

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія ранжування відео на стрімінговій платформі»

Виконала: студент 2-го курсу, групи
ЗКН-22м спеціальності 122

«Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Замкова Т. Д.
(прізвище та ініціали)

Керівник: к.т.н., ст.викл. кафедри КН

Петришин С. І.
(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: к.т.н., доц. кафедри АПТ

Барабан М. В.
(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А. А.


(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця ВНТУ - 2023 рік

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

ЗАТВЕРДЖУЮ
Завідувач кафедри КН

 д.т.н., проф. Яровий А.А.

(підпис)

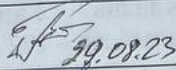
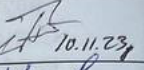
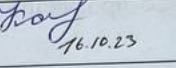
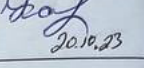
« 29 » 08 2023 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Замковій Тетяні Дмитрівні
(прізвище, ім'я, по батькові)

1. Тема роботи «Інформаційна технологія ранжування відео на стрімінговій платформі»
керівник роботи к.т.н., ст. викл. каф. КН, Петришин С. І.
затверджені наказом вищого навчального закладу від «18» 09 2023 року №247
2. Строк подання студентом роботи 13.11.2023р.
3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтована; формат файлів – xlsx; мінімальна кількість категорій – 3 од.; мінімальна кількість записів в базі даних – 40 од.; тип інтерфейсу користувача інтуїтивно-зрозумілий.
4. Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології ранжування відео на стрімінговій платформі, моделювання інформаційної технології ранжування відео на стрімінговій платформі, структурна організація та особливості програмної реалізації інформаційної технології ранжування відео на стрімінговій платформі, економічна частина, висновки, список використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): структурна схема інформаційної технології; загальна структурна схема; схема алгоритму функціонування модуля ранжування інформаційної технології ранжування відео на стрімінговій платформі; діаграма діяльності системи при встановленні з'єднання з базою даних; UML-діаграма класів; приклад роботи програми; результати роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3 Спеціальна частина	Петришин С. І. к.т.н., ст. викладач каф. КН	 29.09.23	 10.11.23
4 Економічна частина	Кавецький В. В., к.е.н., доц. каф. ЕПВМ	 16.10.23	 20.10.23

7. Дата видачі завдання 29.08 2023 року

КАЛЕНДАРНИЙ ПЛАН

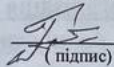
№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування доцільності розробки інформаційної технології ранжування відео на стрімінговій платформі	07.09.23 - 07.09.23	Розділ 1
2	Моделювання інформаційної технології ранжування відео на стрімінговій платформі	08.09.23 - 24.09.23	Розділ 2
3	Структурна організація та особливості програмної реалізації інформаційної технології	25.09.23 - 15.10.23	Розділ 3
4	Підготовка економічної частини	16.10.23 - 20.10.23	Розділ 4
5	Апробація та/або впровадження результатів дослідження	21.10 - 05.11.23	Тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	Пояснювальна записка, графічний матеріал, презентація

Студент


(підпис)

Замкова Т. Д

Керівник роботи


(підпис)

Петришин С. І.

АНОТАЦІЯ

УДК 004.832

Замкова Т. Д. Інформаційна технологія ранжування відео на стрімінговій платформі. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 123 с.

На укр. мові. Бібліогр.: 43 назв; рис.: 19; табл. 16.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології ранжування відео на стрімінговій платформі. Виконано аналіз сучасних програм-аналогів, які використовуються для ранжування відео на стрімінговій платформі, наведено коротку порівняльну характеристику знайдених програм-аналогів, досліджено методи, які можуть бути використані для реалізації поставленої задачі. Проведено обґрунтування вибору методу для ранжування відео на стрімінговій платформі на основі методу RankBoost. Розроблено алгоритм ранжування відео на стрімінговій платформі та відповідне програмне забезпечення на мові програмування Java, в середовищі IntelliJ IDEA. Аналіз роботи програмного забезпечення показав підвищення точності ранжування відео на стрімінговій платформі.

Графічна частина складається з 8 плакатів із результатами моделювання.

У розділі економічної частини здійснено оцінювання комерційного потенціалу розробки інформаційної технології ранжування відео на стрімінговій платформі, проведено оцінювання комерційного потенціалу розробки, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, розраховано період окупності.

Ключові слова: інформаційна технологія, ранжування, стрімінгова платформа, відео.

ABSTRACT

Zamkova T. D. Information technology of video ranking on a streaming platform. Master's work in specialty 122 – Computer science. Vinnytsia: VNTU, 2023. 123 p.

In Ukrainian language. Bibliogr.: 43 titles; Figures: 19; Table 16.

This master's qualification work is devoted to the the development of information technology for video ranking on a streaming platform. An analysis of modern analog programs used for ranking videos on a streaming platform is performed, a brief comparative description of the found analog programs is given, and the methods that can be used to implement the given task are investigated. The justification for choosing a method for ranking videos on a streaming platform based on the RankBoost method is carried out. The video ranking algorithm on the streaming platform and the corresponding software in the Java programming language, in the IntelliJ IDEA environment, were developed. Analysis of the software's performance showed an increase in the accuracy of video ranking on the streaming platform.

In the section of the economic part, the commercial potential of the development of information technology for video ranking on the streaming platform was carried out, the commercial potential of the development is assessed, the costs of research and implementation of the results are predicted, and the payback period is calculated.

The graphic part consists of 8 posters with modelling results.

Keywords: information technology, ranking, testing, streaming platform, video.

ЗМІСТ

ВСТУП	4
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ВІДЕО НА СТРІМІНГОВІЙ ПЛАТФОРМІ	8
1.1 Дослідження принципів ранжування відео на стрімінгових платформах	8
1.2 Визначення факторів, які впливають на ранжування відео.....	9
1.3 Існуючі моделі та методи, що застосовуються для реалізації поставленої задачі.....	15
1.4 Аналіз відомих програмних рішень для ранжування відео на стрімінговій платформі.....	21
1.5 Формулювання вимог та постановка задачі.....	28
1.6 Висновок до розділу 1	30
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ВІДЕО НА СТРІМІНГОВІЙ ПЛАТФОРМІ	31
2.1 Обґрунтування вибору методу ранжування відео на стрімінговій платформі.	31
2.2 Розробка математичної моделі ранжування відео на стрімінговій платформі.	33
2.3 Проектування структури та алгоритму функціонування інформаційної технології ранжування відео на стрімінговій платформі.....	35
2.4 Обґрунтування вибору бази даних для зберігання даних інформаційної технології ранжування відео на стрімінговій платформі.....	42
2.5 Висновок до розділу 2	45
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ВІДЕО НА СТРІМІНГОВІЙ ПЛАТФОРМІ.....	46
3.1 Обґрунтування вибору мови та середовища програмування	46

3.2 Обґрунтування вибору бібліотек та фреймворків для інформаційної технології ранжування відео на стрімінговій платформі.....	49
3.3 Опис процесу програмної розробки.....	56
3.4 Опис класів і функцій інформаційної технології ранжування відео на стрімінговій платформі.....	60
3.5 Тестування та аналіз результатів роботи інформаційної технології ранжування відео на стрімінговій платформі	62
3.6 Висновок до розділу 3	68
4 ЕКОНОМІЧНА ЧАСТИНА	69
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	69
4.2 Розрахунок узагальненого коефіцієнта якості розробки	73
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	75
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	88
4.5 Висновок до розділу 4	92
ВИСНОВКИ.....	93
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	95
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	99
Додаток Б (обов'язковий) Лістинг програми	100
Додаток В (обов'язковий) Ілюстративна частина.....	113
Додаток Г (довідниковий) Інструкція користувача	120

ВСТУП

Актуальність теми. Сучасний розвиток суспільства безпосередньо пов'язаний з необхідністю збирати, обробляти та передавати величезні обсяги інформації. Цей процес перетворює інформацію в цінний товар, що зазвичай має велику вартість. Такий підхід призвів до глобального переходу від індустріального суспільства до інформаційного. Поява всесвітньої павутини призвело до значного зростання міжнародного спілкування в різних сферах людського життя. Сучасне суспільство насичене інформаційними потоками, які природно потрібно обробляти. Саме тому воно не може нормально функціонувати без використання інформаційних технологій, енергетики, транспорту та хімічних технологій.

Ранжування відео важливо для забезпечення задоволення користувачів та забезпечення оптимального досвіду перегляду. Перш за все, це означає показ користувачам найбільш релевантних та цікавих відео згідно з їхніми інтересами, персональними вподобаннями та іншими факторами. Ранжування також може бути використано для просування популярного або нового відео контенту, підвищення залученості та збільшення кількості переглядів.

Метою ранжування відео на стрімінговій платформі є забезпечення належного збалансованого розподілу відео контенту та забезпечення максимальної релевантності та задоволення користувачів.

Інформаційна технологія ранжування відео на стрімінговій платформі є надзвичайно актуальною, оскільки стрімінгові платформи набувають все більшої популярності та стають важливими джерелами відео контенту для мільйонів користувачів по всьому світу. Користувачі очікують, щоб стрімінгова платформа надавала їм персоналізований досвід, показуючи відео, які відповідають їхнім інтересам та вподобанням. Ранжування відео на основі алгоритмів, що враховують контекст користувача, його історію перегляду та взаємодію, допомагає створювати персоналізований контент і підвищує задоволення користувачів.

Ранжування відео допомагає оптимізувати використання обмежених ресурсів стрімінгової платформи, таких як пропускна здатність мережі та потужність серверів. Шляхом показування більш релевантного відео, що має високу ймовірність перегляду, платформа може ефективніше використовувати ресурси та забезпечувати задоволення більшої кількості користувачів. В умовах зростаючої конкуренції в галузі стрімінгових платформ, ефективне ранжування відео є ключовим фактором для залучення нових користувачів та утримання існуючих. Якщо платформа може забезпечити високу якість ранжування, показуючи цікавий та релевантний контент, це дозволяє зберігати та залучати більше користувачів.

Активні дослідження в галузі ранжування відео сприяють вдосконаленню алгоритмів та методів, які використовуються на стрімінгових платформах. Це дозволяє постійно покращувати точність, персоналізацію та релевантність ранжування, щоб забезпечити максимальне задоволення користувачів.

Актуальність інформаційної технології ранжування відео на стрімінговій платформі полягає у потребі створення інструменту для досягнення успіху у сфері стрімінгових послуг, надаючи персоналізований досвід, ефективно використання ресурсів, залучення аудиторії та постійне вдосконалення алгоритмів ранжування. Так як на сьогоднішній день існує проблема, що характеризується недостатньою точністю пошуку релевантного, популярного та якісного відео, розробка інформаційної технології ранжування відео на стрімінговій платформі є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження є підвищення точності пошуку релевантного, популярного та якісного відео.

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

- провести аналіз програм-аналогів та обґрунтувати доцільності розробки інформаційної технології ранжування відео на стрімінговій платформі;
- розробити структуру інформаційної технології ранжування відео на стрімінговій платформі;
- розробити математичну модель ранжування відео на стрімінговій платформі;
- здійснити програмну реалізацію інформаційної технології ранжування відео на стрімінговій платформі;
- провести тестування програми та проаналізувати отримані результати;
- економічно обґрунтувати доцільність розробки інформаційної технології ранжування відео на стрімінговій платформі.

Об'єктом дослідження є процес ранжування відео на стрімінговій платформі.

Предметом дослідження є програмні засоби ранжування відео на стрімінговій платформі.

Методи дослідження. У роботі використано такі методи наукових досліджень: метод системного аналізу для аналізу структури інформаційної технології; методи машинного навчання; метод колаборативної фільтрації, методи контентної фільтрації, гібридні методи; метод аналізу ієрархій; методи об'єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна:

Запропоновано інформаційну технологію ранжування відео на стрімінговій платформі, яка відрізняється від існуючих удосконаленою моделлю, яка містить в собі алгоритм ранжування RankBoost з комбінуванням слабких моделей в одну сильну модель, що дозволяє покращити ранжування, збільшивши точність пошуку релевантного, популярного та якісного відео.

Практичне значення одержаних результатів полягає у такому:

1. Удосконалено модель ранжування відео на стрімінговій платформі.
2. Здійснено програмну реалізацію інформаційної технології

ранжування відео на стрімінговій платформі.

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

Особистий внесок здобувача. Результати даної магістерської кваліфікаційної роботи отримані самостійно. В публікації у співавторстві здобувачу належить дослідження перспектив інформаційної технології ранжування відео на стрімінговій платформі [1, 2].

Апробація результатів магістерської кваліфікаційної роботи. Результати досліджень було апробовано на ЛІІ науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023) м. Вінниці у 2023 р [2].

Публікації магістерської кваліфікаційної роботи. За основними результатами досліджень опубліковано тези доповіді на науково-технічних конференціях [1, 2].

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ВІДЕО НА СТРІМІНГОВІЙ ПЛАТФОРМІ

1.1 Дослідження принципів ранжування відео на стрімінгових платформах

Дослідження принципів ранжування відео на стрімінгових платформах є активною областю досліджень, оскільки ефективне ранжування відео є важливим фактором для залучення та задоволення користувачів [3]. Наукові дослідження та практичні реалізації спрямовані на вдосконалення алгоритмів ранжування з метою покращення якості рекомендацій та відображення контенту для користувачів.

Основні принципи, які вивчаються та досліджуються, включають [4]:

1. **Контентові принципи.** Врахування характеристик та властивостей відео, таких як якість, тривалість, жанр, тематика, аудиторія, мова та інші фактори, що визначають релевантність відео для користувачів. Дослідження спрямовані на розуміння, які аспекти контенту впливають на ранжування і як їх краще враховувати в алгоритмах ранжування.

2. **Контекстові принципи.** Врахування контексту користувача, такого як його історія перегляду, вподобання, географічне розташування, пристрій, час доби та інші фактори. Дослідження зосереджені на виявленні і використанні контекстуальних ознак для покращення персоналізованості та релевантності ранжування відео.

3. **Соціальні принципи.** Врахування впливу соціальних факторів на ранжування відео, таких як популярність, вподобання та взаємодія користувачів. Дослідження спрямовані на розуміння соціальної динаміки та використання відгуків, коментарів, рейтингів та інших соціальних сигналів для ранжування відео.

4. **Фактори взаємодії.** Врахування взаємодії користувача з платформою та самим відео, таких як тривалість перегляду, кількість переглядів, коментарі,

лайки, поділитися тощо. Дослідження спрямовані на розуміння, які фактори взаємодії можуть свідчити про якість та релевантність відео та як їх використовувати для ранжування.

5. Машинне навчання та штучний інтелект. Використання різних методів машинного навчання та штучного інтелекту для побудови моделей ранжування відео. Це можуть бути нейронні мережі, рекомендаційні системи, алгоритми ранжування, засновані на зважуванні факторів та інші підходи [5].

6. Персоналізація. Використання персоналізованого ранжування відео, що враховує індивідуальні інтереси, вподобання та контекст користувача. Це включає аналіз історії перегляду, демографічних даних та інших факторів для надання рекомендацій, які відповідають унікальним потребам кожного користувача.

Дослідження в галузі ранжування відео на стрімінгових платформах є активним напрямком, оскільки вони допомагають покращити персоналізацію, релевантність та задоволення користувачів, що впливає на якість використання платформи та залучення аудиторії.

1.2 Визначення факторів, які впливають на ранжування відео

Ранжування відео на стрімінговій платформі – це процес визначення порядку або позиції, в якій відео контент відображається користувачам на платформі [6]. Коли користувачі відвідують стрімінгову платформу, вони мають можливість переглядати велику кількість відео з різних категорій, жанрів, авторів тощо.

Ранжування відео на стрімінговій платформі зазвичай базується на різноманітних факторах, які впливають на визначення порядку відображення відео [7]. Наведемо кілька загальних факторів, які можуть мати вплив на ранжування відео:

– Популярність та взаємодія. Фактори, такі як кількість переглядів, лайків, коментарів та підписників, можуть використовуватися для визначення

популярності відео. Відео, які мають високі показники цих метрик, можуть мати більшу ймовірність бути відображеними вище в результатах пошуку або рекомендаціях.

- Релевантність та ключові слова. Аналіз ключових слів, тегів та метаданих відео може використовуватися для встановлення його релевантності до запиту користувача. Відео, які містять ключові слова, що відповідають запиту, можуть мати більшу ймовірність бути відображеними вище.

- Історія перегляду та інтереси користувача. Ранжування відео може враховувати історію перегляду, вподобання та взаємодію користувача з платформою. Алгоритми можуть враховувати попередні переглянуті відео, рекомендації та дії користувача, щоб встановити його особисті інтереси та пропонувати відео, які ймовірно будуть цікавими.

- Тривалість та залученість. Довжина відео та рівень залученості глядачів (наприклад, середня тривалість перегляду, відсоток переглянутої частини) можуть впливати на ранжування. Відео, які забезпечують високу залученість та забезпечують задоволення глядачам, можуть мати перевагу.

- Новизна та актуальність. Відео, що містять свіжий або актуальний контент, можуть мати вищий пріоритет в ранжуванні. Нові релізи, трендові теми або актуальні події можуть залучати більше уваги та бути більш видимими на платформі.

Ці фактори можуть варіюватися залежно від конкретної стрімінгової платформи та її алгоритмів ранжування. Комбінація цих факторів допомагає визначити порядок відображення відео на стрімінговій платформі з метою забезпечення максимально релевантного та привабливого контенту для користувачів.

Давайте розглянемо вплив факторів на ранжування відео на прикладі платформи YouTube [9]. Для успішного продвиження відео на YouTube важливо розуміти та враховувати фактори, що визначають його рейтинг. Особа, яка володіє цією інформацією, може забезпечити виведення відео в ТОП-10 або навіть на перше місце за конкретним запитом.

Перший важливий фактор – це коментарі під відео. Дослідження, проведене Браяном Діном [10], показало, що коментарі мають значний вплив на позицію відео в результатах пошуку. Чим більше користувачів висловлюють свої думки під відео, тим вища ймовірність його включення в ТОП.

Розглядаючи коментарі до відео у першій десятці, видно, що їх кількість може досягати тисяч або навіть десятків тисяч. Це стає важливою метою для досягнення успіху. На графіку наведено відображення того, як кількість коментарів користувачів під відео впливає на його позицію (рисунок 1.1).

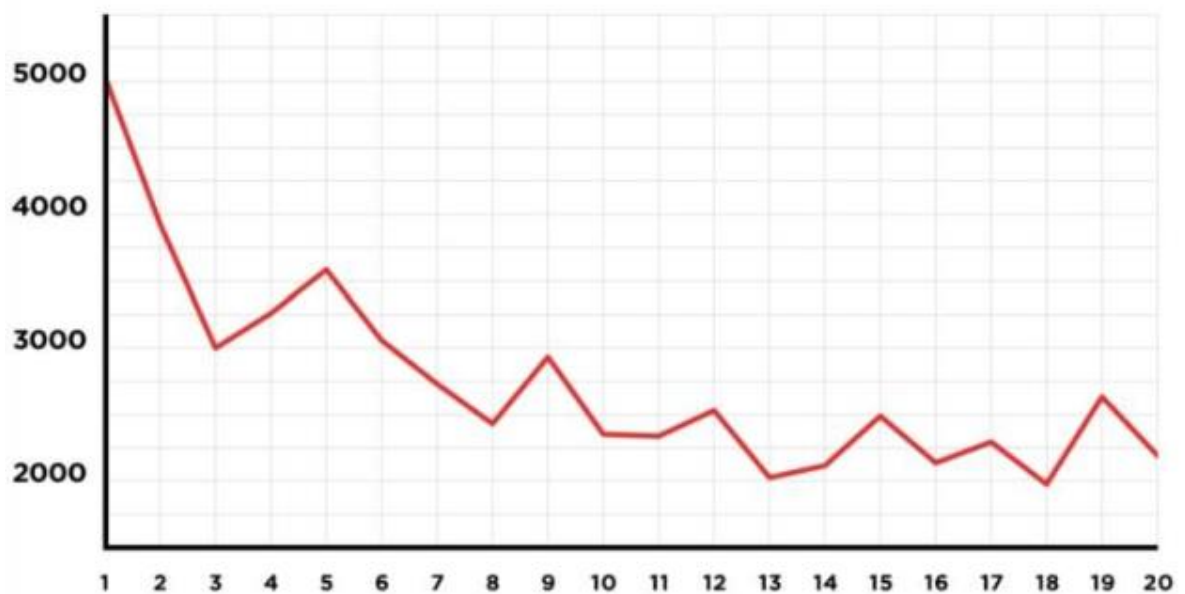


Рисунок 1.1 – Залежність позиції відео від кількості коментарів

Незважаючи на певні зміни в алгоритмах оцінки на YouTube, кількість переглядів залишається важливим фактором. Особливо вагомим є той факт, які перегляди отримує відео протягом перших двох годин після публікації (при умові, що канал не відвідується, то в наступні 2-7 днів).

Важливо враховувати, що шахрайство не призводить до покращення рейтингу. YouTube враховує лише реальні перегляди, лайки та коментарі. Таким чином, відео, яке спробує штучно накрутити ці показники, не тільки не потрапить у ТОП-рейтинг, але й може бути повністю заблоковане.

Також слід зазначити, що кількість переглядів не завжди є гарантією високого рейтингу. Відео з великою кількістю переглядів, але низькою якістю, може не вийти вперед перед відео, яке відзнято професійно та має якісний монтаж.

Лайки також впливають на позицію відео в стрічках YouTube. Проте важливо враховувати, що без реальних переглядів, довгого вмісту та коментарів, просто кількість лайків не забезпечить підняття відео вгору. Це досягається лише в комплексі.

Графік нижче вказує, що відео з 25-55 тисячами лайків посідали перші 10 місць, тоді як інші відео мали менш вигідні позиції (рисунок 1.2).

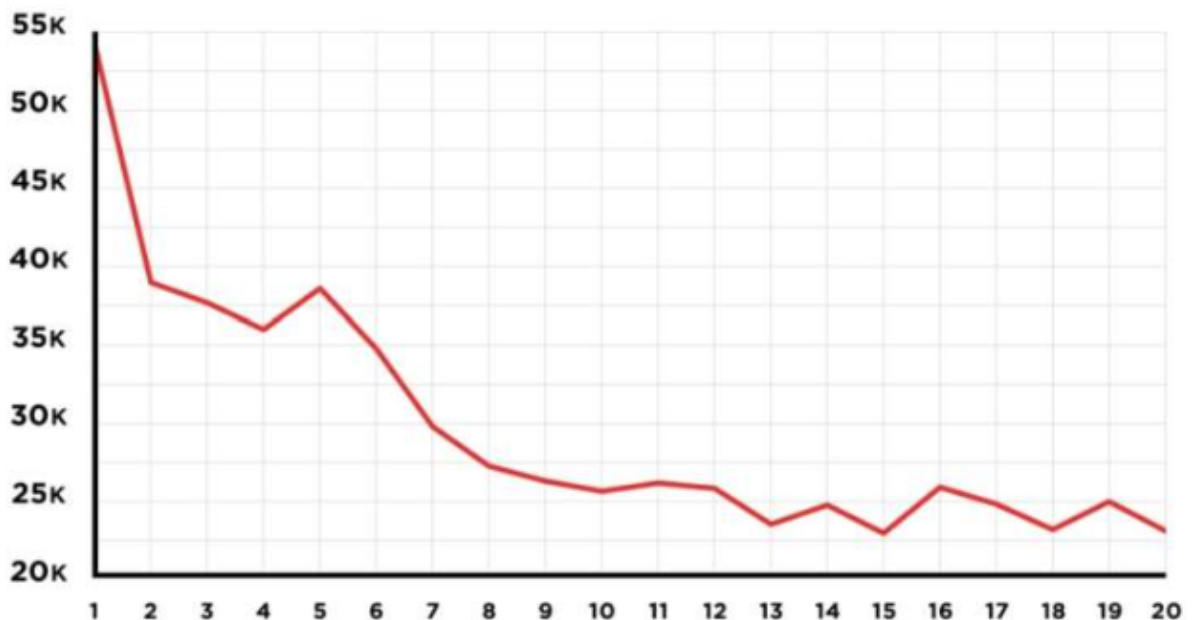


Рисунок 1.2 – Залежність позиції відео від кількості вподобань

Подібно до попереднього випадку, в області YouTube обман не є вдалим способом. Платформа вдосконалилася в розпізнаванні шахрайства і тепер стала менш вразливою до таких практик.

До 2014 року оцінка відео на YouTube головним чином здійснювалася за кількістю переглядів. Проте після цього відбулися зміни в алгоритмах платформи. Нині YouTube акцентує увагу на різних факторах, зокрема на тривалості відео.

Тривалість відео стала одним з ключових чинників, які впливають на рейтинг в пошукових результатах [11]. Чим довше відео, тим вища ймовірність його вдалого позиціонування. YouTube прагне, щоб користувачі проводили на їхньому сайті якомога більше часу. Це стимулює розміщення тривалих відео перед коротшими, сприяючи подовженню часу перебування користувачів на платформі, поліпшенню показників їхньої активності та збільшенню доходів від переглядів реклами.

В середньому, відео на YouTube тривають близько 15 хвилин, але відео, що триває більше 5 хвилин, також може мати високу оцінку.

Термін «утримання глядачів» відноситься до часу, який глядачі витрачають на перегляд відео. Тобто якщо 10-хвилинне відео переглядають до кінця, це вважається нормальним, і YouTube високо розташовує його в пошукових результатах. У випадку, коли глядачі переглядають відео лише кілька секунд або припиняють перегляд без його відновлення, це може вказувати на те, що відео не цікаве або має низьку якість, і YouTube знижує його рейтинг. Важливіше для платформи є утримання глядачів, ніж кількість лайків чи дизлайків.

Ще одним важливим фактором є можливість «Поділитися», яка дозволяє користувачам розміщати відео у соціальних мережах та на інших сайтах. Хоча ця функція впливає на рейтинг, її вплив є відносно невеликим. Залежність позиції відео від кількості людей, що поділилися проілюстрована на рисунку 1.3.

Як і в розглянутих вище випадках, YouTube враховує ряд характеристик (перегляди відео, вподобайки, збереження контенту тощо). Тут також передбачено виявлення шахрайства з розповсюдженнями, YouTube виявляє такі дії і може оштрафувати відео. Платформа вдосконалилася в розпізнаванні шахрайства та обману, що важливо для справедливого ранжування відео.

Отже, можливість «Поділитися» важлива для розповсюдження відео в соціальних мережах та на інших платформах, але її вплив на рейтинг є відносно невеликим.

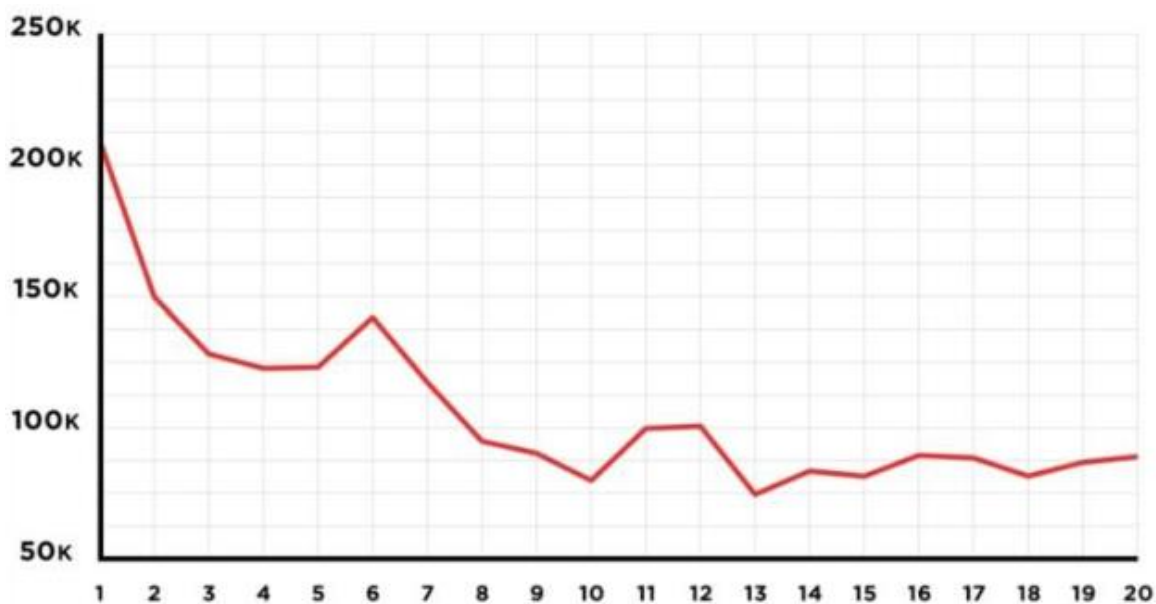


Рисунок 1.3 – Залежність позиції відео від кількості поширень

Інший аспект, який варто враховувати при ранжуванні відео на YouTube, - це підписка на канал. На відміну від колишньої суті кількості передплатників на каналах, ця цифра тепер не має істотного впливу на позицію відео в загальному рейтингу. Проте, підписки, які користувачі здійснюють одразу після перегляду відео, можуть мати суттєвий вплив.

Практика показує, що вміст, після перегляду якого користувачі підписуються на канал, часто займає вищі позиції на YouTube. Важливою також є якість відео, оскільки платформа акцентує увагу на рекламі HD-відео, яке завжди розташоване вище в рейтингу. Висока якість зображення сприяє приємнішому перегляду, і, відповідно, якість впливає на формування рейтингу.

Зокрема, для тих, хто прагне просувати відео на YouTube, рекомендується записувати їх у мінімальній роздільній здатності 1280x720, і навіть краще – у форматі 1920x1080 або 4K. Чим вища якість зображення, тим більше ймовірність, що відео потрапить у топ результатів відеохостингу.

Ранжування відео на стрімінгових платформах, зокрема на YouTube, є складним і багатоаспектним процесом. Важливо зауважити, що конкретні фактори та їх вага у визначенні рейтингу можуть різнитися в залежності від алгоритмів ранжування та стратегії самої стрімінгової платформи. Кожна

платформа може мати власні критерії ранжування, які враховують різноманітні фактори для формування остаточного рейтингу відео. Рекомендації для оптимізації включають створення вмісту, який стимулює підписки, довгий час перегляду та взаємодію глядачів, а також дотримання високих стандартів якості відео.

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

Для ранжування відео на стрімінговій платформі використовуються різні математичні методи і алгоритми. Основна мета полягає в тому, щоб забезпечити користувачам найбільш релевантний та персоналізований контент, враховуючи їхні вподобання та інтереси.

Рекомендаційні системи використовують методи машинного навчання, такі як колаборативний та контентний фільтри, для ранжування відео [13]. Колаборативний фільтр враховує взаємодію користувачів з відео (наприклад, подібність переглядів або оцінок) для рекомендацій. Контентний фільтр аналізує характеристики відео (такі як жанр, ключові слова, теги) для знаходження подібного контенту.

Контентна фільтрація в рекомендаційних системах базується на аналізі поведінки користувачів у системі та враховує збережений вміст [14]. Наприклад, цей метод використовує дані про перегляди статей і блогів, оцінюючи теми, які читає користувач, а також характеристики цих блогів та відео.

З іншого боку, колаборативна фільтрація генерує рекомендації на підставі поведінки попередніх користувачів. Авторизований доступ до ресурсу дозволяє системі збирати інформацію про поведінку користувачів протягом тривалого періоду. Більшість служб пропонують оцінки від користувачів, які зберігаються в базі даних. Такий підхід дозволяє системі подавати рекомендації з урахуванням користувацьких вподобань індивідуально [15]. Загалом, рекомендації ґрунтуються на колективній співпраці користувачів та відборі через фільтрацію

тих, які демонструють схожі уподобання або моделі поведінки, аналогічні користувачу, що шукає конкретний контент. Спільна фільтрація широко використовується в торгових сервісах і соціальних мережах.

Гібридні методи фільтрації комбінують підходи спільної роботи та фільтрації контенту для надання більш точних та індивідуалізованих рекомендацій. Вони можуть використовуватися для ранжування відео на YouTube, аналізуючи готові набори даних та оновлюючи їх у відповідь на постійно змінюючуся колекцію документів на цій платформі. Порівняльні характеристики методів фільтрації даних, що розглядаються в рекомендаційних системах, наведені в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика рекомендаційних методів

	Колаборативна фільтрація	Контентна фільтрація	Гібридна фільтрація
Переваги	Ефективна для рекомендацій вже популярних або схожих об'єктів, не вимагає аналізу контенту	Діє ефективно в умовах холодного старту, здатна рекомендувати унікальний контент	Покращена точність, здатність працювати в різноманітних умовах, компенсація недоліків інших методів
Недоліки	Проблеми з холодним стартом, коли об'єкти ще не отримали достатньо оцінок	Залежність від якості та правильності анотацій, обмежена у випадку нових або малопопулярних об'єктів	Складніше впровадження та налаштування, може вимагати більше обчислювальних ресурсів
Застосування	Блоги Інформаційні портали	Інтернет-магазини Інтернет-аукціони Блоги	Великі інтернет-магазини Складні соцмережі
Приклади	Spotify, Reddit Digg	Medium, Music Genome Project	Amazon, Netflix

Колаборативне ранжування. Цей метод використовує інформацію про взаємодію користувачів з платформою для ранжування відео. Можуть бути використані алгоритми ранжування, такі як RankNet або RankBoost, для

оптимізації порядку відображення відео на основі історичних даних про перегляди, оцінки або реакції користувачів.

Алгоритм RankNet є одним з методів машинного навчання, який використовується для ранжування об'єктів на основі їх порівнянь. Він був запропонований Крісом Буррісом та Томасом Лінденбергом в 2005 році і відноситься до сімейства алгоритмів, відомих як «Ranking SVM» (Support Vector Machine) [16]. Основна ідея RankNet полягає в тому, що він використовує нейронну мережу для навчання моделі, яка може ранжувати об'єкти на основі їх порівнянь. Порівняння відбуваються між парами об'єктів, для яких відомо, який з них більш пріоритетний або важливіший.

Основні кроки алгоритму RankNet:

1. Підготовка даних: Необхідно мати набір даних, що складається з пар об'єктів і їх відповідних порівнянь. Кожній парі присвоюється значення, що вказує, який об'єкт має вищий пріоритет.

2. Визначення функції втрат: RankNet використовує функцію втрат, яка вимірює різницю між прогнозованими і фактичними значеннями пріоритету. Зазвичай використовується функція "cross-entropy loss" або "mean squared error".

3. Навчання моделі: Нейронна мережа RankNet навчається шляхом мінімізації функції втрат за допомогою градієнтного спуску. Ваги нейронної мережі оновлюються під час тренування з метою покращення точності ранжування.

4. Ранжування: Після навчання моделі можна використовувати її для ранжування нових об'єктів. Модель приймає вхідні дані і видає оцінки пріоритету об'єктів.

Основна ідея RankBoost полягає в тому, що він використовує ансамбль слабких ранжуючих моделей (таких як дерева рішень або регресійні моделі) і комбінує їх, щоб отримати більш сильну ранжуючу модель [17]. Кожна слабка модель призначає вагу кожному об'єкту, але ваги оновлюються на кожній ітерації залежно від їх ефективності в порівнянні з фактичними пріоритетами.

Основні кроки алгоритму RankBoost:

1. Підготовка даних: Аналогічно до RankNet, необхідно мати набір даних, що складається з пар об'єктів і їх відповідних порівнянь. Кожній парі присвоюється значення, що вказує, який об'єкт має вищий пріоритет.
2. Ініціалізація ваг: Кожному об'єкту початково присвоюються однакові ваги.
3. Навчання слабких моделей: На кожній ітерації RankBoost навчає слабку модель, яка призначає вагу кожному об'єкту. Модель побудована таким чином, щоб максимізувати кореляцію між порівняннями об'єктів і їх фактичними пріоритетами. Наприклад, можуть використовуватись дерева рішень, які виконують розділення на основі порівнянь.
4. Оновлення ваг: Після кожної ітерації ваги оновлюються відповідно до ефективності слабкої моделі у порівнянні з фактичними пріоритетами. Об'єкти, для яких модель допускає помилки, отримують більшу вагу.
5. Комбінування моделей: На підсумковому етапі RankBoost комбінує всі слабкі моделі в одну сильну ранжуючу модель, використовуючи ваги, які були присвоєні на кожній ітерації.

Алгоритм RankBoost дозволяє покращити ранжування об'єктів шляхом підбору оптимальних ваг і комбінації слабких моделей. Цей метод може бути застосований до ранжування відео на стрімінгових платформах для покращення персоналізації та відповідності вподобанням користувачів.

Машинне навчання: Методи машинного навчання, такі як нейронні мережі, можуть бути використані для ранжування відео [18]. Моделі можуть бути навчені на великому обсязі даних, щоб передбачити інтерес користувача до певного відео та встановити його рейтинг.

Нейронні мережі є потужним і ефективним інструментом для ранжування відео на стрімінговій платформі. Використання нейронних мереж дозволяє враховувати складні залежності між різними факторами і забезпечує гнучкість і адаптивність моделі до змінних умов та контексту [19].

Використання звичайних feedforward нейронних мереж (таких як прямі мережі або мережі з одношаровим персептроном) для ранжування відео

характеризується тим, що модель приймає вхідні дані, такі як характеристики відео та контексту, і видає оцінку ранжування. Цю модель можна навчити за допомогою навчання з вчителем, використовуючи позначені дані ранжування.

Рекурентні нейронні мережі (RNN), такі як Long Short-Term Memory (LSTM) або Gated Recurrent Unit (GRU), використовуються для врахування послідовного контексту інформації у ранжуванні відео. Наприклад, RNN може використовуватись для аналізу коментарів або послідовностей дій користувача, що допомагає врахувати інформацію про сприйняття та взаємодію користувача з відео.

Згорткові нейронні мережі (CNN) використовуються для аналізу візуальних аспектів відео. CNN можуть виявляти образи, об'єкти або особливості відеокадрів та забезпечувати оцінку ранжування на основі цих візуальних аспектів. Використання претренованих моделей CNN, таких як VGG або ResNet, може поліпшити якість ранжування.

Використання attention-механізмів у нейронних мережах для визначення ваги різних характеристик відео або контексту. Attention дозволяє моделі акцентувати увагу на важливих факторах та ігнорувати менш важливі, що сприяє кращому ранжуванню.

Важливо відзначити, що успішне використання нейронних мереж для ранжування відео вимагає великого обсягу позначених даних для навчання та досить обчислювальних ресурсів для тренування та застосування моделей. Крім того, постійна оновлення та переосмислення моделей можуть бути необхідними для забезпечення актуальності ранжування.

Метод аналізу ієрархій (Analytic Hierarchy Process, АНР) може бути використаний для ранжування відео на стрімінговій платформі, особливо якщо враховується предметна оцінка відповідності відео певним критеріям або факторам [20]. АНР є методом прийняття рішень, який дозволяє порівнювати та ранжувати різні альтернативи на основі їх вагомості у відношенні до певних критеріїв.

В контексті ранжування відео, метод АНР може бути застосований наступним чином:

1. Визначення критеріїв ранжування. Спочатку визначаються критерії, які впливають на якість або релевантність відео. Наприклад, це можуть бути критерії, такі як популярність, якість відео, релевантність до користувача тощо.
2. Побудова ієрархії. Створюється ієрархія, де на верхньому рівні знаходиться загальна ціль (наприклад, ранжування відео), наступний рівень складається з критеріїв ранжування, а нижчі рівні представляють альтернативи (конкретні відео).
3. Оцінювання парних порівнянь. За допомогою матриць парних порівнянь користувачі або експерти оцінюють важливість кожного критерію відносно інших критеріїв. Наприклад, користувач може порівняти популярність і якість відео та визначити, який критерій є важливішим.
4. Розрахунок ваг критеріїв. На основі оцінок парних порівнянь виконується математичний аналіз для визначення ваг критеріїв у відношенні один до одного.
5. Ранжування відео. Коли ваги критеріїв визначені, можна застосувати їх до конкретних відео, щоб визначити їх ранг або релевантність. Це можна зробити, порівнюючи кожне відео з кожним критерієм та обчислюючи сумарну оцінку.

Метод АНР надає систематичний підхід до ранжування відео, забезпечуючи більш обґрунтоване прийняття рішень [21]. Однак, варто враховувати, що використання методу АНР може потребувати деякої експертизи та інтерактивного взаємодії з користувачами для збору оцінок та побудови ієрархії.

Розглянуті методи можуть застосовуватись окремо або комбінуватись для досягнення найкращих результатів у ранжуванні відео на стрімінговій платформі. Важливо також враховувати контекст і особливості конкретної платформи та потреби користувачів для досягнення оптимального результату.

Дослідивши методи, що можуть бути використані для ранжування відео, запропоновано використати метод на основі RankBoost, зважаючи на його переваги. Цей підхід буде детальніше описано у другому розділі магістерської роботи.

1.4 Аналіз відомих програмних рішень для ранжування відео на стрімінговій платформі

На сьогоднішній день існує кілька відомих програмних рішень для ранжування відео на стрімінгових платформах. Ці програмні рішення використовуються відомими стрімінговими платформами та базуються на складних алгоритмах, які аналізують різні фактори для ранжування відео та надання користувачам персоналізованих рекомендацій. Кожне рішення має свої унікальні особливості та алгоритми, але загальна мета полягає в тому, щоб забезпечити кращий досвід перегляду для користувачів та підвищити популярність платформи.

YouTube Ranking Algorithm: YouTube, одна з найбільш відомих стрімінгових платформ, використовує свій власний алгоритм ранжування для визначення порядку відео в результатах пошуку та рекомендаціях [22]. Цей алгоритм враховує такі фактори, як кількість переглядів, залучення аудиторії, тривалість перегляду та інші.

Алгоритм ранжування YouTube складний і враховує багато факторів для визначення порядку відео в результатах пошуку та рекомендаціях. Хоча конкретні деталі алгоритму не розкриваються публічно, основні принципи роботи можуть включати наступні етапи:

1. Збір даних. YouTube збирає велику кількість даних про користувачів та їх взаємодію з відео. Це включає інформацію про перегляди, лайки, дизлайки, коментарі, підписки та інші дії користувачів.

2. Врахування релевантності. Алгоритм враховує релевантність відео до запиту користувача або контекстуального середовища. Він аналізує

заголовки, описи, теги та інші метадані відео для визначення його відповідності запиту користувача.

3. Кількість переглядів. Відео з великою кількістю переглядів можуть мати перевагу у ранжуванні. Високі показники переглядів вказують на популярність та цікавість відео.

4. Взаємодія глядачів. Алгоритм враховує рівень взаємодії глядачів з відео, такий як тривалість перегляду, коментарі, лайки та дизлайки. Відео, яке залучає активну аудиторію та викликає позитивні реакції, може мати більш високий рейтинг.

5. Співвідношення взаємодії. Алгоритм також враховує співвідношення між позитивною та негативною взаємодією з відео. Наприклад, відео з багато лайками та мало дизлайків може мати перевагу над відео зі зворотним співвідношенням.

6. Час перегляду. Тривалість перегляду відео є важливим фактором. Відео, яке зберігає глядачів на платформі на тривалий час, може бути високо оціненим.

7. Персоналізація. YouTube також враховує історію перегляду користувача та його вподобання для надання персоналізованих рекомендацій. Алгоритм спробує знайти відео, які відповідають індивідуальним інтересам та передбаченням користувача.

У 2016 році команда YouTube оприлюднила докладну інформацію про використання глибоких нейронних мереж і методів машинного навчання в їхньому алгоритмі рекомендацій [23]. Алгоритм ґрунтувався на майже мільярді параметрів, які розподілені на дві категорії: ті, які стосуються аудиторії (допомагають створити інтерес до пошуку цікавих тем для відео), і ті, які стосуються відео (рейтинги, які допомагають визначити тему, якість та привабливість контенту).

Параметри аудиторії варіювалися від простих (наприклад, вхід користувача до системи) до складних (наприклад, нещодавні пошукові запити користувача) і можуть мати мільйони можливих значень. Вони також поділялися

на одновалентні (наприклад, ідентифікатор відео, яке зацікавило глядача) та багатовалентні (наприклад, реакція користувача на групу відео). При рекомендаціях враховувались не лише параметри самого відео, але й параметри користувача, такі як його інтереси та пошукові запити.

З використанням цих параметрів кожне відео подавалося глядачеві, маючи за мету залучити його до якнайбільшого часу перебування на YouTube. Значуще відзначити, що YouTube регулярно вдосконалює свій алгоритм ранжування, щоб покращити користувацький досвід і відповідати змінюються потреби глядачів. Таким чином, конкретні деталі алгоритму можуть еволюціонувати з часом.



Рисунок 1.4 – Система ранжування рекомендацій YouTube

Отже, алгоритм рекомендацій YouTube є складним системним рішенням, що базується на глибоких нейронних мережах та методах машинного навчання. Параметри аудиторії та відео ретельно враховуються, враховуючи інтереси,

пошукові запити і реакції глядачів. Алгоритм регулярно оновлюється з метою покращення користувацького досвіду і адаптації до змінюючихся потреб аудиторії.

Прозорість у використанні глибоких нейронних мереж та алгоритмів машинного навчання свідчить про стратегічний підхід YouTube до забезпечення якості та різноманітності контенту на платформі

Netflix Recommendation System: Netflix використовує свою власну систему рекомендацій для ранжування відео на своїй платформі [24]. Ця система враховує історію перегляду користувача, оцінки, демографічні дані та інші фактори для надання персоналізованих рекомендацій (рисунок 1.5).

Netflix використовує розширену систему рекомендацій, яка базується на методах машинного навчання та аналізу даних для надання персоналізованих рекомендацій користувачам. Основні принципи роботи системи рекомендацій Netflix включають:

- Колаборативний фільтринг. Система аналізує взаємодію користувачів з відео, таку як оцінки, перегляди та додавання до списку перегляду, для знаходження схожих користувачів. Якщо користувач подібний до іншого користувача та має подібні вподобання, то система може рекомендувати йому відео, які сподобаються іншому користувачеві.

- Змішана модель рекомендацій. Netflix також використовує комбінацію колаборативного фільтрування та контентного аналізу. Контентний аналіз включає аналіз характеристик відео, таких як жанр, актори, режисери, сюжетні лінії та інші фактори, щоб зрозуміти вподобання користувачів. Застосовуючи ці два підходи, система може надати більш точні та релевантні рекомендації.

- Аналіз здатності прогнозування. Netflix використовує алгоритми прогнозування, які аналізують історію перегляду та взаємодію користувача з відео, щоб передбачити, які фільми або серіали можуть сподобатися користувачу в майбутньому. Це допомагає забезпечити більш точні та персоналізовані рекомендації.

- Експерименти та зворотний зв'язок. Netflix постійно проводить експерименти з різними алгоритмами та підходами до рекомендацій, щоб вдосконалювати систему. Крім того, вони залучають зворотний зв'язок від користувачів, збираючи дані про їх реакції та перегляди, щоб виробляти кращі рекомендації в майбутньому.



Рисунок 1.5 – Рекомендації фільмів за допомогою Netflix Recommendation System

Загалом, Netflix Recommendation System поєднує різні методи та алгоритми, щоб надати користувачам індивідуальні, персоналізовані рекомендації, які відповідають їхнім уподобанням та забезпечують зручний досвід перегляду.

TikTok Recommendation Algorithm: TikTok використовує складну систему рекомендацій, яка базується на аналізі поведінки користувача, взаємодії з відео, місцезнаходженні та інших факторах (рисунок 1.6). Цей алгоритм стежить за тим, які відео користувачі дивляться, лайкають, коментують, та надає рекомендації на основі цих взаємодій [25].

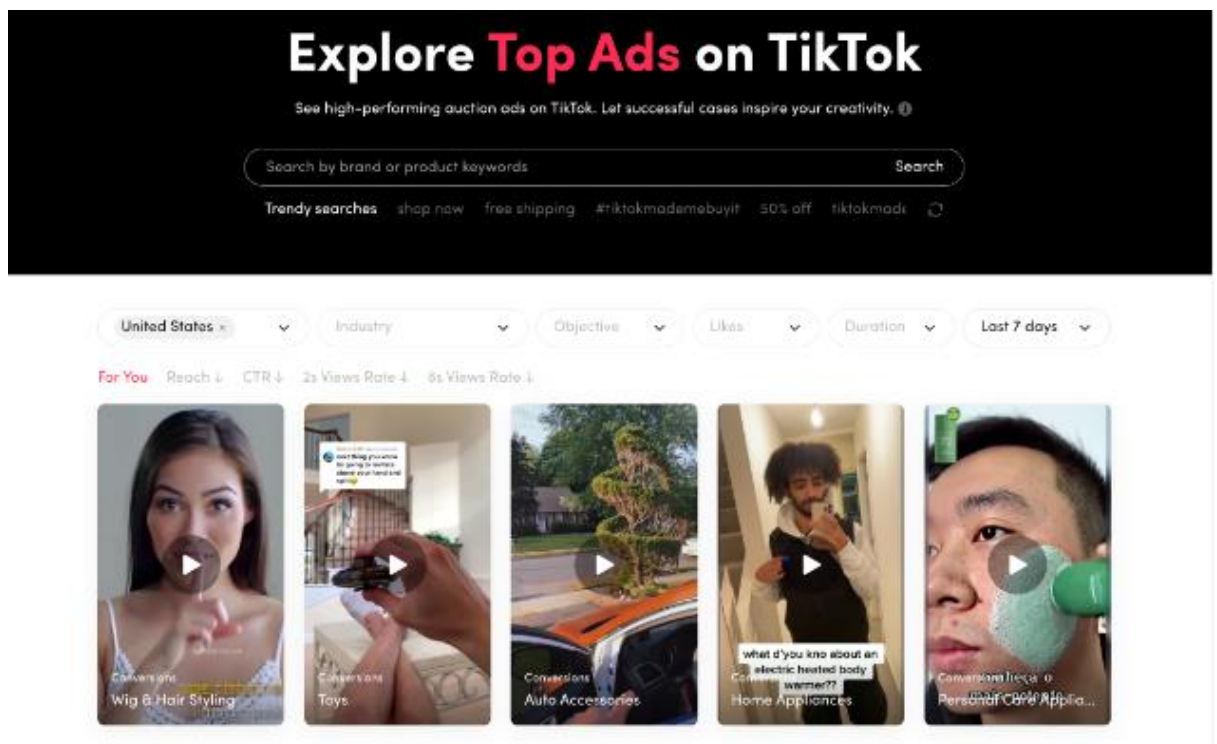


Рисунок 1.6 – Приклад рекомендацій в TikTok

Основними принципами роботи алгоритму є аналіз поведінки користувача, профіль користувача, подібність контенту, популярність та тренди, ретенція глядачів.

TikTok збирає дані про взаємодію користувача з відео, такі як перегляди, лайки, коментарі, репости та тривалість перегляду. Ці дані допомагають алгоритму зрозуміти вподобання та інтереси користувача. Алгоритм також враховує інформацію з профілю користувача, таку як демографічні дані, місцезнаходження та мова. Це допомагає налаштувати рекомендації під конкретного користувача. Алгоритм аналізує відео, які користувачі спостерігають, лайкають або коментують, щоб знайти подібний контент. Це означає, що якщо користувач дивиться багато відео на певну тему, то алгоритм намагається знайти йому подібні відео для рекомендацій.

TikTok також враховує популярність відео та тренди. Відео, які стають вірусними або отримують велику кількість лайків та переглядів, можуть бути включені до стрічки рекомендацій для більш широкої аудиторії. Алгоритм враховує тривалість перегляду відео та реакції глядачів на них. Відео, які

залучають увагу та тримають глядачів на платформі тривалий час, можуть мати перевагу у рекомендаціях.

Amazon Video Personalization: Amazon Video використовує методи машинного навчання та аналізу даних для ранжування відео на своїй платформі (рисунок 1.7). Вони враховують такі фактори, як історія перегляду, оцінки, взаємодія з іншими відео та поведінка користувача для надання персоналізованих рекомендацій [26].

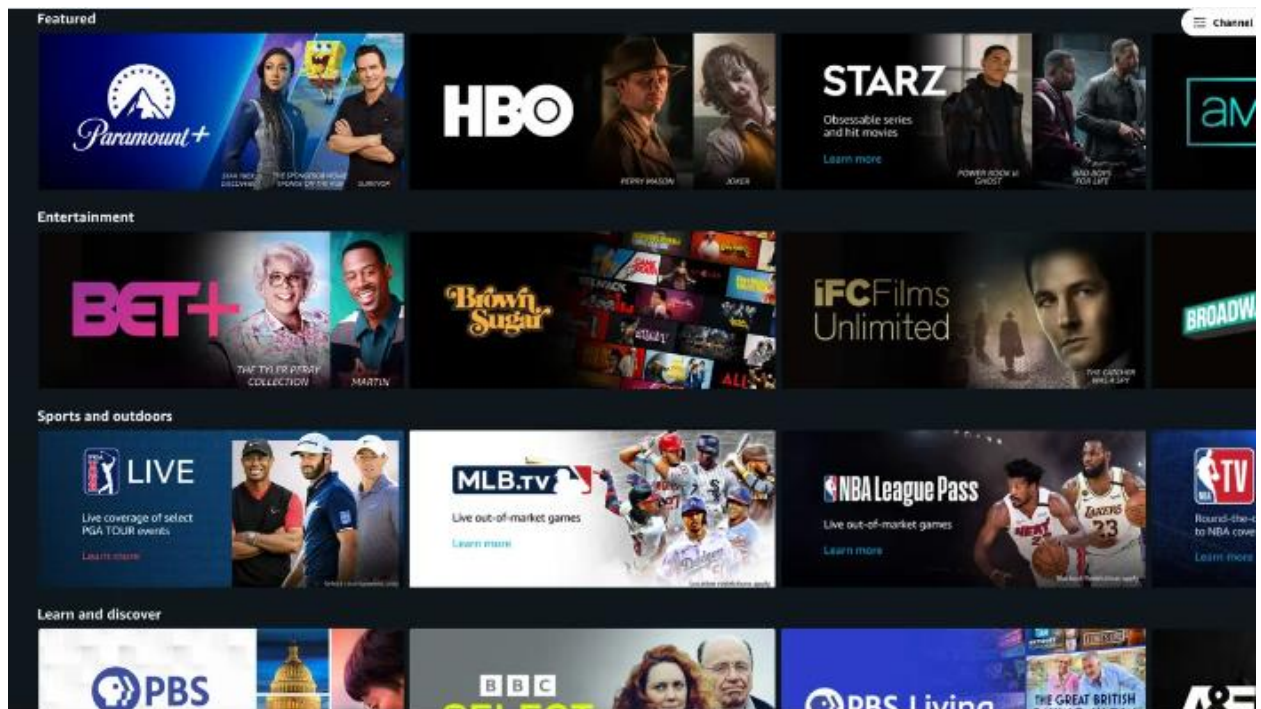


Рисунок 1.7 – Рекомендації фільмів за допомогою Amazon Video

Amazon Video Personalization – це система рекомендацій, яка використовується на платформі Amazon Video для надання персоналізованих рекомендацій користувачам.

Amazon Video збирає різноманітні дані про взаємодію користувачів з платформою, такі як перегляди, оцінки, відгуки, додавання до списку бажань та інші дії. Ці дані використовуються для розуміння вподобань та поведінки користувачів.

Система використовує колаборативний фільтринг для знаходження схожих користувачів із подібними вподобаннями. Якщо два користувачі

спостерігають аналогічні відео або мають спільні інтереси, то система може рекомендувати одному користувачеві відео, які сподобаються іншому.

Amazon Video також аналізує характеристики відео, такі як жанр, актори, режисери, сюжетні лінії та інші фактори, щоб розуміти контент та враховувати вподобання користувачів. Наприклад, якщо користувач подивився багато фільмів одного жанру, система може рекомендувати йому подібні фільми.

Amazon Video використовує дані про поведінку користувача, такі як тривалість перегляду, перемотування, паузи та інші фактори, щоб враховувати зацікавленість користувача в конкретних відео та рекомендувати подібний контент. Amazon Video постійно проводить експерименти з різними моделями та алгоритмами рекомендацій, щоб вдосконалювати систему. Вони також використовують методи машинного навчання для навчання моделей на основі зібраних даних та вдосконалення рекомендацій з часом.

Загалом, Amazon Video Personalization використовує комбінацію колаборативного фільтрування, контентного аналізу та аналізу поведінки користувачів для надання персоналізованих рекомендацій та забезпечення зручного та захоплюючого досвіду перегляду відео.

Отже, переглянувши існуючі рішення щодо ранжування відео на стрімінговій платформі, виникає необхідність у створенні нового рішення у вигляді інформаційної системи ранжування відео на стрімінговій платформі, використовуючи наведені в постановці задачі параметри ранжування, що забезпечить підвищення точності ранжування відео.

1.5 Формулювання вимог та постановка задачі

Магістерська кваліфікаційна робота присвячена реалізації інформаційної технології ранжування відео на стрімінговій платформі. Тобто, необхідно програмно реалізувати алгоритм ранжування відео для подальшого використання на стрімінговій платформі. Інформаційна технологія призначена для користувачів, які потребують системи для ранжування відео. Реалізація

алгоритму ранжування відео за використанням низки факторів, таких як тривалість відеоролика, тематика, кількість переглядів, кількість переглядів за першу добу, кількість переглядів на каналі автора, середня кількість на каналі автора, відсоток аудиторії, що подивилася більше половини відеоролика, відсоток аудиторії, яка подивилася 90% відеоролика, є спрямованою на задоволення потреб користувачів, які шукають ефективний спосіб оцінки та вибору відео на платформі.

Особлива увага приділяється зручності та доступності інформаційної технології для кінцевого користувача, підкреслюючи, що вона повинна бути зрозумілою та не вимагати спеціальних навичок у галузі алгоритмів ранжування. Важливим є також забезпечення максимальної точності у побудові рейтингу відео, щоб користувач міг ефективно обирати контент з урахуванням його індивідуальних уподобань.

Отже, задача цієї роботи полягає у розробці інформаційної технології ранжування відео на стрімінговій платформі.

Вимоги до клієнту:

- головна сторінка з меню;
- перегляд рейтингу у вигляді таблиці;
- можливість вибору категорії;
- можливість додавання нових відео;
- завершення сесії (вихід).

Вимоги до менеджера:

- можливість редагування інформації;
- можливість додавання нової інформації (відео);
- можливість видалення інформації;

Вимоги до серверу:

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

Вхідними даними роботи є відео, та дані про нього: довжина, кількість переглядів та інші дані, що цікавить користувача.

Вихідними даними є список відео у формі рейтингу.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм ранжування відео на стрімінговій платформі, що буде працювати з допустимою точністю.

1.6 Висновок до розділу 1

У даному розділі було досліджено принципи ранжування відео на стрімінгових платформах. Також визначено фактори, які впливають на ранжування відео, зокрема досліджено вплив факторів на ранжування відео на прикладі платформи YouTube. Було проведено аналіз сучасних програм-аналогів, які використовують алгоритми ранжування відео. Було наведено короткий опис основних функцій, які виконують дані програми. Також було досліджено методи, що можуть бути використані для ранжування відео, для чого запропоновано використати метод на основі алгоритму ранжування RankBoost, що є оптимальним, оскільки дозволяє покращити ранжування об'єктів шляхом підбору оптимальних ваг і комбінації слабких моделей. Сформульовано вимоги та здійснено постановку задачі для подальшої розробки.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ВІДЕО НА СТІМІНГОВІЙ ПЛАТФОРМІ

2.1 Обґрунтування вибору методу ранжування відео на стрімінговій платформі

Вибір методу RankBoost для ранжування відео на стрімінговій платформі може бути обґрунтований наступними факторами, які роблять цей метод привабливим для задач ранжування контенту.

Адаптація до складних залежностей. RankBoost – це алгоритм адаптивного ранжування, який може адаптуватися до складних та нелінійних залежностей між ознаками відео та взаємодією користувачів. Це означає, що він може бути ефективним у врахуванні різних чинників, таких як рейтинг відео, кількість переглядів, інтереси користувачів та інші [27].

Навчання на даних. RankBoost базується на машинному навчанні і може вчитися на навчальному наборі даних. Це дозволяє адаптувати ранжування до конкретних умов та вимог платформи, а також до зміни у вподобаннях та поведінці користувачів.

Важливість контексту. RankBoost може бути розроблений для врахування контекстуальних факторів, таких як місцезнаходження користувача, тип пристрою, з якого відбувається перегляд, та час доби. Це допомагає покращити персоналізацію та точність ранжування.

Підтримка різних ознак. RankBoost може легко враховувати різні ознаки відео та користувачів. Це важливо, оскільки для ранжування відео можуть бути важливі різні фактори, такі як якість відео, рейтинг, категорія, демографічні характеристики користувача і багато інших.

Покращена точність ранжування. Метод RankBoost дозволяє покращити точність ранжування за рахунок комбінації багатьох базових класифікаторів. Це може допомогти платформі надавати користувачам більш релевантний контент і підвищити задоволення від використання платформи.

Фідбек і оновлення. RankBoost може бути легко оновлюваним і адаптованим до нових даних та потреб користувачів. Це робить його ефективним для стрімінгових платформ, які постійно змінюються та розвиваються.

Загалом, вибір методу RankBoost для ранжування відео на стрімінговій платформі обґрунтовується його здатністю адаптуватися до складних залежностей та вимог, покращувати точність ранжування та забезпечувати ефективну персоналізацію для користувачів.

Алгоритм ранжування RankBoost може бути побудована наступним чином:

1. Складання навчального набору даних. Спершу потрібно зібрати навчальний набір даних, що складається з прикладів ранжування відео. Кожен приклад повинен включати в себе інформацію про відео (наприклад, рейтинг, кількість переглядів, час викладення), а також змінну, яку потрібно передбачити (наприклад, чи відео буде відображатися користувачеві).

2. Вибір ознак (Features). Визначення ознак відео та користувачів, які будуть використовуватися для ранжування. Це можуть бути, наприклад, рейтинг відео, кількість попередніх переглядів, вподобання користувача тощо.

3. Побудова базових класифікаторів. RankBoost базується на великій кількості багатьох базових класифікаторів. Побудувати базові класифікатори, які оцінюють, наскільки важливою є кожна з визначених ознак для ранжування.

4. Навчання RankBoost. Використовуючи навчальний набір даних та базові класифікатори, провести навчання алгоритму RankBoost. Цей алгоритм буде навчати ваги для базових класифікаторів, з якими вони використовуються для ранжування відео.

5. Ранжування відео. Після навчання алгоритму RankBoost можна використовувати його для ранжування відео. Для кожного відео слід визначити його оцінку на основі ваг, які були навчені під час процесу навчання.

6. Вивід ранжування. Слід вивести відео у відповідному порядку від найвищої до найнижчої оцінки. Користувачам будуть показані відео в цьому порядку на стрімінговій платформі.

7. Оновлення моделі. Необхідно періодично оновлювати модель, використовуючи нові дані та фідбек від користувачів, щоб покращити якість ранжування.

2.2 Розробка математичної моделі ранжування відео на стрімінговій платформі

Математична модель ранжування відео на стрімінговій платформі - це алгоритм або система правил, які визначають порядок, в якому відео контент буде представлений користувачам на платформі. Основна мета такої моделі полягає у забезпеченні користувачам персоналізованого та відповідного контенту, що відповідає їхнім інтересам та вимогам.

Ранжування за допомогою лінійної регресії відбувається завдяки побудові лінійної функції, яка найкращим чином апроксимує залежність між вхідними факторами і вихідним рейтингом [28]. Математично це виражається як:

$$R = \omega_0 + \omega_1 \cdot x_1 + \omega_2 \cdot x_2 + \dots + \omega_n \cdot x_n \quad (2.1)$$

де R – рейтинг відео, $\omega_0, \omega_1, \dots, \omega_n$ – вагові коефіцієнти, які система вивчає під час навчання, x_1, x_2, x_n – фактори, такі як тривалість відеоролика, тематика, кількість переглядів, кількість переглядів за першу добу, кількість переглядів на каналі автора, середня кількість на каналі автора, відсоток аудиторії, що подивилася більше половини відеоролика, відсоток аудиторії, яка подивилася 90% відеоролика

RankBoost - це один з багатьох алгоритмів ранжування, який може бути використаний для побудови ефективної системи рекомендацій на основі даних і взаємодії з користувачами.

У стандартній задачі ранжирування мета полягає в тому, щоб знайти функцію ранжирування для впорядкування заданого набору об'єктів. Такий об'єкт позначається як екземпляр x у домені (або просторі екземплярів) X . Як

форма зворотного зв'язку інформація про те, який екземпляр має бути ранжований вище (або нижче) один одного, надається для кожної пари екземплярів [29]. Цей зворотний зв'язок позначається як функція $\Phi: X \times X \rightarrow R$, де $\Phi(x_0, x_1) > 0$ означає, що x_1 має бути ранжованим вище x_0 і навпаки. Потім потрібно знайти функцію ранжирування $H: X \rightarrow R$, яка максимально узгоджується з Φ , стверджуючи, що x_1 є кращим, ніж x_0 , якщо $H(x_1) > H(x_0)$.

Спершу необхідно визначити функцію втрат. Функція втрат визначає різницю між поточним порядком та бажаним порядком. Однією зі звичайних функцій втрат для ранжування є функція Exponential Loss:

$$L(y, f(x)) = \sum_{i=1}^N \exp(-y_i \cdot f(x_i)) \quad (2.2)$$

де y_i – мітка порядку, $f(x_i)$ – поточний порядок, N – кількість об'єктів.

Алгоритм RankBoost пропонує вивчати функцію ранжирування H шляхом комбінування заданої колекції функцій ранжирування.

RankBoost працює ітераційним способом. У кожному раунді викликається процедура WeakLearn для вибору найкращого слабкого рейтингу з великого набору кандидатів на слабкі рейтинги. Слабкий рейтинг має вигляд

$$h_t: X \rightarrow R, \quad (2.3)$$

$$h_t(x_1) > h_t(x_0) \quad (2.4)$$

Це означає, що екземпляр x_1 має вищий ранг, ніж x_0 у раунді t . Розподіл D_t за $X \times X$ (тобто пари відео) підтримується під час процесу навчання, щоб відобразити важливість правильного ранжування пари. Вага $D_t(x_0, x_1)$ буде зменшена, якщо h_t ранжирує x_0 і x_1 правильно (2.4), і збільшена в іншому випадку. Таким чином, D_t буде прагнути зосередитися на парах, які важко ранжувати.

$$D_{t+1}(x_0, x_1) = \frac{D_t(x_0, x_1) \exp(-\alpha_t(h_t(x_0) - h_t(x_1)))}{Z_t} \quad (2.5)$$

де Z_t – фактор нормалізації:

$$Z_t = \sum_{x_0, x_1} D_t(d_0, d_1) \exp(-\alpha_t(h_t(x_0) - h_t(x_1))) \quad (2.6)$$

Остаточний сильний рейтинг H — це зважена сума вибраних слабких рейтингів у кожному раунді.

$$H(x) = \sum_{t=1}^T \alpha_t h_t \quad (2.7)$$

де T – кількість базових моделей (раундів)

Математична модель ранжування враховуватиме вагу та важливість різних факторів.

Процес навчання RankBoost полягає у мінімізації функції втрат шляхом вибору кращих базових моделей та їх ваг. Ваги об'єктів оновлюються на кожному етапі, щоб краще врахувати помилки попередніх моделей. Після завершення навчання RankBoost можна використовувати для ранжування нових об'єктів.

2.3 Проектування структури та алгоритму функціонування інформаційної технології ранжування відео на стрімінговій платформі

Стрімінгові платформи стали невід'ємною частиною нашого цифрового життя, надаючи безліч можливостей для перегляду відеоконтенту. З урахуванням великої кількості доступного відеоматеріалу стає важливим створити систему ранжування, яка допоможе користувачам знаходити контент, що найкраще відповідає їхнім інтересам та вподобанням.

Спершу при проектуванні структури інформаційної технології для ранжування відео на стрімінговій платформі необхідно визначити, як

проходитиме збір даних. Треба визначити, звідки будуть отримувати дані про відео (тривалість, тематика, перегляди тощо). Це можна реалізувати з допомогою бази даних, що міститиме всю необхідну для цього інформацію. Далі потрібно здійснити обробку та очищення отриманих даних. Здійснити нормалізацію та стандартизацію даних для однорідності.

Потім необхідно розглянути, як кожен фактор впливає на ранжування. Визначити ваги для кожного фактора з урахуванням їхньої важливості. Іншими словами – розробити математичну модель ранжування відео на стрімінговій платформі. Реалізувати алгоритм RankBoost для визначення остаточного рейтингу відео. Врахувати можливість оптимізації та налаштування параметрів алгоритму. Також потрібно врахувати, як змінюються показники з часом. Розробити механізм для оновлення рейтингів в реальному часі.

Необхідно проводити системне тестування для перевірки ефективності та точності ранжування. Запровадити виправлення та вдосконалення на основі отриманих результатів.

Проектування структури інформаційної технології ранжування відео на стрімінговій платформі – це складний процес, який вимагає ретельного планування та розробки. Робота з методами ранжування вимагає великої обчислювальної потужності. Вхід інформаційної технології містить характеристики до відео. Після вибору необхідного жанру, за яким має бути проведено ранжування відбувається формування запитів до бази даних та обробка відповіді. Тоді проводиться ранжування, а на виході ми отримуємо результати ранжування, які задовільняють потреби користувача. Структурна схема інформаційної технології ранжування відео на стрімінговій платформі наведена на рис. 2.1.

Розробка архітектури системи повинна включати компоненти та їх взаємозв'язки. Гарним рішенням є розділення системи на модулі для забезпечення масштабованості та підтримки різних функцій.



Рисунок 2.1 – Структурна схема інформаційної технології

Модульний підхід до розробки ґрунтується на стратегії спрощення та пристосування програми за допомогою передачі модулів та рішень з інших компонентів. Усі функції програми розбиваються на окремі модулі, які взаємодіють через визначені інтерфейси. Кожен модуль є самостійною та логічно завершеною частиною програмного продукту.

Цей підхід сприяє написанню та розумінню логіки програми з більшою ясністю та легкістю. Він також зменшує витрати на обслуговування та модифікацію програми, а також підвищує її стійкість до несправностей. Основний принцип полягає в тому, щоб мати більше модулів із меншою кількістю з'єднань між ними.

В програмі будуть представлені модуль інтерфейсу, модуль ранжування відео, модуль роботи з базою даних. Структура інформаційної технології ранжування відео на стрімінговій платформі зображена на рисунку 2.2.

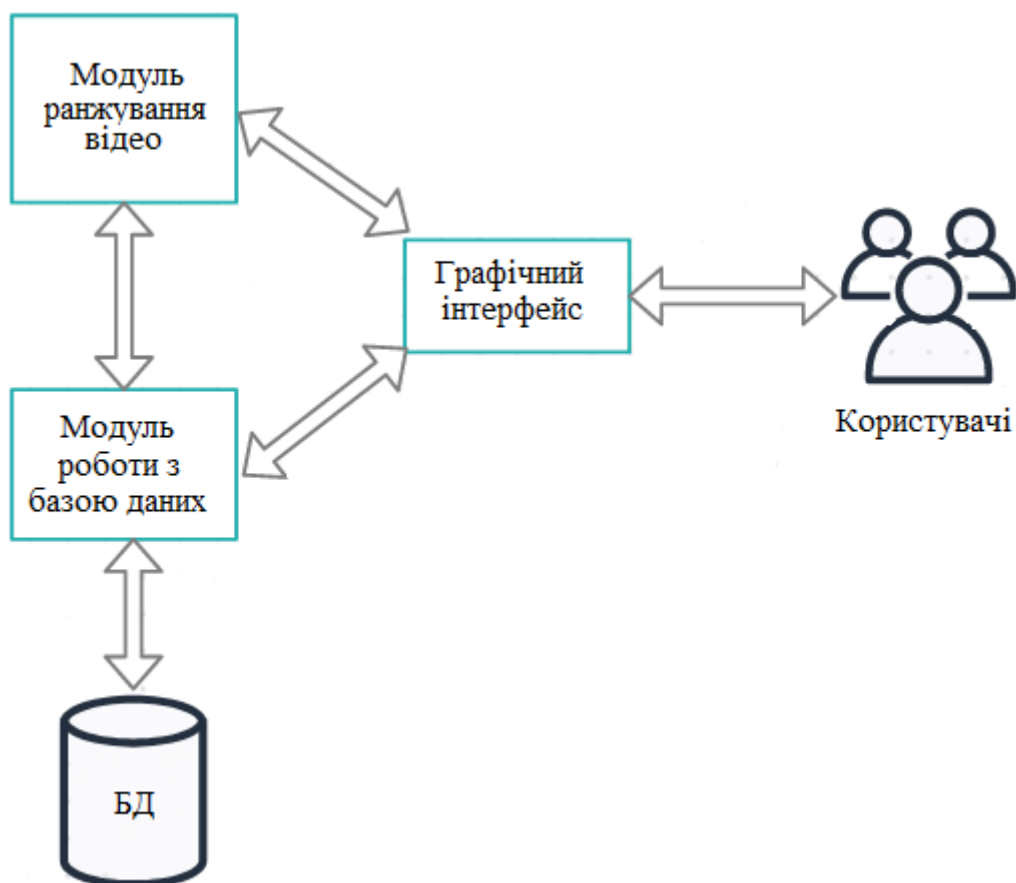


Рисунок 2.2 – Загальна структурна схема

Модуль інтерфейсу відповідає за взаємодію користувача з програмою. Включає в себе графічний інтерфейс для зручного введення параметрів, перегляду результатів та взаємодії з рекомендаціями. Основними його елементами є текстові поля, кнопки, візуалізація результатів.

Модуль ранжування виконує алгоритм ранжування відео за заданими факторами. Обчислює рейтинг для кожного відео на основі параметрів, таких як тривалість відео, тематика, кількість переглядів, кількість переглядів за першу добу тощо. Основними його елементами є блоки коду для обробки параметрів, алгоритм ранжування.

Модуль роботи з базою даних забезпечує зберігання та доступ до даних про відео, необхідних для ранжування. Включає в себе модуль для отримання даних з бази та їх оновлення. Цей модуль містить SQL-запити для створення та управління базою даних, методи для отримання та оновлення інформації.

База даних зберігає дані про відео, необхідні для ранжування, включаючи параметри, такі як тривалість, тематика, кількість переглядів, тощо. Вона повинна забезпечувати швидкий імпорт та автоматичний експорт усієї інформації, необхідної для ранжування.

Таким чином, структура інформаційної технології ранжування відео буде забезпечувати взаємодію користувача з системою через інтерфейс, аналізувати та оцінювати відео за допомогою модуля ранжування та зберігати необхідні дані в базі даних для подальшої роботи та оновлення системи.

Графічне представлення алгоритму функціонування модуля ранжування інформаційної технології ранжування відео на стрімінговій платформі представлено на рис. 3.3.

Алгоритм роботи інформаційної технології включає такі етапи:

1. Запуск інформаційної технології, що призводить до виведення головного меню програми через графічний інтерфейс користувача.
2. Вибір режиму роботи з програмою.
3. Перевірка, чи обраний режим є режимом додавання відео. Якщо це не так, перехід до пункту 8.
4. Зчитування файлу, який користувач завантажив.
5. Обробка системою.
6. Перевірка валідності даних у файлі. У випадку невалідних даних перехід до пункту 13.
7. Збереження завантажених користувачем даних у базу даних. Перехід до пункту 13.
8. Перевірка обраного жанру серед наявних. Якщо введений жанр не існує – перехід до пункту 13.
9. Формування запитів до бази даних для отримання відео певного жанру.
10. Обробка даних системою.
11. Ранжування відео.
12. Виведення результатів ранжування у вигляді таблиці, де полями є назва відео, кількість переглядів, середня кількість переглядів на каналі автора.

13. Якщо користувач бажає продовжити роботу, відбувається перехід до пункту 2. Якщо користувач бажає завершити роботу, відбувається завершення діалогу з програмою.

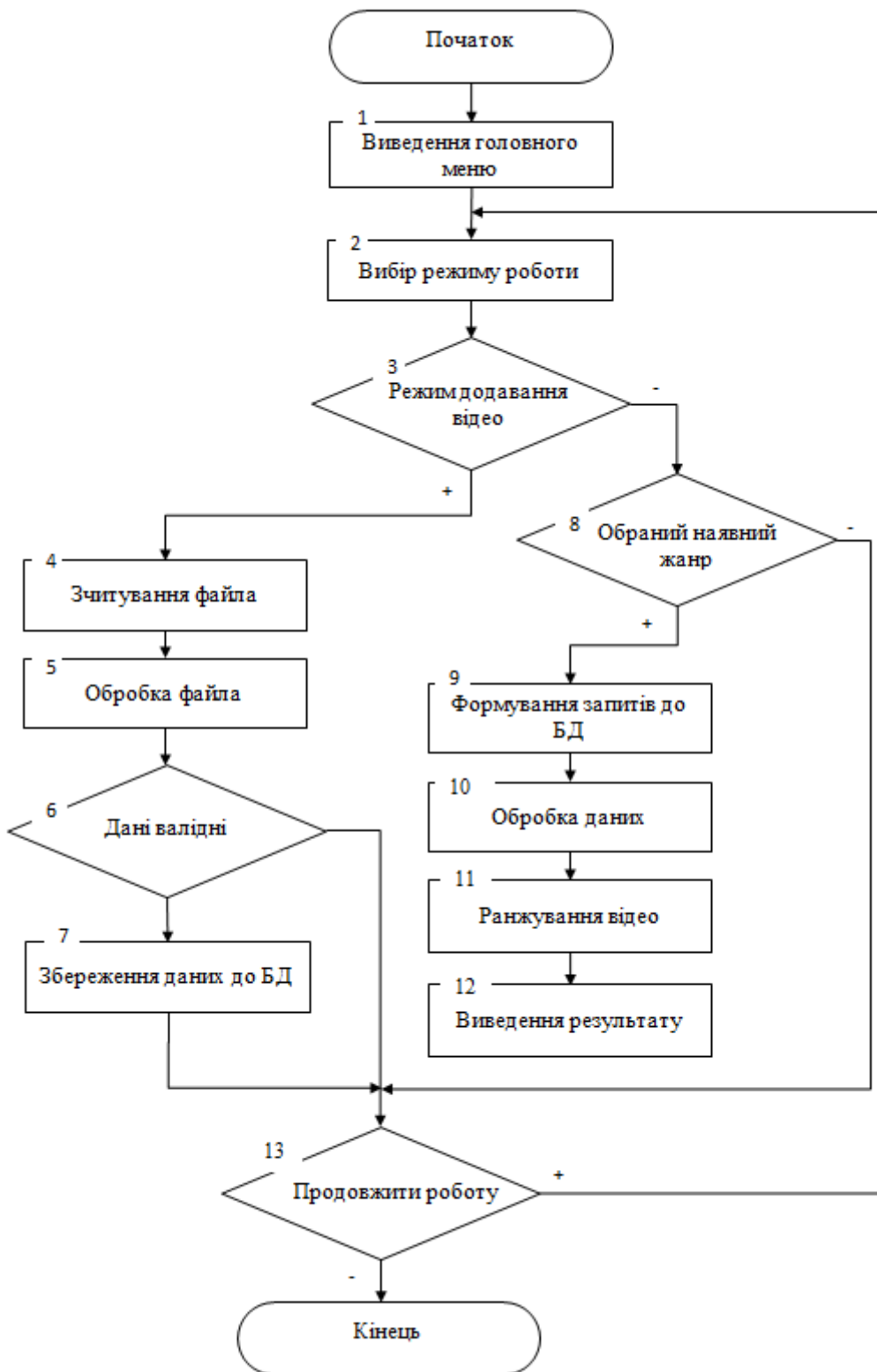


Рисунок 2.3 – Схема алгоритму функціонування модуля ранжування інформаційної технології ранжування відео на стрімінговій платформі

Отже, розроблено схему алгоритму функціонування модуля ранжування інформаційної технології ранжування відео на стрімінговій платформі.

Наступним аспектом проектування інформаційної технології ранжування відео на стрімінговій платформі є опис системи з боку користувача, тобто опис випадків використання технології, функціонал, що буде надавати користувачу з боку системи та опис окремих функціональних блоків. Для цього використаємо UML.

UML (Unified Modeling Language) – це стандартизована мова моделювання, яка використовується для візуального представлення структури та поведінки систем. UML забезпечує стандартні нотації для створення діаграм, які використовуються для аналізу, проектування та документування програмних систем [30].

Діаграма прецедентів в UML використовується для моделювання взаємодії між системою та її користувачами чи іншими системами. Ця діаграма допомагає визначити основні функції системи та визначити варіанти використання або сценарії взаємодії.

Основні елементи діаграми прецедентів включають:

- варіанти використання (Use Cases): представляють функції або дії, які може виконувати система. Кожен екземпляр зазвичай має унікальну назву;
- актори: представляють ролі або сутності, які взаємодіють із системою. Акторами можуть бути користувачі, інші системи або зовнішні сутності;
- відносини між прецедентами та акторами: використовується для визначення взаємодії між акторами та прецедентами. Зазвичай це асоціації, які показують, які прецеденти доступні кожному актору.

Наведемо діаграму прецедентів інформаційної технології ранжування відео на стрімінговій платформі (рис. 2.4). У даній діаграмі «Користувач» є головним актором, який має доступ до таких прецедентів, як «Пошук відео», «Вибір жанру», «Додавання відео», «Фільтрація відео за категоріями» та «Перегляд рекомендацій».

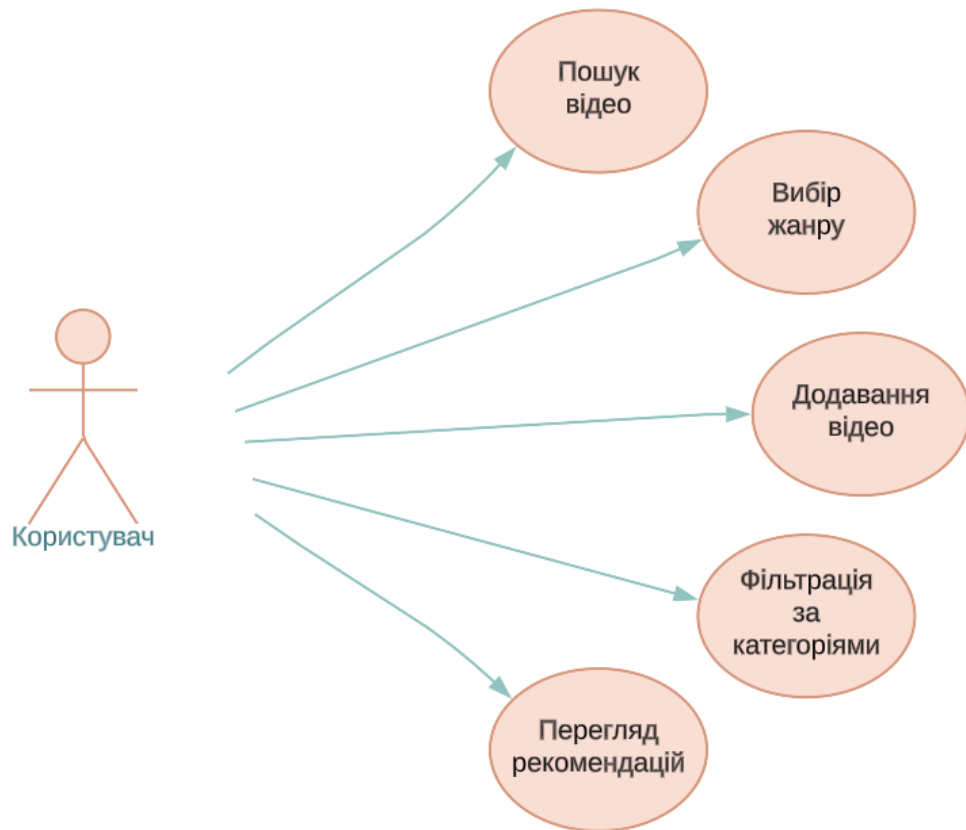


Рисунок 2.4 – Діаграма прецедентів

Як бачимо, на даній діаграмі вказаний увесь необхідний функціонал системи з боку його використання користувачем.

2.4 Обґрунтування вибору бази даних для зберігання даних інформаційної технології ранжування відео на стрімінговій платформі

Для ранжування відео на стрімінговій платформі існує необхідність у використанні сучасної бази даних для забезпечення надійного, швидкого і зручного введення, видалення, обробки та управління інформацією. Використання такої бази даних та системи управління нею дозволить значно скоротити час виконання відповідних операцій, які раніше виконувались вручну.

Вибір реляційної бази даних (РБД) для системи ранжування відео на стрімінговій платформі може бути обґрунтований кількома перевагами, зокрема [31]:

- Структурованість даних: РБД дозволяють організувати дані у вигляді таблиць зі зв'язками між ними. Це допомагає зберігати та обробляти структуровану інформацію, яка зазвичай притаманна даним про відео (такі як тривалість, кількість переглядів тощо).

- Гнучкість запитів: РБД надають потужний SQL-інтерфейс, що дозволяє легко виконувати складні запити до даних. Це корисно при вивченні показників, які впливають на рейтинг відео.

- Гарантії транзакцій: Реляційні бази даних забезпечують гарантії транзакцій, що є важливим для відтворення консистентного стану даних. Це особливо важливо в системах, де важлива точність та актуальність інформації.

- Атомарність, консистентність, ізоляція, довіреність (ACID): РБД використовують принципи ACID, що робить їх надійними та стабільними при обробці даних.

- Відносно невеликий об'єм даних: Якщо очікується, що об'єм даних буде відносно невеликим, реляційна база даних може забезпечити ефективну роботу з ним.

- Зручність управління: Багато реляційних систем баз даних (наприклад, MySQL, PostgreSQL) мають зручний інтерфейс управління та інструменти для адміністрування.

Іншими словами, вибір реляційної бази даних виправданий, коли потрібно забезпечити структуровану та надійну систему зберігання даних з можливістю ефективних операцій читання та запису, а також коли гарантується дотримання принципів ACID. Однак важливо також враховувати конкретні вимоги та обмеження вашого проекту при прийнятті рішення.

На рисунку 2.5 наведено діаграму діяльності при встановленні з'єднання з базою даних.

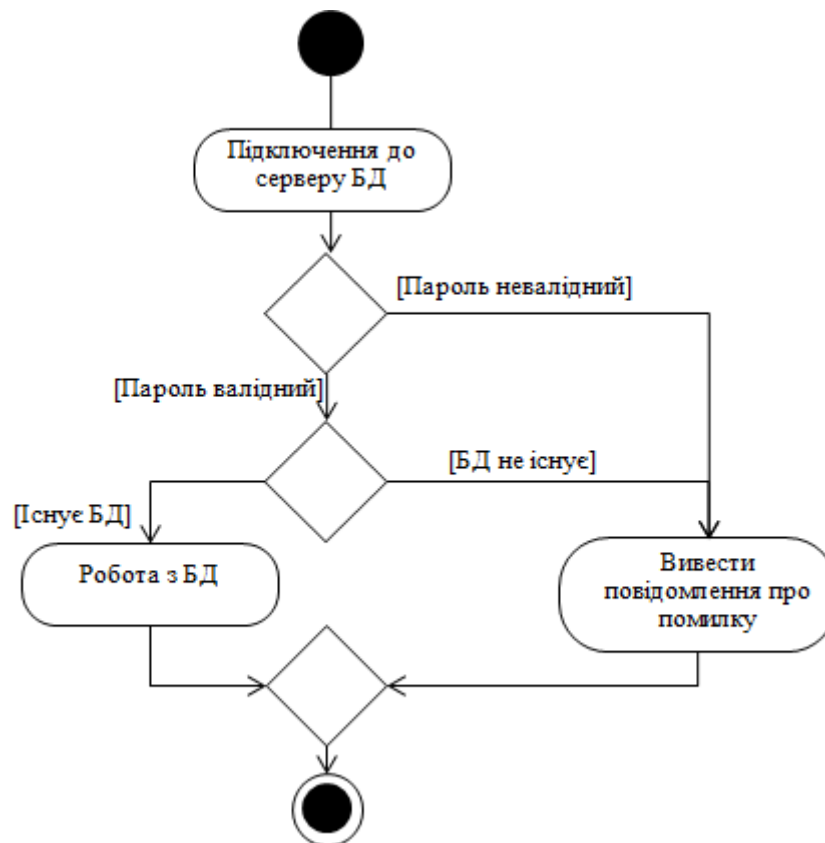


Рисунок 2.5 – Діаграма діяльності системи при встановленні з'єднання з базою даних

Реалізація бази даних відбувається мовою перетворення даних SQL. Перевагами такої мови є: незалежність від конкретної бази даних; наявність стандартів, змістовність.

Вибір мови SQL для маніпулювання даними в інформаційній технології ранжування відео на стрімінговій платформі може бути обґрунтовано кількома ключовими факторами [32]:

- SQL добре підходить для роботи з реляційними базами даних, що використовуються для зберігання даних про відео. Мова SQL надає простий та ефективний спосіб виконання різних операцій, таких як вибірка, вставка, оновлення та видалення даних.

- SQL дозволяє виконувати складні запити, які важливі для ранжування відео. Можливості групування, вибірки, сортування та об'єднання

таблиць дозволяють ефективно взаємодіяти з різними факторами, що впливають на рейтинг.

- SQL є стандартизованою мовою, що дозволяє легко переносити та взаємодіяти з базами даних різних систем. Це робить систему більш гнучкою та незалежною від конкретної технології.

- SQL базується на оптимізації запитів, що робить його ефективним для обробки великої кількості даних. Індиксація та оптимізація запитів можуть покращити продуктивність системи ранжування.

- SQL широко використовується в галузі обробки та аналізу даних, що робить її гарним вибором для інформаційних технологій ранжування відео.

- SQL забезпечує підтримку транзакцій, що важливо для збереження консистентності даних під час операцій ранжування відео.

- SQL має велику спільноту та багато ресурсів для навчання та підтримки, що допомагає вдосконалювати навички та вирішувати проблеми.

Отже, SQL є важливим елементом у розробці систем ранжування відео на стрімінгових платформах.

2.5 Висновок до розділу 2

У другому розділі було розроблено математичну модель інформаційної технології ранжування відео на стрімінговій платформі на основі методу RankBoost – алгоритму машинного навчання, який може бути ефективним для вирішення задач ранжування. Обґрунтовано використання методу для моделювання інформаційної технології. У даному розділі було розроблено структуру програмного забезпечення інформаційної технології, загальну структурну схему та схему алгоритму функціонування інформаційної технології ранжування відео на стрімінговій платформі, наведено діаграму прецедентів. Вибрано реляційну модель даних для програми ранжування відео на стрімінговій платформі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ВІДЕО НА СТРІМІНГОВІЙ ПЛАТФОРМІ

3.1 Обґрунтування вибору мови та середовища програмування

Для розробки інформаційної технології ранжування відео на стрімінговій платформі необхідно вибрати об'єктно-орієнтовану мову програмування, яка матиме значні переваги для досягнення максимальної ефективності цього завдання. Розглянемо такі мови, як C # та Java, щоб остаточно визначити придатність обраної мови.

C# представляє собою мову програмування з об'єктно-орієнтованим підходом, спрямовану на роботу в середовищі .NET [33]. Цю мову створили Андерс Галесберг, Скотт Вілтамут і Пітер Голд під егідою компанії Microsoft. Синтаксис C# подібний до мов програмування C++ та Java. Мова має строгу статичну типізацію і підтримує такі концепції, як поліморфізм, перевантаження операторів, посилання на функції-члени класу, атрибути, події, властивості, винятки та коментарі у форматі XML. C# суттєво відрізняється від своїх попередників, таких як C++, Delphi, Modules і Smalltalk. На відміну від C++, в C# відсутнє спадкування від кількох класів, і він не підтримує тип висновку, що робить його іншим від Haskell. Ці особливості C# були введені для уникнення певних проблем у розробці програмного забезпечення.

Java є мовою програмування, створеною компанією Sun Microsystems у 1995 році, яка виступає ключовим компонентом платформи Java. На сьогоднішній день вона є однією з провідних у галузі розробки програмного забезпечення, особливо в банківському та телекомунікаційному секторах. Java визначається як об'єктно-орієнтована та високотипізована мова програмування. Програми, написані на Java, зазвичай компілюються у спеціальний байт-код, що дозволяє їм працювати на будь-якій архітектурі комп'ютера за допомогою віртуальної машини Java.

Основна концепція Java полягає в мінімізації залежностей від конкретної реалізації. Код, написаний і скомпільований на Java, може використовуватися без перекомпіляції на будь-яких платформах, що підтримують Java. Скомпільований машинний код працюватиме на будь-якій віртуальній машині Java (JVM), незалежно від архітектури базового комп'ютера. Java виникла на основі C і C++, що відбивається у синтаксисі, але вона є мовою програмування вищого рівня [34].

Мови програмування C# і Java є двома популярними мовами, які використовуються для розробки програм та додатків. Обидві мови мають свої переваги і обмеження. Порівняльний аналіз мов програмування C# і Java наведено в таблиці 3.1.

Таблиця 3.1. Порівняльний аналіз мов програмування C# і Java

Назва	C#	Java
Платформонезалежність	має певну платформонезалежність завдяки технології .NET Core і .NET 5+, які дозволяють виконувати C# на різних платформах, включаючи Windows, Linux і macOS. Раніше C# була більш пов'язана з платформою Windows, але це змінилося з введенням .NET Core	відома своєю платформонезалежністю. Код Java компілюється в байт-код, який виконується на віртуальній машині Java (JVM). Це дозволяє виконувати Java-програми на різних операційних системах без модифікацій
Спільнота та Екосистема	має активну спільноту, але вона часто асоціюється з розробкою на платформі Microsoft, включаючи Windows та платформу .NET. .NET має значну кількість бібліотек та фреймворків для різних цілей	має велику та активну спільноту розробників і широкий вибір бібліотек і фреймворків. Вона використовується в різних галузях, включаючи веб-розробку, мобільні додатки, корпоративні системи та ігри
Веб-розробка	C# може бути використана для веб-розробки з використанням ASP.NET і ASP.NET Core	Java використовується для розробки веб-додатків за допомогою фреймворків, таких як Spring і Java EE
Мобільна розробка	може бути використана для розробки мобільних додатків на платформах Xamarin або для розробки під платформу Windows	використовується для розробки мобільних додатків на платформі Android

Продовження таблиці 3.1.

Мовний синтаксис	відома своєю читабельністю та лаконічністю синтаксису	синтаксис Java вважається більшим і розширеним. Деякі розробники віддають перевагу його строгості
Інтегровані розробницькі середовища	Використовується Visual Studio, яке вважається одним із найпопулярніших та потужних IDE для розробки на C#	Використовується інтегроване середовище розробки (IDE) Eclipse, IntelliJ IDEA і NetBeans
Спеціалізовані можливості	має можливості для розробки ігор за допомогою Unity	відома своєю підтримкою для вбудованих систем та розробки на Android

Обираючи між C# і Java, розробникам слід враховувати специфіку проекту, платформу та спільноту, яка вже існує для цих мов програмування. Кожна мова має свої сильні сторони і може бути вибрана відповідно до конкретних потреб та обставин проекту. Для інформаційної технології ранжування відео на стрімінговій платформі бул обрано мову програмування Java, враховуючи всі її переваги, особливо доступність, та велику кількість безкоштовних бібліотек.

Як згадувалось раніше, середовища розробки мовою Java є Eclipse, IntelliJ IDEA і NetBeans. Обираючи середовище розробки для роботи з Java, варто враховувати специфіку проекту, особисті зручності та вподобання. IntelliJ IDEA, наприклад, відома своєю продуктивністю та швидкістю, тоді як Eclipse може бути корисним у випадках, коли потрібно підтримувати декілька мов програмування. NetBeans може бути хорошим вибором для навчання або для проектів з невеликими обсягами розробки [35].

Порівняльний аналіз середовищ розробки мовою Java наведено в таблиці 3.2.

Таблиця 3.2. Порівняльний аналіз середовищ розробки мовою Java

Назва характеристики	Eclipse	IntelliJ IDEA	NetBeans
Тип	Безкоштовний та відкриті джерела	Комерційний (існує безкоштовна Community версія та платна Ultimate версія)	Безкоштовний та відкриті джерела

Продовження таблиці 3.2.

Особливості	Розширюється за допомогою плагінів, має велику спільноту розробників, підтримує різні мови програмування, включаючи Java, C++, та інші. Використовується для веб-розробки, Android-розробки та інших завдань	Фокусується на продуктивності та якості розробки. Має велику кількість корисних функцій, включаючи рефакторинг, автоматичне доповнення коду та інтеграцію з іншими інструментами	Добре підходить для веб-розробки, розробки на Java EE, розробки мобільних додатків. Має легкий інтерфейс і велику кількість плагінів
Інтерфейс	Застарілий і менш інтуїтивний інтерфейс	Сучасний, інтуїтивно зрозумілий інтерфейс	Легкий та дружній інтерфейс, легкий у використанні для початківців
Використання ресурсів	Вимагає більше ресурсів системи, що може вплинути на продуктивність на менш потужних комп'ютерах	Зазвичай має менше вимог до ресурсів системи, що робить його швидшим на менш потужних комп'ютерах	Зазвичай має помірні вимоги до ресурсів системи

З таблиці 3.2 випливає, що середовище розробки IntelliJ IDEA має переваги перед іншими у вигляді великої кількості корисних функцій, швидкості та продуктивності роботи, сучасному та інтуїтивно зрозумілому інтерфейсу, а також продуманості середовища, тому його доцільно обрати для розробки інформаційної технології ранжування відео на стрімінговій платформі.

3.2 Обґрунтування вибору бібліотек та фреймворків для інформаційної технології ранжування відео на стрімінговій платформі

Розглянемо бібліотеки мови програмування Java, що полегшують розробку інформаційної технології ранжування відео на стрімінговій платформі. Перш за все розглянемо бібліотеку Lombok. Дана бібліотека використовується для скорочення коду в класах Java та розширення функціональності мови.

Lombok – це бібліотека для скорочення коду в класах і розширення функціональності мови Java. Підключається до середовища розробки (IDE) або програмного засобу Maven, Gradle як плагін [36].

Lombok надає анотації, такі як «@Getter», «@Setter», «@NoArgsConstructor», «@AllArgsConstructor», «@Data», які автоматично генерують код для геттерів, сеттерів, конструкторів та інших стандартних методів. За допомогою Lombok можна позбутися багатьох повторюваних конструкцій, зокрема для створення об'єктів та їхнього ініціалізації.

Деякі спеціальні анотації, такі як «@Builder» і «@Value», дозволяють реалізувати деякі патерни проектування, наприклад, патерн «Builder» або створення незмінних (immutable) об'єктів.

Lombok добре інтегрований з різними IDE, такими як Eclipse, IntelliJ IDEA та іншими, що дозволяє автоматично обробляти анотації та відображати згенерований код. Lombok дозволяє користувачам використовувати свої власні анотації, які можуть бути визначені для генерації конкретного коду. Використання Lombok може покращити читабельність коду, зменшуючи кількість шаблонного коду та фокусуючись на основній логіці програми.

Плагін усуває необхідність введення вручну коду одного типу розробником, що сприяє уникненню помилок.

Робота бібліотеки ґрунтується на використанні анотацій, які виконують наступні функції:

- значно скорочують шаблонний код, роблячи його більш компактним;
- ефективно економлять час розробників;
- поліпшують читабельність коду, забезпечуючи зрозумілість його структури.

Використання Lombok дозволяє розробникам прискорити процес розробки, зменшити обсяг шаблонного коду та спростити обслуговування проектів. Однак важливо враховувати, що розуміння того, як працює Lombok, може бути корисним при виправленні можливих проблем чи розширенні функціональності.

Далі розглянемо Spring Framework, що є програмою з відкритим вихідним кодом та контейнерами, що реалізують контроль інверсії для платформи Java [37].

Основні можливості Spring Framework можуть бути використані будь-яким додатком Java, проте існують розширення для розробки веб-додатків на платформі Java EE. Незважаючи на те, що Spring Framework не обмежується конкретною моделлю програмування, він завдяки своїм можливостям став популярним у спільноті Java, що розглядає його як альтернативу або навіть доповнення до моделі Enterprise JavaBean (EJB).

Використовуючи Spring, розробники можуть скористатися перевагами POJO, уникнувши необхідності використання Enterprise Java Bean (EJB). Головною перевагою такого підходу є відсутність обов'язковості використання контейнера EJB, такого як сервер додатків, а можливість використовувати надійні контейнери сервлетів, наприклад Tomcat, чи інші комерційні продукти. Spring Framework вражає добре продуманою модульною структурою, що дозволяє розробникам ігнорувати зайві аспекти у даний момент та використовувати лише необхідне.

Замість створення нових рішень для вже вирішених проблем, Spring взаємодіє з існуючими технологіями, такими як ORM-системи (наприклад, Hibernate), Quartz, J2EE, різними бібліотеками для журналів та рядом інших презентацій.

Spring Framework містить декілька модулів, які надають широкий спектр особливостей:

Spring використовує контейнер IoC для управління компонентами та їхніми залежностями. Це дозволяє зменшити залежність між компонентами та полегшити їхню конфігурацію.

Підтримка AOP дозволяє вам розділити основні функції програми від певних аспектів, таких як логування, транзакції, безпека.

Spring MVC (Model-View-Controller): містить вбудований фреймворк для розробки веб-додатків, який використовує паттерн Model-View-Controller для розділення логіки додатку та представлення.

Spring Boot – спрощена версія Spring для швидкої розробки самостійних додатків з конвенціями над конфігурацією.

Spring Data – спрощує роботу з різними джерелами даних, включаючи бази даних, NoSQL та REST-сервіси.

Spring Security – надає засоби для впровадження аутентифікації та авторизації в додатках.

Spring Cloud – додає підтримку для розробки мікросервісних додатків, включаючи конфігурацію, обслуговування реєстрів та інші.

Spring надає спрощений доступ до баз даних через JDBC, а також підтримку ORM (Object-Relational Mapping) інструментів, таких як Hibernate.

Spring дозволяє огортати методи у транзакції, забезпечуючи атомарність та консистентність дій.

Spring надає підтримку для тестування за допомогою інструментів, таких як JUnit та TestNG.

Spring інтегрується з багатьма іншими технологіями, такими як Java EE, JPA, JMS, Quartz, Apache Camel, та інші.

Має можливість розширення функціональності за допомогою власних розширень і модулів.

Ці можливості роблять Spring Framework потужним інструментом для розробки Java-додатків, який дозволяє створювати ефективні, модульні та легко збереженні системи.

Створення графічного інтерфейсу інформаційної технології ранжування відео на стрімінговій платформі відбувається за допомогою бібліотеки Swing.

Swing – це стандартна бібліотека для створення графічних інтерфейсів у мові програмування Java. Swing містить велику кількість компонентів для розробки графічного інтерфейсу користувача (GUI) [38]. Swing представляє собою більш повний інструментарій, порівняно з AWT. Операції з компонентами AWT в цілому аналогічні, хоча вони трошки менші за розміром. Щоб відрізнити компоненти панелі інструментів, компоненти Swing мають літеру «J» перед

своєю назвою (наприклад, JButton, JPanel і так далі). У випадку компонентів AWT ця літера відсутня в початку назви.

Swing і AWT входять в Java Foundation Classes (JFC), які, в свою чергу, складають значну частину стандартної платформи Java. Класи Swing знаходяться в пакеті javax.swing та його підпакетах.

Основні компоненти та класи бібліотеки Swing включають:

JFrame: Основний контейнер для створення вікна програми.

JPanel: Контейнер для розміщення інших компонентів.

JButton: Кнопка, яку можна використовувати для виклику подій.

JLabel: Текстова мітка для відображення тексту або іконки.

JTextField: Поле для введення тексту.

JTextArea: Більше текстове поле для введення та відображення більших обсягів тексту.

JCheckBox, JRadioButton: Компоненти для вибору чи відзначення певних опцій.

JComboBox: Випадаючий список для вибору одного елемента зі списку.

Layout Managers: Менеджери розміщення (Layout Managers), такі як BorderLayout, FlowLayout, GridLayout, допомагають організувати компоненти на вікні.

Event Handling: Механізм обробки подій для реагування на взаємодію з користувачем.

При використанні Swing, можна створювати різні типи візуальних інтерфейсів, включаючи вікна, діалогові вікна, меню, панелі та інше. Низка компонентів та можливостей, які надає Swing, робить його потужним інструментом для розробки графічних інтерфейсів у Java.

Swing має численні переваги, серед яких варто відзначити наступні:

1. Крос-платформеність: будучи побудованою на платформі Java, Swing робить розроблені програми крос-платформовими. Програми, створені за допомогою Swing, можуть запускатися на різних операційних системах без жодних змін.

2. Легкість: компоненти Swing є легкими, спрощуючи користувальницький інтерфейс. Ці компоненти можуть взаємодіяти з інтерфейсом користувача операційної системи, включаючи відображення екранів та пристроїв, а також реагування на користувацькі дії, такі як натискання клавіш та рухи пальців. Swing надає широкий спектр компонентів для створення різноманітних візуальних елементів, таких як кнопки, тексти, таблиці, дерева, та інші. Це дозволяє розробникам легко створювати різні типи інтерфейсів.

3. Підключення: Swing має потужний компонент, який можна розширити для підтримки різних користувацьких інтерфейсів, сприяючи естетичному вигляду програм. Модульна архітектура дозволяє підключати інші реалізації та користувацькі фреймворки для інтерфейсів. Компоненти імпортуються через пакет з назвою `java.swing`.

4. Керований: легко керувати та налаштовувати. Механізм композиції та модель дозволяють змінювати параметри протягом виконання програми. Уніфіковані зміни можна вносити в інтерфейс користувача без жодних змін у програмному коді. Swing надає розширені можливості налаштувань та візуального оформлення інтерфейсів, що дозволяє розробникам створювати естетично приємні та зручні додатки.

Загалом, Swing надійно служить для створення графічних інтерфейсів у Java, і вона залишається популярною серед розробників.

RankLib – це бібліотека для ранжування, яка спрощує розробку та оцінювання моделей ранжування в Java [39]. Вона розроблена для реалізації різноманітних алгоритмів ранжування та надає інтерфейс для тренування, тестування та застосування цих моделей. RankLib містить реалізації різноманітних алгоритмів ранжування, таких як RankNet, RankBoost, AdaRank, Coordinate Ascent, та багато інших. Бібліотека надає простий та зрозумілий API для тренування та застосування моделей. Вона також підтримує роботу з різними форматами даних для навчання та тестування.

RankLib має функціонал для оцінки ефективності моделей ранжування за допомогою різних метрик, таких як NDCG (Normalized Discounted Cumulative

Gain), MAP (Mean Average Precision), та інших. RankLib має можливість використовувати розподілені обчислення для тренування моделей на великих обсягах даних. Як відкритий проект, RankLib має доступний вихідний код, а також активну спільноту, що дозволяє користувачам отримувати підтримку та спільно вдосконалювати бібліотеку.

Використання RankLib може бути корисним для розробників, які шукають простий та ефективний інструмент для ранжування в контексті їхніх проектів. Бібліотека робить доступними імплементації різноманітних методів ранжування, дозволяючи досліджувати та застосовувати їх для покращення результатів ранжування.

ModelMapper – це бібліотека для Java, яка надає зручний спосіб копіювання даних між об'єктами (mapping) [40]. Вона дозволяє автоматично вирівнювати поля об'єктів та виконувати конвертацію значень, спрощуючи рутинні завдання мапінгу.

Основні можливості ModelMapper включають:

Автоматичне вирівнювання полів: ModelMapper може автоматично вирівнювати поля об'єктів за їхніми іменами.

Конвертація значень: вона дозволяє виконувати конвертацію значень між різними типами даних.

Настроювані мапінги: можливість вказати власні мапінги для певних полів або типів даних.

Глибоке копіювання: можливість виконувати глибоке копіювання об'єктів, включаючи їхні асоціації та залежності.

Розширені можливості конфігурації: ModelMapper надає широкі можливості конфігурації для налаштування процесу мапінгу під потреби розробника.

jOOQ (Java Object Oriented Querying) – це бібліотека для мови програмування Java, яка дозволяє зручно та безпечно виконувати SQL-запити до баз даних [41]. Вона дозволяє розробникам використовувати SQL-запити як

об'єктно-орієнтовані конструкції, що дозволяє покращити читабельність та обслуговуваність коду.

3.3 Опис процесу програмної розробки

Для взаємодії з таблицями бази даних, як згадувалося раніше, було використано бібліотеку jOOQ. Дана бібліотека аналізує структуру бази даних і генерує Java-класи, які представляють таблиці, поля, ключі тощо. Це дозволяє працювати з базою даних, використовуючи об'єктно-орієнтований підхід. З її допомогою згенеровано класи VideoRecord, Videos, DefaultCatalog, DefaultSchema, Keys, Tables, а кожен клас починається з

```
/*
 * This file is generated by jOOQ.
 */
```

що спрощує роботу розробника.

Нижче наведено клас VideoDao, що реалізує взаємодію з базою даних.

```
import com.zamkova.jooq.generated.tables.records.VideosRecord;
import lombok.RequiredArgsConstructor;
import org.jooq.DSLContext;
import org.springframework.stereotype.Repository;

import java.util.List;

import static com.zamkova.jooq.generated.Tables.VIDEOS;

@Repository
@RequiredArgsConstructor
public class VideoDao {

    private final DSLContext context;

    public void save(VideosRecord videosRecord) {
        context.insertInto(VIDEOS)
            .set(videosRecord);
    }

    public List<VideosRecord> getByType(String type) {
```



```

        return context.selectFrom(VIDEOS)
            .where(VIDEOS.TOPIC.eq(type))
            .fetch();
    }
}

```

Фрагмент містить клас `DSLContext` – клас бібліотеки `jOOQ` (Java Object Oriented Querying), який представляє собою контекст для створення та виконання SQL-запитів. `jOOQ` використовує DSL (Domain-Specific Language) для конструювання SQL-запитів за допомогою об'єктно-орієнтованого підходу в мові програмування Java.

`DSLContext` надає ряд методів для створення різних частин SQL-запитів, таких як `SELECT`, `INSERT`, `UPDATE`, `DELETE`, а також для виконання цих запитів у базі даних. Цей клас є основним інтерфейсом для взаємодії з `jOOQ`.

В розглянутому фрагменті коду `DSLContext` використовується для створення SQL-запиту `SELECT`. Результати запиту обробляються за допомогою об'єкта `Result`, який представляє собою набір записів (`Record`).

Анотація «`@Repository`» використовується для позначення класів, які виконують роль репозиторію в підході роботи з базою даних, особливо у контексті `Spring Data`.

«`@RequiredArgsConstructor`» – це анотація у проєкті `Lombok` для миттєвого автоматичного створення конструктора з параметрами для всіх фінальних полів класу. Вона спрощує процес написання конструкторів та зменшує кількість boilerplate-коду в програмі.

Розглянемо класи Java, з допомогою яких реалізується зчитування інформації з файла.

«`java.io.File`» – це клас у Java, який представляє абстракцію файлового шляху або каталогу. Цей клас не надає фактичного читання або запису даних у файл, але його основна функція полягає в представленні інформації про файл або каталог та виконанні операцій над ними, таких як перевірка існування, отримання шляху, створення/видалення файлу або каталогу тощо.

«`java.io.FileInputStream`» є класом у Java, який представляє потік для читання даних з файлу байтами. Використовується для читання інформації з файлу, що знаходиться на диску, у вигляді послідовності байтів. Цей клас є частиною пакету «`java.io`».

«`org.apache.poi.xssf.usermodel.XSSFSheet`» є класом в бібліотеці Apache POI, яка використовується для роботи з електронними таблицями у форматі Excel (XLSX). Цей клас входить до складу Apache POI, яка є потужним інструментом для створення, читання і редагування документів Microsoft Office.

Наведемо основні функції `XSSFSheet`:

Читання та запис даних: можна отримувати доступ до комірок, рядків та колонок на аркуші для читання та запису даних.

Форматування: можна застосовувати форматування до комірок, такі як колір фону, шрифт, стиль тексту тощо.

Об'єднання та розбиття комірок: дозволяє об'єднувати та розбивати комірки для форматування даних.

Створення графіків: за допомогою `XSSFSheet` можна створювати графіки в Excel-документах.

Визначення стилів: можливість визначати стилі для різних елементів аркуша.

Робота зі списками: дозволяє вставляти та видаляти рядки, а також працювати з фільтрами.

Реалізація графічного інтерфейсу, як згадувалось раніше, відбувається при використанні `Swing`. Наведемо деякі класи даної бібліотеки.

Клас `ActionEvent` представляє подію дії, таку як натискання кнопки або вибір пункту меню. Цей клас використовується вмістом інтерфейсів подій в `AWT` та `Swing`. Цей клас часто використовується в обробниках подій для обробки дій користувача, наприклад, при взаємодії з кнопками або меню в графічних інтерфейсах користувача.

`ActionListener` – це інтерфейс, який слугує для обробки подій дій, таких як натискання кнопок чи вибір пунктів меню в графічних інтерфейсах користувача.

Клас JFrame є головним вікном програми Swing. Він представляє вікно з можливістю змінювати розмір, закривати та містити інші компоненти.

Клас JLabel використовується для відображення текстової інформації або зображень.

Клас JComboBox реалізує розкривний список для вибору одного або кількох елементів.

Використання цих класів наведено нижче:

```
public class GuiService {

    private final VideoService videoService;

    private JFrame frame;
    private JLabel label;
    private JComboBox comboBox;

    public void createMainPage() {
        frame = new JFrame();
        frame.setTitle("Ранжування відео");
        frame.setSize(400, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 11);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);

        JButton hideBtn = new JButton();
        hideBtn.setText("Додати відео");
        hideBtn.setFont(font1);
        label.add(hideBtn, new GridBagConstraints(0, 0, 2, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.HORIZONTAL, new Insets(10, 0, 10, 0), 0, 0));

        String[] arr = new String[]{"", "Автомобілі", "Відеоігри", "Історія", "Наука",
"Політика", "Спорт", "Технології"};

        comboBox = new JComboBox(arr);
        comboBox.setFont(font1);
        label.add(comboBox, new GridBagConstraints(0, 0, 2, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));
    }
}
```

```

        JButton hide = new JButton();
        hide.setText("Переглянути рейтинг відео");
        hide.setFont(font1);
        label.add(hide, new GridBagConstraints(0, 2, 2, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
        GridBagConstraints.HORIZONTAL, new Insets(10, 0, 10, 0), 0, 0));

        hideBtn.addActionListener(new SelectListener());
        hide.addActionListener(new HideListener());

        frame.add(label);
        frame.setVisible(true);
    }

```

Наведений фрагмент описує створення основної сторінки інформаційної технології ранжування відео на стрімінговій платформі з використанням бібліотеки для створення графічного інтерфейсу користувача Swing.

3.4 Опис класів і функцій інформаційної технології ранжування відео на стрімінговій платформі

Для надання повної інформації про функціональність кожного класу та розкриття залежності та взаємодії між класами наведемо UML-діаграму класів. UML діаграма класів – це графічне представлення структури та зв'язків класів у програмній системі. Основна мета – показати класи, їхні атрибути та методи, а також взаємодію між ними

На рисунку 3.1 зображено загальну UML-діаграму класів інформаційної технології ранжування відео на стрімінговій платформі.

Вона включає такі розроблені класи Video, VideoDao, VideoParser, VideoService, ModelMapperConfig, RankingVideo, RankingTable, GuiService, HideListener, BackListener, SelectListener. Вони взаємодіють між собою, надаючи функціональність для роботи з відео, ранжування та взаємодії з графічним інтерфейсом.

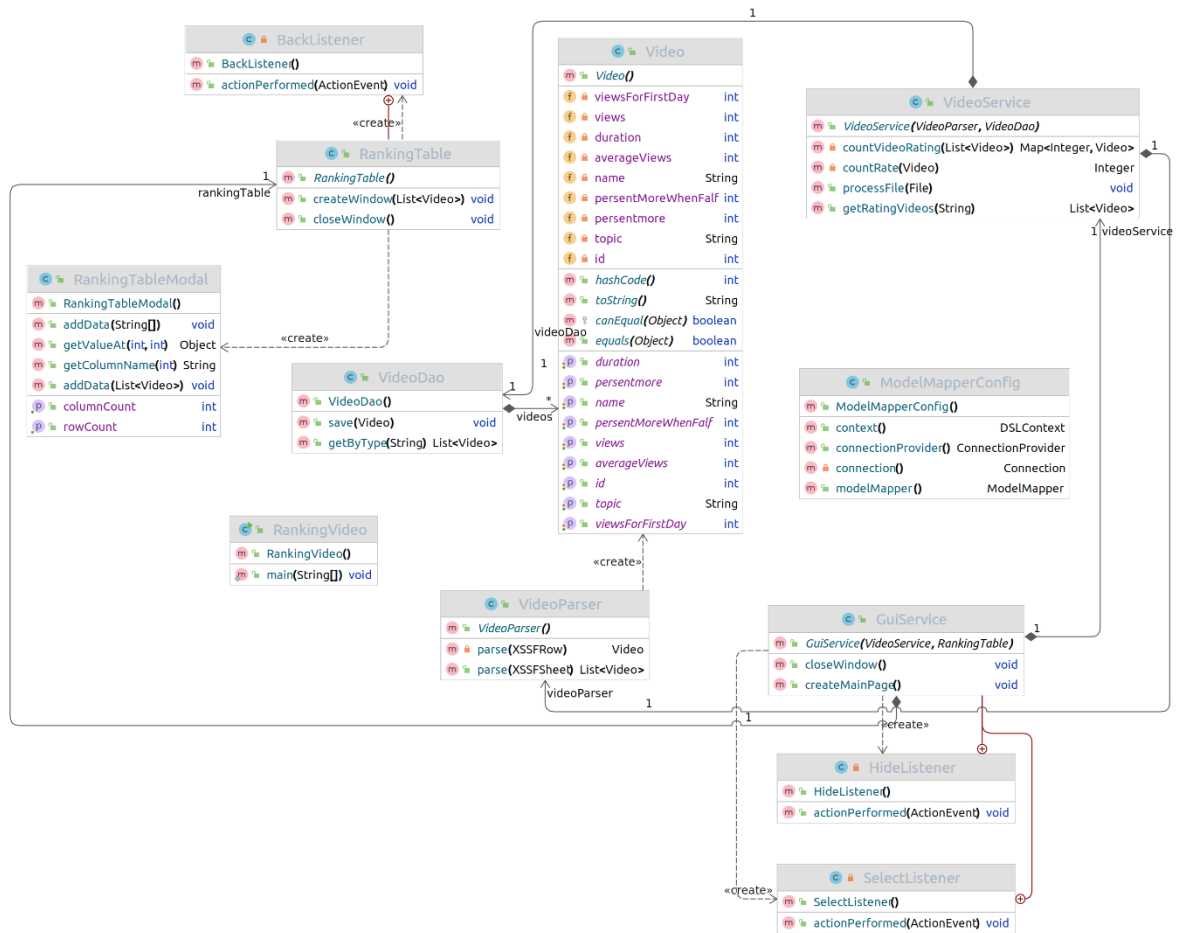


Рисунок 3.1 – Загальна UML-діаграма класів

Клас Video представляє відео на стрімінговій платформі. Він містить основну інформацію про відео, таку як назва, тривалість, автор, кількість переглядів тощо.

VideoDao – клас доступу до даних для відео. Основні його функції завантаження даних про відео з бази даних та збереження даних відео в базу даних. Іншими словами це клас для отримання, оновлення та зберігання інформації про відео в базі даних.

Клас VideoParser – клас для аналізу відео, вилучення метаданих та іншої інформації про відео. Для отримання даних із файлів у форматі xlsx використовується публічний клас XSSFSheet. Цей клас надає можливість взаємодії з електронними таблицями, що дозволяє здійснювати операції зчитування та обробки вмісту цих таблиць.

Клас VideoService – сервісний клас для роботи з відео, що містить алгоритм ранжування відео на стрімінговій платформі. У класі формується рейтинг відео для запису в таблицю.

Клас RankingVideo представляє відео в контексті ранжування, містить дані, що впливають на рейтинг відео. Таблиця з результатами ранжування розроблена в класі RankingTable, що включає функції для сортування та відображення рейтингів відео у таблиці.

Клас ModelMapperConfig – конфігурація ModelMapper для відображення об'єктів між різними моделями.

Клас GuiService – сервіс для взаємодії з графічним інтерфейсом.

Класи HideListener, BackListener та SelectListener відповідають за обробку подій в графічному інтерфейсі. Перший обробляє подію приховування елементів, другий – подію повернення та виходу, третій – подію вибору елементів в інтерфейсі.

Отже, у рамках даного підрозділу було здійснено реалізацію інформаційної технології ранжування відео на стрімінговій платформі, розроблено загальну UML-діаграму класів та описано класи, а також інструменти, використані для їх розробки.

3.5 Тестування та аналіз результатів роботи інформаційної технології ранжування відео на стрімінговій платформі

Після завершення розробки програмного продукту важливо переконатися в його коректності, легкості використання та здатності вирішувати поставлені завдання. Тестування дає змогу переконатися, що ранжування відео відбувається вірно з урахуванням всіх вказаних факторів та враховує різні параметри, такі як кількість переглядів, тривалість, тематика тощо. Проводиться тестування на різних типах відео (короткі, тривалі, популярні, нові).

В результаті проведення навантажувального тестування відбувалося симуляція великої кількості користувачів і виявлено, що технологія впоралася з

навантаженням. Також ранжування залишається ефективним та швидким навіть при великому обсязі даних.

Тестування продуктивності показало, що середній час відгуку менше ніж 100 мілісекунд, а кількість оброблених запитів за секунду – не менше 200, що означає високу продуктивність програмного забезпечення. При довготривалому тестуванні продукт залишався стабільним та не викликав великої кількості помилок.

При запуску програми спочатку відкривається вікно початкової активності, яке включає кнопку «Додати відео», поле із спливаючим меню для введення категорії відео, за якою відбувається ранжування, та кнопку «Переглянути рейтинг відео». Воно є головним меню програми, де користувач може обрати необхідний пункт для подальшої роботи (рисунок 3.2).

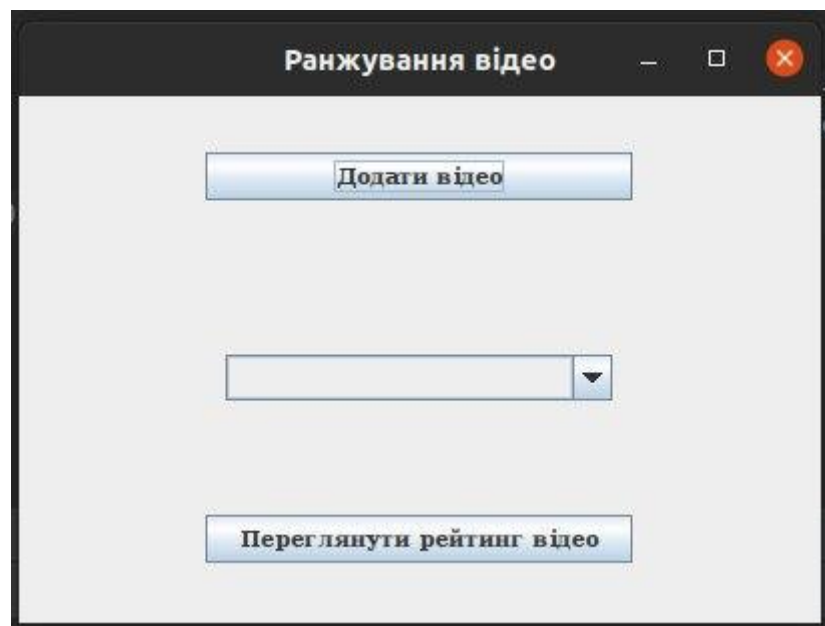


Рисунок 3.2 – Головне вікно програми

Коли користувач натисне кнопку «Додати відео», відкриється діалогове вікно завантаження файлу (рисунок 3.3). Потрібно обрати файл, що містить інформацію про відео для його подальшого опрацювання. У цьому діалоговому вікні відображаються наявні файли, і після вибору одного з них користувач натискає кнопку «Open».

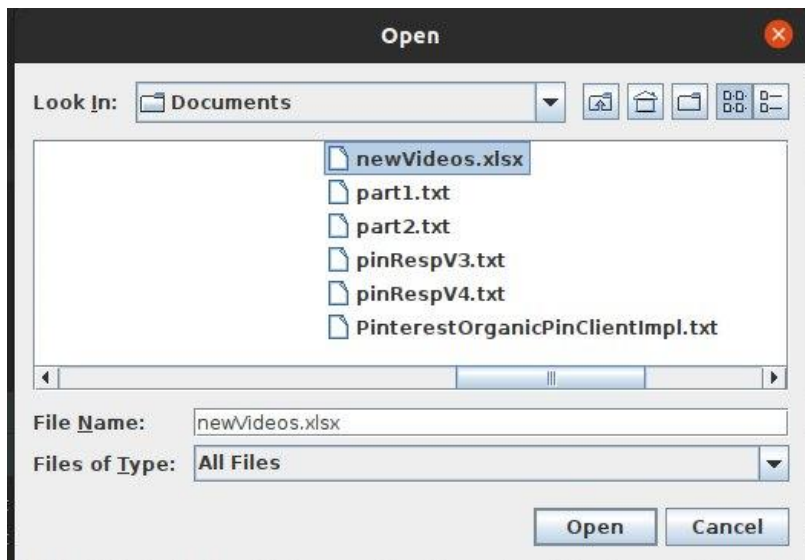


Рисунок 3.3 – Завантаження файла

На рисунку 3.3 видно, що користувач вибирає файл формату `xlsx`, оскільки він містить таблицю з параметрами відео. Програма здійснює зчитування файлу та обробку вмісту. Якщо дані є валідними, вони зберігаються до бази даних.

Для проведення ранжування відео користувач обирає категорію, за якою буде відбуватися ранжування. Наприклад, вибір категорії «Спорт» представлено на рисунку 3.4. Після вибору категорії користувач натискає кнопку «Переглянути рейтинг відео».

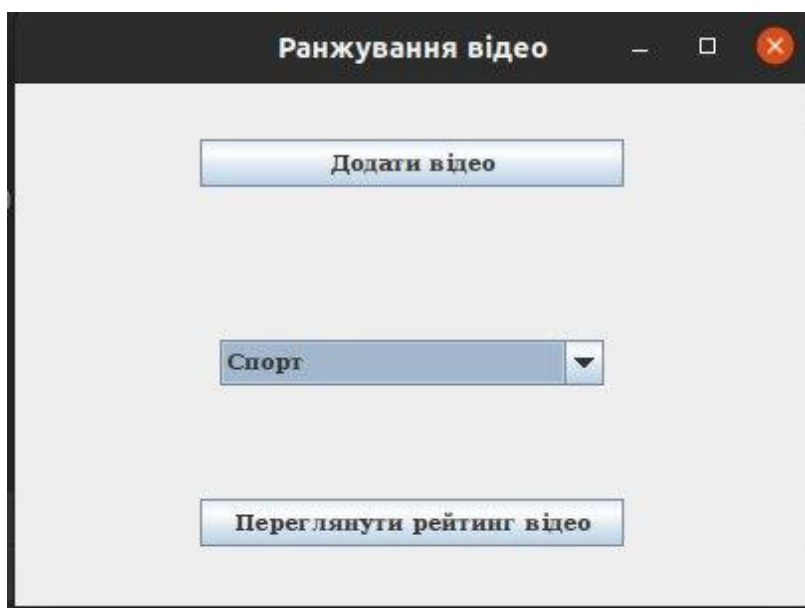
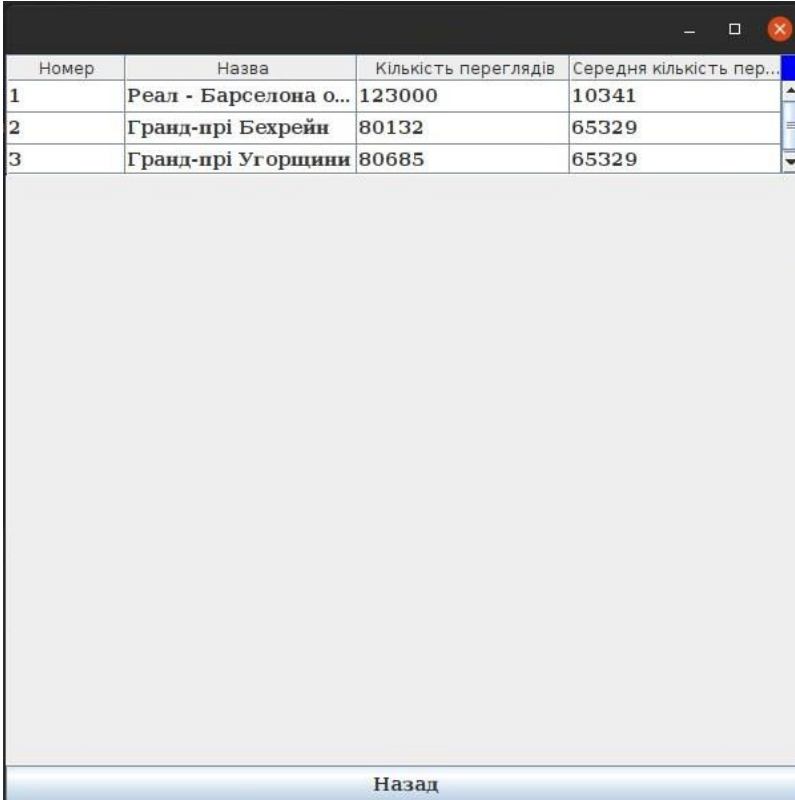


Рисунок 3.4 – Вікно вибору категорії

Результатом ранжування є таблиця з проранжованими відео (рисунок 3.5), яка містить такі поля:

- номер;
- назва;
- кількість переглядів;
- середня кількість переглядів на каналі.



Номер	Назва	Кількість переглядів	Середня кількість пер...
1	Реал - Барселона о...	123000	10341
2	Гранд-прі Бехрейн	80132	65329
3	Гранд-прі Угорщини	80685	65329

Назад

Рисунок 3.5 – Вікно результату ранжування

При виборі користувачем категорії «Історія» з головного меню програми виводиться рейтинг відео за вказаною категорією (рис. 3.6 – 3.7).

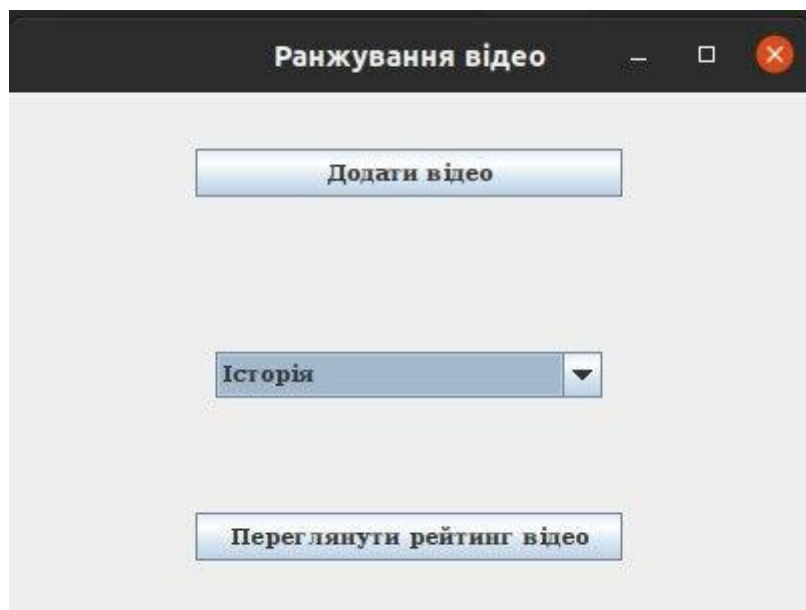


Рисунок 3.6 – Вікно вибору категорії

Номер	Назва	Кількість переглядів	Середня кількість пе...
1	Коротка історія Укра...	265321	102156
2	Війни в Україні	97252	102156
3	Берлінська стіна	218368	201478
4	Історія Гонконга	237879	201478
5	Історія козацтва	74258	102156

Рисунок 3.7 – Вікно результату ранжування

Проведено тестування розробленої інформаційної технології ранжування відео на стрімінговій платформі, що підтвердило коректність її роботи.

Здійснено 500 запусків програмного застосунку, в ході яких були інтенсивно протестовані його функціональні можливості.

Розробивши набір тестових відеофайлів різного типу (короткі, тривалі, різних жанрів і тематик), була застосована інформаційна технологія ранжування відео на стрімінговій платформі. Цей тестовий набір надав змогу докладно оцінити ефективність та надійність розробленого додатку. Результати ранжування були збережені та порівняні з аналогом, на якому використовувався той самий набір тестів.

Тестування також включало використання реальних даних з існуючих стрімінгових платформ для порівняння результатів ранжування. Проводилось зіставлення рангів, присвоєних розробленою технологією, з реальними рейтингами відео на платформах.

У таблиці 3.3 наведено результати тестування розробленого додатку в порівнянні з аналогічним програмним забезпеченням. Цей аналіз дозволив визначити точність ранжування та виявити переваги розробленого продукту в порівнянні з аналогами.

Таблиця 3.3 – Результати тестування розробленого додатку порівняно з аналогом

Назва	Точність ранжування
Розроблений додаток	79,3%
Аналог	74,8%

Отже, з аналізу представлених результатів у таблиці 3.3 випливає, що ранжування за допомогою розробленої інформаційної технології відео на стрімінговій платформі виявило високий рівень точності та ефективності порівняно з аналогом. Проведені тести підтвердили, що розроблений додаток дозволяє досягти покращення точності ранжування релевантного, популярного та якісного відео на платформі.

Відзначимо, що застосування розробленої інформаційної технології призвело до зростання точності пошуку на 4,5%, що вказує на високу ефективність та вдалу реалізацію поставленої мети. Отримані результати свідчать про переваги та конкурентоспроможність розробленого додатку в контексті ранжування відео, що може бути важливим фактором для кінцевих користувачів та платформи в цілому.

3.6 Висновок до розділу 3

В третьому розділі було здійснено обґрунтування мови програмування для реалізації інформаційної технології ранжування відео на стрімінговій платформі. Вибрано об'єктно-орієнтовану мову Java враховуючи всі її переваги, особливо доступність, та велику кількість безкоштовних бібліотек. Інформаційна технологія розроблена в середовищі IntelliJ IDEA, що має багато переваг, зокрема задоволення усіх вимог розробника, розумне доповнення коду. Розглянуто та обґрунтовано використання бібліотек та фреймворків для програмної реалізації інформаційної технології.

Було розроблено UML-діаграму класів програми, наведено опис класів та їх функції. На основі цього було програмно реалізовано інформаційну технологію ранжування відео на стрімінговій платформі.

Проведено тестування інформаційної технології, що підтвердило очікувані результати, підвищивши точність ранжування на 4,5% порівняно з аналогом.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Інформаційна технологія ранжування відео на стрімінговій платформі» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія ранжування відео на стрімінговій платформі» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [42].

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

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 4.2.

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

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

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

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	4	5
2. Ринкові переваги (наявність аналогів)	3	2	3
3. Ринкові переваги (ціна продукту)	3	4	4
4. Ринкові переваги (технічні властивості)	2	3	2
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	4	3	3
7. Ринкові перспективи (конкуренція)	3	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	5	4	5
12. Практична здійсненність (розробка документів)	5	5	4
Сума балів	44	43	43
Середньоарифметична сума балів $СБ_c$	43,3		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [42].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

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

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія ранжування відео на стрімінговій платформі» становить 43,3 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розраховуємо за формулою [43]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має

виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де I_{ni} та I_{ai} – чисельні значення конкретного i -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Швидкодія	с	0,8	0,4	2	0,1
Швидкість доступу до даних	мкс	210	140	1,5	0,4
Кількість обробки запитів на хвилину	шт.	60	100	1,67	0,3
Місце на диску	Мб	1200	200	6	0,1
Потреба в оперативній пам'яті	Гб	16	8	2	0,1

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 2 \cdot 0,1 + 1,5 \cdot 0,4 + 1,67 \cdot 0,3 + 6 \cdot 0,1 + 2 \cdot 0,1 = 2,10.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,10 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія ранжування відео на стрімінговій платформі», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

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

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [42]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 18350,00 \cdot 42 / 21 = 36700,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту з розробки і дослідження програмного продукту	18350,00	873,81	42	36700,00
Консультант (Аналітик з розробки інформаційних систем)	18000,00	857,14	8	6857,14
Інженер-дослідник (програміст 1-ї категорії)	16540,00	787,62	42	33080,00
Технік (без категорії)	7200,00	342,86	21	7200,00
Всього				83837,14

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія ранжування відео на стрімінговій платформі» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=6700,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [42];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_l = 6700,00 \cdot 1,10 \cdot 1,35 / (21 \cdot 8) = 59,22 \text{ грн.}$$

$$Z_{pl} = 59,22 \cdot 4,25 = 251,70 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення офісного обладнання робочого місця розробника програмного забезпечення	4,25	2	1,10	59,22	251,70
Інсталяція програмного забезпечення	6,80	4	1,50	80,76	549,16
Формування дослідної бази даних інформації	16,50	3	1,35	72,68	1199,27
Налагодження програмних блоків	2,10	5	1,70	91,53	192,21
Монтаж серверного обладнання	3,30	4	1,50	80,76	266,50
Монтаж додаткових блоків пам'яті	1,55	3	1,35	72,68	112,66
Тестування системи	12,30	4	1,50	80,76	993,33
Всього					3564,83

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{доп}} = (Z_o + Z_p) \cdot \frac{H_{\text{доп}}}{100\%}, \quad (4.7)$$

де $H_{\text{доп}}$ – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{доп}} = (83837,14 + 3564,83) \cdot 10 / 100\% = 8740,20 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

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

$$Z_n = (Z_o + Z_p + Z_{\text{доп}}) \cdot \frac{H_{zn}}{100\%} \quad (4.8)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (83837,14 + 3564,83 + 8740,20) \cdot 22 / 100\% = 21151,28 \text{ грн.}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія ранжування відео на стрімінговій платформі».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\epsilon j}, \quad (4.9)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$C_{\epsilon j}$ – вартість відходів j -го найменування, грн/кг.

$$M_1 = 3,0 \cdot 220,00 \cdot 1,05 - 0 \cdot 0 = 693,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний А4	220,00	3,0	0	0	693,00
Папір для заміток А5	120,00	4,0	0	0	504,00
Начиння канцелярське	221,00	3,0	0	0	696,15
Органайзер офісний	205,00	4,0	0	0	861,00
Картридж для принтера HP-1600	2150,00	2,0	0	0	4515,00
Диск оптичний (CD-R)	18,00	2,0	0	0	37,80
FLASH-пам'ять (16 ГБ)	132,00	1,0	0	0	138,60
Всього					7445,55

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему «Інформаційна технологія ранжування відео на стрімінговій платформі», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_e = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.10)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$K_e = 2 \cdot 3200,00 \cdot 1,1 = 7040,00$ грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Пам'ять (SSD диск) Samsung 870 QVO 1TB 2.5" V-NAND 4bit MLC (QLC) SATA III (MZ-77Q1T0BW)	2	3200,00	7040,00
Всього			7040,00

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^k C_i \cdot C_{пр.i} \cdot K_i, \quad (4.11)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань устаткування.

$$B_{спец} = 35899,00 \cdot 1 \cdot 1,1 = 39488,90 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Серверне обладнання на основі ПК ZEVS PC 13430U i5 9400F + GTX 1060 3GB	1	35899,00	39488,90
Засоби передачі даних	1	6230,00	6853,00
Всього			46341,90

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{npz} = \sum_{i=1}^k C_{inprz} \cdot C_{npz.i} \cdot K_i, \quad (4.12)$$

де C_{inprz} – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{npz.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{npz} = 7400,00 \cdot 1 \cdot 1,05 = 7770,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Прикладне програмне забезпечення IntelliJ Idea	1	7400,00	7770,00
Прикладне програмне забезпечення системи управління реляційними базами даних MySQL Workbench	1	5320,00	5586,00
Всього			13356,00

4.3.7 Амортизація обладнання, програмних засобів та приміщень

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

$$A_{обл} = \frac{Ц_{б}}{T_{г}} \cdot \frac{t_{вик}}{12}, \quad (4.13)$$

де C_b – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

T_v – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (34499,00 \cdot 2) / (3 \cdot 12) = 1916,61 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника ПЗ (Ноутбук Acer Aspire 7 A715-43G-R92H)	34499,00	3	2	1916,61
Робоче місце інженера-розробника інформаційної технології ранжування відео на стрімінговій платформі	7320,00	7	2	174,29
Робоче місце керівника проекту	6500,00	7	2	154,76
Серверне обладнання на основі ПК ZEVS PC 13430U i5 9400F + GTX 1060 3GB	39488,90	3	2	2193,83
Засоби передачі даних	6853,00	4	2	285,54
Офісна оргтехніка	9350,00	5	2	311,67
ОС Windows 11	6200,00	3	2	344,44

Продовження таблиці 4.11.

Прикладний пакет Microsoft Office 2021 Professional Plus	5800,00	3	2	322,22
Приміщення лабораторії розробки	320000,00	25	2	2133,33
Всього				7836,69

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.14)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,50$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,06 \cdot 320,0 \cdot 7,50 \cdot 0,95 / 0,97 = 144,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника ПЗ (Ноутбук Acer Aspire 7 A715-43G-R92H)	0,06	320,0	144,00

Продовження таблиці 4.12.

Робоче місце інженера-розробника інформаційної технології ранжування відео на стрімінговій платформі	0,10	320,0	240,00
Робоче місце керівника проекту	0,05	320,0	120,00
Серверне обладнання на основі ПК ZEVS PC 13430U i5 9400F + GTX 1060 3GB	0,40	250,0	750,00
Засоби передачі даних	0,04	120,0	36,00
Офісна оргтехніка	0,55	6,0	24,75
Всього			1314,75

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія ранжування відео на стрімінговій платформі» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (Z_o + Z_p) \cdot \frac{H_{св}}{100\%}, \quad (4.15)$$

де $H_{св}$ – норма нарахування за статтею «Службові відрядження», прийнемо $H_{св} = 20\%$.

$$B_{св} = (83837,14 + 3564,83) \cdot 20 / 100\% = 17480,40 \text{ грн.}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.16)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 30\%$.

$$B_{cn} = (83837,14 + 3564,83) \cdot 30 / 100\% = 26220,59 \text{ грн.}$$

4.3.11 Інші витрати

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

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, \quad (4.17)$$

де H_{ie} – норма нарахування за статтею «Інші витрати», прийmemo $H_{ie} = 65\%$.

$$I_e = (83837,14 + 3564,83) \cdot 65 / 100\% = 56811,28 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

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

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.18)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 120\%$.

$$B_{нзв} = (83837,14 + 3564,83) \cdot 120 / 100\% = 104882,37 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія ранжування відео на стрімінговій платформі» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{дод} + Z_n + M + K_v + B_{специ} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_v + B_{нзв}. \quad (4.19)$$

$$B_{заг} = 83837,14 + 3564,83 + 8740,20 + 21151,28 + 7445,55 + 7040,00 + 46341,90 + 13356,00 + 7836,69 + 1314,75 + 17480,40 + 26220,59 + 56811,28 + 104882,37 = 406022,99 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.20)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta=0,9$.

$$3B = 406022,99 / 0,9 = 451136,66 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія ранжування відео на стрімінговій платформі» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	10000	12000	15000	10000

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 145000 осіб;

C_o – вартість послуги у році до впровадження інформаційної системи, прийmemo 52,00 грн;

$\pm \Delta C_o$ – зміна вартості послуги від впровадження результатів, прийmemo зростання на +42,25 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів

від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [42]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.21)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 35\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (42,25 \cdot 145000,00 + 94,25 \cdot 10000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 1683920,78 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (42,25 \cdot 145000,00 + 94,25 \cdot 22000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 1953342,01 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (42,25 \cdot 145000,00 + 94,25 \cdot 37000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 2290118,54 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (42,25 \cdot 145000,00 + 94,25 \cdot 47000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 2514636,23 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (4.22)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau=0,15$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 1683920,78/(1+0,15)^1 + 1953342,01/(1+0,15)^2 + 2290118,54/(1+0,15)^3 + \\ &+ 2514636,23/(1+0,15)^4 = 1464278,94 + 1477007,19 + 1505790,12 + 1437751,43 = \\ &= 5884827,68 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.23)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв}=2$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 451136,66 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 451136,66 = 902273,32 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (4.24)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 5884827,68 грн;

PV – теперішня вартість початкових інвестицій, 902273,32 грн.

$$E_{abc} = III - PV = 5884827,68 - 902273,32 = 4982554,36 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_e = T_{жс} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.25)$$

де E_{abc} – абсолютний економічний ефект вкладених інвестицій, 4982554,36 грн;

PV – теперішня вартість початкових інвестицій, 902273,32 грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_e = T_{жс} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 4982554,36/902273,32)^{1/4} - 1 = 0,60.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min} :

$$\tau_{min} = d + f, \quad (4.26)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,12$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,4.

$\tau_{min} = 0,12 + 0,4 = 0,52 < 0,60$ свідчить про те, що внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія ранжування відео на стрімінговій платформі» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_е}, \quad (4.27)$$

де $E_е$ – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,60 = 1,67 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновок до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія ранжування відео на стрімінговій платформі» становить 43,3 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,10 рази.

Також термін окупності становить 1,67 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія ранжування відео на стрімінговій платформі».

ВИСНОВКИ

Всі завдання, поставлені для реалізації інформаційної технології ранжування відео на стрімінговій платформі виконані в повному обсязі, а саме: провести аналіз програм-аналогів та обґрунтувати доцільність розробки інформаційної технології ранжування відео на стрімінговій платформі; розробити структуру інформаційної технології ранжування відео на стрімінговій платформі; розробити математичну модель ранжування відео на стрімінговій платформі; здійснити програмну реалізацію інформаційної технології ранжування відео на стрімінговій платформі; провести тестування програми та проаналізувати отримані результати; економічно обґрунтувати доцільність розробки інформаційної технології ранжування відео на стрімінговій платформі.

Під час виконання магістерської кваліфікаційної роботи було реалізовано інформаційну технологію ранжування відео на стрімінговій платформі.

Обґрунтовано доцільність розробки інформаційної технології ранжування відео на стрімінговій платформі. Проведено аналіз сучасних аналогів, які використовуються для ранжування відео на стрімінгових платформах. Виявлено проблему недостатньої точності ранжування відео, що довело актуальність розробки інформаційної технології ранжування відео на стрімінговій платформі.

Було розроблено структуру інформаційної технології ранжування відео на стрімінговій платформі, структурну схему функціонування інформаційної технології, спроектовано схему алгоритму роботи інформаційної технології та наведено діаграму прецедентів. Розглянуто модульний підхід до проектування структури інформаційної технології ранжування відео на стрімінговій платформі.

На основі аналізу існуючих моделей та методів, що застосовуються для вирішення поставленої задачі було обґрунтовано використання в інформаційній технології алгоритму ранжування RankBoost, що можна виразити у вигляді вагової комбінації базових моделей, які увійшли в розроблену математичну

модель ранжування відео на стрімінговій платформі, що дозволить підвищити точність ранжування відео.

Здійснено програмну реалізацію інформаційної технології ранжування відео на стрімінговій платформі. Обґрунтовано вибір мови програмування. Розробка інформаційної технології велася на об'єктно-орієнтованій мові програмування Java в середовищі розробки IntelliJ IDEA. Наведено основні бібліотеки і фреймворки вибраної мови, що доцільно використовувати в інформаційній технології. Описано програмні рішення для програмної реалізації інформаційної технології ранжування відео на стрімінговій платформі, детально описано класи та функції, які використовуються в інформаційній технології.

Проведено тестування розробленої програми та підтверджено її працездатність. Порівняно з прототипом точність ранжування зросла на 4,5%, що означає доведення поставленої мети.

Здійснено економічне обґрунтування доцільності розробки інформаційної технології ранжування відео на стрімінговій платформі. Проведено оцінювання комерційного потенціалу розробки. Проведено технологічний аудит із залученням експертів. Згідно висновків експертів, рівень комерційного потенціалу розробки є високим. Здійснено прогнозування витрат на виконання науково-дослідної роботи. Розраховано витрати на заробітну плату та амортизаційні відрахування, витрати на силову електроенергію. Загальні витрати становлять 451136,66 грн.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Замкова Т.Д. Перспективи розробки інтелектуального модулю ранжування відео на стрімінговій платформі. Матеріали LI науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2022) : збірник доповідей. [Електронний ресурс]. Вінниця : ВНТУ, 2022. С. 829–832. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15715> (дата звернення: 26.09.2023).
2. Замкова Т. Д. Перспективи розробки інформаційної технології ранжування відео на стрімінговій платформі / Т. Д. Замкова, С. І. Петришин // Матеріали LI науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023) : збірник доповідей [Електронний ресурс]. – Вінниця : ВНТУ, 2023. – С. 511–513. – Режим доступу: <https://press.vntu.edu.ua/index.php/vntu/catalog/view/788/1373/2632-1> (дата звернення: 26.09.2023). – Назва з екрана.
3. Інформаційні системи і технології. Основні етапи розвитку [Електронний ресурс]. URL: <https://moodle.znu.edu.ua/mod/page/view.php?id=267447> (дата звернення: 26.09.2023).
4. Пошукова оптимізація у 2022 році: основні SEO-тренди [Електронний ресурс]. URL: <https://apollon.guru/seo/trendy-seo-prosuvannya/> (дата звернення: 26.09.2023).
5. Савченко А. С., Синельников О. О. Методи та системи штучного інтелекту: навч. посіб. Київ: НАУ, 2017. 176 с.
6. Катренко А. В., Пастернак О. В. Проблема оптимальності в теорії та практиці прийняття рішень // Вісник Національного університету «Львівська політехніка». 2015. № 829. с. 359–373.
7. Найкращі способи оптимізації відео для пошуку [Електронний ресурс]. URL: <https://www.ranktracker.com/uk/blog/best-ways-to-optimize-your-video-for-search/> (дата звернення: 26.09.2023).

8. Video SEO: A Comprehensive Guide for 2023 [Електронний ресурс]. URL: <https://www.getshow.io/blog/video-seo-optimization-guide/> (дата звернення: 26.09.2023).
9. What is YouTube SEO and 21 Tips To Rank Higher [Електронний ресурс]. URL: <https://rockcontent.com/blog/youtube-seo/> (дата звернення: 26.09.2023).
10. Here's What We Learned About SEO [Електронний ресурс]. URL: <https://backlinko.com/search-engine-ranking> (дата звернення: 26.09.2023).
11. How to get videos to rank [Електронний ресурс]. URL: <https://www.semrush.com/> (дата звернення: 26.09.2023).
12. Leon Yin, Megan Brown. Youtube-data-api Documentation, 2019. [Електронний ресурс]. URL: https://youtube-data-api.readthedocs.io/_/downloads/en/latest/pdf/ (дата звернення: 05.05.2022).
13. Francesco R., Lior R., Bracha S., Paul K. B. Recommender Systems Handbook. Dordrecht: Springer, 2015. 1009 p.
14. Щедрина О. І., Агутін М. М. Інтернет-технології в бізнесі: навч. посіб. Київ: КНЕУ, 2012. 303 с.
15. SEO Cyfe YouTube [Електронний ресурс]. URL: <https://www.cyfe.com/blog/youtube-seo/> (дата звернення: 01.10.2023).
16. Мелешко Є.В., Семенов С.Г., Хох В.Д. Дослідження методів побудови рекомендаційних систем в мережі Інтернет. Інформаційні технології. 2018. №1 (47). С. 131-136.
17. Ch. M. Bishop. Pattern Recognition and Machine Learning. Singapore: Springer Science+Business Media, 2006. 758 p.
18. Кононова К. Ю. Машинне навчання: методи і моделі: підручник. Харків: ХНУ ім. В. Н. Каразіна, 2020. 301с.
19. Добровська Л. М., Добровська І. А. Теорія та практика нейронних мереж: навч. посіб. Київ: НТУУ «КПІ» Вид-во «Політехніка», 2015. 396 с.
20. Кульчицька Х. Б., Предко Л. С. Застосування методу аналізу ієрархій при виборі проекту в поліграфії // Поліграфія і видавнича справа. 2018. №1.

[Електронний ресурс]. URL: http://nbuv.gov.ua/UJRN/Pivs_2018_1_7 (дата звернення: 26.09.2023).

21. Ходаков В., Кирюшатова Е. Застосування методу аналізу ієрархий для підбору кадрів. Проблеми інформаційних технологій с.205

22. How the YouTube Algorithm Works in 2023: The Complete Guide [Електронний ресурс]. URL: <https://blog.hootsuite.com/how-the-youtube-algorithm-works/> (дата звернення: 29.09.2023).

23. Правила і рекомендації YouTube [Електронний ресурс]. URL: https://www.youtube.com/intl/ru_ALL/creators/how-things-work/policies-guidelines/ (дата звернення: 01.10.2023).

24. Recommendation system - Netflix Research [Електронний ресурс]. URL: <https://research.netflix.com/research-area/recommendations> (дата звернення: 29.09.2023).

25. 2023 TikTok Algorithm Explained + Tips to Go Viral [Електронний ресурс]. URL: <https://blog.hootsuite.com/tiktok-algorithm/> (дата звернення: 29.09.2023).

26. Amazon Personalize - Система рекомендацій [Електронний ресурс]. URL: <https://aws.amazon.com/ru/personalize/> (дата звернення: 29.09.2023).

27. Methods to inform the development of concise objectives hierarchies in multicriteria decision analysis [Електронний ресурс]. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0377221719301870> (дата звернення: 06.10.2023).

28. Літнарівч Р.М. Побудова і дослідження математичної моделі за джерелами експериментальних даних методами регресійного аналізу: навч. посіб., Рівне: МЕРУ, 2011. 140 с.

29. G. O. Odu and O. E. Charles-Owaba. Review of Multi-criteria Optimization Methods – Theory and Applications – IOSR Journal of Engineering (IOSRJEN). October 2013.

30. Авраменко В. С., Авраменко А. С. Проектування інформаційних систем: навч. посіб. Черкаси: ЧНУ ім. Б. Хмельницького, 2017. 434 с.

31. Лосєв М. Ю., Федько В. В. Бази даних : навч.-практ. посіб. Харків : ХНЕУ ім. С. Кузнеця, 2018. 233 с.
32. Булатецька Л. В., Булатецький В. В. Мова запитів SQL: конспект лекцій. Луцьк: СНУ ім. Лесі Українки, 2018. 92 с.
33. Коноваленко І.В., Марущак П.О., Савків В.Б. Програмування мовою С# 7.0 : навч. посіб. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 300 с.
34. Галкін О.В., Катеринич Л.О., Шкільняк О.С. Програмування на Java 8: навч. посіб. Київ: ЛОГОС, 2017. 186 с.
35. Цеслів О. В. Розробка додатків в середовищі Java: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2023. 169 с.
36. Project Lombok. [Електронний ресурс]. URL: <https://projectlombok.org/> (дата звернення: 29.09.2023).
37. Spring Framework. [Електронний ресурс]. URL: <https://spring.io/projects/spring-framework> (дата звернення: 29.09.2023).
38. Руденко В.Д. Жугастров О.О. Перші кроки програмування мовою Java: навч. посіб. Харків: Ранок, 2018. 112 с.
39. RankLib [Електронний ресурс]. URL: <https://github.com/codelibs/ranklib> (дата звернення: 29.09.2023).
40. Guide to Using ModelMapper [Електронний ресурс]. URL: <https://www.baeldung.com/java-modelmapper> (дата звернення: 29.09.2023).
41. jOOQ is the best way to write SQL in Java [Електронний ресурс]. URL: <https://github.com/jOOQ/jOOQ> (дата звернення: 29.09.2023).
42. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
43. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія ранжування відео на стрімінговій платформі

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

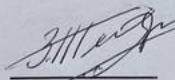
Оригінальність 95,46% Схожість 4,54%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

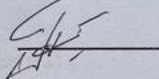
Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Замкова Т.Д.

Керівник роботи

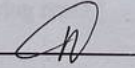


Петришин С.І.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)**Лістинг програми****Клас ModelMapperConfig**

```

package com.zamkovyi.config;

import org.modelmapper.ModelMapper;
import org.springframework.context.annotation.Bean;
import org.springframework.stereotype.Controller;

@Controller
public class ModelMapperConfig {
    @Bean
    public ModelMapper modelMapper() {
        return new ModelMapper();
    }
}

```

Клас VideoDao

```

package com.zamkovyi.dao;

import com.zamkovyi.jooq.generated.tables.records.VideosRecord;
import lombok.RequiredArgsConstructor;
import org.jooq.DSLContext;
import org.springframework.stereotype.Repository;

import java.util.List;

import static com.zamkovyi.jooq.generated.Tables.VIDEOS;

@Repository
@RequiredArgsConstructor
public class VideoDao {

    private final DSLContext context;

    public void save(VideosRecord videosRecord) {
        context.insertInto(VIDEOS)
            .set(videosRecord);
    }

    public List<VideosRecord> getByType(String type) {
        return context.selectFrom(VIDEOS)

```

```

        .where(VIDEOS.TOPIC.eq(type))
        .fetch();
    }
}

```

Клас Video

```

package com.zamkovyi.entity;

import lombok.Data;

@Data
public class Video {
    private int id;
    private String name;
    private int duration;
    private String topic;
    private int views;
    private int viewsForFirstDay;
    private int averageViews;
    private int persentMoreWhenFalf;
    private int persentMore;
}

```

Клас Videos

```

/*
 * This file is generated by jOOQ.
 */
package com.zamkovyi.jooq.generated.tables;

import com.zamkovyi.jooq.generated.DefaultSchema;
import com.zamkovyi.jooq.generated.Keys;
import com.zamkovyi.jooq.generated.tables.records.VideosRecord;

import java.util.Arrays;
import java.util.List;

import javax.annotation.Generated;

import org.jooq.Field;
import org.jooq.ForeignKey;
import org.jooq.Identity;
import org.jooq.Name;

```

```

import org.jooq.Record;
import org.jooq.Schema;
import org.jooq.Table;
import org.jooq.TableField;
import org.jooq.UniqueKey;
import org.jooq.impl.DSL;
import org.jooq.impl.TableImpl;

/**
 * This class is generated by jOOQ.
 */
@Generated(
    value = {
        "http://www.jooq.org",
        "jOOQ version:3.11.3"
    },
    comments = "This class is generated by jOOQ"
)
@SuppressWarnings({ "all", "unchecked", "rawtypes" })
public class Videos extends TableImpl<VideosRecord> {

    private static final long serialVersionUID = -656705460;

    /**
     * The reference instance of <code>videos</code>
     */
    public static final Videos VIDEOS = new Videos();

    /**
     * The class holding records for this type
     */
    @Override
    public Class<VideosRecord> getRecordType() {
        return VideosRecord.class;
    }

    /**
     * The column <code>videos.id</code>.
     */
    public final TableField<VideosRecord, Integer> ID =
        org.jooq.impl.SQLDataType.INTEGER.nullable(false).identity(true), this, "");

    /**
     * The column <code>videos.name</code>.
     */

```

```

public final TableField<VideosRecord, String> NAME = createField("name",
org.jooq.impl.SQLDataType.VARCHAR(90).nullable(false), this, "");

/**
 * The column <code>videos.duration</code>.
 */
public final TableField<VideosRecord, Integer> DURATION = createField("duration",
org.jooq.impl.SQLDataType.INTEGER, this, "");

/**
 * The column <code>videos.topic</code>.
 */
public final TableField<VideosRecord, String> TOPIC = createField("topic",
org.jooq.impl.SQLDataType.VARCHAR(45), this, "");

/**
 * The column <code>videos.views</code>.
 */
public final TableField<VideosRecord, Integer> VIEWS = createField("views",
org.jooq.impl.SQLDataType.INTEGER, this, "");

/**
 * The column <code>videos.viewsForFirstDay</code>.
 */
public final TableField<VideosRecord, Integer> VIEWSFORFIRSTDAY =
createField("viewsForFirstDay", org.jooq.impl.SQLDataType.INTEGER, this, "");

/**
 * The column <code>videos.averageViews</code>.
 */
public final TableField<VideosRecord, Integer> AVERAGEVIEWS =
createField("averageViews", org.jooq.impl.SQLDataType.INTEGER, this, "");

/**
 * The column <code>videos.persentMoreWhenFalf</code>.
 */
public final TableField<VideosRecord, Integer> PERSENTMOREWHENFALF =
createField("persentMoreWhenFalf", org.jooq.impl.SQLDataType.INTEGER, this, "");

/**
 * The column <code>videos.persentMore</code>.
 */
public final TableField<VideosRecord, String> PERSENTMORE = createField("persentMore",
org.jooq.impl.SQLDataType.VARCHAR(45), this, "");

/**

```

```

* Create a <code>videos</code> table reference
*/
public Videos() {
    this(DSL.name("videos"), null);
}

/**
* Create an aliased <code>videos</code> table reference
*/
public Videos(String alias) {
    this(DSL.name(alias), VIDEOS);
}

/**
* Create an aliased <code>videos</code> table reference
*/
public Videos(Name alias) {
    this(alias, VIDEOS);
}

private Videos(Name alias, Table<VideosRecord> aliased) {
    this(alias, aliased, null);
}

private Videos(Name alias, Table<VideosRecord> aliased, Field<?>[] parameters) {
    super(alias, null, aliased, parameters, DSL.comment(""));
}

public <O extends Record> Videos(Table<O> child, ForeignKey<O, VideosRecord> key) {
    super(child, key, VIDEOS);
}

/**
* {@inheritDoc}
*/
@Override
public Schema getSchema() {
    return DefaultSchema.DEFAULT_SCHEMA;
}

/**
* {@inheritDoc}
*/
@Override
public Identity<VideosRecord, Integer> getIdentity() {
    return Keys.IDENTITY_VIDEOS;
}

```



```

}

/**
 * {@inheritDoc}
 */
@Override
public UniqueKey<VideosRecord> getPrimaryKey() {
    return Keys.KEY_VIDEOS_PRIMARY;
}

/**
 * {@inheritDoc}
 */
@Override
public List<UniqueKey<VideosRecord>> getKeys() {
    return Arrays.<UniqueKey<VideosRecord>>asList(Keys.KEY_VIDEOS_PRIMARY,
Keys.KEY_VIDEOS_NAME_UNIQUE);
}

/**
 * {@inheritDoc}
 */
@Override
public Videos as(String alias) {
    return new Videos(DSL.name(alias), this);
}

/**
 * {@inheritDoc}
 */
@Override
public Videos as(Name alias) {
    return new Videos(alias, this);
}

/**
 * Rename this table
 */
@Override
public Videos rename(String name) {
    return new Videos(DSL.name(name), null);
}

/**
 * Rename this table
 */

```

```

@Override
public Videos rename(Name name) {
    return new Videos(name, null);
}
}

```

Класс VideoParser

```

package com.zamkovyi.parser;

import com.zamkovyi.entity.Video;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;

import java.util.List;
import java.util.stream.Collectors;
import java.util.stream.IntStream;

@Component
@RequiredArgsConstructor
public class VideoParser {

    public List<Video> parse(XSSFSSheet sheet) {
        return IntStream.range(1, sheet.getLastRowNum()).boxed()
            .map(i -> parse(sheet.getRow(i)))
            .collect(Collectors.toList());
    }

    private Video parse(XSSFRow row) {
        Video video = new Video();
        video.setName(row.getCell(0).getRawValue());
        video.setDuration(Integer.parseInt(row.getCell(1).getRawValue()));
        video.setTopic(row.getCell(2).getRawValue());
        video.setViews(Integer.parseInt(row.getCell(3).getRawValue()));
        video.setViewsForFirstDay(Integer.parseInt(row.getCell(4).getRawValue()));
        video.setAverageViews(Integer.parseInt(row.getCell(5).getRawValue()));
        video.setPersetMoreWhenFalf(Integer.parseInt(row.getCell(6).getRawValue()));
        video.setPersetMore(Integer.parseInt(row.getCell(7).getRawValue()));
        return video;
    }
}

```

Класс RankingVideo

```

package com.zamkovyi.ranking;

import com.zamkovyi.service.GuiService;

```

```

import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@ComponentScan
@Configuration
public class RankingVideo {

    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(RankingVideo.class);

        GuiService service = context.getBean(GuiService.class);

        service.createMainPage();
    }

}

```

Клас VideoService

```

package com.zamkovyi.service;

import com.zamkovyi.dao.VideoDao;
import com.zamkovyi.entity.Video;
import com.zamkovyi.jooq.generated.tables.records.VideosRecord;
import com.zamkovyi.parser.VideoParser;
import lombok.RequiredArgsConstructor;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.modelmapper.ModelMapper;
import org.springframework.stereotype.Service;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

@Service
@RequiredArgsConstructor
public class VideoService {

```

```

private final VideoParser videoParser;
private final VideoDao videoDao;
private final ModelMapper modelMapper;

public void processFile(File file) {
    try {
        InputStream targetStream = new FileInputStream(file);
        org.apache.poi.xssf.usermodel.XSSFWorkbook workbook = new
XSSFWorkbook(targetStream);
        XSSFSheet sheetAt = workbook.getSheetAt(0);
        List<Video> videos = videoParser.parse(sheetAt);
        videos.stream()
            .map(f -> modelMapper.map(f, VideosRecord.class))
            .forEach(videoDao::save);
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public List<Video> getRatingVideos(String type) {
    List<VideosRecord> videosRecords = videoDao.getByType(type);
    List<Video> videos = videosRecords.stream()
        .map(v -> modelMapper.map(v, Video.class))
        .collect(Collectors.toList());
    Map<Integer, Video> datingToVideoMap = countVideoRating(videos);
    List<Integer> keys = datingToVideoMap.keySet()
        .stream()
        .sorted(Comparator.comparingInt(Integer::intValue))
        .collect(Collectors.toList());
    return keys.stream()
        .map(datingToVideoMap::get)
        .collect(Collectors.toList());
}

private Map<Integer, Video> countVideoRating(List<Video> videos) {
    return videos.stream()
        .collect(Collectors.toMap(video -> countRate(video),
            video -> video));
}

private Integer countRate(Video video) {
    int r1 = video.getViewsForFirstDay() * 1200 / video.getAverageViews();
    int r2 = video.getViews() * 600 / video.getAverageViews();
    int r3 = video.getPersentMoreWhenFalf() * 2;
    int r4 = video.getPersentMore() * 4;
}

```

```

    return r1 + r2 + r3 + r4;
}
}

```

Клас RankingTable

```

package com.zamkovyi.service;

import com.zamkovyi.entity.Video;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

@Component
@RequiredArgsConstructor
public class RankingTable {

    private final GuiService guiService;
    public static JFrame frame;
    private JLabel label;

    public void createWindow(java.util.List<Video> videoList) {
        frame = new JFrame();
        frame.setTitle("Прогноз");
        frame.setSize(600, 600);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);
        Font font = new Font("TimesRoman", Font.BOLD, 14);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        frame.add(label);
        frame.setVisible(true);

        RankingTableModal tmq2 = new RankingTableModal();
        JTable table = new JTable(tmq2);

        JScrollPane tableScrollPane = new JScrollPane(table);
        tmq2.addData(videoList);
        int shur = 0;
        if ((19 + tmq2.getRowCount() * 24) > 500) {

```

```

        shur = 500;
    } else {
        shur = 19 + tmq2.getRowCount() * 24;
    }
    tableScrollPane.setPreferredSize(new Dimension(580, shur));
    tableScrollPane.setBackground(Color.BLUE);
    table.setFont(font);
    table.setRowHeight(24);

    label.add(tableScrollPane, new GridBagConstraints(0, 0, 4, 1, 1, 1,
GridBagConstraints.NORTH,
        GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));

    JButton backBtn = new JButton();
    backBtn.setFont(font);
    backBtn.setText("Назад");
    label.add(backBtn, new GridBagConstraints(0, 0, 4, 1, 1, 1, GridBagConstraints.SOUTH,
        GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));

    backBtn.addActionListener(new BackListener());
}

public void closeWindow() {
    frame.setVisible(false);
    frame.remove(label);
}

private class BackListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent JCom) {
        closeWindow();
        guiService.createMainPage();
    }
}
}

```

Клас GuiService

```

package com.zamkovyi.service;

import com.zamkovyi.entity.Video;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

@Service
@RequiredArgsConstructor
public class GuiService {

    private final VideoService videoService;

    private JFrame frame;
    private JLabel label;
    private JComboBox comboBox;

    public void createMainPage() {
        frame = new JFrame();
        frame.setTitle("Ранжування відео");
        frame.setSize(400, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 11);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);

        JButton hideBtn = new JButton();
        hideBtn.setText("Додати відео");
        hideBtn.setFont(font1);
        label.add(hideBtn, new GridBagConstraints(0, 0, 2, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(10, 0, 10, 0), 0, 0));

        String[] arr = new String[]{"", "Автомобілі", "Відеоігри", "Історія", "Наука", "Політика",
            "Спорт", "Технології"};

        comboBox = new JComboBox(arr);
        comboBox.setFont(font1);
        label.add(comboBox, new GridBagConstraints(0, 0, 2, 1, 0.0, 0.9,
            GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

        JButton hide = new JButton();

```

```

hide.setText("Переглянути рейтинг відео");
hide.setFont(font1);
label.add(hide, new GridBagConstraints(0, 2, 2, 1, 0.0, 0.9, GridBagConstraints.CENTER,
    GridBagConstraints.HORIZONTAL, new Insets(10, 0, 10, 0), 0, 0));

hideBtn.addActionListener(new SelectListener());
hide.addActionListener(new HideListener());

frame.add(label);
frame.setVisible(true);
}

public void closeWindow() {
    frame.setVisible(false);
    frame.remove(label);
}

private class SelectListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent JCom) {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.showDialog(label, "Open");
        File file =fileChooser.getSelectedFile();
        videoService.processFile(file);
    }
}

private class HideListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent JCom) {
        closeWindow();
        String type = (String) comboBox.getSelectedItem();
        java.util.List<Video> videoRating = videoService.getRatingVideos(type);
        //      new HideText(f1, f2).hiding();
//      new MainGUI().createWindow();
    }
}
}

```


Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РАНЖУВАННЯ ВІДЕО НА
СТРІМІНГОВІЙ ПЛАТФОРМІ»

Виконала: студентка 2-го курсу, групи
ЗКН-22м
спеціальності 122 – «Комп'ютерні
науки»



Замкова Т. Д.

(прізвище та ініціали)

Керівник: д.т.н., ст. викл. каф. КН



Петришин С. І.

(прізвище та ініціали)

« 04 » _____ 12 _____ 2023 р.

Вінниця ВНТУ - 2023 рік



Рисунок В.1 – Структурна схема інформаційної технології

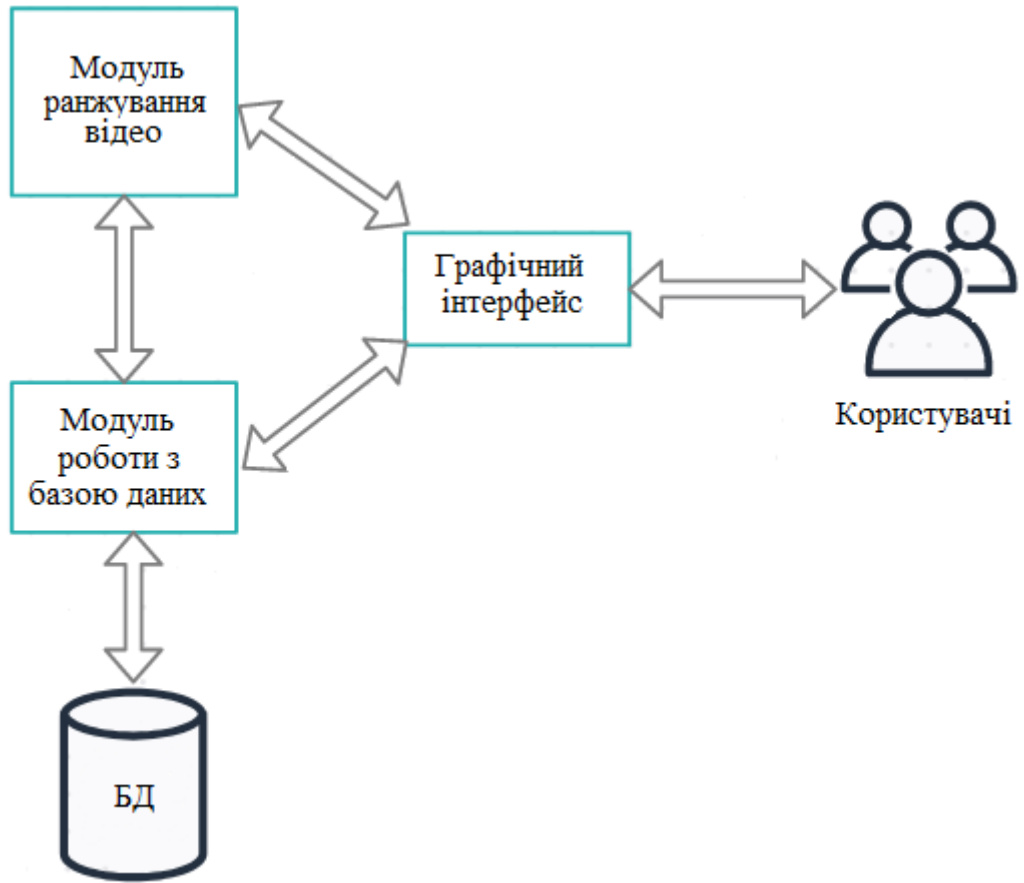


Рисунок В.2 – Загальна структурна схема

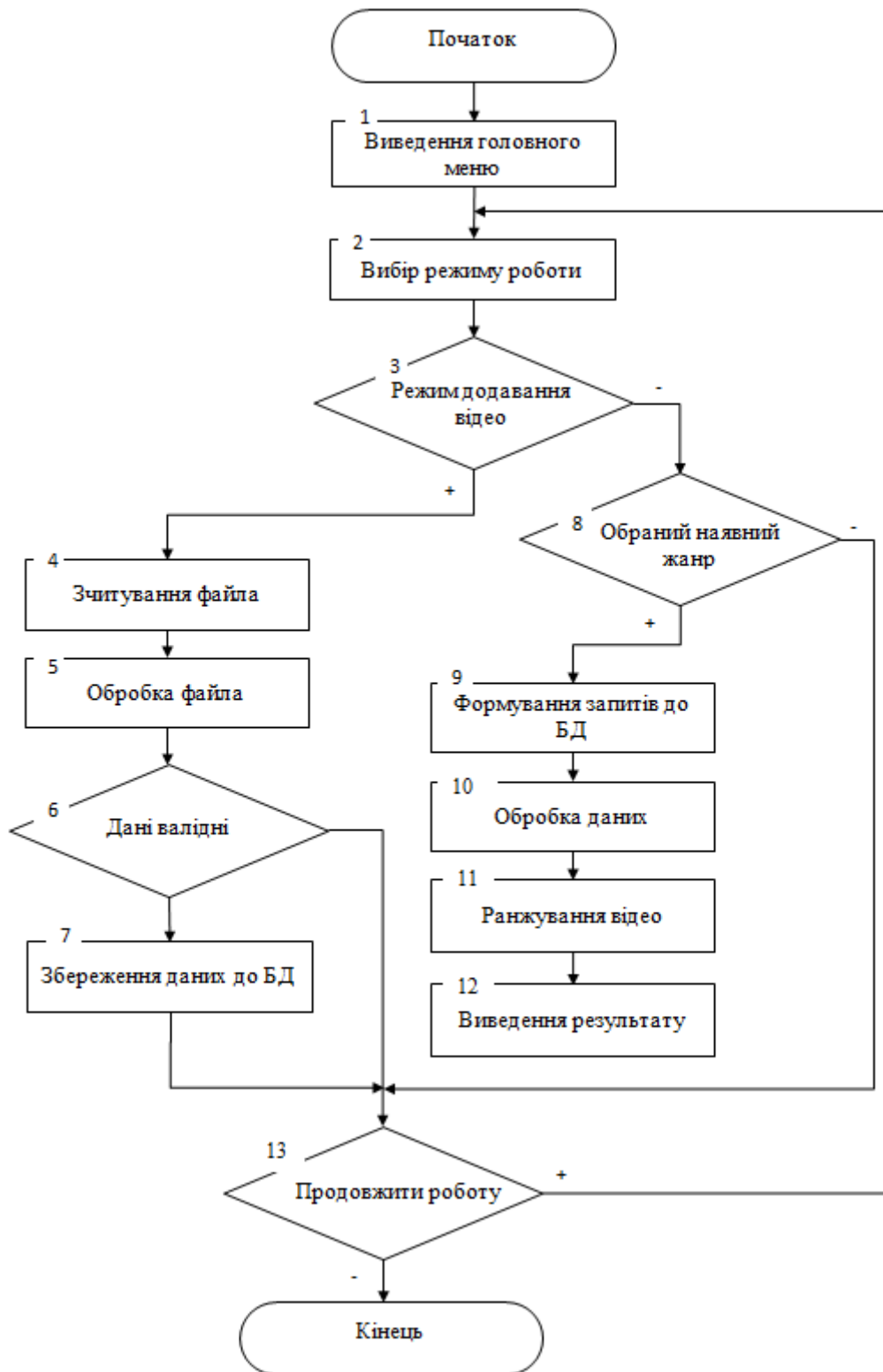


Рисунок В.3 – Схема алгоритму модуля ранжування інформаційної технології ранжування відео на стрімінговій платформі

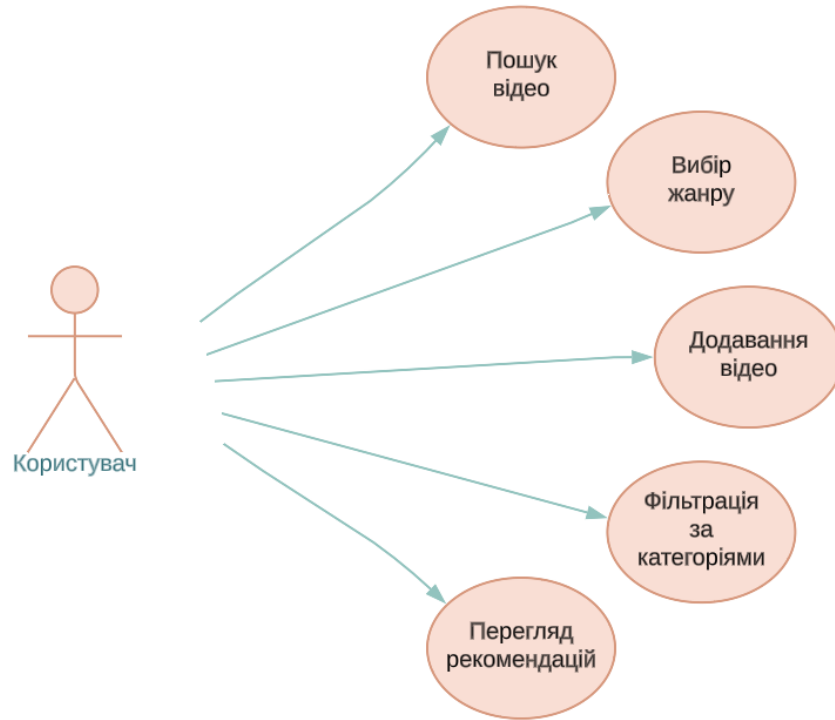


Рисунок В.4 – Діаграма прецедентів

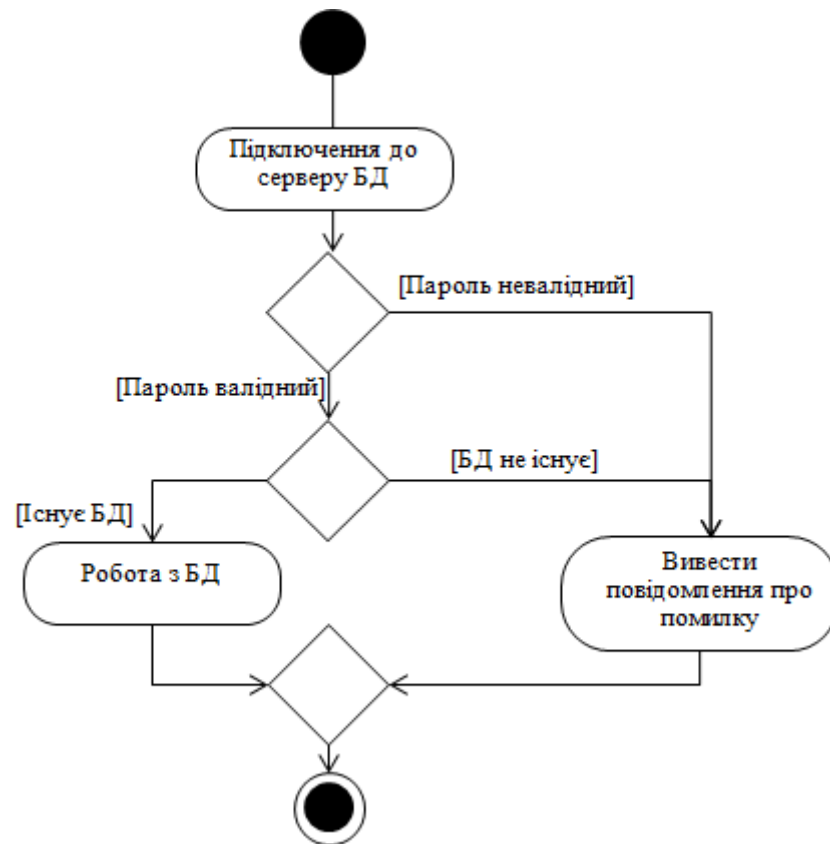


Рисунок В.5 – Діаграма діяльності системи при встановленні з'єднання з базою даних

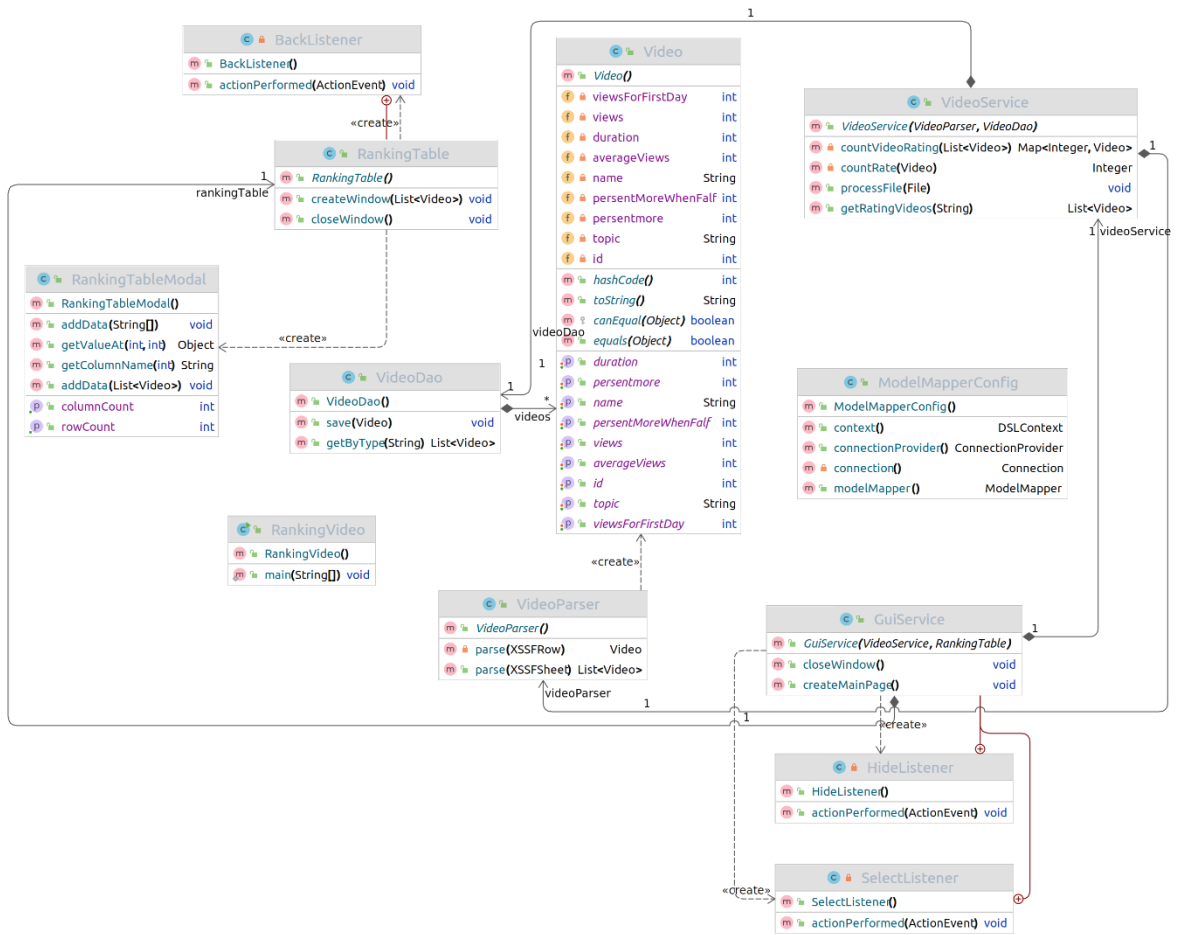


Рисунок В.6 – UML-діаграма класів

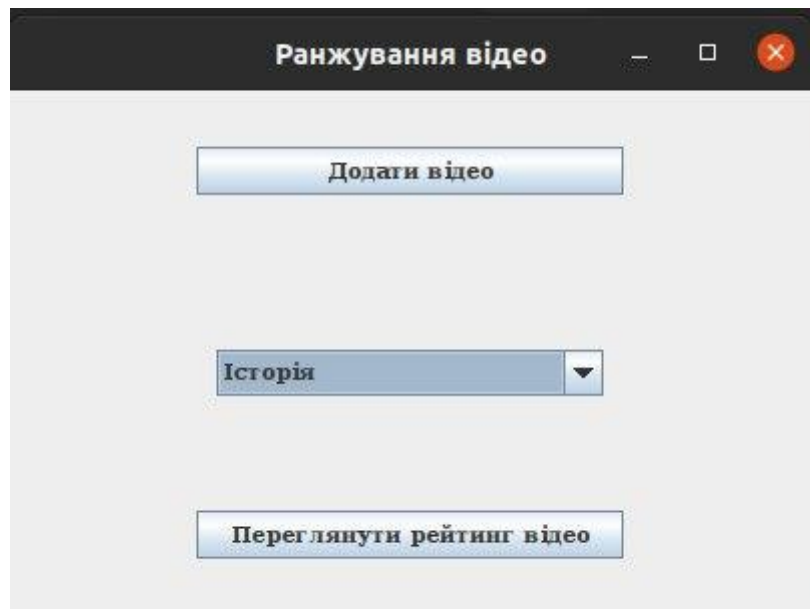
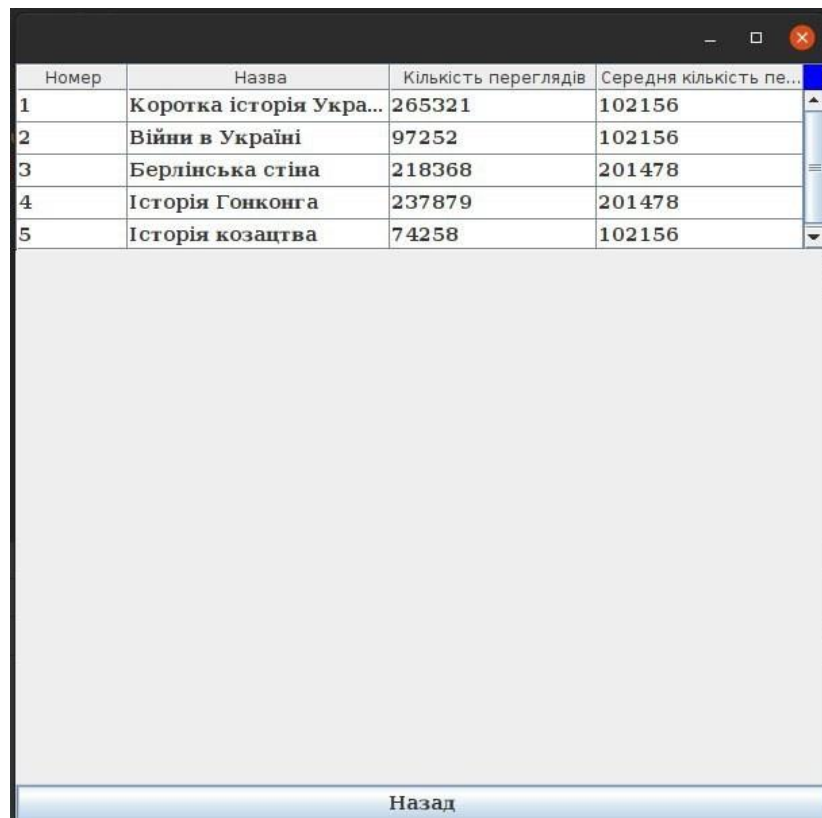


Рисунок В.7 – Приклад роботи програми



Номер	Назва	Кількість переглядів	Середня кількість пе...
1	Коротка історія Укра...	265321	102156
2	Війни в Україні	97252	102156
3	Берлінська стіна	218368	201478
4	Історія Гонконга	237879	201478
5	Історія козацтва	74258	102156

Назад

Рисунок В.8 – Результат роботи програми

Додаток Г (довідниковий)

Інструкція користувача

При запуску програми спочатку відкривається вікно початкової активності, яке включає кнопку «Додати відео», поле із спливаючим меню для введення категорії відео, за якою відбувається ранжування, та кнопку «Переглянути рейтинг відео». Воно є головним меню програми, де користувач може обрати необхідний пункт для подальшої роботи (рисунок Г.1). Необхідно обрати пункт, який цікавить користувача для переходу до подальшої роботи.

Якщо користувач вибрав опцію «Додати відео» відкривається діалогове вікно для завантаження файлу, що містить дані про відео (рисунок Г.2).

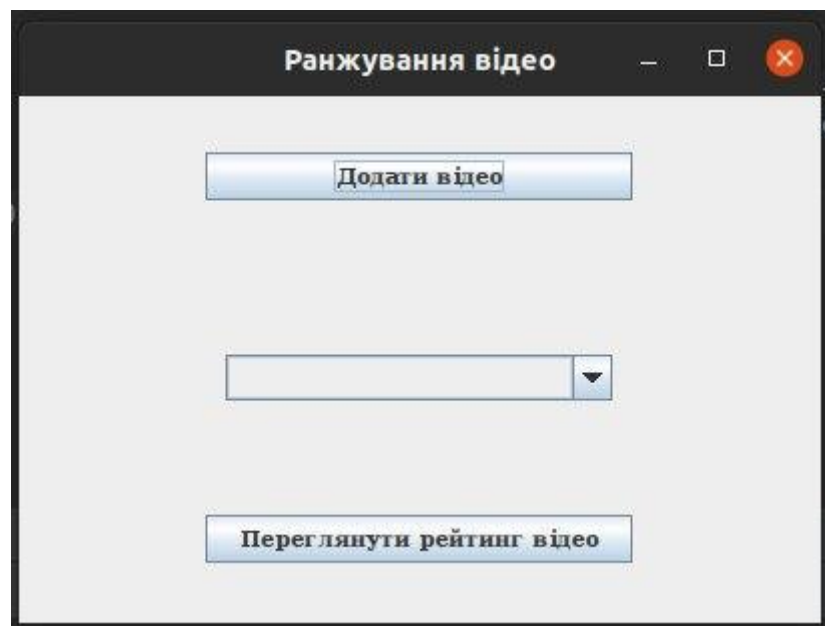


Рисунок Г.1 – Головне вікно програми

Діалогове вікно відображає наявні файли, вибравши один з яких необхідно натиснути кнопку «Open». Як видно з рисунка Г.2, користувач має вибирати файл формату xlsx з назвою «newVideo», в якому міститься таблиця з параметрами відео. Програма зчитує файл та обробляє дані, що містяться в ньому. Якщо дані валідні – відбувається їх збереження до бази даних.

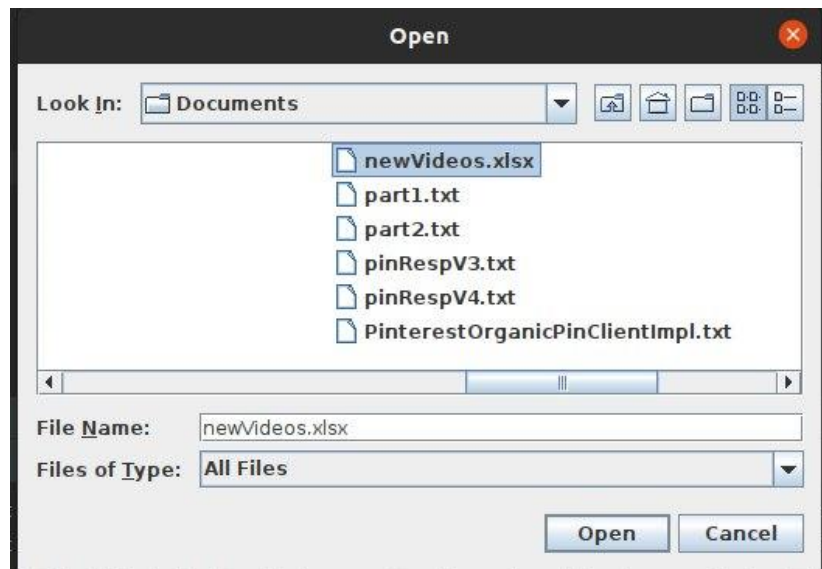


Рисунок Г.2 – Вікно завантаження файла

Для того, щоб проранжувати відео необхідно обрати категорію, за якою буде проводитися ранжування. Користувач, наприклад, обирає категорію «Історія» (рисунок Г.3), після чого натискає кнопку «Переглянути рейтинг відео». Результатом ранжування є таблиця з проранжованими відео (рисунок Г.4).

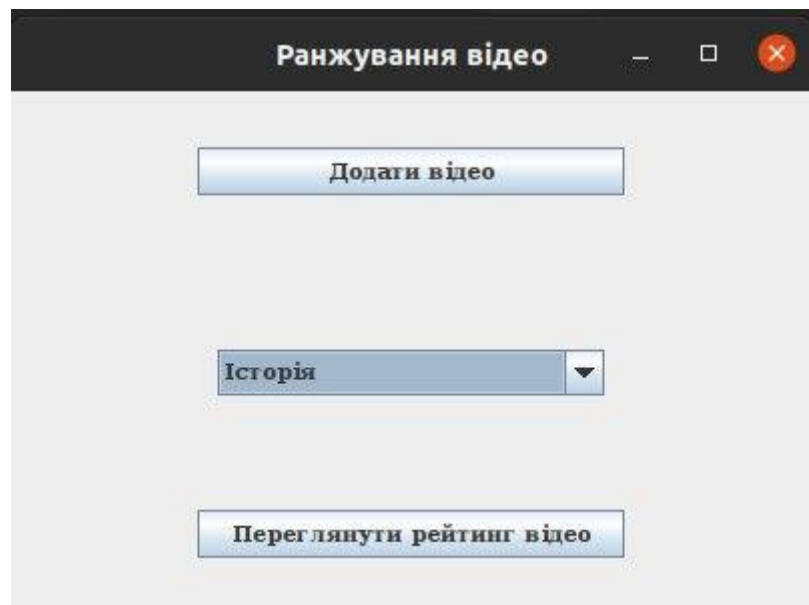
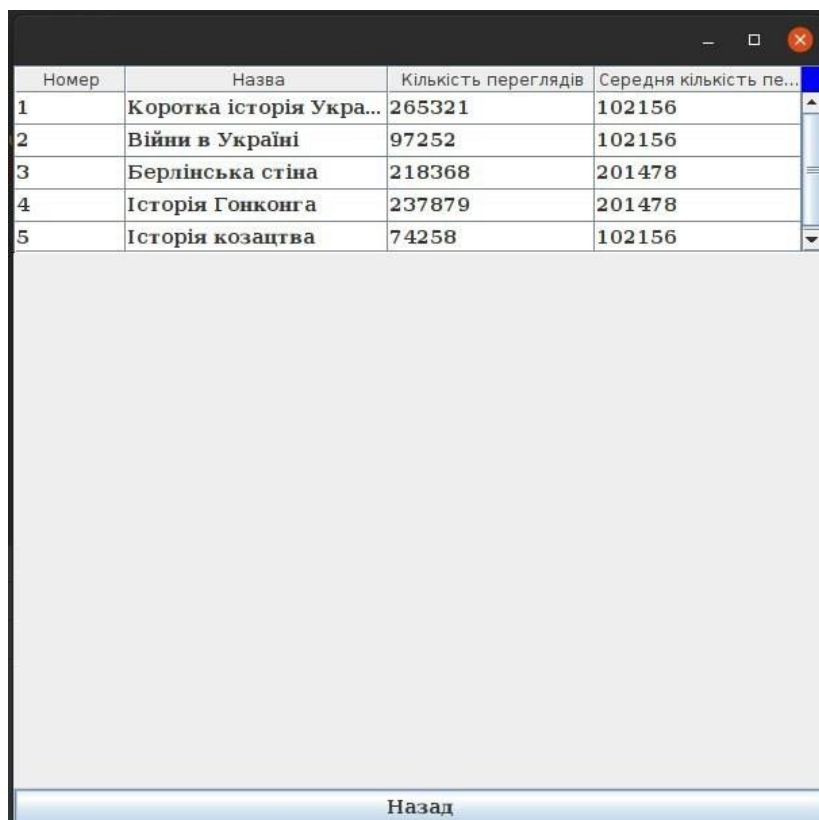


Рисунок Г.3 – Вікно вибору категорії



Номер	Назва	Кількість переглядів	Середня кількість пе...
1	Коротка історія Укра...	265321	102156
2	Війни в Україні	97252	102156
3	Берлінська стіна	218368	201478
4	Історія Гонконга	237879	201478
5	Історія козацтва	74258	102156

Назад

Рисунок Г.4 – Вікно з результатом ранжування

При виборі користувачем категорії «Спорт» з головного меню програми виводиться рейтинг відео за заданою категорією (рис. Г.5 – Г.6).

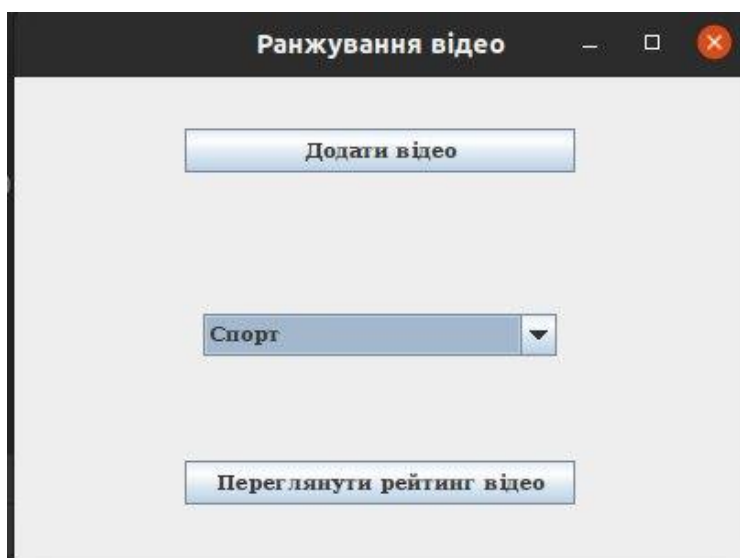
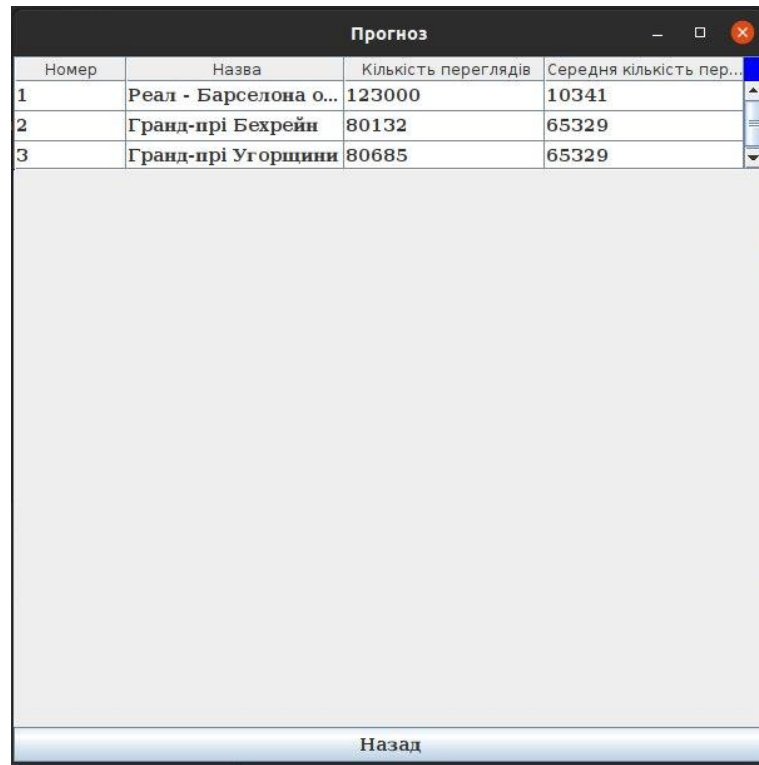


Рисунок Г.5 – Вікно вибору категорії



Номер	Назва	Кількість переглядів	Середня кількість пер...
1	Реал - Барселона о...	123000	10341
2	Гранд-прі Бехрейн	80132	65329
3	Гранд-прі Угорщини	80685	65329

Назад

Рисунок Г.6 – Вікно з результатом ранжування