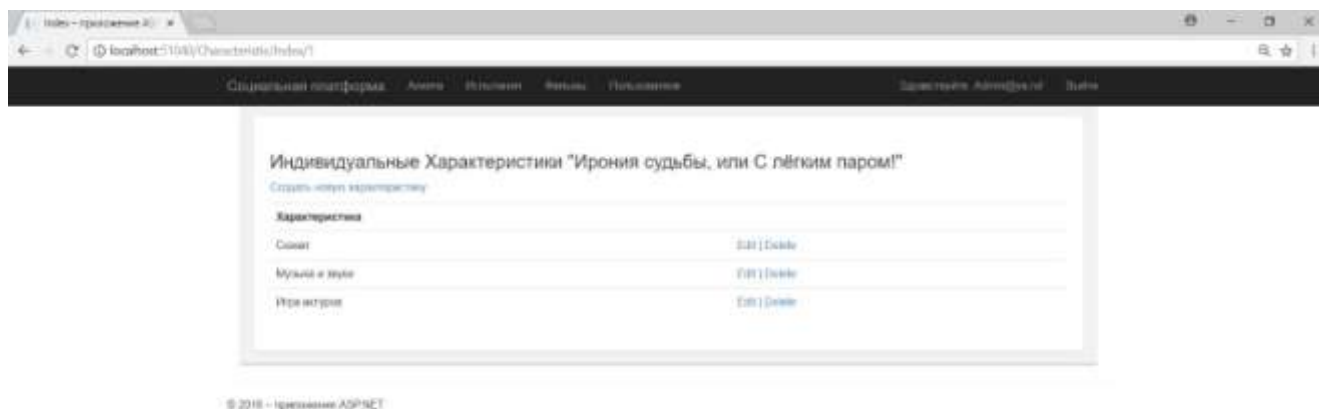


Рисунок 3.16 – Страница индивидуальных характеристик

### **Вывод по разделу три**

Реализована социальная платформа, позволяющая вести учет мнений пользователя и накапливать информацию в базу данных. Описан функционал работы социальной платформы и приведен интерфейс и листинг.

## ЗАКЛЮЧЕНИЕ

Существующая проблема является большим упущением отечественного кинематографа. Имеющаяся возможность снимать качественные и прибыльные кинокартины, которыми можно гордиться, упускается в связи с незнанием пользовательского мнения.

В результате выполнения дипломной работы спроектирована и реализована социальная платформа. Рассмотрена и разобрана существующая проблема, предложен свой подход к ее решению. Сделан анализ существующих решений, разработана логика приложения для решения сложившейся проблемы. Разработана база данных и соответствующий функционал для эффективного способа накопления, обработки и целостного хранения мнений пользователей о выпускаемой продукции, решающей поставленную задачу.

В ходе дипломного проекта достигнуты следующие задачи.

- Проанализирована проблема финансирования.
- Предложено свое решение по решению проблемы.
- Сделан анализ существующих систем.
- Разработана логика будущего приложения.
- Спроектирована и разработана база данных.
- Разработан свой функционал.
- Разработан интерфейс взаимодействия с пользователем.
- Выполнено тестирование.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сартасов, Е.М. Методология и технология проектирования информационных систем: Учебное пособие. – Челябинск: Издательский центр ЮУрГУ, 2014. – 49 с.
2. Емельянова, Н.З. Проектирование информационных систем: учеб. пособие для сред. проф. образования / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. – М.: Форум, 2013. – 432 с.
3. Бондарь, А. Microsoft SQL Server 2014 /Александр Бондарь – БХВ-Петербург 2015г. – 592с.
4. Гарсиа-Молина, Г. Системы баз данных. Полный курс / Г. Гарсиа-Молина, Дж. Ульман, Дж. Уидом. – М.: Вильямс, 2003г. – 1088с.
5. Лабберс, П. HTML5 для профессионалов: мощные инструменты для разработки современных web-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011г. – 272с.
6. Бондарь, А. Microsoft SQL Server 2014 /Александр Бондарь – БХВ-Петербург 2015г. – 592с.
7. Титтел, Э. HTML, XHTML и CSS для чайников, 7-е издание / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011г. – 400 с.
8. Ливингстон, Д. CSS и XHTML: Web – профессионалам / Д. Ливингстон, М. Браун. – Киев: BHV, 2001. 264 с.
9. Шапошников, И. Web-сервисы Microsoft .NET/ Игорь Шапошников Изд-во БХВ-Петербург 2002г. – 334с.
10. Мартин, Ф. UML Основы. Краткое руководство по стандартному языку объектного моделирования / Мартин Фаулер пер. с англ. А. Петухова СПб: Символ–Плюс, 2006. – 192 с.