

Р. О. Чолкан¹, Ю. Є. Кинаш¹, О. Я. Різник¹, О. С. Вітер², В. М. Мицишин²

Національний університет "Львівська політехніка",

¹ кафедра інформаційних технологій видавничої справи,

² кафедра комп'ютеризованих систем автоматки

РОЗРОБЛЕННЯ СИСТЕМИ ДЛЯ ПІДБОРУ ФІЛЬМІВ З ВИКОРИСТАННЯМ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ

<https://doi.org/10.23939/amm2020.01.087>

© Чолкан Р. О., Кинаш Ю. Є., Різник О. Я., Вітер О. С., Мицишин В. М., 2020

Розроблено систему для пошуку фільмів на основі їх параметрів. Використано мову програмування C# 7 на основі фреймворку WPF і технології Json, LinQ, .NET та XAML. Використано алгоритми сортування методом Шелла, швидке сортування та бінарного пошуку для пришвидшення опрацювання інформації. Використання розробленої системи сприятиме економії часу при підборі фільмів за певними критеріями.

Ключові слова: система пошуку фільмів, фільтрація на основі контенту, швидке сортування, сортування методом Шелла, бінарний пошук.

The system for searching movies based on their parameters has been developed. C # 7 programming language based on WPF framework and Json, LinQ, .NET and XAML technology is used. Shell and quick sorting algorithms and binary search are used to speed up information processing. Using the developed system will save time when selecting films according to certain criteria.

Keywords – movie search system, content-based filtering, quick sort, Shell sorting, binary search.

Постановка проблеми

Бажання використання власної бібліотеки для перегляду фільмів чи ігор часто супроводжується довгими пошуками. Для пришвидшення отримання інформації за параметрами пошуку, зазвичай, застосовуються спеціальні алгоритми. Для отримання повноцінної інформації про фільми та ігри слід оперувати їхніми рейтингами. Найбільш широко для пошуку фільмів використовують назву, жанр, задіяних акторів, рейтинг і сюжет. Основу стратегії таких програм становить ідея створення каталогу користувача з улюблених фільмів/серіалів/ігор. Здебільшого, програми функціонують на основі клієнт-серверної взаємодії, тобто з використанням засобів взаємодії з базою даних.

У таких програмах повинні бути реалізовані наступні можливості:

- додавання нового користувача;
- перевірка на персональних вхідних дених вже існуючого користувача;
- додавання улюблених фільмів/серіалів/ігор;
- створення та заповнення окремих бібліотек;
- редагування акаунта з можливістю змінити ім'я, пароль, фото тощо;
- налаштування зовнішнього вигляду програми;
- переглядати та редагувати інформацію про улюблені фільми/серіали/ігри;
- видаляти фільми/серіали/ігри та бібліотеки.

Отже, реалізація пошукової системи для власної бібліотеки фільмів чи ігор за перерахованими категоріями є актуальним завданням.

Аналіз літературних джерел

Підхід клієнт-серверної архітектури набув значної популярності завдяки потужному розвитку мережі Інтернет та зосередження значних обсягів інформації у базах даних, що розташовані на серверах.

Клієнт-серверна архітектура представляє собою концепцію інформаційної мережі, де основну частину ресурсів зосереджено на серверах, що обслуговують клієнтів. Такою архітектурою визначено наступні типи компонентів (рис. 1):

- набір серверів для надання інформаційних послуг програмам, котрі звертаються до них;
- набір клієнтів, котрі використовують доступні сервіси, що надаються серверами;
- мережа, що забезпечує взаємодію між клієнтами та серверами.

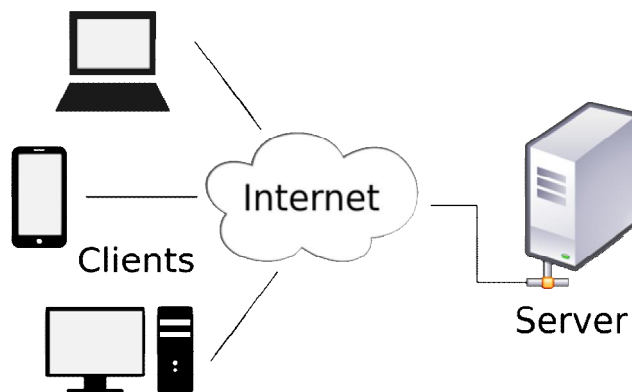


Рис. 1. Клієнт-серверна архітектура

Правила взаємодії між клієнтом і сервером визначаються протоколом обміну (протоколом взаємодії). У моделі клієнт-серверної взаємодії логічно можна виокремити три рівні операцій [1,2]:

- рівень представлення даних, що по суті є інтерфейсом користувача і відповідає за подання даних користувачеві і отримання від нього команд керування;
- прикладний рівень, де реалізовано основну логіку застосунку і на якому виконується необхідна обробка інформації;
- рівень управління даними, де забезпечується зберігання даних та доступ до них.

У дволанковій клієнт-серверній архітектурі передбачається взаємодія двох програмних модулів – клієнтського та серверного. В залежності від розподілу між ними функцій розрізняють:

- модель тонкого клієнта, де усю логіку застосунку та управління даними зосереджено на сервері. Завданням клієнтської програми є забезпечити функції рівня представлення;
- модель товстого клієнта, де сервер виконує керування даними, а обробку інформації та інтерфейс користувача зосереджено на стороні клієнта. Терміном товстих клієнтів також називаються пристрої, де є обмеження за потужністю (мобільні телефони, кишенькові комп'ютери та ін.).

У триланковій клієнт-серверній архітектурі передбачається відокремлення прикладного рівня від управління даними. На відокремленому програмному рівні зосереджується уся логіка прикладного застосунку. Дволанкова архітектура є простішою через обробку всіх запитів одним сервером, але й менш надійною і вимагає високої продуктивності сервера. Структура триланкової архітектури є складнішою і, завдяки розподілу функцій між серверами другого і третього рівнів, забезпечує:

- високий ступінь гнучкості і масштабованості;

- високий рівень безпеки;
- високу ступінь продуктивності.

Основну ідею архітектури «клієнт-сервер» становить поділ мережевого додатку на декілька компонентів, де кожен них реалізує специфічний набір сервісів. Виконання компонентів такого додатку може здійснюватися на різних комп'ютерах з покладеними на них серверними чи клієнтськими функціями. Все це забезпечує підвищення надійності, безпеки і продуктивності мережевих додатків та мережі загалом.

Відповідно до своїх завдань сервер може виконувати різноманітні функції:

- веб-сервера, де відбувається приймання HTTP-запитів від клієнтів. Веб-сервером може виступати, як програмне забезпечення при виконанні функції веб-сервера, так і комп'ютер, де функціонує це програмне забезпечення;
 - сервера застосунків, де виконуються деякі прикладні програми;
 - сервера баз даних для опрацювання запитів з використанням мови SQL;
 - файлового сервера, де зберігається інформація у вигляді файлів;
 - сервера друку, де відбувається надання та управління доступом до принтерів;
 - поштового сервера, де відбувається обслуговування поштових скриньок користувачів;
 - термінального сервера, де клієнти з робочих терміналів отримують доступ до обчислювальних ресурсів надпотужного комп'ютера;
 - сервера віддаленого доступу і VPN сервера, що надають точку входу до мережі для віддалених користувачів;
 - ігрового сервера із забезпеченням зв'язку між клієнтами для надання можливості комунікації в рамках програмної оболонки конкретної гри та ін.

В реалізації стратегії програм при створенні каталогу користувача з улюблених фільмів/серіалів/ігор використовують клієнт-серверну взаємодію з базою даних.

Для ефективного функціонування в програмі передбачається використання наступних алгоритмів [3,4]:

- впорядкування елементів колекцій за допомогою швидкого сортування і сортування Шелла;
- знаходження інформації в БД за допомогою бінарного пошуку.

За основу БД був взятий Json файл з використанням засобів LinQ (Language-Integrated Query) для звертання, вибірки, тощо. Програмне забезпечення реалізовано з використанням мови C# 7 на основі фреймворку WPF (Windows Presentation Foundation) [5-8].

Загалом, існує декілька різновидів LINQ:

- LINQ to Objects: застосовується для роботи з масивами і колекціями;
- LINQ to Entities: використовується при зверненні до баз даних через технологію Entity Framework;
 - LINQ to Sql: технологія доступу до даних в MS SQL Server;
 - LINQ to XML: застосовується при роботі з файлами XML;
 - LINQ to DataSet: застосовується при роботі з об'єктом DataSet;
 - Parallel LINQ (PLINQ): використовується для виконання паралельних запитів;

В пропонованій роботі використано LINQ to Objects та LINQ to Sql.

Мета роботи

Метою роботи є застосування алгоритмів сортування та бінарного пошуку інформації для пришвидшення обробки бази даних фільмів, використання технологій Json, LinQ, .NET та XAML для розроблення оригінального програмного рішення.

Виклад основного матеріалу

В процесі реалізації програмного рішення розроблено інтерфейс користувача. Інтерфейс завантаження розробленої програми із функціями входу та авторизації користувача наведено на рис. 2.

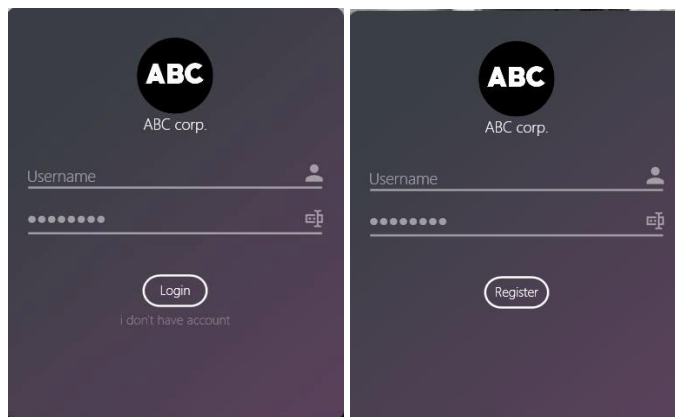


Рис. 2. Вхід у програму (Авторизація / Реєстрація)

Після успішної ідентифікації відбувається перехід у головне меню програми, де показано меню та відображення набору фільмів (рис.3).

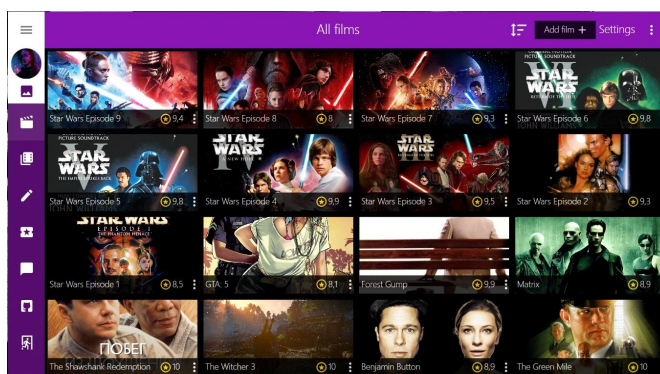


Рис. 3. Головне меню з фреймами даних фільмів

У головному меню можна додавати нові бібліотеки. Інтерфейс сторінки зі створення нових бібліотек показаний на рис. 4. Ми можемо побачити вигляд бібліотеки, виведення кількості фільмів, середнього балу тощо.

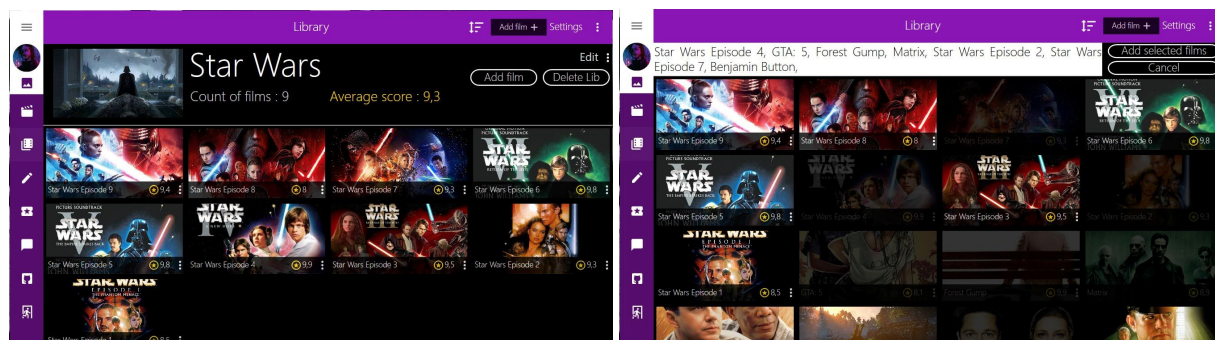


Рис. 4. Сторінки створення нових бібліотек

Під час додавання нових елементів до бібліотеки виконується перевірка наявності його вже у бібліотеці. При спробі додати фільми, які вже наявні в бібліотеці, операція буде відмінена.

На рис. 5. показані вікна, де можна отримати інформацію про акаунт користувача та змінити ім'я, пароль чи фото профілю, а також налаштувати зовнішній вигляд програми з вже готовими прisetами або створити свій вигляд з з використанням кольору для окремих елементів UI.

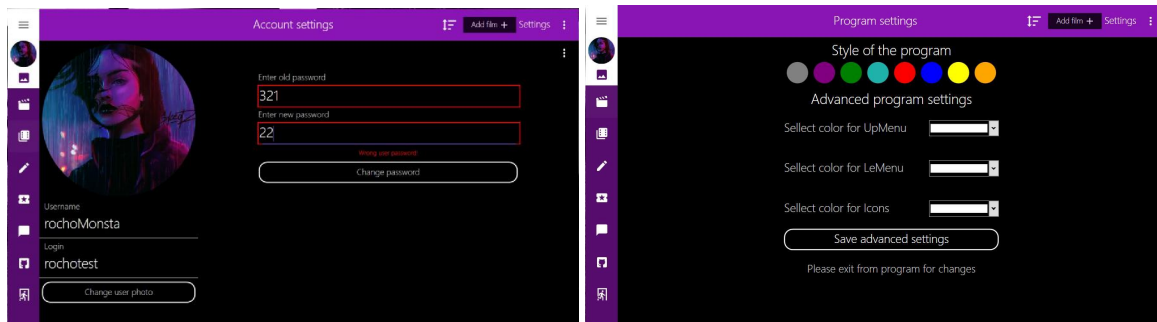


Рис. 5. Налаштування акаунта користувача та інтерфейсу програми

Використання алгоритмів швидкого сортування, сортування методом Шелла та бінарного пошуку застосовується при впорядкуванні елементів з головного меню. На рис. 6. показано реалізацію сортування колекції фільмів відповідно до параметрів.

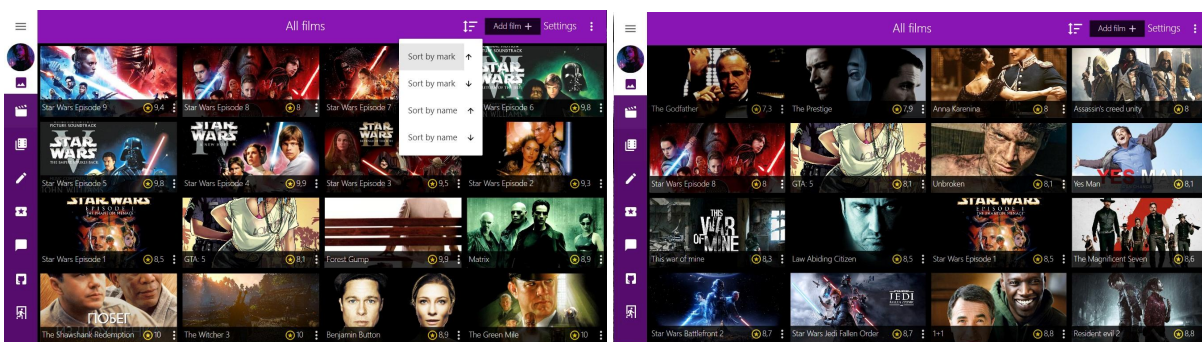


Рис. 6. Меню та результат сортування фільмів

У користувача наявна можливість додавання фільму або будь якого іншого елемента до колекції та перегляд детальної інформації щодо фільму або будь якого іншого елемента. Відповідні вікна сторінок показані на рис. 7.

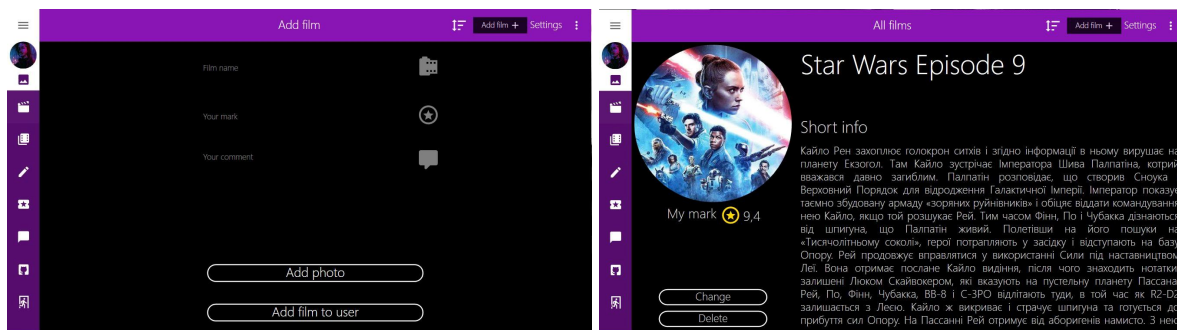


Рис. 7. Сторінки додавання та перегляду фільмів

Отже, ми бачимо відображення короткої інформації про вибраний фільм. Користувач має можливість не лише переглядати, а й редагувати та видаляти інформацію. Швидкість опрацювання

інформації при сортуванні елементів визначається ефективністю самих методів швидкого сортування і Шелла. Знаходження інформації у базі даних значно прискорюється застосуванням алгоритму бінарного пошуку.

Висновок

У роботі досліджено алгоритми сортування та пошуку інформації для пришвидшення обробки записів бази даних фільмів та розроблено оригінальне програмне рішення. Реалізація завдання зроблена мовою програмування C# 7 на основі фреймворку WPF. У роботі використано технології Json, LinQ, .NET та XAML. Для пришвидшення обробки інформації застосовано алгоритми швидкого сортування, сортування методом Шелла та бінарного пошуку.

Отримані результати засвідчують, що розроблена система працює ефективно та надійно. Для виконання дій з користувацьким інтерфейсом у системі передбачено мінімальну кількість кроків.

Список літератури

1. Мартин Фаулер. Шаблоны корпоративных приложений.-К.: Издавничий дім "Диалектика-Вильямс", 2019.-544 с.
2. Гектор Гарсиа-Молина, Джеффри Д. Ульман, Дженнифер Уйдом. Системы баз данных. Полный курс.-К.: Издавничий дім "Диалектика-Вильямс", 2018.-1088 с.
3. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ.-К.: Издавничий дім "Диалектика-Вильямс", 2018.-1328 с.
4. Альфред Ахо, Джон Хопкрофт, Джеффри Ульман. Структуры данных и алгоритмы. Классическое издание.-К.: Издавничий дім "Диалектика-Вильямс", 2019.-400 с.
5. Джозеф Албахари, Бен Албахари. C# 7.0. Справочник. Полное описание языка. -К.: Издавничий дім "Диалектика-Вильямс", 2019.-1024 с.
6. Эндрю Троелсен, Филипп Джепикс. Язык программирования C# 7 и платформы .NET и .NET Core. -К.: Издавничий дім "Диалектика-Вильямс", 2019.-1328 с.
7. Джон Скит. C# для профессионалов: тонкости программирования.-К.: Издавничий дім "Диалектика-Вильямс", 2019.-608 с.
8. Мэтью Мак-Дональд. WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов.-К.: Издавничий дім "Диалектика-Вильямс", 2017.-1024 с.

Reference

1. Martyn Fauler. Shablony korporatyvnykh prylozheniy.-К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2019.-544 s.
2. Gektor Garsya-Molyna, Dzheffry D. Ulman, Dzhennyfer Uydом. Systemy baz dannykh. Polnyj kurs.-К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2018.-1088 s.
3. Tomas X. Kormen, Charlz Y. Lejzerson, Ronald L. Ryvest, Klyfford Shtajн. Algoritmy: postroyeniye y analiz.-К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2018.-1328 s.
4. Alfred Acho, Dzhon Chopkroft, Dzheffry Ulman. Struktury dannykh y algoritmy. Klassycheskoe yzdaniye.-К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2019.-400 s.
5. Dzhozef Albachary, Ben Albachary. C# 7.0. Spravochnyk. Polnoe opysaniye yazyka. -К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2019.-1024 s.
6. Endryu Troelsen, Fylypp Dzhepyks. Yazyk programmyrovaniya C# 7 y platformy .NET y` .NET Core. -К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2019.-1328 s.
7. Dzhon Skyt. C# dlya professyonalov: tonkosty programmyrovaniya.-К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2019.-608 s.
8. Metyu Mak-Donald. WPF: Windows Presentation Foundation v .NET 4.5 s prymeramy na C# 5.0 dlya professyonalov.-К.: Vydavnychyj dim "Dyalektyka-Vylyams", 2017.-1024 s.