

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему: «Розроблення рекомендаційної системи кінотеатру на основі графової
бази даних»

здобувача Бицької Вікторії Віталіївни
(ПІБ)


(Підпис)

Науковий керівник:

к.е.н., доцент



Ситник Н.В.

**Робота допущена до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)
освітньо-професійної програми

_____ Помазун О.М.

“ _____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Тішков Б.О.

“ _____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти **Бицької Вікторії Віталіївни**

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи
на тему: «Розроблення рекомендаційної системи кінотеатру на основі графової бази даних»

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р.
№ 466- ст.

Кваліфікаційна бакалаврська робота виконується на матеріалах

наукових публікацій та матеріалів з мережі інтернет

План кваліфікаційної бакалаврської роботи

Розділ I Характеристика та аналіз предметної області

Розділ II Розробка вимог і моделювання інформаційної системи.

Розділ III Проектування та реалізація компонентів системи.

Об'єкт дослідження процес надання рекомендацій для формування актуального переліку фільмів для кінопрокату, з врахуванням критеріїв вибору, визначених користувачем системи.

Предмет дослідження методи, моделі, алгоритми та процеси розробки рекомендаційної системи для кінотеатру на основі графової бази даних.

Мета кваліфікаційної бакалаврської роботи реалізація рекомендаційної системи для кінотеатру на основі графової бази даних для формування репертуару кінотеатру та збільшення відвідуваності.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I навести характеристику предметної галузі й об'єкта дослідження. Проаналізувати сучасний стан розвитку прокату фільмів з метою популяризації та оптимізації їх прокату. Проаналізувати літературні джерела, що висвітлюють практичний досвід та підходи до розробки рекомендаційних систем.

У розділі II Визначити основні вимоги до рекомендаційної системи та обґрунтувати методи її проєктування. При моделюванні рекомендаційної системи використати сучасні системи автоматизованого інжинірингу. Розробити постановку, алгоритм та інформаційну модель досліджуваної рекомендаційної системи з урахуванням використання графової бази даних Neo4j.

У розділі III виконати проєктування компонентів рекомендаційної системи, зокрема інформаційного, технічного та програмного забезпечення. Змоделювати графову базу даних та реалізувати рекомендаційну систему вибору фільмів для прокату в кінотеатрі, вказавши можливість та доцільність її практичного застосування.

Завдання підготував

науковий керівник



Ситник Ніна Василівна

« 10 » березня 2025 р.

Завдання одержав

здобувач



Бицька Вікторія Віталіївна

« 10 » березня 2025 р..

Відгук
про кваліфікаційну бакалаврську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»
Бицької Вікторії Віталіївни
на тему

«Розроблення рекомендаційної системи кінотеатру на основі графової бази даних»

1. **Актуальність теми:** розробка рекомендаційних систем є актуальною проблематикою сьогодення, так як такі системи є важливими для підвищення ефективності прийняття рішень та прискорення зростання бізнесу в нашому дедалі конкурентнішому онлайн-середовищі.
2. **Позитивні риси кваліфікаційної бакалаврської роботи:** в рамках завдання на кваліфікаційну бакалаврську роботу були проаналізовані існуючі підходи до розробки рекомендаційних систем та обґрунтовано переваги використання графових баз даних для їх розробки. В роботі детально досліджені всі процеси вибору фільмів для прокату в кінотеатрі та побудовані відповідні діаграми з використанням засобів сучасного інжинірингу. Досліджені й структуровані основні бізнес-вимоги до предметної області та змодельована система з орієнтацією на використання графової СКБД Neo4j.
3. **Наявність самостійних розробок автора:** в роботі вибрані графові алгоритми, які дозволили створити ефективну рекомендаційну систему, що враховує критерії менеджерів з прокату фільмів та складні логічні і семантичні взаємозв'язки між даними, яку характеризують досліджувану предметну область.
4. **Цінність теоретичних висновків та практичних рекомендацій:** запропонована рекомендаційна система відбору фільмів для їх прокату в кінотеатрі є безперечно цікавою і практично орієнтованою розробкою, так як розроблена з використанням новітніх інструментальних засобів.
5. **Наявність недоліків:** по представленій роботі можуть бути лише рекомендації щодо використання більшого спектра графових алгоритмів та їх порівняльного аналізу з точки зору ефективності отриманих рекомендацій.
6. **Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК:** У цілому, представлена кваліфікаційна бакалаврська робота відповідає встановленим вимогам методичних вказівок щодо структури, обсягу й змісту та рекомендується до захисту з **високою оцінкою**, а **Бицькій Вікторії Віталіївній** може бути присвоєний освітньо-кваліфікаційний рівень «бакалавр» галузі знань 12 «Інформаційні технології» за спеціальністю 122 «Комп'ютерні науки».

Науковий керівник: к.е.н., професор

кафедри ІСЕ
02.05 2025 р.



Ситник Н.В.

Рецензія
на кваліфікаційну бакалаврську роботу
здобувача вищої освіти
Бицької Вікторії Віталіївни
(прізвище, ім'я, по батькові)

Тема «Розроблення рекомендаційної системи кінотеатру на основі графової бази даних»

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення. Представлена робота має виняткову актуальність у контексті сучасних викликів кінотеатральної індустрії, для якої характерне швидке збільшення кількості кінопродукції, зростання конкуренції з боку інших кінотеатрів, а також різноманіття вподобань цільової аудиторії. Робота є особливо важливою для кінотеатрів, де питання оптимізації розкладу показів для утримання та залучення аудиторії є особливо актуальним.

Якість проведеного дослідження. Авторка роботи продемонструвала глибоке розуміння специфіки роботи кінотеатрів та технічних аспектів створення рекомендаційних систем. Дослідження базується на ґрунтовній теоретичній основі із залученням сучасних наукових джерел. Практична реалізація характеризується системним підходом до розробки як окремих алгоритмічних компонентів, так і архітектури системи в цілому. Поєднання глибокого теоретичного обґрунтування з практичною спрямованістю забезпечило високу якість та результативність проведеного дослідження.

Позитивні риси кваліфікаційної бакалаврської роботи. Серед позитивних рис варто виділити практичну спрямованість проєкту, де розроблена система враховує реальні бізнес-процеси кінотеатру та характеризується ефективністю у здійсненні планування прокату. Також варто відзначити рівень виконання технічної частини проєкту. Створена система вирізняється інтуїтивно зрозумілим користувацьким інтерфейсом, високою швидкістю та здатністю генерувати дійсно релевантні рекомендації.

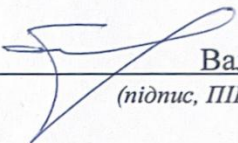
Зауваження. Перспективним напрямом подальших досліджень є проведення комплексного тестування системи в реальних умовах функціонування кінотеатру з детальним аналізом динаміки ключових економічних показників до та після впровадження рекомендаційної системи.

Практична значимість висновків і рекомендацій. Висновки за результатами дослідження мають особливу значущість, адже демонструють високий рівень реалізації розробленої системи. Загалом впровадження рекомендаційної системи на основі графової бази даних забезпечує кінотеатру технологічну перевагу над конкурентами, а також мінімізує ризик додавання неактуальних стрічок до прокату. Робота відповідає кваліфікаційним вимогам освітньо-професійної програми за спеціальністю 122 «Комп'ютерні науки», має практичне значення та заслуговує на оцінку «відмінно».

Кандидат економічних наук,
доцент кафедри інформаційних
технологій, фізико-математичних
та економічних наук Ніжинського
державного університету
імені Миколи Гоголя



Ніжинський державний університет імені Миколи Гоголя	
Підпис <i>Фетісова В.</i>	засвідчую
Провідний фахівець <i>Михайленко Ж.</i>	


Валерій ФЕТІСОВ
(підпис, ПІБ)

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Розроблення рекомендаційної системи кінотеатру на основі
графової бази даних»
Київ: кафедра інформаційних систем в економіці, 2025 р.

Бакалаврська кваліфікаційна робота присвячена проблемі персоналізації кінематографічних рекомендацій, яку пропонується розв'язати шляхом створення рекомендаційної системи для кінотеатру з використанням графових баз даних та графових алгоритмів.

Бакалаврська кваліфікаційна робота складається з трьох розділів, логічно пов'язаних між собою.

У першому розділі представлено характеристику предметної області та об'єкта дослідження, проведено аналіз завдань для розв'язання, а також здійснено огляд існуючих технологічних рішень у галузі проєктування та реалізації рекомендаційних систем.

Другий розділ має проєктний характер та зосереджується на теоретичному обґрунтуванні методології проєктування системи, формулюванні задачі, формуванні її архітектурної концепції, розробці алгоритмічних підходів для генерації рекомендацій в середовищі графової бази даних Neo4j.

Третій розділ носить практичний характер та містить детальний опис компонентів інформаційного, технічного та програмного забезпечення рекомендаційної системи кінотеатру, побудованої на графовій базі даних. Визначено архітектуру інформаційних структур та графових моделей, що застосовуються для реалізації функціоналу персоналізованих кінематографічних рекомендацій.

У висновках сформульовано обґрунтування доцільності створення та практичного впровадження рекомендаційної системи кінотеатру з використанням графових технологій.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 102 сторінки, 13 таблиць, 47 рисунків, список використаних джерел з 20 найменувань, 5 додатків.

«РОЗРОБЛЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ КІНОТЕАТРУ НА ОСНОВІ ГРАФОВОЇ БАЗИ ДАНИХ»

Об'єкт дослідження: процес надання рекомендацій для формування актуального переліку фільмів для кінопрокату, з врахуванням критеріїв вибору, визначених користувачем системи.

Предмет дослідження: методи, моделі, алгоритми та процеси розробки рекомендаційної системи для кінотеатру на основі графової бази даних.

Мета дослідження: реалізація ефективної рекомендаційної системи для кінотеатру на основі графової бази даних, що дозволить підвищити точність та актуальність наданих рекомендацій фільмів, оптимізувати формування репертуару кінотеатру та збільшити відвідуваність кінотеатру.

Відповідно до поставленої мети визначені такі *завдання*:

1. Аналіз предметної області та вже наявного теоретично-практичного досвіду використання існуючих рекомендаційних систем для кінотеатрів.
2. Проєктування моделі даних, що включає визначення ключових сутностей та зв'язків між ними, планування структури властивостей для вузлів та ребер графу та розробку графової моделі для представлення всіх визначених об'єктів та зв'язків.
3. Вибір графової СУБД, що включає порівняльний аналіз різних графових СУБД та вибір оптимальної, враховуючи визначенні функціональні та технічні умови до системи.
4. Розробка ефективних алгоритмів надання рекомендацій на основі базових графових алгоритмів, розробка методів аналізу вподобань глядачів та виявлення прихованих зв'язків для надання найбільш релевантних рекомендацій.
5. Розробка програмного інтерфейсу серверної логіки для роботи з графовою БД, створення сучасного та зручного інтерфейсу користувача для управління рекомендаційною системою.

6. Здійснення процесів тестування, шляхом відтворення різноманітних сценаріїв роботи системи для оцінки точності рекомендацій та оптимізації продуктивності системи.

Теоретична значущість проєкту визначення створенням теоретичного підґрунтя для використання графових моделей у сфері рекомендаційних систем, розробки рекомендаційних алгоритмів з урахуванням специфіки графового представлення даних в умовах динамічної зміни користувацьких вподобань та постійного зростання кількості нового контенту.

Практична значущість визначена можливістю оптимізації процесу вибору фільмів для прокату у кінотеатрах, який буде здійснено з врахування наявних аналітичних даних. Оптимізація вибору кінострічок має на меті залучення більшої кількості відвідувачів, що в свою чергу збільшить прибутковість для кінотеатрів.

Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: кінотеатр, рекомендація, алгоритм, граф, користувач, діаграма, проєктування, система, база даних.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	4
ВСТУП.....	5
РОЗДІЛ 1	8
ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	8
1.1 Характеристика предметної галузі та об'єкта дослідження.....	8
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	13
Висновки до розділу 1	16
РОЗДІЛ 2	18
РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	18
2.1 Аналіз і специфікація вимог до інформаційної системи	18
2.2 Постановка та алгоритм розв'язання задачі.....	25
2.2.1 Постановка задачі	25
2.2.2 Алгоритм розв'язання задачі	35
2.3 Моделювання інформаційної системи.....	45
2.3.1 Моделювання поведінки системи	45
2.3.2 Моделювання структури системи	53
2.3.3 Розподіл вимог за компонентами системи	60
Висновки до розділу 2	66
РОЗДІЛ 3	67
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	67
3.1 Інформаційне забезпечення	67
3.1.1 Загальна характеристика інформаційного забезпечення.....	67
3.1.2 Організація збору і передання первинної інформації.....	67
3.1.3 Структура інформаційних масивів.....	68
3.1.4 Вибір СКБД	69
3.1.5 Інфологічна модель бази (сховища) даних.....	70
3.1.6 Даталогічна модель бази (сховища) даних.....	72
3.2 Технічне забезпечення.....	74
3.2.1 Загальні положення та схема автоматизації.....	74
3.2.2 Структура комплексу технічних засобів	75

3.2.3 Опис автоматизованого робочого місця.....	78
3.3 Програмне забезпечення	78
3.3.1 Структура програмного забезпечення	78
3.3.2 Системне програмне забезпечення	82
3.3.3 Прикладне програмне забезпечення	84
3.3.4 Програмна документація.....	84
3.4 Результати реалізації інформаційної системи.....	85
Висновки до розділу 3	96
ВИСНОВКИ.....	98
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	100
ДОДАТКИ.....	103

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

UML – Unified Modeling Language, уніфікована мова моделювання

АРМ – автоматизоване робоче місце

БД – база даних

ІЗ – інформаційне забезпечення

ІС – інформаційна система

ОС – операційна система

СУБД – система управління базою даних

ВСТУП

Впровадження технологій рекомендаційних систем стрімко поширюється в різноманітних галузях, зокрема набуваючи особливої цінності в індустрії кінопрокату. Кількість фільмів, які випускаються різноманітними кіностудіями щороку, невпинно зростає, у зв'язку з чим і з'являється необхідність автоматизації цього процесу. Актуальність даної теми визначена тим, що на сьогодні більшість кінотеатрів все ще покладаються на класичну модель відбору фільмів для прокату – через складний ланцюжок взаємодії між дистриб'юторами, менеджерами репертуарної політики та керівництвом мережі кінотеатрів. Застарілість такого підходу полягає в тому, що перед затвердженням фільму для прокату необхідно здійснити аналіз кожного фільму та порівняти його характеристики з тими фільмами, яким надає перевагу цільова аудиторія кінотеатру.

Предметом дослідження є методи, моделі, алгоритми та процеси розробки рекомендаційної системи для кінотеатру на основі графової бази даних. Система має надавати можливості для аналізу прокату фільмів за визначений часовий період та обирати фільми, які дали найбільший попит, а відповідно до цього і найбільший касовий збір. Крім цього, необхідно реалізувати функціонал для визначення критеріїв вибору, такі як жанри, режисери, актори тощо. Найбільш оптимальним рішенням для цього є саме графові бази даних, які враховуючи свою гнучку та адаптивну структуру, а також можливість чіткого відображення всіх взаємозв'язків між об'єктами надають численні переваги у порівнянні з традиційними реляційними БД.

Об'єктом дослідження є процес надання рекомендацій для формування актуального переліку фільмів для кінопрокату, з врахуванням критеріїв вибору, визначених користувачем системи.

Метою дослідження є реалізація ефективної рекомендаційної системи для кінотеатру на основі графової бази даних, що дозволить підвищити точність та актуальність наданих рекомендацій фільмів, оптимізувати формування репертуару кінотеатру та збільшити відвідуваність кінотеатру.

Для досягнення кінцевої мети дипломного проєкту визначено чіткі завдання, серед яких:

1. Аналіз предметної області та вже наявного теоретично-практичного досвіду використання існуючих рекомендаційних систем для кінотеатрів.
2. Проєктування моделі даних, що включає визначення ключових сутностей та зв'язків між ними, планування структури властивостей для вузлів та ребер графу та розробку графової моделі для представлення всіх визначених об'єктів та зв'язків
3. Вибір графової СУБД, що включає порівняльний аналіз різних графових СУБД та вибір оптимальної, враховуючи визначенні функціональні та технічні умови до системи.
4. Розробка ефективних алгоритмів надання рекомендацій на основі базових графових алгоритмів, розробка методів аналізу вподобань глядачів та виявлення прихованих зв'язків для надання найбільш релевантних рекомендацій.
5. Розробка програмного інтерфейсу серверної логіки для роботи з графовою БД, створення сучасного та зручного інтерфейсу користувача для управління рекомендаційною системою.
6. Здійснення процесів тестування, шляхом відтворення різноманітних сценаріїв роботи системи для оцінки точності рекомендацій та оптимізації продуктивності системи.

Теоретична значущість проєкту визначення створенням теоретичного підґрунтя для використання графових моделей у сфері рекомендаційних систем, розробки рекомендаційних алгоритмів з урахуванням специфіки графового представлення даних в умовах динамічної зміни користувацьких вподобань та постійного зростання кількості нового контенту.

Практична значущість визначена можливістю оптимізації процесу вибору фільмів для прокату у кінотеатрах, який буде здійснено з врахування наявних аналітичних даних. Оптимізація вибору кінострічок має на меті залучення більшої кількості відвідувачів, що в свою чергу збільшить прибутковість для кінотеатрів.

Загальний обсяг роботи становить 102 сторінки друкованого тексту, 47 рисунків, 13 таблиць, 5 додатків. Список використаних джерел налічує 20 найменування.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Система надання рекомендацій – програмна система, що здійснює комплексний аналіз даних каталогу об'єктів з метою формування персоналізованих рекомендацій [5]. Рекомендаційна система для кінотеатру є централізованою платформою, яка охоплює все, що пов'язано з вибором кінопродукції та здійснює формування персоналізованих рекомендацій. Об'єктами рекомендаційної системи кінотеатру є фільми та користувачі системи.

Ключова задача системи для кінотеатру – забезпечення релевантного та ефективного підбору кінопродукції для кінопрокату, особливо в умовах великого асортименту фільмів та обмеженого часового ресурсу.

Рекомендаційні системи стають важливим інструментом для модернізації кінотеатрів шляхом інтеграції сучасних технологій у рутинний процес планування прокату фільмів. Їх особливістю стає комбінація як загальних методів надання рекомендацій, які диктуються новини трендами у галузі кіно так і індивідуальними потребами цільової аудиторії. Таким чином, з одного боку система враховує актуальні тренди кіноіндустрії, спираючись на об'єктивні показники популярності та актуальності кінострічки, наприклад рейтинги IMDb, новизна кінострічки тощо. З іншого боку, система надає функціональні можливості для отримання персоналізованих рекомендацій, які обумовлені вподобаннями цільової аудиторії, що є характерно для конкретного кінотеатру. Комбінація цих двох підходів дозволяє поєднати два ключові фактори: відображення сучасних глобальних тенденцій кіноринку одночасно з врахування культурних, демографічних та соціальних особливостей місцевих глядачів.

Розробка рекомендаційної системи кінотеатру спрямована на створення сучасної інформаційної системи, що містить актуальні дані про кінострічки,

підтримує ефективні алгоритми надання рекомендацій та надає зручний користувацький інтерфейс для користувачів системи.

Економічний зміст представленої задачі обумовленими оптимізацією фінансових та часових ресурсів. Розробка системи рекомендацій для цієї предметної області робить можливим збільшення прибутку кінотеатру та підвищення рівня наповненості кінотеатрів, залучення нових клієнтів та утримання існуючих, що обумовлено орієнтацією на потреби цільової аудиторії. Оптимізація часових ресурсів обумовлена наданими функціональними можливостями для автоматизації процесу пошуку фільмів за характеристиками, що в свою чергу оптимізує час працівників кінотеатру. Без використання подібних систем, окрім попереднього аналізу показників фільмів, які користувались попитом у конкретному кінотеатрі, працівник має додаткового здійснити багатоетапний пошук подібного контенту за набором різноманітних факторів. У цей час розроблена рекомендаційна система дозволяє автоматизувати цей пошук, залишивши працівнику більше часу на безпосереднє дослідження цільової аудиторії і подальше покращення маркетингових стратегій.

Рекомендаційна система кінотеатру виявляє складні взаємозв'язки рішень, де вибір фільмів для показу пов'язаний з рівнем подібності фільмів на основі спільних характеристик, впливаючи на майбутні рішення про закупівлю контенту та маркетингову стратегію. Завдяки такому підходу кінотеатр може ефективно прогнозувати попит, формуючи цілісну програму показів, яка відповідає інтересам аудиторії в довгостроковій перспективі.

Для прийняття обґрунтованих рішень користувач системи оперує різною інформацією, як внутрішньою (кількість придбаних квитків, зворотній зв'язок відвідувачів), так і зовнішньою (тренди кіноіндустрії, рейтинги фільмів, сезонність попиту). Для обробки цих даних застосовуються комплексні методології, що ґрунтуються на попередньому аналізі поведінки цільової аудиторії та прогнозування на основі базових графових алгоритмів.

Фактори, що впливають на прийняття рішень у системі для надання рекомендацій мають різну вагу для кожного конкретного рекомендаційну

алгоритму, але в глобальному сенсі є подібними. Таким чином, система враховує наявність спільних характеристик між різними об'єктами системи. До цих характеристик відносяться: режисер, жанр, актори, рік та країна виробництва. Крім цього, рейтинг фільму та ступінь його новизни також є тими факторами, що впливають на кінцевий результат рекомендацій.

Система рекомендацій для кінотеатру найчастіше функціонує в умовах відкритих задач або кризових ситуацій. Типовими прикладами таких ситуацій є раптова зміна споживацьких настроїв або поява нових конкурентів. У таких випадках рішення приймаються за умов ризику, тобто коли ймовірності успіху можна приблизно оцінити, або невизначеності, коли прогнозування ускладнене. У випадках нечітких цілей, наприклад балансування між комерційними показниками та мистецькою цінністю програми, система застосовує відкриті рішення, які можуть коригуватися з надходженням нової інформації.

Організаційна структура рекомендаційної системи для кінотеатру включає кілька ключових робочих місць, кожне з яких виконує специфічні функції. Системний адміністратор відповідає за налаштування серверів, виправлення технічних помилок, реалізація доступності до всіх функцій платформи, надання технічної підтримки всім користувачам систем. Адміністратор платформи здійснює управління користувачами системи, забезпечує належний рівень її наповненості, контролює актуальність та результативність рекомендаційних методів, що використовуються. Ще одним об'єктом системи є її кінцевий користувач. Розробка відповідної системи орієнтована, перш за все, менеджерів з прокату або керівників репертуару, які і відповідають за процес формування кінопрокату кінотеатру та планування майбутніх кіносеансів.

Системний адміністратор:

- Здійснює встановлення, налаштування та оновлення серверів платформи;
- Відповідає за технічне обслуговування та діагностику помилок у роботі системи;

- Забезпечує безперебійну роботу платформи, вирішення проблем доступності;
- Надає технічну підтримку для користувачів, що включає оперативну допомогу у випадку технічних збоїв;
- Здійснює контроль над справністю інтеграції платформи з іншими інструментами чи сервісами (наприклад, інтеграції з відкрити кінобазами для отримання актуальної інформації про фільми).

Адміністратор платформи:

- Здійснює управління користувачами, що включає реєстрацію, зміну прав доступу, видалення акаунтів;
- Здійснює контроль за актуальності даних, що є в системі
- Наповнює систему новими фільмами
- Контролює результативність рекомендаційних алгоритмів, які використовуються у системі
- Забезпечує безпеку даних, управління доступом до конфіденційної інформації, здійснює резервне копіювання даних;
- Налаштовує загальні параметри платформи;
- Здійснює моніторинг та вирішення адміністративних запитів (наприклад, відновлення паролів, створення нових курсів тощо).

Користувач системи (менеджер з прокату, або керівник репертуару кінотеатру, який приймає рішення про показ фільмів):

- Пошук фільмів за подібністю; здійснення підбору фільмів, що схожі на ті, що принесли найбільший прибуток на основі аналізу даних попередніх прокатів. Здійснюється з врахуванням з урахуванням жанрової, акторської та режисерської подібності
- Пошук фільмів з використанням на основі різних характеристик (за країною, режисером, акторами тощо) для точного підбору фільмів;
- Ознайомлення з загальними описовими характеристиками запронованих фільмів (режисер, актори, жанри, опис, рік випуску, рейтинг тощо), що необхідні для своєчасного планування репертуару;

- Відстеження прем'єр шляхом інформування про нові релізи для своєчасного планування репертуару;
- Формування особистого списку збережених фільмів для їх майбутньої інтеграції у кінопрокат.

Для рекомендаційної системи важливим аспектом є ефективне управління різними типами інформації для підтримки прийняття рішень на всіх рівнях. Кожен користувач системи має доступ до конкретних даних, які відповідають його ролі, а також потребам звітування та аналізу.

Системний адміністратор відповідає за забезпечення безперебійної роботи платформи та збереження даних. Важливими даними для нього є інформація про стан серверів, наявність помилок на рівні системи, журнал аудиту доступу та звіти про безпеку. Він також здійснює моніторинг продуктивності платформи, визначаючи можливі навантаження, збої або інші технічні проблеми, які потребують негайного вирішення.

Адміністратор платформи має доступ до бази даних, яка містить всю інформацію про фільми та користувачів. Крім цього, саме адміністратор керує правами доступу, які надані користувачам системи, здійснює конфігураційні налаштування платформи, параметри безпеки тощо. Завдяки доступу до адміністративних параметрів, адміністратор може обмежити дії користувачів системи. Наприклад, новий користувач, який не має створеного облікового запису або просто не авторизований у системі не матиме доступу до функціону системи до моменту здійснення процесу авторизації та аутентифікації.

Користувачі системи мають доступ до даних про фільми. Діяльність користувачів у системі відбувається через користувацький web-інтерфейс, який включає сторінки з описами фільмів, а також функціонал для пошуку та фільтрації контенту. Крім того, інтерфейс забезпечує можливість переглядати персоналізовані рекомендації та формувати список фільму для майбутнього прокату.

Розробка рекомендаційної системи для кінотеатру стикається з набором викликів, що знаходяться на перетині технологічної реалізації та бізнес-потреб

кінотеатральної індустрії. Один з них полягає у створенні інтуїтивно зрозумілого інтерфейсу, який дозволяє менеджерам не лише отримувати персоналізовані рекомендації на основі різноманітних параметрів (жанри, актори, режисери, країни виробництва), але й в подальшому ефективно маніпулювати отриманими даними для розробки оптимальної стратегії кінопрокатів. Додаткову технічну складність становить необхідність реалізації гібридного підходу до рекомендацій, що потребує поєднання алгоритмічної точності з гнучкістю людського відбору. Описана задача вимагає створення складної архітектури на технічному рівні. Автоматизовані графові алгоритми повинні безперешкодно взаємодіяти з інтерфейсом користувача, обробляючи два принципово різні типи вхідних даних: структуровані метрики та неструктуровані суб'єктивні рішення менеджерів.

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Для визначення рівня розвитку наявних систем, було проведено аналіз літературних джерел й існуючих досліджень. Велика кількість наукових робіт свідчить про активний розвиток рекомендаційних систем у сфері кінематографу, зокрема із використанням алгоритмів та сучасних підходів на основі графових баз даних. Водночас, незважаючи на наявність потужного теоретичного підґрунтя для розробки подібної платформи, на сьогодні відсутні готові комерційні програмні продукти, що надають користувачам подібний функціонал та є у відкритому доступі. Існуючі програмні рішення, як правило, є частинами великих платформ або використовуються у внутрішніх системах кінотеатрів чи стрімінгових сервісів.

Автор наукової статті «Graph Theory: Applications in Movie Recommendation Systems (Netflix)» аналізує ефективність базових графових алгоритмів та їх результативність для надання рекомендацій на відомій стрімінговій платформі Netflix [8]. В основі рекомендаційного блоку системи лежить графова база даних, що зберігає інформацію про користувачів та фільми у вигляді вершин. Ребрами графу виступають зв'язки між цими елементами, з'єднуючи користувачів з фільмами, які

вони переглянули або оцінили. Як рекомендаційний алгоритм використано метод PageRank, логіка якого зосереджена на вимірюванні важливості об'єктів у структурі графа. Чим більше з'єднань має об'єкт з іншими об'єктами, та чим вищою є вага цих з'єднань, тим важливішим є цей об'єкт. Таким чином, фільми, які мають велику кількість зв'язків з популярними, а відповідно до цього і вагомими для системи, фільмами, отримують вищий рейтинг PageRank. Для вимірювання релевантності рекомендацій, було здійснено статистичний аналіз рівня точності рекомендацій та проведено опитування для оцінки рівня задоволеності користувачів. Обидва види інтерпретації результатів продемонстрували ефективність графового алгоритму PageRank для надання рекомендацій, що відповідають індивідуальним вподобанням користувачів стрімінгової платформи.

Ще одним дослідження, що розглядає новітні методи для побудови рекомендаційних систем є «MoRGH: Movie Recommender System using GNNs on Heterogeneous Graphs» [16]. Об'єктом дослідження є система, побудована на основі гетерогенного графу, тобто графу, що побудований із вузлів різних типів. У графі для описаної системи використовується два типи зв'язків:

1) зв'язок, що демонструє взаємодію між користувачем та фільмом. Граф «користувач-фільм» утворює двочасткову структуру, де вузли представляють користувачів та фільми, а ребра вказують на оцінки, надані користувачами.

2) зв'язок, що демонструє подібність між об'єктами-фільмами. Граф «фільм-фільм» з'єднує подібні фільми на основі подібності синопсису сюжету та жанрової подібності. Для подібності сюжету система використовує попередньо навчені BERT-надбудови, зі застосування косинусної міри подібності, тоді як жанрова подібність розраховується за допомогою алгоритму Жаккара.

У кінцевому графі з'єднуються лише фільми, коефіцієнт подібності яких в обох показниках перевищує значення 0.95, що забезпечує формування змістових зв'язків між елементами системи. Автори зазначають, що подібні моделі працюють швидше за своїх попередників. Крім цього, інтеграція даних методів може стати критично важливою для систем, у яких наявна проблема «холодного старту».

У науковій статті «Hybrid Movie Recommendation System» [5] розглянуто можливі шляхи мінімізації проблеми «холодного старту» та розрідженості даних у системах рекомендацій фільмів. Як стверджують автори, традиційні методи побудови рекомендаційних систем неефективно справляються з великою кількістю даних, особливо у контексті зі зростаючим різноманіттям користувацьких вподобань. Як вирішення цих проблем пропонується використання гібридного методу, що використовує фільтрацію на основі контенту та колаборативну фільтрацію. Фільтрація на основі контенту зосереджена на аналізі метаданих фільму (жанр, режисер, акторський склад) для визначення їх подібності. У цей час колаборативна фільтрація аналізує взаємодію користувачів із фільмами, при цьому зосереджуючись на рейтингах фільмів та поведінці користувачів у системі. Оцінка продуктивності гібридного методу демонструє, що точність наданих рекомендацій рівна значенню 0.85. На противагу цьому, фільтрація на основі контенту досягла точності 0.74, тоді як колаборативна фільтрація – 0.78.

Автори статті «Temporal-Guided Knowledge Graph-Enhanced Graph Convolutional Network for Personalized Movie Recommendation Systems» пропонують інноваційний алгоритм, використання якого вирішить обмеження традиційних рекомендаційних систем, які часто не враховують динамічність уподобань користувачів [2].

Запропонована авторами модель має багат шарову архітектуру, що включає шар вбудовування, шар графової згорткової мережі і повнозв'язані шари. Особливістю алгоритму є введення часового фактору загасання, що надає більшої ваги нещодавнім взаємодіям користувача з контентом. Використання цього часового фактору модифікує початкові рейтинги, пріоритезуючи новіші взаємодії над старішими. У шарі вбудовування здійснюється перетворення ідентифікаторів користувачів і фільмів у вектори фіксованого розміру, а шар GCN оновлює ці вбудовування, використовуючи структуру графа, що забезпечує зрозумілість нелінійні зв'язки між користувачами та фільмами. На фінальному етапі повнозв'язані шари відображають оновлені вбудовування на вихідний прогноз рейтингу. Запропонована модель продемонструвала кращі результати точності

наданих алгоритмів, ніж інші сучасні моделі. У той же час, автори стверджують, що використання графової згорткової мережі може вплинути на підвищення обчислювальної складності та обмежити масштабованість системи зі зростанням кількості даних.

Проведений аналіз демонструє, що наразі пошук нових програмно-технічних рішень для реалізації та забезпечення ефективності роботи рекомендаційних систем залишається актуальним, адже невпинно зростаюча кількість даних та їх різноманіття створюють нові виклики для цієї предметної області.

Висновки до розділу 1

У першому розділі бакалаврського проєкту було проведено всебічний аналіз предметної області та об'єкта дослідження. У розділі сформульовано ключову задачу системи та її економічний зміст, надано опис рішень, що приймають в предметній галузі та взаємозв'язки між ними.

Важливим в контексті проєктованої системи є визначення ролей користувачів системи, а саме: системного адміністратора, адміністратора та кінцевих користувачів. Деталізація ролей користувачів є необхідним для створення гнучкої, адаптивної та зручної системи, адже кожна з цих груп має свої специфічні потреби та функціональні обов'язки, які визначають, як вони взаємодіють із системою.

Під час досліджено предметної області було здійснено аналіз літературних джерел, що слугують теоретично-практичною базою для планування та реалізації подальших проєктних робіт. Велика кількість наукових статей на цю тему свідчить про різноманіття методів та підходів, які застосовуються під час розробки рекомендаційних систем, і надають потужне теоретичне підґрунтя для проведення майбутнього дослідження.

У результаті проведеного аналізу було визначено оптимальний набір технологій та методологічних підходів, що є основою для практичної реалізації рекомендаційної системи кінотеатру в рамках бакалаврської роботи. Детальне дослідження предметної області дозволило обґрунтовано вибрати графові БД як ключову технологію для зберігання та обробки складних взаємозв'язків між

об'єктами, що характеризують сферу кінематографу. Саме графові БД забезпечують значно вищу ефективність порівняно з традиційними реляційними СУБД. Як основу реалізації рекомендаційних методів обрано графові алгоритми Similarity та PageRank. Кожен з алгоритмів пропонує свої переваги в контексті рекомендаційної системи, тому прийнято рішень про можливість їх комбінації. Комбінація алгоритмів PageRank та Similarity, які виконують хоч і різні, але взаємодоповнюючі функції, може значно підвищити точність наданих рекомендацій.

РОЗДІЛ 2

РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної системи

Етап інжинірингу вимог передбачає визначення всіх потреб користувачів та бізнесу. Ця рання стадія розробки вимог встановлює загальне уявлення про те, що може робити система, та переваги, які вона може забезпечити. Надалі кількість та складність вимог доцільно врахувати в техніко-економічному обґрунтуванні, для оцінки того, чи є система технічно та фінансово доцільною.

Процесу формування вимог до системи передують визначення ключових стейкхолдерів та їх бізнес-цілей.

Бізнес-ціль – це чітко сформульований, вимірювальний результат, якого організація прагне досягти за допомогою розробки або впровадження програмного забезпечення, що формулюється з точки зору цінності для бізнесу.

Внутрішніми стейкхолдерами у процесі розробки рекомендаційної системи для кінотеатру виступають керівництво та власники кінотеатрів [17]. Їх бізнес-цілі визначені необхідністю максимізації прибутку кінотеатрів, підвищення рівня заповнюваності залів, оптимізації списку фільмів відповідно до вподобань цільової аудиторії з метою залучення нових у утримання існуючих клієнтів, зростання вартості бізнесу.

До зовнішніх стейкхолдерів належать:

- дистриб'ютори фільмів, бізнес-ціллю яких є збільшення попиту та зростання прибутку від фільмів, які вони пропонують кінотеатрам для прокату;
- інвестори, бізнес-цілі яких зосереджені на отриманні прибутку, мінімізації технічних, фінансових та ринкових ризиків та підвищенні цінності активів;
- державні органи, бізнес-цілі яких обумовлені необхідністю дотримання всіх актуальних нормативно-правових актів та здійснення контролю за дотриманням захисту прав споживачів.

Сформульовані бізнес-цілі слугують підґрунтям до визначення вимог системи, забезпечуючи чітке розуміння кінцевої мети. Вимоги до системи – це опис умов та можливостей, які мають бути реалізовані у системі або її окремих компонентах для задоволення встановлених стандартів та задоволення потреб користувачів системи. Вимоги поділяють на функціональні та нефункціональні.

Функціональні вимоги являють собою специфікації поведінки системи, що описують функції, які система повинна виконувати, а також визначають взаємодію системи з користувачами або іншими системами. У деяких випадках функціональні вимоги можуть також чітко вказувати, чого система не повинна робити [12].

Функціональні вимоги залежать від типу програмного забезпечення, що розробляється, очікуваних користувачів програмного забезпечення та загального підходу, який організація застосовує під час написання вимог. Нижче наведено перелік функціональних вимог до рекомендаційної системи для кінотеатру (таблиця 2.1).

Таблиця 2.1 – Функціональні вимоги системи

Ідентифікатор вимоги	Стисла назва	Повна назва вимоги	Складність	Пріоритетність	Ризикованість	Примітки (детальне пояснення)	Інші властивості
1	2	3	4	5	6	7	8
FREC001	Реєстрація	Реєстрація нового користувача	Середня	Висока	Низька	Обліковий запис нового користувача з username, email, паролем	Аутентифікація, UI/UX

Продовження таблиці 2.1

1	2	3	4	5	6	7	8
FREC00 2	Авторизація	Авторизація зареєстрованого користувача у системі	Середня	Висока	Низька	Вхід до профілю з перевіркою даних	Безпека, UI/UX
FREC00 3	Рекомендації на основі загальної подібності	Формування рекомендацій на основі подібності фільмів за атрибутами	Середня	Висока	Середня	Підрахунок схожості за спільними атрибутами фільму	Рекомендації, Neo4j
FREC00 4	Рекомендації фільмів на основі заданих критеріїв	Персоналізовані рекомендації на основі атрибутів фільму	Висока	Висока	Середня	Рекомендація фільмів за подібністю параметрів: жанру, року, країни, режисера, актора.	Neo4j-запити, графи
FREC00 5	Рекомендація на основі вибірки фільмів	Рекомендації на основі ручного вибору користувачем вибірки фільмів	Висока	Висока	Висока	Користувач обирає декілька фільмів, система знаходить подібні через спільні властивості	Комбіновані фільтри

Продовження таблиці 2.1

1	2	3	4	5	6	7	8
FREC006	Прем'єри	Відображення актуальних або найближчих прем'єр	Низька	Середня	Низька	Інформація з БД, фільтрація за датою випуску	Зовнішні джерела (опц.)
FREC007	Збереження фільмів	Можливість зберігати фільми до персонального списку	Середня	Висока	Низька	Додавання/видалення фільмів до «обраного»	Neo4j
FREC008	Деталі фільму	Виведення повної інформації про фільм	Низька	Висока	Низька	Назва, опис, тривалість, рік, актори, режисер, трейлер тощо	UI/UX, Детальна картка фільму

Джерело: сформовано автором на основі виконаного дослідження

Нефункціональні вимоги являють собою обмеження або критерії якості, які визначають, яким чином система виконує свої функції, включаючи такі аспекти, як час відгуку системи, масштабованість, безпека, зручність використання та надійність. Нефункціональні вимоги часто є більш критичними, ніж окремі функціональні вимоги. Користувачі системи зазвичай можуть знайти способи обійти системну функцію, яка насправді не відповідає їхнім потребам, у той час як невиконання нефункціональної вимоги може означати, що вся система стане непридатною для використання. Хоча часто можна визначити, які компоненти системи реалізують конкретні функціональні вимоги, це часто складніше з

нефункціональними вимогами. Реалізація нефункціональних вимог може бути розподілена по всій системі, адже вони, найчастіше, впливають на загальну архітектуру системи, а не на її окремі компоненти [12].

Перелік нефункціональний вимог до системи наведено у таблиці 2.2.

Таблиця 2.2 – Нефункціональні вимоги системи

Ідентифікатор вимоги	Стисла назва	Повна назва вимоги	Складність	Пріоритетність	Ризикованість	Примітки (детальне пояснення)	Інші властивості
1	2	3	4	5	6	7	8
NFREC001	Кросбраузерність	Система має коректно працювати в популярних браузерах (Chrome, Firefox, Edge тощо)	Низька	Висока	Низька	Актуально для забезпечення широкого охоплення користувачів	Перевірка на останніх версіях браузерів
NFREC002	Адаптивний інтерфейс	Інтерфейс має адаптуватися до різних розмірів екранів (мобільні, планшети, десктоп)	Середня	Висока	Середня	Важливо для забезпечення позитивного UX на будь-якому пристрої	Застосування CSS Media Queries

Продовження таблиці 2.2

1	2	3	4	5	6	7	8
NFREC003	Надійність	Система має працювати стабільно без збоїв або втрати даних	Висока	Висока	Висока	Пов'язано з обробкою винятків, відмовостійкістю сервера, резервним копіюванням	Може вимагати багаторівневу перевірку
NFREC004	Масштабованість	Система повинна легко розширюватися з ростом кількості користувачів або даних	Висока	Висока	Середня	Актуально для роботи з великою кількістю запитів та даних у Neo4j	Враховується при виборі архітектури
NFREC005	Зручність	Інтерфейс має бути інтуїтивно зрозумілим і зручним для кінцевого користувача	Середня	Висока	Низька	Передбачає UX-тестування, використання іконок, підказок, логічної структури інтерфейсу	Може змінюватися після фідбеку

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.1 представлено діаграму, що ілюструє дворівневу ієрархію вимог до системи рекомендацій фільмів, яка поділяється на функціональні та нефункціональні компоненти, що попередньо описані в таблицях 2.1-2.2. Така структурована декомпозиція вимог забезпечує комплексний підхід до проектування системи рекомендацій, чітко визначаючи як функціональні, так і нефункціональні характеристики майбутньої системи.

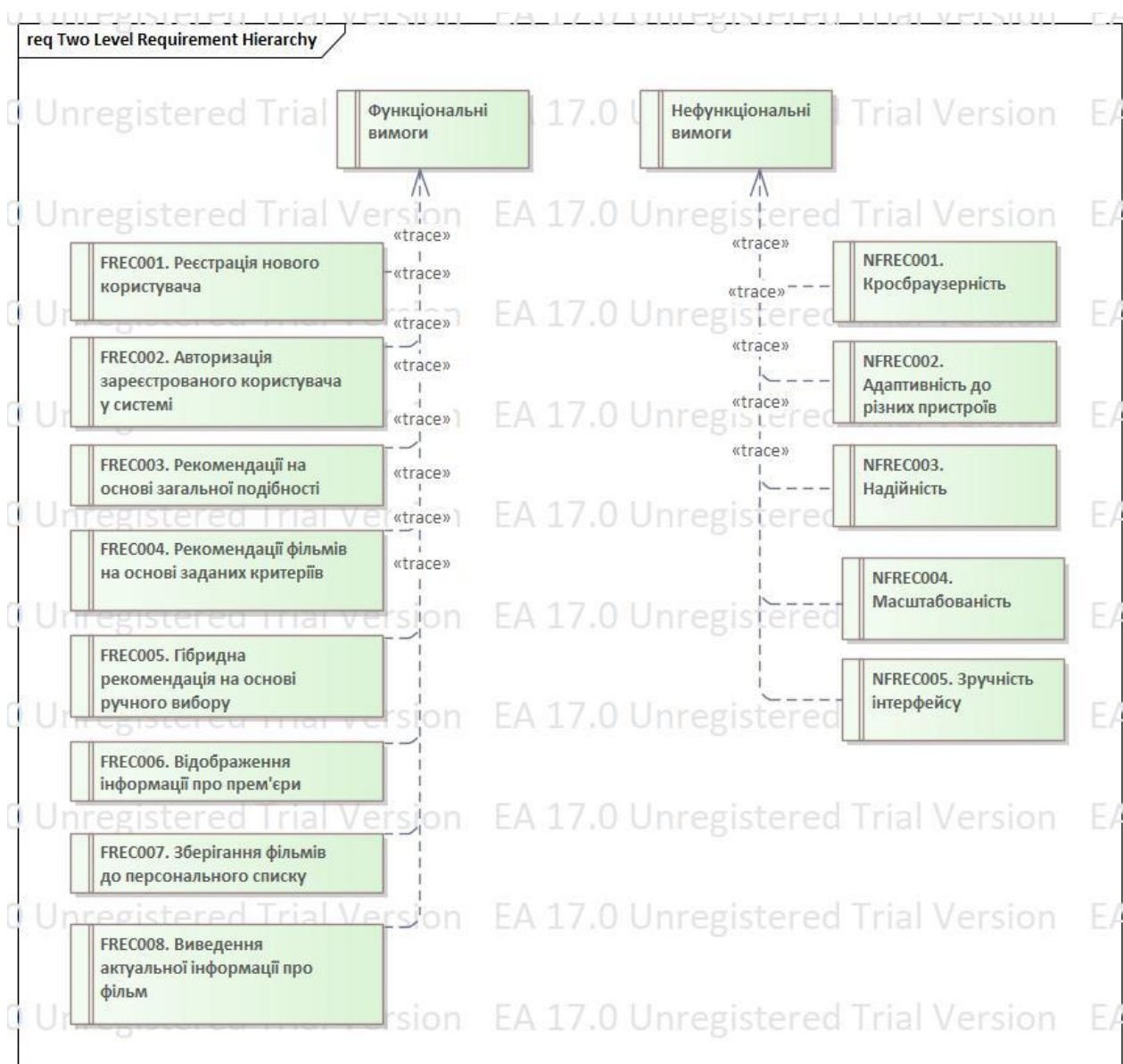


Рисунок 2.1 – Дворівнева ієрархія вимог до системи

Джерело: сформовано автором на основі виконаного дослідження

2.2 Постановка та алгоритм розв'язання задачі

2.2.1 Постановка задачі

Задача реалізації рекомендаційної системи для кінотеатру полягає у створенні централізованої платформи, що здійснює комплексний аналіз даних каталогу фільмів з метою формування персоналізованих рекомендацій для оптимізації кінопрокату, використовуючи графові алгоритми.

Ключова задача системи для кінотеатру – забезпечення релевантного та ефективного підбору кінопродукції для планування майбутнього прокату, особливо в умовах великого асортименту фільмів та обмеженого часового ресурсу менеджерів з репертуару. В умовах, де обсяги кінопродукції невпинно зростають, а самі кінотеатри мають як враховувати тренди кіноіндустрії, так і спиратись на потреби та вподобання відвідувачів, традиційні методи не дозволяють ефективно аналізувати великі обсяги взаємопов'язаної інформації.

Техніко-економічний зміст представленої задачі обумовлений оптимізацією фінансових та часових ресурсів кінотеатру шляхом впровадження ефективного механізму рекомендацій, в основі яких лежить аналіз вподобань цільової аудиторії. Таким чином, техніко-економічна суть задачі полягає у наступному:

1) Збільшення прибутку кінотеатру. Використання рекомендаційних платформ при плануванні кінопрокату дозволяє підвищити рівень наповненості кінозалів за рахунок релевантного підбору фільмів. Крім цього, збільшення прибутку обумовлене залученням нових клієнтів, а також утриманням вже існуючих, зважаючи на чітку орієнтацію на потреби цільової аудиторії, яка характерна для конкретного кінотеатру.

2) Оптимізація часових ресурсів. Автоматизація процесу пошуку фільмів за характеристиками значно оптимізує час працівників кінотеатру, які безпосередньо здійснюють формування прокату. Без використання подібних систем, окрім попереднього аналізу показників фільмів, які користувались попитом у кінотеатрі, працівник має додатково здійснити багатоетапний пошук подібного контенту за

набором різноманітних факторів. Розроблена рекомендаційна система дозволяє автоматизувати цей пошук, залишивши працівнику більше часу на безпосереднє дослідження цільової аудиторії.

3) Підвищення якості прийняття рішень. Система виявляє складні взаємозв'язки між об'єктами системи, де вибір фільмів для показу пов'язаний з рівнем подібності фільмів на основі їх спільних характеристик, впливаючи на майбутні рішення про закупівлю контенту та планування майбутніх маркетингових стратегій. Завдяки графовій структурі кінотеатр може ефективно прогнозувати попит, формуючи цілісну програму показів, яка відповідає інтересам аудиторії в довгостроковій перспективі.

4) Цифрова трансформація та її економічний вплив. Впровадження інтелектуальної рекомендаційної системи є складовою частиною глобальної цифрової трансформації у сфері дозвілля та розваг. Інтеграція сучасних інформаційних технологій у діяльність кінотеатру дозволяє створити новий формат взаємодії з глядачем – персоналізований, зручний та технологічно просунутий.

Обґрунтування потреби у розв'язанні задачі за допомогою ЕОМ полягає у використанні потужностей сучасних комп'ютерних систем для обробки великих масивів даних про фільми та забезпечення високого рівня рекомендацій. Використання графових БД дозволяє виявляти приховані закономірності через аналіз багаторівневих зв'язків між об'єктами системи, що забезпечує кращу ефективність при створенні комплексних репертуарних планів. Система автоматично виявляє складні взаємозв'язки між жанрами, акторами, режисерами, що в свою чергу сприяє підвищенню якості відбору кінопродукції та дозволяє менеджерам оперативно адаптувати репертуар відповідно до змін у запитах цільової аудиторії. Таким чином, інтеграція описаної ЕОМ у сферу репертуарного планування кінотеатру не лише спрощує аналітичну роботу персоналу, а й суттєво підвищує її гнучкість, точність і швидкість. Це, у свою чергу, відіграє ключову роль у контексті ефективного прийняття рішень щодо добору кінопродукції, що є критично важливим у динамічному середовищі сучасної кіноіндустрії, де гнучке

реагування на зміну попиту відіграє ключову роль у забезпеченні конкурентоспроможності закладу.

Щоб краще зрозуміти функціональні можливості рекомендаційної системи для кінотеатру, розглянемо перелік користувачів та об'єктів, управління якими здійснюється у системі. До користувачів системи належать:

Менеджери з прокату/керівники репертуару. Є ключовими користувачами системи, що здійснюють прийняття рішень щодо чи інших фільмів до репертуару кінотеатру. Використання ними рекомендаційної системи обумовлено необхідністю пошуку релевантного контенту відповідно до попередньо проаналізованих вподобань аудиторії, сезонних кінотрендів, подій тощо.

Адміністратор платформи. Відповідальний за щоденну роботу системи: створення та редагування об'єктів, а також за керування правами доступу користувачів. Він забезпечує цілісність і актуальність даних, контролює логічні зв'язки між об'єктами та може здійснювати ручне коригування інформації відповідно до потреб персоналу кінотеатру.

Системний адміністратор. Здійснює підтримку технічної інфраструктури платформи, що включає встановлення та налаштування серверного середовища, регулярне оновлення програмного забезпечення, діагностику та усунення технічних помилок.

До об'єктів системи належать:

Фільми – основні елементи бази даних, кожен з яких має набір атрибутів: назву, дату випуску, тривалість, рейтинг, опис тощо. Об'єкти фільмів пов'язані зв'язками з усіма іншими об'єктами системи, що дозволяє будувати складні запити та знаходити подібність на основі великої різноманітності атрибутів.

Жанри – класифікаційні категорії, до яких належать фільми. Один фільм може бути пов'язаний з кількома жанрами, наприклад: комедія, драма, трилер тощо. Жанри допомагають структурувати каталог і реалізовувати пошук за тематикою кінострічок.

Актори – окремий тип об'єктів, що відображає виконавців головних та другорядних ролей у фільмах, та дозволяє формувати підбірки стрічок за участю певного актора.

Країни – позначають географічне походження фільмів і можуть бути використані як критерій підбору. Наприклад, система бути використана для пошуку виключно українського, японського чи американського кінематографу залежно від вподобань цільової аудиторії, кінотрендів або тематики заходу, який планується провести у кінотеатрі.

Режисери – об'єкти системи, що пов'язують фільми з їхніми постановниками. Завдяки таким зв'язкам система дозволяє формувати рекомендації на основі творчості конкретних режисерів, що особливо актуально для організації тематичних показів, фестивальних програм або ретроспектив.

Користувачі – всі суб'єкти, що взаємодіють із системою. Облік користувачів дозволяє впроваджувати контроль доступу до функцій системи, а також підтримувати персоналізоване середовище роботи, реалізуючи можливість для формування персональних списків збережень.

Автоматизоване розв'язання задачі припиняється у випадку зупинки діяльності організації, яка її підтримує, або внаслідок технічних проблем, що унеможлиблюють доступ до платформи. До таких проблем належать перебої з електропостачанням, несправності серверного обладнання, перебої у роботі системного програмного забезпечення, включно з графовою базою даних, а також випадкове або некоректне видалення критичних об'єктів, як-от фільмів, зв'язків або записів користувачів. Кібератаки, зокрема несанкціонований доступ до облікових записів адміністратора, можуть призвести до втрати керування системою або повного блокування її роботи. Відсутність належного резервного копіювання значно підвищує ризик втрати важливої інформації, адже за відсутності заздалегідь створених резервних копій відновлення даних може бути тривалим і неповним. Також можливі порушення функціональності через застарілість окремих компонентів, несумісність із новими версіями ПЗ. Усі ці ризики можуть бути

мінімізовані за умови налагодженого резервування даних, регулярних оновлень, здійснення технічного моніторингу та системного адміністрування.

Досягнення кінцевої мети задачі пов'язане ще з однією задачею, яка передбачає аналіз цільової аудиторії, який має проводитись за межами системи та передувати етапу формування кінопрокату. Аналіз поведінкових характеристик відвідувачів кінотеатру, їх жанрових уподобань, частоти відвідувань, реакція на попередні покази, вікові та соціальні параметри, дозволяє сформувати повніше уявлення про актуальні запити. Процес дослідження цільової аудиторії може включати як аналіз історичних даних продажів квитків і відвідуваності, так і опитування, відгуки, дані з соціальних мереж чи реакції на маркетингові кампанії. Саме гарне розуміння потреб та вподобань відвідувачів дозволяє визначити, які фільми мають більший потенціал для успішного прокату в конкретному кінотеатрі. Таким чином, детальний попередній аналіз та чіткі механізми рекомендацій дозволяють досягти високої релевантності наданих рекомендацій, що відповідатимуть реальним очікуванням і запитам глядачів, підвищуючи ймовірність успішного заповнення кінозалів.

Вихідні дані системи – це результат, який генерує система після обробки вхідних даних або виконання певної операції. Вони можуть бути представлені в різних формах залежно від типу системи та завдань, які вона виконує.

Вихідні повідомлення відіграють важливу роль у рекомендаційній системі, адже забезпечують ефективну комунікацію між системою та її користувачами. Вихідні повідомлення служать для інформування користувача про результат виконання його дій, зміни у стані системи, появу нових рекомендацій або виникнення помилок, що потребують його уваги. Система повідомлень повинна бути чіткою, своєчасною та зручною для користувачів, що дозволяє мінімізувати непорозуміння, уникнути помилок у взаємодії та сприяти ефективному використанню функціоналу системи. Повідомлення мають містити зрозумілий зміст та бути логічно пов'язаними з діями користувача. Крім цього, вибір каналів передачі та термінів надходження повідомлень значно впливає на ефективність системи та задоволення користувачів. Повідомлення, що надходять надто пізно або

не в тій формі, що очікується, можуть знизити довіру до системи та її сприйняття як зручного інструменту. Перелік вихідних повідомлень, які використовуються у розробленій системі наведено у таблиці 2.3.

Таблиця 2.3 – Вихідні повідомлення

Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність видання	Термін видання і допустимий час затримки	Користувачі інформації
Успішна авторизація в системі	LOGIN_SUCCESS	Автоматичне перенаправлення після входу	Після дії	Негайно (до 1 сек)	Користувач системи
Помилка авторизації	LOGIN_ERROR	Вспиваючий confirm dialog "All fields are required.", "Invalid email address.", "Invalid credentials."	Після дії	Негайно (до 1 сек)	Користувач системи
Успішна реєстрація в системі	SIGIN_SUCCESS	Автоматичне перенаправлення після входу	Після дії	Негайно (до 1 сек)	Користувач системи
Помилка реєстрації	SIGIN_ERROR	Вспиваючий confirm dialog "All fields are required.", "Invalid email address.", "Username or email already exists."	Після дії	Негайно (до 1 сек)	Користувач системи

Продовження таблиці 2.3

Виведення картки фільму	OUT_FILM_INFO	Картка фільму з його характеристика	Після дії	Негайно (до 1 сек)	Користувач системи
Успішне збереження фільму	SAVE_SUCCESS	Активація елементу збереження	Після дії	Негайно (до 1 сек)	Користувач системи
Успішне видалення фільму зі збережених	SAVE_SUCCESS	Деактивація елементу збереження	Після дії	Негайно (до 1 сек)	Користувач системи
Виведення рекомендації на основі загальної подібності	REC_SIMILARITY	Список з 6 фільмів у графічному інтерфейсі	За запитом користувача	2-5 сек	Користувач системи
Виведення рекомендації на основі заданих критеріїв	REC_BY_CRITERIA	Список з 6 фільмів у графічному інтерфейсі	За запитом користувача	2-5 сек	Користувач системи
Виведення рекомендації на основі вибірки фільмів	REC_HYBRID	Список з 6 фільмів у графічному інтерфейсі	За запитом користувача	2-5 сек	Користувач системи

Джерело: сформовано автором на основі виконаного дослідження

Вхідні дані – це інформація або сигнали, які подаються в систему (програму, пристрій, алгоритм чи іншу форму обчислювальної системи) для обробки. Ці дані слугують основою для виконання задачі або отримання певного результату.

Реалізація системи рекомендацій вимагає великої кількості вхідної інформації, що включає в себе різноманітні запити та інтерфейси для збору даних, необхідних для ефективного управління процесом надання рекомендацій.

Різнороманітність форм подання запитів, від простого переходу на сторінку фільму до заповнення складних форм з критеріями пошуку, забезпечує гнучкість системи та можливість адаптації до різних рівнів технічної підготовки користувачів. Система повинна забезпечити швидку та зручну обробку таких запитів у різних форматах, забезпечуючи гнучкість як для адміністраторів, так і для кінцевих користувачів, які регулярно надсилають вхідну інформацію до системи.

Джерела вхідної інформації:

Форма реєстрації нових користувачів. Ця форма містить поля для введення особистих даних. Система управління користувачами забезпечує подальшу роботу з профілями, забезпечуючи надання необхідного функціоналу для різних типів зареєстрованих користувачів.

Форми для отримання рекомендацій. Заповнення та надсилання відповідних форм ініціює їх подальшу обробку на сервері та забезпечує надання найбільш релевантних рекомендацій відповідно до типу користувацького запиту.

Зовнішні джерела інформації. Джерелами інформації про фільми наразі є зовнішні ресурси, такі як IMDb [6], Rotten Tomatoes [11], JustWatch [7], TMDb [14].

Всі описані джерела інформації працюють узгоджено, забезпечуючи повноту та актуальність даних, необхідних для ефективного системи рекомендацій для кінотеатру.

Приклади вхідних повідомлень, які використовуються у розробленій системі наведено у таблиці 2.4.

Таблиця 2.4 – Вхідні повідомлення

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1	2	3	4	5
Реєстрація нового користувача	INP_USER_REGISTER	Форма для реєстрації	За ініціативою користувача	Користувач
Авторизація користувача	INP_USER_LOGIN	Форма для авторизації	За ініціативою користувача	Користувач
Інформація про фільм	INP_FILM_INFO_INPUT	Перехід на сторінку фільму	За ініціативою користувача	Користувач
Рекомендації на основі загальної подібності	INP_REC_SIMILAR	Активація кнопки пошуку "Get Similar"	За ініціативою користувача	Користувач
Рекомендації на основі заданих критеріїв	INP_REC_BY_CRITERIA	Форма із заповненими критеріями пошуку рекомендацій	За ініціативою користувача	Користувач
Рекомендація на основі вибірки фільмів	INP_REC_HYBRID	Активовані фільми з представленої вибірки	За ініціативою користувача	Користувач
Збереження фільмів	INP_USER_SAVE	Активація кнопки збереження	За ініціативою користувача	Користувач

Продовження таблиці 2.4

Видалення фільму зі збережень	INP_USER_UNSAVE	Деактивація кнопки збереження	За ініціативою користувача	Користувач
-------------------------------	-----------------	-------------------------------	----------------------------	------------

Джерело: сформовано автором на основі виконаного дослідження

Інформаційну модель рекомендаційної системи кінотеатру, яка демонструє комплексний підхід до обробки та управління даними для забезпечення персоналізованих рекомендацій користувачам, представлено на рисунку 2.2.

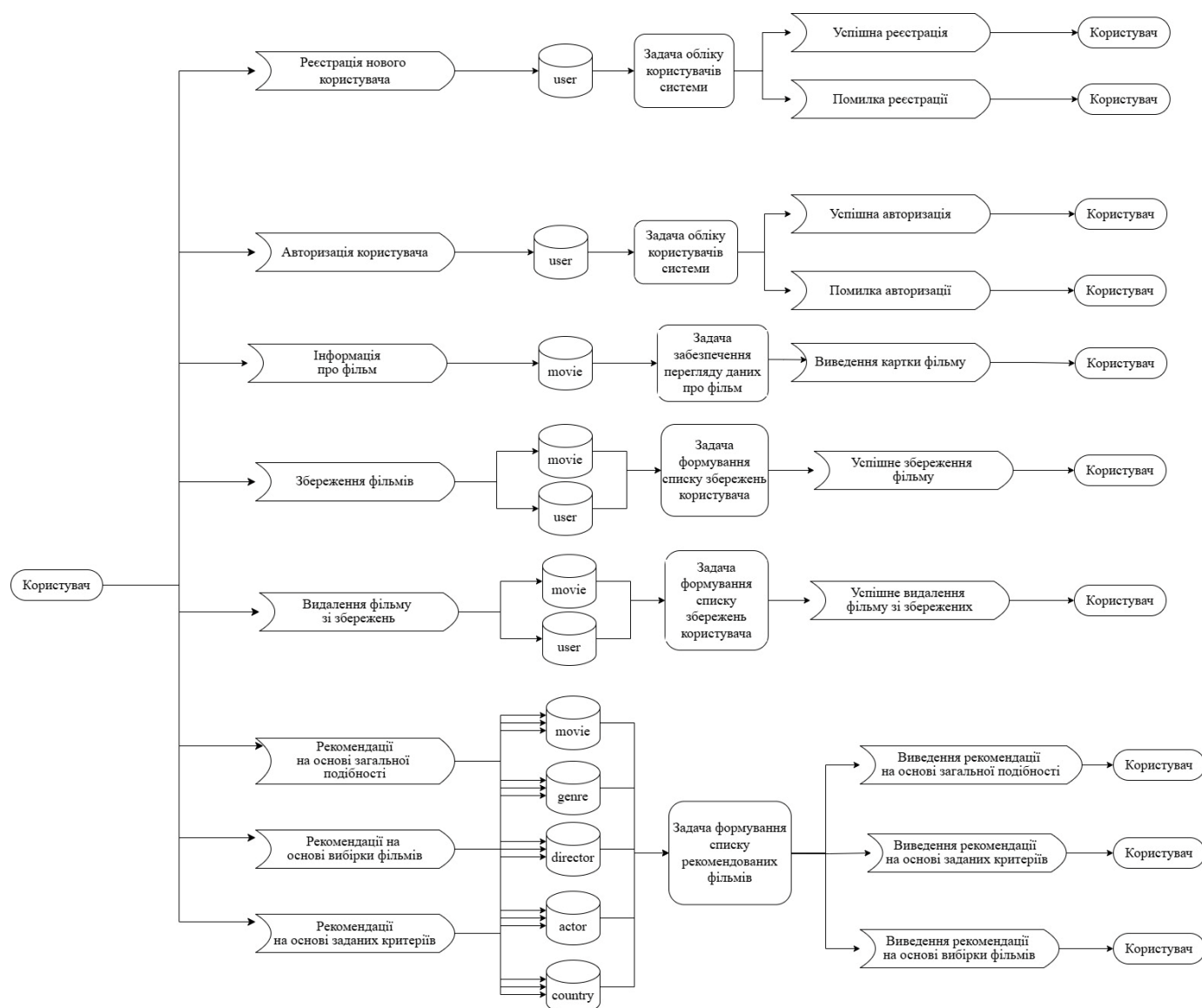


Рисунок 2.2 – Інформаційна модель системи

Джерело: сформовано автором на основі виконаного дослідження

Центральним елементом інформаційної моделі є користувач системи, який ініціює надсилання різних типів запитів та інформації. Перший рівень взаємодії включає реєстрацію та реєстрацію нових користувачів, де система збирає та обробляє базову інформацію про облікові записи та створює персональні профілі. Наступний компонент системи стосується обробки інформації про фільми. Користувачі можуть надсилати запити на отримання інформації про конкретні фільми, що зберігається в графовій базі даних. Система забезпечує виведення детальної інформації про фільми та передає ці дані користувачам системи.

Функціональність збереження та видалення фільмів дозволяє користувачам формувати персональні списки фільмів, які необхідно включити у майбутній кінопрокат. Система обробляє запити на збереження фільмів у персональний список збережень користувача, формуючи зв'язок між користувачем та фільмом, та надає повідомлення про успішне збереження фільму в особистому списку.

Найбільш важливим у системі є саме механізми генерації рекомендацій. Користувач може надіслати запит на отримання трьох різних категорій рекомендацій, залежно від його потреб та вподобань. Результатами виконання кожного з них є сформований список з фільмами, що рекомендуються до прокату.

Таким чином, представлена інформаційна модель рекомендаційної системи для кінотеатрів демонструє ефективну багаторівневу архітектуру, що базується на аналізі подібності фільмів на основі їх спільних характеристик для формування персоналізованих рекомендацій. Система успішно інтегрує процеси реєстрації користувачів, управління інформацією про фільми та обчислення метрик подібності між різними стрічками на основі їхніх атрибутів.

2.2.2 Алгоритм розв'язання задачі

Рекомендаційна система кінотеатру оперує великими обсягами структурованих даних, які організовані у вигляді масивів для здійснення їх обробки з метою надання релевантних рекомендацій. Нижче подано перелік об'єктів

графової БД, що включають дані про фільми та користувачів, класифікацію за жанрами, акторським складом, режисерами, країнами. Перелічені об'єкти формують базу знань системи та використовуються для аналізу графовими алгоритмами для надання персоналізованих рекомендацій (таблиця 2.5).

Таблиця 2.5 – Перелік об'єктів графової БД

Об'єкт	Ідентифікатор	Максимальна кількість записів
Фільми	movies	10000
Жанри	genres	100
Країни	countries	200
Режисери	directors	5000
Актори	actors	50000
Користувачі	users	7000

Джерело: сформовано автором на основі виконаного дослідження

Наразі графова БД для розробленої системи налічує наступну кількість об'єктів:

- 177 об'єктів фільмів, серед яких 119 зарубіжних та 58, які повністю чи частково вироблені в Україні
- 24 об'єкти жанрів
- 38 об'єктів країн
- 202 об'єкти режисерів, серед яких 7 є режисерами та акторами одночасно
- 469 об'єкти акторів, серед яких 7 є режисерами та акторами одночасно
- 2 користувачі

Результатна інформація рекомендаційної системи представлена у вигляді масивів, що містять оброблені дані для різних типів взаємодії з користувачем (таблиця 2.6). Описані масиви формуються для відображення рекомендованих фільмів на головній сторінці з високим рейтингом, надання переліку схожих

фільмів при виборі конкретного кінофільму, результатів розширеного пошуку за заданими критеріями, а також рекомендацій на основі аналізу вибірки фільмів, що обрана користувачем. Розмір масивів результатної інформації оптимізований для забезпечення швидкого відгуку системи та зручності сприйняття інформації користувачем.

Таблиця 2.6 – Перелік масивів результатної інформації

Масив	Ідентифікатор	Максимальна кількість записів
Рекомендовані фільми на головній сторінці	recommended_movies_main	10
Рекомендації фільмів за загальною подібністю	similar_rec	6
Рекомендації фільмів за заданими критеріями	selectsuggest_rec	48
Рекомендації на основі вибірки фільмів	hybrid_rec	9

Джерело: сформовано автором на основі виконаного дослідження

Математичне забезпечення, що лежить в основі рекомендаційної системи, забезпечує точне формування релевантних рекомендацій. До нього належать ряд математичних моделей, формул та методів, що застосовуються для обробки даних про фільм, підрахунку подібності фільмів на основі кількості спільних атрибутів та ваги елементі.

1) Розрахунок загальної подібності фільмів (2.1)

Нехай,

- M – фільм, для якого шукаємо рекомендації
- $R = \{r_1, r_2, \dots, r_n\}$ – множина рекомендованих фільмів
- $G(M) = \{g_1, g_2, \dots, g_n\}$ – множина жанрів фільму M
- $G(R) = \{g_1, g_2, \dots, g_n\}$ – множина жанрів фільмів R
- $C(M) = \{c_1, c_2, \dots, c_n\}$ – множина країн виробництва фільму M
- $C(R) = \{c_1, c_2, \dots, c_n\}$ – множина країн виробництва фільмів R

- $A(M) = \{a_1, a_2, \dots, a_n\}$ – множина акторів фільму M
- $A(R) = \{a_1, a_2, \dots, a_n\}$ – множина акторів фільмів R
- $D(M) = \{r_1, r_2, \dots, r_n\}$ – множина режисерів фільму M
- $D(R) = \{r_1, r_2, \dots, r_n\}$ – множина режисерів фільмів R
- $release_{date}(r_i)$ – дата релізу фільму r_i
- $rate(r_i)$ – рейтинг фільму r_i
- $commonGenres = |G(M) \cap G(R)|$
- $commonCountries = |C(M) \cap C(R)|$
- $sharedActors = |A(M) \cap A(R)|$
- $sharedDirectors = |D(M) \cap D(R)|$

(2.1)

$$score(R) = |G(M) \cap G(R)| + |C(M) \cap C(R)| + |A(M) \cap A(R)| + |D(M) \cap D(R)|,$$

$$\text{при} \begin{cases} R \mid R \neq M, \\ 2020 - 01 - 01 \leq release_{date}(r) \leq today, \\ rate(r) \geq 5, \\ commonGenres \geq 1, \\ commonCountries \geq 1 \end{cases}$$

2) Розрахунок подібності на основі визначених критеріїв (2.2)

Нехай,

- M – фільм, для якого шукаємо рекомендації
- $R = \{r_1, r_2, \dots, r_n\}$ – множина рекомендованих фільмів
- $G(r_i) = \{g_1, g_2, \dots, g_n\}$ – множина жанрів фільму r_i
- $A(r_i) = \{a_1, a_2, \dots, a_n\}$ – множина акторів фільму r_i
- $D(r_i) = \{d_1, d_2, \dots, d_n\}$ – множина режисерів фільму r_i
- $C(r_i) = \{c_1, c_2, \dots, c_n\}$ – множина країн виробництва фільму r_i
- $Y(r_i)$ – рік релізу фільму r_i
- $rate(r_i)$ – рейтинг фільму r_i
- $pr(r_i)$ – PageRank значення фільму r_i
- $today$ – поточна дата
- I – індикаторна функція, де

$I = 1$, якщо критерій визначено

$I = 0$, якщо критерій не визначено

- $S(m, r_i)$ – кількість спільних атрибутів між m і r_i
- $w_y = 14$
- $w_c = 14$
- $w_g = 15$
- $w_d = 30$
- $w_a = 30$
- $c = 20$
- $w_r = 0$
- $w_{pr} = 5$

$$r_i \in R \text{ при } \left\{ \begin{array}{l} R \mid R \neq M, \\ Y(r) \geq 2020, \\ \text{release_date}(r) \leq \text{today} \\ \text{rate}(r) \geq 5, \\ \text{score}(r) > 2, \\ S(m, r) \geq 5 \\ \text{totalScore} \geq 10 \end{array} \right.$$

(2.2)

$$\text{totalScore}(r_i) = w_y \times I_y(r_i) + w_c \times I_c(r_i) + w_g \times I_g(r_i) + w_d \times I_d(r_i) + w_a \times I_a(r_i) + w_s \times S(m, r_i) + w_r \times r(r_i) + w_{pr} \times pr(r_i)$$

3) Розрахунок подібності вибірки фільмів (2.3)

Нехай,

- $M = \{m_1, m_2, \dots, m_n\}$ – множина обраних фільмів
- $R = \{r_1, r_2, \dots, r_n\}$ – множина рекомендованих фільмів
- $G(M) = \{g_1, g_2, \dots, g_n\}$ – множина жанрів фільму M
- $G(R) = \{g_1, g_2, \dots, g_n\}$ – множина жанрів фільмів R
- $A(M) = \{a_1, a_2, \dots, a_n\}$ – множина акторів фільму M
- $A(R) = \{a_1, a_2, \dots, a_n\}$ – множина акторів фільмів R
- $D(M) = \{r_1, r_2, \dots, r_n\}$ – множина режисерів фільму M
- $D(R) = \{r_1, r_2, \dots, r_n\}$ – множина режисерів фільмів R
- $\text{release_date}(r_i)$ – дата релізу фільму r_i
- $\text{rate}(r_i)$ – рейтинг фільму r_i
- today – поточна дата
- $I(\cdot)$ – індикаторна функція, де

$I = 1$, якщо критерій виконується

$I = 0$, якщо критерій не виконується

- $sharedActors = | A(r) \cap A(m) |$
- $sharedDirectors = | D(r) \cap D(m) |$
- $commonGenres = | \bigcup_{m \in M} G(r) \cap G(m) |$

$$r \in R \text{ при } \begin{cases} R \mid R \neq M, \\ 2020 - 01 - 01 \leq release_{date(r)} \leq today, \\ rate(r) \geq 5, \\ score(r) > 2 \end{cases} \quad (2.3)$$

$$score(r_i) = \sum_{m \in M} [3 \times I(A(r) \cap A(m) \neq \emptyset) + 3 \times I(D(r) \cap D(m) \neq \emptyset)] + \begin{cases} 3, & \text{при } commonGenres \geq 2 \\ 1, & \text{при } commonGenres = 1 \\ 0, & \text{при } commonGenres > 1 \end{cases}$$

Таким чином, описані формули становлять фундаментальну математично-теоретичну базу для подальших етапів розробки та імплементації алгоритмічних рішень.

На наступному етапі здійснено побудову алгоритму розв'язання задачі. Візуальне представлення алгоритму є ефективним інструмент для чіткого розуміння логіки роботи програмної системи, що дозволяє структуровано подати етапи взаємодії користувача із системою, визначити послідовність дій і умови переходу між ними. Структурування у форматі блок-схем дозволяє легше виявляти помилки, оптимізувати процеси та забезпечити прозорість функціонування програми. Крім того, алгоритмічна візуалізація сприяє ефективнішому плануванню і впровадженню системи, а також полегшує її подальший супровід. На рисунку 2.3-2.7 подано алгоритми функціонування логічно-функціональних блоків рекомендаційної системи для кінотеатру.

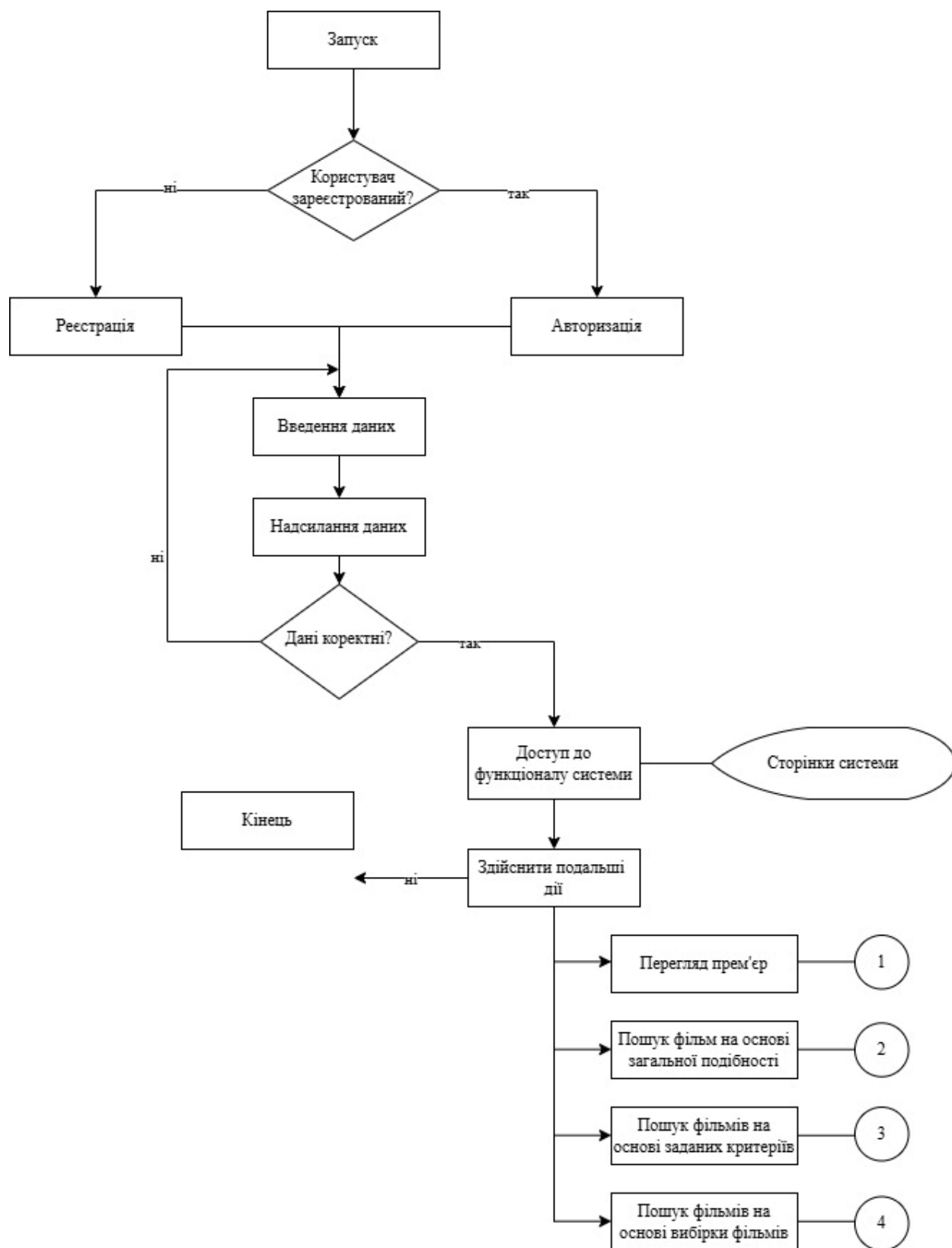


Рисунок 2.3 – Алгоритм авторизації користувача

Джерело: сформовано автором на основі виконаного дослідження

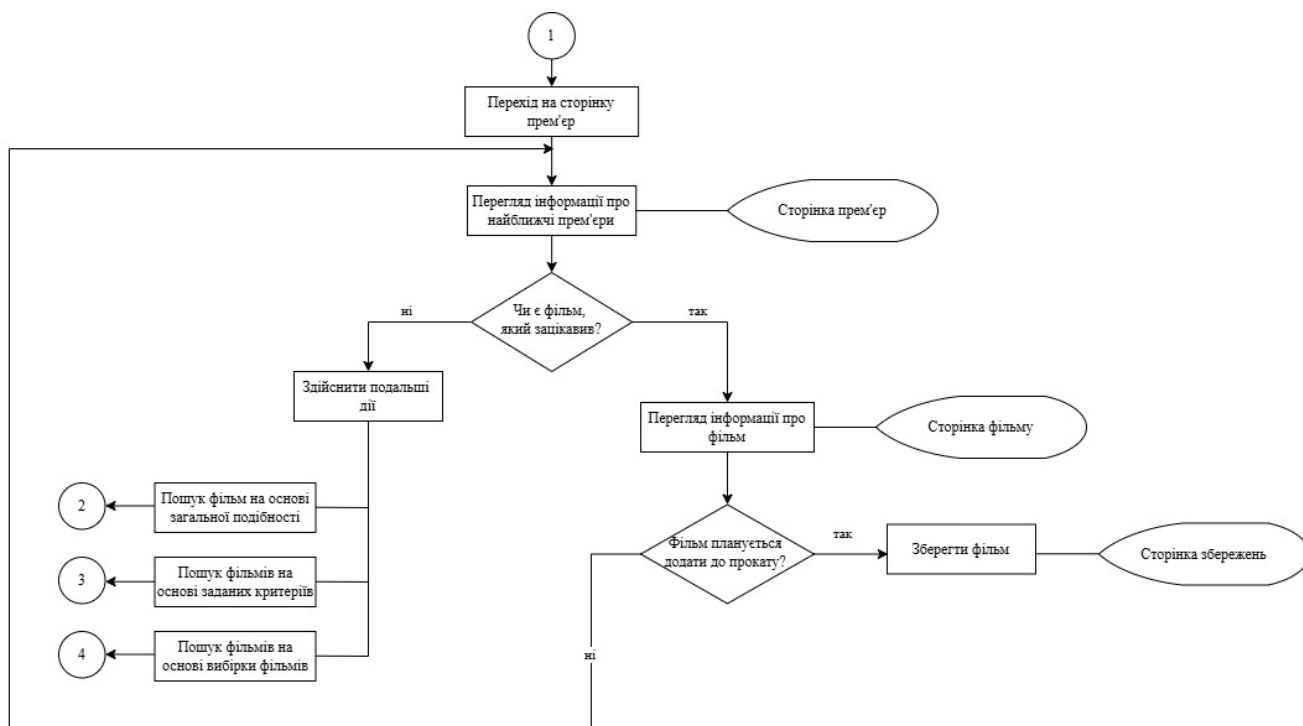


Рисунок 2.4 – Алгоритм взаємодії з списком прем'єр

Джерело: сформовано автором на основі виконаного дослідження

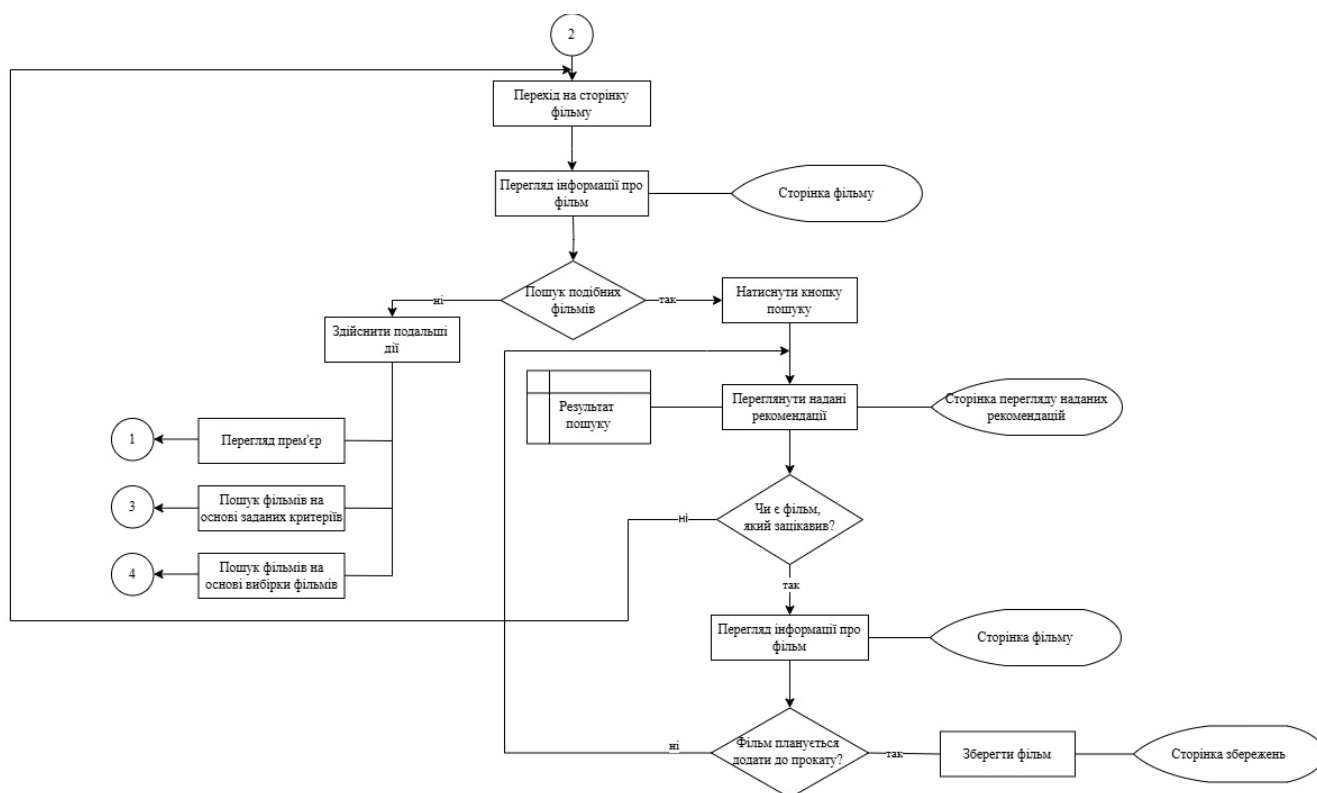


Рисунок 2.5 – Алгоритм пошуку фільмів на основі загальної подібності

Джерело: сформовано автором на основі виконаного дослідження

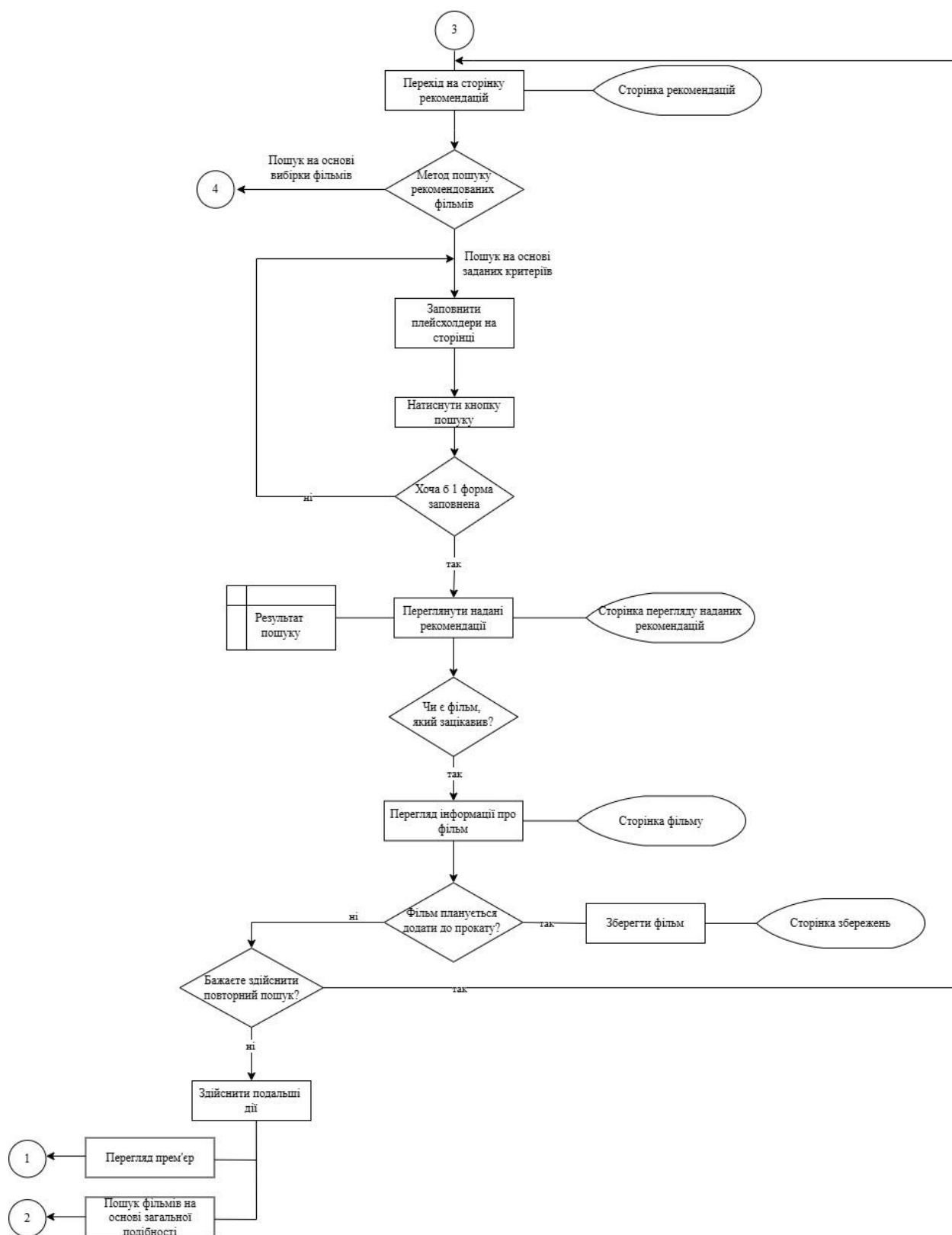


Рисунок 2.6 – Алгоритм пошуку фільмів пошуку фільм на основі заданих критеріїв

Джерело: сформовано автором на основі виконаного дослідження

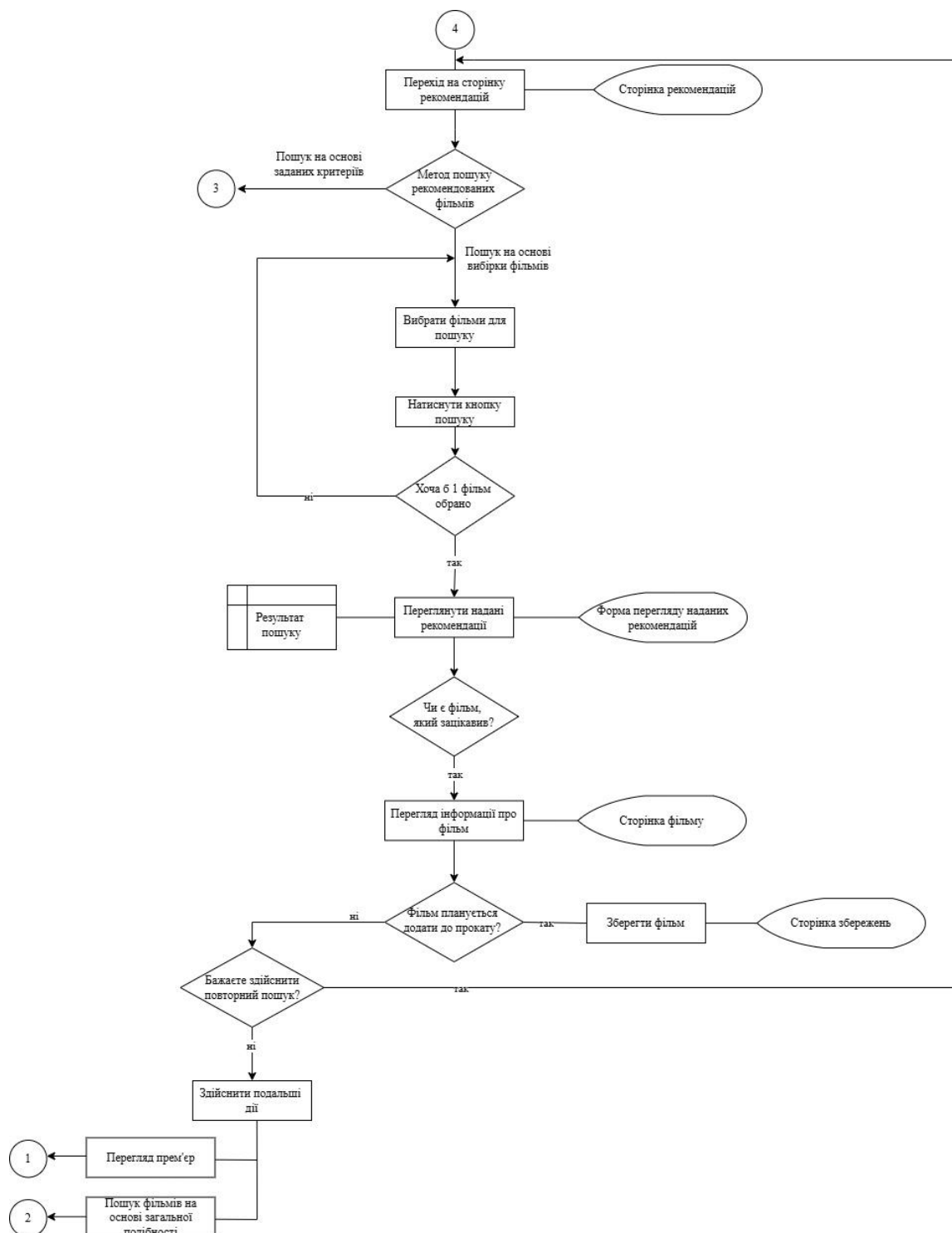


Рисунок 2.7 – Алгоритм пошуку фільм на основі визначеної вибірки

Джерело: сформовано автором на основі виконаного дослідження

Використання блок-схем є ефективним засобом проектування та документування логіки роботи інформаційних систем, адже дозволяють чітко візуалізувати послідовність дій, варіанти розвитку сценаріїв та умови переходів

між етапами, що особливо важливо для систем, орієнтованих на користувача. У контексті рекомендаційних систем блок-схеми сприяють кращому розумінню взаємодії між етапами – від реєстрації та авторизації до генерації персоналізованих рекомендацій.

2.3 Моделювання інформаційної системи

2.3.1 Моделювання поведінки системи

Моделювання структури рекомендаційної системи кінотеатру є фундаментальним та критично важливим етапом розробки, що дозволяє систематично визначити основні компоненти системи, встановити формальні зв'язки між ними та сформувати цілісне бачення архітектури майбутнього програмного рішення. У контексті рекомендаційної системи кінотеатру, яка характеризується високою складністю та різноманітним взаємодіючим компонентам, структурне моделювання системи є важливим етапом для забезпечення цілісності та функціональності системи. Процес моделювання структури системи дає можливість зосередитись на концептуальному логічному рівні системи, при цьому абстрагувавшись від конкретної програмної реалізації. Такий підхід не лише спрощує подальшу розробку, але й забезпечує відповідність системи заявленим функціональним та нефункціональним вимогам. Побудова ефективної моделі значно зменшує кількість потенційних помилок проєктування, забезпечуючи можливість раннього виявлення проблемних місць в архітектурі до початку безпосередньої розробки програмного коду.

Попереднє моделювання структури багатокомпонентної системи рекомендацій є ключовим для забезпечення її масштабованості, продуктивності та адаптивності до змін у вимогах чи умовах використання.

Для моделювання структури системи у даному розділі наведено ряд діаграм, кожна з яких забезпечує свій унікальний погляд на структуру системи, дозволяючи відобразити різні аспекти її організації.

Для точного розуміння функціональних можливостей розроблюваної системи було здійснено детальне моделювання її поведінки з використанням стандартизованих методів UML. Для моделювання поведінки системи було обрано два ключових типи діаграм: діаграму прецедентів (Use Case Diagram) та діаграму послідовностей (Sequence Diagram)

Діаграма Use Case є потужним інструментом для моделювання та аналізу функціональності інформаційної системи, оскільки вона дозволяє зобразити взаємодії між користувачами системи (акторами) та самою системою, підкреслюючи основні операції, що виконуються для досягнення певних цілей. В контексті проєктування рекомендаційної системи для кінотеатру, Use Case діаграма служить для чіткого визначення ролей таких акторів, як користувач та системний адміністратор, а також можливостей, що система має надавати кожному з них. Такий підхід дозволяє систематизувати вимоги до розробки програмного забезпечення і забезпечити відповідність між технічними можливостями платформи і вимогами кінцевих користувачів. Використання цієї діаграми вказує на важливість виокремлення основних процесів і забезпечення їхнього виконання в межах заданих сценаріїв взаємодії, таким чином полегшуючи розуміння того, як система повинна функціонувати в реальних умовах.

Всі представлені діаграми реалізовано у програмному середовищі Enterprise Architect [4].

На рисунку 2.8 наведено Use Case діаграму для проєктованої системи, яка включає опис акторів, що є користувачами системи, і відображення їх поведінки у системі.

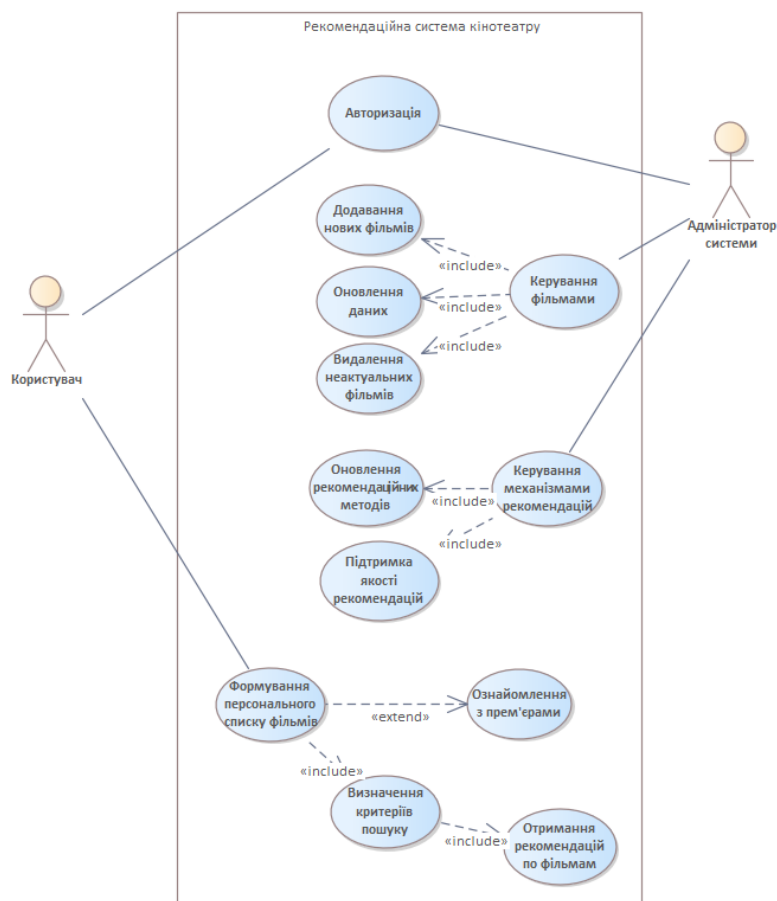


Рисунок 2.8 – Діаграма Use Case для рекомендаційної системи кінотеатру

Джерело: сформовано автором на основі виконаного дослідження

Представлена діаграма ілюструє комплексну архітектуру рекомендаційної системи кінотеатру, взаємодію з функціональними модулями якої здійснюють два ключові актори – Користувач та Адміністратор. Процес взаємодії з системою для обох акторів починається з модуля «Авторизація», який забезпечує розмежування доступу та персоналізацію роботи з системою. Після успішної авторизації Користувач отримує доступ до основних функціональних можливостей системи, орієнтованих на надання актуальних рекомендацій для формування майбутнього прокату. Зважаючи на це, ключовим елементом взаємодії користувача з системою є модуль «Формування персонального списку фільмів», який включає в себе кілька підпорядкованих функцій: «Визначення критеріїв пошуку», що дозволяє користувачам налаштовувати параметри відбору фільмів за власними уподобаннями, та «Отримання рекомендацій до фільмів», що і здійснює

безпосередню видачу персоналізованих пропозицій. Крім цього, модуль «Ознайомлення з прем'єрами» розширює функціональність для формування персонального списку, додаючи можливість для формування свого списку фільмів на основі інформації про майбутні релізи.

Адміністратор системи має доступ до функцій управління контентом та налаштування алгоритмів рекомендацій. Основним прецедентом, з яким пов'язаний адміністратор є «Керування фільмами», який включає в себе наступні підфункції: «Додавання нових фільмів», «Оновлення даних» та «Видалення неактуальних фільмів». Ці операції є критично важливими для підтримання актуальності бази даних системи та забезпечення релевантності наданих рекомендацій. Окрім керування контентом, адміністратор має доступ до модуля «Керування механізмами рекомендацій», який безпосередньо відповідає за налаштування алгоритмічної складової системи. Цей модуль включає в себе підфункції «Оновлення рекомендаційних методів», яка дозволяє адаптувати та вдосконалювати алгоритми рекомендацій відповідно до нових вимог, результатів аналітики та змін у поведінці користувачів, а також «Підтримка якості рекомендацій», що забезпечує моніторинг ефективності системи та вжиття заходів для підвищення точності наданих рекомендацій.

Структура розробленої діаграми демонструє активну двонаправлену взаємодію, що лежить в основі рекомендаційної системи, де з одного боку, вона орієнтована на персоналізований досвід користувача, а з іншого – потребує активного управління та оновлення з боку адміністратора. Всі компоненти системи взаємопов'язані між собою, створюючи цим самим цілісний структурований підхід до процесу надання рекомендацій системою.

Перевагою застосування діаграми Use Case є її здатність до наочного опису складних інформаційних систем та їх функцій без необхідності занурюватися в технічні деталі, що робить діаграму прецедентів інформативною як для внутрішніх, так і для зовнішніх стейкхолдерів. Діаграма допомагає виявити й уточнити ключові потреби різних категорій користувачів, що має вирішальне значення для оптимізації функціональних процесів і налаштування взаємодії в рамках

платформи. Завдяки наочності діаграми, можна швидко виявити можливі прогалини або надлишкові функції, що може зменшити витрати часу на розробку і тестування, сприяючи таким чином підвищенню якості та ефективності роботи всієї системи.

Діаграма послідовностей – діаграма, на якій для деякого набору об'єктів на єдиній тимчасовій осі показані життєві цикли об'єктів (створення – діяльність - знищення) і взаємодія між ними (відправка запитів і отримання відповідей) [18]. Діаграма послідовності використовується для відображення взаємодій між об'єктами в послідовному порядку, в якому ці взаємодії відбуваються. Одним з основних застосувань діаграм послідовності є перехід від вимог, виражених як варіанти використання, до наступного та більш формального рівня уточнення. Варіанти використання часто уточнюються до однієї або кількох діаграм послідовності [3].

Діаграми послідовності є одним із найбільш інформативних інструментів проектування та документування систем, особливо у випадках, коли важливо відобразити взаємодію між компонентами в часі. Основною перевагою діаграм послідовності є їхня здатність детально відображати порядок обміну повідомленнями між різними компонентами системи, що дозволяє краще зрозуміти, який компонент ініціює певну дію, як передаються дані між модулями, та в якій послідовності відбуваються події. Крім цього, діаграми послідовності також відіграють важливу роль у виявленні потенційних проблем на ранніх стадіях розробки, адже допомагають ідентифікувати надмірні залежності між компонентами та знаходити логічні помилки в алгоритмах взаємодії. У контексті методології розробки програмного забезпечення, діаграми послідовності допомагають забезпечити відповідність реалізації початковому дизайну системи та полегшують процес рефакторингу, коли виникає потреба в оптимізації архітектури.

Зважаючи на велику кількість взаємодій, які можуть бути здійснені між користувачем та системою, було прийнято рішення розбити діаграму послідовностей на три діаграми, залежно від категорії взаємодії. Це рішення прийнято з метою підвищення рівня їх інформативності та читабельності. Всі

представлені діаграми складаються з трьох основних компонентів: Frontend (користувацький інтерфейс), серверний інтерфейс, що визначений конкретними блоками (Модуль авторизації, Модуль взаємодії з фільмами та Модуль рекомендацій) та Neo4j database (база даних Neo4j).

На рисунку 2.9 зображено діаграму послідовностей для взаємодії користувача з системою, де серверна частина передбачає фокус на взаємодії з модулем авторизації. Діями, які ініційовано користувачем, в межах даного блоку, є логін, авторизація та вихід з системи.

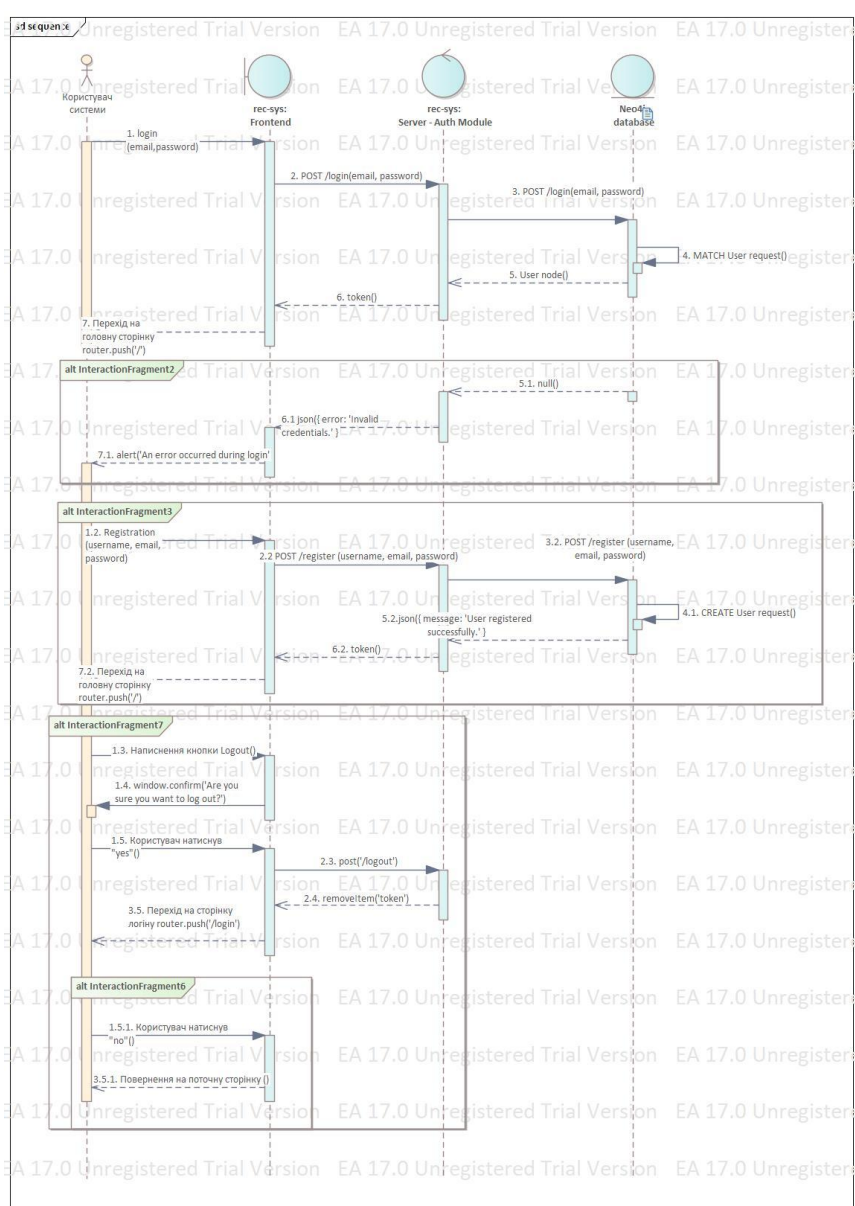


Рисунок 2.9 – Діаграма послідовності (модуль авторизації)

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.10 зображено діаграму послідовностей для взаємодії користувача з системою, де серверна частина передбачає фокус на взаємодії з фільмами. Діями, що ініційовано користувачем, в межах даного блоку є перехід на сторінку прем'єр, для ознайомлення з майбутніми релізами, перехід на сторінки фільмів та збереження фільму в персональний список збережень.

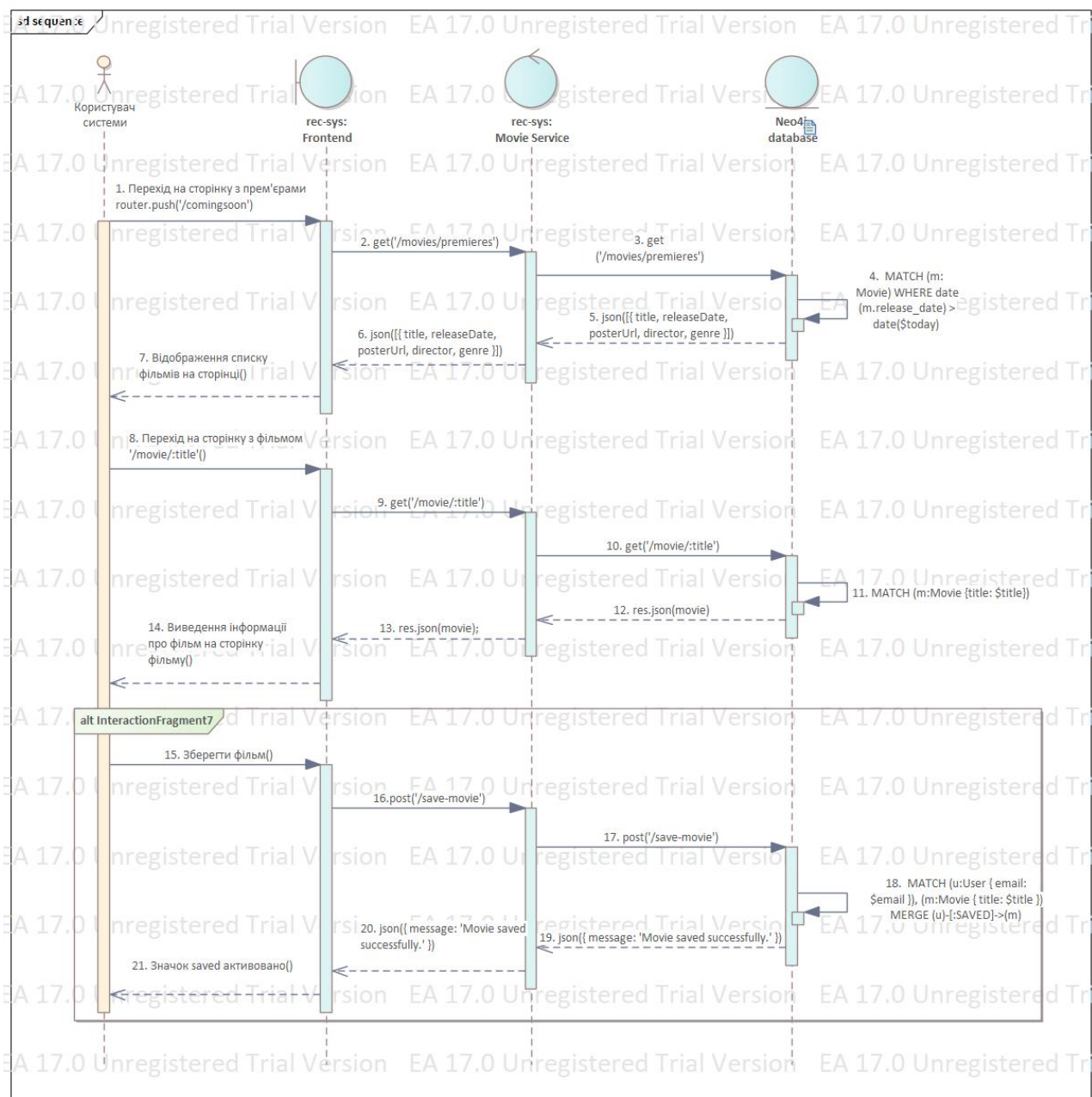


Рисунок 2.10 – Діаграма послідовності (модуль взаємодії з фільмами)

Джерело: сформовано автором на основі виконаного дослідження

На Рисунку 2.11 зображено діаграму послідовностей для взаємодії користувача з системою, де серверна частина передбачає фокус на взаємодії з рекомендаційним модулем. Діями, що ініційовані користувачем, в межах даного блоку є пошук рекомендацій за загальною подібністю між стрічками, заповнення фільтрів-критеріїв та обрання вибірки фільмів для отримання гібридної рекомендації.

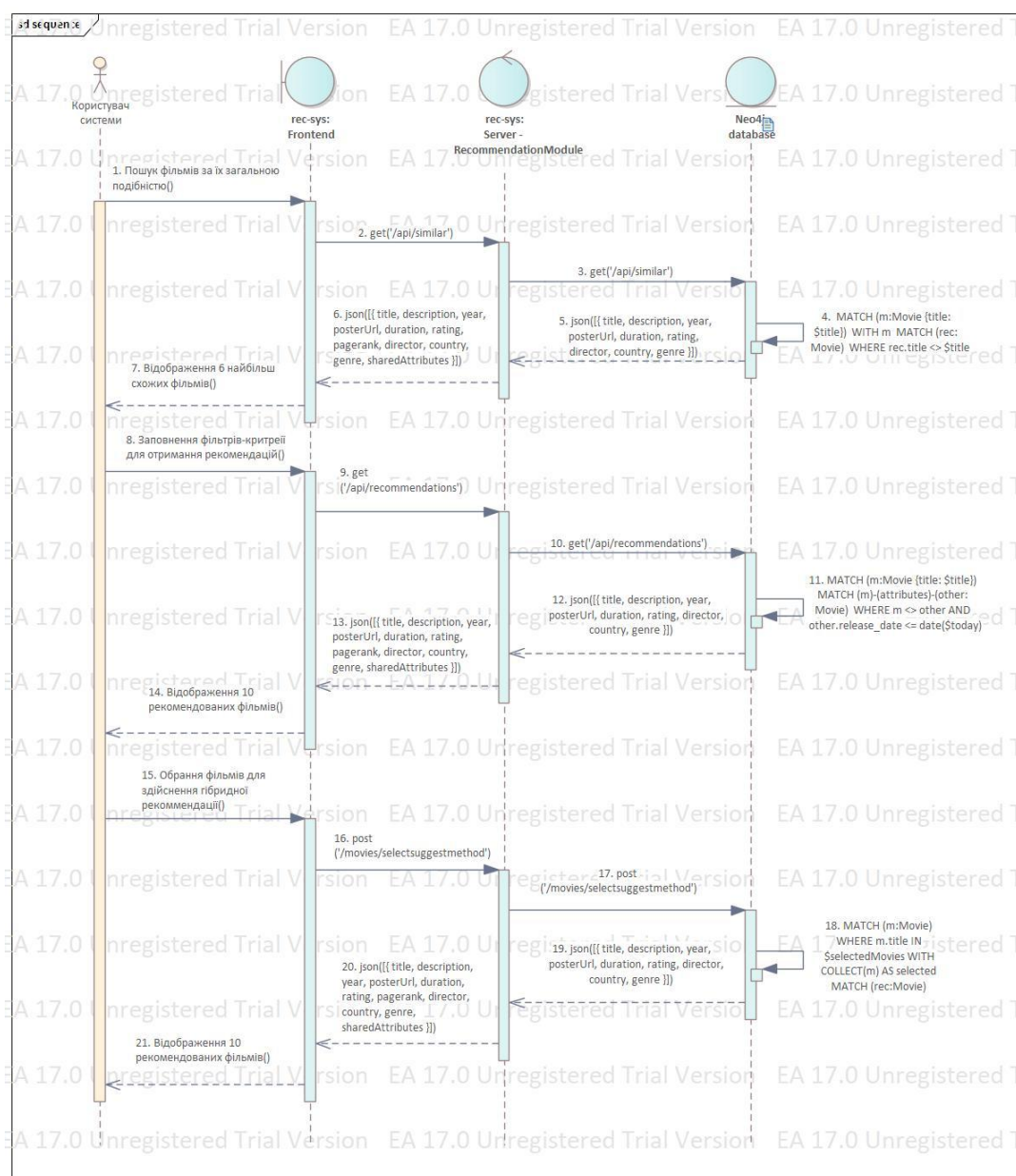


Рисунок 2.11 – Діаграма послідовності (рекомендаційний модуль)

Джерело: сформовано автором на основі виконаного дослідження

Таким чином, представлені у підрозділі діаграми дозволяють формалізувати функціонал рекомендаційної системи у форматі UML діаграм. Представлена діаграма прецедентів використана для визначення основних функціональних вимог до системи через опис взаємодії між акторами та системою в цілому, що дозволяє ідентифікувати ключові сценарії використання системи та встановити функціональні принципи роботи рекомендаційної системи. Діаграма послідовностей, у свою чергу, деталізує динамічну поведінку системи, показуючи хронологічний порядок взаємодії між об'єктами системи під час виконання конкретних прецедентів.

2.3.2 Моделювання структури системи

Моделювання структури є важливим етапом у проєктуванні системи, адже забезпечує чітке розуміння архітектурної організації та взаємозв'язків між компонентами системи. Структурне моделювання дозволяє визначити статичні аспекти системи, включаючи організацію класів, розподіл функціональності між модулями та ієрархічну декомпозицію системних компонентів.

Для всебічного відображення структури системи було розроблено три ключові діаграми: діаграму класів, яка демонструє об'єктну модель системи з деталізацією атрибутів та методів кожного класу, діаграму визначення блоків, що представляє архітектурну декомпозицію системи на функціональні модулі, та діаграму внутрішніх блоків, яка розкриває деталі внутрішньої організації та взаємодії компонентів усередині окремих блоків.

Діаграма класів UML це візуальна нотація, яка використовується для побудови та візуалізації об'єктно-орієнтованих систем. Діаграма класів в уніфікованій мові моделювання – це статична структурна діаграма, що демонструє властивості системи, класи, операції та зв'язки між об'єктами для опису структури системи [1]. Діаграма класів є потужним інструментом у розробці програмного забезпечення, який надає всебічний огляд структури інформаційної системи. Її

значення полягає в здатності представляти складні архітектурні рішення максимально наочно та структуровано.

Важливою перевагою діаграми класів є суттєве скорочення часу обслуговування програмного забезпечення. Маючи чітку візуальну модель взаємозв'язків між компонентами, розробники можуть набагато швидше розуміти структуру системи, що значно спрощує процеси налагодження, модифікації та підтримки коду. Діаграма класів також надає унікальну можливість деталізації та моделювання структур даних, надаючи змогу не просто побачити класи та їхні взаємозв'язки, але й зрозуміти внутрішню логіку побудови програмного забезпечення. Важливою перевагою є її універсальність, адже діаграма класів може бути корисною для структурних, поведінкових діаграм та діаграм взаємодії. Вона охоплює різні аспекти системи, даючи цілісне уявлення про її архітектуру.

UML діаграму класів для проєктованої системи представлено на рисунку 2.12.

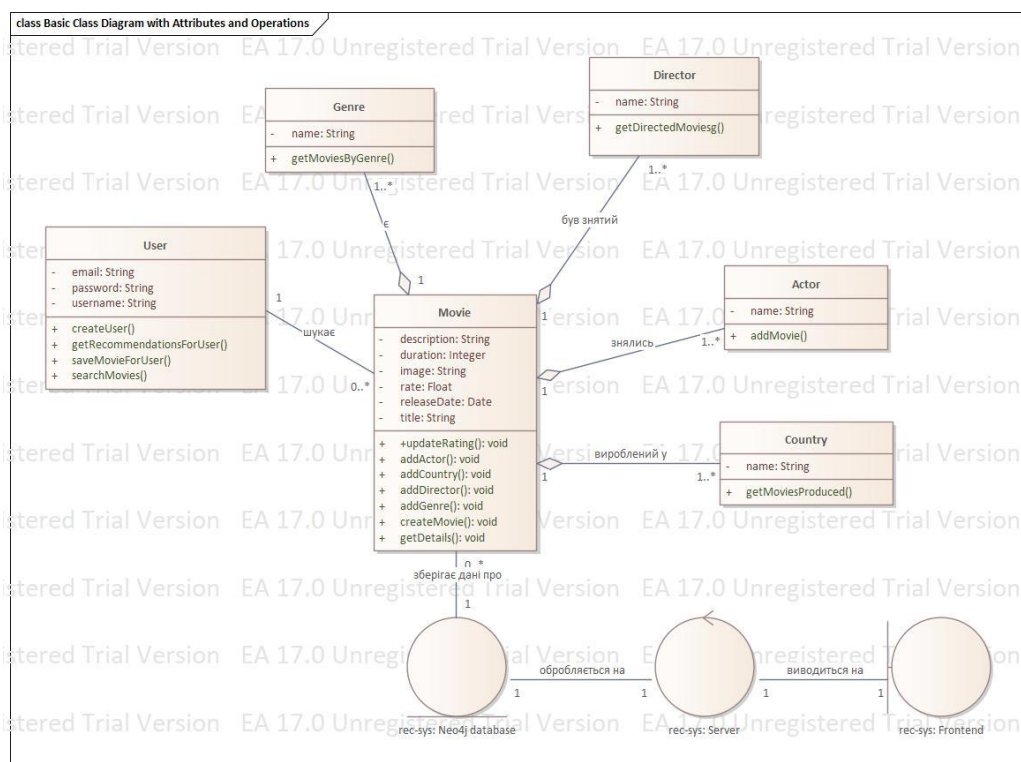


Рисунок 2.12 – Діаграма класів рекомендаційної системи

Джерело: сформовано автором на основі виконаного дослідження

Представлена діаграма класів демонструє архітектуру рекомендаційної системи для кінотеатру, відображаючи взаємозв'язки між ключовими компонентами системи.

Центральним елементом діаграми виступає клас Movie, який зберігає всю необхідну інформацію про фільми та взаємодіє з іншими класами системи. Цей клас містить базові характеристики фільму та методи для маніпуляції даними, включаючи додавання пов'язаних сутностей та отримання детальної інформації про фільм. Movie пов'язаний з класами Genre, Director, Actor та Country, що дозволяє створити повноцінну модель з усіма необхідними для цього зв'язками.

Клас User відповідає за управління даними користувачів системи, реалізуючи функціонал для створення облікових записів, формування персоналізованих рекомендацій, збереження фільмів для майбутнього перегляду та пошуку фільмів за різними критеріями. Зв'язок між User та Movie відображає можливість користувачів взаємодіяти з контентом системи.

Діаграма також демонструє класи Genre, Director, Actor та Country, які розширюють модель даних, додаючи важливі класифікаційні та описові елементи. Ці класи дозволяють групувати фільми за різними критеріями, що є критично важливим для рекомендаційної системи.

З точки зору класів, представлених у діаграмі, можна виділити:

Класи Movie, User, Genre, Director, Actor та Country відносяться до класів-сутностей, оскільки вони моделюють дані з довгим життєвим циклом, які зберігаються в базі даних Neo4j. Вони відображають реальні сутності предметної області кінематографу, їхні атрибути відповідають полям у базі даних, а методи – збереженим процедурам.

Клас Server є класом керування, який відповідає за координацію взаємодій між різними компонентами системи, обробку транзакцій з базою даних, розподіл ресурсів та логіку функціонування системи. Саме сервер здійснює виконання прецедентів системи, визначаючи послідовність дій для реалізації бізнес-логіки. Клас Frontend відноситься до граничних класів, розташованих на межі системи з користувачами, забезпечуючи інтерфейс для взаємодії. Він моделює інтерфейси

системи, дозволяючи користувачам здійснювати пошук фільмів, переглядати інформацію та взаємодіяти з системою рекомендацій.

Взаємозв'язки між класами відображені за допомогою з'єднань з відповідною множинністю, що визначає кількісні аспекти відношень. Наприклад, зв'язок між Movie та Actor з множинністю 1..* означає, що фільм може мати багато акторів, а кожен актор може зніматися в багатьох фільмах. Представлена діаграма класів охоплює всі необхідні аспекти моделювання статичної структури класів системи та їх взаємозв'язків відповідно до поставленого завдання.

Ключовою відмінністю BDD від звичайних діаграм класів є концепція блоків, що представляють собою модульні одиниці, які можуть описувати широкий спектр елементів – від фізичних пристроїв до абстрактних концепцій та алгоритмів. Кожен блок може мати власні структурні елементи, параметри, обмеження та операції, що робить цю нотацію надзвичайно гнучкою для моделювання складних систем [13].

На рисунку 2.13 наведено діаграму визначення блоків рекомендаційної системи, що відображає компоненти системи, їхній вміст, інтерфейси та зв'язки.

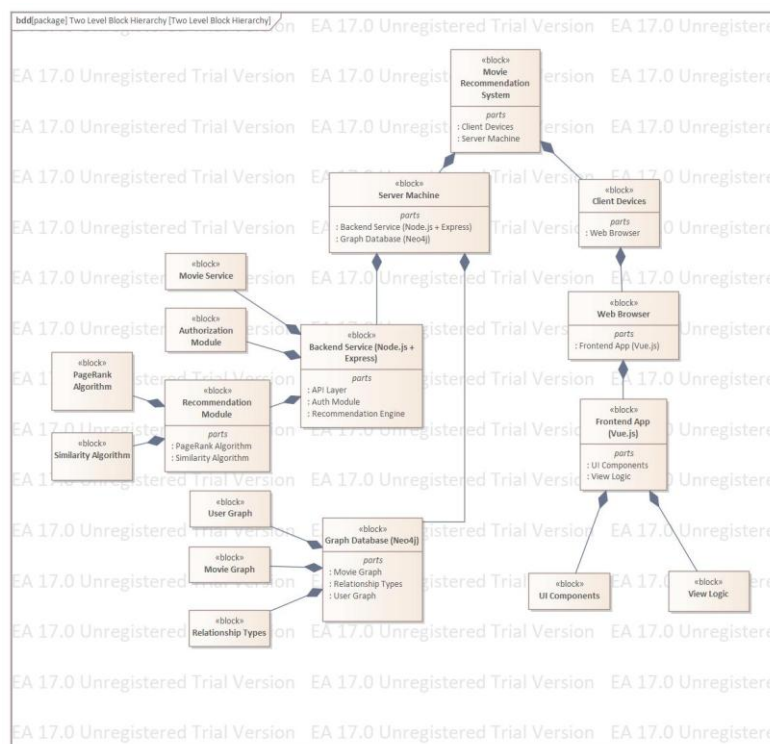


Рисунок 2.13 – Діаграма визначення блоків рекомендаційної системи

Джерело: сформовано автором на основі виконаного дослідження

У контексті рекомендаційної системи кінотеатру BDD забезпечує структуроване представлення архітектури системи через ієрархію взаємопов'язаних компонентів. Центральним елементом діаграми виступає блок «Система рекомендацій кінотеатру», який розгалужується на два основних структурних компоненти – «Сервер» і «Клієнтський пристрій», що відображає класичну клієнт-серверну архітектуру застосунку.

Блок «Сервер» представлений двома невід'ємними компонентами, що формують ядро серверної інфраструктури. Перший компонент – «Backend Service (Node.js+Express)» – реалізує програмну логіку обробки запитів та формування відповідей, використовуючи технологічний стек на основі Node.js із фреймворком Express. Другий компонент, «Графова БД (Neo4j)», забезпечує зберігання та обробку даних у вигляді графової структури, що є оптимальним рішенням для моделювання складних взаємозв'язків між користувачами, фільмами та їхніми властивостями.

«Backend Service» структурно розподілено на три функціональні модулі:

- «Модуль авторизації», який відповідає за процеси ідентифікації користувачів та управління сесіями;
- «Сервіс взаємодії з фільмами», що забезпечує операції з каталогом фільмів та пов'язаними даними;
- «Рекомендаційний модуль», що ж центральним елементом системи, реалізуючи алгоритми персоналізованих рекомендацій.

Рекомендаційний модуль, у свою чергу, інкапсулює два алгоритмічні підходи:

- «PageRank алгоритм», який використовує принципи ранжування вузлів у графі на основі їхньої важливості;
- «Similarity алгоритм», що здійснює обчислення схожості об'єктів за спільними характеристиками та користувацькими вподобаннями.

На клієнтській стороні блок «Клієнтський пристрій» містить компонент «Веб-браузер», який слугує платформою для взаємодії користувача з системою. Всередині веб-браузера функціонує «Frontend App (Vue.js)» – застосунок,

розроблений за допомогою фреймворку Vue.js. Цей фронтенд-застосунок структурно поділяється на:

- «UI компоненти», тобто візуальні елементи інтерфейсу, з якими безпосередньо взаємодіє користувач;
- «Логіка перегляду», що характеризує програмні компоненти, які керують відображенням контенту та обробкою взаємодій.

Така структурна декомпозиція діаграми наочно демонструє не лише компонентну будову системи, але й технологічні рішення, які лежать в основі блоків, формуючи цілісне уявлення про архітектуру рекомендаційної системи кінотеатру.

Діаграма внутрішніх блоків (IBD) є доповненням до попередньої діаграми, та концентрується на внутрішній структурі окремих блоків та їх взаємодії. IBD базується на концепції екземплярів блоків, портів та конекторів, що разом формують детальне представлення системної архітектури. Такий підхід дозволяє зрозуміти та наочно відобразити, як саме відбувається обмін інформацією при формуванні рекомендацій. На рисунку 2.14 зображено діаграму внутрішніх блоків для рекомендаційної системи кінотеатру.

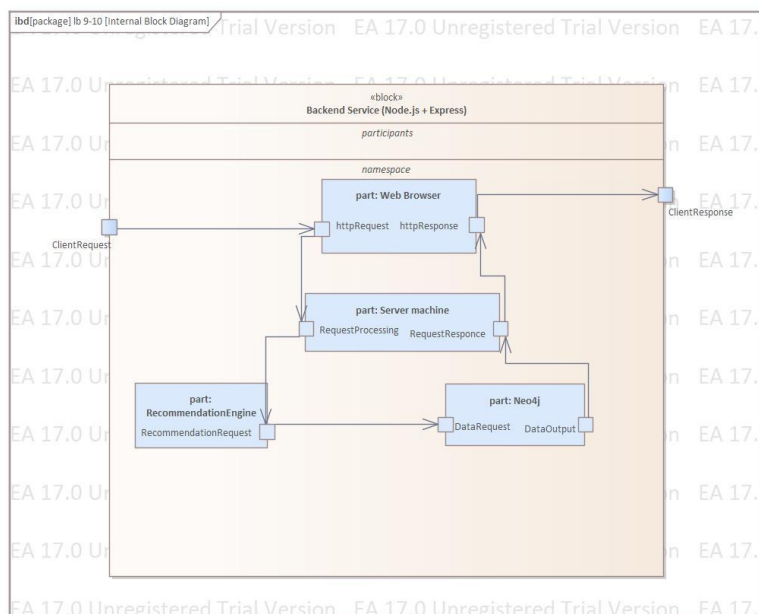


Рисунок 2.14 – Діаграма внутрішніх блоків рекомендаційної системи

Джерело: сформовано автором на основі виконаного дослідження

Загальна структура системи включає чотири основні компоненти, які взаємодіють між собою для забезпечення функціоналу рекомендацій. До них належать:

- Веб-браузер («Web Browser») – інтерфейсна частина, яка ініціює взаємодію через Запит клієнта («ClientRequest»), обробляє HTTP-запит («HttpRequest») і отримує HTTP-відповідь («HttpResponse»), формуючи кінцеву Відповідь («ClientResponse»), яку бачить користувач системи.
- Серверна машина («Server machine») – центральний компонент системи, який виконує обробку запитів («RequestProcessing») та формування відповідей («RequestResponse»).
- Рекомендаційний двигун («RecommendationEngine») – спеціалізований модуль, що отримує запит на рекомендації («RecommendationRequest») на основі даних користувача та фільмів, обробляє цей запит та взаємодіє з БД Neo4j для отримання інформації про фільми, рекомендація яких є найбільш актуальною.
- «Neo4j» – графова БД, що зберігає інформацію про фільми, користувачів та взаємозв'язки між ними. Обробляє Запит даних («DataRequest») від рекомендаційного двигуна та повертає Виведення даних («DataOutput») з необхідними даними.

Потік даних у системі організовано послідовно, адже запит користувача через браузер передається на сервер, який взаємодіє з рекомендаційним модулем для формування релевантних пропозицій. Рекомендаційний модуль запитує дані з Neo4j, отримує результати, які після обробки через сервер повертаються назад до браузера користувача.

Представлені у даному підрозділі діаграми формують цілісну та деталізовану модель структурної архітектури системи, яка охоплює всі рівні її організації від концептуального до імплементаційного.

2.3.3 Розподіл вимог за компонентами системи

Трасування вимог є важливим процесом у розробці програмного забезпечення, що забезпечує відстеження шляху реалізації кожної вимоги через усі етапи життєвого циклу проєкту. Використання такого підходу дозволяє контролювати повноту реалізації функціональності системи, виявляти потенційні невідповідності та оцінювати вплив змін на систему в цілому.

Діаграма трасування – це графічне представлення взаємозв'язків між різними об'єктами, що залучені у процес розробки програмного забезпечення, такими як вимоги, прецеденти використання, тестові сценарії, елементи користувацького інтерфейсу та інші компоненти системи. Вона забезпечує візуалізацію та документування зв'язків, що демонструють, як початкові вимоги до системи прослідковуються через усі етапи проєктування, розробки та тестування до кінцевої реалізації. Діаграма трасування дозволяє простежити походження кожного функціонального елементу системи та перевірити повноту реалізації вимог.

Використання діаграм трасування в контексті рекомендаційної системи кінотеатру є особливо важливим, оскільки дозволяє забезпечити цілісність та узгодженість всіх компонентів складної багаторівневої архітектури.

На рисунку 2.15 зображено діаграму трасування для рекомендаційної системи кінотеатру, що включає вимоги, прецеденти, варіанти тестування, елементи користувацького інтерфейсу та відповідні зв'язки між ними. Представлена діаграма трасування візуалізує складні взаємозалежності між об'єктами рекомендаційної системи, демонструючи, як кожна вимога пов'язана з відповідними прецедентами, тестовими сценаріями та елементами інтерфейсу.

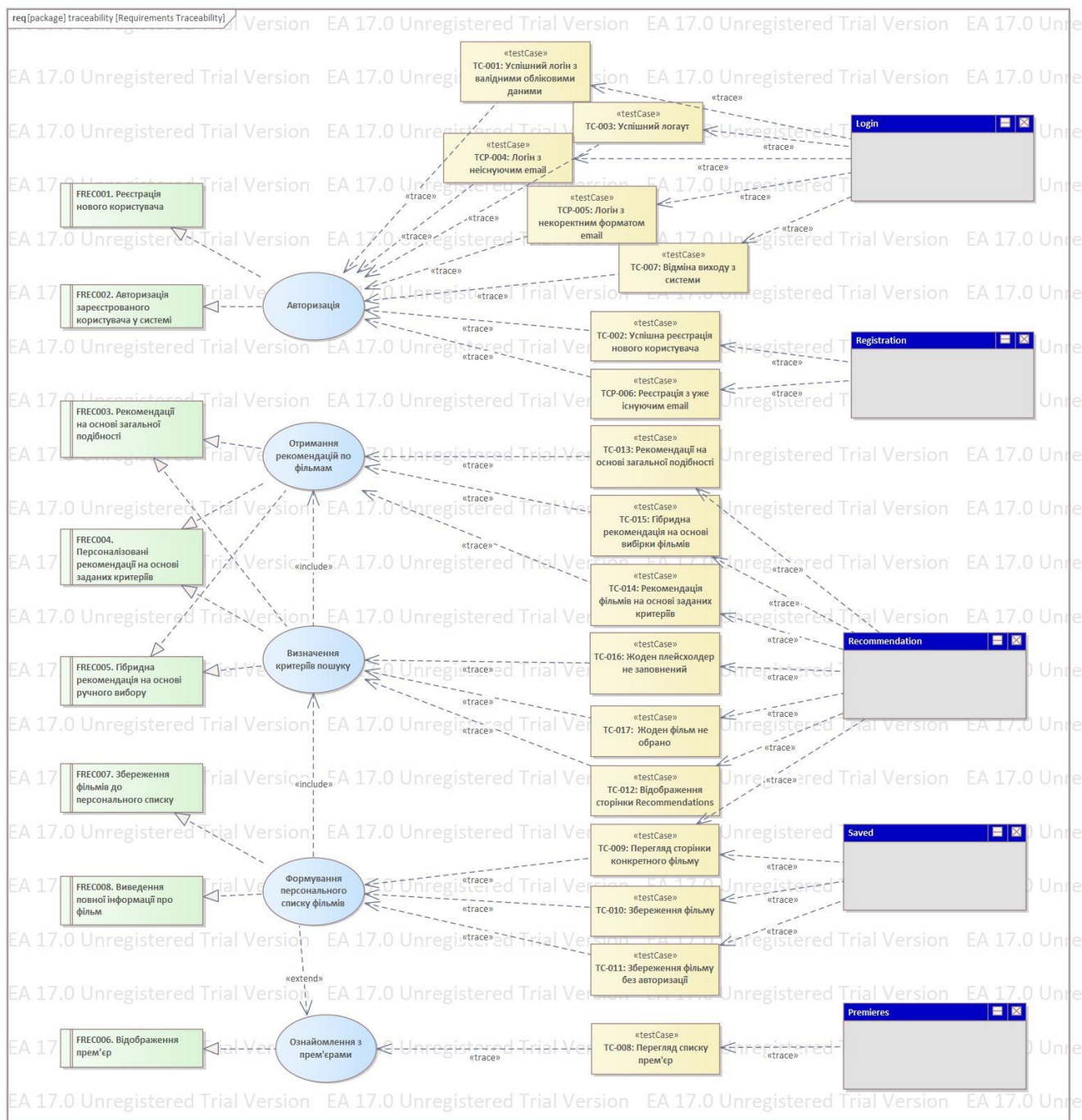


Рисунок 2.15 – Діаграма трасування для рекомендаційної системи кінотеатру

Джерело: сформовано автором на основі виконаного дослідження

Для формалізованого представлення взаємозв'язків між об'єктами діаграми трасування було створено матрицю взаємозв'язків.

Матриця взаємозв'язків (матриця трасування) – це табличне представлення взаємозалежностей між різними об'єктами системи, яке доповнює діаграму

трасування та забезпечує структурований формат для документування зв'язків між вимогами, прецедентами використання, елементами інтерфейсу, тестовими сценаріями та іншими компонентами системи. Матриця організована як двовимірна таблиця, де рядки та стовпці представляють різні типи об'єктів проекту, а на перетині відмічається наявність або відсутність зв'язку між ними, часто з додатковою інформацією про характер цього зв'язку.

Матриці взаємозв'язків виконують функцію деталізації та формалізації інформації, представленої на діаграмі трасування, пропонуючи більш точне та кількісне представлення зв'язків. Використання матриць трасування дозволяє ідентифікувати, які саме вимоги реалізуються через конкретні прецеденти використання; які елементи інтерфейсу відповідають за втілення певних функціональних вимог; які тестові сценарії перевіряють конкретні аспекти системи тощо. На відміну від діаграми трасування, яка надає візуальне представлення з акцентом на загальну структуру взаємозв'язків, матриці взаємозв'язків забезпечують більш деталізований та формалізований підхід до трасування.

У процесі розробки рекомендаційної системи кінотеатру матриці взаємозв'язків стають важливим інструментом для контролю якості та повноти реалізації, дозволяючи систематично перевіряти покриття кожної функціональної вимоги відповідними компонентами системи. Табличний формат представлення матриці взаємозв'язків забезпечує можливість швидкого виявлення прогалин у реалізації, коли певні вимоги залишаються без відповідних прецедентів використання або тестових сценаріїв. Матриці також дозволяють оцінити складність окремих компонентів системи через кількість пов'язаних з ними вимог та ідентифікувати критично важливі елементи, що потребують особливої уваги під час подальшої розробки.

На рисунку 2.16 наведено матрицю взаємозв'язків функціональних вимог та Use Cases системи. Ця матриця є фундаментальною для забезпечення повної відповідності системи всім вимогам, адже дозволяє виявити вимоги, які можуть бути пропущені або недостатньо реалізовані.

Source \ Target	Requirement Specification View::FREC001. Регістрація нового користувача	Requirement Specification View::FREC002. Авторизація зареєстрованого користувача у системі	Requirement Specification View::FREC003. Рекомендації на основі загальної подібності	Requirement Specification View::FREC004. Персоналізовані рекомендації на основі заданих критеріїв	Requirement Specification View::FREC005. Гібридна рекомендація на основі ручного вибору	Requirement Specification View::FREC006. Відображення прем'єр	Requirement Specification View::FREC007. Збереження фільмів до персонального списку	Requirement Specification View::FREC008. Виведення повної інформації про фільм	Two Level Requirement Hierarchy::FREC001. Регістрація нового користувача	Two Level Requirement Hierarchy::FREC002. Авторизація зареєстрованого користувача у системі	Two Level Requirement Hierarchy::FREC003. Рекомендації на основі загальної подібності	Two Level Requirement Hierarchy::FREC004. Рекомендації фільмів на основі заданих критеріїв	Two Level Requirement Hierarchy::FREC005. Гібридна рекомендація на основі ручного вибору	Two Level Requirement Hierarchy::FREC006. Відображення інформації про прем'єри	Two Level Requirement Hierarchy::FREC007. Зберігання фільмів до персонального списку	Two Level Requirement Hierarchy::FREC008. Виведення актуальної інформації про фільм
Use Cases::Авторизація	↑	↑							↑	↑						
Use Cases::Видалення неактуальних фільмів																
Use Cases::Визначення критеріїв пошуку			↑	↑	↑						↑	↑	↑			
Use Cases::Додавання нових фільмів																
Use Cases::Керування механізмами рекомендацій			↑	↑	↑											
Use Cases::Керування фільмами			↑	↑	↑		↑									
Use Cases::Ознайомлення з прем'єрами						↑								↑	↑	
Use Cases::Оновлення даних																
Use Cases::Оновлення рекомендаційних методів																
Use Cases::Отримання рекомендацій по фільмам			↑	↑	↑						↑	↑	↑			
Use Cases::Підтримка якості рекомендацій																
Use Cases::Формування персонального списку фільмів							↑	↑							↑	

Рисунок 2.16 – Матриця трасування вимог та UseCases

Джерело: сформовано автором на основі виконаного дослідження

Матриця взаємозв'язків TestCases та Екранів системи забезпечує зв'язок між тестовими сценаріями та конкретними екранами інтерфейсу рекомендаційної системи кінотеатру. Інтерфейс є основною точкою взаємодії користувача з системою рекомендацій, тому його інтерфейс має реалізовувати весь необхідний функціонал та теж покриватись тестовими сценаріями (рисунок 2.17).

Source \ Target	lab11...AdminPage	lab11...Login	lab11...Premieres	lab11...Recommendation	lab11...Registration	lab11...Saved
traceability::TC-001: Успішний логін з валідними обліковими даними		↑				
traceability::TC-002: Успішна реєстрація нового користувача					↑	
traceability::TC-003: Успішний логат		↑				
traceability::TC-007: Відміна виходу з системи		↑				
traceability::TC-008: Перегляд списку прем'єр			↑			
traceability::TC-009: Перегляд сторінки конкретного фільму				↑		↑
traceability::TC-010: Збереження фільму						↑
traceability::TC-011: Збереження фільму без авторизації						↑
traceability::TC-012: Додавання нового фільму	↑					
traceability::TC-013: Оновлення даних про фільм	↑					
traceability::TC-014: Видалення фільму	↑					
traceability::TC-015: Відображення сторінки Recommendations				↑		
traceability::TC-016: Рекомендації на основі загальної подібності				↑		
traceability::TC-017: Рекомендація фільмів на основі заданих критеріїв				↑		
traceability::TC-018: Гібридна рекомендація на основі вибірки фільмів				↑		
traceability::TC-019: Жоден плейсхолдер не заповнений				↑		
traceability::TC-020: Жоден фільм не обрано				↑		
traceability::TCP-004: Логін з неіснуючим email		↑				
traceability::TCP-005: Логін з некоректним форматом email		↑				
traceability::TCP-006: Реєстрація з уже існуючим email					↑	

Рисунок 2.17 – Матриця трасування TestCases та Екранів системи

Джерело: сформовано автором на основі виконаного дослідження

Матриця взаємозв'язків TestCases та UseCases дозволяє переконатись, що всі функціональні сценарії будуть належним чином протестовані, що є особливо важливим у контексті рекомендаційної системи, адже некоректні рекомендації можуть негативно вплинути на користувацький досвід. Відсутність належного рівня тестування системи може призвести до пропуску критичних помилок в основних функціональних можливостях системи, що безпосередньо впливає на її ефективність (рисунок 2.18).

Source \ Target	Target											
	Use Cases: Авторизація	Use Cases: Видалення неактуальних фільмів	Use Cases: Визначення критеріїв пошуку	Use Cases: Додавання нових фільмів	Use Cases: Керування механізми рекомендацій	Use Cases: Керування фільмами	Use Cases: Ознайомлення з прем'єрами	Use Cases: Оновлення даних	Use Cases: Оновлення рекомендаційних методів	Use Cases: Отримання рекомендацій по фільмам	Use Cases: Підтримка якості рекомендацій	Use Cases: Формування персонального списку фільмів
traceability::TC-001: Успішний логін з валідними обліковими даними	↑											
traceability::TC-002: Успішна реєстрація нового користувача	↑											
traceability::TC-003: Успішний логатут	↑											
traceability::TC-007: Відміна виходу з системи	↑											
traceability::TC-008: Перегляд списку прем'єр							↑					
traceability::TC-009: Перегляд сторінки конкретного фільму												↑
traceability::TC-010: Збереження фільму										↑		↑
traceability::TC-011: Збереження фільму без авторизації												↑
traceability::TC-012: Додавання нового фільму				↑		↑						
traceability::TC-013: Оновлення даних про фільм						↑		↑				
traceability::TC-014: Видалення фільму		↑				↑						
traceability::TC-015: Відображення сторінки Recommendations			↑									
traceability::TC-016: Рекомендації на основі загальної подібності					↑					↑		
traceability::TC-017: Рекомендація фільмів на основі заданих критеріїв					↑							
traceability::TC-018: Гібридна рекомендація на основі вибірки фільмів					↑					↑		
traceability::TC-019: Жоден плейсхолдер не заповнений			↑		↑							
traceability::TC-020: Жоден фільм не обрано			↑		↑							
traceability::TCP-004: Логін з неіснуючим email	↑											
traceability::TCP-005: Логін з некоректним форматом email	↑											
traceability::TCP-006: Реєстрація з уже існуючим email	↑											

Рисунок 2.18 – Матриця трасування TestCases та UseCases

Джерело: сформовано автором на основі виконаного дослідження

Реалізована діаграма трасування та матриці взаємозв'язків створюють цілісну формалізовану систему контролю відповідності рекомендаційної системи кінотеатру всім вимогам. У ході реалізації рекомендаційних систем, де алгоритми рекомендацій, користувацький інтерфейс та бізнес-логіка тісно переплетені, подібна формалізація у вигляді діаграми та матриць трасування дозволяє підтримувати цілісність системи та оперативно реагувати на зміни вимог, забезпечуючи високу якість системи.

Висновки до розділу 2

У другому розділі було здійснено комплексний аналіз вимог та моделювання системи рекомендацій для кінотеатру, що створила міцну теоретичну та проєктну основу для подальшої реалізації системи. Аналіз і специфікація вимог, що ґрунтувались на попередньому визначенні стейкхолдерів та їх бізнес-вимог, дозволили чітко визначити функціональні та нефункціональні характеристики майбутньої системи.

Постановка задачі забезпечила формалізацію досліджуваної області та визначення конкретних цілей системи, що включають надання персоналізованих рекомендацій фільмів на основі попередньо зібраної інформації про вподобання цільової аудиторії. Розроблений алгоритм розв'язання задачі базується на використанні графових структур даних, що дозволяє ефективно виявляти складні взаємозв'язки між фільмами та їхніми атрибутами. Обране математичне забезпечення здійснює точне формування релевантних рекомендацій та включає у себе ряд математичних моделей, формул та методів, що застосовуються для обробки даних про фільм, підрахунку подібності фільмів на основі кількості спільних атрибутів та ваги елементів. Формалізація алгоритму у вигляді блок-схем забезпечує розуміння логіки роботи програмної системи, що дозволяє структуровано подати етапи взаємодії користувача із системою, визначити послідовність дій і умови переходу між ними. Розподіл вимог за компонентами системи створив чіткий план реалізації, де кожен компонент системи отримав конкретні функціональні обов'язки, що забезпечують виконання кінцевих вимог користувачів. Проведена робота створила надійну методологічну базу для практичної реалізації рекомендаційної системи, забезпечивши детальне планування всіх аспектів системи від концептуального рівня до технічної специфікації компонентів. Розроблені моделі та специфікації вимог слугуватимуть основою для ефективної реалізації системи в наступних етапах проєкту, гарантуючи відповідність кінцевого продукту поставленим цілям та очікуванням користувачів.

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

3.1.1 Загальна характеристика інформаційного забезпечення

Інформаційне забезпечення є критичним компонентом будь-якої інформаційної системи, включаючи і рекомендаційні системи. Інформаційне забезпечення передбачає сукупність методів, засобів і процесів, які забезпечують збір, обробку, зберігання, передачу та використання інформації, необхідної для ефективного функціонування системи.

ІЗ рекомендаційної системи для кінотеатру орієнтоване на підтримку процесу прийняття рішень щодо репертуару фільмів, на основі аналізу вподобань цільової аудиторії кінотеатру. Зважаючи на це, ІЗ становить комплексну структуру даних та процедур, основою яких є довідкові, аналітичні та описові дані про фільми, їх жанри, режисерів, акторський склад, країну виробництва тощо.

3.1.2 Організація збору і передання первинної інформації

Збір первинної інформації здійснюється з відкритих джерел, якими є публічні бази даних фільмів. Прикладами такими платформ є IMDb [6], Rotten Tomatoes [11], JustWatch [7], TMDb [14]. Крім цього, джерелом первинної інформації можуть слугувати внутрішні каталоги дистриб'юторів. Дані зберігаються у системі у форматі графової структури, де кожен об'єкт системи представлений окремим вузлом, що пов'язані між собою зв'язками у вигляді ребер графа. У цьому контексті, база даних фільмів у системі виступає не просто сховищем даних, а і графом знань, що не лише зберігає інформацію про об'єкти, а і моделює семантику зв'язків між сутностями, відображаючи їх смислові відношення. Зважаючи на це,

граф знань дозволяє не лише відповідати на прямі запити про наявні дані (наприклад, «Знайти всі мультфільми»), але й здійснювати логічне виведення нових знань, аналізуючи приховані залежності між об'єктами системи. Такий підхід наближає розроблену рекомендаційну систему до онтологічного моделювання, де знання є логічно зв'язними фактами, що піддаються інтерпретації.

3.1.3 Структура інформаційних масивів

Структура інформаційних масивів рекомендаційної системи для кінотеатру визначається специфікою графової організації даних та необхідністю ефективного представлення складних взаємозв'язків між сутностями кінематографічної предметної області. Представлення масивів у форматі графової структури дозволяє природним чином моделювати реальні взаємозв'язки між фільмами, їх жанрами, режисерами, акторами та країнами, створюючи основу для інтелектуального аналізу кінематографічної інформації та генерації рекомендацій для планування репертуару кінотеатру. Опис структури інформаційних масивів для рекомендаційної системи кінотеатру наведено у таблиці 3.1.

Таблиця 3.1 – Структура сутностей (вузлів графі)

Масив	Атрибут	Ідентифікатор	Формат	Обов'язкове поле	Умова на значення	Логічні зв'язки
1	2	3	4	5	6	7
Movie	Назва фільму	title	String	так	унікальне	HAS_GENRE, DIRECTED_BY, PRODUCED_IN, STARS, SAVE

Продовження таблиці 3.1

1	2	3	4	5	6	7
Movie	Дата релізу	release_date	Date	так	$1990 \leq \text{дата} \leq 2050$	
	Опис	description	Char	ні	≤ 1500 символів	
	Рейтинг	rate	Float	ні	від 0 до 10	
	Тривалість	duration	Int	так	> 0	
	Зображення	image	URL	ні	-	
User	Електронна адреса	email	String	так	унікальне, валідна email-адреса	
	Ім'я користувача	username	String	так	унікальне	SAVE
	Пароль	password	String	ні	-	
Genre	Назва жанру	name	String	так	унікальне	HAS_GENRE
Country	Назва країни	name	String	так	унікальне	PRODUCED_IN
Actor	Ім'я актора	name	String	так	унікальне	STARS
Director	Ім'я режисера	name	String	так	унікальне	DIRECTED_BY

Джерело: сформовано автором на основі виконаного дослідження

3.1.4 Вибір СКБД

Реалізація системи відбувається з використанням графової СКБД Neo4j. Однією з переваг обраної СКБД є мова запитів Cypher з читабельним синтаксисом, що подібний до природної мови. У порівнянні з іншими графовими СКБД, які

використовують менш дружні до користувача мови, як-от Gremlin або власні DSL (Domain-Specific Languages), Cypher вирізняється простотою та зручністю. У результаті, вона значно пришвидшує розробку, тестування і підтримку запитів, що є особливо важливим у проектах з інтенсивною аналітикою. Важливою перевагою Neo4j у контексті реалізації рекомендаційних систем є наявність вбудованої бібліотеки Graph Data Science, що надає доступ до великої кількості графових алгоритмів, серед яких PageRank, Node Similarity, Community Detection тощо. Усі алгоритми є оптимізованими до масштабованості системи та здатні ефективно обробляти великі графи з великою кількістю вузлів та зв'язків.

3.1.5 Інфологічна модель бази (сховища) даних

Важливим етапом на етапі вибору інформаційного забезпечення є створення структурованої інфологічної моделі системи, для якої буде розроблятися БД. Основною складовою інфологічної моделі є атрибути, які потрібно проаналізувати і певним чином згрупувати для подальшого зберігання в БД.

Суть інфологічного моделювання полягає у виділенні інформаційних об'єктів (сутностей) системи, які підлягають зберіганню в БД, а також у визначенні характеристик об'єктів і взаємозв'язків між ними. Інформаційний об'єкт – це деяка сутність, яку необхідно зберігати в БД з точки зору прикладної програми чи користувача. Кожний об'єкт описується його властивостями. Властивості об'єктів описують атрибути, що входять до їх складу. Атрибут являє собою елементарну, логічно неподільну, поіменовану одиницю інформації [20].

Інфологічну модель бази даних для рекомендаційної системи кінотеатру наведено на рисунку 3.1.

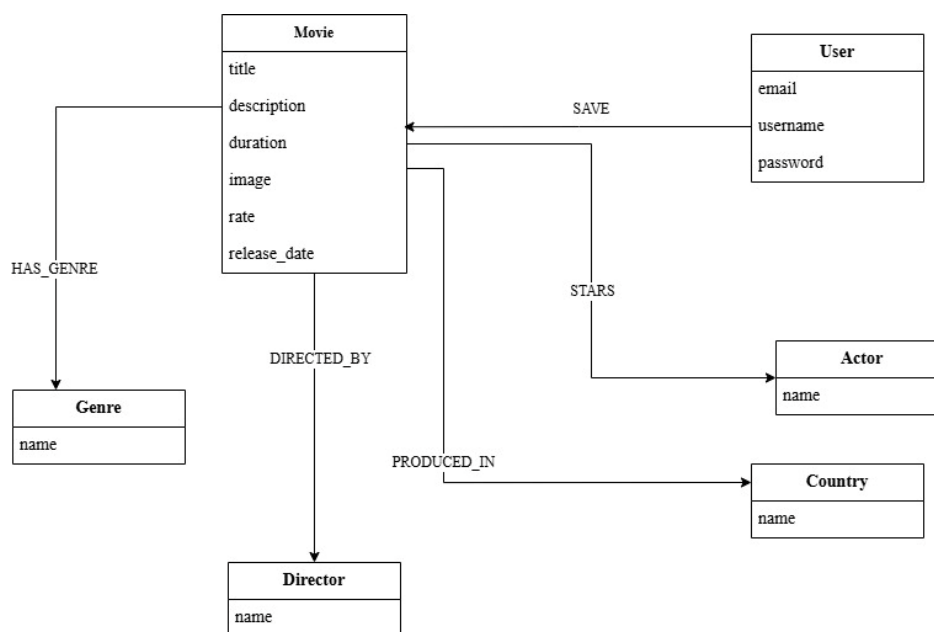


Рисунок 3.1 – Інфологічна модель рекомендаційної системи

Джерело: сформовано автором на основі виконаного дослідження

Представлена на рисунку 3.1 інфологічна модель охоплює ключові сутності-вузли, серед яких Movie, User, Genre, Actor, Director, Country і взаємозв'язки між ними, які у графовій моделі реалізуються у вигляді ребер.

Центральним об'єктом моделі є сутність-вузол фільму (Movie), який характеризується рядом атрибутів, серед яких його назва, опис, тривалість, зображення, рейтинг та дата релізу. Сутність-вузол Movie пов'язана з іншими вузлами за допомогою типізованих зв'язків:

- HAS_GENRE, що веде до вузла жанру (Genre)
- DIRECTED_BY, що зв'язує з вузлом режисера (Director)
- PRODUCED_IN, що вказує на країну виробництва фільму (Country)

Сутність-вузол актора (Actor) пов'язана з фільмами через зв'язок STARS та дозволяє враховувати подібність фільмів за акторським складом під час формування рекомендацій.

Сутність-вузол User має зв'язок SAVE, який відображає фільми збережені користувачем до їх персонального списку.

Описана модель охоплює всі необхідні атрибути для кожної сутності та відповідає принципу нормалізації графових моделей, де основне зосередження здійснюється на семантиці об'єктів та зв'язків між ними.

3.1.6 Даталогічна модель бази (сховища)даних

На основі інфологічної моделі бази даних у обраній СКБД реалізовано її даталогічну модель. На етапі даталогічного проєктування здійснюється перехід від інфологічної моделі ПО до логічної (даталогічної) моделі, що підтримується засобами конкретної СКБД. Даталогічна модель являє собою базу даних, структуровану на логічному рівні та орієнтовану на конкретну СКБД [19].

Даталогічну модель бази даних для рекомендаційної системи кінотеатру представлено на рисунку 3.2.



Рисунок 3.2 – Даталогічна модель рекомендаційної системи

Джерело: сформовано автором на основі виконаного дослідження

Представлена даталогічна модель описує, як інформація з інфологічної моделі представлена безпосередньо у СКБД Neo4j, з урахуванням її логічної структури та зв'язків між сутностями.

Структура даталогічної моделі системи розроблена на прикладі фільму «Дикий робот», сутність якого розміщена в центрі схеми та відображена у вигляді фіолетового кола.

Країнами, які брали участь у виробленні фільму є Японія та США, що відображені у вигляді червоних кіл. Між назвою фільму та країною походження передбачено формування зв'язку PRODUCED_IN, що направлений від сутності фільму до сутності країни.

Жанри, до яких належить фільму (у межах цього прикладу – Наукова фантастика та Мультфільм) зображено блакитними колами. Між назвою фільму та його жанром передбачено формування зв'язку HAS_GENRE, що направлений від сутності фільму до сутності жанру.

Режисери, що працювали над фільмом зображено помаранчевими колами (у межах цього прикладу режисером стрічки є Кріс Сандерс). Між назвою фільму та режисером передбачено формування зв'язку DIRECTED_BY, що направлений від сутності фільму до сутності режисеру.

Актори, що знімались у фільмі зображено у вигляді кола світло-коричневого кольору. Між назвою фільму та акторами передбачено формування зв'язку STARS, що направлений від сутності фільму до сутності актора.

Користувачі системи відображено у форматі кола зеленого кольору. Між користувачем та актором передбачено формування зв'язку SAVE, що направлений від сутності користувача до сутності фільму.

Приклади Cypher-запитів для завантаження системи наведено у Додатку А.

3.2 Технічне забезпечення

3.2.1 Загальні положення та схема автоматизації

Технічне забезпечення є невід’ємною складовою частиною інформаційної системи, що забезпечує її коректну, стабільну та продуктивну роботу. Воно охоплює всі апаратні засоби, необхідні для функціонування платформи, обробки даних, зберігання інформації, забезпечення мережевого підключення, а також захисту даних і системи в цілому. Основна мета технічного забезпечення – створення надійної, масштабованої та високопродуктивної інфраструктури, яка зможе обробляти великі обсяги даних і забезпечувати стабільний доступ користувачів до всіх функцій додатку. Розглянуте обладнання підібране з урахуванням специфіки платформи, вимог до обробки запитів, захисту даних тощо.

Реалізована рекомендаційна система для кінотеатру передбачає наявність відданого серверу, що включає її ядро та базу даних. Доступ користувачів до функціоналу системи здійснюється шляхом використання сайту системи через веб-браузер за умови доступу до мережі Інтернет. Взаємодія всіх описаних елементів представлена на схемі автоматизації для рекомендаційної системи кінотеатру (рисунок 3.3).

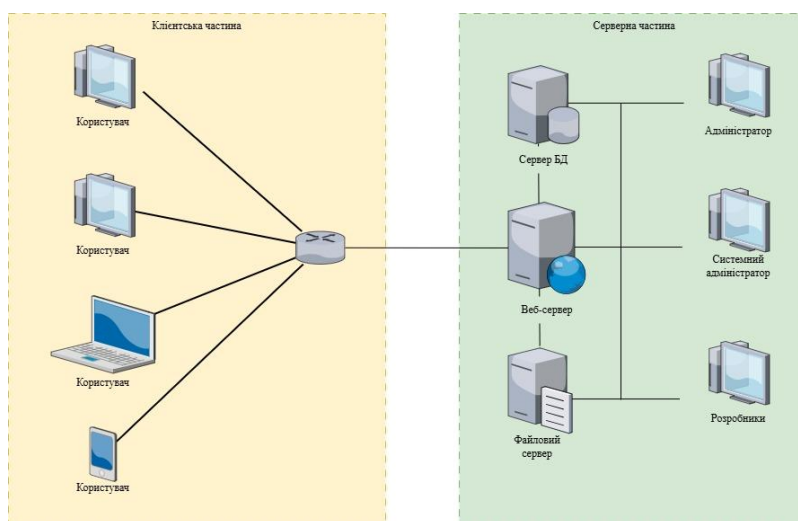


Рисунок 3.3 – Схема автоматизації рекомендаційної системи

Джерело: сформовано автором на основі виконаного дослідження

3.2.2 Структура комплексу технічних засобів

Реалізація рекомендаційної системи є складним технічним завданням, яке вимагає значних обчислювальних ресурсів для забезпечення високої швидкості обробки великих обсягів даних, таких як рейтинги, уподобання користувачів та характеристики фільмів. Нижче наведено характеристики технічного обладнання та обґрунтування його використання у контексті проєктованої системи (таблиця 3.2).

Таблиця 3.2 – Технічне обладнання для реалізації рекомендаційної системи

Категорія	Елемент	Характеристики	Обґрунтування вибору
1	2	3	4
Серверне обладнання	Сервер баз даних	Процесор: Intel XEON Gold	Висока продуктивність, підтримка багатопотокової обробки даних, швидкий доступ до великих обсягів інформації та можливість обслуговування численних запитів одночасно.
	Система резервного копіювання	Накопичувач: NAS (Network Attached Storage), 5 TB, RAID 5	Захист даних завдяки резервуванню. RAID 5 забезпечує відмовостійкість і дозволяє відновлювати дані у разі збою одного з дисків.
Мережева інфраструктура	Маршрутизатор	Модель: Lancom 1800EFW	Підтримує високошвидкісний інтернет, забезпечує безпеку з'єднань завдяки вбудованому брандмауеру
	Комутатор	Модель: Cisco CBS220-24P-4G-EU	Комутатор із підтримкою Gigabit Ethernet для швидкої передачі даних, PoE (Power over Ethernet) для живлення підключених пристроїв.

Продовження таблиці 3.1

1	2	3	4
Мережева інфраструктура	Канал зв'язку	Тип: Оптоволоконний інтернет, швидкість 1 Гбіт/с	Висока пропускна здатність дозволяє мінімізувати затримки, забезпечуючи швидкий доступ до серверів і стабільну роботу системи.
Клієнтське обладнання	Персональні комп'ютери/ноутбук и	ОС: Windows 7+, macOS 10.13+, Linux з ядром 4.x+	Ці характеристики забезпечують сумісність із системою та гарантують коректну роботу програмного забезпечення
		Процесор: Intel Pentium 4 / AMD Athlon 64 або новіший	
		ОЗП: 2 ГБ (рекомендовано 4 ГБ)	
		Пам'ять: 500 МБ вільного місця	
		Дисплей: роздільна здатність від 1024x768	
	Мобільні пристрої	ОС: Android 10+ або iOS 12+	
Інші засоби	Система безперебійного живлення (UPS)	Модель: APC Smart-UPS, 2200 ВА	Гарантує безперервну роботу серверів і мережевого обладнання у випадку перебоїв з електропостачанням, запобігаючи втраті даних і пошкодженню обладнання.

Джерело: сформовано автором на основі виконаного дослідження

Описане технічне забезпечення демонструє комплексний підхід до створення інфраструктури, що здатна витримувати високі навантаження та забезпечувати

надійність і безперервність роботи системи. Серверне обладнання, представлене потужним процесором Intel XEON Gold та значним обсягом оперативної пам'яті 64 GB DDR4, створює надійне підґрунтя для обробки масивів даних, забезпечуючи високошвидкісну багатопотокову обробку інформації та високу швидкість відклику системи. Система резервного копіювання на базі NAS з конфігурацією RAID 5 постає критично важливим елементом технічного забезпечення, оскільки унеможливорює втрату важливої інформації та гарантує максимальну відмовостійкість інформаційної системи. Такий підхід до зберігання даних є особливо актуальним у контексті рекомендаційної системи, де цілісність та збереженість всіх об'єктів та зв'язків між ними набуває важливого значення.

Мережева інфраструктура, представлена професійним обладнанням Lanscom та Cisco, демонструє прагнення до максимальної швидкодії та безпеки комунікацій. Використання оптоволоконного інтернету зі швидкістю 1 Гбіт/с створює технологічне середовище, здатне підтримувати складні взаємодії з системою.

Система першочергово розроблена для використання на персональних комп'ютерах та ноутбуках, забезпечуючи оптимальну функціональність при відносно невибагливих технічних характеристиках (Windows 7+, macOS 10.13+, Linux з ядром 4.x+, процесор рівня Intel Pentium 4 або вище, від 2 ГБ оперативної пам'яті та лише 500 МБ вільного місця на диску). Це дозволяє охопити максимально широке коло користувачів, включаючи власників застарілих моделей комп'ютерів. Водночас, завдяки впровадженню веб-технологій та крос-платформного підходу, система демонструє високу адаптивність до різних пристроїв – користувачі мобільних телефонів та планшетів з Android 10+ та iOS 12+ можуть повноцінно працювати з рекомендаційною системою платформою через браузер, що забезпечує високий рівень її доступності.

Система безперебійного живлення APC Smart-UPS з потужністю 2200 ВА є додатковим страховим механізмом, який гарантує стабільність роботи всієї інформаційної системи навіть у разі непередбачуваних електротехнічних форс-мажорів, що є особливо важливим для запобігання будь-яким технічним збоїв.

3.2.3 Опис автоматизованого робочого місця

Автоматизоване робоче місце (АРМ) працівника, який забезпечує підтримку та управління рекомендаційною системою кінотеатру, повинно бути оснащено сучасними технічними засобами, які забезпечують стабільну роботу, високу продуктивність і зручність адміністрування. Основним елементом є персональний комп'ютер із продуктивним багатоядерним процесором (наприклад, Intel Core i5, AMD Ryzen 7 або еквівалентні), оперативною пам'яттю обсягом не менше 16 ГБ та SSD-накопичувачем об'ємом від 512 ГБ для швидкого доступу до даних. Обов'язковою є наявність високоякісного монітора з роздільною здатністю не нижче 1920x1080 для комфортної роботи з веб-інтерфейсами та графами. Підключення до мережі Інтернет має бути стабільним, з пропускнуою здатністю не менше 50 Мбіт/с, для взаємодії з серверною частиною системи, перевірки оновлень та здійснення контроль за станом сервісів.

Кількість оснащених АРМ залежить безпосередньо від кількості працівників, що здійснюють управління системою, де на кожного працівника має бути відведене окреме АРМ.

3.3 Програмне забезпечення

3.3.1 Структура програмного забезпечення

Програмне забезпечення системи складається з двох основних категорій: системного та прикладного програмного забезпечення, які забезпечують ефективність у процесі розробки системи.

Системне програмне забезпечення формує базову платформу для роботи всіх компонентів системи. До цієї категорії належить операційна система Windows 11, яка забезпечує управління апаратними ресурсами комп'ютера та надає інтерфейс для взаємодії з користувачем. Крім цього, системне ПЗ також включає у себе

технічний стек для розробки проєкту, системи контролю версій, середовище розробки, СКБД, засоби прототипування тощо.

До прикладного програмного забезпечення належать ПЗ загального призначення, RAD-засоби, та програми та здійснення тестування функціональності системи. Прикладне ПЗ потребує попереднього створення текстових сценаріїв.

Сценарії – це наративні описи взаємодій між користувачами та системою. Конкретність сценаріїв допомагає користувачам та розробникам виробити спільне розуміння функціональності запропонованої системи; проте збільшення рівня конкретності призводить до потенційно необмеженої кількості сценаріїв для системи.

Сценарії отримують з опису цілей системи та користувача, а також потенційних перешкод, що блокують ці цілі. Таким чином, сценарії уточнюють дизайн-мислення кількома способами:

- Стейкхолдери можуть порівнювати конкуруючі системні пропозиції, які відрізняються ступенем автоматизації або стилем взаємодії користувача з системою, розглядаючи основні сценарії;
- Якщо мережа цілей передбачає, що ціль може бути досягнута кількома різними способами, сценарії можуть бути корисними для порівняння альтернативних варіантів дій;
- Сценарії корисні для обмірковування наслідків перешкод, що блокують досягнення цілей, та можливих механізмів захисту від перешкод або пом'якшення їхнього впливу, коли вони виникають [10].

Нижче наведено сценарії для рекомендаційної системи кінотеатру, що формалізовані у вигляді таблиць 3.3-3.7.

Таблиця 3.3 – Сценарій реєстрації нового користувача

Назва сценарію	Реєстрація нового користувача
Учасники процесу	Користувач, Адміністратор системи, Система

Продовження таблиці 3.3

Цілі	1. Зареєструвати новий обліковий запис для доступу до системи; 2. Перевірити введені дані, переконатися у відсутності дублюючих облікових записів
Сценарій успішний	1. Користувач відкриває сторінку реєстрації 2. Користувач вводить обов'язкові поля: username, email, password та натискає «Зареєструватися» 3. Користувач потрапляє на головну сторінку
Результат	Користувач успішно зареєстрований; доступ до сторінок системи надано
Розширення	1а) Не всі поля заповнені Результат: система повертає повідомлення про помилку
	2а) Користувач з таким username або email вже існує Результат: система повертає повідомлення про помилку
	3а) Помилка під час роботи БД або внутрішня помилка сервера Результат: система повертає повідомлення про помилку

Джерело: сформовано автором на основі виконаного дослідження

Таблиця 3.4 – Сценарій авторизації користувача

Назва сценарію	Авторизація користувача
Учасники процесу	Користувач, Адміністратор системи, Система
Цілі	1. Увійти до системи для доступу до сторінок з обмеженим доступом для незареєстрованих користувачів; 2. Ідентифікувати користувача та надати токен для авторизованого доступу
Сценарій успішний	1. Користувач відкриває сторінку входу 2. Користувач вводить валідні email та password та натискає «Увійти» 3. Користувач потрапляє на головну сторінку
Результат	Користувач авторизований; токен надано

Продовження таблиці 3.4

Розширення	1a) Не всі поля заповнені Результат: система повертає повідомлення про помилку
	2a) У базі даних немає користувача з введеним email Результат: система повертає повідомлення про помилку
	3a) Неправильний пароль Результат: система повертає повідомлення про помилку
	4a) Помилка під час роботи БД або внутрішня помилка сервера Результат: система повертає повідомлення про помилку

Джерело: сформовано автором на основі виконаного дослідження

Таблиця 3.5 – Сценарій отримання списку прем'єр

Назва сценарію	Отримання списку прем'єр фільмів
Учасники процесу	Користувач, Система
Цілі	1. Отримання списку майбутніх прем'єр фільмів, що вийдуть після сьогоднішньої дати 2. Знайти у базі даних фільми, що ще не вийшли у прокат
Сценарій успішний	1. Користувач відкриває сторінку "Прем'єри" 2. Користувач ознайомлюється з переліком майбутніх прем'єр
Результат	Користувач отримує список прем'єр
Розширення	1a) У базі даних немає фільмів з майбутньою датою релізу Результат: система повертає порожній список. Користувач бачить повідомлення, що найближчим часом прем'єр не заплановано
	2a) Помилка під час роботи БД або внутрішня помилка сервера Результат: система повертає повідомлення про помилку
	3a) Деякі дані про фільми відсутні Результат: система підставляє значення «Unknown»

Джерело: сформовано автором на основі виконаного дослідження

Таблиця 3.6 – Сценарій збереження фільму користувачем

Назва сценарію	Збереження фільму користувачем у список обраних
Учасники процесу	Користувач, Система
Цілі	1. Додати вибраний фільм до свого списку збережених 2. Створити зв'язок між користувачем і вибраним фільмом у БД
Сценарій успішний	1. Користувач переглядає сторінку з карткою фільму 2. Користувач натискає на іконку «Зберегти» на картці обраного фільму
Результат	Фільм додано до списку збережень користувача
Розширення	1а) Помилка під час роботи БД або внутрішня помилка сервера Результат: система повертає повідомлення про помилку

Джерело: сформовано автором на основі виконаного дослідження

Таблиця 3.7 – Сценарій отримання рекомендацій

Назва сценарію	Отримання рекомендації зі схожими фільмами
Учасники процесу	Користувач, Система
Цілі	1. Отримати список фільмів, схожих на обраний 2. Знайти фільми, що мають найбільше спільних атрибутів з заданим фільмом
Сценарій успішний	1. Користувач переглядає сторінку з карткою фільму 2. Користувач натискає на кнопку «Get Similar» на картці обраного фільму.
Результат	Користувач отримав список рекомендованих фільмів
Розширення	1а) Назву фільму не передано Результат: система повертає повідомлення про помилку
	2а) Помилка під час роботи БД або внутрішня помилка сервера Результат: система повертає повідомлення про помилку

Джерело: сформовано автором на основі виконаного дослідження

3.3.2 Системне програмне забезпечення

Для забезпечення функціонування платформи було обрано комплекс програмних інструментів, що охоплює всі необхідні аспекти для розробки сучасних

рекомендаційних систем. Вибір системного ПЗ базується на принципах сумісності, продуктивності та зручності використання, що дозволяє створити стабільну та ефективну платформу для розробки. Представлена сукупність програм забезпечує повний цикл роботи над проектом.

Системне ПЗ для розроблюваної системи включає:

1. Операційна система: Windows 11 – операційна система Microsoft, що забезпечує стабільну платформу для роботи всіх компонентів системи розробки.
2. Засоби захисту: Windows Defender – вбудований антивірус Windows, що забезпечує захист системи від шкідливого програмного забезпечення, веб-загроз та несанкціонованого доступу в режимі реального часу.
3. Середовище розробки та інструменти версіонування: Visual Studio Code – редактор коду з підтримкою розширень, інтелектуального автодоповнення та інтегрованого терміналу. GitHub – розподілена система контролю версій, що забезпечує зберігання коду, відстеження змін та управління гілками проекту.
4. Система керування базами даних: Neo4j Desktop – графова СКБД з графічним інтерфейсом, що дозволяє створювати, адмініструвати та візуалізувати графові бази даних. Включає вбудовані засоби моніторингу, Browser для виконання Cypher-запитів та плагіни для розширення функціональності.
5. Засоби проектування та прототипування: Figma – хмарна платформа для UI/UX дизайну, що забезпечує створення інтерактивних прототипів, системи дизайну та автоматичної генерації CSS-коду.
6. Технологічний стек розробки: Vue.js – JavaScript-фреймворк для створення користувацьких інтерфейсів з реактивною архітектурою та компонентним підходом. Node.js + Express – серверне середовище виконання JavaScript та мінімалістичний веб-фреймворк для розробки RESTful API та веб-додатків. HTML/CSS - фундаментальні технології веб-розробки для створення користувацького інтерфейсу.

3.3.3 Прикладне програмне забезпечення

Рекомендаційна система для кінотеатру потребує відповідного прикладного програмного забезпечення, яке забезпечує підтримку всіх етапів життєвого циклу проекту. Вибір прикладного ПЗ ґрунтується на принципах функціональної повноти, сумісності та ефективності. Прикладне ПЗ для розроблюваної системи включає:

1. ПЗ загального призначення: пакет програм Microsoft Office, що включає текстовий редактор Word для створення проєктної документації, технічних специфікацій та звітів, а також табличний процесор Excel для виконання розрахунків проєкту.

2. ПЗ для тестування та перевірки веб-систем: ряд браузерів для забезпечення кросбраузерної сумісності та тестування функціональності системи, серед яких Google Chrome, Microsoft Edge, Opera.

3. RAD-засоби для проєктування та моделювання: інструменти швидкої розробки архітектури проєкту, до яких належать:

- Draw.io – веб-редактор для створення UML-діаграм, блок-схем, архітектурних схем та діаграм баз даних з можливістю експорту в різні формати.
- Enterprise Architect – платформа моделювання з повною підтримкою UML, функціоналом для управління вимогами та роботою над архітектурними моделями системи.

3.3.4 Програмна документація

Технічна документація рекомендаційної системи охоплює широкий спектр аспектів для комплексного опису функціоналу та архітектури системи, створюючи надійну основу для ефективної експлуатації та подальшого розвитку платформи. Документація сприяє успішному впровадженню системи в операційні процеси кінотеатру та забезпечує якісну підтримку як кінцевих користувачів, так і технічного персоналу. Додаток Б містить приклади програмного коду.

Специфікація програмного продукту визначає наступні параметри:

- Найменування системи – UCine. Рекомендаційна система для кінотеатрів з графовою архітектурою.
- Розробник – Бицька Вікторія
- Технологічний стек включає: Neo4j, Express.js, Vue.js, Git.
- Інформація про експлуатацію: постійна підтримка з боку розробника.
- Інформація про мінімальні параметри технічних засобів для роботи системи

Визначені області застосування системи включають:

- Надання рекомендацій на основі схожості фільмів;
- Рекомендації за заданими критеріями;
- Рекомендації на основі вибірки фільмів;
- Управління персональними списками збережених фільмів;
- Відображення актуальної інформації про прем'єри.

У Додатку В надано інструкцію користувача для рекомендаційної системи кінотеатру.

3.4 Результати реалізації інформаційної системи

Результатом реалізації є веб-додаток рекомендаційної системи для кінотеатру, що забезпечує персоналізований підбір кінофільмів на основі графових алгоритмів Similarity та PageRank. Розроблена система включає комплекс взаємопов'язаних сторінок та функціональних модулів, що забезпечують повний цикл взаємодії користувача з рекомендаційним сервісом.

Система автентифікації користувачів, що включає сторінки авторизації та реєстрації нових облікових записів (рисунки 3.4-3.6). При створенні нового акаунту відповідна вершина користувача автоматично записується до бази даних, що забезпечує можливість подальшого формування персональних списків користувачів. Процес авторизації реалізовано з комплексною системою безпеки, що включає валідацію електронної пошти за допомогою validator.js, хешування паролів через bcrypt для захисту облікових даних, та генерацію JWT (JSON Web

Token) токенів для забезпечення безпечних сесій користувачів. Система перевіряє наявність користувача в базі даних Neo4j, порівнює хешовані паролі та у випадку успішної автентифікації створює новий токен.

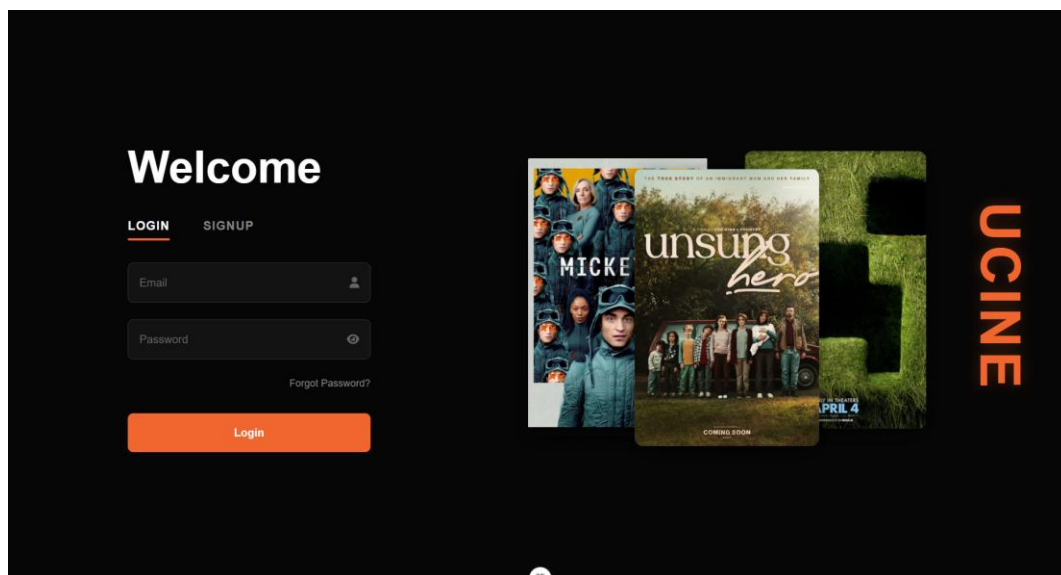


Рисунок 3.4 – Сторінка авторизації

Джерело: реалізовано автором

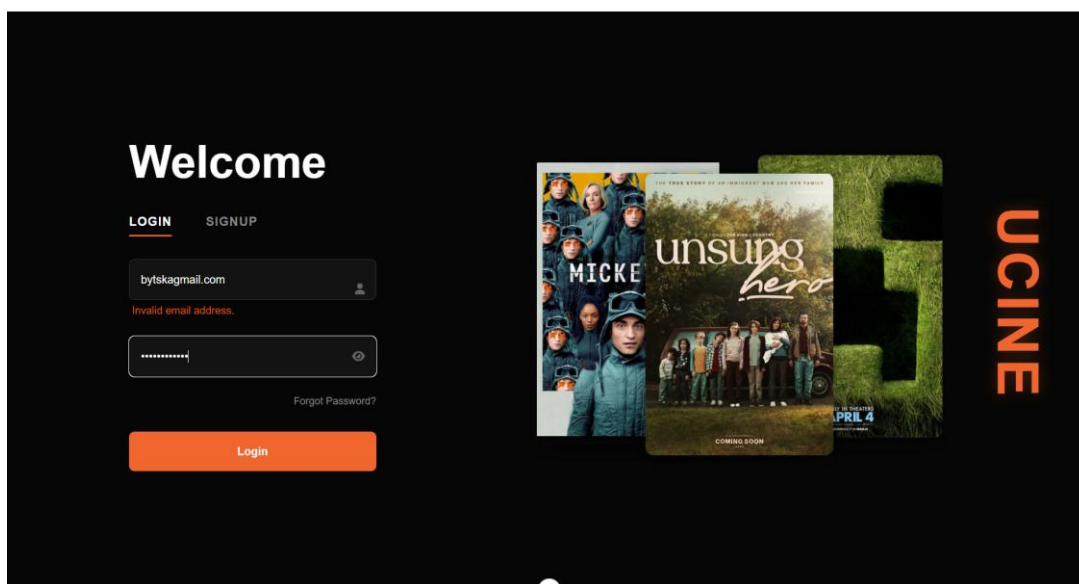


Рисунок 3.5 – Сторінка авторизації. Введення невірної електронної адреси

Джерело: реалізовано автором

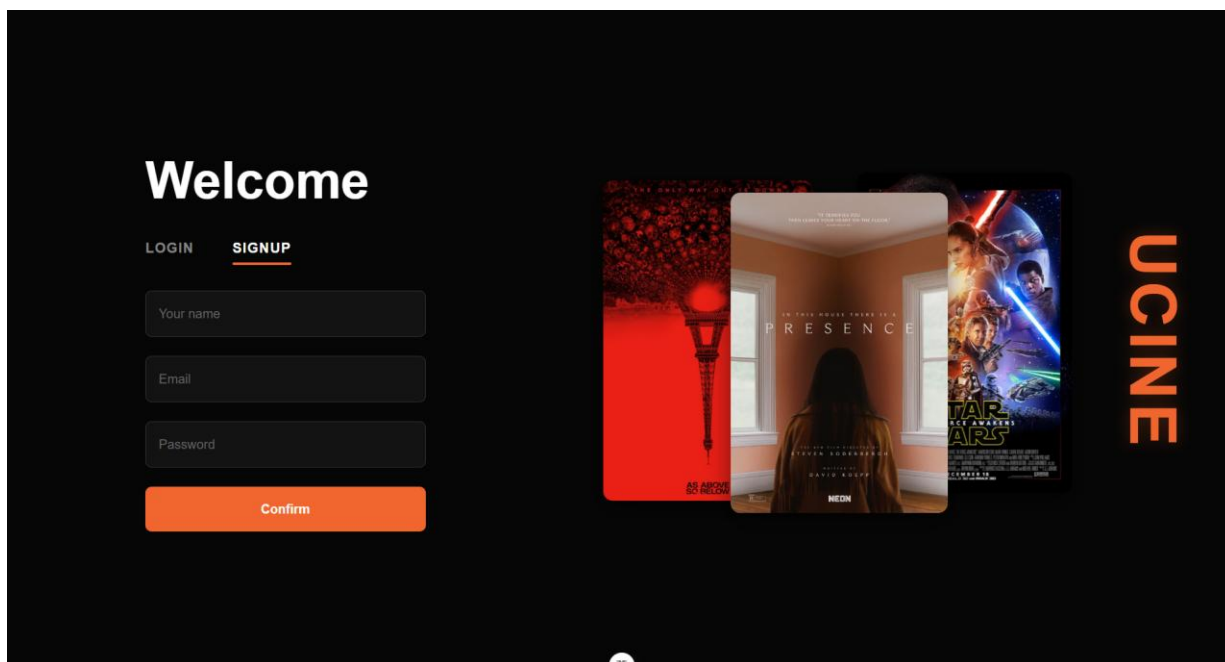


Рисунок 3.6 – Сторінка реєстрації

Джерело: реалізовано автором

Головна сторінка слугує центральним елементом системи, що містить інформацію про систему, її функціонал та переваги. Інтерфейс головної сторінки оптимізований для швидкого переходу до різних розділів системи, а також включає карусель із фільмами, що мають високий рейтинг та можуть бути потенційно цікавими для користувачів системи (рисунок 3.7).

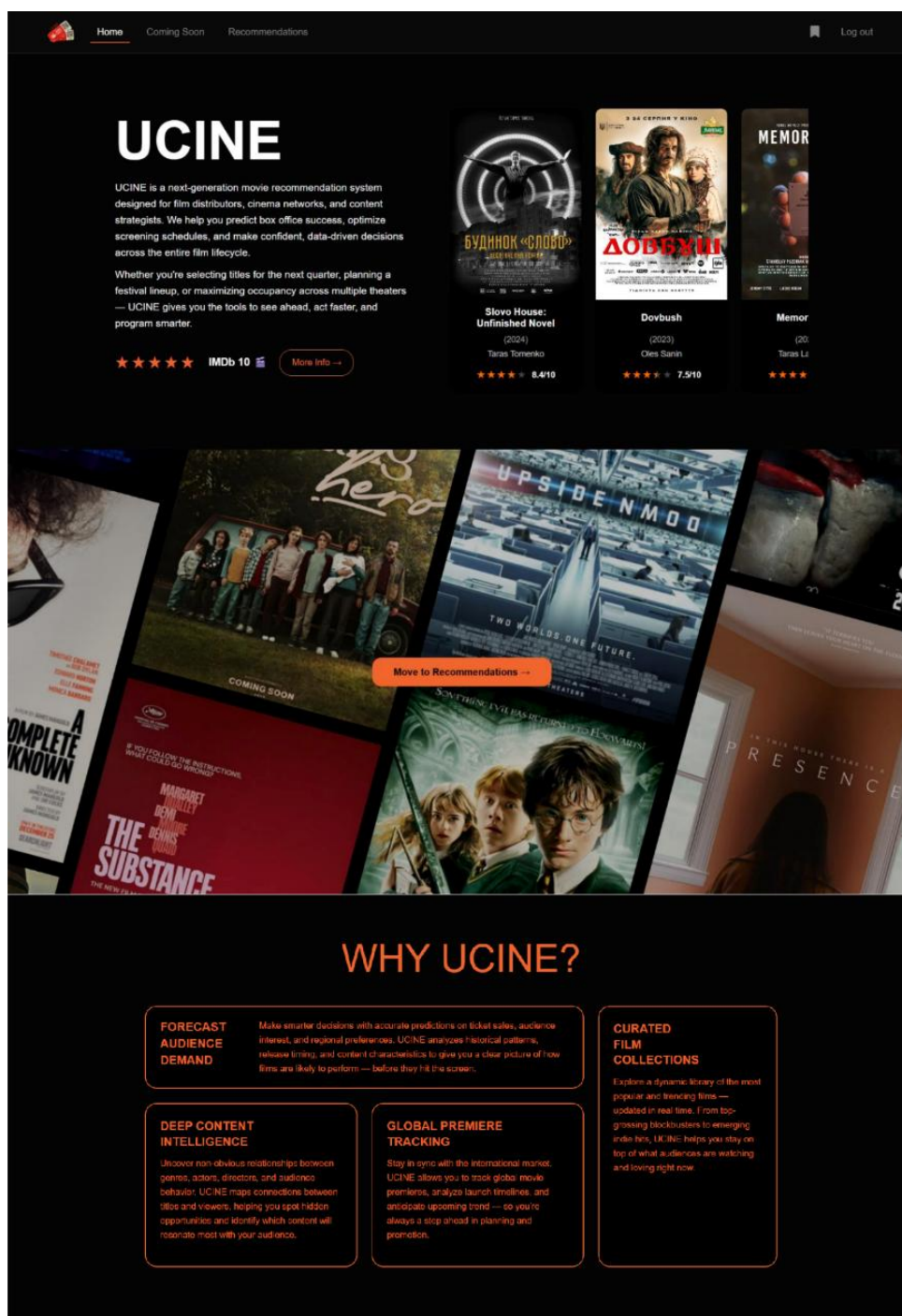


Рисунок 3.7 – Головна сторінка рекомендаційної системи

Джерело: реалізовано автором

Сторінка прем'єр демонструє актуальні новинки кінопрокату та майбутні релізи, дозволяючи користувачам бути в курсі найсвіжіших кінематографічних подій та планувати майбутні перегляди (рисунок 3.8).

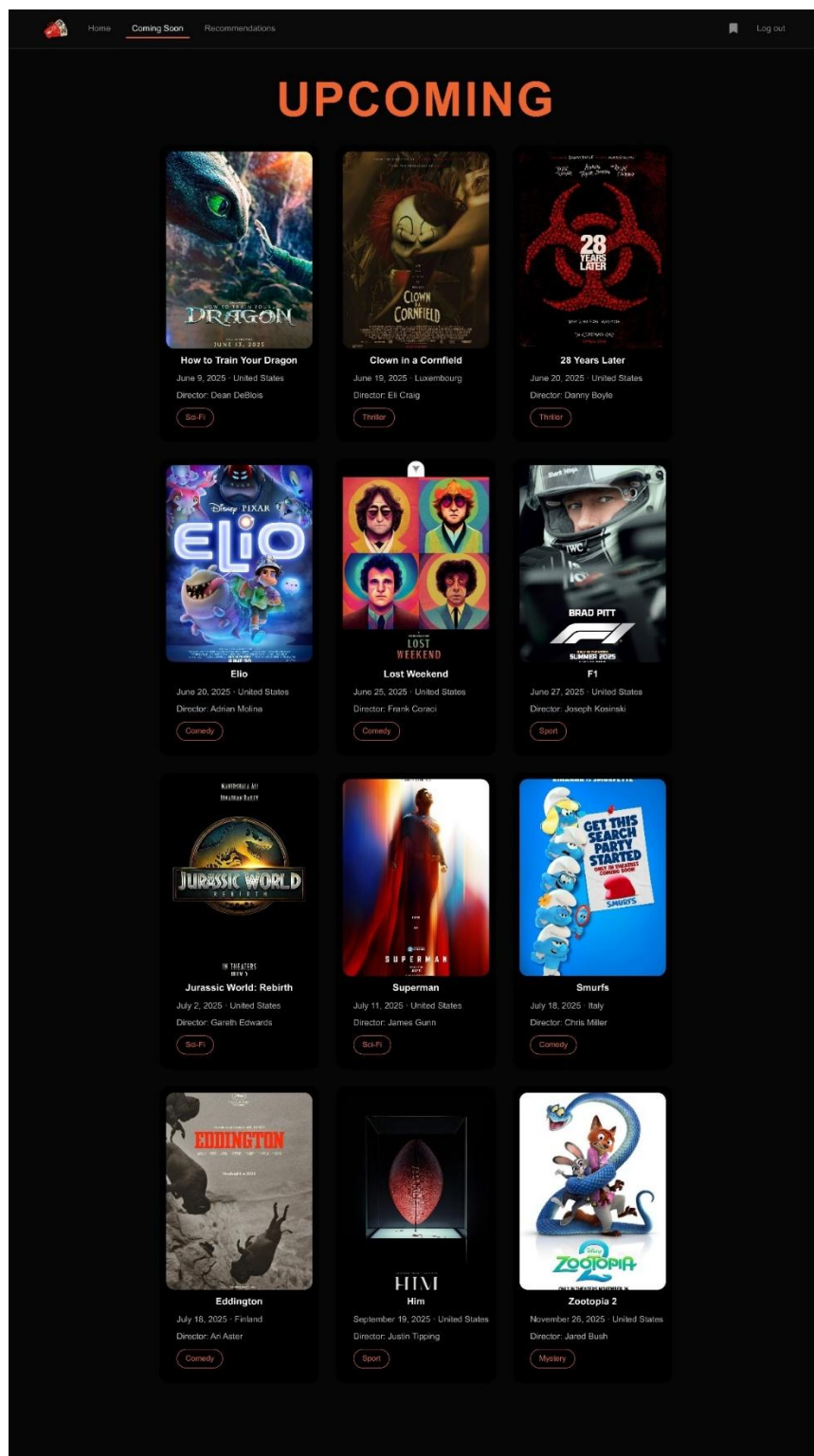


Рисунок 3.8 – Сторінка з прем'єрами

Джерело: реалізовано автором

Рекомендаційна сторінка є ключовим функціональним модулем системи, що реалізує два алгоритми підбору фільмів. Перший метод дозволяє користувачам задавати власні критерії пошуку через заповнення спеціальних полів,

забезпечуючи гнучкість у формуванні запитів. Метод SelectSuggest пропонує користувачеві випадкову вибірку з п'ятнадцяти фільмів, з яких мають бути обрані ті, які попередньо були цікаві аудиторії. Якщо у представленій вибірці немає таких кінострічок, то реалізовано можливість оновлення вибірки без перевантаження сторінки (рисунки 3.9-3.12).

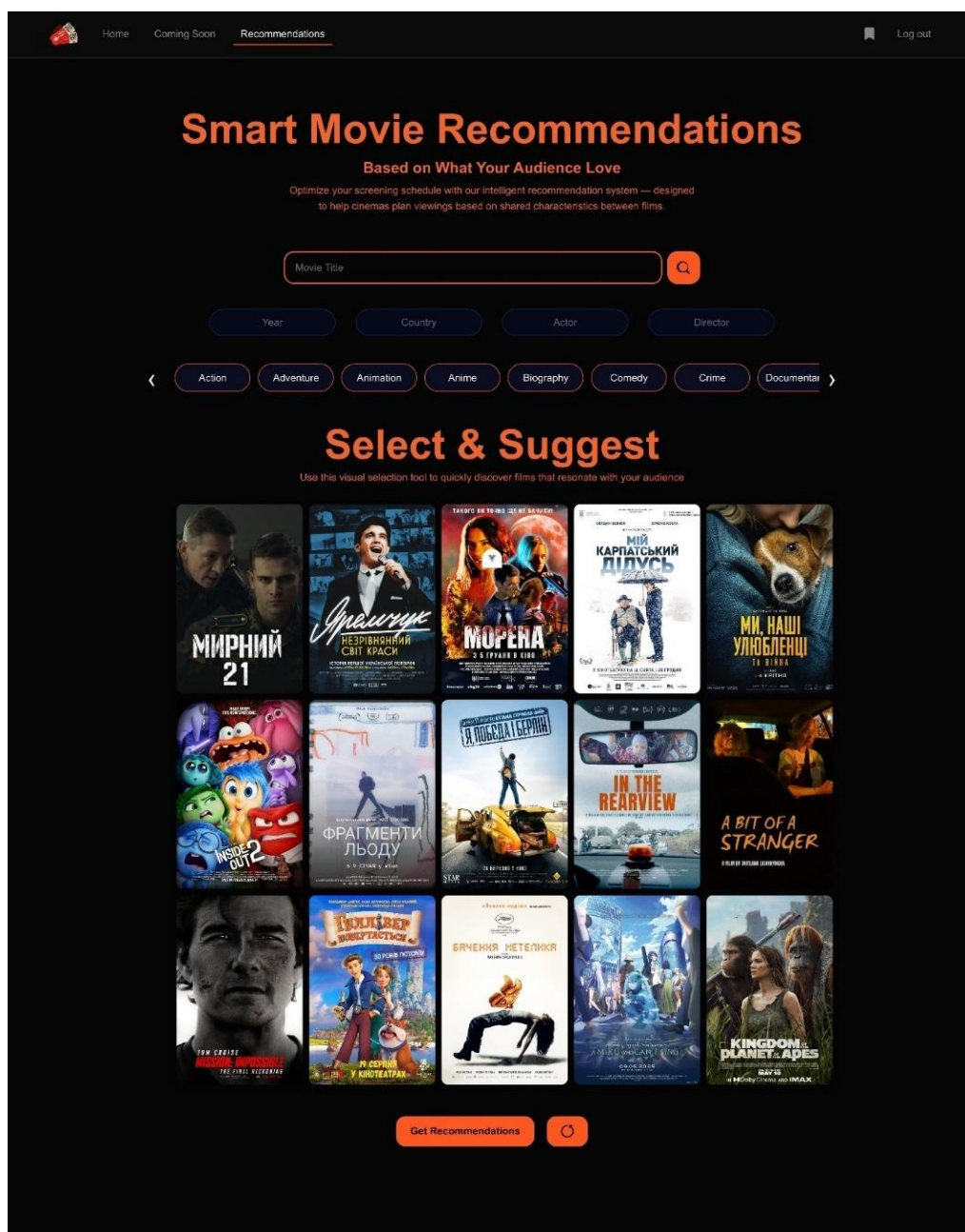


Рисунок 3.9 – Сторінка з формами для рекомендацій на основі заданих критеріїв та вибірки фільмів

Джерело: реалізовано автором

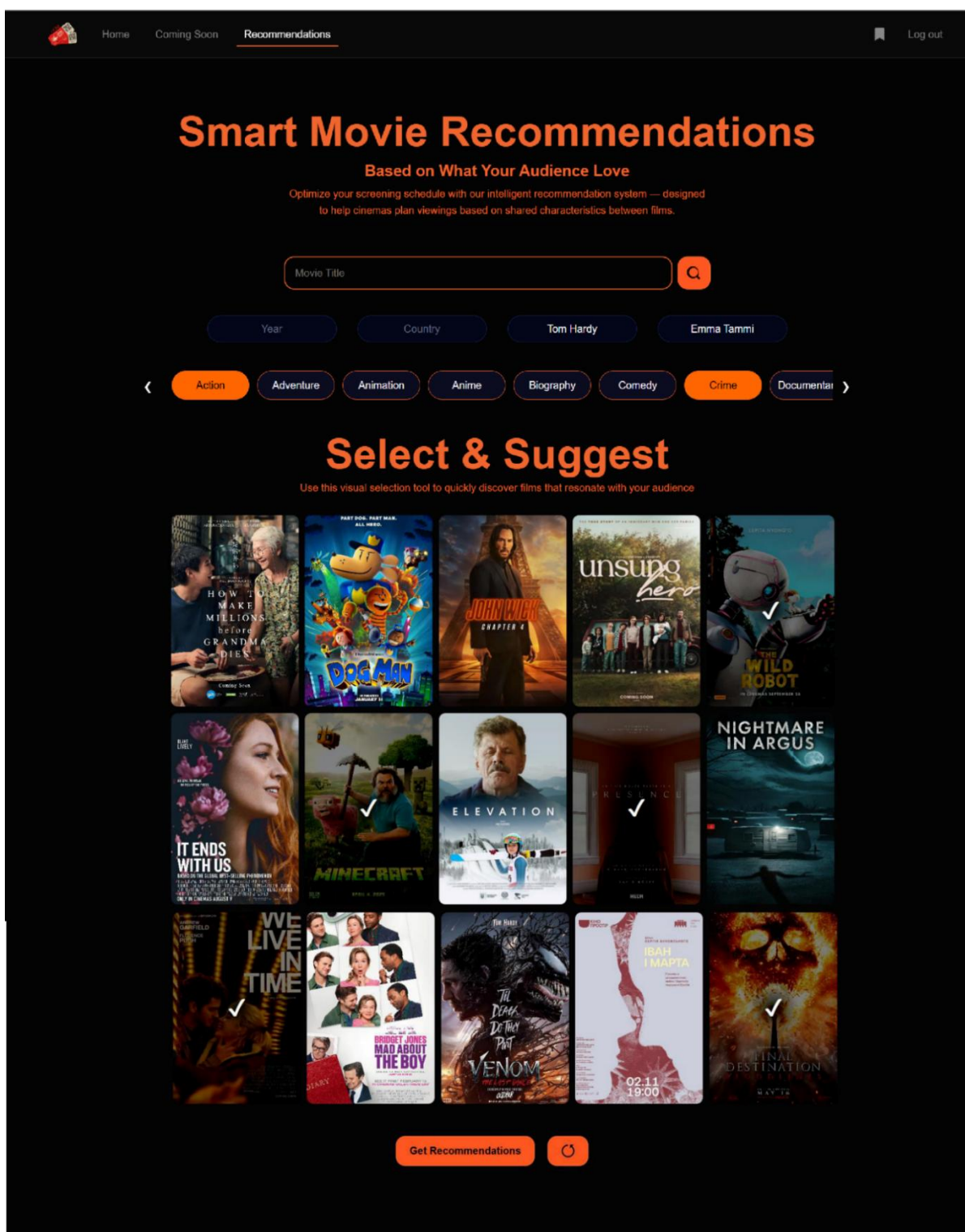


Рисунок 3.10 – Заповнені рекомендаційні форми

Джерело: реалізовано автором

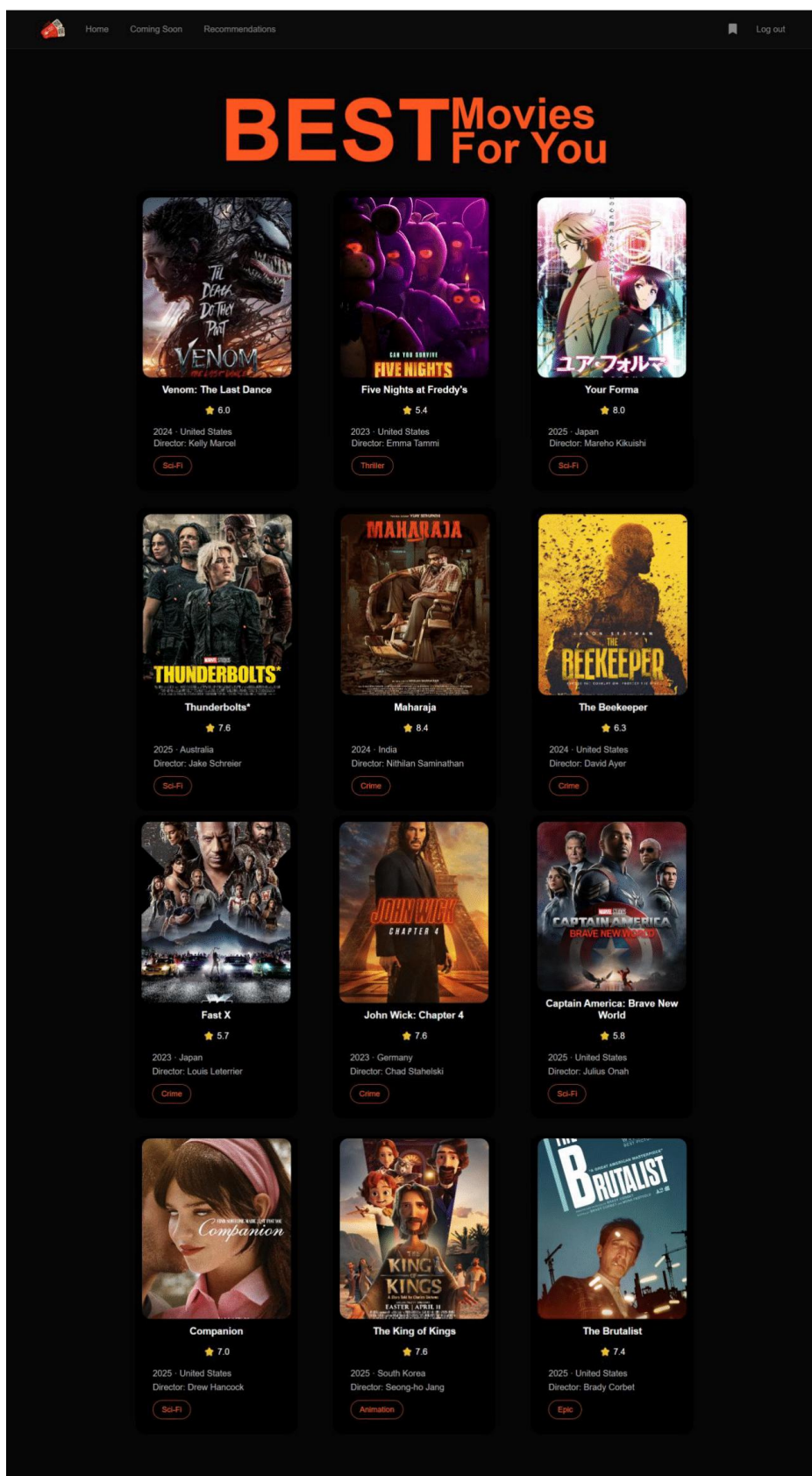


Рисунок 3.11 – Сторінка з наданими рекомендаціями на основі заданих критеріїв

Джерело: реалізовано автором

neo4j\$ MATCH (m:Movie) WHERE m.release_date ≤ date("2025-06-08") AND m.rate ≥ 5.0 AND m.release_date.year ≥ 2020 CALL { WITH m OPTIO...

	title	description
1	"Venom: The Last Dance"	"In Venom: The Last Dance, Tom Hardy returns as Venom, one of Marvel's greatest and most complex characters, for the final film in the trilogy. Eddie and Venom are on the run. Hunted b
2	"Five Nights at Freddy's"	"In the year 2000, a desperate young man takes a job at an infamous, long-abandoned pizzeria with a dark past, bringing his little sister along. As they uncover the restless spirits and veng
3	"Your Forma"	"In 1992, a pandemic of viral encephalitis shook the world. The medical technology that saved humanity evolved into "Your Forma"-an invasive information device implanted into the brain.
4	"Thunderbolts**"	"After finding themselves ensnared in a death trap, an unconventional team of antiheroes must go on a dangerous mission that will force them to confront the darkest corners of their pasts
5	"Maharaja"	"A simple and soft spoken barber in a small town reports the theft of Lakshmi (his dustbin) to the police. Initially they are angry and amused and take it lightly but when he keeps insisting t
6	"The Beekeeper"	"Adam Clay is a beekeeper, but someone very important to him is scammed and driven to harm. However, what these power and money hungry scammers don't know, is that Clay is an ag
7	"Fast X"	"Over many missions and against impossible odds, Dom Toretto (Vin Diesel) and his family have outsmarted, out-nerved and outdriven every foe in their path. Now, they confront the most
8	"John Wick: Chapter 4"	"Reviewers say 'John Wick: Chapter 4' is lauded for its intense action, impressive stunts, and Keanu Reeves' performance. The film's high-octane action, stunning visuals, and expanded u
9	"Captain America: Brave New World"	"Some time after the events of The Falcon and the Winter Soldier, Sam Wilson has now fully embraced his title as the new Captain America. Now Sam is summoned to the White House as
10	"Companion"	"Since Iris and Josh met in a supermarket, the two have been inseparable. Now they want to spend the weekend together with Josh's friends at the luxurious lake estate of Sergey. His mis
11	"The King of Kings"	"A father tells his son the greatest story ever told, and what begins as a bedtime tale becomes a life-changing journey. Through vivid imagination, the boy walks alongside Jesus, witnessin

Started streaming 12 records after 483 ms and completed after 492 ms.

Рисунок 3.12 – Результати виконання рекомендації на основі заданих критеріїв у середовищі Neo4j

Джерело: реалізовано автором

Сторінка детальної інформації про фільм надає повну характеристику обраного кінофільму та включає функцію Get Similar, що генерує рекомендації схожих за характеристиками фільмів на основі алгоритму подібності (рисунки 3.13-3.15).



Рисунок 3.13 – Сторінка фільму

Джерело: реалізовано автором

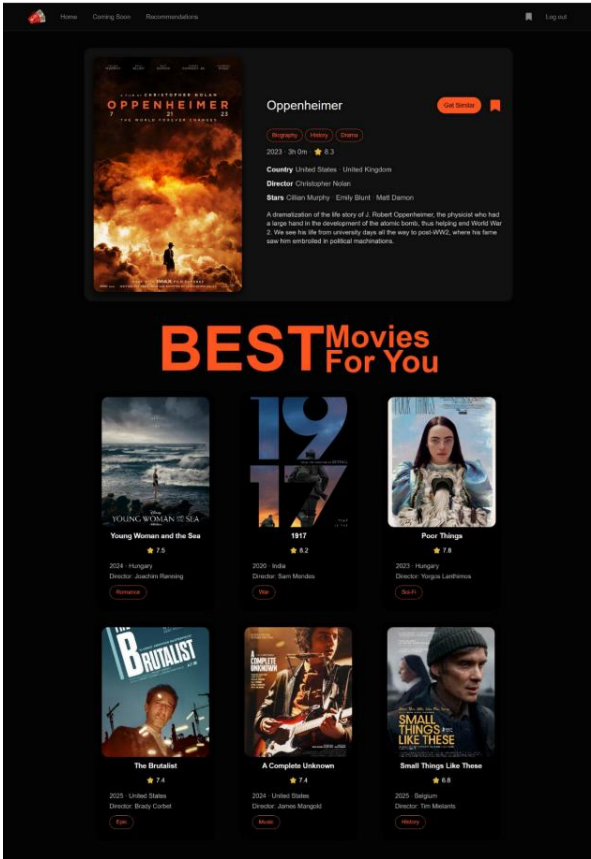


Рисунок 3.14 – Надані рекомендації на основі загальної подібності

Джерело: реалізовано автором

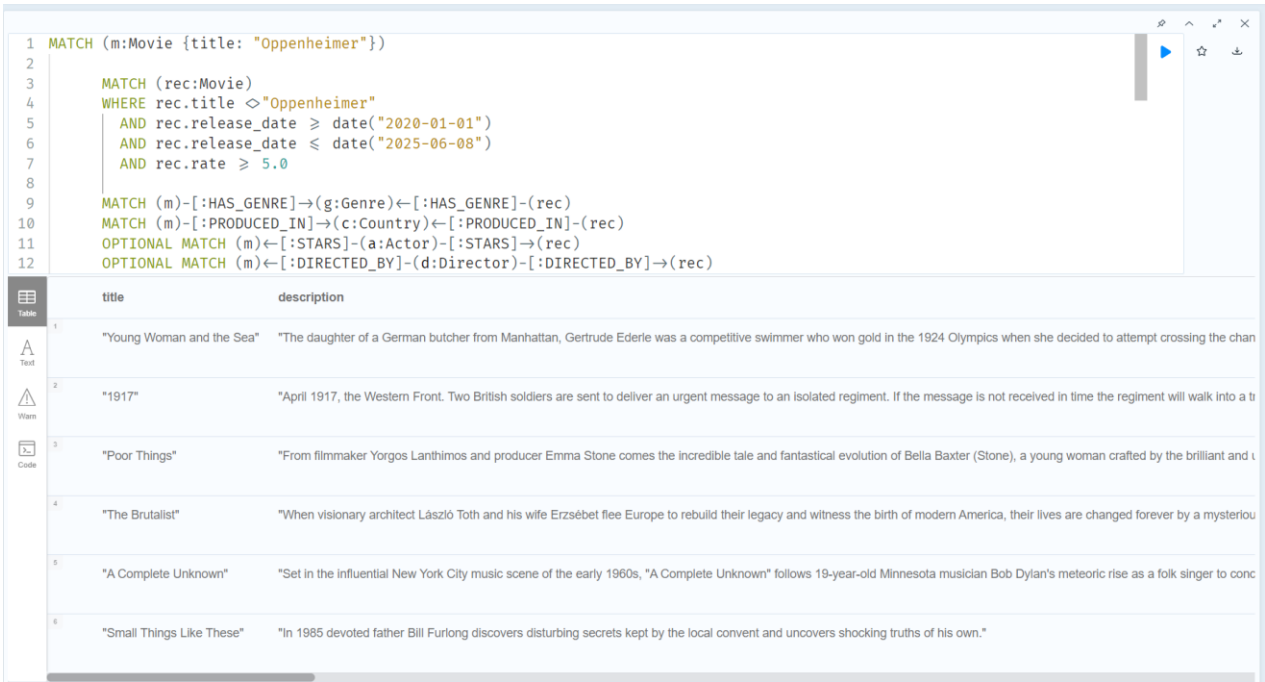


Рисунок 3.15 – Результати виконання рекомендації на основі загальної подібності у середовищі Neo4j

Джерело: реалізовано автором

Сторінка збережених фільмів дозволяє користувачам формувати власний список збережень улюблених кінострічок, при цьому в базі даних автоматично створюються відповідні зв'язки між користувачем та обраними фільмами. Використання списку збережень надає швидкий доступ до обраних фільмів та фактично служить списком «до перегляду», дозволяючи користувачам зберігати цікаві фільми для майбутнього кінопрокату (рисунки 3.16).

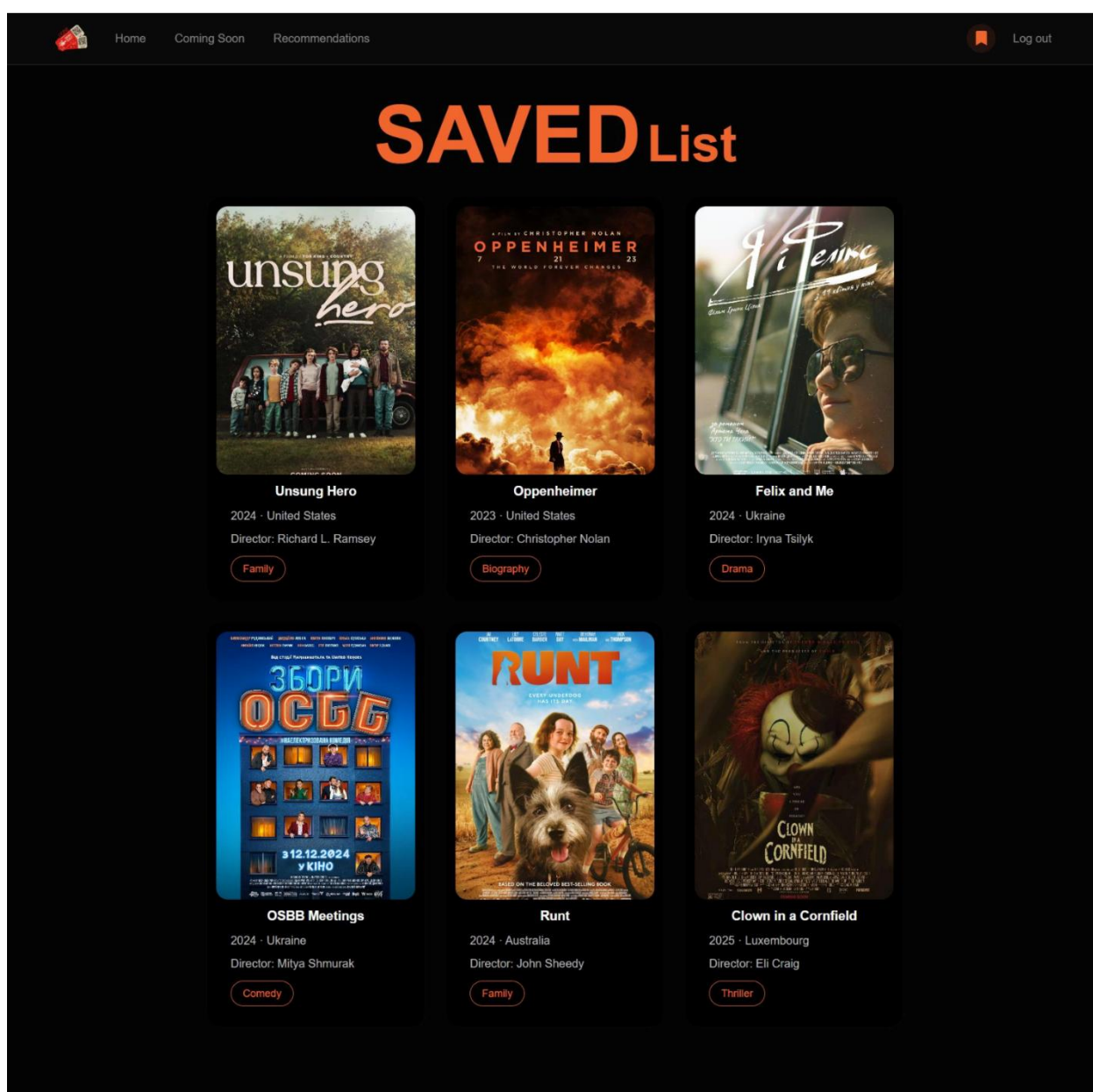


Рисунок 3.16 – Сторінка збережень користувача

Джерело: реалізовано автором

При зростанні кількості фільмів у базі та збільшенні числа користувачів, критично важливим є забезпечення можливості масштабування системи. Масштабування рекомендаційної системи на основі графової БД буде передбачає горизонтальне розширення за рахунок розподілу даних між декількома серверними вузлами. Графова структура з інформацією про фільми, акторів, режисерів та жанри може бути ефективно шардована для забезпечення паралельної обробки запитів від множини менеджерів одночасно.

Алгоритми PageRank та Similarity потребують значних обчислювальних ресурсів, тому їх виконання планується оптимізувати через кешування результатів та асинхронну обробку у фоновому режимі, що дозволить системі швидко надавати рекомендації менеджерам без затримок на повторні розрахунки.

Висновки до розділу 3

У третьому розділі було детально розглянуто процеси проєктування та реалізації всіх ключових компонентів рекомендаційної системи кінотеатру. Комплексний підхід до розробки охопив чотири основні напрямки забезпечення системи, який включає аналіз та вибір інформаційного, технічного, програмного забезпечення та практичну реалізацію рекомендаційної системи.

В рамках дослідження інформаційного забезпечення було спроектовано ефективну архітектуру збору та обробки даних, розроблено систему класифікації та кодування інформації про фільми та користувачів, а також створено оптимальну структуру інформаційних масивів. Обґрунтовано вибір графової бази даних Neo4j як найбільш підходящого рішення для зберігання та обробки складних взаємозв'язків між користувачами та кінофільмами у форматі графової моделі. Побудовано інфологічну та даталогічну моделі БД, що забезпечують ефективне зберігання та швидкий доступ до інформації.

Технічне забезпечення системи включає розробку комплексної схеми автоматизації, структури технічних засобів та організацію автоматизованих робочих місць.

Програмне забезпечення розробленої системи базується на поєднанні передових технологій веб-розробки, що забезпечують високу продуктивність рекомендаційної системи. На стороні серверної логіки використовується Node.js, який являє собою середовище виконання JavaScript, що дозволяє створювати швидкі та асинхронні веб-додатки. У поєднанні з Express.js, легким та гнучким фреймворком для побудови RESTful API, забезпечується ефективне управління маршрутами, обробка запитів та взаємодія з базою даних. На клієнтській стороні реалізовано інтерфейс за допомогою Vue.js, який дозволяє створювати динамічні та реактивні веб-компоненти. Завдяки компонентному підходу до побудови інтерфейсу, Vue забезпечує гнучкість та можливість повторне використання коду. Архітектура програмного забезпечення реалізована з урахуванням принципів модульності та розділення функціоналу, що дозволяє незалежно розробляти, тестувати та масштабувати окремі частини системи. Використання такого підходу є важливим в контексті підтримки системи, подальшого впровадження нових функцій та адаптації до змін у вимогах користувачів або технологіях.

Крім цього, створено інтуїтивно зрозумілий користувацький інтерфейс з реалізацією трьох різних алгоритмів рекомендацій, системою автентифікації користувачів та персоналізованими функціями збереження фільмів. Впровадження алгоритмів на основі методів Similarity та PageRank забезпечує високу точність генерованих рекомендацій.

ВИСНОВКИ

Кваліфікаційна бакалаврська робота на тему «Розробка рекомендаційної системи кінотеатру на основі графової бази даних» є теоретично-практичним дослідженням, що представляє інноваційне технологічне рішення для залучення та утримання аудиторії в умовах інтенсивного зростання обсягів кінопродукції. Використання графових баз даних для моделювання складних взаємозв'язків кіноіндустрії представляє передовий підхід, який дозволяє виявляти приховані закономірності та надавати більш точні рекомендації порівняно з традиційними методами, що використовуються у процесі планування кінопрокату та потребували ручного підбору фільмів керівниками репертуару. Розробка рекомендаційної системи не лише відповідає сучасним технологічним трендам, але й адресує реальні бізнес-потреби кінотеатрів у підвищенні ефективності операційної діяльності та покращенні якості обслуговування клієнтів.

У першому здійснено всебічний аналіз предметної області, існуючих методів та алгоритмів для реалізації рекомендаційних платформ, що дозволило створити теоретичну базу для розробки системи та визначення найбільш перспективних напрямків її реалізації.

Другий розділ присвячено детальному аналізу та специфікації вимог до рекомендаційної системи. Було визначено ключових стейкхолдерів та їх бізнес-вимоги, на базі яких сформовано функціональні та нефункціональні вимоги до системи. Постановка задачі чітко окреслила основні цілі системи: надання персоналізованих рекомендацій фільмів на основі аналізу подібності кінострічок. Математичне забезпечення та розроблені алгоритми задачі забезпечують чітке розуміння логіки роботи програмної системи. Розроблений алгоритм розв'язання задачі базується на графових методах обробки даних, що дозволяє ефективно виявляти приховані зв'язки між користувачами та контентом. Моделювання поведінки системи включало створення діаграм використання, активності та послідовностей, що детально описують взаємодію між компонентами системи. Структурне моделювання охопило проєктування архітектури системи з використанням компонентних та розгортання діаграм. Моделювання поведінки та

структури системи включало створення ряду діаграм, що забезпечують структурне подання взаємодії користувачів з системою та архітектурної організації та взаємозв'язків між компонентами системи.

Третій розділ демонструє практичну реалізацію рекомендаційної системи через детальне проєктування всіх її компонентів. Інформаційне забезпечення включало проєктування первинних документів та структури інформаційних масивів. Особливу увагу приділено вибору графової СУБД, що здійснювався на основі порівняльного аналізу різних рішень з урахуванням специфічних вимог до системи. Інфологічна та даталогічна моделі бази даних відображають складну структуру взаємозв'язків між користувачами, фільмами, жанрами, акторами та іншими сутностями кінематографічної сфери. Обране технічне забезпечення реалізує коректну, стабільну та продуктивну роботу системи та охоплює всі апаратні засоби, необхідні для функціонування платформи, обробки даних, зберігання інформації, забезпечення мережевого підключення, а також захисту даних і системи в цілому. Обране програмне забезпечення розробленої системи базується на поєднанні передових технологій веб-розробки, що забезпечують високу продуктивність рекомендаційної системи.

У перспективі подальшого розвитку системи варто розглянути можливість її інтеграції з зовнішніми кінобазами, що забезпечить наповнення та оновлення даних у системі без втручання адміністратора. Крім цього, доцільним буде дослідити можливість інтеграції сучасних методів машинного навчання, такі як глибокі нейронні мережі та методи природної мови обробки для аналізу текстових описів фільмів, що забезпечить ще вищу ефективність наданих рекомендацій.

Всі поставлені завдання були виконані, а отримані результати задовольняються початкову мету роботи як за обсягом реалізованої функціональності, так і за якістю технічного виконання. Система готова до практичного використання та має міцну основу для подальшого розвитку та вдосконалення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Al-Fedaghi S. Classes in Object-Oriented Modeling (UML):Further Understanding and Abstraction. *ResearchGate*. URL: https://www.researchgate.net/publication/352007234_Classes_in_Object-Oriented_Modeling_UML_Further_Understanding_and_Abstraction.
2. Chen C.-Y., Huang J.-J. Temporal-Guided Knowledge Graph-Enhanced Graph Convolutional Network for Personalized Movie Recommendation Systems. *Future Internet*. 2023. Vol. 15, no. 10. URL: https://www.researchgate.net/publication/374323289_Temporal-Guided_Knowledge_Graph-Enhanced_Graph_Convolutional_Network_for_Personalized_Movie_Recommendation_Systems.
3. Donald Bell. UML's sequence diagram. URL: <https://www.fing.edu.uy/inco/cursos/ingsoft/iis/files/UML-Sequence%20diagram.pdf>.
4. Enterprise Architect. *SparxSystems*. URL: <https://sparxsystems.com/>.
5. Hybrid Movie Recommendation System / R. M. H. Mohamad et al. 2023. URL: https://www.researchgate.net/publication/373868051_Hybrid_Collaborative_Movie_Recommendation_System.
6. IMDb: Ratings, Reviews, and Where to Watch the Best Movies & TV Shows. *IMDb*. URL: <https://www.imdb.com/>.
7. JustWatch - The Streaming Guide. *JustWatch*. URL: <https://www.justwatch.com/>.
8. Kamila I. C., Prihandini R. M., Agatha A. B. Graph Theory: Applications in Movie Recommendation Systems (Netflix)». 2024. URL: https://www.researchgate.net/publication/381504142_Graph_Theory_Applications_in_Movie_Recommendation_Systems_Netflix.
9. Kar P., Roy M., Datta S. Overview of Recommendation Systems. 2025. URL: https://www.researchgate.net/publication/381364266_Overview_of_Recommendation_Systems.

10. Potts C. Using schematic scenarios to understand user needs. *DIS '95: Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*. 1995. P. 247–256. URL: <https://dl.acm.org/doi/pdf/10.1145/225434.225462>.
11. Rotten Tomatoes: Movies | TV Shows | Movie Trailers | Reviews. *Rotten Tomatoes*. URL: <https://www.rottentomatoes.com/>.
12. Sommerville I. Software engineering. 10th ed. Pearson Education Limited. 2016. 775 p. URL: <https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf>.
13. SysML FAQ: what is a block definition diagram (BDD)?. *SysML.org*. URL: <https://sysml.org/sysml-faq/what-is-block-definition-diagram.html>.
14. The Movie Database. *The Movie Database (TMDB)*. URL: <https://www.themoviedb.org/>.
15. Ye Y. An overview of knowledge graph-based recommendation systems. *Applied and Computational Engineering*. 2024. Vol. 73, no. 1. P. 57–68. URL: https://www.researchgate.net/publication/382041455_An_overview_of_knowledge_graph-based_recommendation_systems.
16. Ziaee S. S., Rahmani H., Nazari M. MoRGH: movie recommender system using GNNs on heterogeneous graphs. *Knowledge and Information Systems*. 2024. URL: https://www.researchgate.net/publication/377459571_MoRGH_Movie_Recommender_System_using_GNNs_on_Heterogeneous_Graphs.
17. Власенко Т. Визначення пливу внутрішніх і зовнішніх стейкхолдерів на розвиток підприємства. *Herald of Khmelnytskyi National University. Economic sciences*. 2023. Т. 322, № 5. С. 21–27. URL: <https://doi.org/10.31891/2307-5740-2023-322-5-3>.
18. Єфремов М., Єфремов Ю., Єфремов В. Проектування програмного забезпечення з використанням UML. *Конференції Державного університету «Житомирська політехніка»*. 2016. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2016/06/12.pdf>.

19. Ситник Н.В. Проектування баз і сховищ даних: Навч. Посібник. – К.:КНЕУ, 2004. – 348 с.
20. Ситник Н.В., Краснюк М.Т. Проектування баз і сховищ даних: Навч.-метод. посіб. для самос. вивч. дисц. – К.: КНЕУ, 2005. – 264 с.

ДОДАТКИ

Додаток А

Cypher-запити для завантаження системи

```
// Beetlejuice Beetlejuice
MERGE (beetlejuice:Movie {
  title: "Beetlejuice Beetlejuice",
  release_date: date("2024-09-06"),
  description: "After an unexpected family tragedy, three generations of the
Deetz family return home to Winter River. Still haunted by Beetlejuice, Lydia's
life is turned upside down when her rebellious teenage daughter, Astrid, discovers
the mysterious model of the town in the attic and the portal to the Afterlife is
accidentally opened. With trouble brewing in both realms, it's only a matter of
time until someone says Beetlejuice's name three times and the mischievous demon
returns to unleash his very own brand of mayhem.",
  rate: 6.6,
  duration: 105,
  image: "https://res-
console.cloudinary.com/diczrhtoh/thumbnails/v1/image/upload/v1745441255/NjQwcHgtQm
VldGxlanVpY2VfQmVldGxlanVpY2VfXzI4MjAyNF8yOV91bnhwcDQ=/drilldown"
})
MERGE (timburton:Director {name: "Tim Burton"})
MERGE (comedy:Genre {name: "Comedy"})
MERGE (fantasy:Genre {name: "Fantasy"})
MERGE (horror:Genre {name: "Horror"})
MERGE (usa:Country {name: "United States"})
MERGE (france:Country {name: "France"})
MERGE (michaelkeaton:Actor {name: "Michael Keaton"})
MERGE (winonaryder:Actor {name: "Winona Ryder"})
MERGE (catherineohara:Actor {name: "Catherine O'Hara"})
MERGE (jennaortega:Actor {name: "Jenna Ortega"})

MERGE (beetlejuice)-[:DIRECTED_BY]->(timburton)
MERGE (beetlejuice)-[:HAS_GENRE]->(comedy)
MERGE (beetlejuice)-[:HAS_GENRE]->(fantasy)
MERGE (beetlejuice)-[:HAS_GENRE]->(horror)
MERGE (beetlejuice)-[:PRODUCED_IN]->(usa)
MERGE (beetlejuice)-[:PRODUCED_IN]->(france)
MERGE (beetlejuice)-[:STARS]->(michaelkeaton)
MERGE (beetlejuice)-[:STARS]->(winonaryder)
MERGE (beetlejuice)-[:STARS]->(catherineohara)
MERGE (beetlejuice)-[:STARS]->(jennaortega)

// Oppenheimer
MERGE (oppenheimer:Movie {
  title: "Oppenheimer",
  release_date: date("2023-07-01"),
  description: "A dramatization of the life story of J. Robert Oppenheimer, the
physicist who had a large hand in the development of the atomic bomb, thus helping
end World War 2. We see his life from university days all the way to post-WW2,
where his fame saw him embroiled in political machinations.",
  rate: 8.3,
  duration: 180,
  image: "https://res-
console.cloudinary.com/diczrhtoh/thumbnails/v1/image/upload/v1745506210/b3BwZW5oZW
ltZXJfcG9zdGVyXzE2NTg0MTE2MDE1OTNfaThuZ3Rm/drilldown"
})
WITH oppenheimer
```

```
MATCH (drama:Genre {name: "Drama"})
MATCH (christophernolan:Director {name: "Christopher Nolan"})
MATCH (usa:Country {name: "United States"})
MATCH (uk:Country {name: "United Kingdom"})

MERGE (biography:Genre {name: "Biography"})
MERGE (history:Genre {name: "History"})
MERGE (cillianmurphy:Actor {name: "Cillian Murphy"})
MERGE (emilyblunt:Actor {name: "Emily Blunt"})
MERGE (mattdamon:Actor {name: "Matt Damon"})

MERGE (oppenheimer)-[:DIRECTED_BY]->(christophernolan)
MERGE (oppenheimer)-[:HAS_GENRE]->(biography)
MERGE (oppenheimer)-[:HAS_GENRE]->(drama)
MERGE (oppenheimer)-[:HAS_GENRE]->(history)
MERGE (oppenheimer)-[:PRODUCED_IN]->(usa)
MERGE (oppenheimer)-[:PRODUCED_IN]->(uk)
MERGE (oppenheimer)-[:STARS]->(cillianmurphy)
MERGE (oppenheimer)-[:STARS]->(emilyblunt)
MERGE (oppenheimer)-[:STARS]->(mattdamon)
```

Серверна логіка обробки POST-запиту для збереження фільму

```
app.post('/save-movie', authenticate, async (req, res) => {
  const { title } = req.body

  // Валідація вхідних даних
  if (!title || title.trim() === '') {
    return res.status(400).json({ error: 'Movie title is required.' })
  }

  const session = driver.session()
  try {
    const email = req.user.email

    const checkMovieQuery = `
      MATCH (m:Movie { title: $title })
      RETURN m
    `

    const movieResult = await session.run(checkMovieQuery, { title })

    if (movieResult.records.length === 0) {
      return res.status(404).json({ error: 'Movie not found.' })
    }

    const checkSavedQuery = `
      MATCH (u:User { email: $email })-[r:SAVED]->(m:Movie { title: $title })
      RETURN r
    `

    const savedResult = await session.run(checkSavedQuery, { email, title })

    if (savedResult.records.length > 0) {
      return res.status(409).json({ error: 'Movie already saved.' })
    }

    const saveQuery = `
      MATCH (u:User { email: $email }), (m:Movie { title: $title })
      MERGE (u)-[:SAVED]->(m)
      RETURN m {
        .title,
        .release_date,
        .description,
        .rate,
        .duration,
        .image
      } as movie
    `

    const result = await session.run(saveQuery, { email, title })

    const savedMovie = result.records[0]?.get('movie')

    return res.status(200).json({
      message: 'Movie saved successfully.',
      movie: savedMovie,
    })
  } catch (error) {
    console.error('Error saving movie:', error)
    return res.status(500).json({ error: 'Internal server error.' })
  } finally {
    await session.close()
  }
})
```

Програмна реалізація сторінки, на яку виводяться рекомендації

```
<template>
  <div class="recommendations-page">
    <h2 class="similar-title">
      <span class="best">BEST</span>
      <span class="movies-for-you">Movies<br />For You</span>
    </h2>

    <div v-if="isOnlyTitle" class="based-on-title">
      Recommendations based on <strong class="clickable"
@click="goToMovie(route.query.title)">{{ route.query.title }} → </strong>
    </div>

    <div v-if="loading" class="loading">Loading...</div>

    <div v-else-if="movies.length === 0" class="loading">No results found</div>

    <div v-else class="movies-grid">
      <div
        v-for="(movie, index) in movies"
        :key="movie.title + index"
        class="movie-card"
        @click="goToMovie(movie.title)"
      >
        <div class="poster-container">
          
        </div>

        <div class="movie-info">
          <h2 class="movie-title">{{ movie.title }}</h2>

          <p v-if="movie.rating" class="rating">
            ★ {{ movie.rating.toFixed(1) }}
          </p>

          <div class="year-country" v-if="movie.year || movie.country">
            {{ movie.year?.low ?? movie.year ?? 'Unknown' }} · {{ movie.country }}
          </div>

          <p v-if="movie.director" class="director">
            Director: {{ movie.director }}
          </p>

          <p v-if="movie.genres.length" class="genre">
            {{ movie.genres[0] }}
          </p>
        </div>
      </div>
    </div>
  </div>
</template>

<script setup>
import { ref, onMounted, computed } from 'vue';
import { useRoute, useRouter } from 'vue-router';

const route = useRoute();
const router = useRouter();

const movies = ref([]);
const loading = ref(true);
```

```

const goToMovie = (title) => {
  router.push({ name: 'FilmDetail', params: { title } });
};

const isOnlyTitle = computed(() => {
  const query = route.query;
  return (
    'title' in query &&
    query.title &&
    Object.keys(query).length === 1
  );
});

onMounted(async () => {
  try {
    const hasQueryParams = Object.keys(route.query).length > 0;

    if (hasQueryParams) {
      const params = new URLSearchParams(route.query).toString();
      const res = await
fetch(`http://localhost:3000/api/recommendations?${params}`);

      const data = await res.json();

      movies.value = Array.isArray(data)
        ? data.map(movie => ({
            title: movie.title,
            description: '',
            genres: movie.genres || [],
            actors: [],
            rating: movie.rating || 0,
            posterUrl: movie.posterUrl,
            year: movie.year,
            country: movie.country,
            director: movie.director
          }))
        : [];
    } else {
      const storedSelectedMovies = localStorage.getItem('selectedMovies');
      if (!storedSelectedMovies) {
        console.warn('No selected movies found in localStorage. ');
        loading.value = false;
        return;
      }

      const selectedMovies = JSON.parse(storedSelectedMovies);

      const res = await fetch('http://localhost:3000/movies/selectsuggestmethod',
{
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ selectedMovies }),
  });

      const data = await res.json();

      movies.value = Array.isArray(data)
        ? data.map(movie => ({
            title: movie.title,
            description: '',
            genres: movie.genres || [],
            actors: [],
            rating: movie.rating || 0,

```

```

        posterUrl: movie.posterUrl,
        year: movie.year,
        country: movie.country,
        director: movie.director
    )))
    : [];
    }
} catch (error) {
    console.error('Failed to fetch recommendations:', error);
} finally {
    loading.value = false;
}
});
</script>

```

```

<style scoped>
.recommendations-page {
    padding-bottom: 10px;
}

.loading {
    font-size: 1.5rem;
    font-weight: 600;
    color: #ff5722;
    text-align: center;
    padding: 30px 0;
    user-select: none;
    font-family: 'Inter', sans-serif;
}

.similar-title {
    display: flex;
    align-items: flex-start;
    gap: 0rem;
    font-size: 3rem;
    color: #ff5722;
    margin-bottom: 2rem;
    justify-content: center;
    flex-wrap: wrap;
    margin-top: 3rem;
}

.best {
    font-weight: bold;
    font-size: 10rem;
    line-height: 1;
}

.movies-for-you {
    display: flex;
    flex-direction: column;
    justify-content: center;
    font-weight: bold;
    font-size: 5rem;
    line-height: 0.8;
    margin-top: 1rem;
}

.based-on-title {
    cursor: pointer;
    color: #ff5722;
    margin-bottom: 30px;
}

```

```

    margin-bottom: 30px;
    font-size: 1rem;
    font-family: 'Inter', sans-serif;
    max-width: 1100px;
    margin-left: auto;
    margin-right: auto;
    padding: 0 1rem;
    text-align: center;
}
.based-on-title:hover {
    text-decoration: none;
}

.clickable {
    cursor: pointer;
}

.movies-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
    gap: 2rem;
    max-width: 1100px;
    margin: 0 auto 150px;
    padding: 0 1rem;
    justify-items: center;
}

.movie-card {
    width: 100%;
    max-width: 300px;
    background: #000;
    border-radius: 20px;
    padding: 12px;
    color: var(--text-light);
    font-family: 'Inter', sans-serif;
    display: flex;
    flex-direction: column;
    align-items: stretch;
    height: 560px;
}

.movie-card:hover {
    transform: translateY(-8px);
    box-shadow: 0 12px 24px rgba(255, 87, 34, 0.3);
}

.poster-container {
    position: relative;
    overflow: hidden;
    border-radius: 8px;
    height: 420px;
}

.poster {
    width: 100%;
    height: 100%;
    object-fit: cover;
    border-radius: 16px;
}

.movie-info {
    display: flex;

```

```

    flex-direction: column;
    justify-content: flex-start;
    flex-grow: 1;
    gap: 0px;
    margin-top: 12px;
}

.movie-title {
    margin: 0;
    font-size: 18px;
    font-weight: bold;
    line-height: 1.2;
    text-align: center;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: normal;
    margin-bottom: 15px;
}

.year-country,
.director {
    color: #bbb;
    font-size: 1rem;
    line-height: 1;
    margin-left: 20px;
    margin-top: 10px;
}

.rating {
    text-align: center;
    font-size: 16px;
    margin-bottom: 10px;
}

.genre {
    margin-top: auto;
    padding: 6px 16px;
    border: 1px solid var(--primary-orange);
    border-radius: 9999px;
    color: var(--primary-orange);
    font-size: 0.9rem;
    font-weight: 500;
    max-width: fit-content;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    transition: all 0.2s ease;
    margin-left: 20px;
}

.genre:hover {
    background-color: var(--primary-orange);
    color: black;
    cursor: default;
}

/* 1024px */
@media (max-width: 1024px) {
    .similar-title {
        font-size: 2.5rem;
    }
}

```

```

.best {
  font-size: 8rem;
}

.movies-for-you {
  font-size: 4rem;
}

.movies-grid {
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  padding: 0 0.5rem;
}

.movie-card {
  max-width: 280px;
}

.movie-title {
  font-size: 16px;
}

.rating {
  font-size: 14px;
}

.genre {
  font-size: 0.85rem;
}
}

/* 768px */
@media (max-width: 768px) {
  .similar-title {
    font-size: 2rem;
    flex-direction: column;
    text-align: center;
  }

  .best {
    font-size: 6rem;
  }

  .movies-for-you {
    font-size: 3.5rem;
  }

  .movies-grid {
    grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
  }

  .movie-card {
    max-width: 250px;
  }

  .movie-title {
    font-size: 14px;
  }

  .movies-for-you {
    white-space: normal;
    line-height: 1.2;
  }

  .rating {

```

```
        font-size: 12px;
    }

    .genre {
        font-size: 0.8rem;
    }
}

/* 480px */
@media (max-width: 480px) {
    .similar-title {
        font-size: 1.5rem;
    }

    .best {
        font-size: 4rem;
    }

    .movies-for-you {
        font-size: 3rem;
    }

    .movies-grid {
        grid-template-columns: 1fr;
        padding: 0 1rem;
    }

    .movie-card {
        max-width: 100%;
        height: auto;
    }

    .movie-title {
        font-size: 12px;
    }

    .rating {
        font-size: 10px;
    }

    .genre {
        font-size: 0.75rem;
    }
}

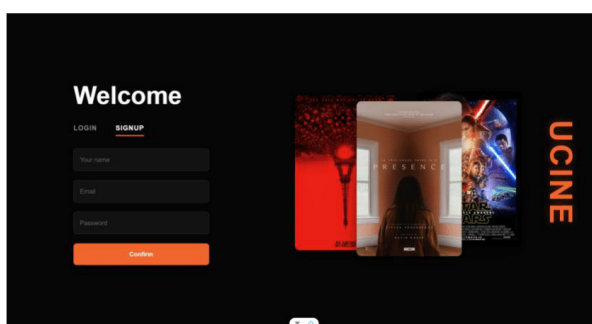
</style>
```

ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА

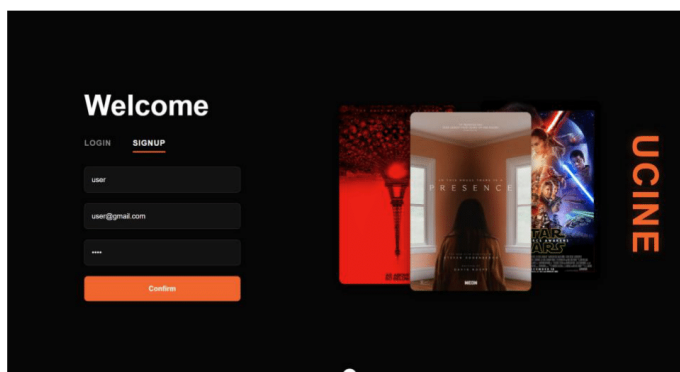
UCINE – це система рекомендацій фільмів нового покоління, розроблена для кіномереж, кінодистриб'юторів та контент-стратегів. Ми допомагаємо вам прогнозувати успіх касових зборів, оптимізувати графіки показів та приймати впевнені рішення на основі даних протягом усього життєвого циклу фільму.

1. Реєстрація

- 1) **Відкрийте сторінку UCINE**
- 2) **Оберіть вкладку SIGNUP.** Переконайтесь, що ви на вкладці SIGNUP, а не LOGIN. Вона має бути підкреслена оранжевою лінією.



- 3) **Введіть ваше ім'я.** У перше поле, під написом "Your name", введіть своє ім'я.
- 4) **Введіть електронну пошту.** У друге поле впишіть свій email (він має бути дійсним).
- 5) **Створіть пароль.** У третє поле введіть бажаний пароль. Рекомендується використовувати комбінацію літер, цифр та символів для безпеки.
- 6) **Натисніть кнопку Confirm.** Після заповнення всіх полів натисніть помаранчеву кнопку **Confirm** для завершення реєстрації.

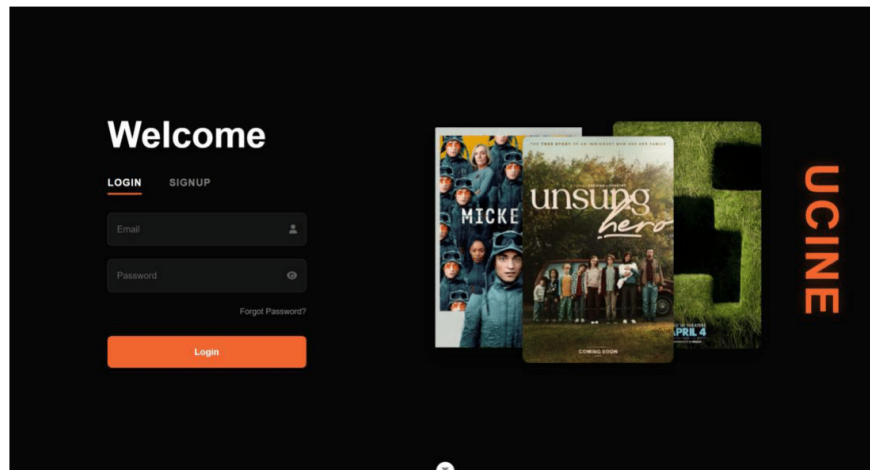


- 7) **У разі правильного виконання всіх дій обліковий запис створено.** Користувача направлено на головну сторінку системи.
- 8) **У випадку неправильного введення даних повторіть дії.**

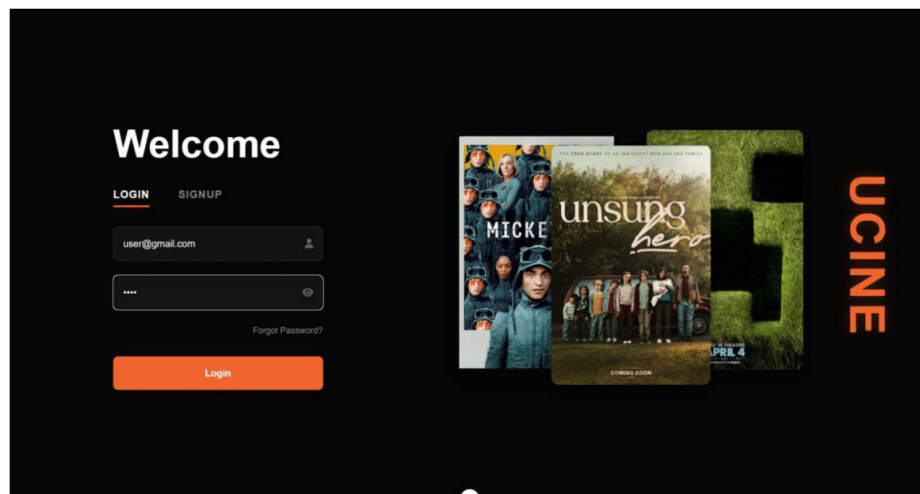
Рисунок В.1 – Реєстрація користувача

2. Авторизація

- 1) **Перейдіть на сайт UCINE.** Відкрийте веб-сторінку, де розміщена форма входу (як на скріншоті).
- 2) **Оберіть вкладку LOGIN.** Переконайтесь, що активна вкладка LOGIN (вона підкреслена оранжевою лінією).



- 3) **Введіть електронну пошту.** У перше поле введіть email, який ви використовували при реєстрації.
- 4) **Введіть пароль.** У друге поле введіть ваш пароль.
- 5) **Натисніть кнопку Login.** Коли все заповнено, натисніть помаранчеву кнопку Login для входу до свого облікового запису.



- 6) **У разі правильного виконання всіх дій авторизацію виконано.** Користувача направлено на головну сторінку системи.
- 7) **У випадку неправильного введення даних повторіть дії.**

Рисунок В.2 – Авторизація користувача

3. Головна сторінка

- 1) Після виконання авторизації чи реєстрації користувача **направлено на головну сторінку**. На ній користувач може ознайомитись з інформацією про систему, її призначення, функціонал та переваги, що вона надає.



- 2) Справа розміщено фільми з високим рейтингом, які можуть бути цікавими для цільової аудиторії кінотеатру. Для перегляду всіх карток наведіть на карусель з фільмами, зажіміть праву кнопку миші та свайпніть справа на ліво.

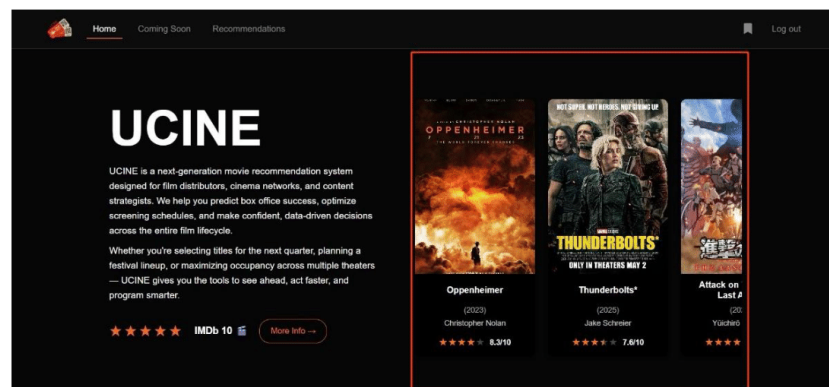


Рисунок В.3 – Головна сторінка

4. Перегляд інформації про фільм

- 1) Клікніть по постеру або назві фільму.
- 2) Ознайомтесь з інформацією про фільм

На сторінці фільму ви побачите:

- Назву, жанри
- Рік
- Тривалість
- Рейтинг
- Країну-виробника
- Режисера
- Акторів
- Опис сюжету

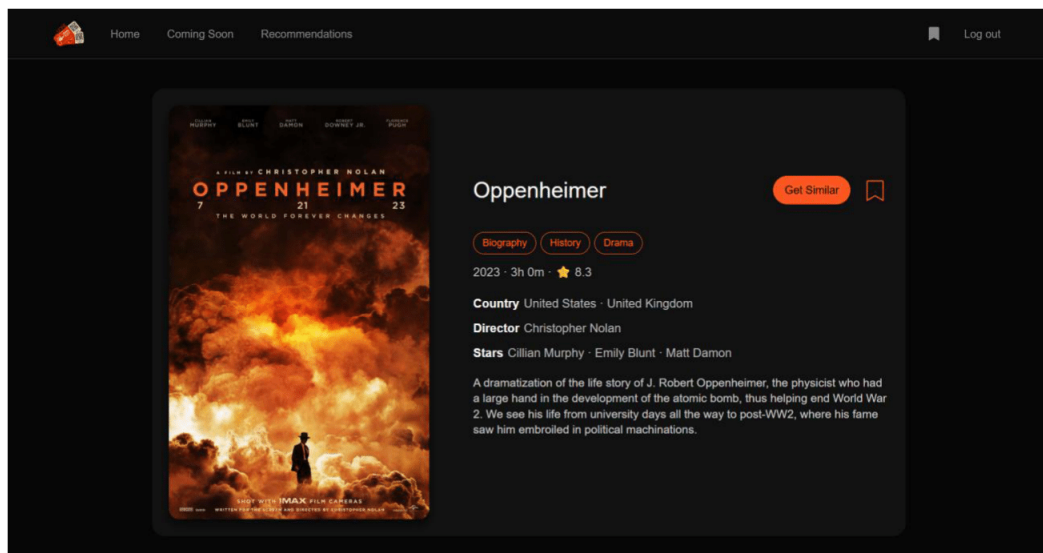
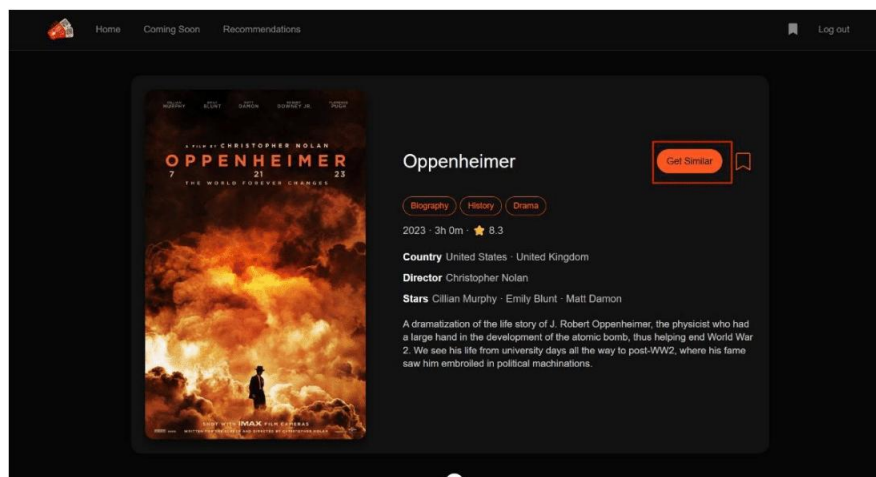


Рисунок В.4 – Сторінка фільму

5. Знайти подібні фільми

- 1) Клікніть по постеру або назві фільму для переходу на його сторінку. Натисніть кнопку “Get Similar”.



- 2) Ознайомтесь з фільми, які подібні на той, що вами обрано.

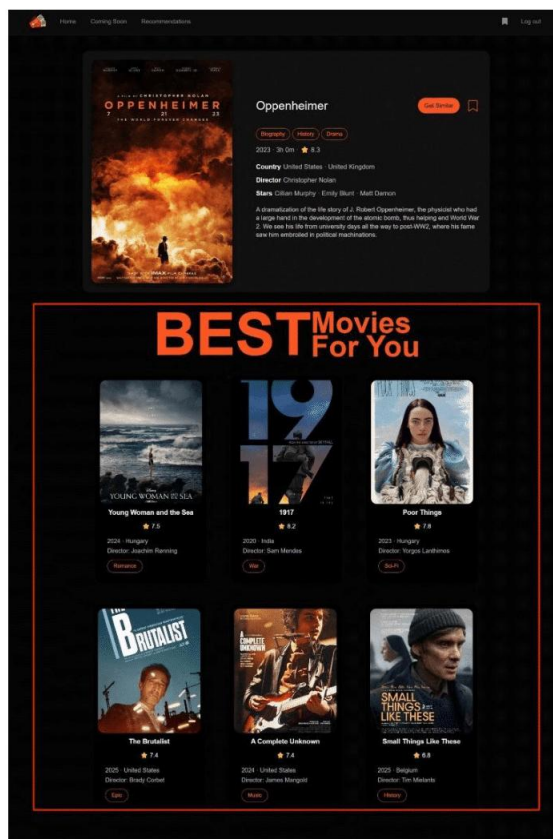
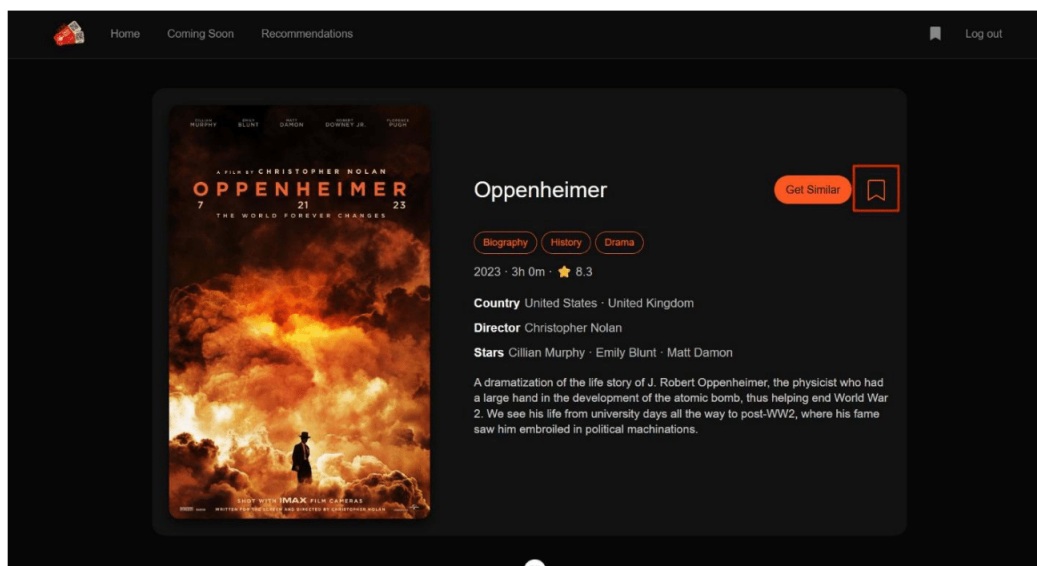


Рисунок В.5 – Рекомендації на основі загальної подібності

6. Збереження фільму

- 1) Клікніть по постеру або назві фільму для переходу на його сторінку. Натисніть на іконку “Save”.



- 2) Після настикання на іконку “Save” вона змінить свій колір на помаранчевий. Фільм збережено

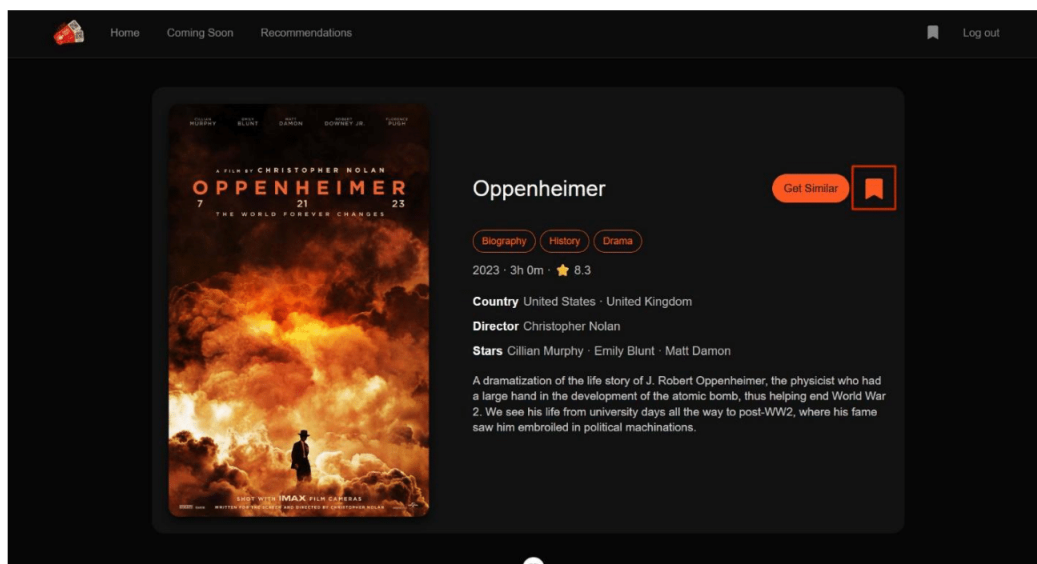
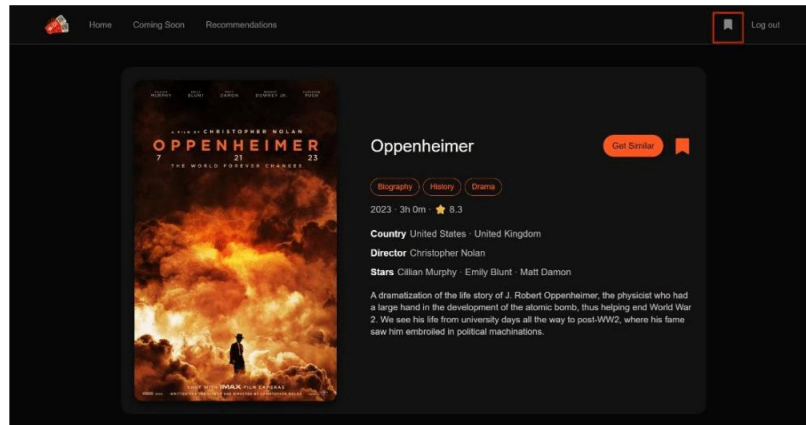


Рисунок В.6 – Збереження фільму. Сторінка фільму

- 3) Для перегляду списку збережених фільмів натисніть на іконку “Save” у правому верхньому куті екрану.



- 4) Перегляньте список фільмів, які зберегли.

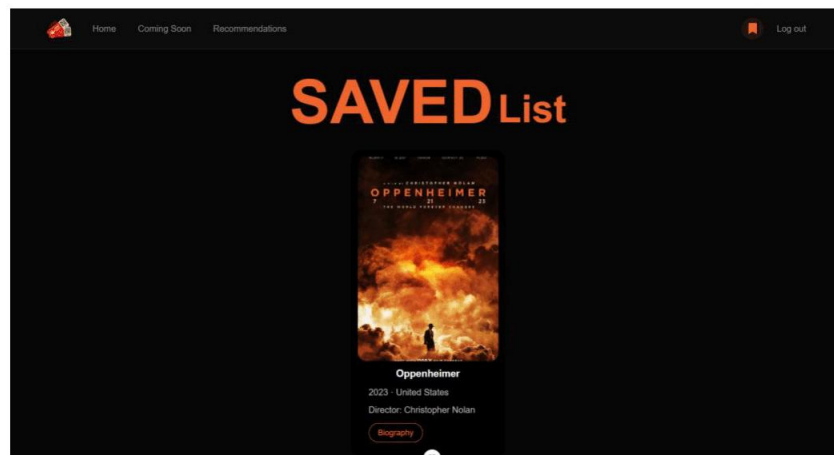
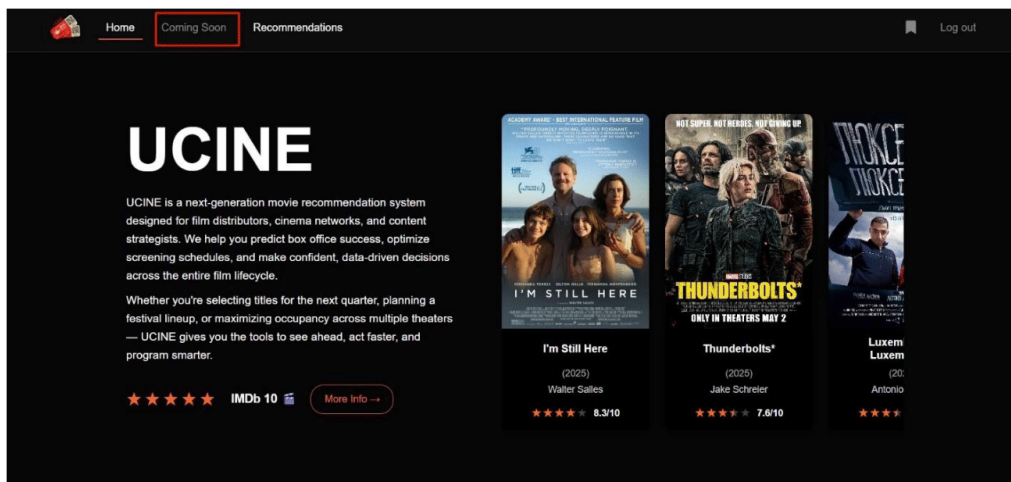


Рисунок В.7 – Збереження фільму. Список збережень

7. Прем'єри

- 1) Для переходу на сторінку прем'єр натисніть на кнопку «Coming Soon» у лівому верхньому куті.



- 2) Ознайомтесь з інформацією про актуальні новинки кінопрокату та майбутні релізи.

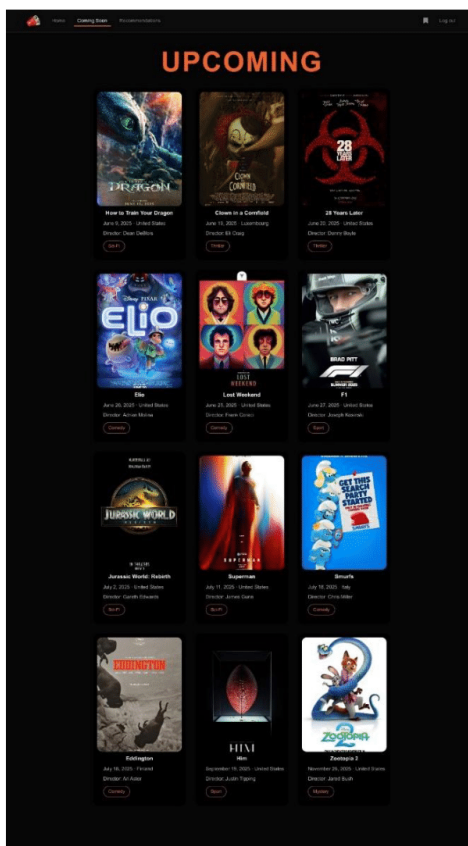
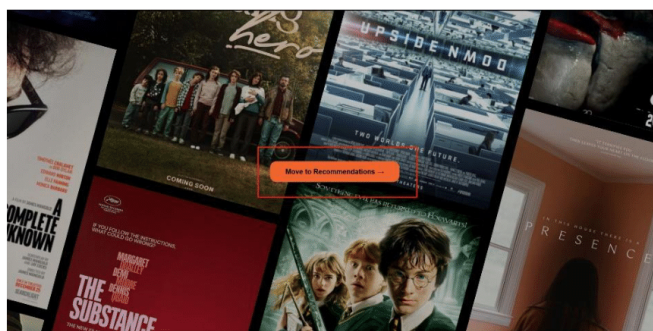
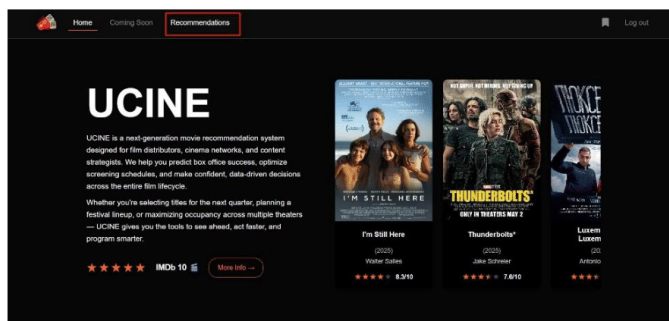


Рисунок В.8 – Сторінка прем'єр

8. Рекомендації за категоріями

- 1) **Перейдіть на сторінку Рекомендацій**, натиснувши на кнопку «Recommendations» у лівому верхньому екрані, або натиснувши «Move to Recommendations →» на головній сторінці.



- 2) **Заповніть хоча б один із плейсхолдерів:**

- Введіть назву фільму у поле "Movie Title", якщо необхідна рекомендація, що базується на конкретному фільмі
- Оберіть рік фільму
- Оберіть країну (наприклад, *Ukraine*)
- Оберіть актора (наприклад, *Emma Stone*)
- Оберіть режисера (наприклад, *Emma Tammi*)
- Оберіть жанр (наприклад, *Action*, *Comedy*, *Animation* тощо)

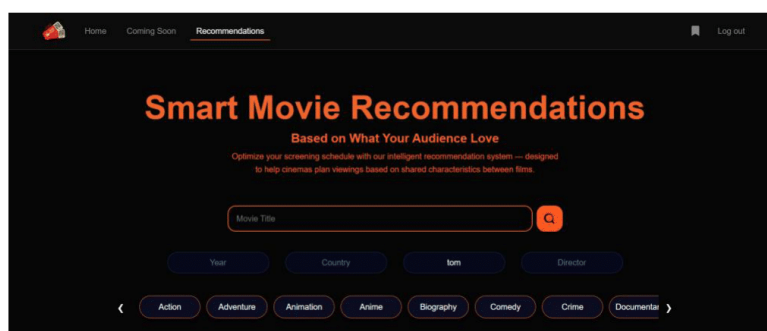
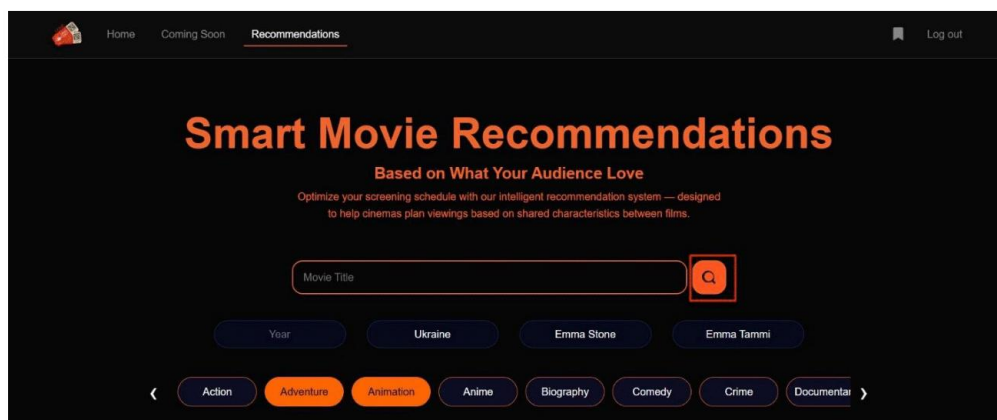


Рисунок В.9 – Рекомендації за заданими критеріями

- 3) Натисніть на значок лупи праворуч від поля для введення назви, щоб запустити пошук.



- 4) На екрані з'являться фільми, які відповідають обраним характеристикам.

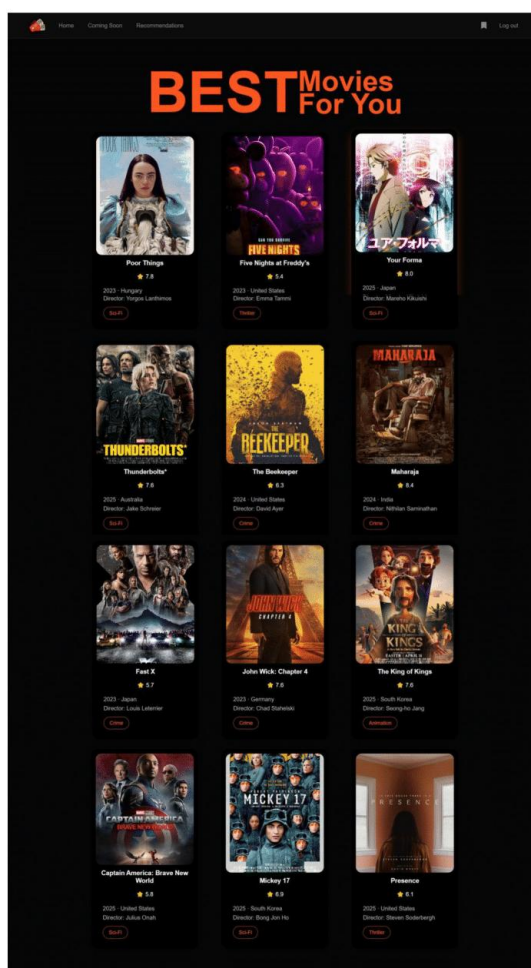
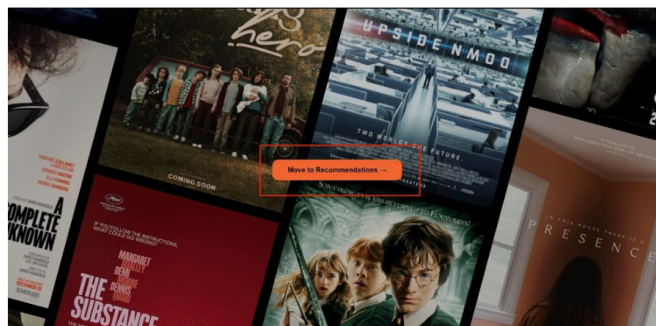
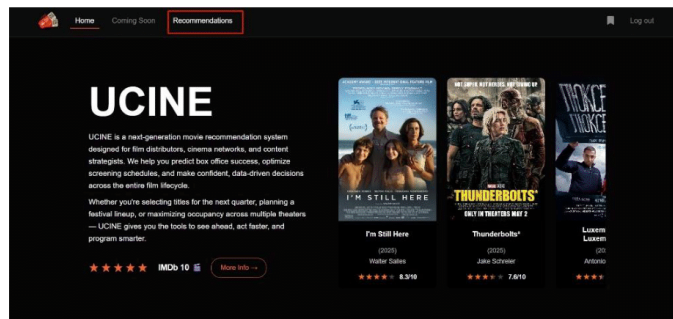


Рисунок В.10 – Рекомендації за заданими критеріями

9. Рекомендацій за вибіркою фільмів

- 1) **Перейдіть на сторінку Рекомендацій**, натиснувши на кнопку «Recommendations» у лівому верхньому екрані, або натиснувши «Move to Recommendations →» на головній сторінці.



- 2) **Оберіть ті фільми**, що попередньо були цікаві цільовій аудиторії кінотеатру, натиснувши на них.
- 3) Якщо **користувач помилково обрав невірний фільм**, то необхідно натиснути на нього повторно, тим самим вилючивши його з обраних.

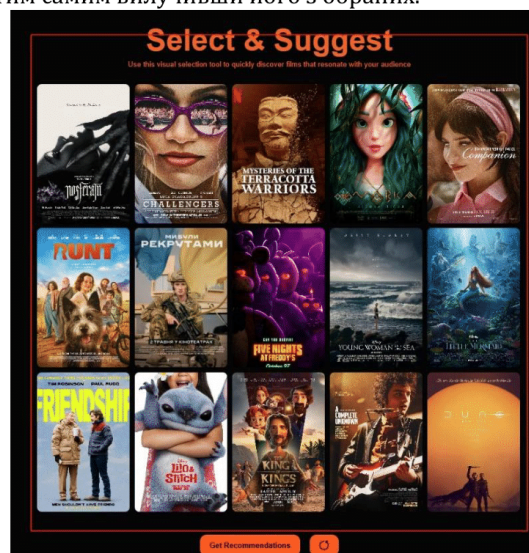
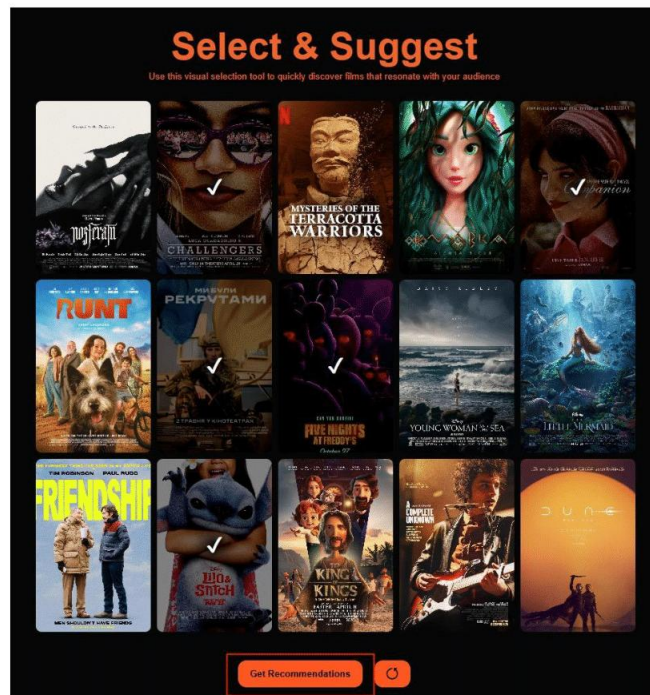


Рисунок В.11 – Рекомендації за вибіркою фільмів

4) Після вибору фільмів натисніть на кнопку «Get Recommendations»



5) На екрані з'являться фільми, побудовані на вибірці обраних фільмів.

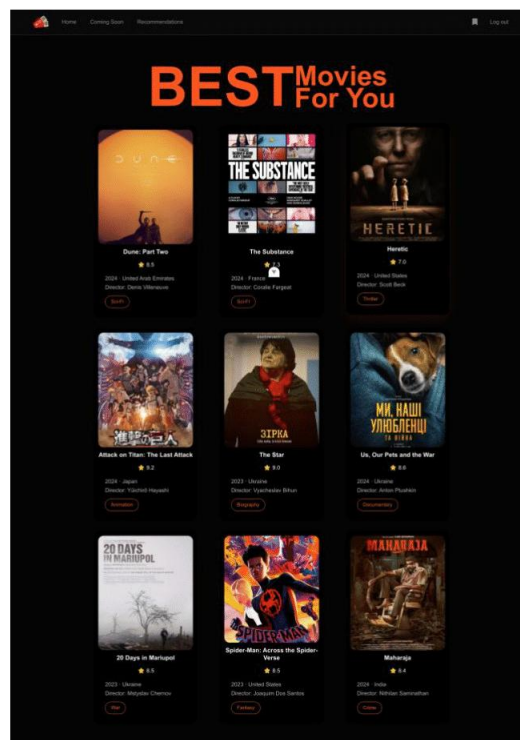


Рисунок В.12 – Рекомендації за вибіркою фільмів

- 6) Якщо у представленій вибірці фільмів немає таких, що користувались популярністю у кінотеатрі, скористайтесь функцією оновлення вибірки, натиснувши на відповідний елемент.

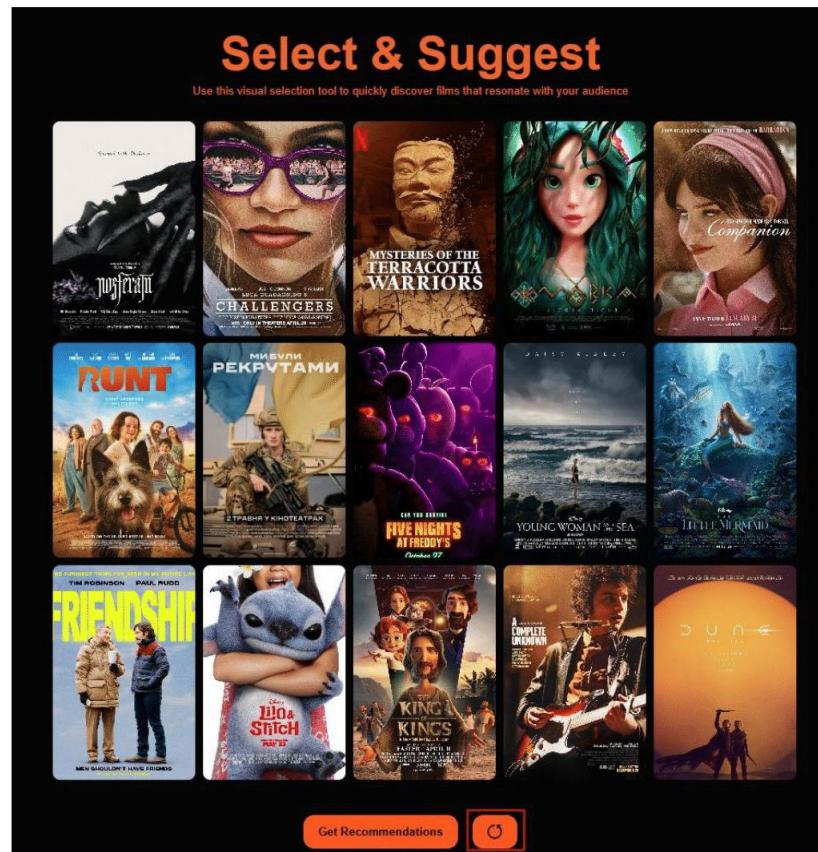


Рисунок В.13 – Рекомендації за вибіркою фільмів

Бицька В.В., студентка
Київський національний економічний
університет імені Вадима Гетьмана
vika.bytska@ukr.net

ГРАФОВІ БАЗИ ДАНИХ ЯК ОСНОВА ДЛЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У КОНТЕКСТІ АВТОМАТИЗОВАНОГО ІНЖИНІРИНГУ

У сучасному світі спостерігається безпрецедентне зростання обсягів даних. За даними статті «Amount of Data Created Daily», кількість контенту, який генерується щодня, досяг позначки 402.74 мільйонів терабайт та продовжує невпинно зростати, створюючи виклики для систем, що мають на меті аналізувати дані та виводити закономірності, враховуючи складні взаємозв'язки між об'єктами даних [2]. Одночасно з цим, спостерігається підвищення вимог користувачів щодо персоналізації контенту та послуг. У зв'язку з цим, рекомендаційні системи стали невід'ємною частиною сучасних цифрових сервісів та платформ. Серед технологічних рішень особливо виділяються графові бази даних, що є ефективною технологією для побудови рекомендаційних систем, надаючи потужні інструменти для моделювання зв'язків та структур даних у вигляді графів.

Метою дослідження є теоретичне та практичне обґрунтування переваг використання графових баз даних як фундаменту для побудови сучасних рекомендаційних систем в умовах автоматизованого інжинірингу. Дослідження спрямоване на виявлення обмежень традиційних підходів шляхом розкриття потенціалу графового представлення даних для моделювання складних взаємозв'язків та виявлення прихованих закономірностей

Традиційні підходи до побудови рекомендаційних систем, засновані на реляційних базах даних та класичних алгоритмах колаборативної фільтрації, зіштовхуються з низкою обмежень у сучасних вимогах. Причиною є високий рівень структурованості схем реляційних баз даних, що обмежує можливості моделювання багатовимірних зв'язків між об'єктами. У зв'язку з тим, що всі елементи розділені на різні таблиці, для отримання рекомендацій необхідне використання багаторівневих JOIN-операцій, що знижують швидкість обробки запитів до системи зі зростанням глибини зв'язків. Ще однією проблемою є можливість масштабування рекомендаційних систем, що побудовані на основі реляційних баз даних. Вертикальне масштабування являє собою збільшення обчислювальної потужності одного сервера, але при цьому робить цей метод масштабування економічно неефективним, зважаючи на кількість об'єктів та зв'язків, які зберігаються у рекомендаційних системах. Крім цього, значні обсяги пам'яті, яких потребують сучасні алгоритми також можуть перевищувати можливості одного сервера. Горизонтальне масштабування (шардинг) передбачає розподіл даних та обчислювального навантаження між кількома серверами, що працюють паралельно. Такий метод створює виклики, пов'язані із забезпеченням цілісності зв'язків між сутностями, які розподілені на різних шардах, та може мати негативні наслідки для надання релевантних рекомендацій.

У порівнянні з реляційними, графові бази даних пропонують ряд переваг, які є критичними для реалізації ефективної рекомендаційної системи. Однією з них є простота схем об'єктів та зв'язків, що зображені у вигляді вершин та ребер графа і не обтяжені складними об'єднаннями об'єктів за допомогою зовнішніх ключів. Наступною перевагою є гнучкість графових моделей, що дозволяє легко додавати, змінювати або видаляти вузли та ребра без значних змін у схемі [3]. Реалізація подібних рішень є складним процесом у реляційних системах, де дочірні таблиці залежать від батьківських, ускладнюючи будь-які модифікації даних. Досягнення високого рівня масштабованості відбувається різними способами: наприклад, графова система Neo4j використовує індекс-вільну суміжність, що дозволяє швидко переходити від одного вузла до іншого без необхідності індексного пошуку, забезпечуючи продуктивність при обході графа. У графовій моделі ArangoDB використовується інший підхід, адже вузли зберігаються як

документи. Ребра графа теж збережено як окремі документи зі спеціальними атрибутами, що вказують на початковий та кінцевий вузли. Подібна архітектура є ефективним методом для забезпечення горизонтального масштабування системи, розподіляючи дані між серверами [1].

Рекомендаційні системи, за основу яких взято графову базу даних, можна побудувати використовуючи прості графові алгоритми, серед яких PageRank, Similarity та Community Detection. Для реалізації алгоритму PageRank об'єкти системи, що представлені вузлами графа, об'єднані зв'язками, що формуються на основі спільних характеристик, вподобань однією групою користувачів тощо. Основний принцип роботи PageRank полягає у визначенні того, наскільки «важливим» є кожен вузол у системі. Важливість вузла залежить від двох факторів: кількості інших вузлів, що мають зв'язок з даним вузлом та вага цих зв'язків. Об'єкт вважається більш важливим, якщо на нього посиляється багато інших об'єктів, особливо якщо ці об'єкти самі по собі мають високу значущість у системі. У контексті рекомендаційних систем PageRank допомагає виявити найбільш значущі об'єкти, які потім будуть рекомендовані користувачам.

Алгоритми Similarity в графових рекомендаційних системах вимірюють схожість між вузлами на основі їхніх властивостей або структурних характеристик у графі. Основою для вимірювання можуть бути як метадані, які характеризують окремий об'єкт (назва, ціна, опис), так і поведінкові дані користувача (оцінки, коментарі, історії переглядів). Одними з ключових методів для реалізації алгоритму є методи косинусної подібності та алгоритм Жаккара.

Використання алгоритму Community Detection у рекомендаційних системах здійснює процес пошуку груп користувачів або об'єктів, які мають тісніші зв'язки між собою, ніж з іншими елементами системи. Основна логіка полягає в представленні користувачів та об'єктів як вузлів графа, де зв'язки між ними відображають взаємодії (наприклад, користувач купив товар або вподобав фільм). На основі цього алгоритм аналізує структуру графа, щоб знайти кластери з більш щільними внутрішніми зв'язками. Таким чином, Community Detection допомагає знаходити нетривіальні зв'язки між користувачами та об'єктами, що підвищує точність та персоналізацію рекомендацій.

Кожен з алгоритмів пропонує свої переваги в залежності від предметної області та типу рекомендаційної системи. Для забезпечення вищої точності наданих рекомендацій, варто розглянути комбінацію двох алгоритмів. Комбінація алгоритмів PageRank та Similarity, які виконують хоч і різні, але взаємодоповнюючі функції, може значно підвищити точність наданих рекомендацій.

Сучасний рівень розвитку графових баз даних робить їх однією з найефективніших технологій для реалізації рекомендаційних систем, демонструючи численні переваги у порівнянні з традиційними реляційними моделями. Застосування базових графових алгоритмів та можливість їх комбінації дозволяє реалізувати функціональні системи для надання рекомендацій, адаптувавши їх до специфіки та вимог різноманітних предметних областей.

Список використаних джерел

1. Порівняння графових баз даних Neo4j, ArangoDB та OrientDB: яку обрати?. EPAM. URL: <https://careers.epam.ua/blog/comparison-of-graph-databases-neo4j-arangodb-orientdb>.
2. Duarte F. Amount of Data Created Daily (2025). *Exploding Topics*. URL: <https://explodingtopics.com/blog/data-generated-per-day>.
3. Jatin. What is a Graph Database? Use Cases and Advantages | Decube. *Decube: Data Trust Platform for the AI Era*. URL: <https://www.decube.io/post/graph-database-concept>.

Науковий керівник: Ситник Н.В., к.е.н., доцент.

Звіт подібності

метадані

Назва організації
Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок
Розроблення рекомендаційної системи кінотеатру на основі графової бази даних

Автор
Науковий керівник / Експерт
Бицька Вікторія ВіталіївнаСитник Н.В.

підрозділ
кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		26
Інтервали		0
Мікропробіли		9
Білі знаки		0
Парафрази (SmartMarks)		48

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз		Копір тексту
ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	98 0.64 %
2	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3923cf3566a2/download	43 0.28 %
3	https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram/	41 0.27 %
4	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	34 0.22 %
5	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3923cf3566a2/download	29 0.19 %

6	https://stanislava-shelin.webnode.com.ua/news/tema-5/	22 0.14 %	
7	https://kneu.edu.ua/userfiles/Faculty_of_Information_Systems_and_Technology/kaf%20ise/Metod_rekom_en_6_050101.doc	20 0.13 %	
8	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	16 0.10 %	
9	https://kneu.edu.ua/userfiles/Faculty_of_Information_Systems_and_Technology/kaf%20ise/Metod_rekom_en_6_050101.doc	16 0.10 %	
10	https://ir.kneu.edu.ua/server/api/core/bitstreams/38362e99-7f5f-449c-879b-cb24b773a81d/content	16 0.10 %	
з бази даних RefBooks (0.00 %)			
ПОРЯДКОВИЙ НОМЕР		ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
з домашньої бази даних (0.04 %)			
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	Інформаційна система автоматизації процесів малого підприємства на прикладі кав'ярні 6/11/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	6 (1) 0.04 %	
з програми обміну базами даних (0.26 %)			
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	Сопрун О.В._CACM-21_робота.docx 10/31/2022 National University "Lviv Politechnika" (NULP2)	30 (3) 0.19 %	
2	2014_705010101_Okis_Andrii_Vasylovych_20194 10/25/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	5 (1) 0.03 %	
3	Вдосконалення менеджменту та адміністрування інноваційною діяльністю в закладах СОЗ.docx 11/29/2022 Sumy National Agrarian University (SNAU)	5 (1) 0.03 %	
з Інтернету (3.83 %)			
ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	148 (3) 0.96 %	
2	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3923cf3566a2/download	99 (4) 0.64 %	
3	https://kneu.edu.ua/userfiles/Faculty_of_Information_Systems_and_Technology/kaf%20ise/Metod_rekom_en_6_050101.doc	51 (3) 0.33 %	
4	https://stanislava-shelin.webnode.com.ua/news/tema-5/	50 (4) 0.32 %	
5	https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram/	41 (1) 0.27 %	
6	https://revolution.allbest.ru/programming/00694221_0.html	38 (3) 0.25 %	
7	https://ir.kneu.edu.ua/bitstreams/22acc4f6-0461-4d18-a9a4-ff2fbd8c10ff/download	26 (3) 0.17 %	
8	https://conf.ztu.edu.ua/wp-content/uploads/2016/06/12.pdf	20 (2) 0.13 %	