МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация 1-40 01 01 Программное обеспечение информационных технологий (программирование интернет-приложений)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема «Программное средство «MovieDB»»

Исполнитель

студент 2 курса группы 4 Конопацкий Александр Александрович

(Ф. И. О.)

Руководитель работы ассистент Рауба А. А.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Рауба А. А.

(подпись)

Минск 2020

Содержание

[Введение 3](#_Toc41264057)

[1. Аналитический обзор литературы и постановка задачи 4](#_Toc41264058)

[1.1 Анализ прототипов 4](#_Toc41264059)

[1.2 Постановка задачи 6](#_Toc41264060)

[2. Анализ требований к программному средству и разработка функциональных требований 7](#_Toc41264061)

[2.1 Определение требований к программному средству 7](#_Toc41264062)

[2.2 Описание средств разработки 7](#_Toc41264063)

[2.3 Описание функциональности ПС 8](#_Toc41264064)

[2.4 Спецификация функциональных требований 8](#_Toc41264065)

[3. Проектирование и создание программного средства 10](#_Toc41264066)

[3.1 Модель базы данных 10](#_Toc41264067)

[3.2 Структура проекта и выполняемые функции 11](#_Toc41264068)

[3.2.1 Регистрация и авторизация 13](#_Toc41264069)

[3.2.2 Просмотр списка фильмов 14](#_Toc41264070)

[3.2.3 Уведомление о выходе в прокат и поиск 14](#_Toc41264071)

[3.2.4 Отображение статистики 15](#_Toc41264072)

[3.2.5 Настройка параметров работы с приложением 16](#_Toc41264073)

[3.3 Взаимоотношения между классами 16](#_Toc41264074)

[4. Тестирование 18](#_Toc41264075)

[4.1 Тестирование авторизации и регистрации 18](#_Toc41264076)

[4.2 Тестирование функций приложения 21](#_Toc41264077)

[5. Руководство по использованию 22](#_Toc41264078)

[5.1 Руководство по установке 22](#_Toc41264079)

[5.2 Руководство по использованию 22](#_Toc41264080)

[Заключение 28](#_Toc41264081)

[Список литературных источников 29](#_Toc41264082)

[Приложение А 30](#_Toc41264083)

[Приложение Б 31](#_Toc41264084)

[Приложение В 32](#_Toc41264085)

[Приложение Г 35](#_Toc41264086)

# **Введение**

Фильмы являются неотъемлемой частью досуга многих людей независимо от их происхождения, возраста, пола. Но все эти люди всегда задаются одним и тем же вопросом, что бы посмотреть?

На этот вопрос можно найти ответ на сайтах, в которых нам предоставляют обширный каталог со всевозможными фильмами, и опираясь на такие приложения, разрабатывался мой проект.

Для успешной реализации курсового проекта необходимо:

* провести анализ соответствующей литературы;
* ознакомиться с прототипами программных средств выбранной мной темы;
* определить функциональные требования;
* продумать структуру базы данных;
* продумать структуру проекта;
* реализовать программное средство;
* протестировать программное средство;
* написать руководство пользователя.

Содержание данной пояснительной записки отражает все этапы выполнения моего курсового проекта.

# **Аналитический обзор литературы и постановка задачи**

Для того чтобы окончательно определиться с постановкой задачи моего курсового проекта, необходимо проанализировать прототипы программных средств выбранной темы.

## **1.1 Анализ прототипов**

*IMDb* [1] – крупнейшая в мире база данных и сайт о кинематографе. В базе собрана иформация о более чем 450 тыс. фильмов и телесериалов, а также о 2 млн. актеров, режиссеров и других профессионалов кино. Сайт imdb.com постоянно входит в сотню самых популярных сайтов.

Плюсы:

* удобный и понятный интерфейс;
* возможность ставить оценки и писать рецензии;
* особая система подсчёта рейтинга фильма;
* возможность редактирования информации в базе данных любым пользователем.

Минусы:

* отсутстует desktop версия;
* cайт является англоязычным.

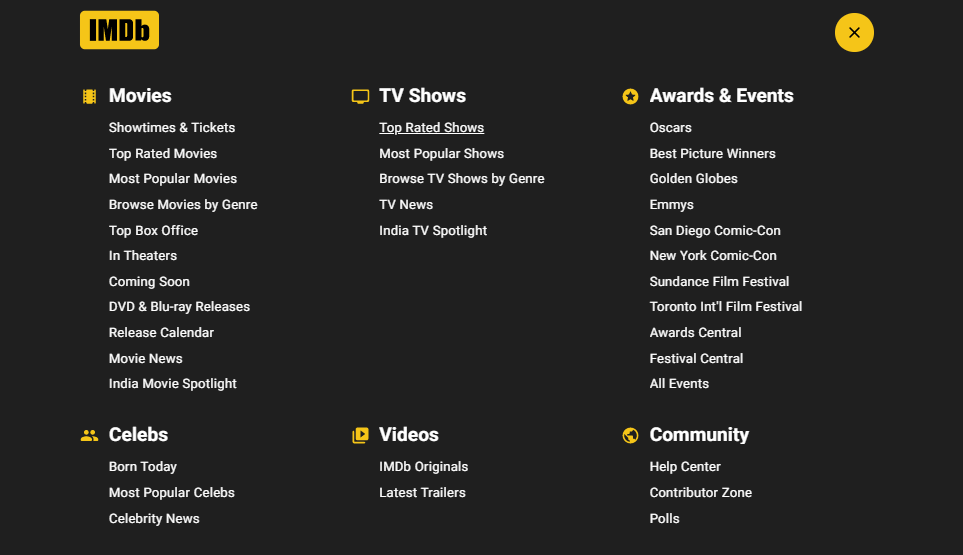


Рисунок 1.1 – Интерфейс сайта «IMDb»

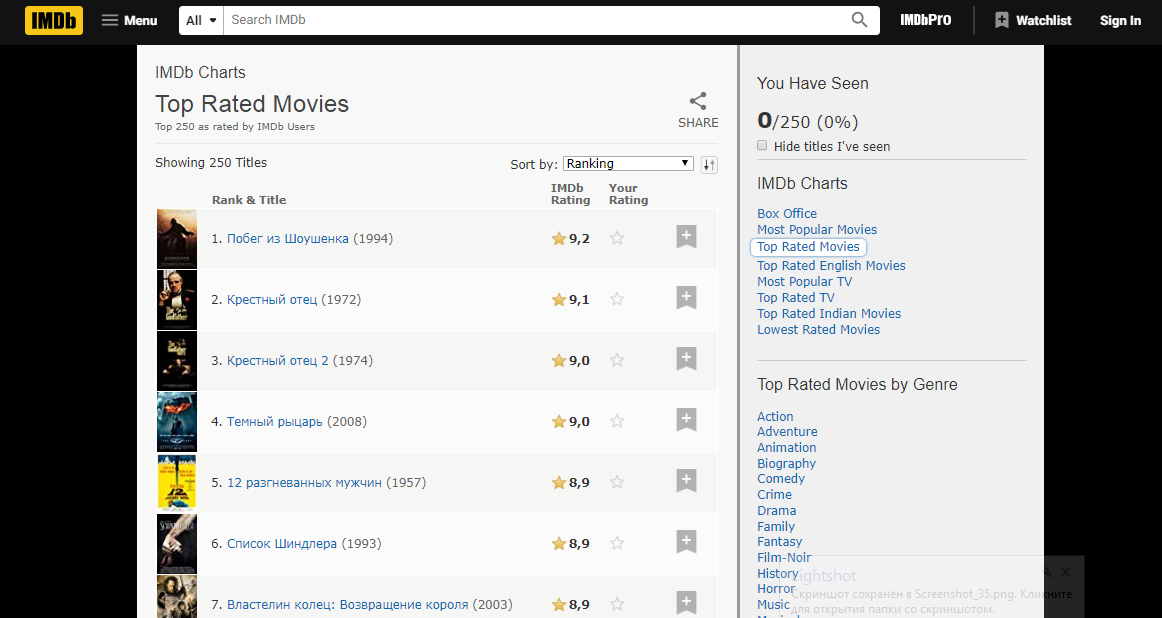


Рисунок 1.2 – Интерфейс сайта «IMDb»

*Кинопоиск* [2] – крупнейший русскоязычный интернет-сервис о кино. С 2018 года также доступен онлайн-кинотеатр с несколькими тысячами фильмов, сериалов, мультфильмов.

Плюсы:

* удобный и простой интерфейс;
* возможность оставлять оценки и писать рецензии;
* возможность смотреть фильмы онлайн;
* присутствует приложения для Android и iOS;
* возможность покупки билетов в кино через web и мобильные версии приложения;

Минусы:

* ограниченный функционал в бесплатной версии.

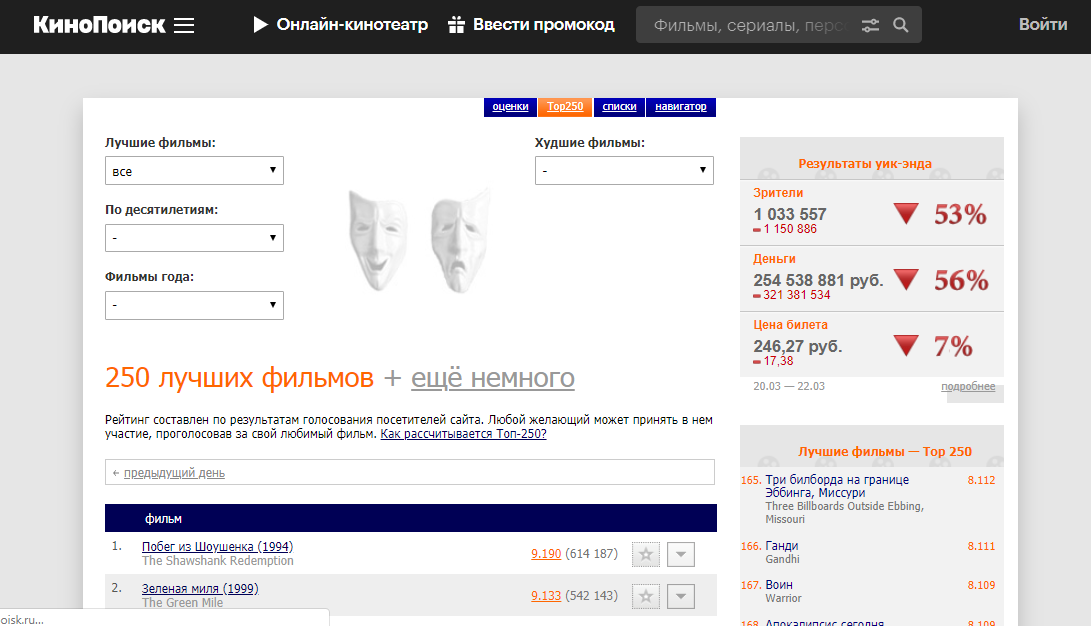


Рисунок 1.3 – Интерфейс сайта «КиноПоиск»

## **1.2 Постановка задачи**

Проанализировав прототипы приложений, я выделил основную задачу и функциональные требования проекта.

Основной задачей моего курсового проекта является разработка десктопного приложения, позволяющего пользователю просматривать списки фильмов, ставить оценки, писать комментарии. Вся информация должна храниться в централизованной базе данных. Программное средство должно содержать профиль авторизованного пользователя с его контактными данными. Интерфейс должен быть простым и удобным для использования.

Функциональные требования описаны в главе 2.

# **Анализ требований к программному средству и разработка функциональных требований**

Для того чтобы приступить к этапу разработки проекта, необходимо четко сформулировать функциональные требования к программному средству.

## **2.1 Определение требований к программному средству**

Функционально ПС должно выполнять следующие задачи:

* регистрация и авторизация пользователей;
* создание профиля пользователя;
* сохранение информации в базе данных;
* осуществление поиска фильмов по названию;
* разделение фильмов на категории;
* возможность ставить оценки и писать комментарии;
* выводить статистику о поставленных оценках;
* уведомление о выходе фильма в прокат.

## **2.2 Описание средств разработки**

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2017;
* программная платформа .NET Framework 4.7;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* технология Entity Framework;
* SQL Server Express LocalDB;
* концепции визуального оформления Material Design;
* библиотека для построения графиков OxyPlot.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс — Windows Presentation Foundation (WPF) [5], предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом. В основе графической технологии WPF лежит мощная инфраструктура, основанная на DirectX. Это является одним из основных отличий WPF от более ранней технологии создания пользовательских интерфейсов — Windows Forms.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом — С#, разработанный для создания приложений на платформе Microsoft .NET Framework [6].

Чтобы осуществлять связь между базой данных и приложением на C# необходим посредник. И именно таким посредником является технология Entity Framework [7]. Она предоставляет собой объектно-ориентированную технологию доступа к данным, является object-relational mapping (ORM) решением для платформы .NET Framework. Entity Framework предоставляет возможность взаимодействия с объектами посредством LINQ to Entities. Центральной концепцией Entity Framework является понятие сущности или entity. Сущность представляет набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

Для оформления стиля приложения использовалась концепция визуального оформления Material Design [8], которая позволяет разрабатывать удобные и приятные в использовании приложения. Material Design — стиль дизайна программного обеспечения и приложений, разработанный компанией Google для унификации интерфейсов всех ее продуктов и сервисов.

Построения графиков происходило с использованием кроссплатформенной библиотеки OxyPlot [9]. Библиотека основана на .NET и имеет открытый исходный код. Ее цель состояла в том, чтобы предоставить графическую модель, которая проста в использовании и открыта для расширения, с возможностью изменения внешнего вида графиков путем изменения простых свойств.

## **2.3 Описание функциональности ПС**

Описание функциональности программного средства представлено с помощью UML-диаграммы вариантов использования и диаграммы последовательности (Приложение А).

Спецификация функциональности программного средства представлена в следующем пункте.

## **2.4 Спецификация функциональных требований**

Для функциональности ПС необходимо создание базы данных для хранения информации приложения. Подробно о базе данных описано в главе 3.

В программном средстве «MovieDB» при запуске необходимо реализовать регистрацию и авторизацию пользователей для дальнейшего использования приложения. Для авторизации входными параметрами являются логин и пароль пользователя, которые содержатся в базе данных. Для того чтобы зарегистрироваться необходимо ввести логин, ФИО, email и пароль. Введенные данные, успешно прошедшие валидацию, заносятся в базу данных, а пользователь получает доступ к приложению.

Данные о пользователе включаются в профиль пользователя. В профиле можно изменить данные, загрузить фото, а также заполнить пустые поля соответствующей информацией. Все данные о пользователе должны автоматически обновляться в базе данных.

В профиле пользователя также должны находиться настройки приложения: смена языка интерфейса и смена темы приложения.

Пользователь может просмотреть список фильмов, распределённых по двум категориям, просмотреть подробную информацию о каждом фильме. Написать комментарий, поставить оценку, отметить просмотренные и любимые фильмы. После добавления комментария, пользователь может его редактировать.

Все данные должны заноситься в базу данных в соответствующие таблицы.

Необходимо реализовать возможность сортировки фильмов по категориям, статистики о поставленных оценках в виде графика, уведомление о выходе фильма в прокат, а также поиск фильмов по названию.

Кнопка для выхода из приложения должна возвращать пользователя к окну авторизации.

# **Проектирование и создание программного средства**

## **3.1 Модель базы данных**

Для реализации поставленной задачи была создана локальная база данных MovieDataBase. База данных состоит из 3 таблиц, перечисленных ниже.

На рисунке 3.1 представлена структура таблицы «Users», которая содержит информацию о пользователе: логин, пароль, ФИО и контактные данные.

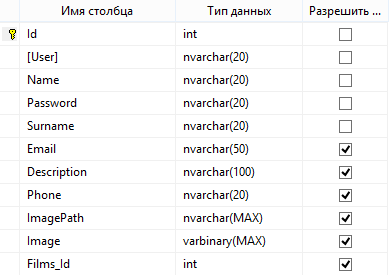


Рисунок 3.1 – Структура таблицы «Users»

Сведения о фильмах можно посмотреть в таблице Films, которая представлена на рисунке 3.2. Таблица содержит полную информацию о фильме.

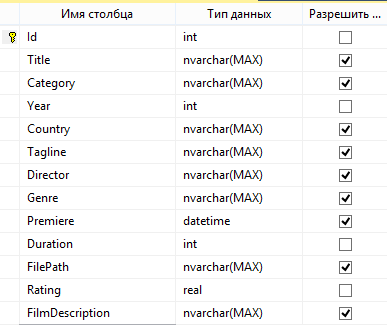


Рисунок 3.2 – Структура таблицы «Films»

На рисунке 3.3 представлена структура таблицы для хранения дополнительной информации о фильмах от пользователей.

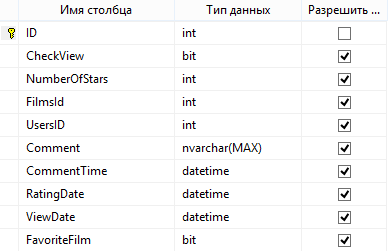


Рисунок 3.3 – Структура таблицы «Attachments»

Взаимосвязь таблиц базы данных представлена в Приложении Б.

Таблица Attachments имеет зависимости в виде внешних ключей FilmsId и UserdId, которые сопоставляются с первичными ключами таблиц Films и Users соответственно.

Таблицы соединены связями «один ко многим», т. е. несколько строк из дочерней таблицы зависят от одной строки в родительской таблице.

## **3.2 Структура проекта и выполняемые функции**

Структура проекта представлена на рисунке 3.5.

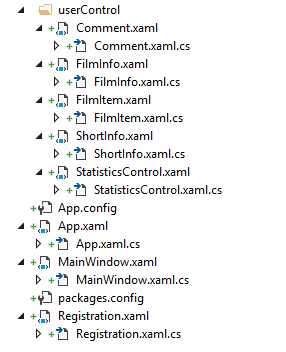
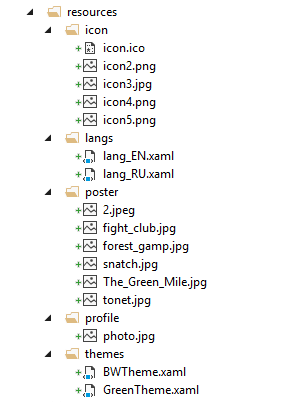
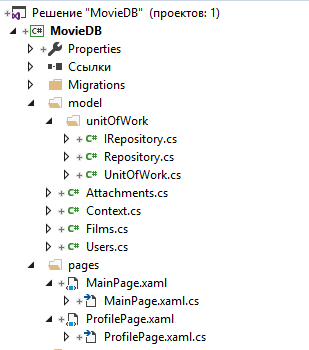


Рисунок 3.5 – Структура проекта

Данное приложение предоставляет доступ к базе данных с фильмами и выполняет следующие функции.

* регистрация и авторизация пользователя;
* просмотр списков фильмов по категориям и дополнительной информации о фильмах;
* уведомление о выходе фильма в прокат и поиск;
* отображение статистики;
* настройка параметров работы приложения.

### **3.2.1 Регистрация и авторизация**

Регистрация и авторизация пользователя происходит в окне Registration, имеющем следующую структуру, отображенную на рисунке 3.6.



Рисунок 3.6 – Структура класса Registration

Класс Registration содержит обработчик события Register\_Click, который является главным в процессе регистрации, и обработчик события LogIn\_Click – в авторизации (Приложение В).

При нажатии на кнопку регистрации происходит проверка всех строк для ввода на пустоту и корректность вводимых данных с помощью методов CheckEmail(), CheckNames(), CheckPassword(). Если вводимые данные корректны и пароли совпадают, то создается новый пользователь и происходит вход в приложение. Если вводимые данные не являются корректными, то строки изменяют цвет на красный и добавляется подсказка с правилами ввода к каждой некорректной строке.

При авторизации идет проверка введенных значений на пустоту и корректность данных, если проверка пройдена, то происходит поиск на соответствие введенных значений со значениями в базе данных и вход в приложение, иначе система уведомляет об ошибках.

### **3.2.2** **Просмотр списка фильмов**

Добавление фильмов происходит при загрузке приложения.

На рисунке 3.7 представлена структура класса MainPage:

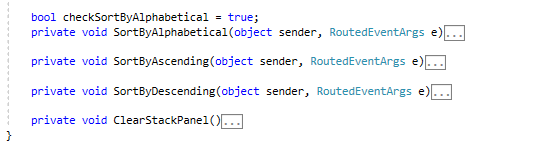
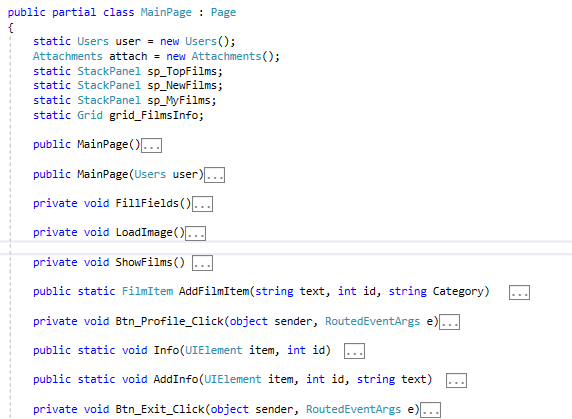


Рисунок 3.7 – Структура класса MainPage

Вывод фильмов происходит в методе AddFilmItem(…). Создается объект класса FilmItem и в зависимости от категории добавляется на страницу.

### **3.2.3** **Уведомление о выходе в прокат и поиск**

Уведомление о выходе фильма в прокат реализовано в классе MainWindow в методе AddAlert(), поиск фильмов по названию осуществляется в обработчике события Txt\_Search\_TextChanged.

Структура класса MainWindow показана на рисунке 3.8.

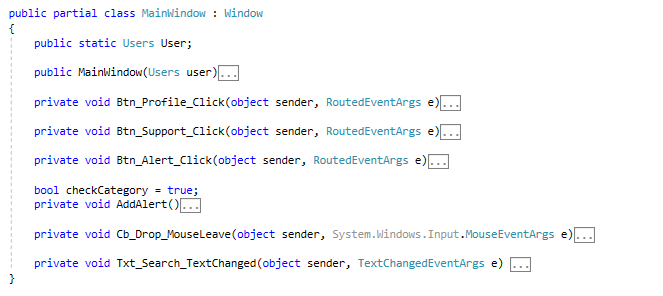


Рисунок 3.8 – Структура класса MainWindow

Листинги главных функций представлены в Приложении В.

Уведомление о выходе новых фильмов происходит путём сравнения текущей даты и даты выхода. Если дата выхода меньше, то появится уведомление о выходе фильма и обновится счётчик.

Поиск фильмов происходит при каждом изменении вводимой строки с помощью регулярного выражения.

### **3.2.4** **Отображение статистики**

График для отображения статистики реализован с помощью пользовательского элемента управления StatisticsControl (рисунок 3.9).

Добавление статистического графика происходит в конструкторе класса ProfilePage (рисунок 3.10)

Отображение статистики осуществляется с помощью подключения библиотеки OxyPlot для построения графика по поставленным оценкам.

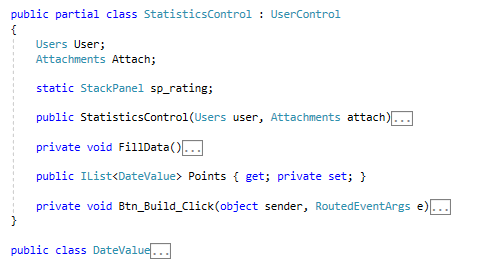


Рисунок 3.9 – Структура класса StatisticsControl

В обработчике события Btn\_Build\_Click(…) происходит считывание выбранного диапазона дат и проверка на правильность диапазона. Координатами графика являются дата и оценка фильма в этот день.

### **3.2.5 Настройка параметров работы с приложением**

Настройки работы с приложением реализуются в классе ProfilePage и представлены на рисунке 3.10.

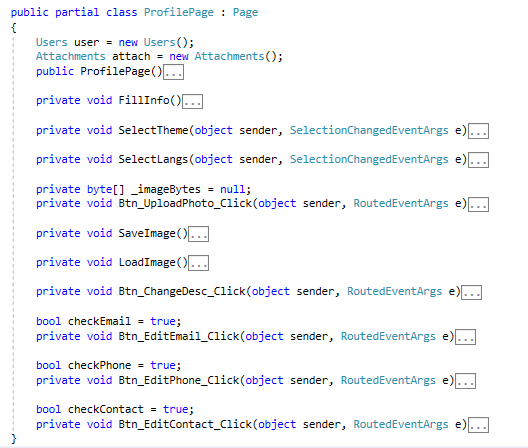


Рисунок 3.10 – Структура класса ProfilePage

## **3.3 Взаимоотношения между классами**

Взаимоотношения между классами можно представить в виде диаграммы классов (Приложение Г).

Классы друг с другом связаны ассоциативными связями. Ассоциация показывает, что объекты одной сущности (класса) связаны с объектами другой сущности таким образом, что можно перемещаться от объектов одного класса к другому.

Так как в проекте используется технология Entity Framework, классы Users, Films, Attachments сопоставляются с аналогичными таблицами базы данных.

Слой доступа к данным представлен паттернами Репозиторий и Unit of Work.

Паттерн Репозиторий позволяет абстрагироваться от конкретных подключений к источникам данных, с которыми работает программа, и является промежуточным звеном между классами, непосредственно взаимодействующими с данными, и остальной программой. Реализацией паттерна является универсальный интерфейс IRepository<TEntity> и универсальный класс Repository<TEntity>.

Паттерн Unit of Work служит для упрощения работы с множеством репозиториев и гарантирует, что все репозитории будут использовать один и тот же контекст данных.

Программное средство «MovieDB» является страничным приложением. Главным окном является MainWindow, главной страницей MainPage и страницей профиля ProfilePage. Окно Registrarion применяется для авторизации и регистрации пользователей.

Для представления фильмов, уведомлений, комментарий, графика статистики и дополнительной информации служат пользовательские элементы управления FilmItem, ShortInfo, Comment, StatisticsControl, FilmInfo.

# **4. Тестирование**

## **4.1 Тестирование авторизации и регистрации**

В момент регистрации возможна ситуация, когда пользователь вводит некорректные данные. Обработка таких данных происходит с помощью добавления подсказок к строкам ввода и сменой цвета введенных данных. Примеры обработки исключительных ситуаций в окне регистрации представлены на рисунках 4.1 – 4.4.

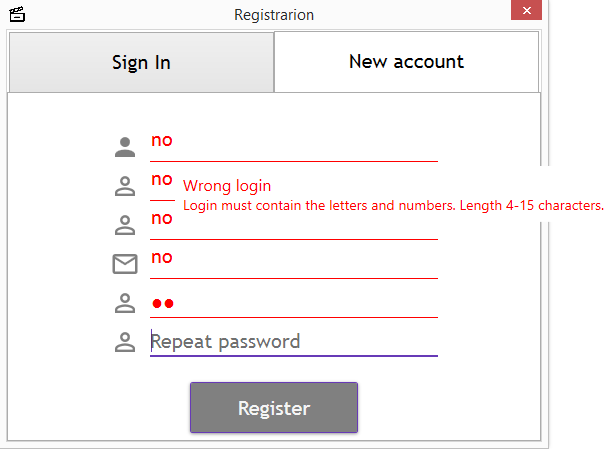


Рисунок 4.1 – Обработка некорректного логина

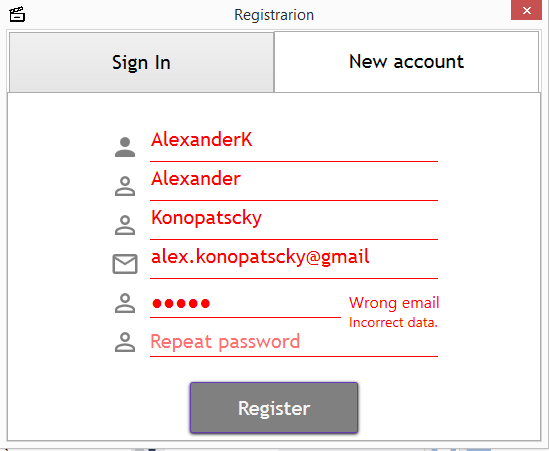


Рисунок 4.2 – Обработка некорректного email

Обработка некорректного пароля представлена на рисунке 4.3.

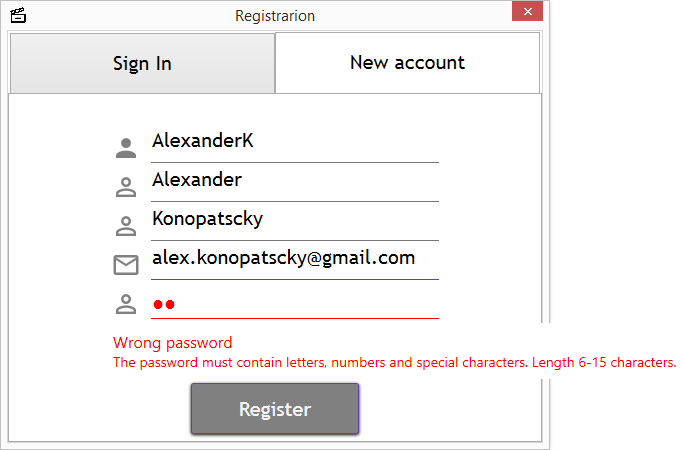


Рисунок 4.3 – Обработка некорректного пароля

Для входа в приложение необходимо, чтобы повторное введение пароля совпадало с первоначальным. Обработка такой ситуации представлена на рисунке 4.4.

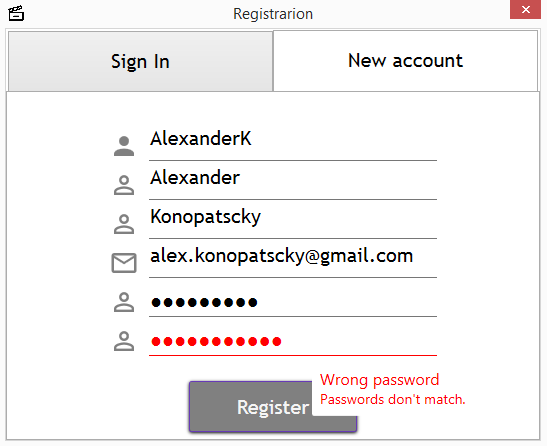


Рисунок 4.4 – Обработка несовпадающих паролей

Обработка пустых значений в окне авторизации также обрабатывается путем изменения цвета строк и добавлением подсказок при наведении на соответствующую строку. Пример обработки такого исключения представлен на рисунке 4.5.

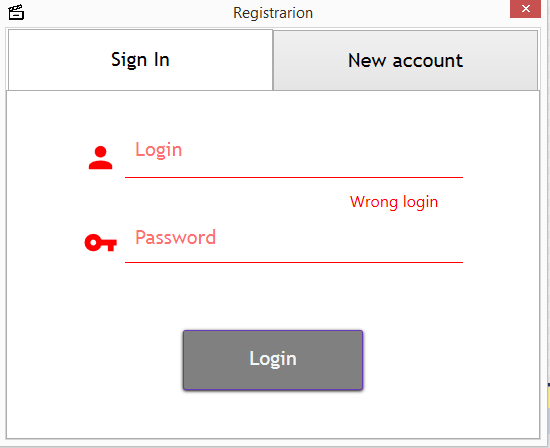


Рисунок 4.5 – Обработка ввода пустых данных

При авторизации возможна ситуация, когда пользователь вводит неверные данные пользователя, который ещё не зарегистрирован. Обработка данного исключения продемонстрирована на рисунке 4.6.

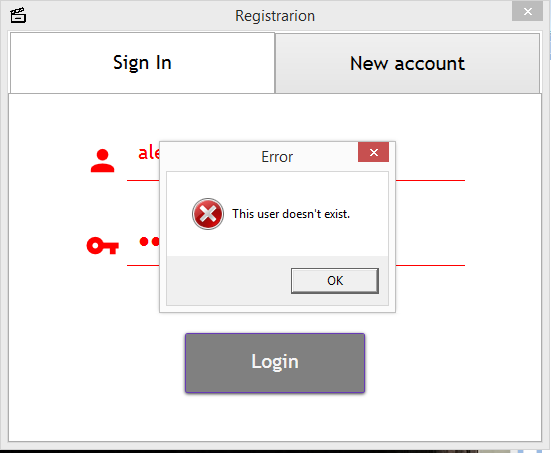


Рисунок 4.6 – Тестирование авторизации

## **4.2 Тестирование функций приложения**

Если мы попытаемся добавить ещё один комментарий, система этого не допустит. Обработка данного исключения продемонстрирована на рисунке 4.7.

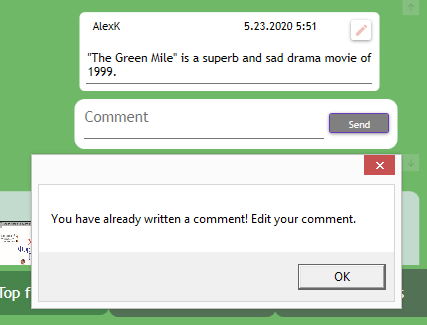


Рисунок 4.7 – Тестирование добавления второго комментария

Если при построении статистического графика мы выберем неправильный диапазон дат, то ошибка будет обработана следующим образом:

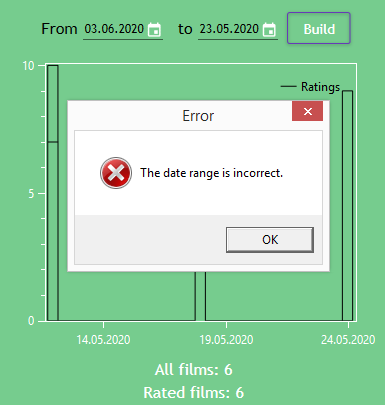


Рисунок 4.8 – Тестирование диапазона дат при построении графика

# **5. Руководство по использованию**

## **5.1 Руководство по установке**

В решении проекта есть папка SetupMovieDB, которая предназначена для установки программного средства «MovieDB». С помощью Debug\SetupMovieDB.msi необходимо установить приложение в выбранную вами папку. Для запуска приложения необходимо найти папку, куда вы установили программное средство, и запустить MovieDB.exe.

## **5.2 Руководство по использованию**

После запуска приложения пользователь должен зарегистрироваться или войти под зарегистрированной ранее учетной записью для дальнейшего использования ПС.

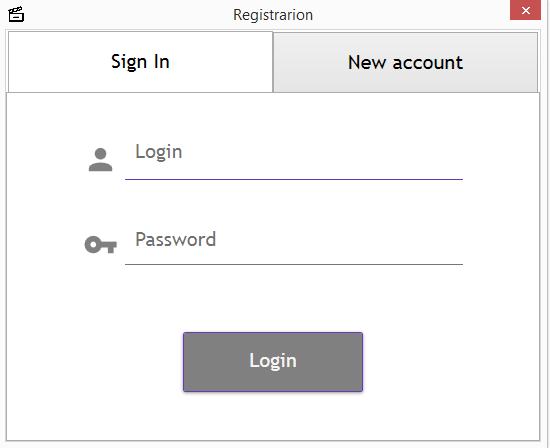


Рисунок 5.1 – Окно авторизации

При регистрации указывается имя и фамилия пользователя, адрес электронной почты и пароль.

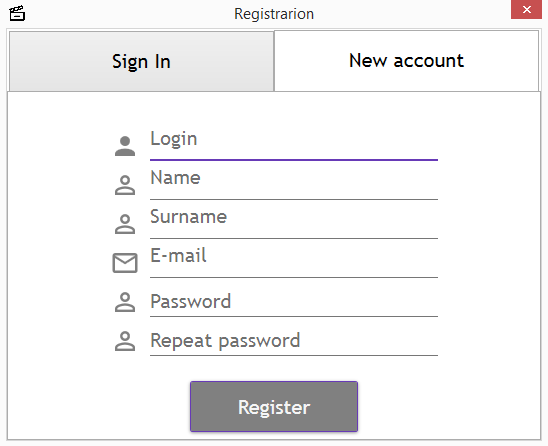


Рисунок 5.2 – Окно регистрации

При успешном входе в аккаунт появляется главная страница, в которой пользователь видит панель профиля, список фильмов, строку поиска и кнопку уведомлений. Данная страница представлена на рисунке 5.3.

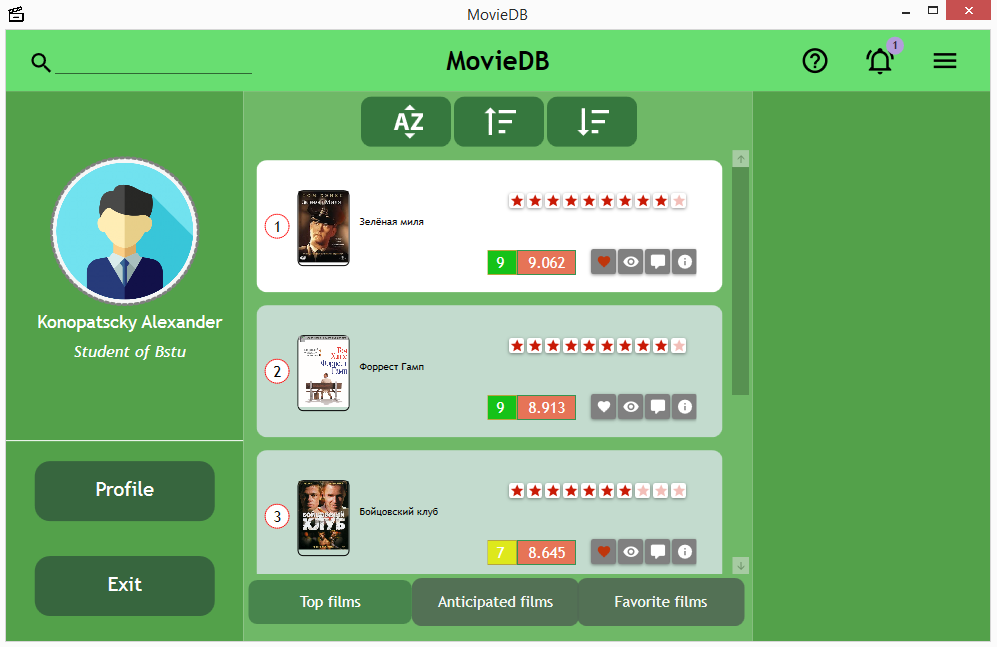


Рисунок 5.3 – Главная страница

Пользователь может посмотреть дополнительную информацию о фильме, нажав на специальную кнопку (рисунок 5.4).

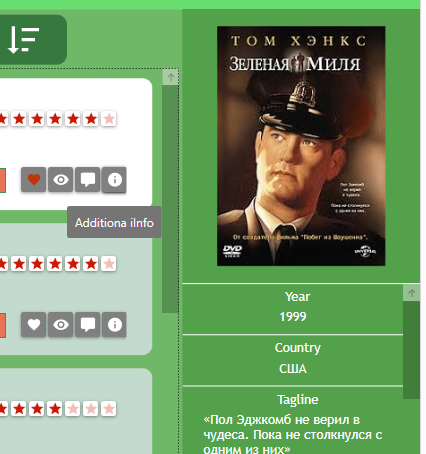


Рисунок 5.4 – Просмотр дополнительной информации

Для добавления комментарий необходимо нажать на соответствующую кнопку на панели, появится выпадающее меню. Затем нужно написать комментарий и нажать кнопку отправить (рисунок 5.5).

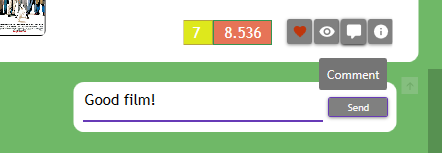


Рисунок 5.5 – Добавление комментария

Для редактирования комментария необходимо нажать на соответствующую кнопку (рисунок 5.6). После завершения редактирования – нажать второй раз, так изменения будут сохранены.

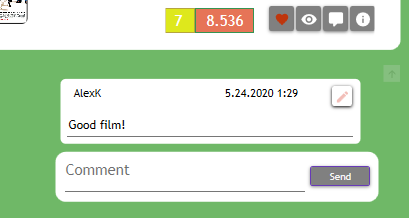


Рисунок 5.6 – Изменение комментария

Для добавления фильма в категорию избранное, нужно нажать на соответствующую кнопку (рисунок 5.7).

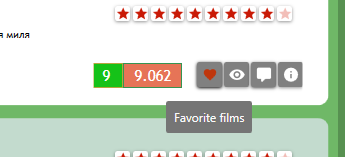


Рисунок 5.7 – Добавление в категорию избранное

Чтобы отметить просмотренный фильм, нужно нажать на соответствующую кнопку (рисунок 5.8). После этого изменится фон элемента.

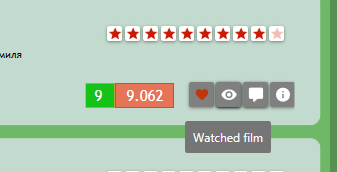


Рисунок 5.8 – Отметить просмотренный фильм

Чтобы поставить оценку, нужно нажать на соответствующую звезду (рисунок 5.9).

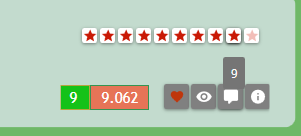


Рисунок 5.9 – Поставить оценку

Чтобы перейти на профиль пользователя с его личной информацией, необходимо нажать кнопку “Profile” на панели профиля.

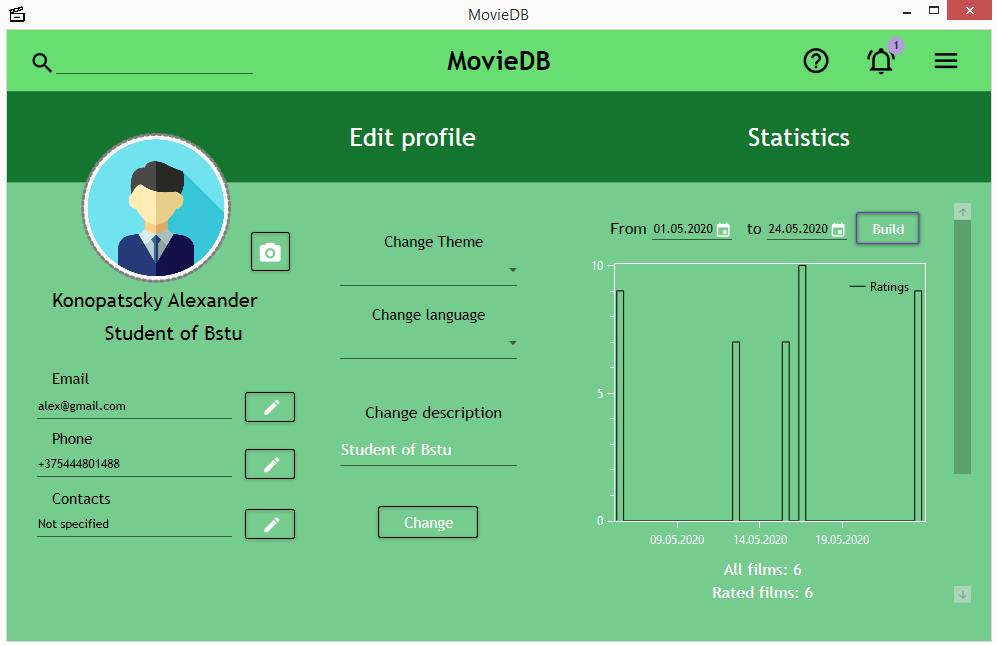


Рисунок 5.10 – Страница профиля

В профиле содержится информация, введенная при регистрации, а также есть возможность добавить дополнительные данные или изменить существующие. Для загрузки или обновления фото профиля необходимо нажать на соответствующую кнопку.

На странице профиля можно изменить тему и язык интерфейса ПС.

Смена темы представляет собой смену цветовой гаммы ПС. По умолчанию в приложении установлена тема Green.

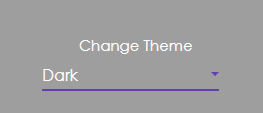


Рисунок 5.11 – Смена темы

Программное средство поддерживает 2 языка интерфейса: русский и английский. По умолчанию интерфейс на английском языке.

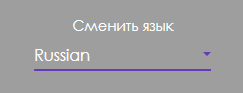


Рисунок 5.12 – Смена языка

Также страница профиля содержит статистический график, который по умолчанию показывает количество поставленный оценок за неделю. Диапазон можно изменить, выбрав даты и нажав двойным кликом на кнопку для построения.

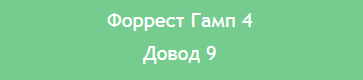
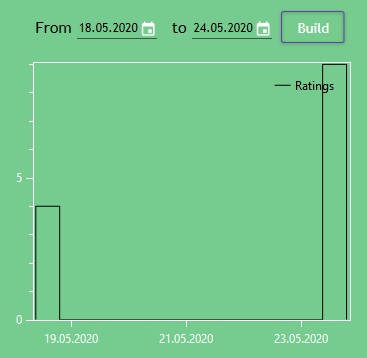


Рисунок 5.13 – Статистический график

# **Заключение**

В ходе выполнения курсовой работы было разработано программное средство «MovieDB» на языке C# с использованием технологий Entity Framework, WPF, концепции визуального оформления Material Design и библиотеки для построения графиков OxyPlot.

При разработке программного средства были выполнены все пункты из указанного списка предполагаемого основного функционала приложения, а именно:

* составлена база данных;
* сделано разделение фильмов по двум основным спискам;
* реализована возможность ставить оценки и писать комментарии;
* есть возможность получать уведомление о выходе фильма в прокат;

Данное программное средство имеет удобный и понятный интерфейс. Результаты проведенного тестирования показали, что приложение обрабатывает все исключительные ситуации.

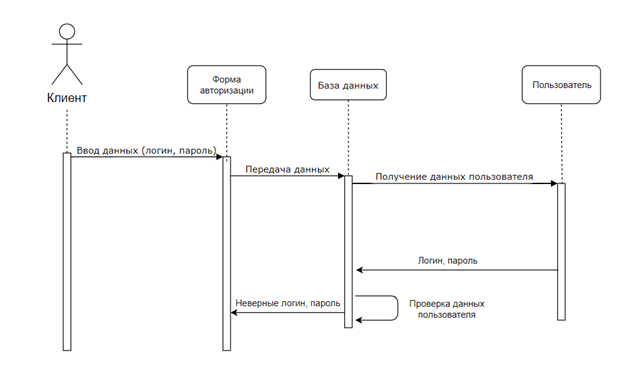
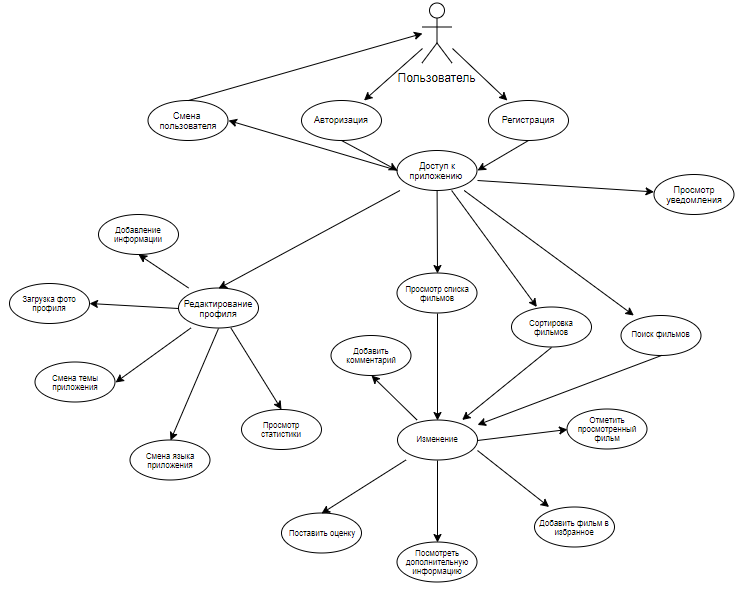
Инструкция по установке программного средства «MovieDB» и руководства пользователя приведены.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

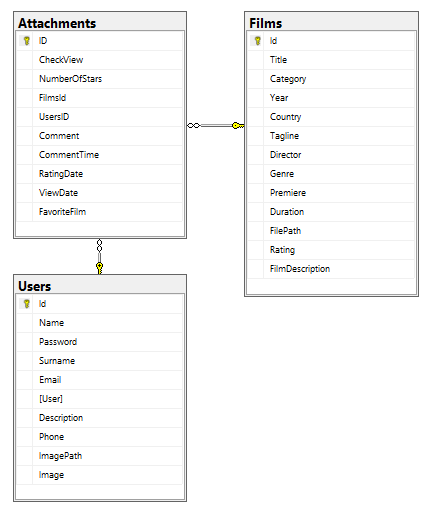
# **Список литературных источников**

1. Программное средство IMDb [Электронный ресурс] – Режим доступа: <https://www.imdb.com/>
2. Программное средство Кинопоиск [Электронный ресурс] – Режим доступа: <https://www.kinopoisk.ru/>
3. Мэтью Макдональд — WPF: Windows Presentation Foundation в .NET 4.0 с примерами на C#
4. Язык C# и .NET Framework [Электронный ресурс] — Режим доступа: <http://professorweb.ru/my/csharp/charp_theory/level1/infonet.php>
5. Руководство по ADO.NET Entity Framework 6.0 [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/entityframework/>
6. OxyPlot Documentation [Электронный ресурс] – Режим доступа: <https://www.oxyplot.org/>
7. Material Design [Электронный ресурс] – Режим доступа: <https://material.io/design/>

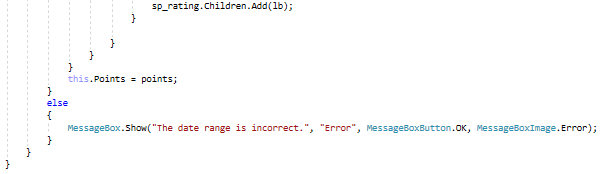
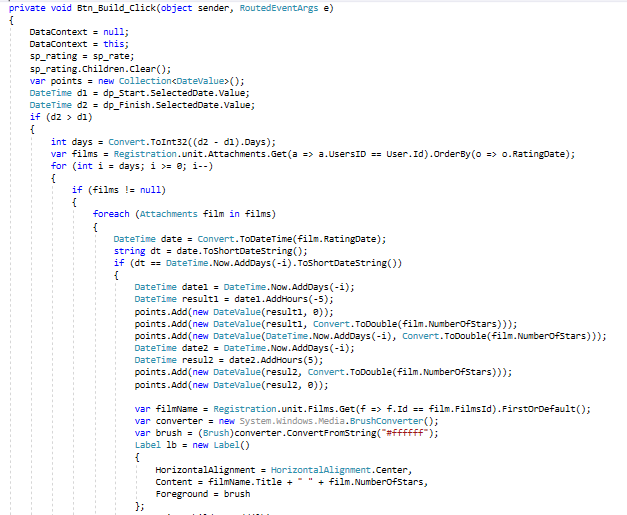
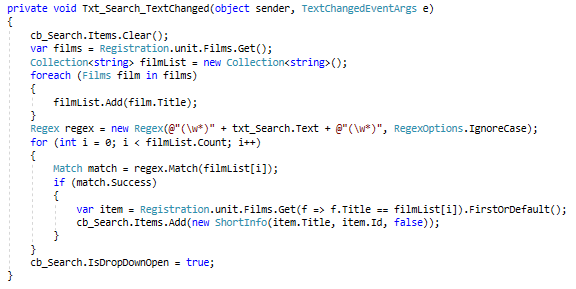
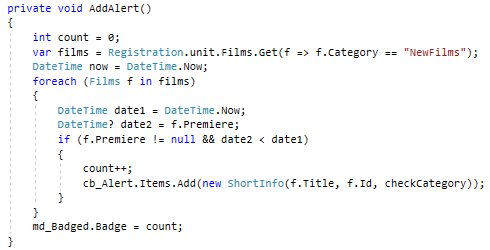
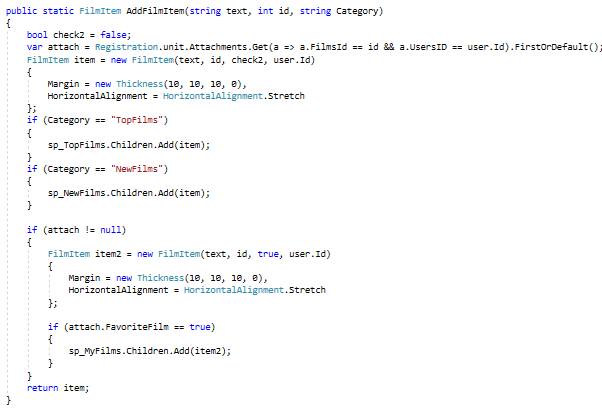
# **Приложение А**



# **Приложение Б**



# **Приложение В**



# **Приложение Г**

