

## **АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ ПРОЦЕССОМ УЧЁТА МЕДИАФАЙЛОВ «МЕДИАТЕКА»**

**О.В. Рычка, А.С. Полятыкин**

Донецкий национальный технический университет

кафедра программной инженерии,

E-mail: axpltkn@yandex.ru

*Рычка О.В., Полятыкин А.С. Автоматизация управления процессом учёта медиафайлов «Медиаотека». Работа посвящена разработке клиент-серверной системы ведения учёта медиафайлов с возможностью оценки и комментирования. Целью работы является проектирование и разработка данной системы.*

*Rychka O.V., Polyatykin A.S. Automation of management of the process of accounting for media files "Mediatek". The work is devoted to the development of a client-server system for keeping records of media files with the ability to evaluate and comment. The aim of the work is to design and develop this system.*

### **Общая постановка проблемы**

Ведение учёта медиафайлов с возможностью оценивания широким кругом пользователей позволяет проследить динамику изменения популярности медиафайлов разного рода и ответить на такие вопросы как «какой музыкальный жанр самый популярный?», «какой художник написал наибольшее количество картин?», «в скольких фильмах сыграл тот или иной актёр?» т.д. Автоматизация учёта этого процесса с сопровождением хорошо спроектированной базы данных повышает эффективность учёта, в том числе благодаря исключению непреднамеренных и преднамеренных внесений неправильных данных людьми, уменьшению объема записей благодаря структуре реляционной базы данных и т.д. Также можно выделить такие преимущества автоматизации учёта как: перенесение части работы на машины, что позволяет работать удаленно, и это позволяет получать в любой момент времени оперативные данные с сервера.

В границах темы данной работы пользователи получают информацию о фильмах, музыке, картинах/фотографиях, текстах/книгах, каждый из них может прокомментировать любой из медиафайлов и поставить оценку от 1 до 5, таким образом на основании всех оценок пользователей система выведет среднюю оценку и постепенно можно будет сделать вывод о популярности каждого медиа-элемента. Естественно, если есть база данных, к которой есть возможность доступа у большого числа людей, то настоятельно рекомендуется организовать разграничение прав доступа для каждой из ролей, коих в данной системе шесть, о которых будет рассказано ниже.

Для уменьшения требований со стороны клиентских машин и повышения безопасности большинство логических операций над данными выполняется на сервере с помощью языков SQL и PL/pgSQL.

### **Исследования**

#### **1. Обзор существующих систем учёта и оценивания мультимедиа**

На сегодняшний день существует множество систем учёта мультимедиа. Но в большинстве своём они охватывают только один тип медиаданных. Чаще всего это фильмы, реже – музыка, а картины, фото, книги встречаются ещё реже.

Зачастую это сайты, которые помимо информации позволяют посмотреть фильм или прослушать музыку.

KinoGo[1] – популярный сайт для просмотра фильмов. Здесь собраны разные фильмы, приведено и описание, оценки, комментарии. Поиск фильмов очень гибкий благодаря системе фильтров по широкому спектру критериев.

Крупнейший в мире портал Internet Movie Database (IMDB)[2], которой владеет компания Amazon.com предоставляет функционал по теме данной работы. Здесь собраны данные о нескольких миллионах разных кинопродуктов. Пользователи могут оценивать фильмы, на чём строятся рейтинги, которые позволят пользователям быстро найти интересующие их фильмы или сериалы.

Muzter[3] – сервис, который составляет рейтинг музыкальных произведений. Здесь нет возможности комментирования произведений, а рейтинг строится не на основании оценок, а на количестве прослушиваний отдельных песен. Есть информация об альбоме песен, дате релиза и жанрах.

35photo[4] – портал, где фотографы выкладывают свои работы для оценивания пользователями. Как и в большинстве подобных порталов – здесь есть возможность комментирования фотографий и поиск по жанрам.

Книгопоиск[5] – рейтинговый портал книг, который позволяет пользователям оценивать книги и искать их по жанрам и другим критериям.

## **2. Общие требования к системе**

Разрабатываемая система рассчитана на большое количество пользователей для комментирования, обсуждения и оценивания фильмов, музыки, книг и картин/фотографий. Должны быть распределены роли в системе, такие как администратор, модератор, контент-менеджер и простой пользователь.

Система должна быть спроектирована на основе клиент-серверной архитектуры. Серверная часть может быть распределённой или централизованной и представлять собой СУБД. Модель клиента должна соответствовать «тонкому клиенту», а большую часть вычислений должен брать на себя сервер. Учитывая модель тонкого клиента, клиентское приложение должно иметь небольшой размер, небольшие системные требования.

Так как система предполагает удалённое взаимодействие клиента и сервера – необходимо интернет соединение, а для меньшей нагрузки на сеть, количество передаваемой информации должно быть уменьшено до необходимого минимума.

Предупреждение исключительных ситуаций настоятельно рекомендуется для повышения устойчивости к сбоям и надёжности системы.

Учитывая вышеперечисленные требования, программный продукт должен соответствовать следующим требованиям:

- система должна иметь клиент-серверную архитектуру;
- сервер представляет собой СУБД;
- клиентская часть ПО должна легко устанавливаться на ПК или вообще быть переносной и не требовать установки;
- клиентская часть должна иметь доступ к интернету для связи с сервером;
- должно быть минимизировано количество информации, передаваемое от сервера к клиенту и наоборот;
- желательны обработки исключительных ситуаций;
- серверная часть должна брать на себя большую часть вычислительных операций.

## **3. Общая структура приложения**

В основе работы системы будет лежать так называемая модель взаимодействия клиент-сервер, которая позволяет разделять функционал и вычислительную нагрузку между клиентскими приложениями и серверным приложением. Между собой клиент и сервер взаимодействуют в сети Интернет или в любой другой компьютерной сети при помощи

различных сетевых протоколов.

Данная система имеет двухзвенную архитектуру, то есть клиент и сервер напрямую между собой взаимодействуют, без посредников.

В роли серверной части будет выступать СУБД PostgreSQL, а в роли клиентских приложений – приложения, написанные на языке C# с применением технологии WPF.

Такая архитектура позволяет хранить и использовать сложные связи между таблицами данных в БД, предоставляя пользователю просто для понимания интерфейс.

В качестве СУБД для реализации серверной части был выбран PostgreSQL[7]. Данная СУБД была выбрана по причине широких возможностей для реализации решения различных задач, не только выборки данных, но и других процедур над данными. PostgreSQL имеет набор инструментов, который расширяет стандарты SQL. Реляционная модель данных поддерживает декларативные ограничения целостности уровня домена (типа данных), уровня отношения и уровня базы данных, а также поддерживает операторы манипулирования отношениями. PostgreSQL предоставляет множество различных возможностей, достаточно надежна и имеет хорошие характеристики по производительности. Данная СУБД выгодно отличается среди многих других. Она обладает практически всеми возможностями, которые есть в других базах данных (коммерческих или Open Source), а также некоторыми дополнительными.

PostgreSQL обеспечивает ссылочную целостность сам, без необходимости реализации этого вручную, т.е. если есть запись в таблице, которая должна ссылаться на запись из другой таблицы, то программисту не нужно явно проверять наличие записи, на которую пытается сослаться первая запись – СУБД сама проверит и предпримет соответствующие действия, которые уже можно настроить под свои нужды.

Немаловажной особенностью этой СУБД является поддержка некоторых языков программирования, которые могут не только облегчить, но и вообще дать возможность к реализации некоторых функций. Например, поддержка PL/pgSQL позволяет объявлять переменные в функциях и использовать условные операторы if...else.

Выбор языка и технологий для написания клиентской стороны обоснован гибкостью настроек пользовательского интерфейса и наличием эффективных средств для реализации связи с СУБД PostgreSQL. Взаимосвязь клиентских приложений и базы данных будет осуществлено с помощью сторонней библиотеки Npgsql[8].

Используемый в WPF язык разметки XAML, являющийся подмножеством XML позволяет использовать создавать декларативный пользовательский интерфейс. Хотя весь пользовательский интерфейс можно описать на языке C# (или другом языке .Net), технология WPF использует другой подход, следуя которому интерфейс описывается на языке XAML, а поведение программы записывается в коде. Это позволяет более эффективно разделять логику и интерфейс приложения что очень положительно сказывается на модифицируемости программы и возможности разделения её на модули.

WPF берет на себя компоновку элементов интерфейса подстраивая его под разные разрешения. Это происходит благодаря тому, что элементы интерфейса могут привязываться не к размерам в пикселях, а к относительному размеру экрана.

Ну и наконец инструменты привязки данных в WPF приложениях имеют широкие возможности, что существенно уменьшает количество строк кода и упрощает его анализ.

## **Выводы**

Система позволяет содержать базу данных медиа-элементов, комментировать и оценивать большому количеству людей, и на основе этих оценок выстраивать рейтинги. При современной популярности и разнообразии фильмов, музыки, графических данных и книг такая существует множество подобных популярных систем.

Перед разработкой данной программной системы был проведён анализ предметной

области и подтверждена целесообразность такой системы.

Созданная программная система полностью соответствует, выдвинутым на этапе проектирования, требованиям.

В перспективе в систему может быть добавлена дополнительная информация, например, файлы-изображения, являющиеся логотипами фильмов, или изображения обложек альбомов, это может положительно повлиять на популярность системы, так как разнообразие в графическом интерфейсе поддерживает интерес пользователей и легче воспринимается глазом. Также может быть реализована возможность редактирования комментариев и изменение оценки медиа-элемента.

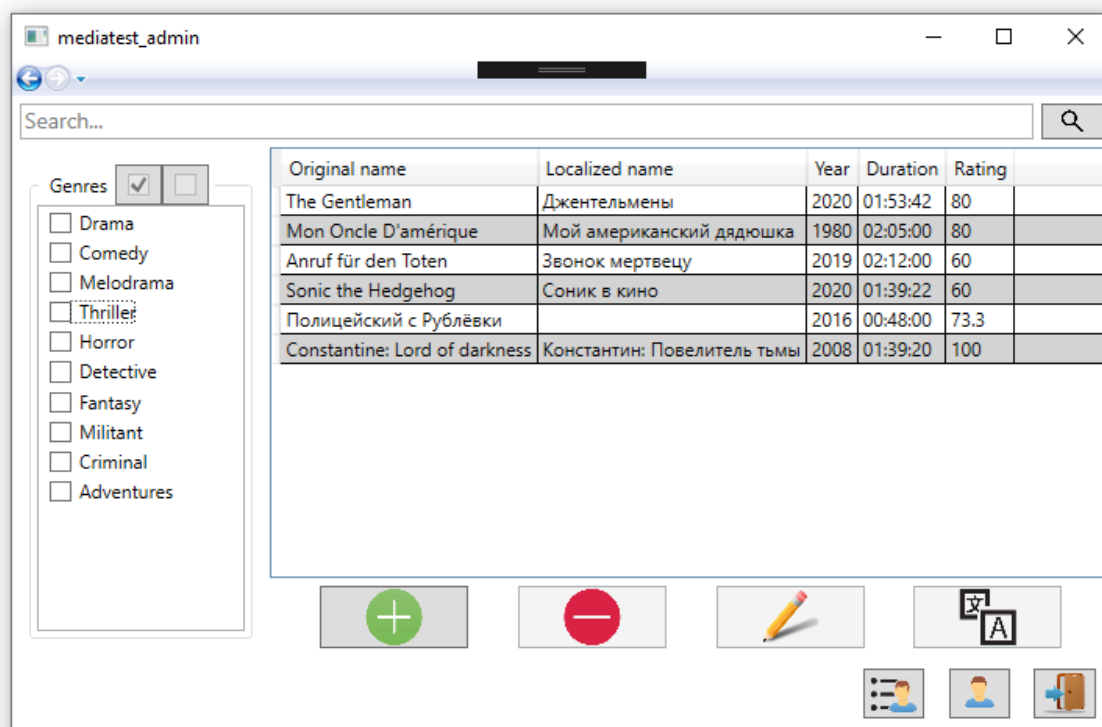


Рисунок 1 – Страница просмотра фильмов

### Литература

1. KinoGo – URL: <https://kinogo.by/> (дата обращения: 25.04.2020).
2. IMDB – URL: <https://www.imdb.com/> (дата обращения: 25.04.2020).
3. Muzter – URL: <https://muzter.net/> (дата обращения: 25.04.2020).
4. 35photo – URL: <https://ru.35photo.pro/> (дата обращения: 25.04.2020).
5. Knigopoisk – URL: <https://knigopoisk.org/> (дата обращения: 25.04.2020).
6. Описание структуры клиент-серверной архитектуры – URL: [https://studopedia.ru/3\\_194267\\_arhitektura-klient--server.html](https://studopedia.ru/3_194267_arhitektura-klient--server.html) (дата обращения: 03.05.2020)
7. PostgreSQL – URL: <https://www.postgresql.org>. (дата обращения: 03.05.2020)
8. Документация к сторонней библиотеке Npgsql – URL: <https://www.npgsql.org> (дата обращения: 03.05.2020).
9. Документация PostgreSQL: Процедурный язык PL/pgSQL – URL: <https://postgrespro.ru/docs/postgresql/9.6/plpgsql-overview#plpgsql-advantages> (дата обращения: 03.05.2020).