

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра інженерії програмного забезпечення

## **Пояснювальна записка**

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ  
ФІЛЬМОТЕКОЮ МОВОЮ C#»**

Виконала: студентка 5 курсу, групи ППЗ– 51  
спеціальності

121 Інженерія програмного забезпечення  
(шифр і назва спеціальності)

Окупнярек К. О.

(прізвище та ініціали)

Керівник Жебка В.В.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Нормоконтроль \_\_\_\_\_

(прізвище та ініціали)

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрямок підготовки -121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

О.В. Негоденко

“16” лютого 2022 року

### **ЗАВДАННЯ** НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Окупнярек Катерина Олегівна

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для управління фільмотекою мовою С#»

Керівник роботи проф.каф. ПЗ, д.т.н., доц. Жебка В.В.,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 1 8 ” лютого 2022 року № 22.

2. Строк подання студентом роботи 03.06.2022

3. Вихідні дані до роботи:

3.1. Мова програмування С#;

3.2. Середовище розробки Visual Studio;

3.3. Технологія Windows Presentation Foundation (WPF).

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1. Проаналізувати та виділити переваги розробленого ПЗ;

4.2. Обґрунтувати обрані програмні застосунки та шляхи розробки додатку;

4.3. Розробити програмне забезпечення та користувацький інтерфейс;

4.4. Зробити висновки за результатами роботи.

5. Перелік графічного матеріалу

1. Логотипи технологій;

2. Діаграми діяльності, стану, прецедентів;

3. Схеми;

4. Скріни програми.

6. Дата видачі завдання 11.04.2022

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04-14.04	
2	Вивчення та аналіз задачі	15.04-17.04	
3	Розробка структури додатку	18.04-21.04	
4	Розробка дизайну та графічних елементів	22.04-25.04	
5	Програмна реалізація системи	26.04-05.05	
6	Налагодження програми	05.05-07.05	
7	Вступ, висновки, реферат	07.05-10.05	
8	Розробка обов'язкових демонстраційних матеріалів	11.05-15.05	
9	Попередній захист роботи	16.05-01.06	
10	Подання роботи в деканат	03.06	

Студент Окупнярек К.О.  
( підпис ) (прізвище та ініціали)

Керівник роботи Жебка В. В.  
( підпис ) (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 67 сторінок, включає 40 рисунків, 4 таблиці та 12 посилань.

Метою дипломної роботи є покращення процесу управління роботою фільмотеки за допомогою створеного програмного додатку.

Об'єкт дослідження – процес управління фільмотекою.

Предмет дослідження – програмний додаток на мові C#.

Було створено десктопний додаток за допомогою мови програмування C# у інтегрованій середі розробки Microsoft Visual Studio. Додаток працює з онлайн-базою даних The Movie Database (TMDB) .

Розроблено програмний додаток управління фільмотекою з функціями відображення даних, пошук за назвою, жанром і датою випуску, відображення детальної інформацію, створення власного списку та його відображення.

Додаток створено для звичайних користувачів для управління фільмотекою. Додаток працює на комп'ютерах з операційною системою Windows версій не нижче Windows 7.

Ключові слова: фільмотека, продукти кіно, управління, MediaLibrary, C#, WPF.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ</b> .....	7
<b>ВСТУП</b> .....	9
<b>1. АНАЛІЗ І ВИДІЛЕННЯ ПЕРЕВАГИ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	11
<b>1.1. Постановка задачі розробки</b> .....	11
<b>1.2. Аналіз існуючого програмного забезпечення</b> .....	12
1.2.1. Мобільний додаток «Фільмотека» .....	12
1.2.2. Мобільний додаток «MovieDiary».....	14
1.2.3. Мобільний додаток «More.tv» .....	16
1.2.4. Необхідність розробки альтернативи.....	17
<b>2. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	19
<b>2.1. Моделювання програмного забезпечення</b> .....	19
2.1.1. Діаграма прецедентів .....	20
2.1.2. Діаграма станів.....	22
2.1.3. Діаграма діяльності .....	22
<b>2.2. Обґрунтування вибору засобів реалізації програмного комплексу</b> .....	24
2.2.1. Середовище розробки Visual Studio.....	25
2.2.2. Мова програмування C# .....	27
2.2.3. Фреймворк .Net Framework.....	30
2.2.4. Технологія Windows Presentation Foundation (WPF) .....	32
2.2.5. Мова розмітки XAML .....	34
2.2.6. Онлайн-база даних The Movie Database (TMDB) .....	36
2.2.7. Шаблон Model-View-ViewModel.....	37
<b>2.3. Структура програмного забезпечення</b> .....	38
<b>3. ТЕХНОЛОГІЧНИЙ РОЗДІЛ</b> .....	40
<b>3.1. Керівництво користувача</b> .....	40

3.1.1. Запуск програми .....	40
3.1.2. Вікно авторизації.....	40
3.1.3. Головна сторінка додатку .....	42
3.1.4. Сторінка «Фільми».....	43
3.1.5. Сторінка «Серіали» .....	43
3.1.6. Сторінка «Зірки кіно» .....	44
3.1.7. Сторінка «Вибране» .....	45
3.1.8. Пошук за назвою .....	45
3.1.9. Пошук фільмів за жанром та роком випуску.....	47
3.1.10. Пошук серіалів за жанром та роком випуску .....	48
3.1.11. Перегляд детальної інформації про фільм.....	48
3.1.12. Перегляд детальної інформації про актора.....	51
3.1.13. Перегляд детальної інформації про серіал.....	52
3.1.14. Робота зі списком вибраного .....	54
3.1.15. Налаштування додатку.....	55
<b>3.2. Тестування програмного продукту .....</b>	<b>56</b>
3.2.1. Мета тестування .....	56
3.2.2. Результати тестування .....	56
<b>ВИСНОВКИ .....</b>	<b>60</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>	<b>61</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення.

TMDB – це онлайн-база даних фільмів та серіалів.

DVD – носій інформації у вигляді диска, зовні схожий з компакт-диском.

CRUD – 4 основні функції управління даними «створення, читання, оновлення і вилучення».

ОС – операційна система.

Android - операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux.

YouTube - популярний відеохостинг, що надає послуги розміщення відеоматеріалів.

UML – уніфікована мова моделювання.

MVVM – шаблон Model-View-ViewModel.

Серіалізація – процес перетворення будь-якої структури даних у послідовність бітів.

Десеріалізація – це процес відновлення об'єкта з байт.

JSON – це текстовий формат обміну даними між комп'ютерами.

Windows – узагальнювальна назва операційних систем для комп'ютерів, розроблених корпорацією Microsoft.

MacOS – серія пропріетарних графічних операційних систем корпорації Apple Inc.

Linux – загальна назва UNIX-подібних операційних систем на основі однойменного ядра.

Mono – багато-платформне вільне відкрите втілення системи .NET, яке відповідає стандартам ECMA, включаючи серед іншого і компілятор C#, і Common Language Runtime.

Xamarin – це фреймворк для кроссплатформної розробки мобільних додатків (iOS, Android, Windows Phone) з використанням мови C#.

API – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Фреймворк – інфраструктура програмних рішень, що полегшує розробку складних систем.



## ВСТУП

У сучасному житті існує безліч фільмів, серіалів та інших продуктів кіно, тож для зберігання і швидкого користування ними існують онлайн-бази даних. Онлайн-база даних — електронна розподілена база даних, доступна з локальної або мережі Інтернет. До найбільш відомих баз належать Internet Movie Database(IMDb), The Movie Database (TMDB).

Метою дипломної роботи є покращення процесу управління роботою фільмотеки за допомогою створеного програмного додатку.

Завдання:

1. Проаналізувати наявні програмні продукти для управління фільмотекою.
2. Визначити технічне завдання.
3. Здійснити моделювання додатку.
4. Обґрунтувати вибрані технології для реалізації додатку.
5. Описати структуру програми.
6. Створити керівництво користувача.
7. Здійснити тестування продукту.
8. Зробити висновки.

Об'єкт дослідження – процес управління фільмотекою.

Предмет дослідження – програмний додаток на мові програмування C#.

Розроблена в дипломній роботі програма MediaLibrary призначена для користувачів комп'ютерів і забезпечує швидке й просте управління та роботу з фільмотекою.

Завдяки використанню сучасних технологій розроблене програмне забезпечення має не складний і зрозумілий інтерфейс, швидко працює й надає актуальну інформацію.

Середовищем розробки програмного забезпечення вибрано Visual Studio 2019, мовою програмування — C#. Всі дані будуть братися і зберігатися в онлайн-базі

даних — The Movie Database (TMDB). Програма працює на персональних комп'ютерах з операційною системою Windows версій не нижче Windows 7.

Розроблене програмне забезпечення призначене для звичайних користувачів.

У першому розділі пояснювальної записки проведено аналіз і виділення переваги розробленого програмного забезпечення. У другому — обґрунтовано вибір програмних засобів розробки системи та описано програмну реалізацію. У третьому розділі описано тестування додатку.

# 1. АНАЛІЗ І ВИДІЛЕННЯ ПЕРЕВАГИ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 1.1. Постановка задачі розробки

У минулому столітті для зберігання продуктів кіно використовували відеокасети, потім на заміну прийшли DVD диски, далі на ринку з'явилися USB-накопичувачі. А на теперішній час взагалі відпала необхідність в таких носіях, адже можна знайти всю необхідну інформацію на просторах глобальної мережі Інтернет.

У сучасному житті існує безліч фільмів, серіалів та інших продуктів кіно, тож для зберігання і швидкого користування ними існують онлайн-бази даних. Онлайн-база даних — електронна розподілена база даних, доступна з локальної або мережі Інтернет. До найбільш відомих баз належать Internet Movie Database(IMDb), The Movie Database (TMDB).

На сьогоднішній день існує велика кількість фільмотек, які відрізняються за дизайном, наповненням, тематикою, проте можливості в них однакові. Дуже складно знайти фільмотеку, яка відрізняється функціями. Тому поставлено завдання розробити щось нове та цікаве для користувача.

Тому доречним є розробка й створення на основі сучасних комп'ютерних технологій програмного продукту для перегляду інформації про фільми та серіали з можливістю створення власних списків і перегляд повної інформації про вибраний об'єкт.

Розроблюваний програмний додаток повинен забезпечувати виконання таких функцій:

- відображення фільмів, серіалів, акторів;
- отримання повної інформації про вибраний об'єкт;
- пошук по назві, по жанру та року випуску;
- перегляд трейлерів;
- авторизація і реєстрація користувача;

- підтримка української та англійської мови;
- можливість вибору теми інтерфейсу;
- створення власних списків;
- CRUD-операції над списками;
- зручний та зрозумілий інтерфейс.

Для роботи програми потрібен комп'ютер з операційною системою Windows і постійний доступ до Інтернету.

Потенційними користувачами створюваного продукту є звичайні користувачі, які мають комп'ютер з операційною системою Windows.

## **1.2. Аналіз існуючого програмного забезпечення**

На сьогоднішній день представлена невелика кількість десктопних застосунків для управління фільмотекою. Більшість сервісів представлені як веб-сайти або мобільні додатки. Кожен з них має переваги й недоліки.

Нижче проаналізовано додатки фільмотек: «Фільмотека», «MovieDiary» та «More.tv»

### **1.2.1. Мобільний додаток «Фільмотека»**

«Фільмотека» – це мобільний додаток, розроблений для ОС Android. Призначений для перегляду фільмів, мультфільмів та відеороликів з сервісу YouTube. Вигляд додатку подано на рисунках 1.1 і 1.2.

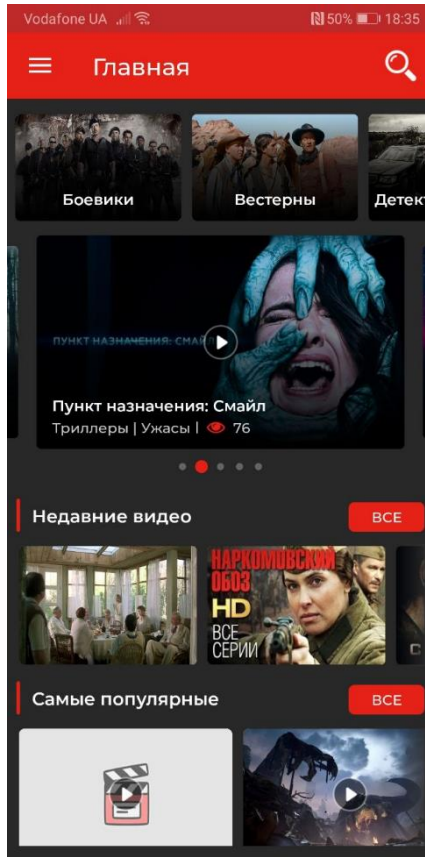


Рисунок 1.1 – Головна сторінка додатку



Рисунок 1.2 – Детальна інформація про вибраний фільм

Додаток має наступний функціонал:

- перегляд фільмів, мультфільмів та відеороликів;
- перегляд детальної інформації;
- пошук за жанром;
- пошук за назвою;
- зміна теми інтерфейсу;
- додавання вибраних фільмів до списку.

Переваги додатку:

- не порушує право власності, оскільки весь контент отримується через YouTube API;
- зручний та зрозумілий інтерфейс.

Недоліки додатку:

- відсутність реєстрації та авторизації;
- інформація про вибраний фільм не повноцінна;
- відсутня інформація про акторів;
- відсутня функція зміни мови інтерфейсу;
- відсутня інформація про серіали;
- некоректно працює пошук.

### 1.2.2. Мобільний додаток «MovieDiary»

«MovieDiary» – це особистий трекер фільмів. Призначений для зберігання власних списків фільмів та серіалів, не призначений для перегляду їх. Вигляд додатку подано на рисунках 1.3 і 1.4.

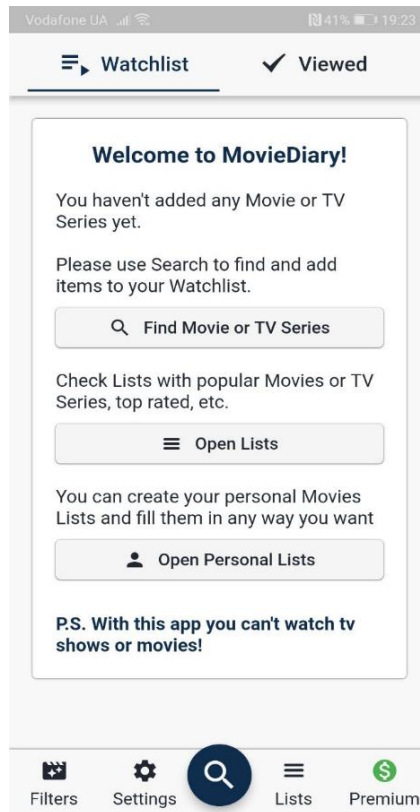


Рисунок 1.3 – Головна сторінка додатку



Рисунок 1.4 – Списки доступних фільмів

Додаток має наступний функціонал:

- перегляд списків фільмів та серіалів;
- перегляд детальної інформації;
- пошук за жанром, за назвою;
- реєстрація та авторизація;
- додавання вибраних об'єктів до списку.

Переваги додатку:

- не порушує право власності.

Недоліки додатку:

- відсутня функція зміни мови інтерфейсу;
- відсутня функція зміни теми інтерфейсу;
- не зрозумілий користувацький інтерфейс.

### 1.2.3. Мобільний додаток «More.tv»

«More.tv» – це онлайн-сервіс. Призначений для перегляду фільмів та серіалів за підпискою. Вигляд додатку подано на рисунку 1.5.



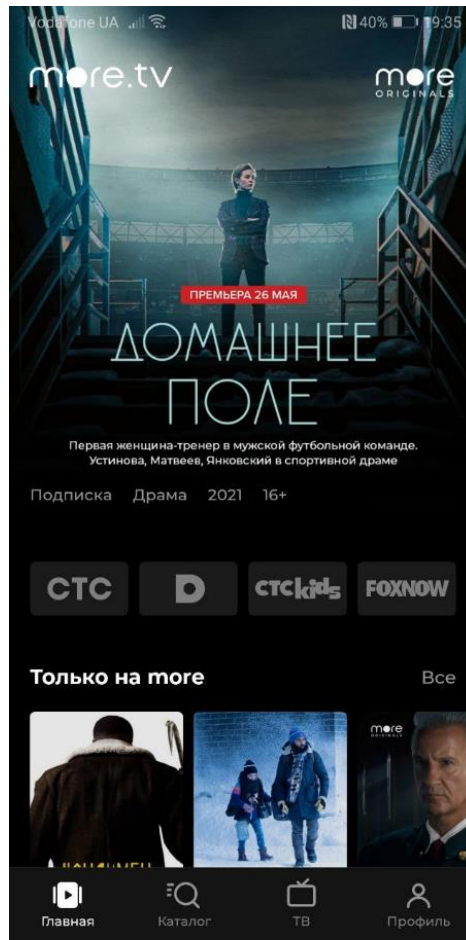


Рисунок 1.5 – Головна сторінка додатку

Додаток має функціонал подібний до перших двох мобільних додатків.

Переваги додатку:

- не порушує право власності;
- наявність підписки.

Недоліки додатку:

- відсутня функція зміни мови інтерфейсу;
- відсутня функція зміни теми інтерфейсу;
- платна підписка.

#### 1.2.4. Необхідність розробки альтернативи

Було розглянуто кілька додатків для управління та роботи фільмотеки. Брався до уваги функціональний набір додатків, було проаналізовано переваги й недоліки кожного з них (табл. 1.1).

Таблиця 1.1 – Характеристика аналогів

Додаток	Функція зміни мови інтерфейсу	Функція зміни теми інтерфейсу	Наявність підписки	Перегляд відео	Інформація про акторів
More.tv	-	-	+	+	+
MovieDiary	-	-	-	-	-
Фільмотека	-	+	-	+	-

Головною необхідністю розробки додатку управління фільмотекою MediaLibrary є те, що подібних рішень для ОС Windows не так вже й багато, тому є затребуваним створення альтернативи.

Основними недоліками розглянутих програмних засобів є:

- відсутність інформації стосовно акторів;
- відображення неповної інформації вибраного об'єкта;
- відсутність вибору мови інтерфейсу;
- відсутність вибору теми інтерфейсу.

Основними перевагами розробленого продукту є:

- швидкість роботи;
- локалізація додатку;
- безкоштовний додаток;
- зручний та зрозумілий інтерфейс.

Таким чином, проблема розробки нових програм для управління фільмотекою на ОС Windows залишається актуальною.

## 2. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1. Моделювання програмного забезпечення

Модель являє собою спрощений вигляд реальності, креслення системи, яке може включати детальні плани і більш абстрактні види, так би мовити «з висоти пташиного польоту». Якісні моделі завжди містять елементи, які мають значний вплив на результат, але не елементи, які не мають значення на цьому рівні абстракції.

Кожну систему можна описати з різних точок зору за допомогою різних моделей, кожна з яких є семантично закритою абстракцією системи. Модель може бути структурною, підкреслюючи організованість системи, або моделлю поведінки, що відображає її динаміку.

Модель ПЗ будується для того, щоб краще розуміти розроблювану систему. Моделювання дозволяє вирішити чотири різні завдання:

1. Візуалізувати систему в її поточному або бажаному для замовника стані.
2. Описати структуру або поведінку системи.
3. Отримати шаблон, що дозволяє сконструювати систему.
4. Документувати прийняті рішення, використовуючи отримані моделі.

Чим більша і складніша система, тим більшого значення набуває моделювання при її розробленні. Без моделі складну систему неможливо сприймати як єдине ціле (навіть програмний еквівалент розплідника значно виграє від використання моделювання).

Людське сприйняття складних сутностей обмежене. Моделювання системи чи об'єкта може звузати проблему, зосереджуючись на певних аспектах одночасно. Важку задачу завжди легше вирішити, якщо розбити її на менші завдання. Моделювання покращує можливості людського інтелекту.

Розроблений додаток буде моделюватись за допомогою уніфікованої мови моделювання UML. Вона застосовується до широкого спектру програмних систем проектування, різних областей застосування, типів організацій, рівнів можливостей,

розмірів проекту. UML описує об'єкти в єдиному синтаксисі, тому незалежно від того, де ви намалюєте діаграму, будь-хто, хто знайомий з цією графічною мовою, знатиме правила — навіть в іншій країні..

### 2.1.1. Діаграма прецедентів

Діаграма прецедентів наочно зображує різні сценарії взаємодії між акторами (користувачами) і прецедентами (випадками використання), описує функціональні аспекти (бізнес-логіку) системи.

Прецедентні діаграми не тільки відіграють важливу роль у спілкуванні між збирачами вимог і потенційними користувачами.

Діаграми прецедентів додаються бізнес-логікою та детальними специфікаціями прецедентів як вихідна інформація, яку учасники проекту успішно використовують на всіх його фазах (виникнення, проектування, програмування, тестування, документація).

Добре продуману та повну специфікацію прецеденту легко перевести в тестові випадки.

Елементи діаграми прецедентів:

- Актор — користувач.
- Прецедент — випадок використання, дія. Позначається овалом.
- Граничні межі системи — охоплюють усі випадки використання у системі. Позначається прямокутником.

Елементи взаємодії:

- Використовує — користувач виконує дію.
- Включає — один прецедент використовує іншого.
- Розширює — представлення дочірніх прецедентів.
- Вимагає — наступний прецедент вимагає виконання попереднього.
- Схожий — прецеденти подібні, але описують різну функціональність.
- Рівнозначний — подібна функціональність, але користувач сприймає, як різну.



Рисунок 2.1 – Умовні позначення діаграми прецедентів

Розроблений додаток відповідає наступній діаграмі прецедентів, зображеній на рисунку 2.2.

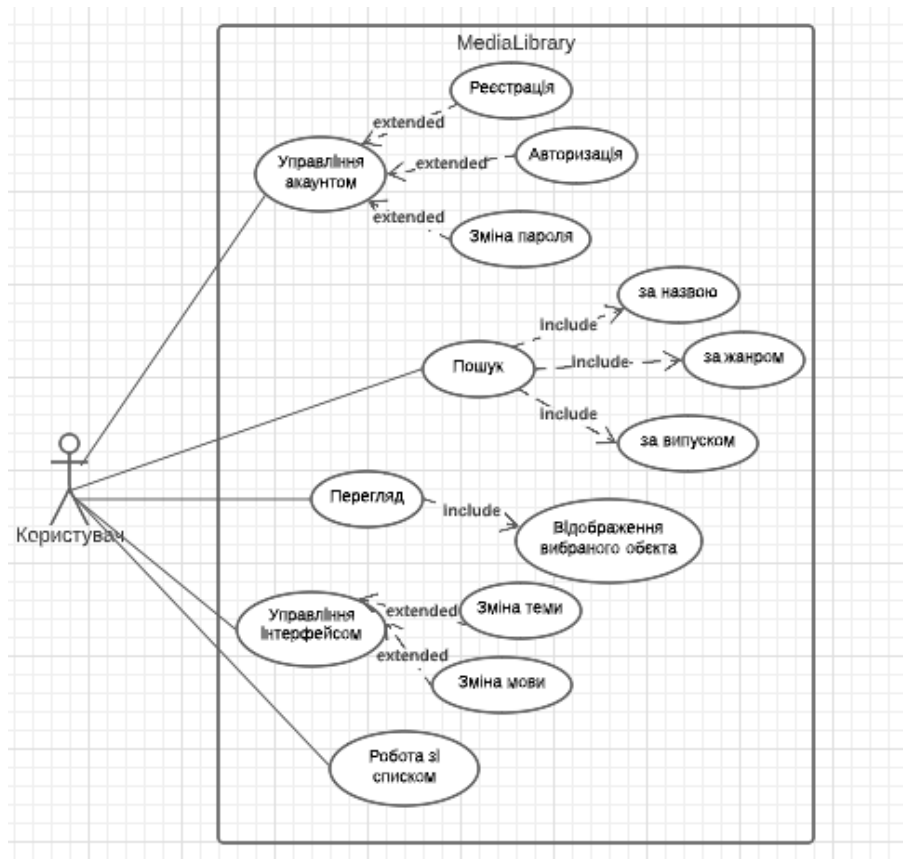


Рисунок 2.2 – Діаграма прецедентів розробленого додатку

### 2.1.2. Діаграма станів

Діаграми стану допомагають описати поведінку окремого об'єкта.

Аналітики часто використовують діаграми станів, щоб описати послідовність переходу об'єктів з одного стану в інший.

Діаграма станів покаже нам усі можливі стани, в яких може перебувати об'єкт, і процес зміни станів через зовнішні впливи.

Розроблений додаток відповідає наступній діаграмі станів, зображеній на рисунку 2.3.

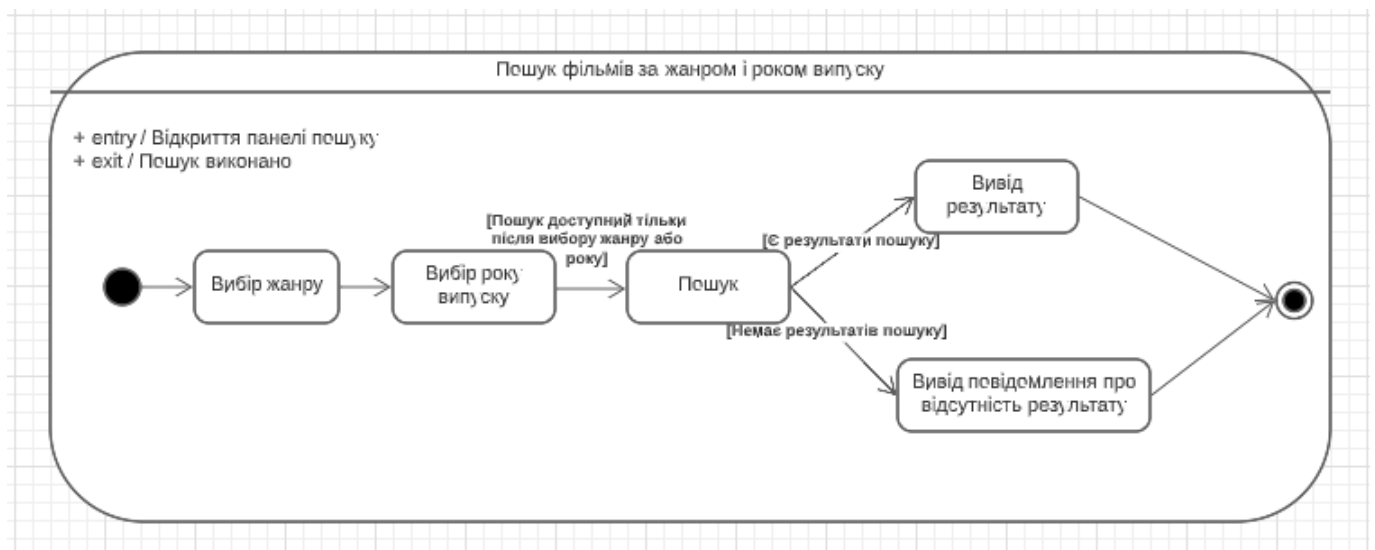


Рисунок 2.3 – Діаграма станів для пошуку фільмів за жанром і роком випуску

### 2.1.3. Діаграма діяльності

Діаграми діяльності використовуються для моделювання послідовного робочого процесу великих видів діяльності, акцентуючи увагу на послідовності дій і відповідних умовах, які ініціюють дії. Стан діяльності відноситься до виконання кожного кроку робочого процесу.

Діаграма діяльності представлена графіком, з'єднаним стрілками. Стрілки йдуть від початку до кінця, представляючи послідовність дій. Чорне коло представляє початковий стан робочого процесу. Кінцевий стан зображений обведеним чорним колом. Закруглені прямокутники, що представляють виконані дії, описані текстом у кожному прямокутнику.

Діамантова форма використовується для представлення рішення, що є ключовою концепцією діаграми діяльності. Після завершення діяльності повинен бути обраний перехід (або набір послідовних дій) з набору альтернативних переходів для всіх випадків використання.

Стрічки синхронізації, що вказують на початок або завершення одночасних дій, використовуються для представлення паралельних підтоків.

Розроблений додаток відповідає наступній діаграмі станів, зображеній на рисунку 2.4.

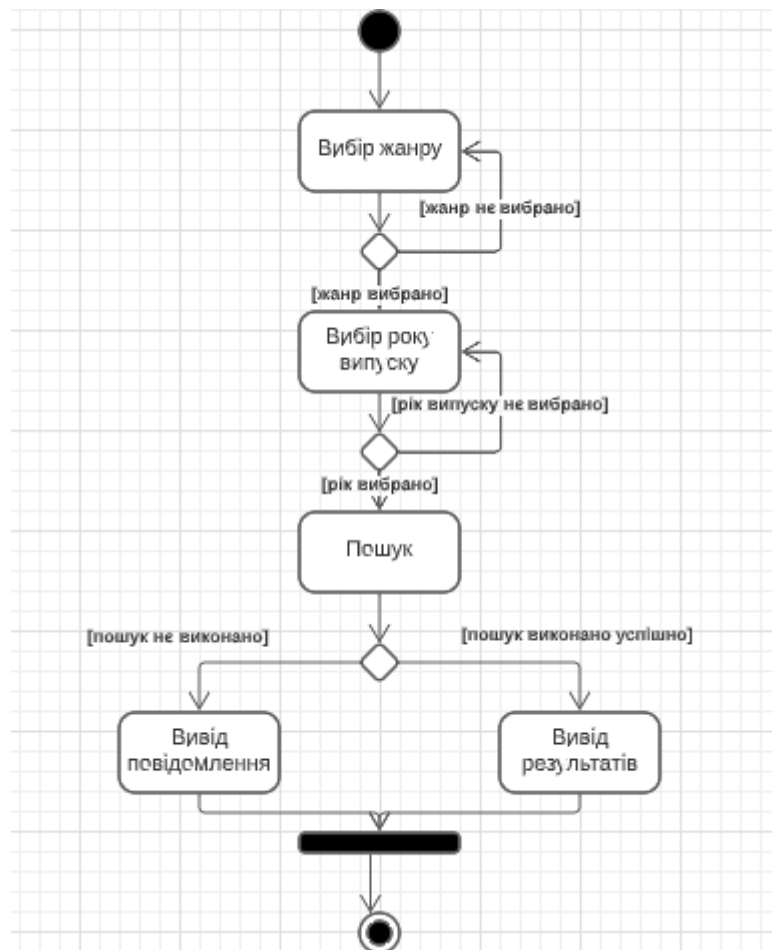


Рисунок 2.4 – Діаграма діяльності для пошуку фільмів за жанром і роком випуску

## 2.2. Обґрунтування вибору засобів реалізації програмного комплексу

Для розробки програмних продуктів необхідно вибрати технології та інструменти, які спрощують вашу роботу та забезпечують інструменти, необхідні для виконання поставлених завдань. Вибір правильного підходу до технології та впровадження програмного забезпечення безпосередньо впливає на час розробки, якість, швидкість та якість продукту.

Для розробки додатку MediaLibrary було вибрано такі засоби:

- середовище розробки Microsoft Visual Studio 2019;
- мова програмування C#;
- фреймворк .Net Framework;
- технологія Windows Presentation Foundation (WPF);
- мова розмітки XAML;
- онлайн-база даних The Movie Database (TMDB).

Також для створення додатку буде використано шаблон проектування Model-View-ViewModel(MVVM) і концепція Single-window. При розробці були додатково використані такі бібліотеки:

- MahApps.Metro.2.4.9 для графічного інтерфейсу користувача;
- Newtonsoft.Json.13.0.1 для серіалізації та десеріалізації;
- System.Windows.Interactivity.WPF.2.0.20525 для додавання команд до об'єктів;
- ControlzEx.4.4.0 для створення власного елемента.

Вибір технологій зумовлений тим, що ці інструменти дають найбільшу функціональність серед конкурентів, саме тому були обрані технології - C#, XAML, бібліотека MahApps.Metro. В якості онлайн-бази даних вибрано The Movie Database (TMDB).

Мовою програмування високого рівня було обрано C#, адже вона має багато бібліотек, що допомагають спростити процес налаштування застосунку, а також



стандартизують архітектуру та стиль написання коду, що важливо для можливого розширення функціоналу іншими розробниками.

Середовище розробки Microsoft Visual Studio також використовує технологію IntelliSense, яка значно прискорює написання коду, покращує обробку коду та надає методи вирішення помилок розробки.

Базою даних було обрано The Movie Database (TMDB) в першу чергу через швидкодію, адже при накопиченні даних це відіграє важливу роль у продуктивності роботи.

Додаток управління фільмотекою працює на комп'ютерах з операційною системою Windows версій не нижче Windows 7.

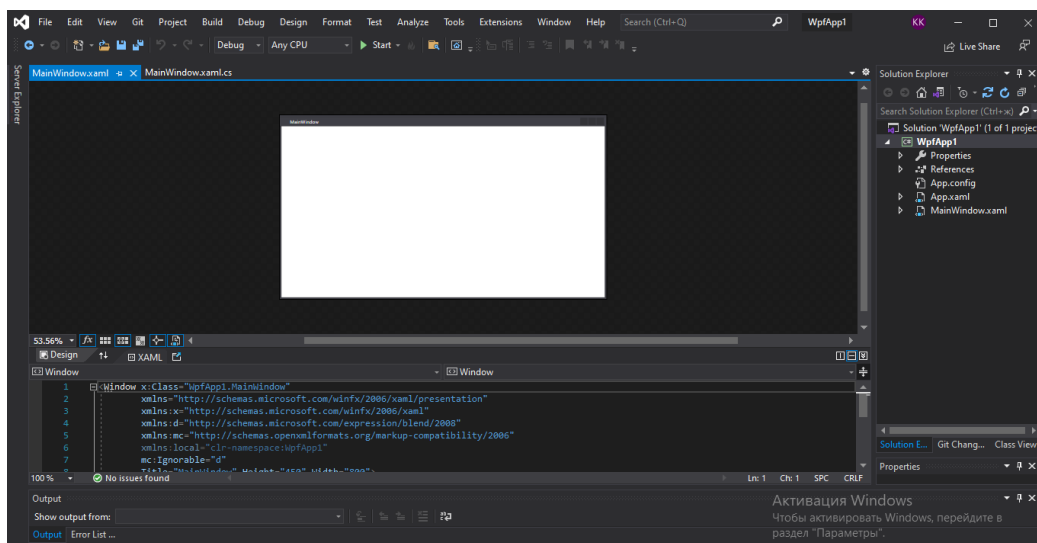
### 2.2.1. Середовище розробки Visual Studio

Microsoft Visual Studio — це сімейство продуктів Microsoft, яке включає інтегроване середовище розробки програмного забезпечення та багато інших інструментів. MSE Visual Studio .NET IDE — це набір інструментів та засобів для розробки різних типів додатків (консольні, Windows, мобільні, веб-додатки) та сервісів. MS Visual Studio — це середовище мультипрограмування, яке підтримує кілька мов програмування, особливо C++, C# (табл. 2.1).

Таблиця 2.1 – Характеристики середовища програмування Visual Studio

Параметр	Характеристика
Розробник	Microsoft
Стабільний випуск	2022 (11.0.507.1)
Написано на мові	C++
Операційна система	Microsoft Windows
Доступні мови	англійська, китайська, іспанська, італійська, корейська, німецька, російська, французька
Тип	інтегроване середовище розробки

Visual Studio включає в себе редактор вихідного коду, який підтримує технологію IntelliSense і забезпечує легкий рефакторинг коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так само як і відладчик машинного рівня. Інші вбудовані інструменти включають редактор форм, веб-редактор, конструктор класів і конструктор схем бази даних для спрощення графічного інтерфейсу користувача програми. Visual Studio дозволяє створювати та підключати сторонні доповнення (плагіни), щоб розширити функціональність майже на всіх рівнях, включаючи додавання підтримки систем контролю версій вихідного коду, таких як Subversion і Visual Source Safe, додавання нових наборів інструментів (наприклад, для редагування та візуального проектування коду на об'єктно орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення) (рисунк 2.5).



Рисунк 2.5 – Вікно середовища програмування Visual Studio

Системні вимоги до середовища наведені в таблиці 2.2.

Таблиця 2.2 – Системні вимоги до Visual Studio

Параметр	Характеристика
Процесор	1,6 ГГц

## Продовження таблиці 2.2

ОЗП	512 Мб
Дисплей	1024x768
Жорсткий диск	900 Мб
Операційна система	Windows XP, Server 2003, Vista, Seven, Ten

## Переваги Visual Studio:

- підтримка міжплатформ: Windows, Linux, Mac;
- вбудований Web-сервер;
- надійна архітектура;
- Intelli-Sense;
- Freeware: Без витрат - мабуть, найкраща ризика для всіх програмістів там, ще більше для організацій;
- підтримка безлічі мов при розробці;
- можливості налагодження.

## Недоліки Visual Studio:

- складна IDE;
- неможливість відладчика (Microsoft Visual Studio Debugger) відстежувати в коді режиму ядра.

## 2.2.2. Мова програмування C#

C# (вимовляється C-Sharp) — це об'єктно-орієнтована мова програмування з безпечною системою типів для платформи .NET. Розроблено Андерсом Галесбергом, Скоттом Вілтамутом і Пітером Голдом під егідою Microsoft Research (належить Microsoft).

C# є близьким родичем мови програмування Java. Мова Java була створена компанією Sun Microsystems, коли глобальний розвиток інтернету поставив завдання розподілених обчислень. Java заснована на популярній мові C++ і виключає потенційно небезпечні речі (наприклад, вказівників без контролю виходу за межі).

Для розподілених обчислень була створена концепція віртуальної машини та машинно-незалежного байт-коду, який є своєрідним посередником між вихідним текстом програми та апаратними інструкціями комп'ютера чи іншого розумного пристрою.

Java набула значної популярності і також ліцензована Microsoft. Проте з часом Sun почала звинувачувати Microsoft у тому, що та при створенні свого клону Java робить її сумісною виключно з платформою Windows, що суперечило концепції машинно-незалежного середовища виконання та порушувало ліцензійну угоду. Microsoft відмовилася виконувати вимоги Sun, тому з'ясування відносин набуло судового статусу. Суд постановив, що позиція Sun була справедливою, і зобов'язав Microsoft відмовитися від неліцензійного використання Java.

У цьому випадку Microsoft вирішила використати своє домінування на ринку для створення власного аналога Java – мови, якою компанія буде володіти повністю. Ця новостворена мова називається C#. Вона успадковує концепції віртуальної машини Java (середовище .NET), байт-коду (MSIL) і вищої безпеки вихідного коду програми, а також враховує досвід використання програм Java.

Нововведенням C# є можливість легше взаємодіяти з кодом, написаним іншими мовами, ніж його попередники, що важливо при створенні великих проектів. Якщо програми різними мовами працюють на платформі .NET, .NET відповідає за сумісність програм (тобто типи даних, зрештою).

Сьогодні C# є флагманською мовою Microsoft, оскільки він використовує нові можливості .NET. Інші мови програмування, хоча й підтримуються, вважаються такими, що мають прогалини у спадковості у використанні .NET. Рядки в C# є типами посилань.

Основні важливі рішення, які Microsoft реалізувала на мові програмування C#, такі:

- компонентно-орієнтований підхід до програмування (часто типовий для ідеології Microsoft .NET);

- властивості як засіб інкапсуляції даних (характерно також в цілому для ООП);
- обробка подій (з розширеннями, включаючи обробку винятків, зокрема оператор try);
- делегати (вказівники на функції, розроблені на C і C#);
- індексатори (індексатори - оператори індексації для доступу до елементів класу контейнера);
- перевантажені оператори (розробка PLO);
- оператор foreach (обробляє всі елементи класу колекції, подібно до Visual Basic);
- механізми боксування та розпакування для перетворення типів;
- атрибути (засіб оперування метаданими в СОМ-моделі);
- прямокутний масив (група елементів, доступ до яких здійснюється за номером індексу з однаковою кількістю стовпців і рядків).

Ймовірно, найважливіше в С# полягає в тому, що код, який вона генерує, може виконуватися лише під час виконання .NET. Наприклад, неможливо створити класичний сервер Som за допомогою С#. Відповідно до термінології Microsoft, код, призначений для виконання в середовищі виконання .NET, є керованим кодом. Бінарні файли, які містять керовані файли, називаються збірками.

#### Переваги С#:

- підтримка більшості продуктів Microsoft;
- безкоштовність ряду інструментів для невеликих компаній та деяких індивідуальних розробників - Visual Studio, хмара Azure, Windows Server, Parallels Desktop для Mac Pro та ін.;
- типи даних мають фіксований розмір (32-бітний int і 64-бітний long), що підвищує «мобільність» мови та спрощує програмування, тому що ви завжди знаєте точно, з чим ви маєте справу;
- автоматичне «складання сміття»;

- велика кількість синтаксичного «цукру» — спеціальних конструкцій, розроблених для розуміння та написання коду;
- низький поріг входження.

Недоліки мови програмування C#:

- пріоритетна орієнтованість на платформу Windows;
- мова безкоштовна тільки для невеликих фірм, індивідуальних програмістів, стартапів та учнів.

### 2.2.3. Фреймворк .Net Framework

.NET Framework(читається дот-нет) — це програмна технологія, надана Microsoft як платформа для створення звичайних програм і веб-додатків. Це багато в чому є продовженням ідей і принципів, закладених у технології Java. Однією з філософій .NET є сумісність служб, написаних різними мовами. Хоча ця функція рекламується Microsoft як перевага .NET, платформа Java має ті самі можливості.

Кожна бібліотека (колекція) в .NET має свою версію, що дозволяє вирішувати можливі конфлікти між колекціями різних версій.

.NET — це кросплатформна технологія, яка в даний час реалізується як DotGNU, варіант технології Linux у Microsoft Windows, FreeBSD (від Microsoft) та проекту Mono (за угодою між Microsoft і Novell).

Захист авторських прав — це створення середовища виконання (CLR) для програм .NET. Багато компаній можуть безкоштовно використовувати компілятори .NET для різних мов.

.NET ділиться на дві основні частини - середовище виконання (по суті, віртуальна машина) і засоби розробки.

Середовище розробки .NET: Visual Studio .NET (C++, C#, J#), SharpDevelop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse має додаток для розробки програм .NET. Програми також можна розробляти в текстовому редакторі та використовувати консольний компілятор.

Як і технологія Java, середовище розробки .NET створює байт-код, який призначений для виконання віртуальною машиною. Рідна мова введення в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або скорочено IL. Використання байт-коду дозволяє отримати кросплатформний рівень на рівні скомпільованого проекту (відповідно до .NET: асамблея), а не на рівні необробленого тексту, як у C. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора.

Слід зазначити, що один з перших компіляторів Java JIT також був розроблений Microsoft (зараз J2SE 1.3, створений на Java з 8 травня 2000 року, використовується як основна, більш просунута багаторівнева компіляція - Sun HotSpot). Сучасні методи динамічної компіляції дозволяють досягти рівня продуктивності, подібного до традиційних «статичних» компіляторів (наприклад, C++), і проблеми з продуктивністю часто залежать від якості компілятора. Стек технологій .NET Framework показаний на рисунку 2.6.

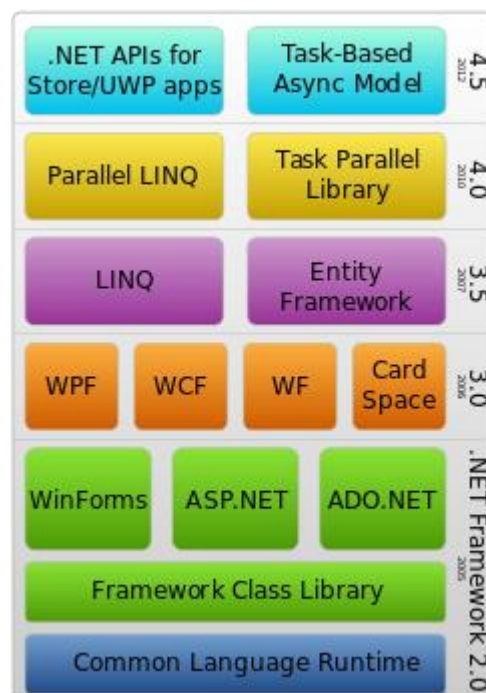


Рисунок 2.6 – Стек технологій .NET Framework

#### 2.2.4. Технологія Windows Presentation Foundation (WPF)

Технологія WPF (Windows Presentation Foundation) є частиною екосистеми платформи .NET і є підсистемою для побудови графічних інтерфейсів.

При створенні традиційних програм WinForms традиційні частини Windows (наприклад, User32 і GDI+) відповідають за елементи керування малюванням і графіку, тоді як програми WPF засновані на DirectX. Це ключова особливість візуалізації графіки в WPF: за допомогою WPF велика частина роботи з відтворення графіки, будь то найпростіші кнопки чи складні 3D-моделі, покладається на графічний процесор на відеокарті, що також дозволяє використовувати апаратне прискорення графіки.

Важливою особливістю є використання декларативної мови розмітки на основі XML: можна створювати розширені графічні інтерфейси, використовуючи декларативні оголошення інтерфейсу або керований код C#, VB.NET і F# або їх комбінацію.

Перша версія, WPF 3.0, була випущена в 2006 році разом з .NET Framework 3.0 і Windows Vista. Відтоді платформа WPF стала частиною екосистеми .NET і зростає разом із .NET Framework. Наприклад, останньою версією .NET Framework зараз є .NET 6, яка повністю підтримує WPF.

##### Переваги WPF

- використання традиційних мов платформи .NET - C#, F# і VB.NET для створення логіки програми;
- можливість декларативно визначати графічні інтерфейси з використанням спеціальної мови розмітки XAML на основі xml, яка є альтернативою програмному створенню графіки та елементів керування, а також можливість комбінувати XAML і C#/VB.NET;
- незалежність від роздільної здатності екрана: оскільки в WPF всі елементи вимірюються в одиницях, що не залежать від пристрою, програми на WPF можуть легко масштабуватися до різних екранів з різною роздільною здатністю.



- нові функції, які важко реалізувати в WinForms, наприклад, використання таких елементів, як створення 3D-моделей, прив'язування даних, стилів, шаблонів, тем, тощо;
- хороша взаємодія з WinForms, завдяки чому, наприклад, у додатках WPF можна використовувати традиційні елементи керування WinForms;
- багаті можливості створення різноманітних додатків: мультимедіа, 2D та 3D-графіка та багаті вбудовані елементи керування, а також можливість створення нових елементів, створення анімації, прив'язки даних, стилів, шаблонів, тем, тощо;
- апаратне прискорення графіки — незалежно від того, чи працюєте ви в 2D чи 3D, графікою чи текстом, усі компоненти програми перетворюються на об'єкти, які Direct3D розуміє, а потім візуалізуються процесором на відеокарті, що призводить до покращення продуктивності та більш гладкої графіки;
- створено програми для багатьох сімейств Windows.

У той же час WPF також має певні обмеження. Хоча підтримується 3D-візуалізація, для створення програм з великою кількістю 3D-зображень, особливо ігор, краще використовувати інші інструменти - спеціальні фреймворки, такі як DirectX або Monogame або Unity.

Також варто відзначити, що кількість програм на WPF і споживання пам'яті під час роботи в середньому дещо вищі в порівнянні з програмами на Windows Forms. Але це компенсується ширшими графічними можливостями та підвищенням продуктивності при малюванні графіки.

Крім того, хоча WPF працює на платформі .NET 5/6, через природу WPF та його залежність від компонентів Windows, наразі програми можна створювати лише на WPF під Windows.

Графічно архітектуру WPF зображено на рисунку 2.7.

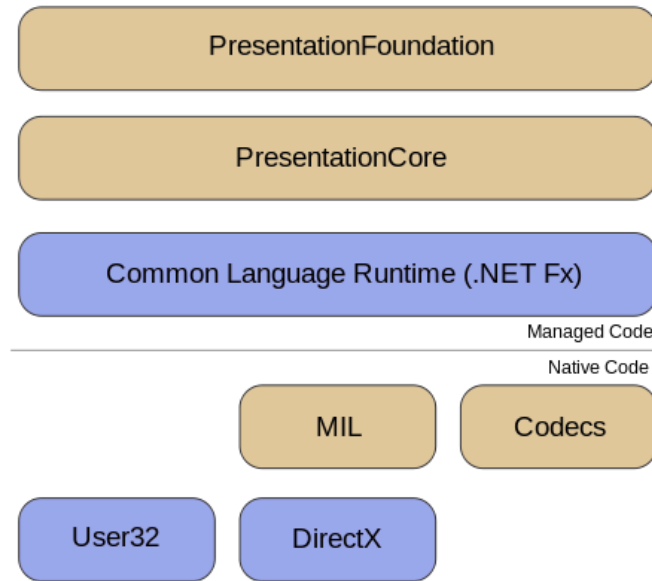


Рисунок 2.7 – Архітектура WPF

### 2.2.5. Мова розмітки XAML

XAML (Extensible Application Markup Language) — це мова розмітки на основі XML для декларативного програмування, розроблена Microsoft. Приклад коду написаного на мові XAML зображено на рисунку 2.8.

```

<Window x:Class="ContentPropertySample.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="MainWindow" Height="400" Width="500" >
<Grid x:Name="ParentGrid">
  <Button Height="50" Margin="10,10,350,310" Content="Hello XAML" />
  <Button Height="80" Margin="10,80,300,170">
    <Rectangle Height="60" Width="120" Fill="Green"></Rectangle>
  </Button>
  <Button Margin="10,201,100,40">
    <StackPanel Orientation="Horizontal">
      <Ellipse Height="60" Width="60" Fill="Red"/>
      <TextBlock TextAlignment="Center"><Run Text=" Red Circle"/></TextBlock>
      <TextBlock TextAlignment="Center"><Run Text="      "/></TextBlock>
      <Rectangle Height="60" Width="120" Fill="Green"></Rectangle>
      <TextBlock TextAlignment="Center"><Run Text=" Green
Rectangle"/></TextBlock>
    </StackPanel>
  </Button>
</Grid>
</Window>
  
```

Рисунок 2.8 – Фрагмент коду написаного на мові XAML

Модель програми Vista включає об'єкт Application. Його набір властивостей, методів і подій дозволяє об'єднувати веб-документи у відповідні програми. Об'єкт Application контролює виконання програми та генерує події для коду користувача. Програмна документація написана на XAML. Однак найголовнішим у використанні XAML є опис інтерфейсу користувача. Логіка програми все ще контролюється процедурним кодом (C#, VB тощо). XAML можна використовувати в програмах на базі браузера та в додатках для настільних комп'ютерів.

XAML включає чотири основні категорії елементів: панелі, елементи керування, елементи, пов'язані з документом, і графіку. Оголошує 7 типів панелей, які встановлюють принципи відображення елементів, вбудованих у них. Такі властивості, як властивості в об'єктно-орієнтованих мовах, використовуються для визначення положення елемента відносно батьківської панелі. Цей синтаксис не дуже підходить для порад CSS, але настільні програмісти будуть знайомі з ним.

Оголошена XAML програма може містити багато сторінок. Елемент керування PageViewer дозволяє розбивати вміст на сторінки. Елемент ContextMenu допомагає створити навігаційне меню програми. Код процедури можна помістити безпосередньо у файл XAML або призначити під час чернетки.

XAML широко використовується в .NET Framework 3.0, особливо в Windows Presentation Foundation (WPF), Windows Workflow Foundation (WWF) і Silverlight. У WPF XAML використовується як мова розмітки інтерфейсу користувача для визначення елементів інтерфейсу користувача, прив'язки даних, підтримки подій та інших властивостей. У WF можна використовувати XAML для визначення робочого процесу.

Файли XAML можна створювати та редагувати за допомогою інструментів візуального дизайну, таких як Microsoft Expression Blend, Microsoft Visual Studio, WPF Visual Designer тощо. Ви також можете створювати їх за допомогою стандартних текстових редакторів, редакторів коду, таких як XAMLPad, або графічних редакторів, таких як Vectropy.

Все, що створено або реалізовано в XAML, можна виразити за допомогою більш традиційних мов .NET, таких як C# або Visual Basic.NET. Однак ключовим аспектом цієї технології є зниження складності інструментів обробки XAML, оскільки XAML заснований на XML. Тому існує багато продуктів для створення програм на основі XAML. Оскільки XAML заснований на XML, розробники та дизайнери можуть працювати з вмістом одночасно без компіляції.

#### 2.2.6. Онлайн-база даних The Movie Database (TMDB)

The Movie Database (TMDB) — це створена спільнотою база даних про фільми та серіали. Кожна частина даних була додана дивовижною спільнотою, починаючи з 2008 року. Сильний міжнародний фокус і широта даних TMDB значною мірою не має собі рівних. Логотип зображений на рисунку 2.9.



Рисунок 2.9 – Логотип The Movie Database

#### Переваги TMDB

- З кожним роком, починаючи з 2008 року, кількість внесків до бази збільшується, оскільки платформу використовують понад 400 000 розробників і компаній, TMDB став першим джерелом метаданих.

- Поряд із обширними метаданими для фільмів, телешоу та людей, також доступна одна з найкращих добірок плакатів високої роздільної здатності та фанарту. У середньому щодня додається понад 1000 зображень.
- База даних міжнародна. Хоча офіційно підтримується 39 мов, також доступні обширні регіональні дані. Кожен день TMDB використовується в більш ніж 180 країнах.
- Спільнота не має собі рівних. Між співробітниками та модераторами спільноти завжди існує зв'язок.
- Надійна платформа. Кожен день сервісом користуються мільйони людей, а обробляється понад 3 мільярди запитів.

Недоліки:

- для отримання даних потрібно мати код доступу;
- не доступна реєстрація та зміна даних профілю через код.

### 2.2.7. Шаблон Model-View-ViewModel

Шаблон Model-View-ViewModel — застосовується для проектування архітектури програми. Спочатку був представлений спільноті Джоном Госсманом (John Gossman) в 2005 як модифікація шаблону Presentation Model. MVVM орієнтований на сучасні платформи розробки, такі як Windows Presentation Foundation, Silverlight від компанії Microsoft, ZK framework.

MVVM використовується для поділу моделі та її уявлення, що необхідно, тому що дозволяє змінювати їх окремо один від одного. Наприклад, програміст задає логіку роботи з даними, а дизайнер відповідно працює з інтерфейсом користувача.

Паттерн MVVM ділиться на три частини(рисунок 2.11):

- Модель (Model), так само, як у класичній MVC, Модель є фундаментальними даними, необхідні для роботи програми.
- Вигляд/Представлення (View) - це графічний інтерфейс, тобто вікно, кнопки і т.п. Вигляд є передплатником на подію зміни значень властивостей або команд, що надаються Моделью Виду. Якщо у Моделі Вида змінилася будь-

яка властивість, то вона сповіщає всіх передплатників про це і Вид у свою чергу витягує оновлене значення властивості з Моделі Вида. Якщо користувач впливає на будь-який елемент інтерфейсу, Вигляд викликає відповідну команду, надану Моделью Вида.

- Модель виду (ViewModel, що означає Model of View) є з одного боку абстракцією Вида, а з іншого надає обгортку даних з Моделі, які підлягають зв'язування. Тобто вона містить Модель, яка перетворена до Вида, а також містить команди, якими може користуватися Вигляд, щоб впливати на Модель.

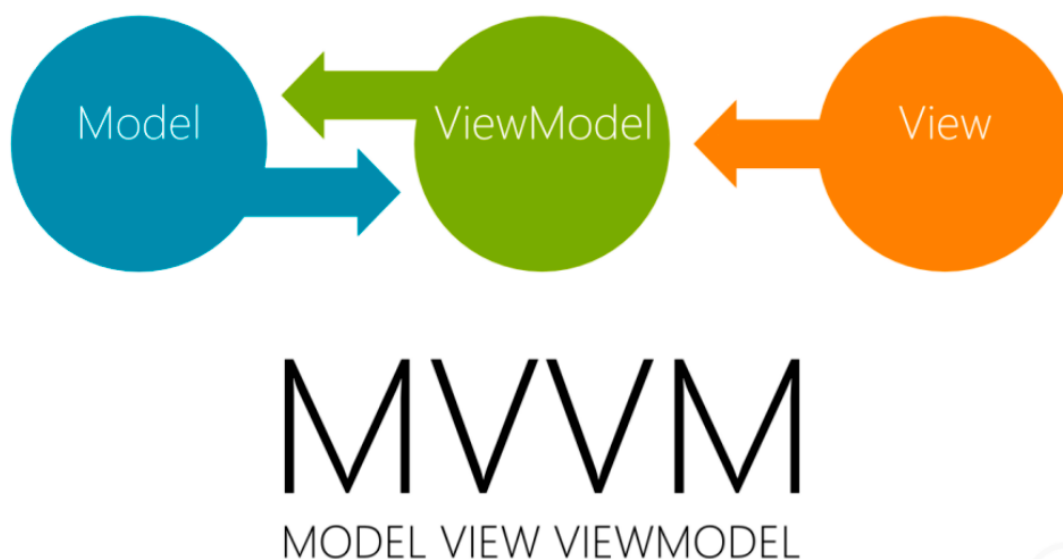


Рисунок 2.11 – Структура MVVM

### 2.3. Структура програмного забезпечення

Програмний додаток розроблявся з використанням мови програмування C# платформи .NET Framework 4.7.2 за шаблоном WPF. Структура додатку зображена на рисунку 2.10.

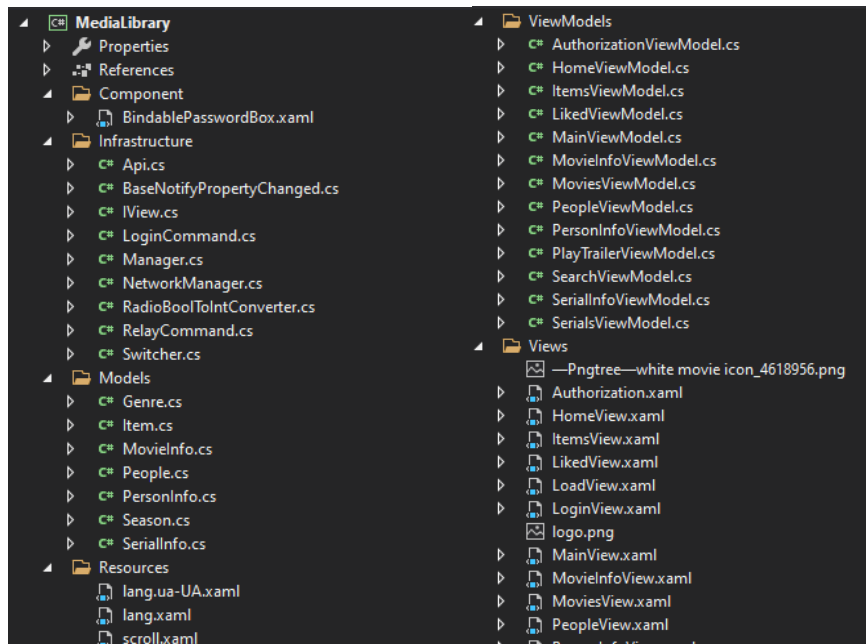


Рисунок 2.10 – Структура розробленого додатку

В папці Component знаходиться файл з власним елементом.

В папці Infrastructure зберігаються всі допоміжні класи для отримання даних з бази даних, для роботи з командами, для перетворення типів.

В папці Resources знаходяться словники, які використовуються для перекладу інтерфейсу.

В папці Models знаходяться всі класи для зберігання даних з бази.

В папці ViewModels прописана вся логіка роботи на сторінках.

В папці Views прописаний тільки графічний інтерфейс.

## 3. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 3.1. Керівництво користувача

Щоб показати процес взаємодії додатку з користувачем наведемо приклади виконання додатку. На рисунках 5.1-5.24 приведені скріншоти роботи з програмою.

#### 3.1.1. Запуск програми

При запуску програми на екрані з'являється заставка додатку(рисунок 3.1).



Рисунок 3.1 – Заставка додатку

При першому запуску програми відображається вікно авторизації(рисунок 3.2), в іншому випадку відображається головна сторінка додатку.

#### 3.1.2. Вікно авторизації

Для входу в додаток потрібно ввести логін і пароль. При вводиті невірному паролю виводить повідомлення про це(рисунок 3.3).



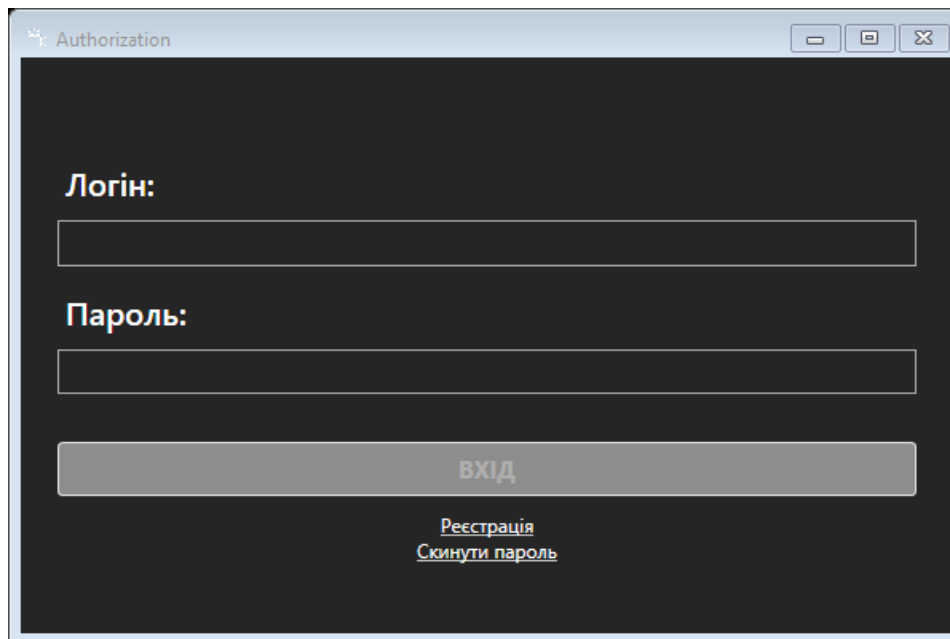


Рисунок 3.2 – Вікно авторизації

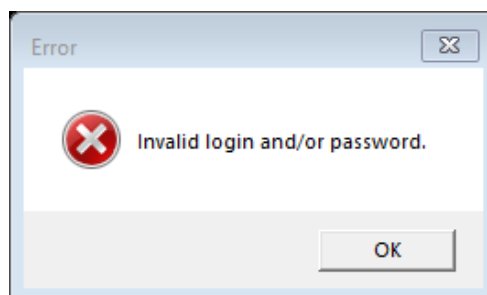


Рисунок 3.3 – Повідомлення про введення невірного паролю або логіну

При першому вході потрібно пройти реєстрацію(рисунок 3.4). Для реєстрації потрібно пройти на сайт онлайн-бази даних The Movie Database (TMDB). Після заповнення необхідних даних потрібно повернутися до додатку і ввести свій логін і пароль.

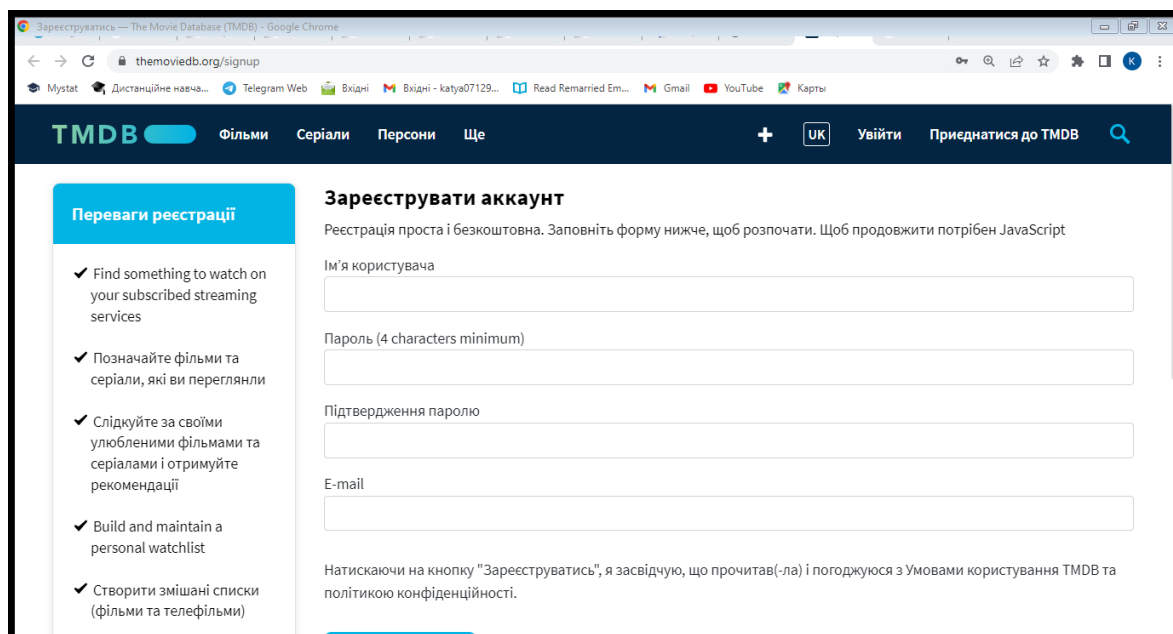


Рисунок 3.4 – Реєстрація

При випадку забуття паролю потрібно перейти на сайт онлайн-бази даних і скинути пароль.

### 3.1.3. Головна сторінка додатку

Вигляд головної сторінки зображено на рисунку 3.5.

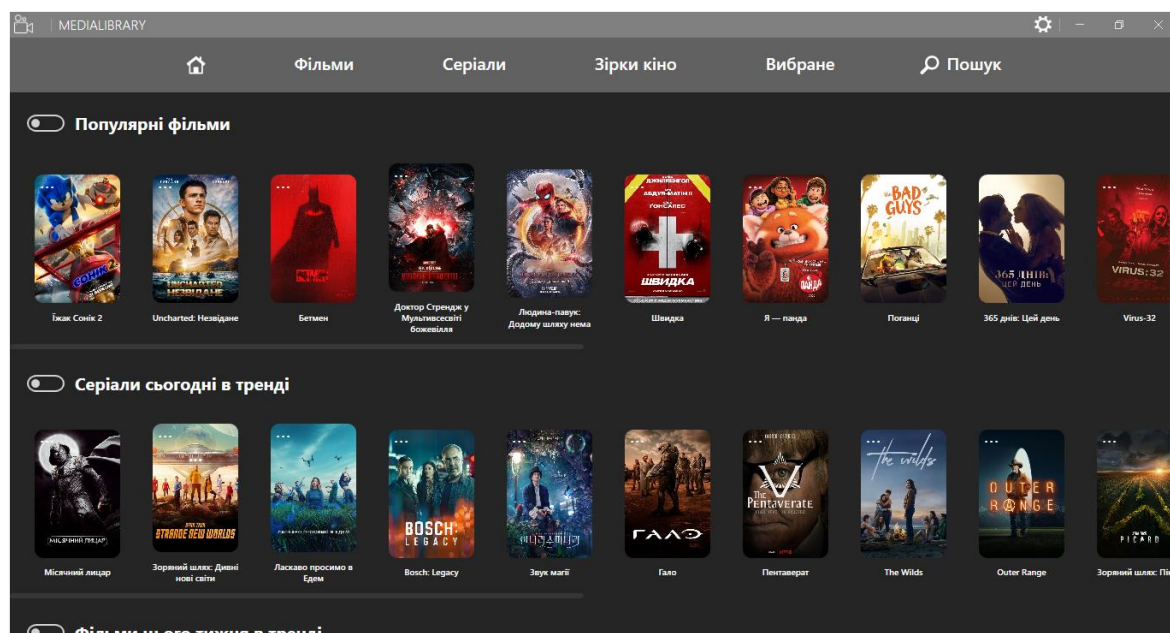


Рисунок 3.5 – Головна сторінка додатку

На головній сторінці знаходиться навігаційна панель, яка надає можливість переходу між сторінками додатку. Також на сторінці відображаються популярні фільми та серіали і тренди кіно.

### 3.1.4. Сторінка «Фільми»

Вигляд сторінки «Фільми» зображено на рисунку 3.6.

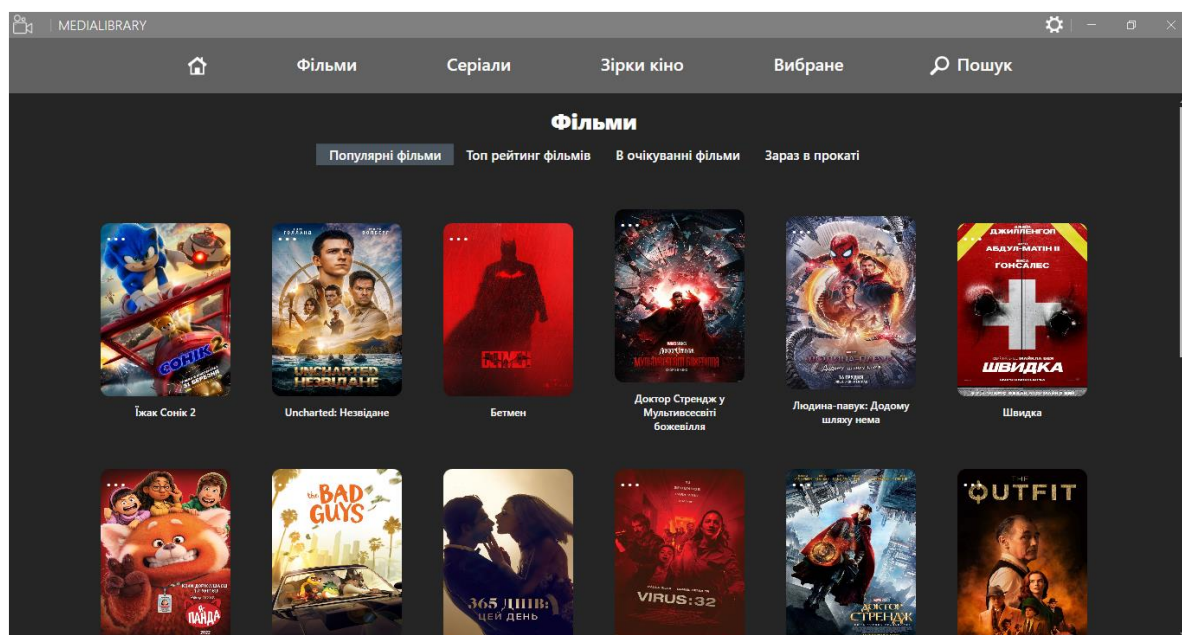


Рисунок 3.6 – Сторінка «Фільми»

На цій сторінці є можливість різні категорії фільмів. Для відображення більшої кількості фільмів знизу сторінки є кнопка «Більше».

### 3.1.5. Сторінка «Серіали»

Сторінка «Серіали» по вигляду і функціоналу схожа на сторінку «Фільми»(рисунок 3.7).

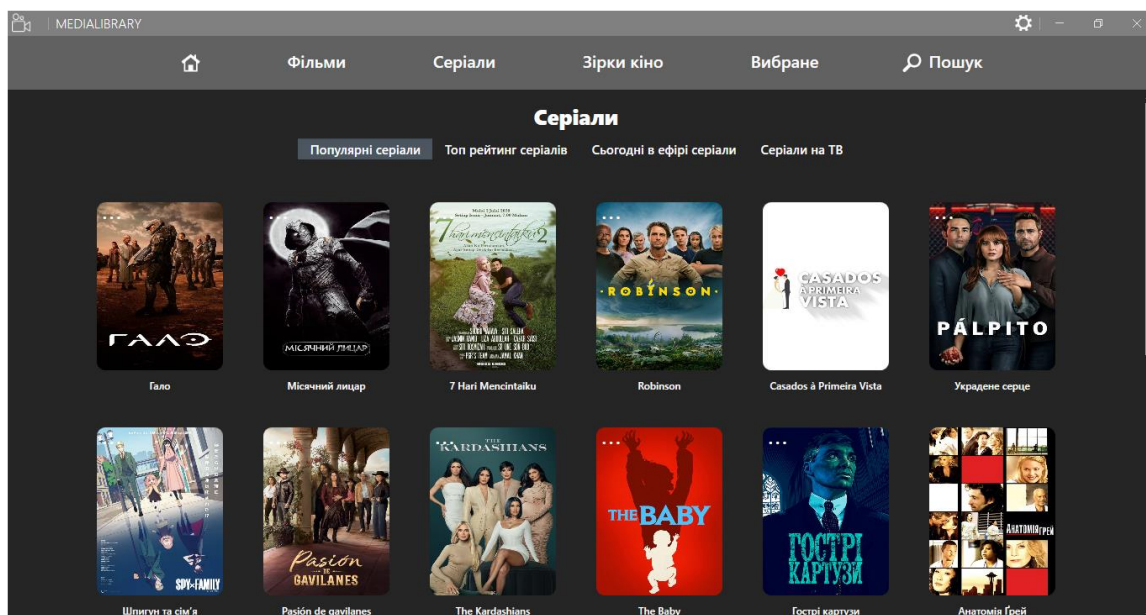


Рисунок 3.7 – Сторінка «Серіали»

### 3.1.6. Сторінка «Зірки кіно»

На сторінці «Зірки кіно» відображається список популярних акторів (рисунк 3.8).

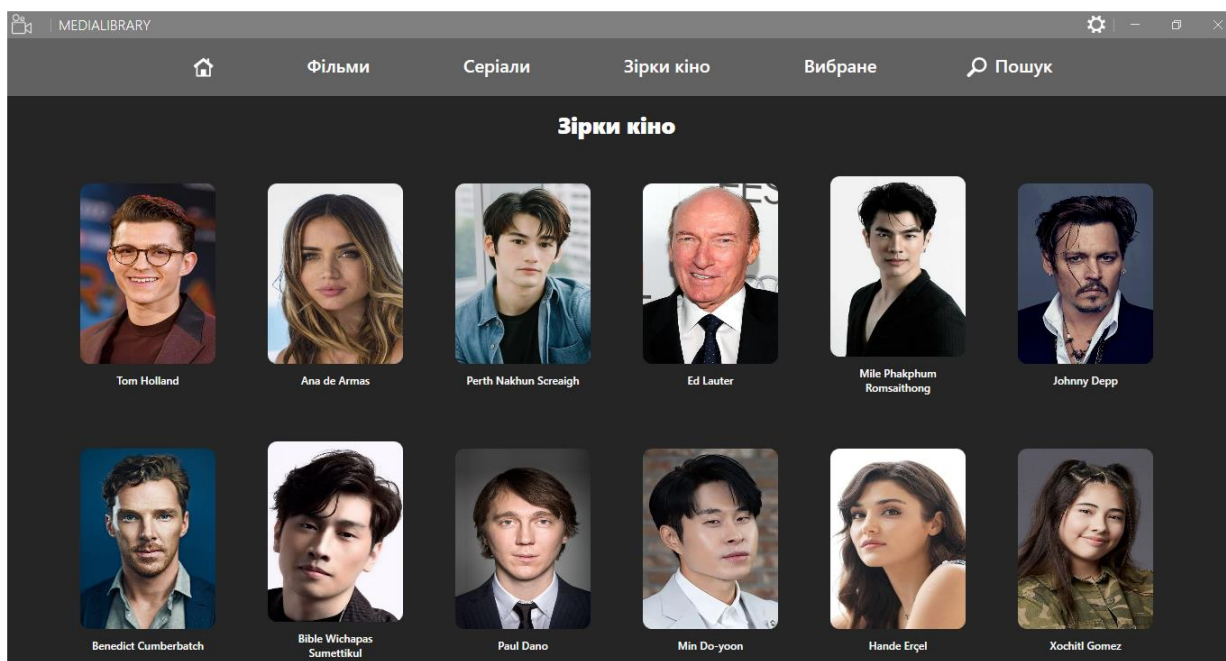


Рисунок 3.8 – Сторінка «Серіали»

Для відображення більшої кількості акторів в кінці сторінки є кнопка «Більше».

### 3.1.7. Сторінка «Вибране»

На сторінці «Вибране» відображується список вибраних користувачем фільмів та серіалів(рисунок 3.9).

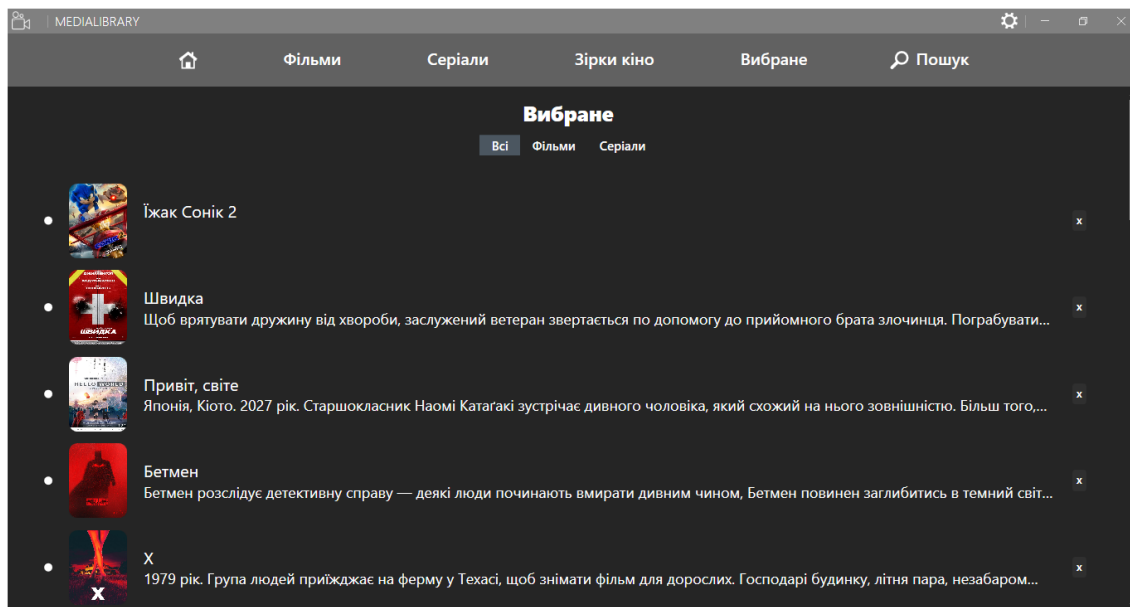


Рисунок 3.9 – Сторінка «Вибране»

Також передбачена можливість видалення елементів зі списку.

### 3.1.8. Пошук за назвою

Щоб з'явилося вікно пошуку, потрібно натиснути «Пошук» на навігаційній панелі(рисунок 3.10).

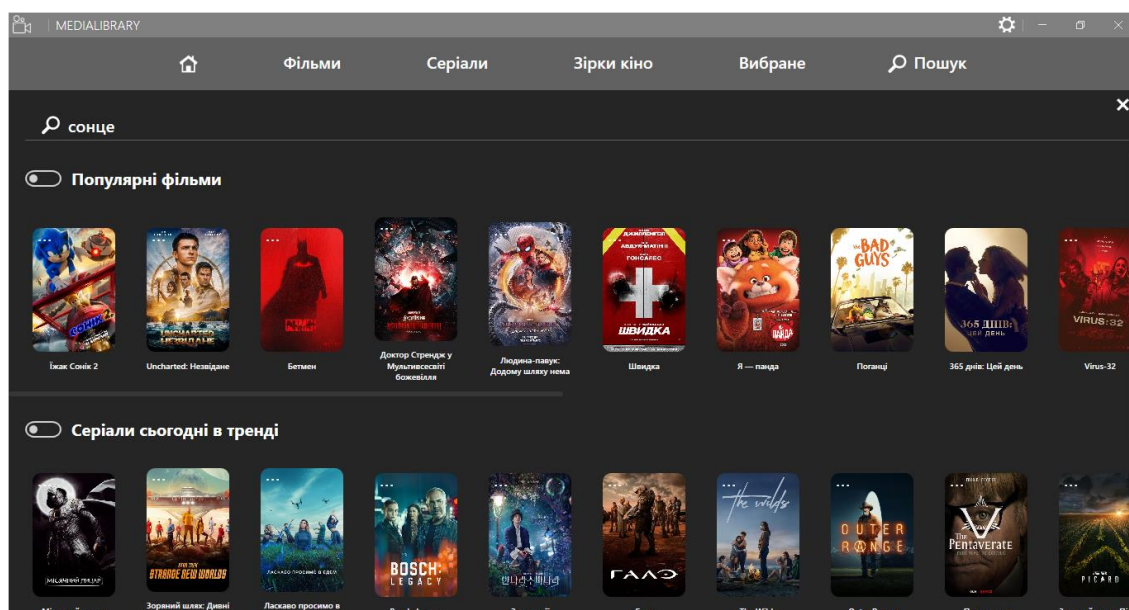


Рисунок 3.10 – Пошук за назвою

Для пошуку потрібно ввести назву фільму, серіалу, або актора і натиснути клавішу Enter. Після цього буде здійснено пошук і виведено результати(рисунок 3.11).

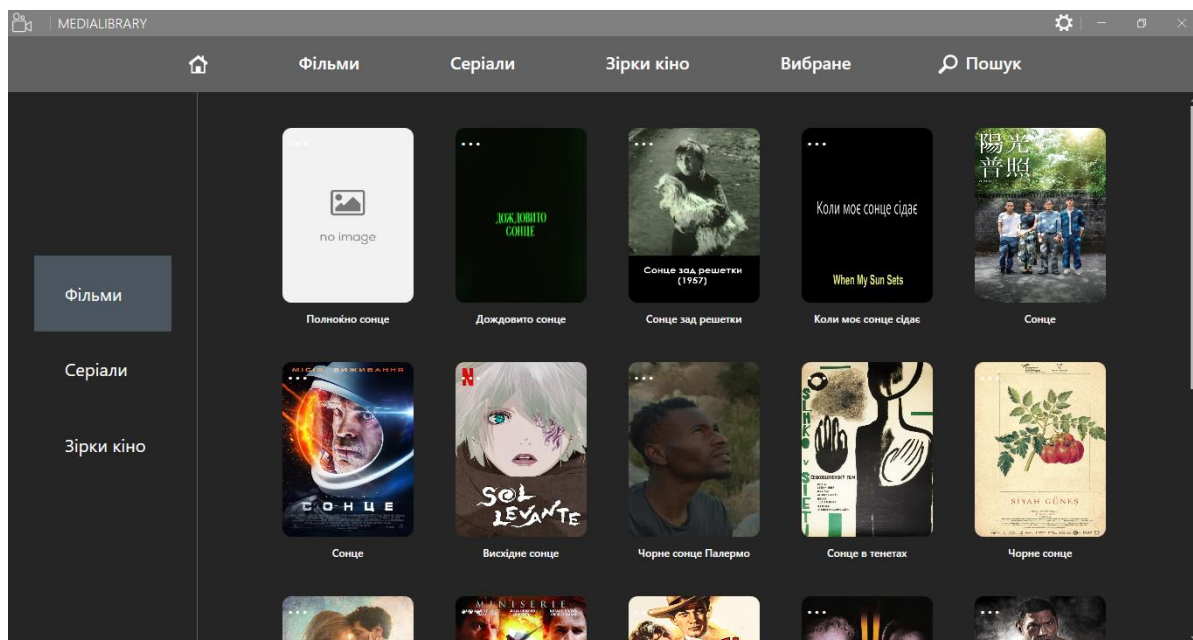


Рисунок 3.11 – Результати пошуку за назвою

### 3.1.9. Пошук фільмів за жанром та роком випуску

Щоб з'явилася панель пошуку за жанром та роком випуску, потрібно навести курсор на «Фільми» на навігаційній панелі(рисунок 3.12).

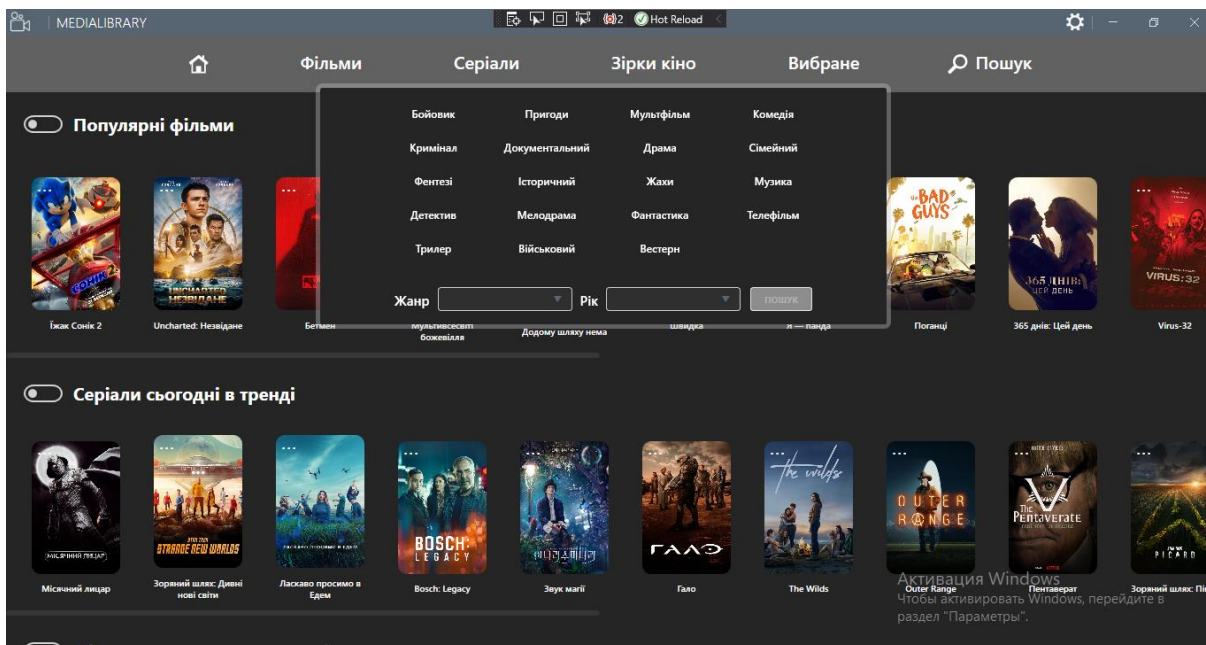


Рисунок 3.12 – Панель пошуку фільмів за жанром та роком випуску

Для пошуку фільмів за жанром можна вибрати жанр зі списку або вибрати жанр з випадаючого списку та натиснути кнопку «Пошук».

Для пошуку за роком випуску потрібно вибрати рік з випадаючого списку і натиснути пошук.

Кнопка «Пошук» буде доступна тільки в тому випадку, коли буде вибрано або жанр або рік.

Вигляд результатів пошуку зображено на рисунку 3.13.

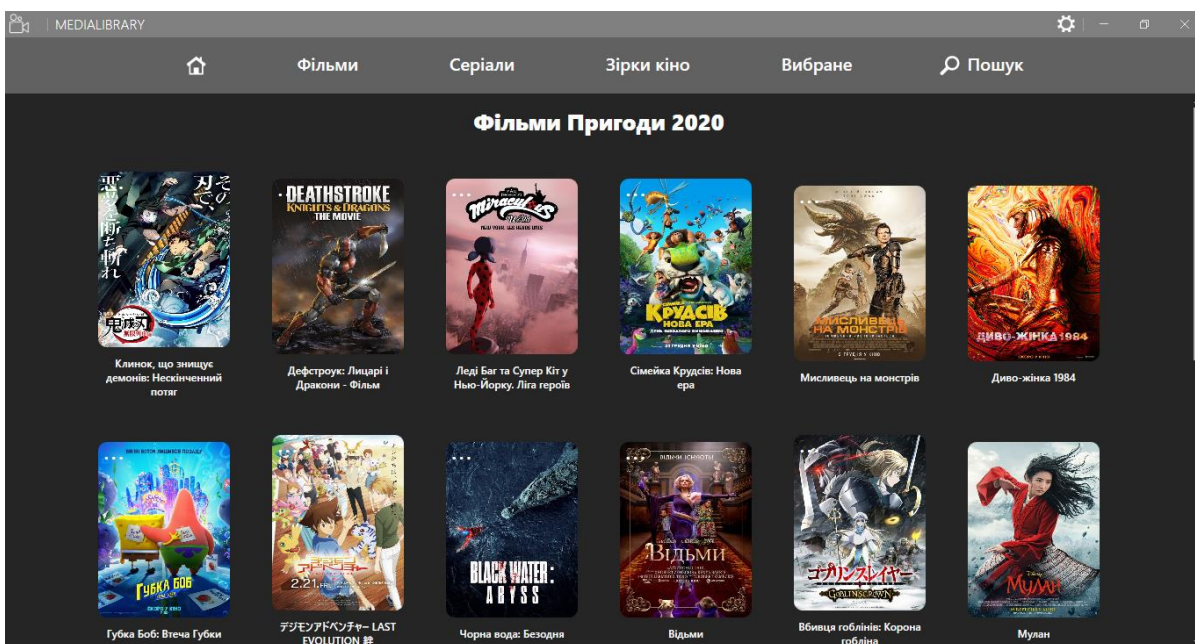


Рисунок 3.13 – Результати пошуку за жанром та роком випуску

### 3.1.10. Пошук серіалів за жанром та роком випуску

Для пошуку серіалів за жанром та роком випуску подібний до пошуку фільмів(рисунок 3.14).

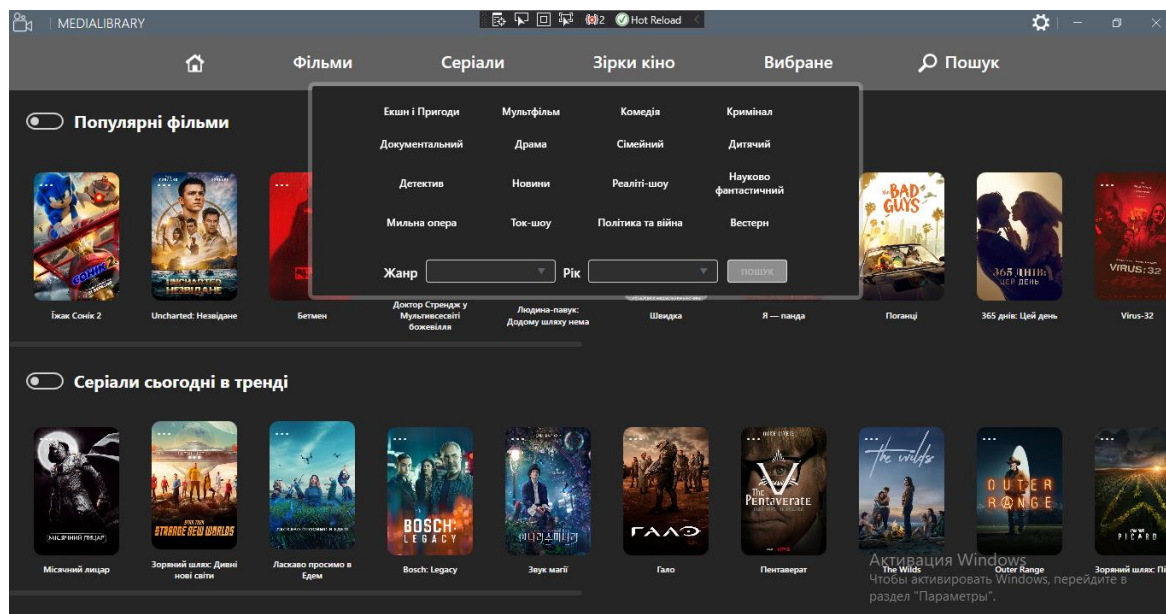


Рисунок 3.14 – Панель пошуку серіалів за жанром та роком випуску

### 3.1.11. Перегляд детальної інформації про фільм



Для отримання детальної інформації про фільм потрібно натиснути на будь-який фільм зі списку, після цього відкриється вікно(рисунок 3.15).

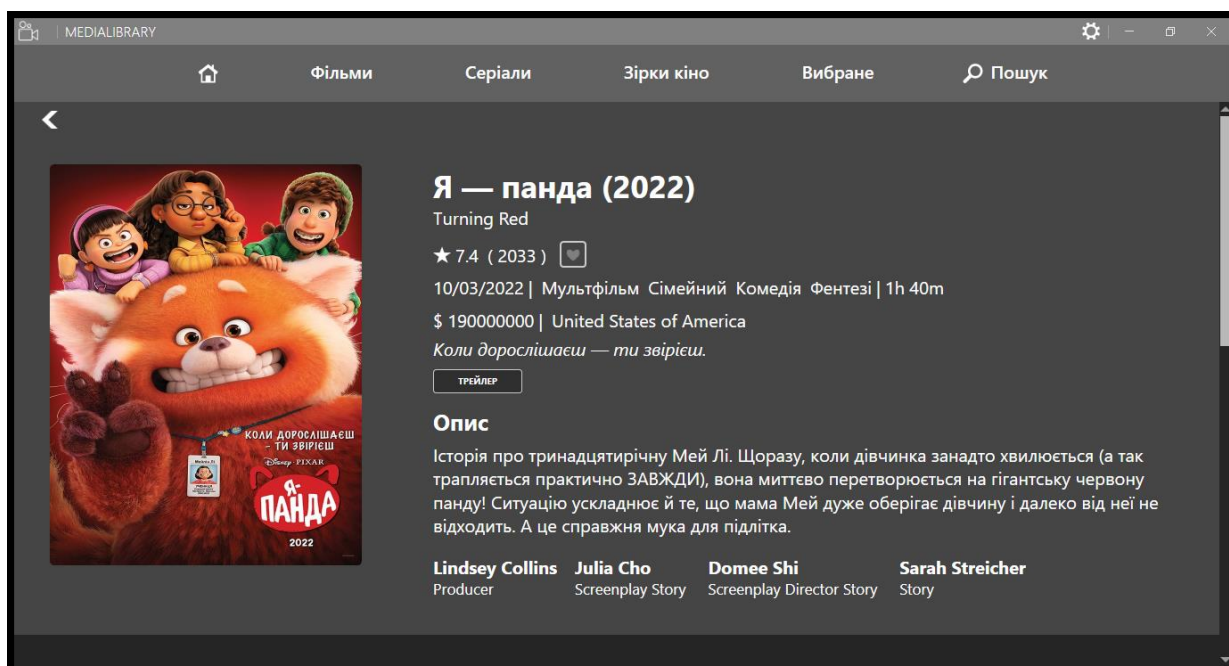


Рисунок 3.15 – Вікно детальної інформації про фільм

На цьому вікні відображається такі дані:

- постер;
- назва;
- оригінальна назва;
- рейтинг;
- кількість користувачів, які оцінили фільм;
- дата випуску;
- жанр;
- тривалість фільму;
- країна;
- слоган;
- трейлер;
- опис;
- команда зйомок;

- акторський склад(рисунок 3.16);
- схожі фільми;
- рекомендації.

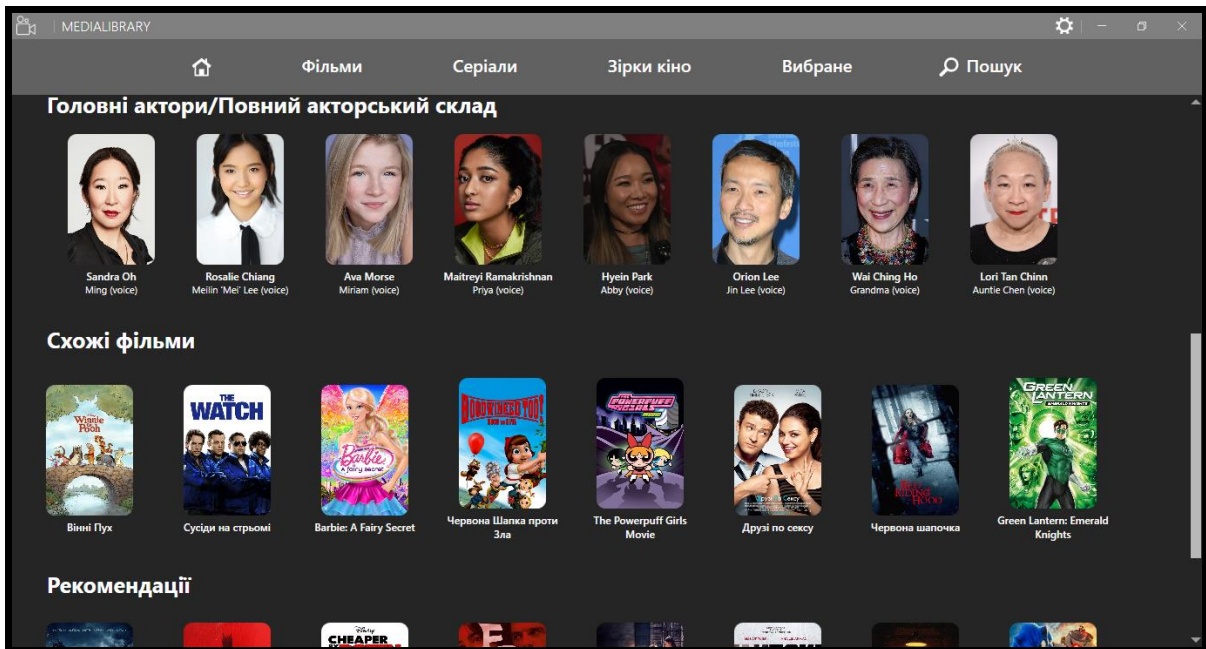


Рисунок 3.16 – Вікно детальної інформації про фільм

Також доступні функції:

- перегляд головних акторів;
- перегляд повного списку акторів;
- перегляд трейлера(рисунок 3.17);
- додавання або видалення до списку вибране;
- повернутися назад.

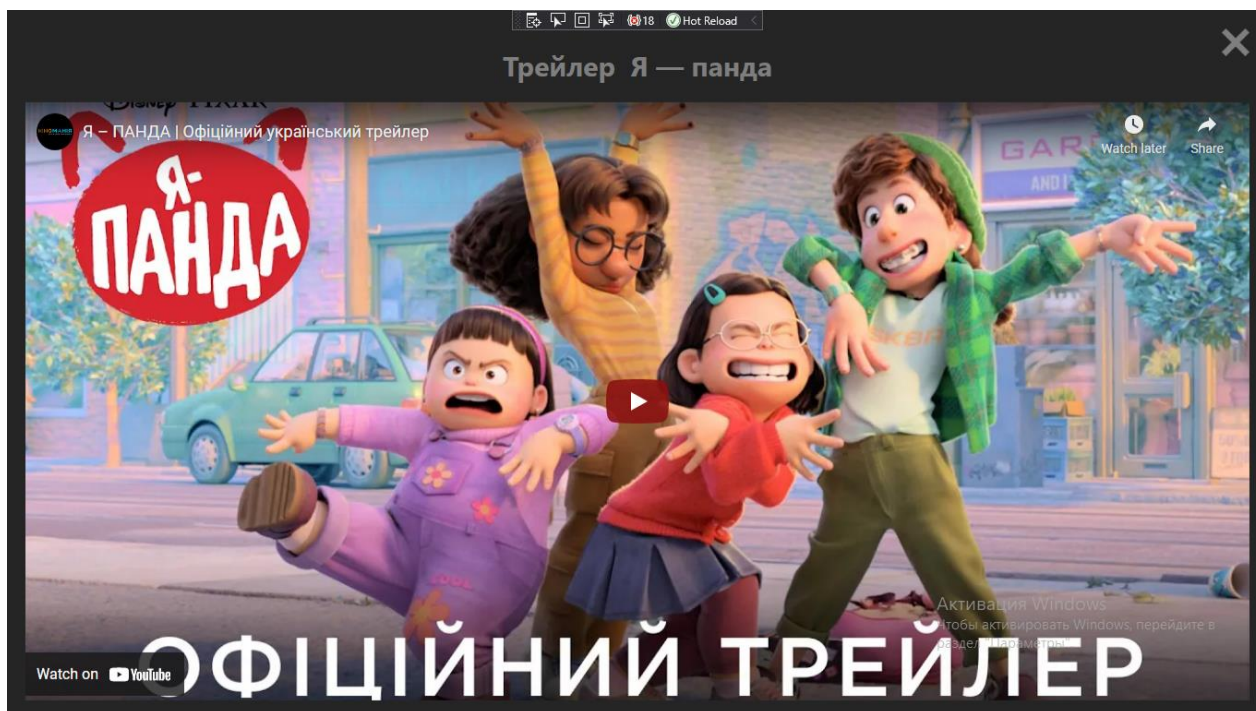


Рисунок 3.17 – Перегляд трейлеру

### 3.1.12. Перегляд детальної інформації про актора

Для отримання інформації про актора потрібно натиснути на будь-якого актора зі списку і відкриється сторінка з детальною інформацією (рисунок 3.18).



Рисунок 3.18 – Детальна інформація про актора

На цьому вікні буде надана така інформація:

- дата народження;
- коротка біографія;
- акторська діяльність, яка включає фільми та серіали(рисунок 3.19).
- 

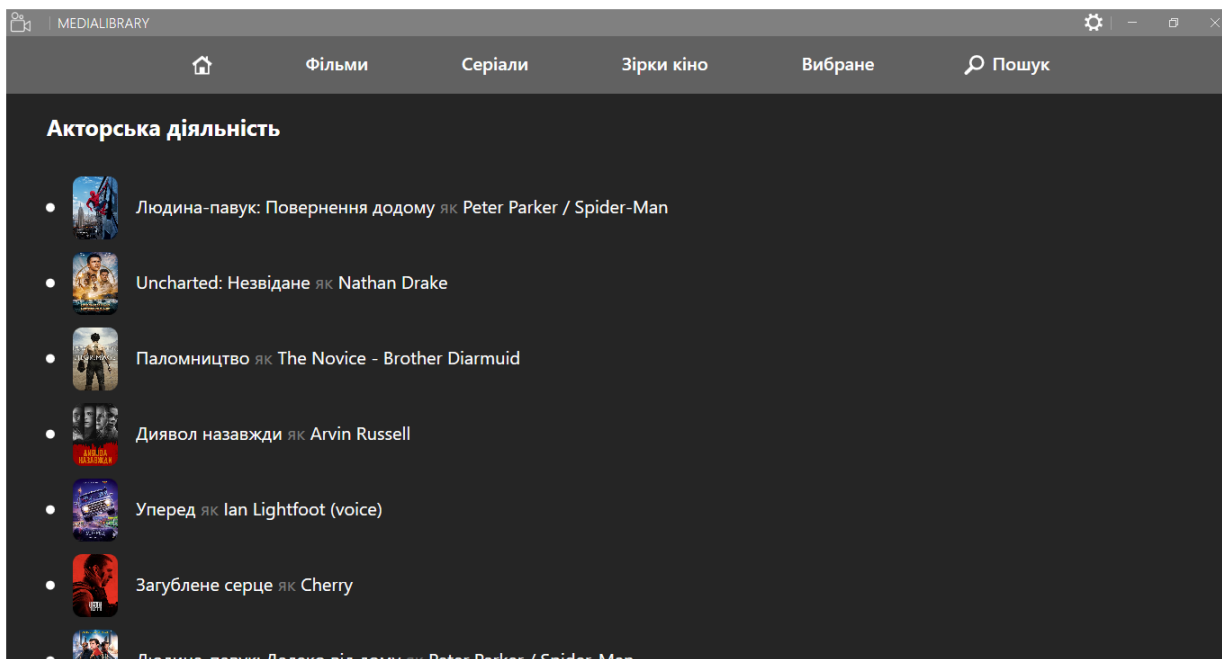


Рисунок 3.19 – Акторська діяльність актора

Також на сторінці є функція повернутися до попереднього вікна.

### 3.1.13. Перегляд детальної інформації про серіал

Для отримання детальної інформації про серіал потрібно натиснути на будь-який серіал зі списку і відкриється сторінка з детальною інформацією про серіал(рисунок 3.20).



Рисунок 3.20 – Детальна інформація про серіал

На сторінці відображається подібні дані, що на сторінці про детальну інформацію про фільм.

Також доступний функціонал:

- перегляд акторського складу(рисунок 3.21);
- перегляд випущених сезонів;
- перегляд схожих серіалів;
- перегляд рекомендацій;
- додавання або видалення серіалу зі списку вибраного;
- повернення до попередньої сторінки.

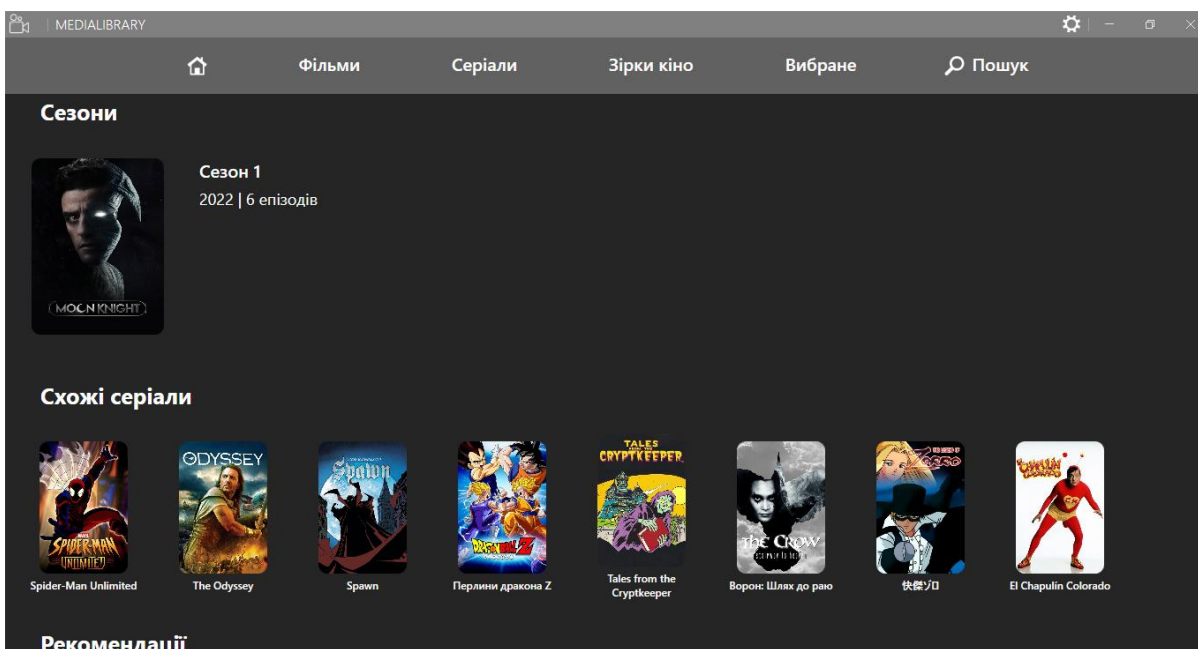


Рисунок 3.21 – Детальна інформація про серіал

### 3.1.14. Робота зі списком вибраного

Для додавання або видалення фільму або серіалу зі вибраного списку, потрібно використовувати кнопку «...», яка знаходиться в лівому кутку об'єктів зі списків(рисунок 3.22).

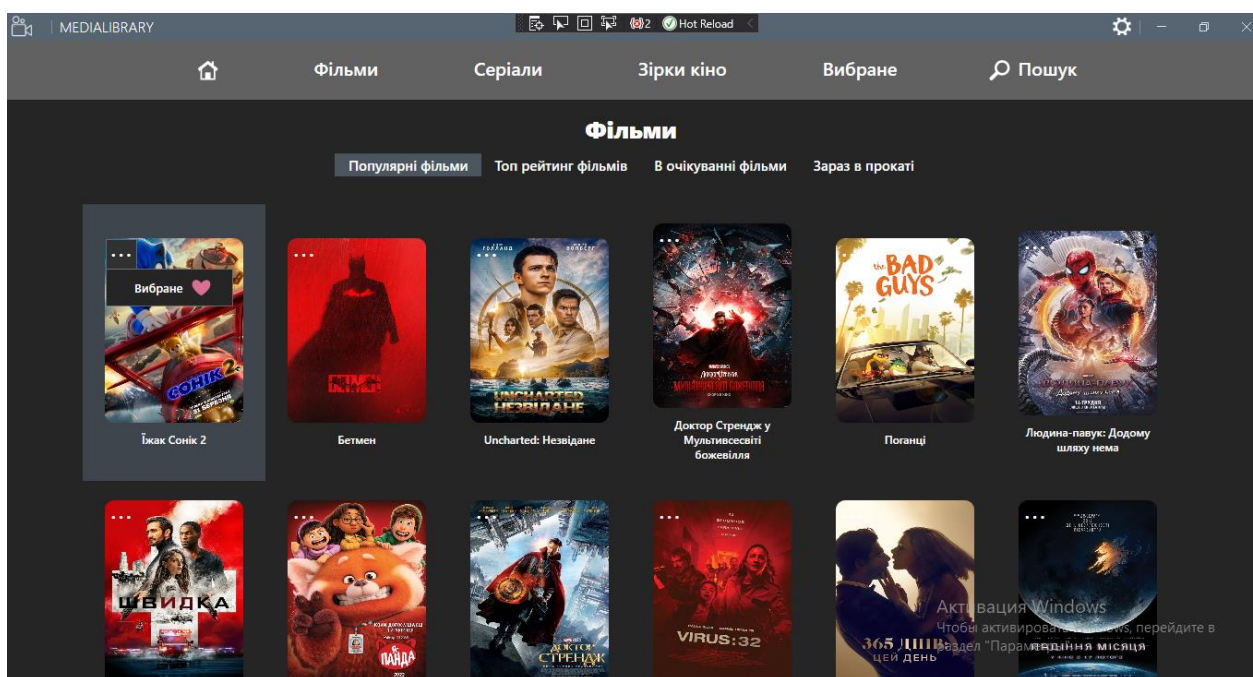


Рисунок 3.22 – Перегляд статусу фільму

Якщо об'єкт додано до списку серце буде рожевого кольору, в інакшому випадку сірого кольору. Для зміни статусу потрібно натиснути на сердечко.

### 3.1.15. Налаштування додатку

Для відкриття панелі для налаштування потрібно натиснути на кнопку з іконкою шестерня, яка знаходиться в правому кутку вікна(рисунок 3.23).

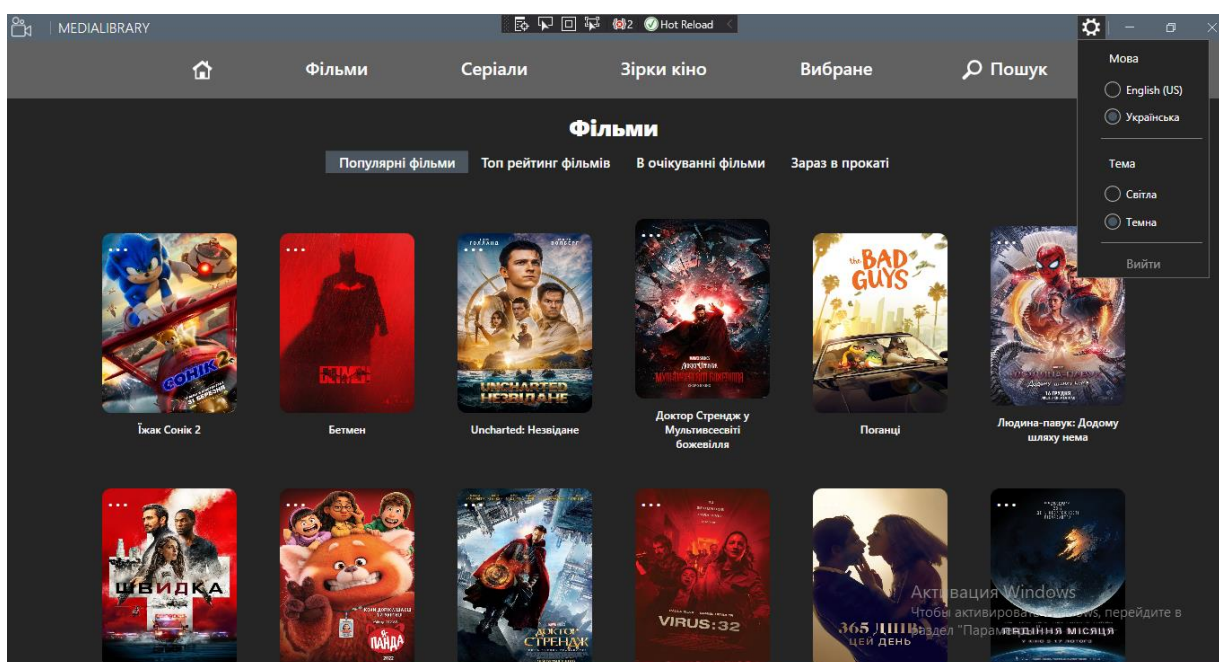


Рисунок 3.23 – Налаштування додатку

За допомогою цієї панелі можна здійснити такі функції:

- зміну мови інтерфейсу;
- зміна тема інтерфейсу(рисунок 3.24);
- вийти з акаунту.

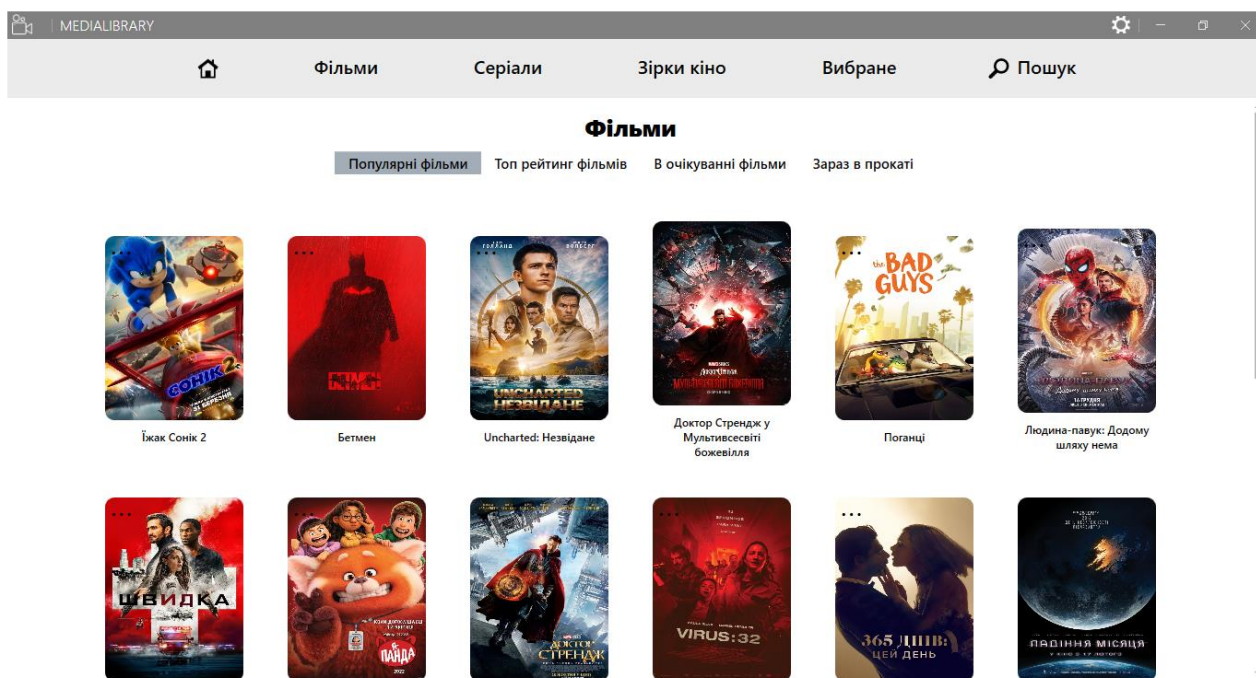


Рисунок 3.24 – Світла тема інтерфейсу

## 3.2. Тестування програмного продукту

В цьому розділі буде наведено опис тест кейсів, які дозволять забезпечити відповідність розробленого програмного продукту попередньо виставленим функціональним вимогам.

### 3.2.1. Мета тестування

Метою тестування являється перевірка відповідності програмної реалізації дипломного проекту вимогам поставленого завдання

### 3.2.2. Результати тестування

В процесі тестування було перевірено відповідність програмної реалізації комплексу задач вимогам поставленого завдання. Перелік тестувань основних функціональних можливостей програмної реалізації дипломного проекту наведено у таблиці 3.1.



Таблиця 3.1 – Тест кейси для розробленого додатку

Id	Summary	Pre-condition	Steps	Expected Result	Status
1	Авторизація	1. Запуск програми	1. Ввести логін 2. Ввести пароль 3. Натиснути кнопку «Вхід»	У разі правильно введених даних відбудеться відкриття головної сторінки проекту, в інакшому випадку – виведення повідомлення про неправильно введені дані і очищення полів для вводу паролю і логіну.	Pass
2	Реєстрація	1. Запуск програми	1. Натиснути на гіперпосилання «Реєстрація» 2. Заповнити всі необхідні поля 3. Натиснути кнопку «Реєстрація» 4. Активувати акаунт через електронну пошту 5. Повернутися до додатку 6. Ввести логін і пароль 7. Натиснути «Вхід»	Відкриття головної сторінки додатку	Pass
3	Зміна мови інтерфейсу	1. Запуск додатку	1. Натиснути на кнопку з іконкою шестерня 2. Вибрати необхідну мову	Весь інтерфейс та всі дані зміниться на вибрану мову	Pass

Продовження таблиці 3.1

4	Зміна теми інтерфейсу	1.Запуск програми	1. Натиснути на кнопку з іконкою шестерня 2.Вибрати необхідну тему інтерфейсу	Зміна теми інтерфейсу на вибрану	Pass
5	Вийти з акаунту	1.Запуск програми	1. Натиснути на кнопку з іконкою шестерня 2. Натиснути на кнопку «Вийти»	Відкриття вікна авторизації	Pass
6	Пошук за назвою	1.Запуск програми	1. На навігаційній панелі натиснути «Пошук» 2. Ввести назву для пошуку 3. Натиснути Enter	Вивід результатів пошуку	Pass
7	Пошук фільмів за жанром	1.Запуск програми	1.Навести курсор на панелі на «Фільми» 2. На відкритій панелі зі списку вибрати потрібний жанр	Відкриття сторінки зі списком фільмів вибраного жанру	Pass
8	Перегляд інформації про вибраного актора	1. Запуск	1.Натиснути «Зірки кіно» на навігаційній панелі 2. Натиснути на потрібного актора зі списку	Відкриття сторінки з детальною інформацією про актора	Pass

Продовження таблиці 3.1

9	Видалити фільм з вибраного списку	1. Запуск	1. Натиснути «Вибране» на навігаційній панелі 2. Натиснути вкладку «Фільми» 3. Натиснути на кнопку «x» навпроти потрібного фільму	Видалення вибраного фільму зі списку	Pass
10	Добавити серіал до вибраного списку	1. Запуск	1. Натиснути «Серіали» на навігаційній панелі 2. Натиснути на кнопку «...» потрібного серіалу 3. Натиснути на сердечко	Сердечко стане рожевого кольору, що означає додавання серіалу до списку	Pass
11	Перегляд трейлеру	1. Запуск	1. Натиснути на необхідний фільм або серіал 2. На відкритій сторінці на кнопку «Подивитися трейлер»	Відкриття вікна з трейлером	Pass
12	Перегляд повного акторського складу фільму	1. Запуск	1. Натиснути на необхідний фільм 2. На відкритій сторінці натиснути на надпис «Повний акторський склад»	Відображення повного списку акторів	Pass

## ВИСНОВКИ

У процесі реалізації дипломного проекту було розглянуто предметне середовище. Проект присвячений розробці десктопного додатку, який буде надавати доступ до фільмотеки для управління та роботи над нею.

При описі предметного середовища було описано процеси діяльності, також була створена функціональна модель системи, яка представлена діаграмою варіантів використання. Також створено діаграми станів та діяльності для розуміння логіки роботи програми.

Був проведений пошук та аналіз функцій найвідоміших аналогів, на основі якого були зроблені висновки про те, що жоден з проаналізованих аналогів не задовольняє усім вимогам.

Також було сформовано постановку задачі (зокрема призначення та цілі розробки).

В розділі з описом розробки програмного забезпечення було розглянуто засоби розробки, які використовувалися у процесі роботи над дипломним проектом.

В підрозділі структура програмного забезпечення були наведені ключові класи програмного продукту, та пояснено для чого вони використовуються, та чому була обрана саме така архітектурна реалізація.

В технологічному розділі було описано послідовність дій, які може виконувати користувач, а також наведено набір тест-кейсів, проведення яких дозволили впевнитися у тому, що розроблений програмний продукт відповідає попередньо виставленим програмним вимогам та вимогам технічного завдання

## ПЕРЕЛІК ПОСИЛАНЬ

1. Історія кіно - Все про кіно. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://sites.google.com/site/kinovsepro/istoria-kino>  
Дата звернення: 01.05.2022.
2. Історія кіно: від зародження до сучасності - Наука - 2022. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://uk.warbletoncouncil.org/historia-del-cine-2420> Дата звернення: 01.05.2022.
3. Моделювання програмного забезпечення. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу <https://evergreens.com.ua/ua/articles/uml-diagrams.html> Дата звернення: 03.05.2022.
4. What is Unified Modeling Language (UML)? - Visual Paradigm. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> Дата звернення: 03.05.2022.
5. Середовище та мова програмування - Студопедия. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: [https://studopedia.ru/19\\_316706\\_seredovishche-ta-mova-programuvannya.html](https://studopedia.ru/19_316706_seredovishche-ta-mova-programuvannya.html)  
Дата звернення: 03.05.2022.
6. Visual Studio Community - The complete C# tutorial. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://csharp.net-tutorials.com/uk/371/getting-started/visual-studio-community/> Дата звернення: 04.05.2022.
7. C# Tutorial (C Sharp) - W3Schools. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.w3schools.com/cs/index.php>  
Дата звернення: 04.05.2022.
8. What is Model-View-ViewModel (MVVM)? - TechTarget. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу:

<https://www.techtarget.com/whatis/definition/Model-View-ViewModel> Дата звернення: 04.05.2022.

9. Основи тестування програмного забезпечення | Prometheus. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: [https://courses.prometheus.org.ua/courses/course-v1:LITS+115+2017\\_T4/about](https://courses.prometheus.org.ua/courses/course-v1:LITS+115+2017_T4/about) Дата звернення: 10.05.2022.
10. Види Тест Кейсів. Стани Тест Кейсів. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://qalearning.com.ua/theory/lectures/material/test-cases-bugtracking-systems-severity-priority/> Дата звернення: 10.05.2022.
11. Окупнярек К. О. Розробка програмного забезпечення для управління фільмотекою за допомогою мови програмування с#. *Застосування програмного забезпечення в інфокомунікаційних технологіях*: матеріали Наук.-тех. конф., м. Київ, 20—22 бер. 2022 р. / Держ. ун-т телекомунікацій, 2022. С. 20-23.
12. Окупнярек К. О. Розробка програмного забезпечення для управління фільмотекою за допомогою мови програмування с#. *Наукова молодь - 2021*: матеріали ІХ Всеукр. Наук.-практ. конф., м. Київ, 27—30 лист. 2021 р. / Ін-т геохімії н.с. НАН України, Ін-т інформ. технологій і засобів навчання НАПН України. Київ, 2021. С. 122—123.

## Додаток А

### Лістинг програми для відображення акторів

#### PersonInfo.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MediaLibrary.Models
{
    class PersonInfo
    {
        private string profile_path;
        public int Id { get; set; }
        public string Name { get; set; }
        public string Profile_path
        {
            get => profile_path;
            set
            {
                if (value != null)
                {
                    profile_path = "https://image.tmbd.org/t/p/w600_and_h900_bestv2" + value;
                }
                else
                {
                    profile_path = "https://nuft.edu.ua/assets/images/people/no-image.jpg";
                }
            }
        }
        public string Biography { get; set; }
        public string Birth_day { get; set; }
        public string Place_of_birth { get; set; }
        public List<Item> Acting { get; set; }
    }
}

```

#### PersonInfoView.xaml

```

<UserControl x:Class="MediaLibrary.Views.PersonInfoView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:vm="clr-namespace:MediaLibrary.ViewModels"
    xmlns:i="http://schemas.microsoft.com/expression/2010/interactivity"
    xmlns:iconPacks="http://metro.mahapps.com/winfx/xaml/iconpacks"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:MediaLibrary.Views"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800">

    <UserControl.DataContext>
        <vm:PersonInfoViewModel></vm:PersonInfoViewModel>
    </UserControl.DataContext>

    <Grid>

```

```

<ScrollViewer Name="scroll" VerticalScrollBarVisibility="Auto">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="auto"></ColumnDefinition>
      <ColumnDefinition></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="auto"></RowDefinition>
      <RowDefinition Height="auto"></RowDefinition>
      <RowDefinition></RowDefinition>
    </Grid.RowDefinitions>
    <Grid.Resources>
      <Style TargetType="TextBlock">
        <Setter Property="FontSize" Value="20"></Setter>
        <Setter Property="Padding" Value="5 5 5 5"></Setter>
      </Style>
    </Grid.Resources>
    <StackPanel Orientation="Horizontal" Grid.Row="0" Grid.ColumnSpan="2"
Background="{DynamicResource MahApps.Brushes.Gray8}">
      <i:Interaction.Triggers>
        <i:EventTrigger EventName="PreviewMouseLeftButtonDown">
          <i:InvokeCommandAction Command="{Binding BackCommand}"/>
        </i:EventTrigger>
      </i:Interaction.Triggers>
      <iconPacks:PackIconModern Width="20"
        Height="20"
        Cursor="Hand"
        Margin="30 10 10 0"
        Foreground="White"
        Background="Transparent"
        VerticalAlignment="Center"
        Kind="ChevronLeft" />
    </StackPanel>
    <Border Grid.Row="1" Grid.ColumnSpan="2" Background="{DynamicResource
MahApps.Brushes.Gray8}" />
    <StackPanel Grid.Row="1" Margin="40" Grid.Column="0">
      <Border Width="350" Height="450" CornerRadius="5" VerticalAlignment="Center">
        <Border.Background>
          <ImageBrush Stretch="Fill" ImageSource="{Binding
Person.Profile_path}"/>
        </Border.Background>
      </Border>
    </StackPanel>
    <StackPanel Margin="40" Grid.Row="1" Grid.Column="1">
      <TextBlock FontSize="35" TextWrapping="Wrap" FontWeight="Bold" Text="{Binding
Person.Name}"></TextBlock>
      <TextBlock Margin="0 10 0 0">
        <Run FontWeight="Medium" Text="{DynamicResource Birthday}"></Run>
        <Run Text="{Binding Person.Birth_day,
StringFormat='d/MM/yyyy'}"></Run>
      </TextBlock>
      <TextBlock>
        <Run FontWeight="Medium" Text="{DynamicResource Place}"></Run>
        <Run Text="{Binding Person.Place_of_birth}"></Run>
      </TextBlock>
      <TextBlock Margin="0 7 0 0" TextWrapping="Wrap" Text="{Binding
Person.Biography}"></TextBlock>
    </StackPanel>
    <StackPanel Margin="0 0 0 20" Grid.Row="2" Grid.ColumnSpan="2">
      <TextBlock Text="{DynamicResource Acting}" Margin="40 15 0 10" FontSize="25"
FontWeight="Bold"></TextBlock>
      <StackPanel Margin="40 15 0 10">

```



```

SelectedMovie}" >
    <ListView ItemsSource="{Binding Person.Acting}" SelectedItem="{Binding
    <ListView.ItemsPanel>
        <ItemsPanelTemplate>
            <StackPanel></StackPanel>
        </ItemsPanelTemplate>
    </ListView.ItemsPanel>
    <ListView.ItemTemplate>
        <DataTemplate>
            <StackPanel>
                <i:Interaction.Triggers>
                    <i:EventTrigger
EventName="PreviewMouseLeftButtonDown">
                        <i:InvokeCommandAction Command="{Binding
DataContext.DisplayMovieInfoCommand, RelativeSource={RelativeSource
Mode=FindAncestor,AncestorType=UserControl}}">
                            CommandParameter="{Binding}"/>
                        </i:EventTrigger>
                    </i:Interaction.Triggers>
                <StackPanel Cursor="Hand" Orientation="Horizontal"
Margin="0 7 0 5">
                    <iconPacks:PackIconModern Width="10"
Height="10"
Cursor="Hand"
Foreground="White"
Background="Transparent"
VerticalAlignment="Center"
Kind="MoonFull" />
                    <Border Margin="20 0 0 0" Width="50" Height="70"
CornerRadius="10 ">
                        <Border.Background>
                            <ImageBrush Stretch="Fill"
ImageSource="{Binding Poster_path}"/>
                        </Border.Background>
                    </Border>
                    <TextBlock Margin="20" FontSize="20">
                        <Run Text="{Binding Title}"></Run>
                        <Run Foreground="{DynamicResource
MahApps.Brushes.Gray}"
                            Text="{DynamicResource As}"></Run>
                        <Run Text="{Binding Character}"></Run>
                    </TextBlock>
                </StackPanel>
            </StackPanel>
        </DataTemplate>
    </ListView.ItemTemplate>
</ListView>
</StackPanel>
<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">
    <i:Interaction.Triggers>
        <i:EventTrigger EventName="PreviewMouseLeftButtonDown">
            <i:InvokeCommandAction Command="{Binding ScrollUpCommand}"/>
        </i:EventTrigger>
    </i:Interaction.Triggers>
    <iconPacks:PackIconModern Width="20"
Height="20"
Cursor="Hand"
Margin="30 10 10 0"
Foreground="White"
Background="Transparent"
VerticalAlignment="Center"
Kind="Upload" />

```

```

        </StackPanel>
    </StackPanel>
</Grid>
</ScrollView>

</Grid>
</UserControl>

```

## PersonInfoViewModel.cs

```

using MediaLibrary.Infrastructure;
using MediaLibrary.Models;
using MediaLibrary.Views;
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Controls;
using System.Windows.Input;

namespace MediaLibrary.ViewModels
{
    class PersonInfoViewModel : BaseNotifyPropertyChanged
    {
        private MainViewModel main;
        private Switcher switcher;
        private int id;
        private UserControl previousView;
        public UserControl PreviousView
        {
            get => previousView;
            set
            {
                previousView = value;
                NotifyProperty();
            }
        }
        private PersonInfoView view;
        public PersonInfoView View
        {
            get => view;
            set
            {
                view = value;
                NotifyProperty();
            }
        }
        public int Id
        {
            get => id;
            set
            {
                id = value;
                NotifyProperty();
                Person = Api.GetPerson(Id);
            }
        }
        private PersonInfo person;
        public PersonInfo Person
        {

```

```

        get => person;
        set
        {
            person = value;
            NotifyProperty();
        }
    }

    public PersonInfoViewModel()
    {
        InitCommand();
    }

    private void InitCommand()
    {
        BackCommand = new RelayCommand(x => RedirectToBack());
        DisplayMovieInfoCommand = new RelayCommand(RedirectToMovieInfo);
        ScrollUpCommand = new RelayCommand(x => View.scroll.ScrollToHome());
    }

    public ICommand BackCommand { get; set; }
    public ICommand DisplayMovieInfoCommand { get; set; }
    public ICommand ScrollUpCommand { get; set; }

    public void RedirectToBack()
    {
        main = (MainViewModel)System.Windows.Application.Current.MainWindow.DataContext;
        switcher = new Switcher(main);
        switcher.Switch(previousView);
        previousView = null;
    }
    public void RedirectToMovieInfo(object obj)
    {
        main = (MainViewModel)System.Windows.Application.Current.MainWindow.DataContext;
        switcher = new Switcher(main);
        Item item = obj as Item;
        if(item.Media_type=="movie")
        {
            switcher.SwitchToMovieInfo(item.Id);
        }
        else
        {
            switcher.SwitchToSerialInfo(item.Id);
        }
        View.scroll.ScrollToHome();
    }
}
}
}

```