

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук


МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:


«Інформаційна технологія надання рекомендацій щодо підбору фільмів»

Виконала: студентка 2-го курсу, групи
КН-21м

Спеціальності 122 – «Комп'ютерні
науки»



Шевчук О. Р.
(прізвище та ініціали)

Керівник: к.т.н., ст. викл. каф. КН

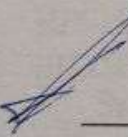

Озеранський В. С.
(прізвище та ініціали)

«15» 12 2022 р.

Опонент: к.т.н., доц каф. КСУ


Юхимчук М.С.
(прізвище та ініціали)

«15» 12 2022 р.


Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А. А.

(прізвище та ініціали)

«16» 12 2022 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Системи штучного інтелекту

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
д.т.н., проф. Яровий А.А.

(підпис)

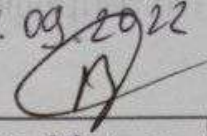

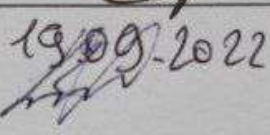
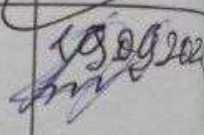
"15" 09 2022 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ
Шевчук Олені Русланівні

(прізвище, ім'я, по батькові)

1. Тема роботи «Інформаційна технологія надання рекомендацій щодо підбору фільмів»
керівник роботи к.т.н., ст. викл. каф. КН, Озеранський В. С.
затверджені наказом вищого навчального закладу від «14» 09 2022 року №203
2. Строк подання студентом роботи 18 листопада 2022 року
3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтоване програмування (ООП); мінімальна кількість питань до користувача – 6 ум. од.; мінімальна кількість записів в базі даних – 30 од.; тип інтерфейса користувача - інтуїтивно зрозумілий.
4. Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології надання рекомендацій щодо підбору фільмів підбору фільмів, моделювання інформаційної технології надання рекомендацій щодо підбору підбору фільмів, програмна реалізація інформаційної технології надання рекомендацій щодо підбору фільмів, економічна частина, висновки, список використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): модель дерева рішень для вибору фільмів; схема загального алгоритму функціонування системи; схема алгоритму формування запитань; структура інформаційної технології; діаграма прецедентів системи; UML-діаграма класів; приклад роботи програми.

6. Консультанти розділів роботи

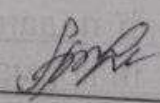
Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Озеранський О. В., к.т.н., ст. викл. каф. КН	14.09.2022 	14.09.2022 
4	Нікіфорова Л. О., к.е.н, доц. каф. ЕПВМ	19.09.2022 	19.09.2022 

7. Дата видачі завдання 14.09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування доцільності розробки	14.09.2022 - 01.10.2022	Розділ 1
2	Моделювання інформаційної технології надання рекомендацій щодо підбору фільмів	02.10.2022 - 16.10.2022	Розділ 2
3	Програмна реалізація інформаційної технології надання рекомендацій щодо підбору фільмів	17.10.2022 - 07.11.2022	Розділ 3
4	Підготовка економічної частини	08.11.2022 - 21.11.2022	Розділ 4
5	Апробація та/або впровадження результатів дослідження	23.11.2022 - 01.12.2022	тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.2022 - 14.12.2022	Пояснювальна записка, графічний матеріал, презентація

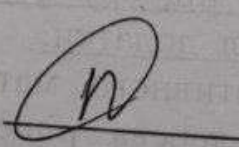
Студентка


(підпис)

Шевчук О.Р.

(прізвище та ініціали)

Керівник роботи


(підпис)

Озеранський В. С.

(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.42

Шевчук О. Р. Інформаційна технологія надання рекомендацій щодо підбору фільмів. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 111 с.

На укр. мові. Бібліогр.: 43 назв; рис.: 18; табл. 9.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології надання рекомендацій щодо підбору фільмів. Виконано аналіз сучасних програм-аналогів, які використовуються для підбору фільмів, наведено коротку порівняльну характеристику знайдених програм-аналогів, досліджено методи, які можуть бути використані для реалізації поставленої задачі. Проведено обґрунтування моделі підбору фільмів. Як базовий метод запропоновано застосувати метод дерева прийняття рішень. Розроблено алгоритм підбору фільмів та відповідне програмне забезпечення на мові програмування Java, в середовищі IntelliJ IDEA. Аналіз роботи програмного забезпечення показав підвищення ефективності прийняття рішень щодо вибору фільмів.

Графічна частина складається з 6 плакатів із результатами моделювання.

У розділі економічної частини здійснено оцінювання комерційного потенціалу розробки інформаційної технології надання рекомендацій щодо підбору фільмів, проведено оцінювання комерційного потенціалу розробки, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 478 167,3 грн, розраховано період окупності – 0,69 року.

Ключові слова: інформаційна технологія, прийняття рішень, дерева рішень.

ABSTRACT

Shevchuk OR Information technology for providing recommendations on movie selection. Master's thesis in the specialty 122 - Computer Science, educational program - Artificial Intelligence Systems. Vinnytsia: VNTU, 2023. 111 p.

In Ukrainian language. Bibliogr .: 43 titles; fig .: 18; table 9.

This master's thesis is devoted to the development of information technology for the selection of films. The analysis of modern analog programs used for the selection of films is performed, a brief comparative description of the found analog programs is given, the methods that can be used to implement the task are investigated. The substantiation of the film selection model is substantiated. As a basic method, it is proposed to use the decision tree method. A film selection algorithm and corresponding software in the Java programming language have been developed in the IntelliJ IDEA environment. Analysis of the software showed an increase in the effectiveness of decision-making on the choice of films.

The graphic part consists of 6 posters with simulation results.

In the section of the economic part the assessment of the commercial potential of information technology development of film selection was carried out, the commercial potential of development was assessed, the costs of scientific work and implementation of results were estimated, which amounted to 478 167,3 UAH, the payback period is calculated to be 0.69 years.

Keywords: information technology, decision making, decision trees.

ЗМІСТ

ВСТУП	6
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ ФІЛЬМІВ.....	9
1.1 Обґрунтування використання методів штучного інтелекту для підбору фільмів.....	9
1.2 Існуючі моделі та методи, що застосовуються для реалізації поставленої задачі.....	13
1.3 Аналіз відомих програмних рішень надання рекомендацій щодо підбору фільмів.....	21
1.4 Постановка задачі дослідження.....	28
1.5 Висновок до розділу 1	29
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ ФІЛЬМІВ.....	30
2.1 Обґрунтування вибору базового методу надання рекомендацій щодо підбору фільмів	30
2.2 Розробка математичної моделі надання рекомендацій щодо підбору фільмів.....	35
2.3 Розробка алгоритмів функціонування інформаційної технології надання рекомендацій щодо підбору фільмів.....	39
2.4 Висновок до розділу 2	43
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ ФІЛЬМІВ.....	44
3.1 Розробка структури інформаційної технології надання рекомендацій щодо підбору фільмів	44
3.2 Обґрунтування вибору мови та середовища програмування.....	47
3.3 Обґрунтування вибору бібліотек та фреймворків для розробки інформаційної технології надання рекомендацій щодо підбору фільмів	50

3.4	Опис класів і функцій інформаційної технології надання рекомендацій щодо підбору фільмів	54
3.5	Тестування інформаційної технології надання рекомендацій щодо підбору фільмів.....	56
3.6	Висновок до розділу 3	60
4	ЕКОНОМІЧНА ЧАСТИНА	61
4.1	Проведення комерційного та технологічного аудиту науково-технічної розробки	61
4.2	Розрахунок витрат на здійснення науково-дослідної роботи.....	66
4.2.1	Витрати на оплату праці	65
4.2.2	Відрахування на соціальні заходи	68
4.2.3	Амортизація обладнання, програмних засобів та приміщень	69
4.2.4	Паливо та енергія для науково-виробничих цілей	70
4.2.5	Інші витрати	71
4.2.6	Накладні (загальновиробничі) витрати	72
4.3	Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	73
4.4	Розробка чи суттєве вдосконалення інформаційної технології для використання масовим споживачем.....	74
4.5	Висновок до розділу 4	78
	ВИСНОВКИ.....	79
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	82
	Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі UNICHECK	85
	Додаток Б (обов'язковий) Лістинг програми	86
	Додаток В (обов'язковий) Ілюстративна частина.....	97
	Додаток Г (довідниковий) Інструкція користувача	104
	Додаток Д Акт впровадження.....	107

ВСТУП

Актуальність теми. На даний час існує проблема інформаційного перевантаження, що перешкоджає пошуку необхідного контенту для користувачів, а також проблема пріоритетності і персоналізації, що може бути вирішена за допомогою системи надання рекомендацій.

Кіноіндустрія на сьогоднішній день – важлива складова сфери культури. Кіно допомагає особистості відгородитися від проблем, зануритися в іншу реальність. У ХХІ столітті кіно – не лише спосіб розважитися, а може бути джерелом для початку мислення, аналізу, а також пізнання.

У даний час знято безліч фільмів і справжньою проблемою стає вибір фільму для перегляду. Збір інформації з різних сайтів та сервісів займає багато часу та потребує систематизації різноманітних даних, щоб зробити процес підбору фільмів максимально простим та гнучким. Для вирішення цієї проблеми послужить інформаційна технологія вибору фільмів, яка допоможе користувачеві швидко підібрати фільм для перегляду відповідно до заданих умов та побажань.

Об'єм даних у світі зростає, і швидкість їх зростання перевищує швидкість зростання ресурсів, необхідних для обробки цих даних. Сьогодні людство повністю поглинуте інформаційним простором: фільми, книги, статті, новини тощо, і важко визначити, який продукт більше підходить конкретній людині. Відповідно, з'явилась потреба в технологіях, які можуть оперативнo обробляти великі обсяги даних та виділяти лише ту інформацію, що є корисною для конкретного користувача. Одними з найпотужніших технологій вирішення таких задач є системи надання рекомендацій та експертні системи.

Для таких задач варто використовувати інформаційну технологію надання рекомендацій щодо підбору фільмів. Ця система зможе аналізувати усі вподобання і побажання користувача, та параметри для вибору найкращого фільму. Актуальність теми дослідження полягає в використанні сучасних інформаційних технологій, що не лише реалізує простоту використання

експертних систем кінцевими користувачами, а й допоможе більш ефективно прийняти рішення щодо вибору фільмів, тому дана тема є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка спеціалізованих засобів штучного інтелекту на основі інтелектуального аналізу даних та машинного навчання» і плану наукової та навчально-методичної роботи кафедри.

Мета та завдання роботи. Метою роботи є підвищення точності надання рекомендацій підбору фільмів.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі**:

- провести аналіз програм аналогів, систем-аналогів та обґрунтування доцільності розробки інформаційної технології підбору фільмів;
- розробити математичну модель інформаційної технології підбору фільмів;
- розробити структуру інформаційної технології підбору фільмів;
- здійснити програмну реалізацію інформаційної технології підбору фільмів;
- провести тестування програми для підбору фільмів та проаналізувати отримані результати;
- економічно обґрунтувати доцільність розробки інформаційної технології надання рекомендацій щодо підбору фільмів.

Об'єкт дослідження – це процес надання рекомендацій щодо підбору фільмів.

Предмет дослідження – інформаційні технології надання рекомендацій щодо підбору фільмів.

Методи дослідження. У роботі використано такі методи наукових досліджень: метод системного аналізу для аналізу структури інформаційної системи; методи підбору рекомендацій; метод колаборативної фільтрації, методи

експертних оцінок, метод аналізу ієрархій; методи математичної статистики; методи об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів полягає в наступному:

- удосконалено інформаційну технологію надання рекомендації щодо підбору фільмів, яка відрізняється від існуючих удосконаленим методом аналізу ієрархій для надання рекомендацій та методу дерева рішень, що дозволяє максимально задовольнити потреби користувача та забезпечує підвищення точності надання рекомендацій щодо вибору фільму.

Практичне значення одержаних результатів полягає у такому:

1. Удосконалено алгоритм підбору фільмів, що підвищило точність надання рекомендацій щодо підбору фільмів.
2. Здійснено програмну реалізацію інформаційної технології надання рекомендацій щодо підбору фільмів.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Результати магістерської кваліфікаційної роботи отримані самостійно. У публікації у співавторстві здобувачу належить дослідження перспектив інформаційної технології підбору фільмів [1, 2].

Апробація результатів магістерської кваліфікаційної роботи. Результати досліджень було апробовано на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці : дослідження, проблеми, перспективи» (НТКП ВНТУ–2022) м. Вінниці у 2022 р. [1].

Публікації магістерської кваліфікаційної роботи. За основними результатами досліджень опубліковано тези доповіді на конференціях [1].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ ФІЛЬМІВ

1.1 Обґрунтування використання методів штучного інтелекту для підбору фільмів

Штучний інтелект (ШІ) став дуже популярним у сучасному світі. Це моделювання природного інтелекту в машинах, які запрограмовані на навчання та імітацію дій людей. Ці машини здатні вчитися з досвідом та виконувати людські завдання [2].

ШІ дозволяє автоматизувати повторювані процеси навчання і пошуку за рахунок використання даних. Однак ШІ відрізняється від роботизації, в основі якої лежить застосування апаратних засобів. Мета ШІ – не автоматизація ручної праці, а надійне і безперервне виконання численних великомасштабних комп'ютеризованих задач. Така автоматизація вимагає участі людини для початкового налаштування системи та правильної постановки питань.

Також ШІ здійснює глибший аналіз великих обсягів даних за допомогою нейромереж з безліччю прихованих рівнів. Для моделей глибокого навчання необхідна величезна кількість даних, тому що саме на їх основі вони і навчаються [3].

ШІ найчастіше класифікується на такі поняття:

- ШІ вузького спектру (Artificial Narrow Intelligence) – галузь ШІ, яка перевершує виконання окремих завдань, повторюючи людський інтелект, та основну концепцію ШІ. Цей тип знань є в системах розпізнавання мови та голосових помічниках;

- сильний ШІ (Artificial General Intelligence) – ШІ, мета якого загальна і ефективність якого можна застосовувати до різноманітних завдань. Цей тип ШІ може вдосконалюватись завдяки навчанню і за можливостями є найближчим до людського мозку;

- штучний супер інтелект (Artificial Super Intelligence) – перевищуючи людський інтелект, ця концепція ШІ є набагато складнішою, ніж будь-яка інша система штучного інтелекту або навіть людський мозок. Головною рисою ASI є те, що він може міркувати про абстракції, про які люди не здатні думати. Його нейронна мережа перевищує мільярди нейронів людини.

Деякими типами технологіями ШІ є [4]:

- машинне навчання (Machine Learning) – це програма ШІ, яка надає комп'ютерним системам можливість автоматичного навчання та вдосконалення на основі досвіду без явного програмування. ML зосереджується на розробці алгоритмів, які можуть аналізувати дані та робити прогнози. Окрім використання для прогнозування музики, які можуть сподобатися користувачам, машинне навчання застосовується у сферах охорони здоров'я, фармації та наук про життя, щоб допомогти діагностувати захворювання, інтерпретувати медичні зображення та пришвидшити розробку ліків;

- глибоке навчання (Deep Learning) – це підмножина машинного навчання, що використовує штучні нейронні мережі, які навчаються, обробляючи дані. Штучні нейронні мережі імітують біологічні нейронні мережі в мозку людини. Кілька шарів штучних нейронних мереж працюють разом, щоб визначити єдиний вихід з багатьох входів, наприклад, ідентифікуючи зображення обличчя з мозаїки плитки. Машини навчаються шляхом позитивного та негативного підкріплення завдань, які вони виконують, що вимагає постійної обробки та підсилення для прогресу.

- нейронні мережі (Neural Network) включають в себе глибоке навчання. Як уже згадувалося, нейронні мережі – це комп'ютерні системи, створені за зразком нейронних зв'язків у мозку людини. Штучним еквівалентом нейрона людини є персептрон. Подібно до того, як пучки нейронів створюють нейронні мережі в мозку, стеки персептронів створюють штучні нейронні мережі в комп'ютерних системах. Нейронні мережі навчаються, обробляючи навчальні приклади. Цей процес аналізує дані багато разів, щоб знайти асоціації та надати значення раніше

невизначеним даним. За допомогою різних моделей навчання, як позитивне підкріплення, машину навчають, що вона успішно ідентифікувала об'єкт;

- когнітивне обчислення (Cognitive Computing) – ще один важливий компонент ШІ. Його мета - імітувати та покращити взаємодію між людьми та машинами. Когнітивні обчислення прагнуть відтворити процес мислення людини за допомогою комп'ютерної моделі, в даному випадку, розуміючи людську мову та значення зображень. Разом когнітивні обчислення та штучний інтелект прагнуть наділити машини людською поведінкою та здібностями до обробки інформації;

- обробка природної мови (Natural Language Processing), дозволяє комп'ютерам інтерпретувати, розпізнавати та виробляти людську мову та мовлення. Кінцева мета NLP – забезпечити безперебійну взаємодію з машинами, якими ми користуємося щодня, навчаючи системи розуміти людську мову в контексті та виробляти логічні відповіді;

- комп'ютерний зір (Computer Vision) – це техніка, що реалізує глибоке навчання та ідентифікацію шаблону для інтерпретації змісту зображення; включаючи графіки, таблиці та зображення у документах PDF, а також інший текст та відео.

З розвитком технологій з'являється все більше можливостей цікаво провести час не виходячи з дому, для цього достатньо мати тільки комп'ютер чи телефон і доступ до швидкісного інтернету. Поступово навіть кінотеатри переміщуються прямо до наших домів – їх успішно замінюють потокові сервіси, як-от Netflix, Amazon, YouTube та інші, онлайн кінотеатри Megogo, Google Movies. Багато сервісів активно знімають фільми та серіали власного виробництва – щоб привабити нових користувачів та вигідно відрізнятись від конкурентів.

Таким чином, щороку кількість доступних до перегляду фільмів та серіалів на будь-яку тематику невпинно зростає. Але водночас все складніше і складніше підібрати фільм чи серіал, який би неодмінно сподобався: по-перше, чим більше фільмів знімають, тим важче знайти серед них якісний продукт, а також обрати

поміж кількох фільмів із цікавим описом і гарними оцінками. По-друге, всі фільми які входять до топ 100 найкращих фільмів за версією глядачів/критиків якогось сайту, це здебільшого одні й ті самі фільми, які всі разом складають невелику частину усіх доступних до перегляду, і які скоріше за все передивилися вже всі друзі і знайомі. А фільмів і серіалів настільки багато, що передивитися всі не вийде ніколи, можна тільки здогадуватись чи сподобається певний фільм конкретному користувачу, судячи з опису, оцінок та коментарів інших користувачів.

Отже, обрати щось швидко з такого різноманіття просто неможливо, вибір може затягнутися на години, аж доки вже й не буде часу дивитися обраний фільм.

Системи рекомендації фільмів виявилися потужним інструментом для надання корисних пропозицій фільмів для користувачів. За допомогою таких систем надаються рекомендації, які допомагають користувачам справитись із надмірною кількістю інформації, а також допомагають швидко та зручно підібрати відповідні фільми. На відміну від персональних комп'ютерів, мобільні послуги приділяють більше уваги питанню швидкості надання послуг, що, у свою чергу, вимагає швидкої обробки та обчислень з боку провайдерів послуг. Таким чином, рекомендації фільмів у мобільних сервісах повинні виконувати якісну роботу як з точки зору точності наданих рекомендацій, так і з точки зору швидкості обробки інформації.

Система рекомендацій є підкласом системи фільтрації інформації для передбачення «рейтингу» або «уподобань» користувача. Такі системи використовують у різних областях і найчастіше визнають як генератори списків відтворення для відео та музичних послуг, таких як Netflix, YouTube і Spotify, рекомендації щодо продуктів для таких послуг, як Amazon, або рекомендації щодо вмісту для соціальних медіа-платформ, таких як Facebook і Twitter.

Формування системи рекомендації фільмів – це всеохоплююче і складне завдання, яке передбачає аналіз різних вподобань користувачів, різноманітності фільмів та інше. Тому для вирішення проблеми надання якісних рекомендацій у сучасному світі уже запропоновано багато методів.

У загальному, користувачі більше схильні вибирати фільми, яким надала перевагу більшість людей, і відмовляються від фільмів, які більшості людей не сподобались. Рішення на основі досвіду інших людей приймається користувачами з метою спрощення і покращення власного досвіду користування системою. В цьому випадку доцільно використовувати рейтинг IMDb [5].

База даних фільмів в Інтернеті (англ. Internet Movie Database, IMDb) – найбільша база даних та вебсайт про кінематограф. База даних значною мірою заповнюється добровольцями, це чимось нагадує концепцію Вікі. У базі зараз зібрана інформація про 10.1 млн фільмів і телесеріалів, є інформація майже про 11,5 млн акторів, режисерів та інших професіоналів кіно зі всього світу.

За даними рейтингового агентства «Alexa», сайт imdb.com постійно входить до сотні найпопулярніших сайтів Всесвітньої павутини. Основна мова інтерфейсу сайту – англійська, українського інтерфейсу сайт не має; проте назви фільмів поступово перекладаються [6].

Зареєстровані користувачі на сайті сайт imdb.com отримують персональні рекомендації фільмів на основі оцінених ними або доданих до списку перегляду фільмів. Рекомендуються ті фільми, які подобаються користувачам із схожими смаками. Для кожної рекомендації вказуються інші фільми чи серіали, на основі яких вона була зроблена.

1.2 Існуючі моделі та методи, що застосовуються для реалізації поставленої задачі

Існує багато методів підтримки прийняття рішень, класифікація яких відбувається за різними критеріями, поділяючи на методи багатокритеріальної оптимізації та багатокритеріального аналізу.

Кожен метод має свої переваги та недоліки, проте кожен може бути корисним залежно від конкретної ситуації.

Метод аналізу ієрархій визначає коефіцієнти важливості критеріїв, цілей, оцінки альтернатив. Внаслідок підсумовування множення коефіцієнтів впливу

на оцінки альтернатив. Даний метод є популярним, завдяки гнучкості, відсутності жорстких вимог і обмежень.

Метод аналізу ієрархії, розроблений Томасом Сааті, – один з найпопулярніших підходів у вирішенні багатокритеріальних задач [7].

Структура методу показана на рисунку 1.1.

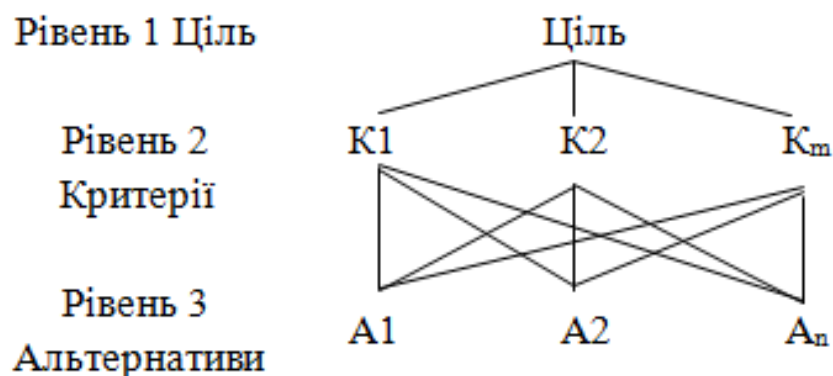


Рисунок 1.1 – Структурна схема методу аналізу ієрархій

Алгоритм складається з наступних кроків:

1. Декомпозиція проблеми: демонстрація у вигляді ієрархії;
2. Заповнення матриць попарних порівнянь на всіх рівнях ієрархії;
3. Перевірка узгодження у матриці попарних порівнянь і обчислення локальних пріоритетів об'єктів на кожному рівні ієрархії;
4. Обчислення глобальних пріоритетів і визначення кращих варіантів вирішення проблеми.

Кожну матрицю необхідно перевірити на узгодженість за формулою 1.1:

$$CI = \frac{\lambda_{max} - m}{m - 1}, \quad (1.1)$$

де m – порядок матриці попарних порівнянь; λ_{max} – найбільше власне число матриці попарних порівнянь.

Множина альтернатив визначається за допомогою одного з семи алгоритмів:

- коефіцієнт кореляції Пірсона;
- метод SVD;
- метод найближчого сусіда;
- K-means метод;
- TF-IDF;
- метод гібридної фільтрації;
- Apriori.

Дерево прийняття рішень використовується в статистиці та аналізі даних для моделей прогнозування. Структура дерева містить такі елементи: «листя» та «гілки». На «гілках» дерева рішень записуються атрибути, від яких залежить цільова функція, у «листі» записуються значення цільової функції, а в інших вузлах – атрибути, за якими випадки різні. Щоб класифікувати новий випадок, потрібно спуститися по дереву до листка і дати відповідне значення. Такі дерева рішень широко використовуються в аналізі даних. Мета полягає в тому, щоб створити модель, яка передбачає значення цільової змінної з урахуванням кількох вхідних змінних [8].

Алгоритм для побудови нечіткого дерева рішень для прийняття рішень щодо відповідності підбору фільму до вподобань користувача примає на вхід множину навчальних даних D . Алгоритм складається з наступних кроків:

Крок 1. Генерування кореня дерева як набору всіх даних D , задання ступеня впевненості μ для нечіткої множини, яка відповідає оцінці фільму користувачем.

Крок 2. При поточному вузлі t з набором нечіткої множини даних D задовольняє одну з наступних умов:

- частка набору даних класу C_k більша або рівна θ_r :

$$\frac{|D^{C_k}|}{|D^C|} \geq \theta_r; \quad (1.2)$$

- кількість наборів даних менше θ_n :

$$|D| < \theta_n; \quad (1.3)$$

- немає більше атрибутів для класифікації, вузол вважається листом.

Крок 3.1. Розрахунок приросту інформації та вибір атрибуту, що максимізує розглянуті атрибути.

Крок 3.2. Розбиття D на нечіткі множини D_1, D_2, \dots, D_m згідно з обраним максимізованим атрибутом, де значення ступеню впевненості буде розраховуватись добутком ступенем приналежності множині D на значення $F_{max,j}$ для A_{max} в D . При чому A_{max} видаляється з подальшого розгляду. Для предметної області підбору фільму слід зазначити, що деякі з атрибутів можуть мати необхідність використання двовимірного масиву в якості елементів нечітких множин, що може спричинити деякі складнощі. Наприклад, якщо матимемо атрибут «доріжка», то необхідно перевести матрицю приналежності деякого атрибуту доріжка до окремого числа – значення впевненості μ_a . Для цього використаємо зведення матриці:

$$\mu_{a_1} = \frac{1}{q} \sum_i^n \sum_j^n h_{ij} f_{ij}^{(a_1)} \in [0; 1], \quad (1.4)$$

де $H = ||h_{ij}||$ - матриця хобі, $F_{a_1} = |f_{ij}^{(a_1)}|$ – матриця типізації доріжки. m в даному випадку буде дорівнювати кількості комірок матриці доріжки.

Крок 3.3. Генерація нових вузлів для нечітких множин.

Крок 3.4. Заміна D на D_j ($j = 1, 2, \dots, q$) та перехід до пункту 2.

Приклад дерева рішень для такої ситуації наведено на рисунку 1.2. При чому перша точка прийняття рішення – це визначення доріжки, а друга – визначення рейтингу IMDb.

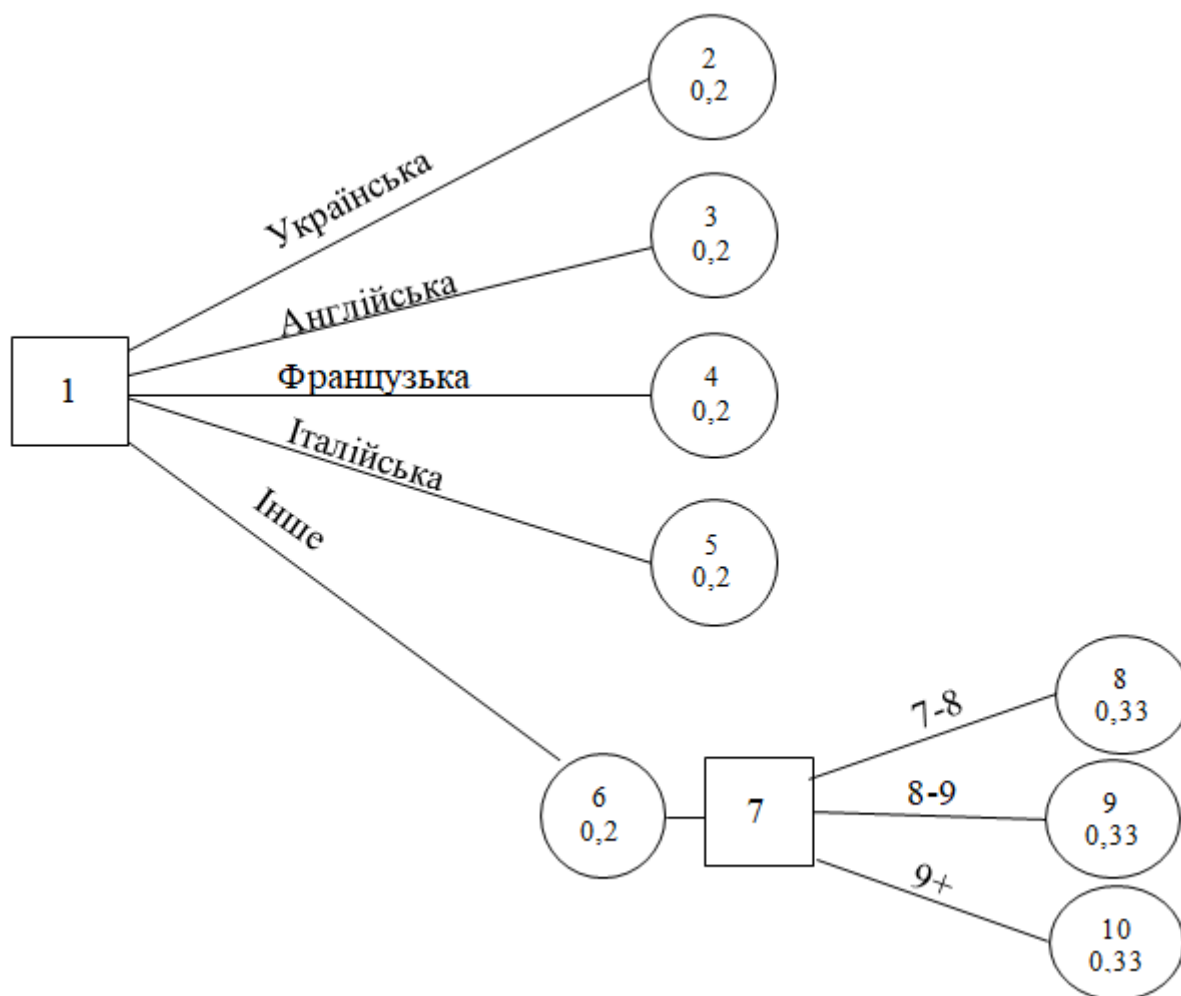


Рисунок 1.2 – Структурна схема моделі дерева рішень для ситуації, коли не важлива звукова доріжка.

Нечітка логіка – математичний розділ, що являє собою поєднання класичної логіки та теорії множин. Вперше представлений Лотфі Заде в 1965 році як розділ, який вивчає об'єкти з функцією належності елемента до набору, який приймає значення в інтервалі $[0, 1]$, а не просто 0 або 1. На основі цієї концепції, вводяться логічні операції над нечіткими множинами та формулюється поняття лінгвістичної змінної, якими є нечіткі множини.

Предмет нечіткої логіки – дослідження суджень з точки зору невизначеності, які подібні до суджень у звичайному розумінні, та їх застосування в комп'ютерних системах.

Теорія невизначеного прийняття рішень у розмитих обставинах відрізняється від проблеми прийняття рішень загалом тим, що один або кілька елементів моделі рішення даються нечіткими множинами.

Традиційний розмитий логічний підхід до прийняття рішень базується на принципі Беллмана-Заде, який розглядає нечіткі рішення як поєднання розмитих цілей та обмежень. У 1970 році Беллман і Заде опублікували статтю «Прийняття рішень у нечіткому середовищі», в якій досліджувався процес прийняття рішень за невизначених обставин, коли цілі та обмеження ставляться в розмитих умовах. Прийняття рішень – це вибір альтернативи, яка відповідає як розмитим цілям, так і межах. З цієї точки зору цілі та обмеження, пов'язані з рішенням, симетричні. Це розмиває різницю між ними та рішенням, яке можна представити як поєднання невизначених цілей та меж [9].

Коротко наведемо приклад відмінних переваг систем із нечіткою логікою порівняно з іншими:

- можливість оперативно працювати з набором вхідних даних: наприклад, постійно змінюються значення з часом (динамічні завдання), значення, що не можна встановити однозначно (результати статистичних опитувань, рекламних кампаній тощо);
- можливість нечіткої формалізації критеріїв оцінки та порівняння: оперування критеріями "більшість", "можливо", переважно "тощо";
- можливість проводити якісну оцінку як попередніх даних, так і отриманих результатів: ви керуєте не лише фактичними значеннями даних, але й ступенем їх вірогідності та їх розподілом;
- можливість швидко моделювати складні динамічні системи та їх порівняльний аналіз з певним ступенем точності: працюючи відповідно до принципів поведінки системи, описаних нечіткими методами, ви не витрачаєте багато часу на з'ясування точних значень змінних та рівнянь, як вони описані, по-друге, ви можете описати різні варіанти початкових умов.

Баєсові методи були розроблені в результаті численних спроб вчених визначити проблеми статистичного аналізу поведінки різних процесів та знайти

їх рішення, застосовуючи основу байєсівської методології – теорему Байєса. Використання цієї теореми має ряд передумов, основна з яких – наявність певних зв'язків між ймовірністю явищ різної природи та специфікаціями будь-якого явища на бажаному рівні [10].

Байєсівська методологія відрізняється від інших підходів тим, що ще до отримання даних дослідник визначає рівень своєї впевненості у можливих моделях і згодом подає їх у вигляді певних ймовірностей [11].

Детальніше розгляньте переваги та недоліки класифікатора Байєса.

Переваги класифікатора Байєса включають в себе наступні пункти:

- Класифікація, включаючи багатокласну, проста та швидка. Це дозволяє передбачити ймовірність для кількох значень цільової змінної.

- Коли припущення щодо незалежності виконуються, НБК перевершує інші алгоритми, такі як логістична регресія, що вимагає менших даних про навчання.

- НБК працює краще за категоричними ознаками, ніж тверді. Для безперервних ознак передбачається нормальний розподіл, що є досить сильним припущенням.

- Класифікація в режимі реального часу. НБК вчиться дуже швидко, саме тому його можна використовувати для обробки даних у режимі реального часу.

Як недоліки класифікатора Байєса зазначають:

Якщо тестовий набір даних має певне значення атрибута, він не відображається в навчальному наборі даних, тоді встановіть модель на нуль, і це значення навряд чи зможе зробити передбачення. Це явище відоме як "нульова частка". Цю проблему можна вирішити згладжуванням. Ще одне обмеження НБК – прийняття функції незалежності. Насправді набори абсолютно незалежних функцій надзвичайно рідкісні.

Байєсівські довірчі мережі застосовуються у сферах, яким характерна спадкова невизначеність [12]. Вона може бути обумовлена:

- неповним розумінням предметної області;

- неповні знання;
- характеристика проблеми випадковістю.

Отже, байєсівські мережі довіри (БМД) застосовують для моделювання ситуацій, які в певному сенсі містять невизначеність. Для байєсівських довірчих мереж інколи користується популярністю інша назва причинно-наслідкової мережі, в якій випадкові події пов'язані між собою причинно-наслідковими зв'язками.

Причинно-наслідкові зв'язки спрощують оцінку ймовірностей подій. У звичайному світі оцінка найчастіше проводиться в напрямленні від «спостерігача» до «спостереження», або від «ефекту» до «наслідку», що, як правило, важче оцінити, ніж напряму «наслідок \rightarrow ефект», тобто у напрямку від наслідку.

Проведемо оглядовий приклад мережі, в якій ймовірність вершини e в різних станах (e_k) залежить від станів (c_i, d_j) вершин d і c і визначається виразом

$$p(e_k) = \sum_i \sum_j p(e_k | c_i, d_j) \times p(c_i, d_j), \quad (1.5)$$

де $p(e_k | c_i, d_j)$ – ймовірність перебування в стані e_k залежно від станів c_i, d_j . Оскільки події, зображені вершинами d і c незалежні, то

$$p(e_k | c_i, d_j) = p(c_i) \cdot p(d_j). \quad (1.6)$$

Проаналізуємо інший приклад, що застосовує більш складну мережу що представляє умовну незалежність подій. Для оцінки вершин c і d застосовуються ті самі взаємозв'язки, що й для обчислення $p(e_k)$, тоді:

$$p(c_i) = \sum_m \sum_n p(c_i | A_{1m}, B_{1n}) \times p(A_{1m}) \times p(B_{1n}), \quad (1.7)$$

$$p(d_j) = \sum_m \sum_n p(d_j | A_{2m}, B_{2n}) \times p(A_{2m}) \times p(B_{2n}). \quad (1.8)$$

З розглянутих виразів випливає, що вершина «e» умовно не залежить від вершин A_1 , A_2 , B_1 , B_2 , через відсутність стрілок, що практично з'єднують ці вершини.

Переглянувши ці приклади, можна спробувати точніше охарактеризувати основні поняття, що застосовуються для БМД. БМД - це спрямований ациклічний графік із наступними особливостями:

- подіями виступає кожна вершина, а також кожна подія описана випадковою величиною, що здатна перебувати у декількох станах;
- усі вершини, що мають зв'язок з «батьківським», описуються таблицею умовних ймовірностей чи функціями умовних ймовірностей;
- інші вершини, що не мають зв'язків з «батьківськими вершинами», ймовірності її станів безумовні (граничні).

Інакше кажучи, в байєсівських довірчих мережах вершини є випадковими величинами, а дуги – це ймовірнісні залежності, що визначаються за допомогою умовних таблиць ймовірностей. Таблиця умовних ймовірностей кожної вершини містить ймовірності станів цієї вершини за умови станів її «батьків».

Враховуючи перераховані переваги та недоліки розглянутих методів можна зробити висновок про доцільність використання поєднання методу аналізу ієрархії та алгоритму на основі дерева рішень у майбутній математичній моделі підбору фільмів.

1.3 Аналіз відомих програмних рішень надання рекомендацій щодо підбору фільмів

Існує багато онлайн сервісів, що надають рекомендації стосовно вибору фільму, багато з яких також є платформами для їх перегляду. Необхідно створити інформаційну технологію, яка буде задовольняти користувача високою точністю та усуне недоліки розглянутих аналогів. Для порівняльного аналізу

характеристик програм-аналогів розглянемо наступні програми, які представлені у таблиці 1.1.

Таблиця 1.1 – Представлення порівняльних характеристик існуючих програм-аналогів

Програмна система	Переваги	Недоліки
Джарвіс	Система оцінювання фільмів від 1 до 5 при використанні 11 параметрів; Інтеграція з сервісами Megogo, Youtube, Vimeo. Оцінка настрою користувача	Лише ті фільми, у яких присутня українська доріжка
Netflix	Пропонує персональні рекомендації на основі раніше переглянутих фільмів та серіалів; Є списки найкращих фільмів відповідно певної характеристики, причому для різних користувачів ці списки відрізняються	Кількість фільмів та серіалів обмежена; Не можна оцінити чи прокоментувати фільм
IMDB	Відображення рейтингу критиків, користувачів та загального біля кожного фільму; Багато списків найкращих фільмів за різними характеристиками.	Англомовний ресурс.
Movie Expert	Щоденні підбірки фільмів; Розширений профіль користувача; Функція додавання в список.	Відсутність фільтрації по акторах; Не можна оцінити або коментувати фільм.

Джарвіс – сервіс, що допомагає уникнути багатогодинних пошуків фільмів та забезпечує підбір фільму, що який має сподобатись користувачеві.

Вибір фільмів здійснюється за декількома параметрами, деякі з унікальних властивостей сервісу [13]:

- інформація тільки про ті фільми, у яких присутня українська доріжка;
- підбір базується на великій кількості характеристик;
- розширена система оцінювання фільмів: одинадцять параметрів оцінюються від 1 до 5. Наприклад, часто фільми мають багато жанрів, один з

яких комедія. Коли користувач починає перегляд, то фільм виявляється абсолютно не смішним. Тут фільм оцінюється глядачами саме як смішний від 1 до 5, тому можна порекомендувати перелік смішних фільмів, а не тих, в яких просто вказано жанр комедія;

- для зареєстрованих користувачів є функції “відкласти на потім” та «прибрати з видачі»;

- спеціальні фільтри для сімей з дітьми «КДП» і «КДП>13» – Коли Діти Поруч, фільми, які комфортно дивитися батькам у присутності дітей.

На сайті «Джарвіс» розробники зібрали стрічки, у яких є українська озвучка чи дубляж українською мовою. Обрати стрічку можна не лише по жанрах. Система пропонує обрати настрій, із яким повинен бути фільм.

Сервіс не пропонуватиме відповідні до настрою жанри. «Джарвіс» аналізуватиме, як оцінили фільм глядачі і за настроєм публіки буде визначати у яку категорію його вносити. Часто фільми мають багато жанрів, один з яких – комедія.

Також у цьому сервісі можна обрати часовий період та країну, у яких сайт шукатиме фільм.

На сайті можна реєструватися через електронну пошту або Facebook. Приклад роботи сервісу зображено на рисунку 1.3.

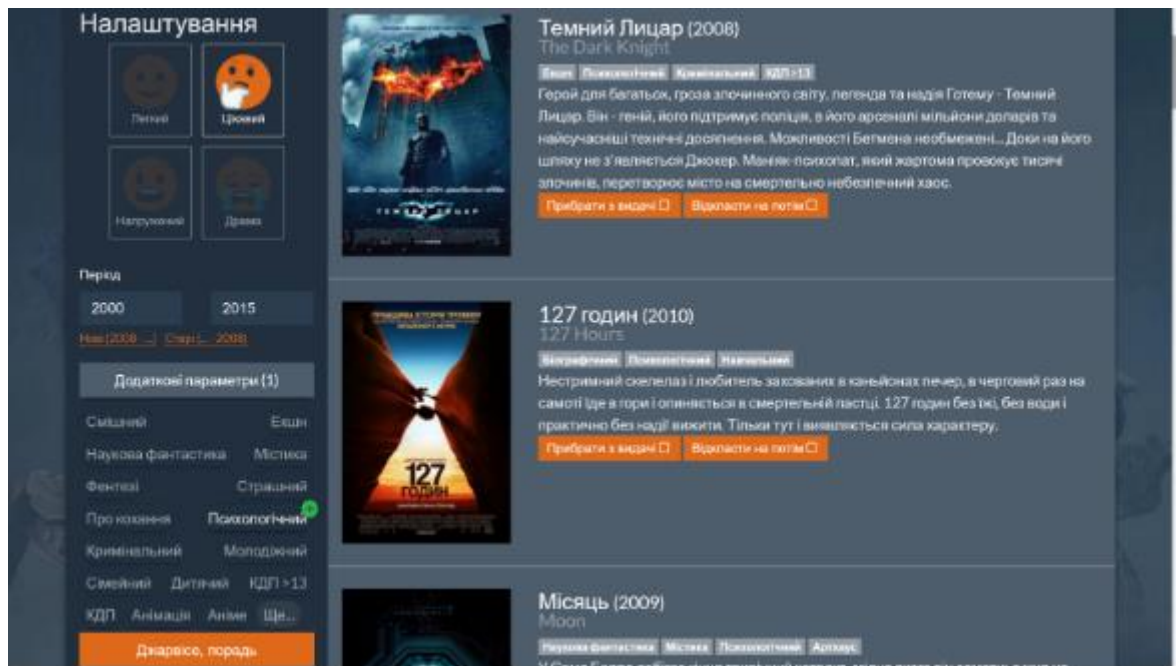


Рисунок 1.3 – Загальний вигляд інтерфейсного вікна системи «Джарвіс»

Netflix – потоковий сервіс, що дозволяє переглядати велику кількість фільмів та серіалів. Для кожного фільму чи серіалу можна переглянути список подібних, а також відсоток подібності із цим фільмом чи серіалом. Є списки найпопулярніших фільмів, фільмів із найбільшою кількістю переглядів, найкращих фільмів для кожного жанру, причому для різних користувачів ці списки відрізняються [14].

До переваг сервісу можна віднести, що фільми та серіали можна додати до власного списку, поставити їм негативну чи позитивну оцінку. Також сервіс пропонує персональні рекомендації на основі раніше переглянутих фільмів та серіалів.

До недоліків сервісу Netflix слід віднести:

- кількість фільмів та серіалів досить обмежена;
- не можна оцінити чи прокоментувати фільм.

IMDB – онлайн база даних інформації про фільми та серіалами, включаючи акторів, людей пов’язаних із створенням фільмів, короткий опис сюжету, рецензії критиків та глядачів [15]. Приклад надання рекомендацій щодо вибору фільму за допомогою IMDB зображено на рисунку 1.4.

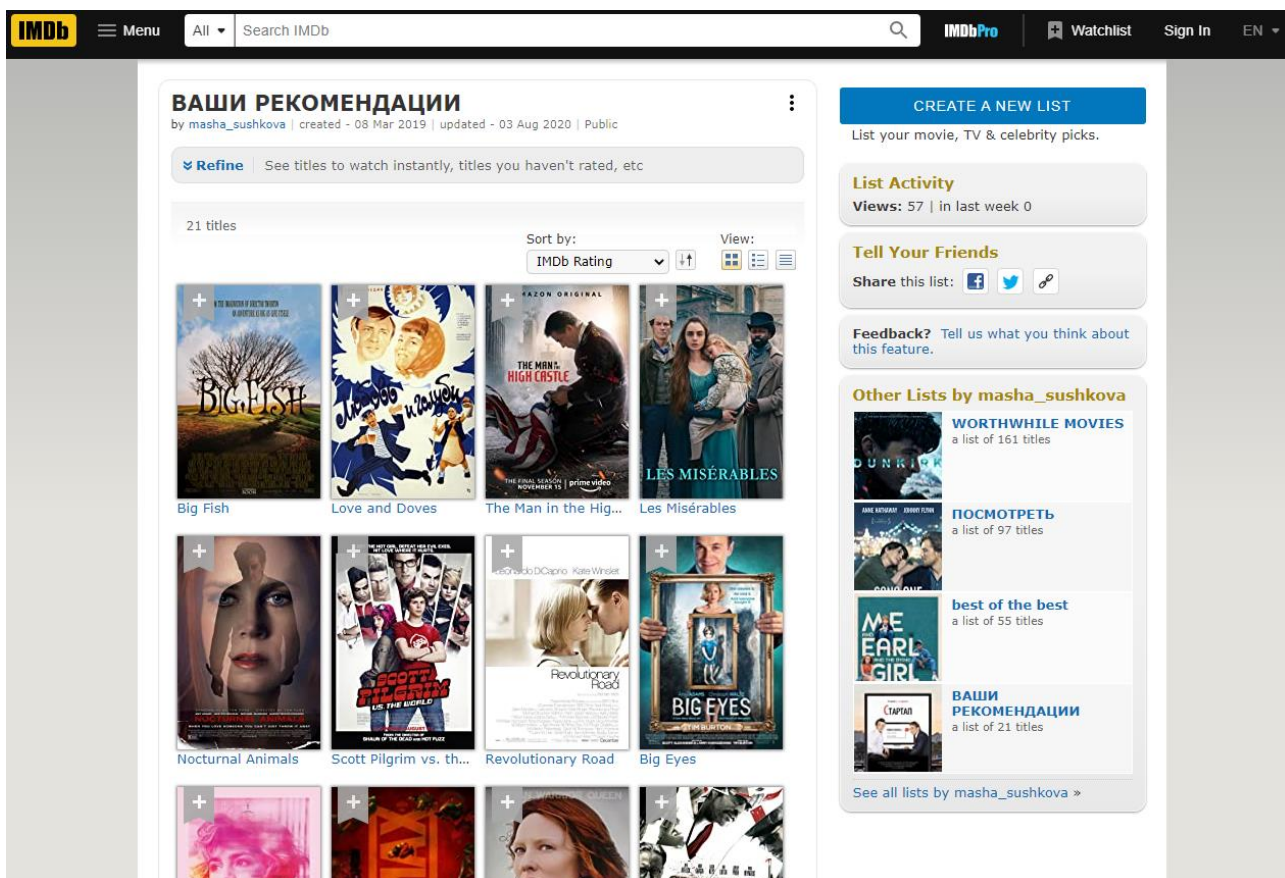


Рисунок 1.4 – Загальний вигляд інтерфейсного вікна системи IMDb

До переваг сервісу IMDb можна віднести:

- Всі користувачі можуть переглядати сторінки фільмів, людей пов'язаних із створенням фільмів та акторів;
- Зареєстровані користувачі можуть створювати нові сторінки, а також редагувати вже існуючі;
- Кожний зареєстрований користувач має вказати ім'я, але може заборонити іншим користувачам його переглядати. У профілі користувача вказується як довго він зареєстрований на сайті, його власні оцінки фільмам (якщо користувач захоче їх показати), значки за доповнення інформації про фільми;
- Користувачі можуть оцінювати фільми за шкалою від 1 до 10;
- Оцінки користувачів враховуються у середньому зваженому рейтингу, який відображається біля назви фільму, використовуються спеціальні фільтри для уникнення фальсифікацій;

- Для кожного фільму відображається рейтинг критиків, користувачів, та загальний;

- Є окремі списки фільмів у прокаті, а також тих, що скоро виходять;

- Фільми можна додати до списку, який можна переглянути;

- Є багато списків найкращих фільмів за різними характеристиками;

- Доступна цікава інформація про фільми, а також новини;

- Зареєстровані користувачі отримують персональні рекомендації фільмів на основі оцінених ними або доданих до списку перегляду фільмів. Рекомендуються ті фільми, які подобаються користувачам із схожими смаками. Для кожної рекомендації вказуються інші фільми чи серіали, на основі яких вона була зроблена.

Недолік сервісу:

- Застосунок доступний лише англійською мовою.

Для тих, хто використовував сервіси з рейтинговою оцінкою, які, однак, не змогли допомогти в пошуку відповідних фільмів, було створено стартап Movie Expert. Кінострічки з високим рейтингом були або переглянуті раніше, або не цікаві. А фільми із середнім рейтингом було важко оцінити і відібрати. По-перше, коментарі і рецензії ділилися на два повноцінних табори: за і проти. По-друге, сам процес підбору фільмів займав досить багато часу (адже щороку тільки в Голлівуді випускають більше 800 фільмів і серіалів), і в більшості випадків часу на перегляд кінострічки просто не залишалося [16].

Так і прийшла думка створити сайт зі спеціальним алгоритмом, метою якого є: відображення ряду найпопулярніших продуктів кінематографу згідно з інтересами конкретного користувача, базуючись на оцінках раніше вподобаних фільмів; надати можливість ранжування рекомендацій фільмів з огляду на додаткові параметри. Вигляд інтерфейсного вікна системи зображено на рисунку 1.5.

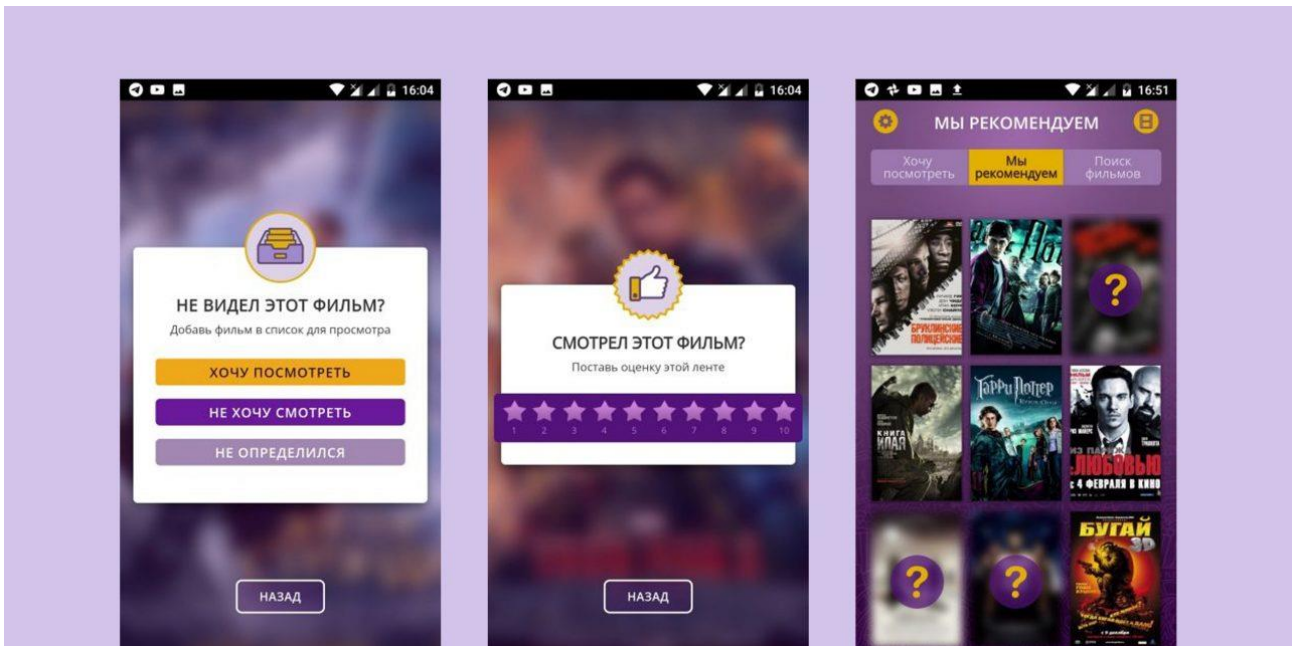


Рисунок 1.5 – Загальний вигляд інтерфейсного вікна системи Movie Expert

Переваги сервісу Movie Expert:

- Щоденні підбірки фільмів на різноманітні тематики — відмінна альтернатива свайпам і рекомендаціям.
- Розширений профіль зі статистикою за своїми оцінками і визначенням улюблених жанрів, а також система досягнень користувача з бонусами (критик, кіноман, сценарист, експерт).
 - Тепер у видачу можна додати серіали.
 - Пошук фільму: можна відразу знайти кіно і додати його в список «хочу подивитися» або оцінити переглянуті раніше фільми. Додана функція «поділитися фільмом з другом» (посилання, при переході на яке відкривається сторінка фільму).
 - PRO-режим – можливість отримати повний доступ до фільмів зі списку «Рекомендації», які обирає для користувача система на основі його оцінок. Деякі картки фільмів у цьому списку знаходяться під знаком питання. Якщо потрібно дізнатися, який це фільм, можна або подивитися рекламу, або купити один з PRO-режимів, і всі фільми будуть відкриті. Також можна побачити розширену статистику за своїми оцінками.

Отже, проаналізувавши системи надання рекомендацій щодо вибору фільму слід зазначити, що все ще існує потреба у створенні україномовного ресурсу для надання персоналізованих рекомендацій фільмів, що матиме високу ефективність надання рекомендацій стосовно вибору фільму. Серед проаналізованих систем найбільш близькою за характеристиками є програма «Джарвіс», тому саме її вибрано в якості прототипу.

1.4 Постановка задачі дослідження

Задача магістерської кваліфікаційно роботи (МКР) полягає у розробці інформаційної технології надання рекомендацій щодо підбору фільму.

Вимоги до клієнта:

- варіанти відповідей на питання;
- рекомендації фільмів;
- завершення сесії (вихід).

Вимоги до адміністратора:

- можливість редагування інформації;
- можливість додавання нової інформації (питань, фільмів);
- можливість видалення інформації.

Вимоги до серверу:

- зберігання даних користувача через зв'язок з базою даних;
- видача необхідних даних по запиті від клієнта.

Вхідними даними інформаційної технології надання рекомендацій щодо підбору фільму є наступні:

- 1) Бажаний жанр.
- 2) Рік виходу.
- 3) Країна створення.
- 4) Вікове обмеження.
- 5) Обмеження.
- 6) Режисер.

- 7) Звукова доріжка.
- 8) Рейтинг IMDb [16].

Вихідними даними є рекомендації для користувача щодо вибору фільмів, на основі аналізу його відповідей на поставлені питання.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм підбору фільмів.

1.5 Висновок до розділу 1

У даному розділі обґрунтовано використання ШІ для підбору фільмів. Досліджено різні методи та моделі вирішення поставленої задачі та вибрано алгоритм дерева прийняття рішень, у поєднанні з методом аналізу ієрархій. Проведено аналіз систем-аналогів, які створені для надання рекомендацій щодо вибору фільмів та формування рейтингової оцінки фільму та коротко охарактеризовано їх головні переваги та недоліки, а саме відсутність орієнтованості в багатьох аналогах на україномовного користувача, обмежена кількість фільмів та серіалів, можливість ефективного надання рекомендацій лише завдяки існуючому профілю та при реєстрації, що зумовлює проблему «холодного старту». Здійснено постановку задачі та сформовано вимоги до інформаційної технології підбору фільмів.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ ФІЛЬМІВ

2.1 Обґрунтування вибору базового методу надання рекомендацій щодо підбору фільмів

Традиційний метод ієрархічного аналізу запропонував Т. М. Сааті. Проблема структурована як ієрархія. На вершині ієрархії зазвичай знаходиться загальна мета, на наступному рівні – цілі, нижче – підцілі, потім критерії, підкритерії; на нижньому рівні – альтернативи. По-перше, існує покоління альтернатив, з яких ви повинні вибрати найкращу альтернативу або розмістити всі елементи. Цей рівень зменшує набір усіх можливих альтернативних угруповань залежно від рівня оцінки за рахунок обмеження кількості еквівалентних об'єктів [17].

Альтернативи також порівнюються за окремими критеріями. Попарне порівняння – це спосіб визначення коефіцієнтів значущості критеріїв та альтернатив. Результат порівняння оцінюється за шкалою. На основі цих порівнянь вони обчислюють коефіцієнти значущості критеріїв, рейтинг альтернатив і знають загальний рейтинг як велику суму оцінок критеріїв.

На якість порівняння може впливати кваліфікація осіб, які приймають рішення, і (або) властивості елементів ієрархії.

Основними перевагами методу парного порівняння є:

- відбувається порівняння кожної альтернативи за більшою кількістю пріоритетних факторів, що підвищує точність оцінки та відкриває можливість перевірки якості досліджуваного предмета, ніж при використанні інших методів;
- експерт не акцентує увагу на всіх альтернативах відразу, а лише на двох, які він щойно порівняв – це полегшує роботу;
- метод дає змогу отримати не тільки середню оцінку, надану кожним експертом, але й дисперсію цієї оцінки, що дозволяє провести більш глибокий економіко-математичний аналіз;

– метод відрізняється простотою та відповідністю інтуїтивним уявленням.

Основними недоліками цього методу є [18]:

- великий об'єм інформації;
- обмеження кількості одночасно порівнюваних альтернатив (бажано не більше 9 од.; це пов'язано з тим, що звичайній людині важко зробити раціональний вибір, якщо кількість об'єктів вибору перевищує 7 ± 2 од.);
- неможливість повного опису невизначеності системи переваг осіб, які приймають рішення;
- неможливість використання у випадку кількох осіб, які приймають рішення, та наявності систем суперечливих переваг.

Метод аналізу ієрархій може приймати до уваги матеріальні та нематеріальні чинники, обчислювати кількісні параметри та якісні характеристики, об'єктивні та суб'єктивні дані. Такий метод підходить для ранжування об'єктів, фактори яких представлені якісними характеристиками, однак для інформаційної технології надання рекомендацій щодо підбору фільмів можна використовувати даний метод для оцінки значущості критеріїв [19].

Алгоритм методу складається з наступних пунктів:

1. Побудова матриці попарних порівнянь.
2. Знаходження нормованого вектора матриці.
3. Знаходження індексів узгодженості.
4. Визначення найбільшого власного числа, індексу узгодженості та відношення узгодженості.
5. Визначення глобальних пріоритетів.
6. Визначення найкращої альтернативи за максимальним значенням глобального пріоритету.

Дерево прийняття рішень – графічний метод, що дозволяє пов'язати точки прийняття рішення, можливої стратегії A_i і її наслідків $E_{i,j}$ з можливими факторами, умовами зовнішнього середовища. Побудова дерева рішень починається з більш раннього рішення, потім зображуються можливі дії і наслідки кожної дії (подія), потім знову приймається рішення (вибір напрямку

дії) і далі до тих пір, поки всі логічні наслідки результатів не будуть вичерпані [20].

Дерево рішень будується за допомогою п'яти елементів [21]:

- момент прийняття рішення;
- точка виникнення події;
- зв'язок між рішеннями і подіями;
- імовірність настання події (сума ймовірностей в кожній точці повинна дорівнювати 1);
- очікуване значення (наслідки) – кількісне вираження кожної альтернативи, розташоване в кінці гілки.

Структура дерева рішень містить такі елементи: «листя» і «гілок». На «гілках» вказуються записані атрибути, від яких залежить цільова функція, в «листі» записані значення цільової функції, а в інших вузлах – атрибути, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листа і видати відповідне значення. Подібні дерева рішень широко використовуються в інтелектуальному аналізі даних. Мета полягає в тому, щоб створити модель, яка прогнозує значення цільової змінної на основі декількох змінних на вході [22].

Приклад переваг дерев рішень:

- Прості для розуміння та тлумачення. Можливість візуалізації дерев рішень.
- Потрібна підготовка вхідних даних. Інші методи часто вимагають нормалізації даних, потрібно створити фіктивні змінні та видалити порожні значення. Однак потрібно зауважити, що цей модуль не підтримує відсутні значення.
- Вартість використання дерева (тобто прогнозування даних) є логарифмічною за кількістю точок даних, що використовуються для навчання дерева.
- Здатність до обробки як числових, так і категоріальних даних. Однак реалізація scikit-learn поки що не підтримує категоріальні змінні. Інші методи, як

правило, спеціалізуються на аналізі наборів даних, які мають лише один тип змінних.

- Здатність вирішувати проблеми з кількома виходами.
- Використовує модель білої коробки. Якщо дана ситуація спостерігається в моделі, пояснення умови легко пояснюється булевою логікою. Навпаки, у моделі чорного ящика (наприклад, у штучній нейронній мережі) результати можуть бути складнішими для інтерпретації.
- Можливо перевірити модель за допомогою статистичних тестів. Це дає змогу враховувати надійність моделі.
- Добре працює, навіть якщо його припущення дещо порушені справжньою моделлю, з якої були створені дані.

Наприклад, для отримання рекомендації в ситуації, коли немає значення звукова доріжка для користувача, буде розглядатись така модель дерева (рис. 2.1), де номер у квадраті позначає номер точки прийняття рішення, а число в колі – точки можливостей з вказаною ймовірністю існування події. При чому перша точка прийняття рішення – це визначення доріжки, а друга – визначення рейтингу IMDb.

Регулювання глибини дерева – це техніка, яка дозволяє зменшувати розмір дерева рішень, видаляючи ділянки дерева, які мають маленьку вагу [22].

Найбільша проблема алгоритму – розмір. Невелике дерево може не охопити важливу інформацію. Але майже неможливо спрогнозувати, додавання якого вузла дозволить значно зменшити помилку. Ця проблема відома як «ефект горизонту». Проте, загальна стратегія обмеження дерева зберігається, тобто якщо вузол не несе додаткової інформації – він видаляється [8].

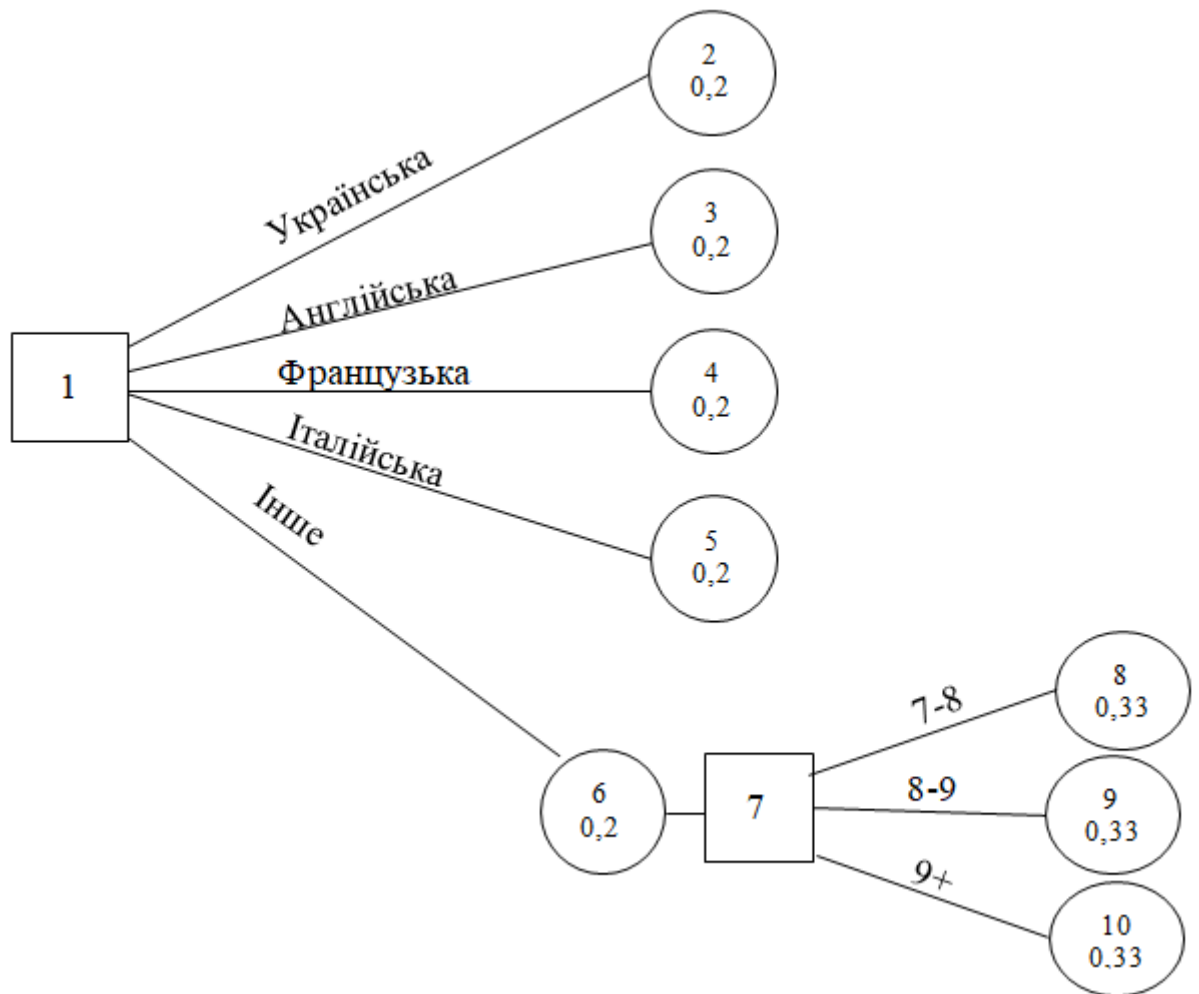


Рисунок 2.1 – Модель дерева рішень для ситуації, коли не важлива звукова доріжка

Регулювання глибини дерева необхідне для того, щоб зменшити розмір навчальної моделі дерева без зменшення точності її прогнозу або за допомогою перехресної перевірки. Є багато методів регулювання глибини дерева, які відрізняються вимірюванням оптимізації продуктивності.

Скорочення дерева може здійснюватися зверху вниз або знизу вгору. У першому випадку – обрізка починається з кореня, в іншому – скорочується число листя дерева. Один з найпростіших методів регулювання – зменшення помилки обмеження дерева. Зміни починаються з листя. Якщо це не впливає на точність передбачення, то зміна зберігається [18].

Таким чином, оптимізоване дерево прийняття рішень допоможе правильно сформулювати модель складних ситуацій для вибору фільму.

2.2 Розробка математичної моделі надання рекомендацій щодо підбору фільмів

Приведемо приклад абстрактного означення дерева рішень. Нехай B – деяка непорожня множина та F – деяка множина функцій, які визначені на B і такі, що набувають значення з множини $E_k = \{0, 1, \dots, k - 1\}$. Функції з F називатимемо перевітками, а пару $U = (B, F)$ – системою перевірок. Система U називається скінченною, якщо множина перевірок F скінченна, та нескінченною у протилежному випадку. Задача над U – це набір вигляду $z = (v, f_1, \dots, f_n)$, де $f_1, \dots, f_n \in F$, $E_k \rightarrow \omega$ та $\omega = \{0, 1, 2, \dots\}$. Задача z полягає у визначенні за довільним елементом $b \in B$ значення $z(b) = v(f_1(b), \dots, f_n(b))$. Дерево рішень над U – це кореневе дерево, для кожного листка якого є відповідне число з ω – результатом роботи дерева рішень, кожній внутрішній вершині відповідає перевірка з F , кожній дузі – число з E_k – значення перевірки, при якій здійснюється перехід по цій дузі. Дугам, що виходять з однієї вершини, відповідають попарно різні значення.

Дерево рішень інформаційної технології підбору фільмів наведено на рисунку 2.2.

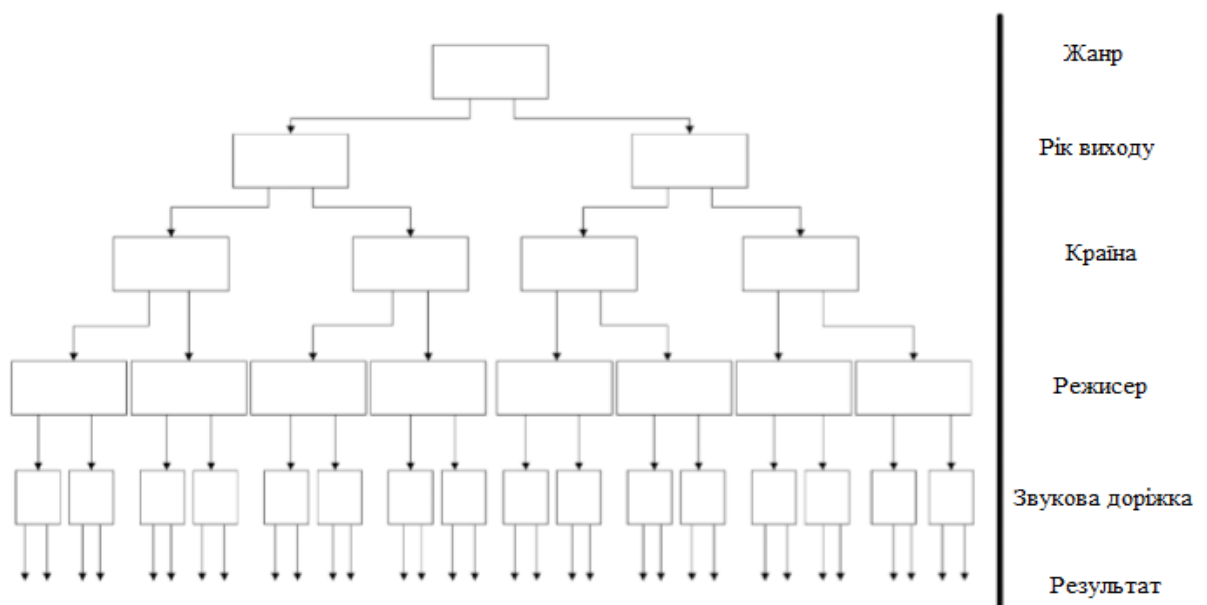


Рисунок 2.2 – Структурна схема дерева рішень інформаційної технології підбору фільмів

У процесі аналізу даних дерева рішень можуть бути використані для вирішення проблеми вибору пристрою як математичні та обчислювальні методи для опису, класифікації та узагальнення набору даних, які можна записати таким чином:

$$(x, Y) = (x_1, x_2, x_3 \dots x_k, Y) . \quad (2.1)$$

Залежна змінна Y – цільова змінна, яку потрібно проаналізувати, класифікувати та узагальнити. Вектор x складається з вхідних змінних x_1, x_2, x_3 тощо, які використовуються для виконання цього завдання [20].

Далі математична модель вводить поняття методу аналізу ієрархій для надання рекомендацій, яке можна використовувати для визначення критеріїв корисності, коли існує невизначеність після використання дерева рішень.

Нехай відомо $P = \{P_1, P_2, \dots, P_k\}$ – серія альтернатив, які піддають багатокритеріальному аналізу; $G = \{G_1, G_2, \dots, G_k\}$ – набір критеріїв, завдяки яким оцінюються варіанти. Завданням багатокритеріального аналізу є організація елементів множини P відповідно до критеріїв множини G .

Позначимо число, яке перебуває в діапазоні $[0,1]$ $MGi(P_j)$, таке число оцінює альтернативу $P_j \in P$ відповідно до критерію $G_i \in G$: для більшого числа $MGi(P_j)$ відповідатиме краща альтернатива P_j за критерієм G_i . Тому критерій G_i зазвичай визначають за використання нечіткої множини на універсальній множині варіантів P :

$$\tilde{G}_i = \left\{ \frac{\mu_{G_i}(P_1)}{P_1}, \frac{\mu_{G_i}(P_2)}{P_2}, \dots, \frac{\mu_{G_i}(P_k)}{P_k} \right\}, \quad (2.1)$$

де $MGi(P_j)$ – міра відповідності елемента P_j нечіткій множині G_i .

Для знаходження показників релевантності нечіткого набору (2.2) зручно застосувати метод розробки функції релевантності при порівнянні альтернатив. Використання такого методу для кожного критерію передбачає формування

матриці порівняння між парами альтернатив. Остаточна кількість відповідних матриць дорівнює кількості критеріїв.

Найкращою альтернативою вважають ту, яка найкраща за всіма критеріями. Нечітке рішення D обраховується перетином підкритеріїв:

$$\tilde{D} = \tilde{G}_1 \cap \tilde{G}_2 \cap \dots \cap \tilde{G}_n = \left\{ \frac{\min_{i=1,n} \mu_{G_i}(P_1)}{P_1}, \frac{\min_{i=1,n} \mu_{G_i}(P_2)}{P_2}, \dots, \frac{\min_{i=1,n} \mu_{G_i}(P_k)}{P_k} \right\}. \quad (2.2)$$

Зважаючи на обраховану нечітку множину D , найкращою альтернативою вважають ту, яка має найбільшу міру належності:

$$D = \arg \max(\mu_D(P_1), \mu_D(P_2), \dots, \mu_D(P_k)). \quad (2.3)$$

При невідповідності критеріїв міру належності нечіткої множини D використовують наступну формулу:

$$\mu_D(P_j) = \min_{i=1,n} (\mu_{G_i}(P_1))^{a_i}, \quad (j = \overline{1, k}), \quad (2.4)$$

де a_i – коефіцієнт відносної важливості критерія G_i , $a_1 + a_2 + \dots + a_n = 1$.

Коефіцієнт a_i в формулі (2.4) вказує на нечітку множину, що відповідає мірі важливості критерія G_i . Коефіцієнти відносної важливості критеріїв можуть визначатися різними способами, як приклад, це може відбуватися шляхом порівняння пар альтернатив за шкалою Сааті [17].

Реалізація функції приналежності, використовуючи метод попарного порівняння кожної пари альтернатив універсальної множини, відбувається за оцінюванням фахівцем переваги одного елемента над іншим. Такі попарні порівняння записують у вигляді матриці:

$$A = \begin{matrix} & \cdot & U_1 & U_2 & \dots & U_n \\ U_1 & \left[\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ U_n & a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right. & & & & \end{matrix}, \quad (2.5)$$

де a_{ij} – рівень переваги альтернативи u_i над альтернативою $u_j()$, що визначається за допомогою десятибальної шкали Сааті, де 1 ставлять тоді, коли u_i не переважає u_j ; 3 – коли u_i має невелику перевагу над u_j ; оцінку 5 ставлять при істотній перевазі u_i над u_j ; оцінку 7 – коли альтернатива u_i явно переважає альтернативу u_j ; 9 – коли u_i абсолютно переважає u_j ; в решті проміжними оцінками є бали 2, 4, 6, 8, де: 2 – трохи слабша перевага, 4 – коли u_i сильніше переважає u_j , 6 – істотніше переважає і 8 – значно переважає.

Матрицю попарних порівнянь називають діагональною ($a_{ij}=1$) та обернено симетичною ($a_{ij}=1/a_{ji}$).

Міра приналежності вважається рівною відповідно координатам власного вектора $W=(w_1, w_2, \dots, w_n)^T$ матриці парних порівнянь A :

$$m(u_j) = w_j. \quad (2.6)$$

Власний вектор знаходять з наступної системи рівнянь:

$$\begin{cases} AW = \lambda_{max}W, \\ w_1 + w_2 + \dots + w_n = 1 \end{cases}, \quad (2.7)$$

де λ_{max} – найбільше власне значення матриці A .

Отже, поєднуючи методи дерева рішень та методу аналізу ієрархій, можна досягти підвищення ефективності прийняття рішення щодо вибору фільму, що відповідає меті роботи.

2.3 Розробка алгоритмів функціонування інформаційної технології надання рекомендацій щодо підбору фільмів

На рисунку 2.3 зображено алгоритм роботи телеграм телеграм-бота, що реалізує інформаційну технологію підбору фільму.

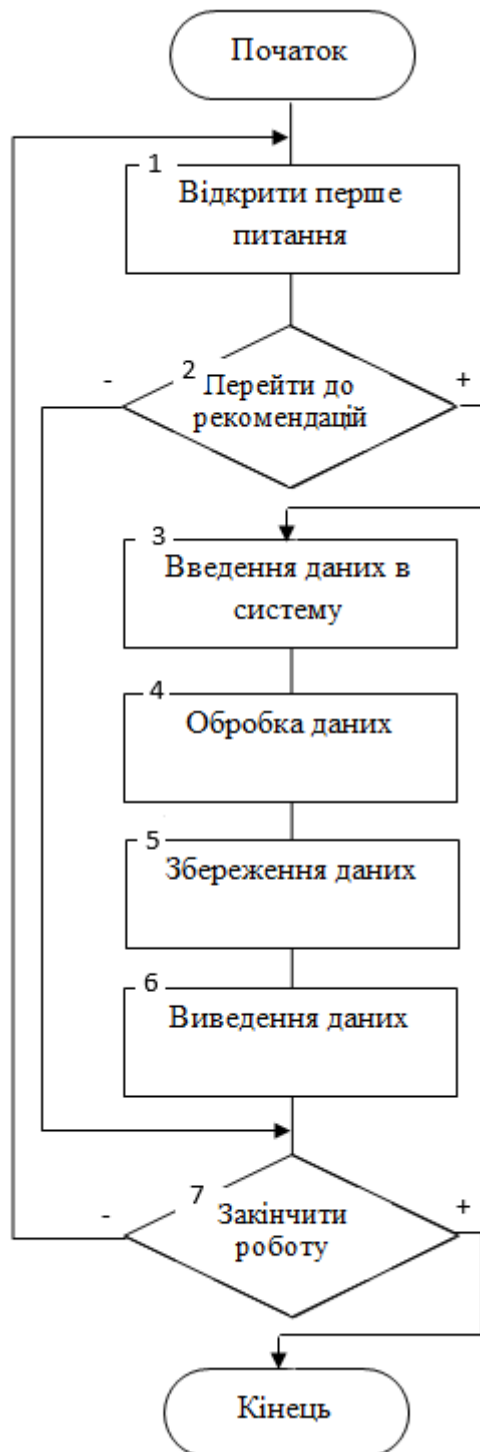


Рисунок 2.3 – Структурна схема алгоритму роботи телеграм-бота

Розглянемо алгоритм роботи телеграм-бота, що реалізує інформаційну технологію підбору фільму. Даний алгоритм полягає в наступному:

1. Користувач за допомогою Telegram API запускає бота, щоб отримати рекомендацію за допомогою експертної системи.
2. Початок циклу опитування користувача системою.
3. Користувач покроково відповідає на запитання системи починаючи від загальних.
4. Система здійснює перевірку вхідних даних, аналізує чи є передбачені рекомендації при відповідності даним характеристикам.
5. За допомогою відповідних логічних правил з бази правил формуються наступні запитання.
6. Вихідні результати у вигляді значень рекомендацій стосовно вибору фільму на кожний із доступних у базі правил варіант надаються користувачу через телеграм-бот.
7. Здійснюється перевірка на готовність до закінчення діалогу з системою.

Формування запитів користувачів показано на рисунку 2.4. Воно відбувається за алгоритмом діалогу на базі готових стандартних значень параметрів.

Вхідними даним для розробки інформаційної технології підбору фільмів є наступні питання:

- 1) Вкажіть бажаний жанр.
- 2) Вкажіть рік виходу.
- 3) Вкажіть країну створення.
- 4) Важливе вікове обмеження?
- 5) Вкажіть обмеження.
- 6) Оберіть режисера.
- 7) Якій звуковій доріжці надаєте перевагу?
- 8) Оберіть рейтинг IMDb.

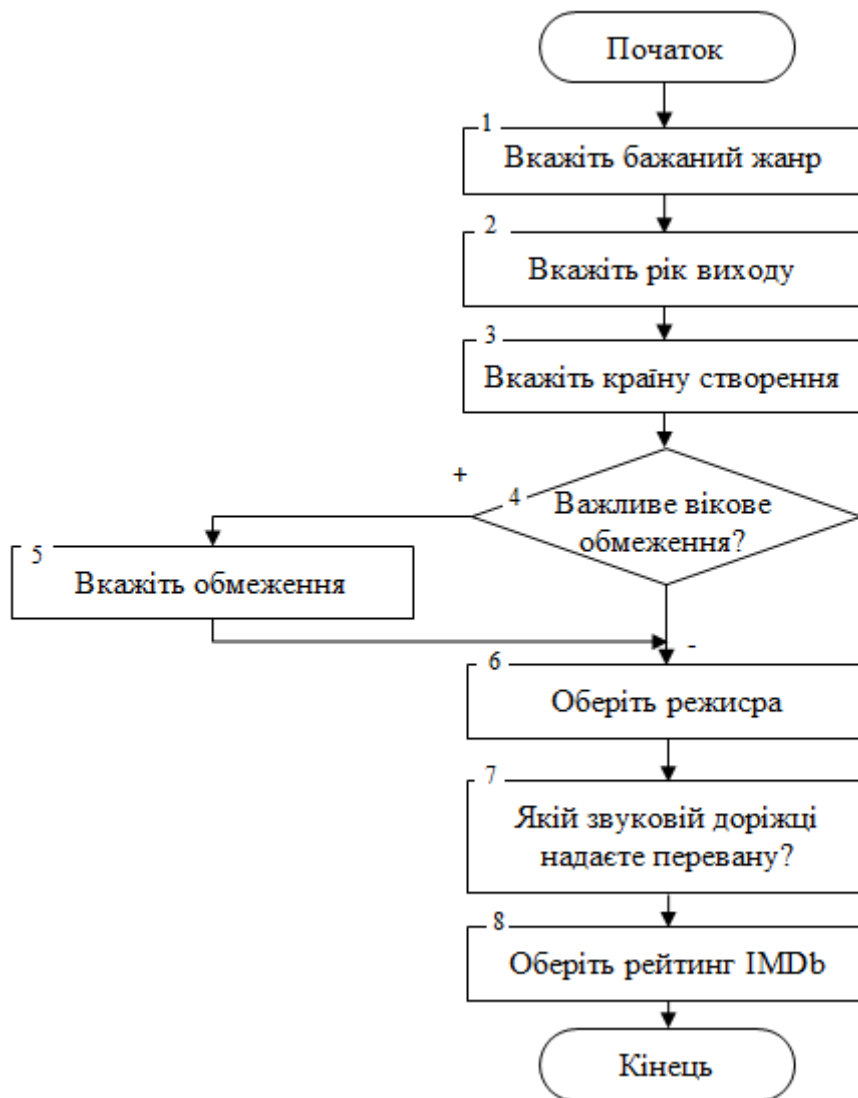


Рисунок 2.4 – Структурна схема алгоритму формування запитань

Розроблені запитання дозволять максимально точно вирішити проблему підбору фільму.

Розглянемо структурну схему загального алгоритму функціонування інформаційної технології надання рекомендацій щодо підбору фільмів (рис. 2.5).

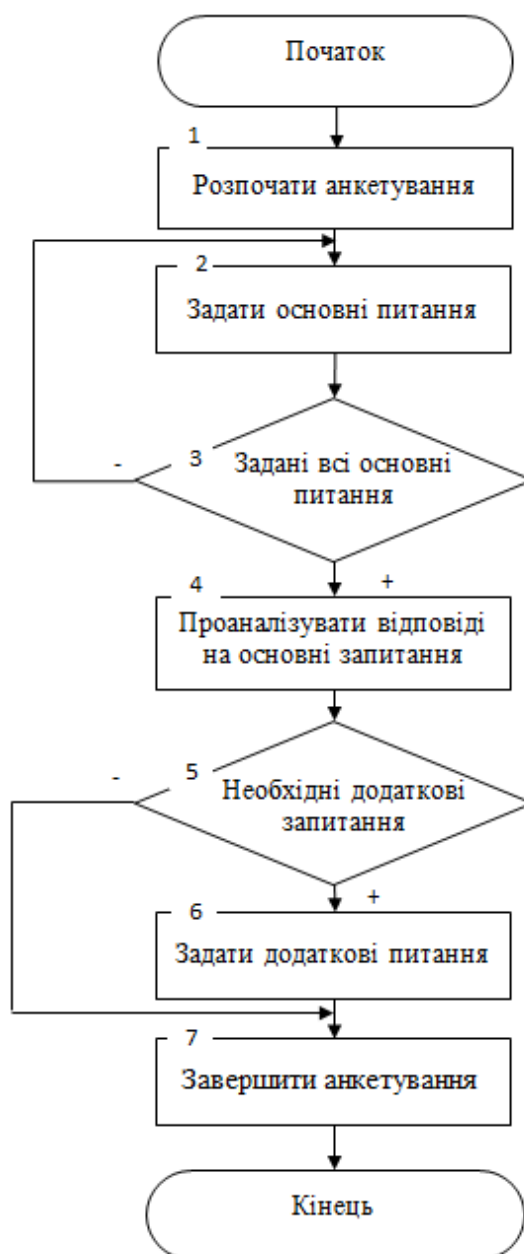


Рисунок 2.5 – Структурна схема загального алгоритму функціонування інформаційної технології надання рекомендацій щодо підбору фільмів

Даний алгоритм складається з наступних етапів:

1. Після початку роботи з програмою розпочинається анкетування користувача.
2. Задання основних питань користувачу.
3. Цикл опитування користувача системою для визначення вподобань користувача.
4. Аналіз відповідей на пройдений цикл опитування.

5. Перевірка на необхідність задати додаткові запитання.
6. Користувач покроково відповідає на додаткові запитання системи.
7. Система аналізує відповіді на додаткові запитання та формує результати.

Виведення результатів опитування у вигляді рекомендацій стосовно вибору фільму, що відповідає вподобанням користувача.

2.4 Висновок до розділу 2

Для розробки інформаційної технології підбору фільму необхідно застосовувати методи, що здатні забезпечити підвищення ефективності прийняття рішення щодо вибору фільму. Згідно специфіки задачі було запропоновано математичну модель для даної інформаційної технології на основі методу аналізу ієрархій для надання рекомендацій та методу дерева рішень, що дозволяє максимально задовольнити потреби користувача та забезпечує високу ефективність прийняття рішень щодо вибору фільму.

Розроблено алгоритми функціонування системи та схему формування запитань.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ ФІЛЬМІВ

3.1 Розробка структури інформаційної технології надання рекомендацій щодо підбору фільмів

Структуру програмного забезпечення інформаційної технології підбору фільмів наведено на рисунку 3.1.

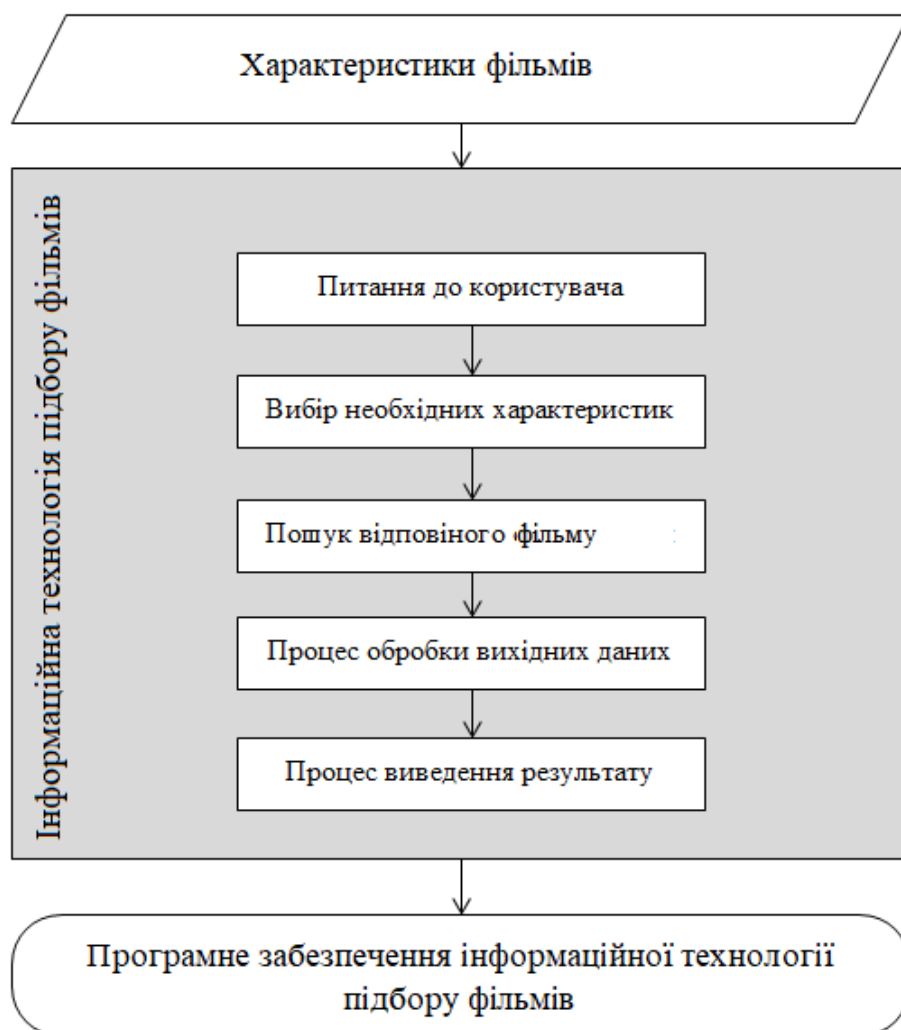


Рисунок 3.1 – Структура інформаційної технології надання рекомендацій щодо підбору фільмів

Реалізацію інформаційної технології підбору фільмів було вирішено зробити у вигляді чат-боту, а вибір платформи, яка найкраще підходить для підтримки чат-бота, впав на месенджер Telegram. Використовувати бота на цих платформах досить просто, потрібно вибрати його з каталогу, перейти за посиланням або знайти його за іменем за допомогою пошуку та почати спілкування з ним. Щоб запуснути бота, потрібно ввести команду /start (старт), або буде запропоновано натиснути кнопку «старт» на віртуальній клавіатурі. Для роботи можна використовувати програмовані кнопки віртуальної клавіатури, завдяки яким бот перетворюється на міні-додаток з інтуїтивно зрозумілим інтерфейсом. Тоді бот надішле інформацію про себе, інструкції, а також список або відобразить доступні команди або кнопки на екрані пристрою [18].

Telegram на сьогоднішній день пропонує тисячі чат-ботів на будь-який смак – від вивчення іноземних мов до спілкування з незнайомими людьми. Боти стали особливо популярними завдяки творцям месенджера Telegram, які одними з перших надали багато можливостей для розробників у цьому плані. Насправді програмісти-фрілансери мають можливість створювати інтерактивні інтерфейси з дуже широкими можливостями.

Алгоритм утиліт ботів досить простий. Повідомлення, команди та запити, надіслані користувачами, передаються програмному забезпеченню, що працює на серверах розробників. Анонімний проксі-сервер Telegram керує шифруванням та надає інформацію між утилітою та користувачем.

Взаємодія між користувачем і ботом виглядає наступним чином:

Користувач бота передає йому команду -> Бот надсилає команду серверу -> Програма на сервері обробляє запит, отриманий від бота -> Сервер дає відповідь боту -> Бот відображає відповідь на екран програми для користувача.

І цей цикл повторюється знову і знову, коли користувач натискає кнопки та взаємодіє з будь-яким ботом Telegram.

Спілкування із серверами відбувається за допомогою простого інтерфейсу HTTPS, який є спрощеною версією API Telegram. Крім того, цей інтерфейс може називатися каталогом програм або алгоритмом бота.

На рисунку 3.2 зображена структура інформаційної технології підбору фільмів.

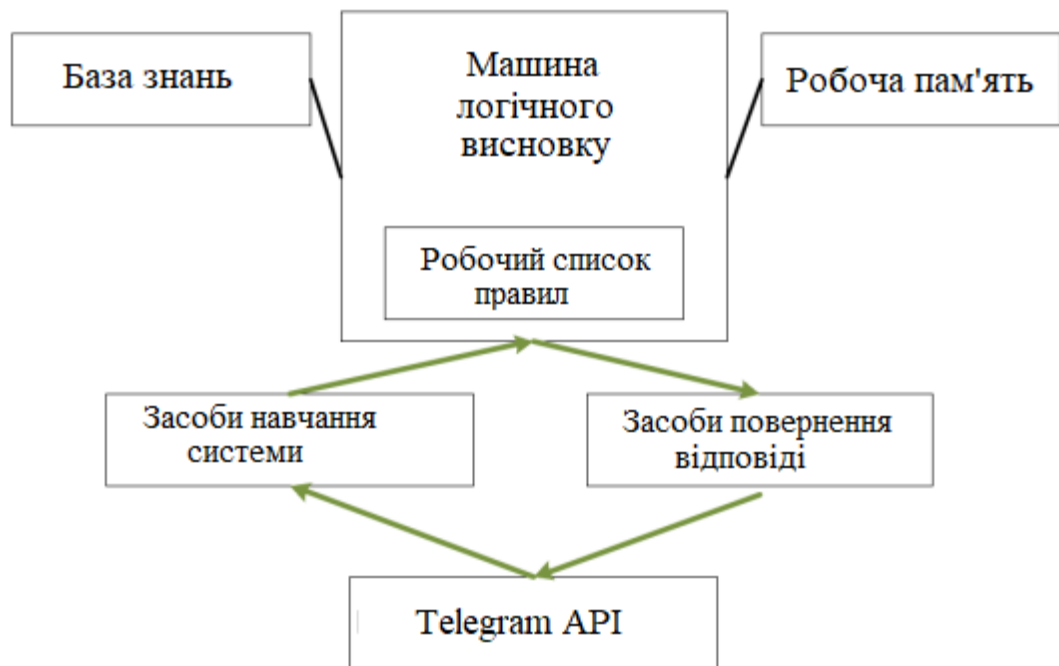


Рисунок 3.2 – Структурна схема інформаційної технології надання рекомендацій щодо підбору фільмів

Інформаційна технологія складається з описаних нижче компонентів:

- Telegram API. Механізм, за допомогою якого відбувається спілкування користувача і експертної системи.
- Засіб передачі відповіді від експертної системи до користувача. Компонент, що дозволяє користувачу отримати відповідь від системи.
- Машина логічного висновку. Програмний компонент, який забезпечує формування логічного висновку за певними правилами.
- Робочий список правил. Створений машиною логічного висновку і розташований за пріоритетами список правил, шаблони яких задовольняють фактам або об'єктам, що перебувають в робочій пам'яті.
- Засіб навчання. Автоматизований спосіб, що дозволяє користувачеві вводити знання в систему, а не залучати до вирішення завдання явного кодування знань інженера по знаннях.

3.2 Обґрунтування вибору мови та середовища програмування

Для реалізації інформаційної технології підбору фільмів вагому роль відіграє правильний вибір мови програмування, адже це впливає на швидкість та простоту розробки, а також від доцільного використання мови програмування залежить ефективність використання інформаційної технології кінцевим споживачем. Найчастіше для розробки програм у вигляді Telegram ботів застосовують мови програмування Java та Python. Саме тому необхідно розглянути більш детально ці дві мови програмування.

Java є однією з найпопулярніших мов програмування. Зараз нею користуються близько 9 мільйонів людей, а самі програми на Java використовуються на семи мільярдах пристроїв. Багато починаючі програмісти задаються питанням, навіщо вивчати цю мову і з чого почати [18].

Java – це універсальна мова програмування, яка використовується всюди, і найчастіше в таких сферах, як: мобільні програми, ігри, наукові розробки, мобільні телефони, ігрові приставки, суперкомп'ютери.

Java є основою майже всіх типів мережеских програм. У багатьох дослідженнях її називають мовою програмування №1 серед розробників.

Серед основних особливостей Java необхідно відзначити її кросплатформеність, ком'юніті, надійність, об'єктно-орієнтованість, відносну простоту та гнучкість. Розглянемо ці особливості більш детально.

Кросплатформеність. У Java є спеціальна програма, яка виконує код. Це віртуальна машина Java. Якщо програмний код написаний один раз, він працює з будь-якою апаратною платформою чи операційною системою: від смарт-карт до програм для розумного дому.

Спільнота. Оскільки Java є досить поширеною мовою, якою користується велика кількість розробників, можна знайти рішення практично для будь-якої проблеми, з якою ви можете зіткнутися під час роботи з Java. Тисячі бібліотек і форумів стануть у нагоді. На GitHub, наприклад, є відкриті проекти та

документація, а на форумі Stack Overflow ви можете звернутися за допомогою до спільноти.

Надійність. Мова Java строго типізована. Тобто будь-яка змінна або вираз вже має певний тип під час компіляції, що полегшує пошук певних проблем. Компілятор сам підказує програмісту, де він робить помилки.

Об'єктно-орієнтована. Усі бібліотеки, коли-небудь написані для Java, є класами, відповідальними за функціональність мови. Будь-який Java-додаток - це набір класів, які описують різні об'єкти. Це добре, оскільки дозволяє створювати складні, але прості в обслуговуванні програми. А взагалі Java є багатопарадигмальною мовою, тобто підтримує багато принципів програмування, що дозволяє ефективно вирішувати різноманітні завдання [19].

Відносна простота. Java не так легко зрозуміти, як Python, але вона набагато легша, ніж C або C++. Мовні функції оновлюються повільно, тому ви можете легко оновити до нових версій – вам не доведеться вчити все заново. Java – це строго типізована мова, що означає, що новачок завжди матиме можливість побачити помилку в коді під час компіляції.

Гнучкість. На Java можна розробити програму будь-якої складності.

Технічно Python – це об'єктно-орієнтована мова програмування високого рівня з вбудованою динамічною структурою, головним чином для підтримки розробки програмного забезпечення та Інтернет-додатків. Це надзвичайно цікаво в області швидкої розробки додатків, оскільки мова пропонує параметри динамічного введення тексту та функції динамічного зв'язування.

Python є відносно простою мовою, вона вимагає унікального синтаксису, який зосереджується на читанні. Розробники можуть читати та перекладати код Python набагато легше, ніж інші мови. У свою чергу, це зменшує витрати на обслуговування та розробку додатків, оскільки дозволяє командам працювати разом без значних мовних та досвідчених бар'єрів.

Крім того, Python підтримує використання модулів і пакетів, що означає, що програми можна розробляти в модульному стилі, а код можна повторно використовувати в проектах. Коли ви розробили потрібний модуль або пакет,

його можна масштабувати для використання в інших проектах, і ці модулі легко імпортувати чи експортувати.

Однією з найбільш багатообіцяючих переваг Python є те, що стандартна бібліотека та інтерпретатор є у вільному доступі, як у двійковій, так і у вихідній формі. Також немає ексклюзивності, оскільки Python і всі необхідні інструменти доступні на всіх основних платформах. Тому це привабливий варіант для розробників. Це робить Python доступним майже кожному [20].

Python – мова програмування загального призначення, тому її можна використовувати майже для будь-чого. Що ще важливіше, це інтерпретована мова, що означає, що написаний код фактично не перекладається в машиночитаний формат під час виконання.

Хоча більшість мов програмування виконують це перетворення перед виконанням програми. Цей тип мови також називають «мовою сценаріїв», оскільки спочатку він був призначений для використання в тривіальних проектах.

Концепція «мови сценаріїв» значно розвинулася з моменту її появи, тому що Python тепер використовується для написання великих бізнес-додатків, а не лише звичайних. Ця залежність від Python ще більше зросла зі зростанням популярності Інтернету. Переважна більшість веб-програм і платформ покладаються на Python, включаючи пошукову систему Google, YouTube і систему веб-торгівлі Нью-Йоркської фондової біржі (NYSE). Це чітко показує, що мова має бути досить серйозною, якщо вона працює в системі обміну.

NASA також використовує Python для програмування свого космічного обладнання та систем. Python також можна використовувати для обробки тексту, відображення чисел або зображень, вирішення наукових рівнянь і збереження даних. Python використовується для керування багатьма речами, які можуть знадобитися користувачеві або доступні на ПК чи мобільному пристрої [21].

Отже, розглянувши мови програмування, можна зробити висновок, що програмну реалізацію інформаційної технології підбору фільмів необхідно вести мовою програмування Java, що має перевагу у вигляді швидкої продуктивності,

порівняно з мовою Python, а також простотою розробки та наявністю бібліотек для створення телеграм-боту.

Було вирішено використовувати IntelliJ IDEA як програмний інструмент для написання програми. Це одне з найпопулярніших інтегрованих середовищ розробки, яке використовується для написання коду на багатьох мовах програмування (Java, Python, Kotlin тощо). Функції IntelliJ IDEA включають наступне:

- допомога в кодуванні (автозаповнення коду, зручна навігація по коду);
- інтегрована інтеграція для багатьох інструментів і технологій (Gradle, Maven, Git);
- можливість розширення базових функціональних можливостей (екосистема плагінів);
- підтримка багатьох мов програмування.

Отже, враховуючи перераховані вище переваги, для розробки інформаційної технології підбору фільмів було вибрано об'єктно-орієнтовану мову програмування Java та середовище розробки IntelliJ IDEA.

3.3 Обґрунтування вибору бібліотек та фреймворків для розробки інформаційної технології надання рекомендацій щодо підбору фільмів

Хоча Java – проста у використанні та вивченні мова, комплексність створення, налагодження та розгортання додатків Java досягла небувалих висот. Це сталося через величезну кількість змінних, з якими стикаються сучасні розробники мобільних та веб-додатків при роботі з такими поширеними технологіями, як платформи потокової передачі музики або мобільних платежів. Розробник, який пише простий додаток для певної галузі, змушений мати справу з безліччю бібліотек, модулів, журналів помилок і бібліотек обробки, інтеграцій з веб-службами, а також знати C#, Java, HTML та інші мови. Це призвело до закономірної потреби у спрощенні розробки програм Java, щоб заощадити час та

гроші розробників. З метою цього було створено платформи, фреймворки, бібліотеки та конфігурації, що полегшують розробнику роботу.

Існує велика кількість бібліотек для написання ботів Telegram. Кожен із них має свої недоліки та переваги: деякі з них підтримують обмежену кількість функцій Telegram, деякі доступні лише на одній мові програмування тощо. В якості такої бібліотеки було вирішено використовувати «telembots», яка є офіційною бібліотекою розробників месенджера Telegram і має велике співтовариство розробників та підтримується багатьма мовами програмування (Java, Python, JavaScript тощо).

Вона забезпечує доступ до більшості функцій Telegram, включаючи: обробку повідомлень користувачів, надсилання файлів різних форматів і розмірів тощо.

Приклад використання бібліотек для розробки інформаційної технології підбору фільму наведено в наступному коді:

```
package com.example.filmesgradle.bot;

import com.example.filmesgradle.processor.UpdateProcessor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.bots.TelegramLongPollingBot;
import org.telegram.telegrambots.meta.api.objects.Update;

@Component
public class TelegramBot extends TelegramLongPollingBot {

    @Value("${telegram.bot.username}")
    private String username;

    @Value("${telegram.bot.token}")
    private String token;

    private UpdateProcessor updateProcessor;
```

```

@Autowired
public void setUpdateProcessor(UpdateProcessor updateProcessor) {
    this.updateProcessor = updateProcessor;
}

@Override
public String getBotUsername() {
    return username;
}

@Override
public String getBotToken() {
    return token;
}

@Override
public void onUpdateReceived(Update update) {
    updateProcessor.process(update);
}
}

```

В якості основи для створення сервісу було обрано Spring Boot. Це одна з найпопулярніших бібліотек мови Java, яка підтримує інтеграцію з багатьма технологіями та використовує готовий сервер Tomcat для розгортання програми.

Spring Boot – це фреймворк на основі Java з відкритим кодом, який використовується для створення мікросервісу. Він розроблений Pivotal Team і використовується для створення автономних і готових до виробництва Spring додатків.

Spring Boot дозволяє легко створювати автономні додатки на основі Spring продуктивного класу, які можна «просто запускати». Більшість програм Spring Boot потребують мінімальної конфігурації Spring.

Особливості конфігурації Spring:

- Можливе створення автономних програм Spring;
- вставка Tomcat, Jetty або Undertow безпосередньо (не потрібно розгортати WAR-файли);
- надавання впевнених «початкові» залежностей, щоб спростити конфігурацію збірки;
- автоматичне налаштування Spring і бібліотеки сторонніх розробників, коли це можливо;
- можливо надати готові до виробництва функції, такі як показники, перевірки працездатності та зовнішня конфігурація;
- абсолютна відсутність генерації коду та жодних вимог щодо конфігурації XML.

Інформаційна технологія підбору фільмів містить базу даних, в якій зберігається інформація про фільми, що можуть зацікавити користувача. Доступ до цієї бібліотеки забезпечується за допомогою використання бібліотеки jOOQ, що надає розробнику можливість писати SQL-запити як ланцюжок викликів методів.

Бібліотека створює класи моделі для таблиць бази даних. З такими класами запити стають типобезпечними. Під час компіляції виявляються помилки невідповідності типу параметра, відсутність стовпців у таблицях, тощо. При використанні звичайних рядків для запитів SQL такі помилки виникають під час запуску програми, внаслідок чого її робота стає некоректною. Недоліком бібліотеки є неповна підтримка синтаксису SQL і неможливість розширити перетворення результатів запиту в об'єкти Java. Ця бібліотека має такі функції, як отримання готового до використання рядка запиту SQL без його виконання. Це дає змогу створити адаптер для використання типобезпечних запитів в асинхронному контексті [22].

3.4 Опис класів і функцій інформаційної технології надання рекомендацій щодо підбору фільмів

Інформаційна технологія підбору фільмів включає взаємодію користувача (актора) та системи. Детальна діаграма прецедентів для опису моделі функціональних відношень між користувачем та прецедентами в системі зображена на рисунку 3.3.

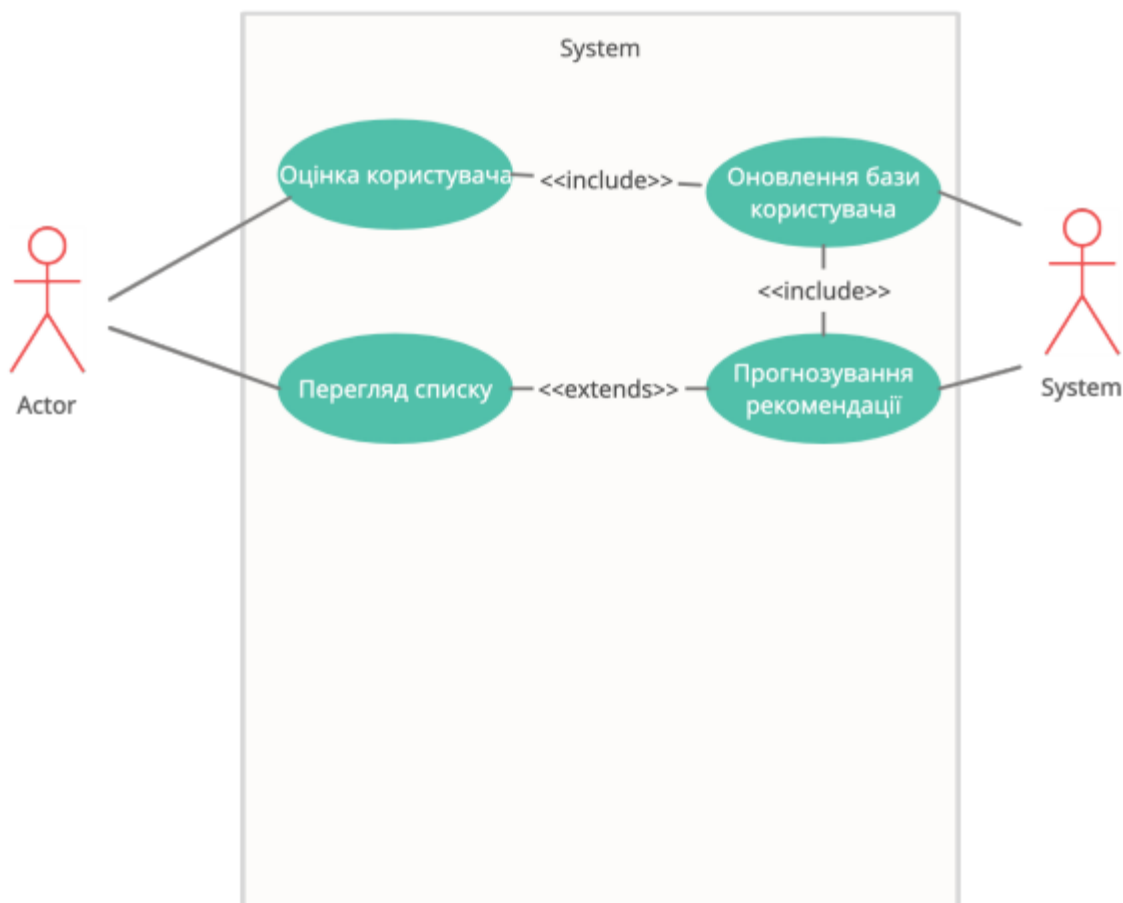


Рисунок 3.3 – Діаграма прецедентів інформаційної технології надання рекомендацій щодо підбору фільмів

Розроблена UML-діаграма інформаційної технології підбору фільмів зображена на рисунку 3.4.

В UML-діаграмі наведено наступні класи: FilmEsGradleApplication, MessageHandler, TelegramBot, AnswerCache, AnswerWeriantCache, QuestionCache, ModelMapperConfig, FilmDao, Film, TestResult, FilmsRecord,

Films, DefaultCatalog, DefaultSchema, Keys, Tables, MessageSender, MessageSenderImpl, UpdateProcessorImpl, UpdateProcessor, TestResultService, TestService.

Клас QuestionCache містить питання до користувача, такі як «Вкажіть бажаний жанр», «Вкажіть рік виходу», «Вкажіть країну створення», «Важливе вікове обмеження», «Вкажіть обмеження», «Ведіть улюбленого актора», «Вкажіть тривалість», «Оберіть режисера», «Якій звуковій доріжці надаєте перевагу», «Оберіть рейтинг IMDb», а також відповідність питання до можливих варіантів відповідей на них. Натомість в класі AnswerWeriantCache міститься набір усіх можливих відповідей на запропоновані запитання, наприклад: «Драма», «Фантастика», «Бойовик», «Пригоди», «Комедія», «Триллер», «Жахи», «Інше», «1990-2000», «2000-2010», «2010-2020», «2020-2022» та інші.

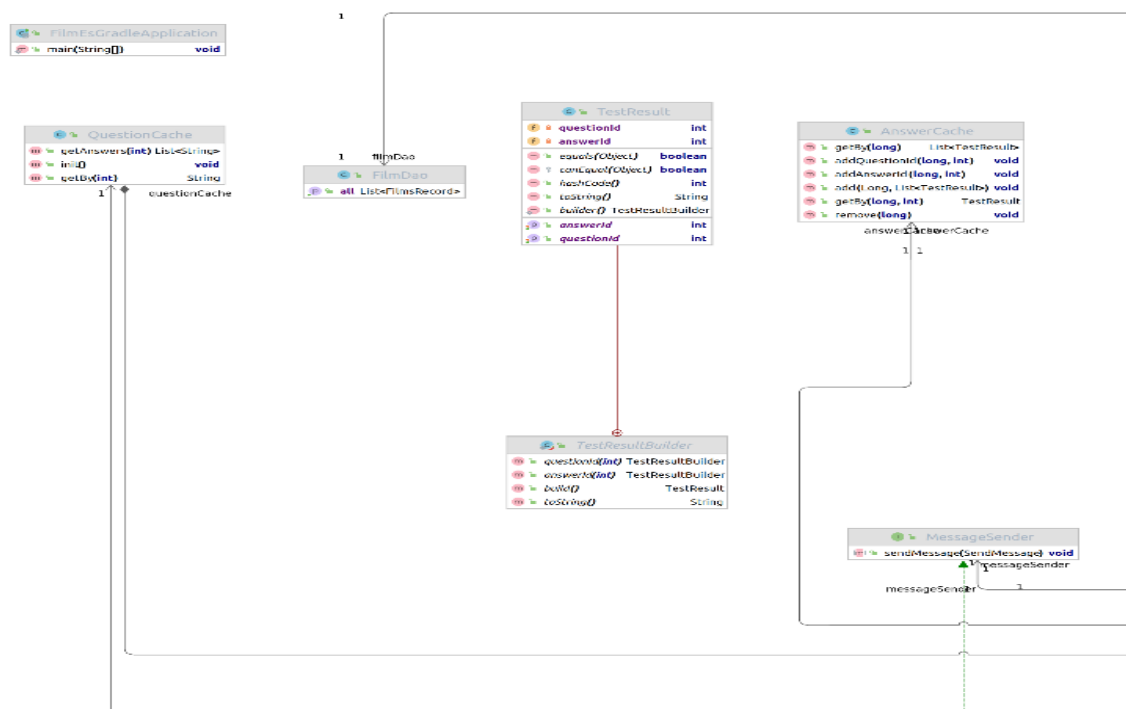


Рисунок 3.4 – Частина UML-діаграми інформаційної технології надання рекомендацій щодо підбору фільмів

Класи MessageHandler та FilmDao – класи, створені за допомогою бібліотеки Lombok, а клас TelegramBot – Spring Boot. Класи MessageSenderImp

та TelegramBot відповідають взаємодію з платформою Telegram. Доцільно використовувати цей фреймворк TelegramBots-Spring-Boot-Starter для реалізації інтерфейсу Telegram Bot API.

Клас MessageHandler відповідає за надсилання відповідей та повідомлень користувачу. Клас TestService відповідає за організацію процесу тестування,

У класі TestResultService знаходиться логіка програми, що відповідає за прийняття рішення щодо вибору фільму. Клас містить інтелектуальний алгоритм підбору фільму за допомогою поєднання дерева рішень та методу аналізу ієрархій.

Отже, у рамках даного підрозділу було здійснено реалізацію інформаційної технології підбору фільму, наведено діаграму прецедентів системи, розроблено загальну UML-діаграму класів та описано деякі з них.

3.5 Тестування інформаційної технології надання рекомендацій щодо підбору фільмів

Тестування – процес перевірки відповідності заявленим вимогам до продукту та фактично реалізованим функціональним можливостям, який відбувається шляхом спостереження за його роботою в штучно створених ситуаціях та на обмеженому наборі вибраних тестів у визначений спосіб.

У таблиці 3.1 наведено результати тестування інформаційної технології, за якими здійснювалась оцінка відповідності інформаційної технології поставленим вимогам.

Таблиця 3.1 – Результати тестування програмного забезпечення

Тип характеристики	Значення
Швидкодія	0,4 с
Швидкість доступу до даних	140 мкс
Швидкість обробки запитів	100 од/хв
Використання пам'яті	1024 Мб

На рисунку 3.5 наведено початок роботи з телеграм-ботом, коли для отримання першого (початкового) питання інформаційної технології підбору фільму необхідно ввести «/start», що запускає бота.



Рисунок 3.5 – Загальний вигляд інтерфесного вікна «Початок роботи»

На рисунку 3.6 зображене початкове питання про бажаний жанр, де передбачені кнопки з можливими варіантами відповіді: «Драма», «Фантастика», «Бойовик», «Комедія», «Пригоди», «Триллер», «Жахи», «Інше».

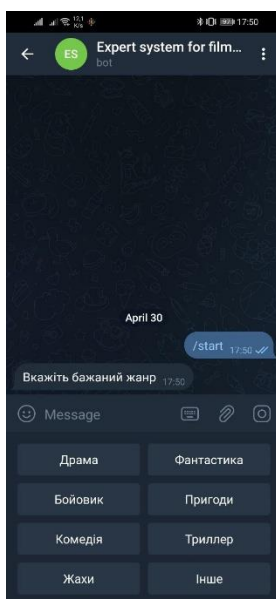


Рисунок 3.6 – Загальний вигляд інтерфесного вікна «Перше питання системи»

На рисунку 3.7 і 3.8, наведено вікна, що містять наступні питання системи та результат, проведеного опитування у вигляді рекомендованих для перегляду фільмів, що відповідають побажанням користувача.

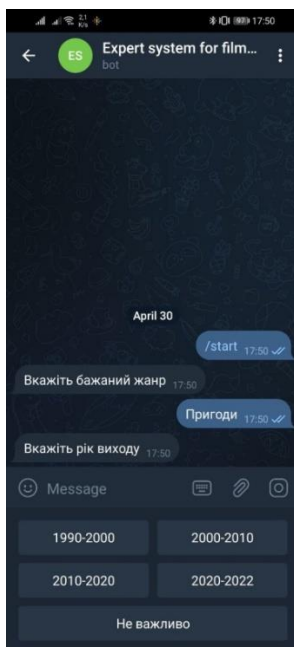


Рисунок 3.7 – Загальний вигляд інтерфесного вікна «Наступне питання щодо даної ситуації»

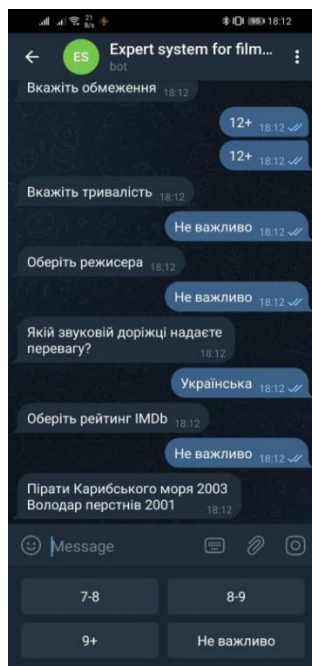


Рисунок 3.8 – Загальний вигляд інтерфесного вікна «Результат роботи інформаційної технології надання рекомендацій щодо підбору фільму»

Тестування результатів підбору фільмів відбувалося за допомогою опитування групи респондентів, що складалася зі 100 людей. Кожен з них підбирав фільм за допомогою розробленої інформаційної технології та додатку «Джарвіс», після чого мали змогу оцінити відповідність фільму їх запити. Оцінювання відбувалося за 10-ти бальною шкалою. Наведено результати тестування в таблиці 3.2, де буде наведено кількість користувачів, які ставили відповідну оцінку кожній з розглянутих систем підбору фільму.

Таблиця 3.2 – Оцінювання результатів респондентами

Оцінка відповідності підбору фільму	Кількість респондентів розробленого додатку	Кількість респондентів аналогу «Джарвіс»
1	3	3
2	1	2
3	7	5
4	3	8
5	2	7
6	15	14
7	5	19
8	14	16
9	28	17
10	22	9

З таблиці 3.2 випливає, що більше користувачів поставили високі оцінки розробленому додатку. Обрахувавши точність роботи інформаційної технології, виходячи з наведених даних, виявимо, що розроблений додаток має точність 75,7% проти 71,6% аналогу «Джарвіс», що означає покращення точності підбору фільму на 4,1%.

3.6 Висновок до розділу 3

В третьому розділі було розроблено структуру інформаційної технології підбору фільмів, структуру програмного забезпечення інформаційної технології підбору фільмів.

Здійснено обґрунтування вибору мови програмування для реалізації інформаційної технології підбору фільмів. Інформаційну технологію розроблено на об'єктно-орієнтованій мові програмування Java, враховуючи її кросплатформність, об'єктно-орієнтованість та простоту розробки, в середовищі розробки IntelliJ IDEA, що має переваги у вигляді глибокого розуміння коду, задоволенні усіх вимог розробника, розумного доповнення коду. Розроблено та описано діаграму класів, розглянуто методи та атрибути розроблених класів. Програмно реалізовано інформаційну технологію підбору фільмів.

Тестування точності вибору фільму відбувалось за допомогою групи респондентів, які оцінювали точність рекомендації підбору фільму інформаційною технологією та аналогом. У ході тестування виявлено збільшення точності надання рекомендації щодо підбору фільму на 4,1% порівняно з аналогом, що значно покращить досвід користування з програмою для користувача. Отже мету МКР досягнуто.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу інформаційної технології підбору фільмів, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів, якими можуть бути провідні викладачі випускової або спорідненої кафедри чи інші відомі фахівці. Не рекомендується залучати експертами керівника магістерської кваліфікаційної роботи та завідувача відповідної випускової кафедри. В якості програми-аналога взято програму «Джарвіс» з вартістю 711 400 грн.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [23].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за п'ятибальною шкалою)					
	0	1	2	3	4
1	2	3	4	5	6
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження таблиці 4.1

Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижча за ціни аналогів	Ціна продукту значно нижча за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Продовження таблиці 4.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5ти років. Термін окупності інвестицій більший 5ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій менший 3-х років

Продовження таблиці 4.1

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що потребує значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту потребує незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	---	--	--	---

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	3	4
2. Ринкові переваги (наявність аналогів)	4	4	3
3. Ринкові переваги (ціна продукту)	3	4	4
4. Ринкові переваги (технічні властивості)	4	3	4
5. Ринкові переваги (експлуатаційні витрати)	3	4	4
6. Ринкові перспективи (розмір ринку)	4	4	3
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	3	4	2

Продовження таблиці 4.2

9. Практична здійсненність (наявність фінансів)	3	3	4
10. Практична здійсненність (необхідність нових матеріалів)	4	4	3
11. Практична здійсненність (термін реалізації)	3	3	3
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	41	42	40
Середньоарифметична сума балів $СБ_c$	$СБ_c = \frac{\sum_i^3 СБ_i}{3} = 41$		

За результатами розрахунків, наведених в таблиці 4.2, робиться висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використовують рекомендації, наведені в табл. 4.3.

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ$, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вищий середнього
21...30	Середній
11...20	Нижчий середнього
0...10	Низький

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високою, що досягається за рахунок удосконаленому алгоритму, який поєднує в собі алгоритм на основі методу аналізу ієрархій для надання рекомендацій та методу дерева рішень, що дозволяє максимально задовольнити потреби користувача та забезпечує високу ефективність прийняття рішень щодо вибору фільму.

Також, використання розробленої програми є значно дешевшим для кінцевого споживача. Якщо для аналога затрати складають 380 грн щомісячно,

то для розроблюваної програми щомісячні витрати плануються на рівні 180-200 грн.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної, дослідноконструкторської, конструкторсько-технологічної роботи, створенням дослідного зразка і здійсненням виробничих випробувань, під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуються за такими показниками [24]:

- витрати на оплату праці;
- відрахування на соціальні заходи;
- матеріали;
- паливо та енергія для науково-виробничих цілей;
- витрати на службові відрядження;
- спецустаткування для наукових (експериментальних) робіт;
- програмне забезпечення для наукових (експериментальних) робіт;
- витрати на роботи, які виконують сторонні підприємства, установи і організації;
- інші витрати;
- накладні (загальновиробничі) витрати.

4.2.1 Витрати на оплату праці

До витрат на оплату праці належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в

організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_i = 1 \frac{k \times M_{ni} \times t_i}{T_p}, \quad (4.1)$$

де k – кількість посад дослідників, залучених до процесу досліджень; M_{ni} – місячний посадовий оклад конкретного дослідника, грн; t_i – кількість днів роботи конкретного дослідника, дн.; T_p – середня кількість робочих днів в місяці, $T_p=21 \dots 23$ дні. Проведені розрахунки потрібно звести до таблиці.

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Кількість днів роботи	Витрати на заробітну плату, грн
<i>Керівник проекту</i>	35 000	1 590,9	25	39 773
<i>Науковий співробітник</i>	20 000	909,1	25	22 727
Всього				62 500

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розраховується як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_d = (Z_o + Z_p) \times \frac{H_{\text{дод}}}{100\%}, \quad (4.2)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати.

$$З_д = (62500 * 12 \% / 100 \%) = 7\ 500 \text{ (грн.)}$$

4.2.2 Відрахування на соціальні заходи

До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне страхування та для здійснення заходів щодо соціального захисту населення (ЄСВ – єдиний соціальний внесок).

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$H_з = (З_о + З_р + З_{дод}) \times \frac{H_{зп}}{100\%}, \quad (4.3)$$

де $H_{зп}$ – норма нарахування на заробітну плату.

$$H_з = (62\ 500 + 7\ 500) \cdot 22 \% / 100 \% = 15\ 400 \text{ (грн.)}$$

4.2.3 Амортизація обладнання, програмних засобів та приміщень

До амортизації обладнання, програмних засобів та приміщень відносять амортизаційні відрахування по кожному виду обладнання, устаткування та інших приладів і пристроїв, а також програмного забезпечення для проведення науково-дослідної роботи, в дослідній організації або на підприємстві.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \times \frac{T_{вик}}{12}, \quad (4.4)$$

$$A_{\text{обл}} = \frac{51210}{2} \times \frac{1,25}{12} = 2667,$$

де C_b – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн; $t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців; T_e – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки необхідно звести до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Комп'ютер (Ноутбук ASUS ROG Strix G17 44999 грн, монітор Lenovo 24" S24E-20 6211грн)	51210	2	1,25	2667
Приміщення	800000	20	1,25	4166
Всього				6833

4.2.4 Паливо та енергія для науково-виробничих цілей

До палива та енергії для науково-виробничих цілей належать витрати на придбання у сторонніх підприємств, установ і організацій будь-якого палива, що витрачається з технологічною метою на проведення досліджень. Це у разі виконання енергоємних наукових досліджень за методом прямого внесення витрат і досягає значної питомої ваги у собівартості досліджень.

Витрати на силову електроенергію (B_e) розраховують за формулою:

$$B_e = \sum \frac{W_{yi} \times t_i \times C_e \times K_{впi}}{\eta_{i=1}}, \quad (4.5)$$

$$B_e = \sum \frac{0,096 \times 200 \times 6.2 \times 0,9}{\eta_{i=1}} = 107,136,$$

$$B_e = \sum \frac{0,036 \times 200 \times 6.2 \times 0,9}{\eta_{i=1}} = 40,176,$$

$$B_e = \sum \frac{0,024 \times 60 \times 6.2 \times 0,9}{\eta_{i=1}} = 8,04,$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год; C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії); $K_{впi}$ – коефіцієнт, що враховує використання потужності, $K_{впi} < 1$; η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Проведені розрахунки необхідно звести до таблиці.

$$C_e = (C_{опт} + C_{розп} + C_{пост}) \left(1 + \frac{ПДВ}{100\%}\right), \quad (4.6)$$

$$C_e = (4,76606 + 0,19947 + 0,19947) \left(1 + \frac{20\%}{100\%}\right) = 6,2,$$

де $C_{опт}$ - середня оптова ціна електроенергії, яка визначається оператором ринку (без ПДВ), грн за 1 кВт·год; $C_{розп}$ - вартість розподілу електроенергії окремою енергорозподільною компанією (без ПДВ), грн за 1 кВт·год; $C_{пост}$ - вартість постачання електроенергії від енергорозподільної компанії до конкретного споживача (без ПДВ), грн за 1 кВт·год. ПДВ - величина податку на додану вартість, %, у 2022 році ПДВ=20%.

Таблиця 4.6 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук (ASUS ROG Strix G17)	0,096	200	500,58
Монітор (Lenovo 24"S24E-20)	0,036	200	187,72
Освітлення	0,024	60	37,54
Всього			725,85

4.2.5 Інші витрати

До інших витрат належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Інші витрати розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (З_{\text{о}} + З_{\text{р}}) \times \frac{H_{\text{ів}}}{100\%}, \quad (4.7)$$

де $H_{\text{ів}}$ – норма нарахування за статтею «Інші витрати».

$$I_{\text{в}} = 62500 * 55\% / 100\% = 67999,8 \text{ (грн.)}.$$

4.2.6 Накладні

До накладних (загальновиробничих) витрат належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Накладні витрати розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (Z_o + Z_p) \times \frac{H_{\text{нзв}}}{100\%}, \quad (4.8)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{\text{нзв}} = 62500 * 125\% / 100\% = 78125 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{доо}} + Z_n + M + K_v + B_{\text{снец}} + B_{\text{прг}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сн}} + I_s + B_{\text{нзв}}. \quad (4.9)$$

$B_{\text{заг}} = 62500 + 7500 + 15400 + 6833 + 725,85 + 67999,8 + 78125 = 239083,65$
грн.

Загальні витрати ZB на завершення науково-дослідної (науковотехнічної) роботи та оформлення її результатів розраховуються за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (4.10)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науководослідної роботи. Так, якщо науково-технічна розробка знаходиться на стадії: науководослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta =0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$.

$$ЗВ = 239083,65 / 0,5 = 478\,167,3 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки [25].

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- а)* вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- б)* зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 4-х років після її впровадження);
- в)* кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- г)* визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу.

При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

4.4 Розробка чи суттєве вдосконалення інформаційної технології для використання масовим споживачем

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta C_0 \cdot N + C_0 \cdot \pm\Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.11)$$

де $\pm\Delta C_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу; N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки; C_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_0 = C_0 \pm \Delta C_0$; C_0 – вартість програмного продукту у році до впровадження результатів розробки; ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик; λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%,

а коефіцієнт $\lambda = 0,8333$. p – коефіцієнт, який враховує рентабельність продукту;
 ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 2160 грн. за програмний продукт, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 400 грн. Кількість проданих підписок на програмний продукт також збільшиться: протягом першого року – на 4000 шт., протягом другого року – на 6000 шт., протягом третього року на 10000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*400 + (2160 + 400)*4000)*0,8333*0,3) * (1 - 0,18) = 2\,099\,116 \text{ грн.}$$

$$\Delta\Pi_2 = (0*400 + (2160 + 400)*(4000 + 6000)*0,8333*0,3) * (1 - 0,18) = 5\,247\,790 \text{ грн.}$$

$$\Delta\Pi_3 = (0*400 + (2160 + 400)*(4000 + 6000 + 10000)*0,8333*0,3) * (1 - 0,18) = 10\,495\,580 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 17 842 486 грн.

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-дослідної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.12)$$

де $\Delta\Pi_i$ збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної роботи, грн; T – період часу, протягом якого виявляються результати впровадженої науково-дослідної роботи, роки; τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\text{ПП} = (2\,099\,116 / (1 + 0,1)^1) + (5\,247\,790 / (1 + 0,1)^2) + (10\,495\,580 / (1 + 0,1)^3) = 1\,908\,287 + 4\,337\,016 + 7\,885\,485 = 14\,130\,788 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \times \text{ЗВ}, \quad (4.13)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}} = 2 \dots 5$, але може бути і більшим; ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 478167,3 = 956\,334,6 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV, \quad (4.14)$$

$$E_{\text{абс}} = 14\,130\,788 - 956\,334,6 = 13\,174\,453,4 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої

норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B . Для цього використаємо формулу:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.15)$$

$T_{ж}$ життєвий цикл наукової розробки, роки.

$$E_B = \sqrt[3]{(1 + 13\,168\,159,56 \div 956\,334,6)} - 1 = 1,45$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.16)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,23 \dots 0,25)$; f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,5)$.

$$\tau_{min} = 0,25 + 0,05 = 0,29.$$

Так як $E_B > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}, \quad (4.17)$$

$$T_{\text{ок}} = 1 / 1,45 = 0,69 \text{ р.}$$

Оскільки < 3 -х років, а саме термін окупності рівний 0,69 року, то фінансування даної наукової розробки є доцільним.

4.5 Висновок до розділу 4

Економічна частина даної роботи містить розрахунок витрат на розробку інформаційної технології підбору фільмів, сума яких складає 478 167,3 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є високо конкурентоспроможним.

ВИСНОВКИ

- Проведено аналіз програм аналогів, систем-аналогів та обґрунтування доцільності розробки інформаційної технології підбору фільмів;
- розроблено математичну модель підбору фільмів інформаційної технології надання рекомендацій щодо підбору фільму;
- розроблено структуру інформаційної технології підбору фільмів;
- здійснено програмну реалізацію інформаційної технології підбору фільмів;
- проведено тестування програми для надання рекомендацій щодо підбору фільму та проаналізувати отримані результати;
- економічно обґрунтовано доцільність розробки інформаційної технології надання рекомендацій щодо підбору фільмів.

У ході виконання магістерської кваліфікаційної роботи реалізовано інформаційну технологію надання рекомендацій щодо підбору фільмів. Для досягнення мети були поставлені такі завдання: провести аналіз програм аналогів, систем-аналогів та обґрунтувати доцільність розробки інформаційної технології надання рекомендацій щодо підбору фільмів; розробити математичну модель надання рекомендацій щодо підбору фільмів; розробити структуру інформаційної технології надання рекомендацій щодо підбору фільмів; здійснити програмну реалізацію інформаційної технології надання рекомендацій щодо підбору фільмів; провести тестування програми та проаналізувати отримані результати; економічно обґрунтувати доцільність розробки інформаційної технології підбору фільмів.

Обґрунтовано доцільність розробки інформаційної технології підбору фільмів. Проведено аналіз сучасних аналогів та систем-аналогів, які створені для надання рекомендацій щодо вибору фільмів та формування рейтингової оцінки фільму та коротко охарактеризовано їх головні переваги та недоліки, а саме відсутність орієнтованості в багатьох аналогах на україномовного користувача, обмежена кількість фільмів та серіалів, можливість ефективного надання

рекомендацій лише завдяки існуючому профілю та при реєстрації, що зумовлює проблему «холодного старту», що довело актуальність розробки інформаційної технології підбору фільмів.

На основі аналізу існуючих моделей та методів, що застосовуються для вирішення поставленої задачі було обґрунтовано використання в інформаційній технології методи, що здатні забезпечити підвищення ефективності прийняття рішення щодо вибору фільму. Згідно специфіки задачі було запропоновано математичну модель для даної інформаційної технології на основі методу аналізу ієрархій для надання рекомендацій та методу дерева рішень, що дозволяє максимально задовольнити потреби користувача та забезпечує високу ефективність прийняття рішень щодо вибору фільму.

Було розроблено структуру інформаційної технології надання рекомендацій щодо підбору фільмів та спроектовано схему алгоритму роботи розроблюваної програми. Наведено алгоритм формування запитань.

Здійснено програмну реалізацію інформаційної технології надання рекомендацій щодо підбору фільмів. Обґрунтовано вибір мови програмування. Розробка інформаційної технології велася на об'єктно-орієнтованій мові програмування Java в середовищі розробки IntelliJ IDEA. Розглянуто бібліотеки та фреймворки, які доцільно використовувати в інформаційній технології підбору фільмів, детально описано класи та функції, які використовуються в інформаційній технології.

Проведено тестування розробленої програми та підтверджено її працездатність. Порівняно з прототипом виявлено збільшення точності підбору фільму на 4,1%, що означає досягнення поставленої мети.

Здійснено економічне обґрунтування доцільності розробки інформаційної технології підбору фільмів. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, знайдено термін окупності витрат для виробника та економічний ефект для споживача при використанні такої розробки. У результаті аналізу розрахунків можна зробити

висновок, що розробка у виробництві та використання дешевша за аналог і є висококонкурентоспроможною. Період окупності складе близько 0,69 роки. Загальні витрати становлять 481 314,22 грн.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шевчук О.Р.; Озеранський В.С. Information technology for providing recommendations on movie selection. – Всеукраїнська науково-практична інтернет-конференція «Молодь в науці : дослідження, проблеми, перспективи», Ukraine, May. 2022. Available at: <https://conferences.vntu.edu.ua/index.php/all-fbtegp/all-fbtegp2022/paper/view/1578960/137808>
2. IMDb [Електронний ресурс]. URL: <https://www.imdb.com/> (дата звернення: 14.04.2022).
3. Яровий А. А. Експертні системи. Частина 2 : навчальний посібник / Яровий А. А., Арсенюк І. Р., Месюра В. І. – Вінниця : ВНТУ, 2017. – 106 с.
4. Олійник А.О. Побудова асоціативних правил на основі інтелектуального стохастичного пошуку / А.О. Олійник // Математичні машини і системи. – 2015. – № 4. – с. 45-56, URL: http://nbuv.gov.ua/UJRN/MMS_2015_4_6 (дата звернення: 20.02.2022).
5. Сніжинський М.В., Нескородева Т.В. Аналіз даних про рейтинг фільмів на платформі оцінювання IMDb засобами мови R. Матеріали всеукраїнської науковопрактичної конференції для студентів, аспірантів та молодих вчених "Прикладні інформаційні технології" (29 квітня 2020 року) - Вінниця: ДонНУ імені Василя Стуса. с.57- 58.
6. J. Cendrowska (1987). PRISM: An algorithm for inducing modular rules. *International Journal of ManMachine Studies*. 27(4):349-370.
7. Циганков О. В. Основи проектування систем штучного інтелекту: Конспект лекцій / Одеський національний політехнічний університет. Одеса, 2006. 195 с.
8. Сорока П. М. Вирішення проблем прийняття рішень в умовах невизначеності за допомогою дерева рішень / П. М. Сорока, О. Є. Іларіонов // Вісник Нац. техн. ун-ту "ХПІ" : зб. наук. пр. Сер. : Механіко-технологічні системи та комплекси. – Харків : НТУ "ХПІ", 2016. – № 50 (1222). – С. 106-111.

9. Сокур Д. В. Нечітке дерево розв'язків як метод автоматичного аналізу даних / Д. В. Сокур; наук. кер. О. Л. Блохін // Наукові розробки молоді на сучасному етапі : тези доповідей XVII Всеукраїнської наукової конференції молодих вчених та студентів (26-27 квітня 2018 р., Київ). - Київ : КНУТД, 2018. - Т. 2 : Мехатронні системи і комп'ютерні технології. Ресурсозбереження та охорона навколишнього середовища. - С. 493-494.
10. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. – Запоріжжя: ЗНТУ, 2008. – 341 с.
11. Гаврилова Т.А. Базы знаний интеллектуальных систем/Т.А. Гаврилова, В.Ф. Хорошевский – СПб.: Питер, 2001. – 384 с.
12. Месюра В.І., Яровий А.А., Арсенюк І.Р. Експертні системи. Частина 1. Навчальний посібник. – Вінниця: ВНТУ, 2006.– 114 с.
13. Алексєєв А.А., Костіна Н.І., Кононець О.Я. Фінансово-економічні експертні системи. Навчальний посібник // За ред. Н.І. Костіної. – К.: Видавничий дім „Скарби”, 2004. – 208 с.
14. Джонс М.Т. Программирование искусственного интеллекта в приложениях. Пер. с англ. Осипов А.И. – М.: ДМК Пресс, 2004. – 312 с.
15. Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям (+CD): Учебное пособие. 2-е изд., испр. – СПб: Питер, 2013. – 704 с.: ил.
16. Левитин, Ананий В. Алгоритмы: введение в разработку и анализ.: Пер. с англ. – М.: Издательский дом “Вильямс”, 2006. – 576 с.: ил. – Парал. тит. англ.
17. Кушлик-Дивульська О.І. Основи теорії прийняття рішень. Навчальний посібник. /О.І. Кушлик-Дивульська, Б.Р. Кушлик. – Київ: НТУУ «КПІ», 2014. – 94с.
18. Копитко М. Ф., Іванків К.С. Основи програмування мовою Java: тексти лекцій. Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. 83 с.
19. Калініченко А. В. Порівняння інтегрованих середовищ розробки додатків JAVA із відкритим кодом: ECLIPSE та INTELLIJ IDE. Foss Lviv 2013.

20. IntelliJ Idea - середовище розробки для Java. [Електронний ресурс]. URL: <https://ualinux.com/uk/ubuntu-apps-programming/idea-ic> (дата звернення: 23.05.2022).
21. Spring Framework. URL: <https://spring.io/projects/spring-framework> (дата звернення: 23.05.2022).
22. Spring Boot. URL: <https://spring.io/projects/spring-boot> (дата звернення: 23.05.2022).
23. Технологічний аудит [Електронний ресурс]. URL: <https://studfile.net/preview/7336428/> (дата звернення: 25.11.22).
24. Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи [Електронний ресурс]. URL: <https://web.posibnyku.vntu.edu.ua>. (дата звернення: 25.11.22).
25. Ефективність наукових досліджень [Електронний ресурс]. URL: <https://studfile.net/preview/6872280/page:25/>. (дата звернення: 27.11.22).

Додаток А (обов'язковий)

Результат перевірки на плагіат в онлайн-системі UNICHECK



Ім'я користувача:
Озеранський В.С. КН

ID перевірки:
1013314845

Дата перевірки:
16.12.2022 12:39:15 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
16.12.2022 12:47:13 EET

ID користувача:
62038

Назва документа: 122МКР-ШевчукОР2022

Кількість сторінок: 57 Кількість слів: 10059 Кількість символів: 77248 Розмір файлу: 798.90 KB ID файлу: 1013073238

12.5% Схожість

Найбільша схожість: 7.89% з джерелом з Бібліотеки (ID файлу: 1011439675)

0.08% Джерела з Інтернету	3	Сторінка 59
12.5% Джерела з Бібліотеки	88	Сторінка 59

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

28.1% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 5%)

9.28% Вилучення з Інтернету	123	Сторінка 60
21.9% Вилученого тексту з Бібліотеки	233	Сторінка 61

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 11

Додаток Б (обов'язковий)

Лістинг програми

TestResultService

```

package com.example.filmesgradle.service;

import com.example.filmesgradle.cache.AnswerCache;
import com.example.filmesgradle.cache.AnswerWeriantCache;
import com.example.filmesgradle.dao.FilmDao;
import com.example.filmesgradle.entity.Film;
import com.example.filmesgradle.entity.TestResult;
import com.example.filmesgradle.jooq.generated.tables.records.FilmsRecord;
import com.example.filmesgradle.messagesender.MessageSender;
import lombok.RequiredArgsConstructor;
import org.modelmapper.ModelMapper;
import org.springframework.stereotype.Service;
import org.springframework.util.CollectionUtils;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;

import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;
import java.util.stream.Stream;

@Service
@RequiredArgsConstructor
public class TestResultService {

    private final AnswerCache answerCache;
    private final AnswerWeriantCache answerWeriantCache;
    private final FilmDao filmDao;
    private final ModelMapper modelMapper;
    private final MessageSender messageSender;

    private static final String NEXT_STRING = System.lineSeparator();

    public void cleanCash(Message message) {
        answerCache.remove(message.getChatId());
    }

    public void showResult(Message message) {
        List<Film> films = getFilmsByTestingResult(answerCache.getBy(message.getChatId()));
        StringBuilder sb = new StringBuilder();
        films.stream()
            .limit(3)
            .forEach(film -> {
                sb.append(film.getName());
                sb.append(" ");
                sb.append(film.getYear());
                sb.append(NEXT_STRING);
            })
    }

```



```

    );
    SendMessage sendMessage = SendMessage.builder()
        .text(sb.toString())
        .chatId(String.valueOf(message.getChatId()))
        .build();
    messageSender.sendMessage(sendMessage);
}

private List<Film> getFilmsByTestingResult(List<TestResult> testResults) {
    String ganr = getTestResult(testResults, 1);
    String ages = getTestResult(testResults, 2);
    String country = getTestResult(testResults, 3);
    String limitation = getTestResult(testResults, 5);
    int limitationInt = limitation.isEmpty() ? 0 : Integer.parseInt(limitation.substring(0, limitation.length() -
1));
    String lengths = getTestResult(testResults, 7);
    String retings = getTestResult(testResults, 10);
    List<Integer> ageList = splitToInt(ages);
    List<Integer> reting = splitToInt(retings);
    List<Integer> length = splitToInt(lengths);
    List<FilmsRecord> filmsRecords = filmDao.getAll();
    List<Film> films = filmsRecords.stream()
        .map(f -> modelMapper.map(f, Film.class))
        .collect(Collectors.toList());
    if (!ganr.isEmpty()) {
        films = films.stream()
            .filter(film -> film.getGanr().toLowerCase().contains(ganr.toLowerCase()))
            .collect(Collectors.toList());
    }
    if (!CollectionUtils.isEmpty(ageList)) {
        films = films.stream()
            .filter(film -> film.getYear() > ageList.get(0))
            .filter(film -> film.getYear() < ageList.get(1))
            .collect(Collectors.toList());
    }
    if (!country.isEmpty()) {
        films = films.stream()
            .filter(film -> film.getCountry().toLowerCase().contains(country.toLowerCase()))
            .collect(Collectors.toList());
    }
    films = films.stream()
        .filter(film -> film.getLimitation() >= limitationInt)
        .collect(Collectors.toList());
    if (!CollectionUtils.isEmpty(length)) {
        films = films.stream()
            .filter(film -> film.getLength() > length.get(0))
            .collect(Collectors.toList());
    }
    if (length.size() == 2) {
        films = films.stream()
            .filter(film -> film.getLength() < length.get(1))
            .collect(Collectors.toList());
    }
    if (!CollectionUtils.isEmpty(reting)) {
        films = films.stream()
            .filter(film -> film.getRating() > reting.get(0))
            .collect(Collectors.toList());
    }
}

```

```

    }
    if (reting.size() == 2) {
        films = films.stream()
            .filter(film -> film.getRating() < reting.get(1))
            .collect(Collectors.toList());
    }
    return films;
}

private String getTestResult(List<TestResult> testResults, int questionId) {
    List<Integer> ansId = testResults.stream()
        .filter(tr -> tr.getQuestionId()==questionId)
        .map(TestResult::getAnswerId)
        .collect(Collectors.toList());
    if (CollectionUtils.isEmpty(ansId)) {
        return "";
    }
    if (ansId.get(0).equals(13) || ansId.get(0).equals(8)) {
        return "";
    }
    return answerWeriantCache.getBy(ansId.get(0));
}

private List<Integer> splitToInt(String text) {
    if (text.isEmpty()) {
        return List.of();
    }
    if (text.contains("+")) {
        String start = text.substring(0, text.length() - 1);
        return List.of(Integer.parseInt(start));
    }
    String[] split = text.split("-");
    return List.of(Integer.parseInt(split[0]), Integer.parseInt(split[1]));
}
}
}

```

TestService

```

package com.example.filmesgradle.service;

import com.example.filmesgradle.cache.AnswerCache;
import com.example.filmesgradle.cache.AnswerWeriantCache;
import com.example.filmesgradle.cache.QuestionCache;
import com.example.filmesgradle.entity.TestResult;
import com.example.filmesgradle.messagesender.MessageSender;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.ReplyKeyboardMarkup;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardRow;

import java.util.ArrayList;
import java.util.List;

import static java.util.Objects.isNull;

```

```

@Service
@RequiredArgsConstructor
public class TestService {

    private final AnswerCache answerCache;
    private final MessageSender messageSender;
    private final QuestionCache questionCache;
    private final AnswerVeriantCache answerVeriantCache;

    public void startTest(Message message) {
        questionCache.init();
        answerCache.add(message.getChatId(), new ArrayList<>());
        answerCache.addQuestionId(message.getChatId(), 1);
        createMessageWithQuestion(message.getChatId(), 1);
    }

    public void saveAnswer(Message message) {
        String text = message.getText();
        int answerId = answerVeriantCache.getBy(text);
        answerCache.addAnswerId(message.getChatId(), answerId);
    }

    public void saveAnswerIfNeded(Message message) {
        String text = message.getText();
        List<TestResult> testResults = answerCache.getBy(message.getChatId());
        TestResult lastAnswer = testResults.get(testResults.size()-1);
        if (lastAnswer.getAnswerId() == 0) {
            int answerId = answerVeriantCache.getBy(text);
            answerCache.addAnswerId(message.getChatId(), answerId);
        }
    }

    public boolean isTestFinished(Message message) {
        List<TestResult> testResults = answerCache.getBy(message.getChatId());
        if (testResults == null){
            return false;
        }
        TestResult lastAnswer = testResults.get(testResults.size()-1);
        if (lastAnswer.getQuestionId() == 10) {
            return true;
        }
        return false;
    }

    private void createMessageWithQuestion(Long chatId, int questionId) {
        String questionText = questionCache.getBy(questionId);
        List<String> answers = questionCache.getAnswers(questionId);

        ReplyKeyboardMarkup markup = createMarkup(answers);

        SendMessage sendMessage = SendMessage.builder()
            .text(questionText)
            .chatId(String.valueOf(chatId))
            .replyMarkup(markup)
            .build();
        messageSender.sendMessage(sendMessage);
    }
}

```

```

}

private ReplyKeyboardMarkup createMarkup(List<String> answers) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    if (answers.size() == 2) {
        KeyboardRow row = createRow(answers.subList(0, 1));
        KeyboardRow row2 = createRow(answers.subList(1, 2));
        markup.setKeyboard(List.of(row, row2));
    }
    if (answers.size() == 3) {
        KeyboardRow row = createRow(answers.subList(0, 1));
        KeyboardRow row2 = createRow(answers.subList(1, 2));
        KeyboardRow row3 = createRow(answers.subList(2, 3));
        markup.setKeyboard(List.of(row, row2, row3));
    }
    if (answers.size() == 4) {
        KeyboardRow row = createRow(answers.subList(0, 2));
        KeyboardRow row2 = createRow(answers.subList(2, 4));
        markup.setKeyboard(List.of(row, row2));
    }
    if (answers.size() == 5) {
        KeyboardRow row = createRow(answers.subList(0, 2));
        KeyboardRow row2 = createRow(answers.subList(2, 4));
        KeyboardRow row3 = createRow(answers.subList(4, 5));
        markup.setKeyboard(List.of(row, row2, row3));
    }
    if (answers.size() == 6) {
        KeyboardRow row = createRow(answers.subList(0, 2));
        KeyboardRow row2 = createRow(answers.subList(2, 4));
        KeyboardRow row3 = createRow(answers.subList(4, 6));
        markup.setKeyboard(List.of(row, row2, row3));
    }
    if (answers.size() == 8) {
        KeyboardRow row = createRow(answers.subList(0, 2));
        KeyboardRow row2 = createRow(answers.subList(2, 4));
        KeyboardRow row3 = createRow(answers.subList(4, 6));
        KeyboardRow row4 = createRow(answers.subList(6, 8));
        markup.setKeyboard(List.of(row, row2, row3, row4));
    }

    markup.setResizeKeyboard(true);
    return markup;
}

private KeyboardRow createRow(List<String> answers) {
    KeyboardRow row = new KeyboardRow();
    // List<KeyboardButton> keyboardButtons = answers.stream()
    //     .map(s -> KeyboardButton.builder()
    //         .text(s)
    //         .build())
    //     .collect(Collectors.toList());
    row.addAll(answers);
    return row;
}

public void nextQuestion(Message message) {

```

```

long chatId = message.getChatId();
List<TestResult> testResults = answerCache.getBy(message.getChatId());
TestResult lastAnswer = testResults.get(testResults.size()-1);
int lastQId = lastAnswer.getQuestionId();
if (lastQId == 4) {
    if (testResults.get(testResults.size()-1).getAnswerId() == 18){
        answerCache.addQuestionId(message.getChatId(), 5);
        createMessageWithQuestion(chatId, 5);
        return;
    }else{
        answerCache.addQuestionId(message.getChatId(), 6);
        createMessageWithQuestion(chatId, 6);
        return;
    }
}
if (lastQId < 10) {
    answerCache.addQuestionId(message.getChatId(), lastQId + 1);
    createMessageWithQuestion(chatId, lastQId + 1);
    return;
}
}

public boolean isTestStarted(Message message) {
    return !isNull(answerCache.getBy(message.getChatId()));
}
}

```

AnswerCache

```

package com.example.filmesgradle.cache;

import com.example.filmesgradle.entity.TestResult;
import org.springframework.stereotype.Component;

import javax.inject.Singleton;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Component
@Singleton
public class AnswerCache {

    private Map<Long, List<TestResult>> chatIdToTestResultsTest = new HashMap<>();

    public void addQuestionId(long chatId, int questionId) {
        TestResult build = TestResult.builder().questionId(questionId).build();
        chatIdToTestResultsTest.get(chatId).add(build);
    }

    public void addAnswerId(long chatId, int answerId) {
        TestResult last = chatIdToTestResultsTest.get(chatId).get(chatIdToTestResultsTest.get(chatId).size()-1);
        last.setAnswerId(answerId);
    }

    public TestResult getBy(long chatId, int questionId){
        return chatIdToTestResultsTest.get(chatId).stream()

```

```

        .filter(tr -> tr.getQuestionId() == questionId)
        .findFirst()
        .get();
    }

    public List<TestResult> getBy(long chatId){
        return chatIdToTestResultsTest.get(chatId);
    }

    public void remove(long chatId){
        chatIdToTestResultsTest.remove(chatId);
    }

    public void add(Long chatId, List<TestResult> list) {
        chatIdToTestResultsTest.put(chatId, list);
    }
}

AnswerWeriantCache

```

```

package com.example.filmesgradle.cache;

import org.springframework.stereotype.Component;

import javax.inject.Singleton;
import java.util.HashMap;
import java.util.Map;

@Component
@Singleton
public class AnswerWeriantCache {

    private Map<Integer, String> answerMap = new HashMap<>();

    public void init(){
        if (!answerMap.isEmpty()){
            return;
        }
        answerMap.put(1, "Драма");
        answerMap.put(2, "Фантастика");
        answerMap.put(3, "Бойовик");
        answerMap.put(4, "Пригоди");
        answerMap.put(5, "Комедія");
        answerMap.put(6, "Триллер");
        answerMap.put(7, "Жахи");
        answerMap.put(8, "Інше");
        answerMap.put(9, "1990-2000");
        answerMap.put(10, "2000-2010");
        answerMap.put(11, "2010-2020");
        answerMap.put(12, "2020-2022");
        answerMap.put(13, "Не важливо");
        answerMap.put(14, "США");
        answerMap.put(15, "Японія");
        answerMap.put(16, "Франція");
        answerMap.put(17, "Великобританія");
        answerMap.put(18, "Так");
        answerMap.put(19, "Ні");
        answerMap.put(20, "12+");
    }
}

```

```

    answerMap.put(21, "14+");
    answerMap.put(22, "16+");
    answerMap.put(23, "18+");
    answerMap.put(24, "60-90");
    answerMap.put(25, "90-120");
    answerMap.put(26, "120-150");
    answerMap.put(27, "150+");
    answerMap.put(28, "Стівен Спілберг");
    answerMap.put(29, "Мартін Скорсезе");
    answerMap.put(30, "Альфред Хічкок");
    answerMap.put(31, "Джеймс Кемірон");
    answerMap.put(32, "Спайк Лі");
    answerMap.put(33, "Українська");
    answerMap.put(34, "Англійська");
    answerMap.put(35, "Французька");
    answerMap.put(36, "7-8");
    answerMap.put(37, "8-9");
    answerMap.put(38, "9+");
}

public String getBy(int id){
    return answerMap.get(id);
}

public int getBy(String test){
    return answerMap.keySet().stream()
        .filter(k -> answerMap.get(k).equalsIgnoreCase(test))
        .findFirst().get();
}
}

```

QuestionCache

```

package com.example.filmesgradle.cache;

import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;

import javax.inject.Singleton;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Component
@Singleton
@RequiredArgsConstructor
public class QuestionCache {

    private final AnswerWeriantCache answerWeriantCache;

    private Map<Integer, String> questionMap = new HashMap<>();

    public void init(){
        if (!questionMap.isEmpty()){
            return;
        }
        answerWeriantCache.init();
        questionMap.put(1, "Вкажіть бажаний жанр");
    }
}

```

```

questionMap.put(2, "Вкажіть рік виходу");
questionMap.put(3, "Вкажіть країну створення");
questionMap.put(4, "Важливе вікове обмеження?");
questionMap.put(5, "Вкажіть обмеження");
questionMap.put(6, "Ведіть улюбленого актора");
questionMap.put(7, "Вкажіть тривалість");
questionMap.put(8, "Оберіть режисера");
questionMap.put(9, "Якій звуковій доріжці надаєте перевагу?");
questionMap.put(10, "Оберіть рейтинг IMDb");
}

public String getBy(int id){
    return questionMap.get(id);
}

public List<String> getAnswers(int qId){
    if (qId == 1){
        return List.of(answerWeriantCache.getBy(1), answerWeriantCache.getBy(2),
answerWeriantCache.getBy(3), answerWeriantCache.getBy(4),
answerWeriantCache.getBy(5), answerWeriantCache.getBy(6), answerWeriantCache.getBy(7),
answerWeriantCache.getBy(8));
    }
    if (qId == 2){
        return List.of(answerWeriantCache.getBy(9), answerWeriantCache.getBy(10),
answerWeriantCache.getBy(11),
answerWeriantCache.getBy(12), answerWeriantCache.getBy(13));
    }
    if (qId == 3){
        return List.of(answerWeriantCache.getBy(14), answerWeriantCache.getBy(15),
answerWeriantCache.getBy(16),
answerWeriantCache.getBy(17), answerWeriantCache.getBy(13));
    }
    if (qId == 4){
        return List.of(answerWeriantCache.getBy(18), answerWeriantCache.getBy(19));
    }
    if (qId == 5){
        return List.of(answerWeriantCache.getBy(20), answerWeriantCache.getBy(21),
answerWeriantCache.getBy(22), answerWeriantCache.getBy(23));
    }
    if (qId == 7){
        return List.of(answerWeriantCache.getBy(24), answerWeriantCache.getBy(25),
answerWeriantCache.getBy(26),
answerWeriantCache.getBy(27), answerWeriantCache.getBy(13));
    }
    // TODO
    if (qId == 8){
        return List.of(answerWeriantCache.getBy(28), answerWeriantCache.getBy(29),
answerWeriantCache.getBy(30), answerWeriantCache.getBy(31),
answerWeriantCache.getBy(32), answerWeriantCache.getBy(13));
    }
    if (qId == 9){
        return List.of(answerWeriantCache.getBy(33), answerWeriantCache.getBy(34),
answerWeriantCache.getBy(35), answerWeriantCache.getBy(8));
    }
    if (qId == 10){
        return List.of(answerWeriantCache.getBy(36), answerWeriantCache.getBy(37),
answerWeriantCache.getBy(38), answerWeriantCache.getBy(13));
    }
}

```



```

    }
    return List.of();
}
}

```

MessageHandler

```

package com.example.filmesgradle.bot;

import com.example.filmesgradle.service.TestResultService;
import com.example.filmesgradle.service.TestService;
import lombok.RequiredArgsConstructor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.meta.api.objects.Message;

@Component
public class MessageHandler {

    private TestService testService;

    private TestResultService testResultService;

    @Autowired
    public void setTestService(TestService testService) {
        this.testService = testService;
    }

    @Autowired
    public void setTestResultService(TestResultService testResultService) {
        this.testResultService = testResultService;
    }

    public void apply(Message message) {
        if (testService.isTestFinished(message)) {
            testService.saveAnswerIfNeded(message);
            testResultService.showResult(message);
            testResultService.cleanCash(message);
            return;
        }
        if (testService.isTestStarted(message)) {
            testService.saveAnswer(message);
            testService.nextQuestion(message);
            return;
        }
        testService.startTest(message);
        return;
    }
}

```

TelegramBot

```

package com.example.filmesgradle.bot;

import com.example.filmesgradle.processor.UpdateProcessor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.bots.TelegramLongPollingBot;
import org.telegram.telegrambots.meta.api.objects.Update;

```

```
@Component
public class TelegramBot extends TelegramLongPollingBot {

    @Value("${telegram.bot.username}")
    private String username;
    @Value("${telegram.bot.token}")
    private String token;

    private UpdateProcessor updateProcessor;

    @Autowired
    public void setUpdateProcessor(UpdateProcessor updateProcessor) {
        this.updateProcessor = updateProcessor;
    }

    @Override
    public String getBotUsername() {
        return username;
    }

    @Override
    public String getBotToken() {
        return token;
    }

    @Override
    public void onUpdateReceived(Update update) {
        updateProcessor.process(update);
    }
}
```

Додаток В (обов'язковий)**ІЛЮСТРАТИВНА ЧАСТИНА****«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДБОРУ ФІЛЬМІВ»**

Виконала: студентка 2-го курсу, групи
КН-21м

Спеціальності 122 – «Комп'ютерні
науки»

Шевчук О. Р.

(прізвище та ініціали)

Керівник: д.т.н., ст. викл. каф. КН

Озеранський О. В.

(прізвище та ініціали)

«_____» _____ 2022 р.

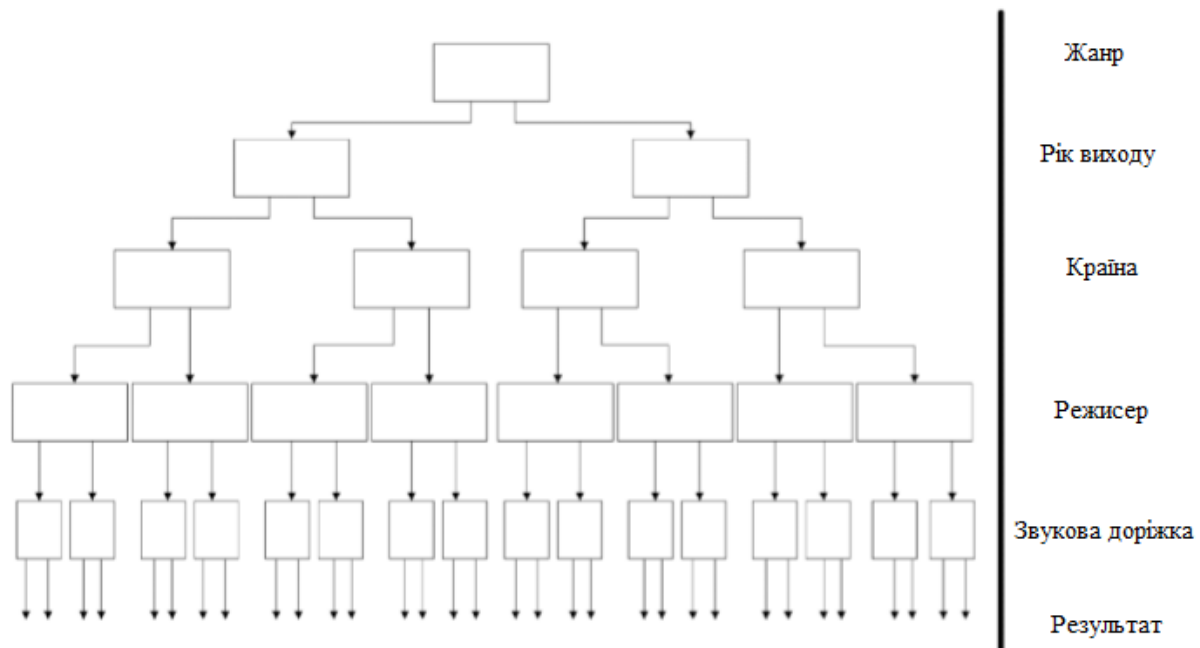


Рисунок В.1 – Структурна схема моделі дерева рішень для вибору
фільмів

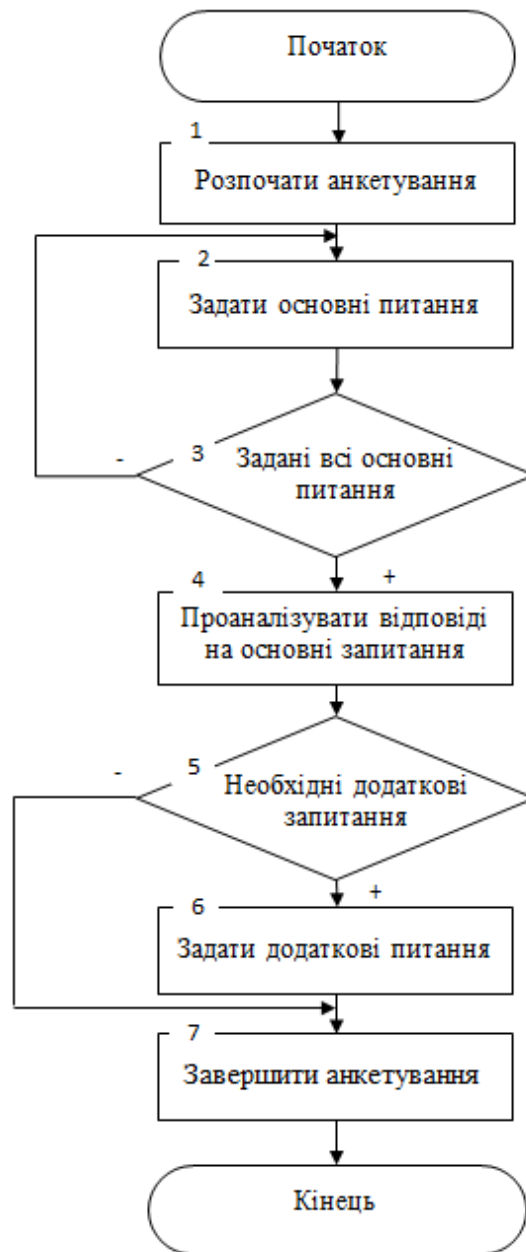


Рисунок В.2 – Структурна схема загального алгоритму функціонування інформаційної технології надання рекомендацій щодо підбору фільмів

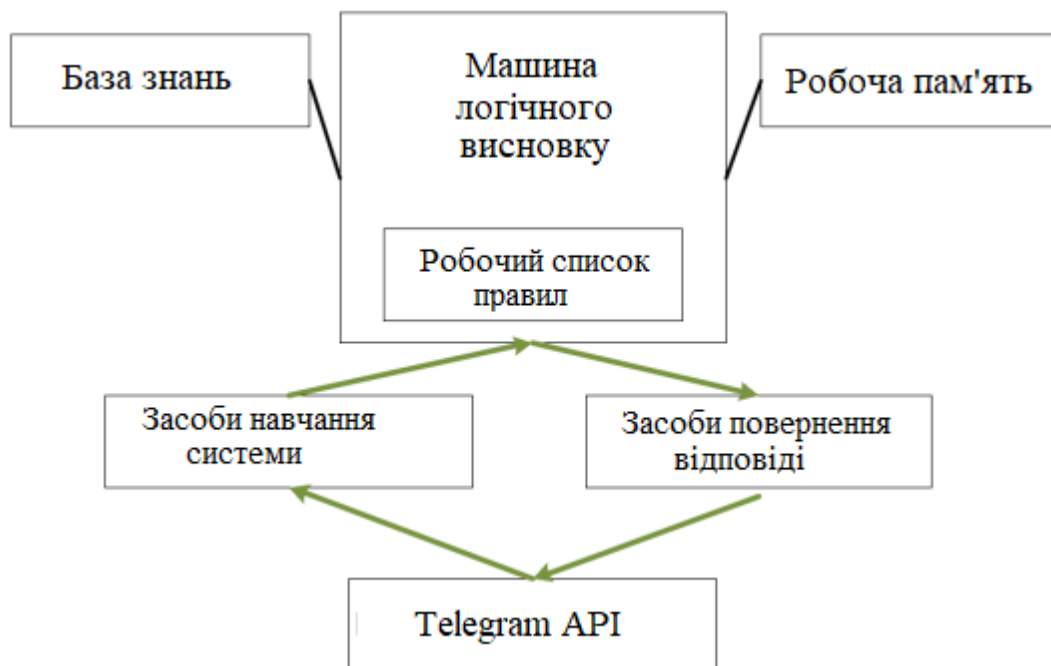


Рисунок В.3 – Структура інформаційної технології надання рекомендацій щодо підбору фільмів

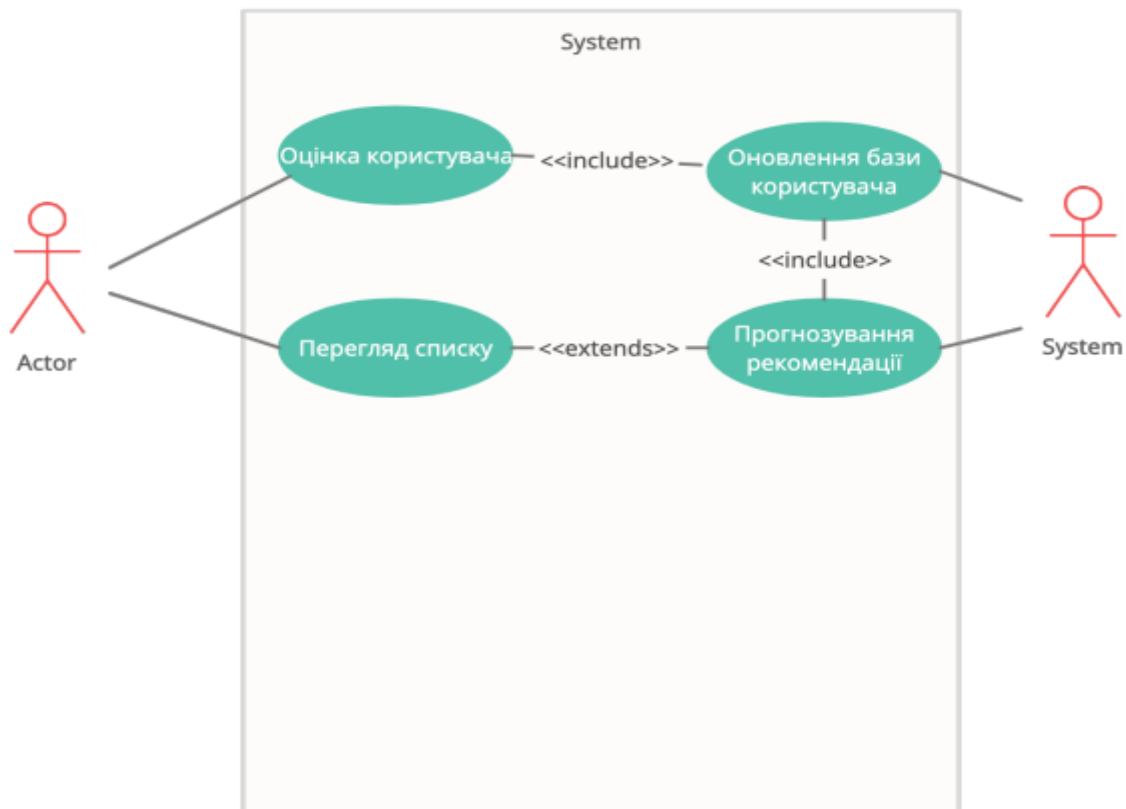


Рисунок В.4 – Діаграма прецедентів інформаційної технології надання рекомендацій щодо підбору фільмів

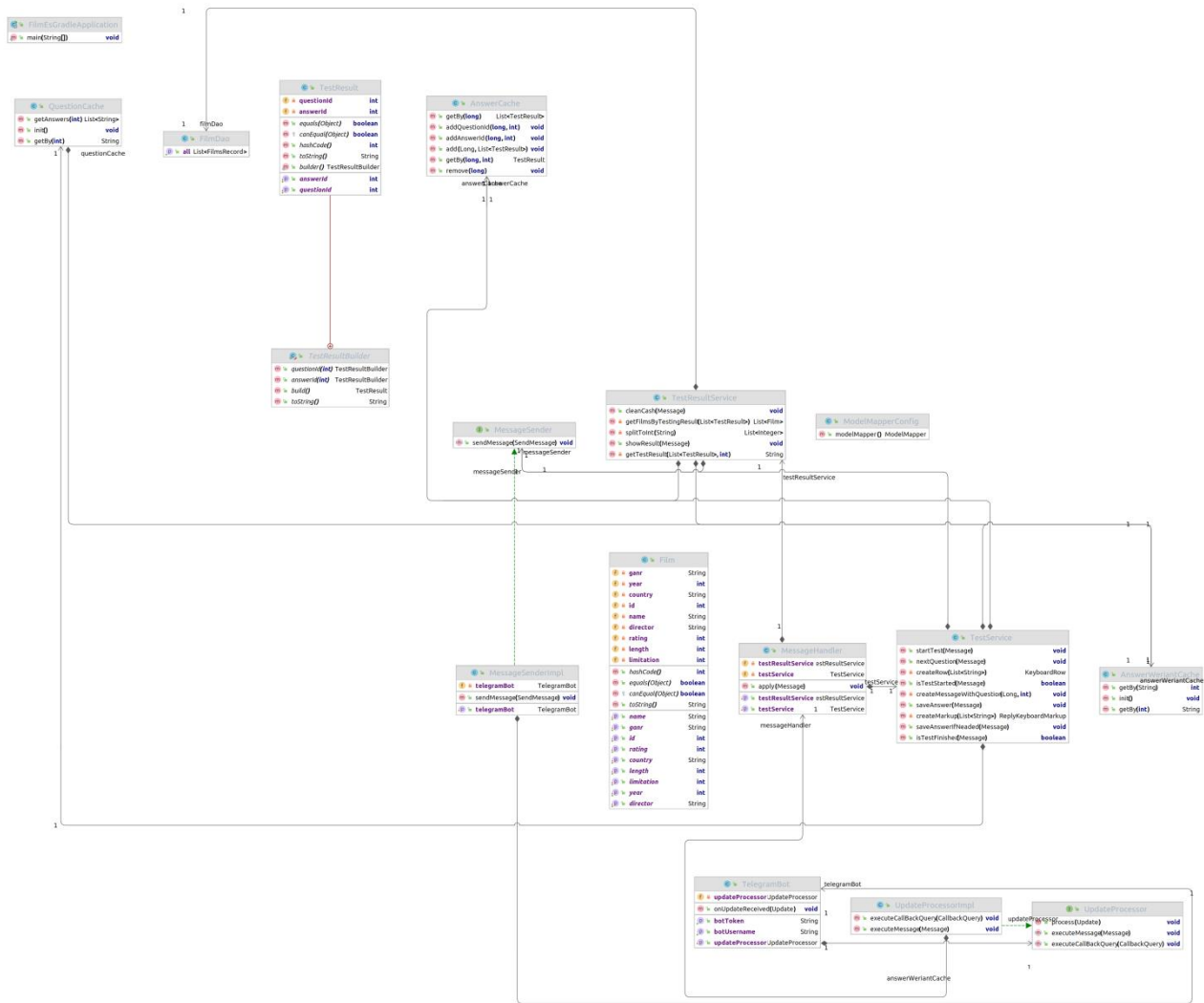


Рисунок В.5 – UML-діаграма класів

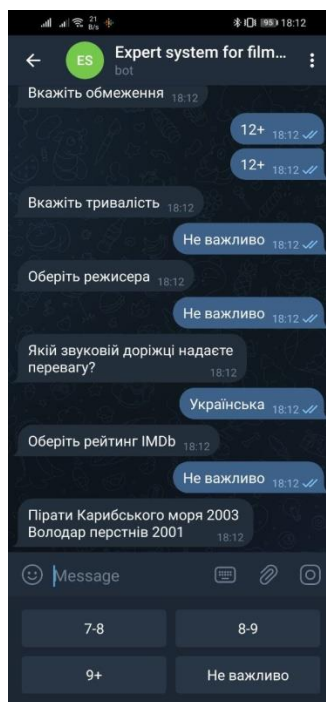


Рисунок В.6 – Загальний вигляд Приклад роботи програми



Рисунок В.7 – Структурна схема алгоритму формування запитань

Додаток Г (довідниковий) Інструкція користувача

Для запуску інформаційної технології підбору фільмів необхідно відкрити Telegram бот і ввести «/start», що запускає бота (рисунок Г.1).

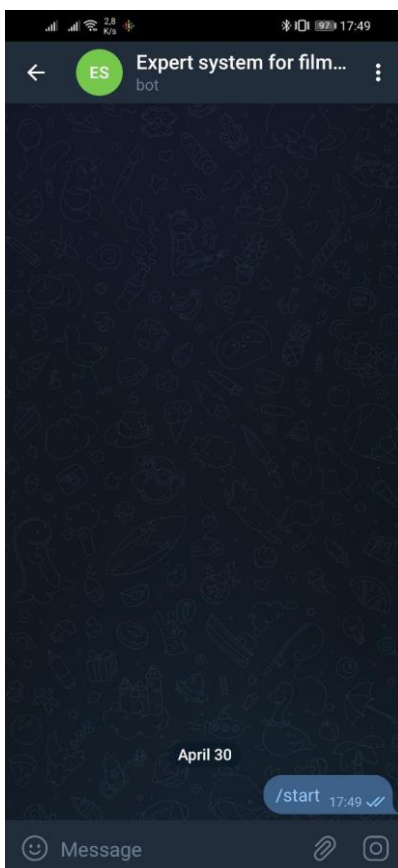


Рисунок Г.1 – Загальний вигляд інтерфесного вікна «Початкок діалогу з ботом»

Після відкриття першого питання програма запропонує варіанти відповідей, відповідно до яких будуть поставлені наступні запитання (рисунок Г.2, Г.3)

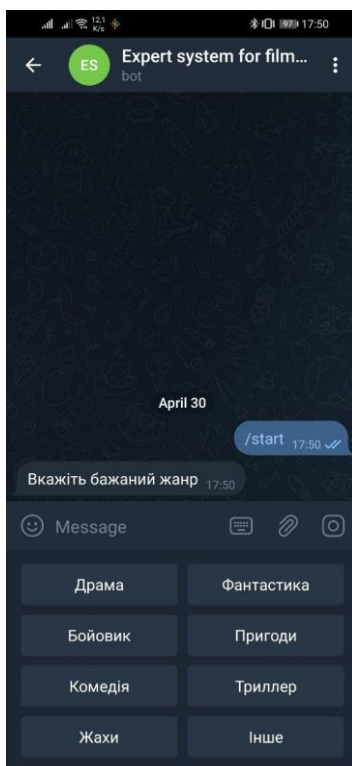


Рисунок Г.2 – Загальний вигляд інтерфесного вікна «Вкладка з першим питанням до користувача»

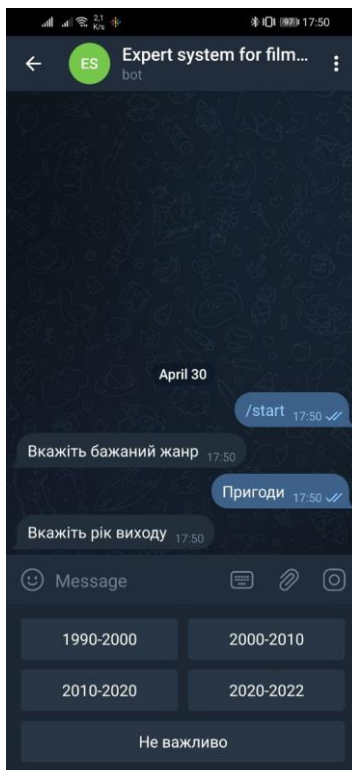


Рисунок Г.3 – Загальний вигляд інтерфесного вікна «Вкладка з наступним питанням щодо даної ситуації»

Після того, як користувач відповість на всі запитання, експертна система надання рекомендацій для вибору фільму запропонує рекомендації (Рисунок Г.4).

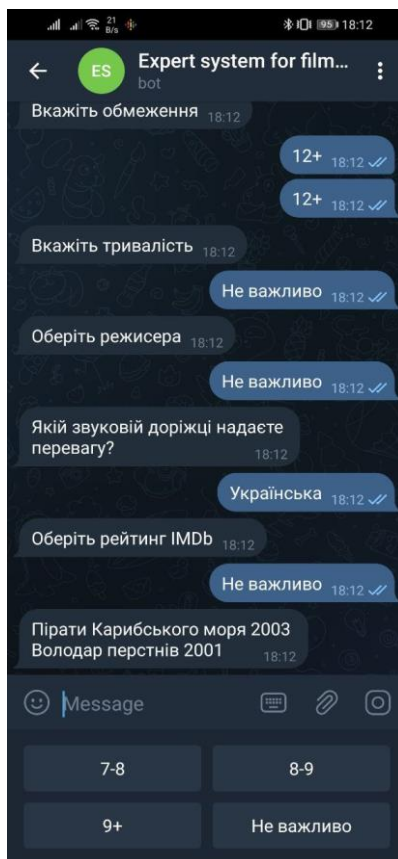


Рисунок Г.4 – Загальний вигляд інтерфесного вікна «Результат роботи інформаційної технології підбору фільму»

Додаток Д
Акт впровадження



МАЛЕ НАУКОВО-ВИРОБНИЧЕ ПІДПРИЄМСТВО "ТОВ "ІТІ"
Україна, 21021, м.Вінниця, вул. Келецька, 56

№ 24 від 02 12 2022.

ДОВІДКА

Дана Шевчук Олені Русланівні в тому, що результати, одержані нею в процесі виконання магістерської кваліфікаційної роботи, а саме алгоритми та програмні засоби надання рекомендацій щодо підбору фільмів, планується використати в розробках ТОВ «ІТІ».

Зам. директора ТОВ «ІТІ» Бодяк В.М.

