

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Системотехніки  
(повна назва)

**АТЕСТАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

«Дослідження особливостей роботи рекомендаційних систем при застосуванні  
штучного інтелекту»  
(тема)

Виконав:

Студент 2 курсу, групи СПРМ-19-1

Спеціальність 122 – Комп'ютерні науки  
(код і повна назва напрямку)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування  
(повна назва освітньої програми)

Лобинцев А. А.

(прізвище, ініціали)

Керівник проф. Перова І. Г.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри \_\_\_\_\_  
(підпис)

Гребеннік І. В.  
(прізвище, ініціали)

2020 р.

## Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Системотехніки  
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – Комп'ютерні науки  
(код і повна назва)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Лобинцеву Андрію Андрійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження особливостей роботи рекомендаційних систем при застосуванні штучного інтелекту»

затверджена наказом по університету від « 02 » 11 2020 р. № 1516Ст

2. Термін подання студентом роботи (проекту) 18.12.2020 р.

3. Вихідні дані до роботи (проекту) теоретичні відомості про аналіз даних у предметній області рекомендує систем, штучного інтелекту та ігрової індустрії

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити) \_\_\_\_\_

1. Аналіз предметної області

2. Аналіз існуючих методів добутку знань

3. Аналіз існуючих алгоритмів

4. Формування вимог до системи

5. Проектування системної реалізації

6. Реалізація прототипу системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів)

5.1 Взаємодія сервісів в системі (1 аркуш формату А4). 5.2 Контекстна діаграма системи (1 аркуш формату А4). 5.3 Діаграма декомпозиції контекстної діаграми (1 аркуш формату А4). 5.4 Концептуальна DFD системи (1 аркуш формату А4). 5.5 Діаграма декомпозиції процесу «Робота системи надавання рекомендацій» (1 аркуш формату А4). 5.6 Діаграма варіантів використання системи (1 аркуш формату А4). 5.7 Діаграма класів (1 аркуш формату А4). 5.8 Діаграма послідовності дій для надавання рекомендацій (1 аркуш формату А4).

6. Консультанти розділів роботи (проєкту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	02.11.20	
2	Аналіз завдання, підбір літератури	02.11.20-05.11.20	
3	Аналіз літератури з досліджуваної проблеми	05.11.20-08.11.20	
4	Аналіз технічних засобів для реалізації	08.11.20-10.11.20	
5	Проектування системи	10.11.20-15.11.20	
6	Програмна реалізація	15.11.20-30.11.20	
7	Оформлення пояснювальної записки	30.11.20-08.12.20	
8	Перевірка на плагіат	09.12.20	
9	Представлення на рецензування	10.12.20	
10	Представлення атестаційної роботи в ДЕК	18.12.20	

Дата видачі завдання   02     11   2020 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Перова І. Г.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Атестаційна робота: 55 с., 17 рис., 5 табл., 1 формула, 21 джерело, 1 додаток.

### РЕКОМЕНДАЦІЙНА СИСТЕМА, ШТУЧНИЙ ІНТЕЛЕКТ, ВІДЕОІГРОВА ІНДУСТРІЯ, АНАЛІЗ, РОЗРОБКА, ПРОГРАМУВАННЯ

Метою дослідження є отримання найбільш достовірних та точних рекомендацій за допомогою розробленої системи.

Задачею дослідження є проектування системи, яка за допомогою обробки різних вхідних даних під час різних ігрових ситуацій буде надавати рекомендації музичного супроводження.

У результаті роботи спроектована система надання рекомендацій музичного супроводження в різних станах користувача та реалізований її прототип.

## **ABSTRACT**

Attestation work: 55 p., 17 pic., 5 table, 21 source, 1 formula, 1 application.

**RECOMMENDATION SYSTEM, ARTIFICIAL INTELLIGENCE, VIDEO GAME INDUSTRY, ANALYSIS, DEVELOPMENT, PROGRAMMING**

The goal of research is to get the most reliable and accurate recommendations using the developed system.

The task of research is focused on designing a system a system that, by processing different input data during different game situations, will provide recommendations for musical accompaniment.

As a result of work, the system of providing musical accompaniment recommendations in different user states was designed and its prototype was implemented.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ .....	6
ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1 Опис предметної області .....	9
1.1.1 Опис рекомендуємих систем.....	9
1.1.2 Опис відеоігрової індустрії .....	10
1.1.3 Опис штучного інтелекту .....	11
1.2 Опис сучасних проблем .....	13
1.3 Аналіз існуючих методів добутку знань .....	14
1.4 Аналіз існуючих алгоритмів.....	17
1.5 Постановка задачі .....	22
2 РОЗРОБКА МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ .....	23
2.1 Розробка вимог до системи .....	23
2.2 Визначення функціональних вимог до системи.....	24
2.3 Розробка моделі потоків даних системи .....	27
2.4 Діаграма варіантів використання.....	29
2.5 Діаграма класів системи .....	30
2.6 Діаграма послідовності дій.....	34
3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ .....	37
3.1 Обґрунтування вибору мови програмування .....	37
3.2 Обґрунтування вибору СУБД .....	37
3.3 Моделювання бази даних системи .....	38
3.4 Проведення досліджень .....	41
ВИСНОВКИ.....	44
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	45
Додаток А .....	47
Додаток Б.....	56

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

БД – база даних;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект;

AI (Artificial Intelligence) – штучний інтелект;

ANI (Artificial Narrow Intelligence) - штучний інтелект вузького спектру;

AGI (Artificial General Intelligence) - сильний штучний інтелект;

ASI (Artificial Super Intelligence) - штучний супер інтелект;

CASE (Computer Aided Software Engineering) – сукупність методів і засобів автоматизованого проектування інформаційних систем;

MRS (Music Recommendation System) - рекомендаційна система музики;

CI (Confidence Intervals) - інтервали довіри;

ML (Machine Learning) - машинне навчання;

DL (Deep Learning) - глибоке навчання;

NL (Neural Network) - нейронні мережі;

CC (Cognitive Computing) - когнітивне обчислення;

NLP (Natural Language Processing) - обробка природної мови;

CV (Computer Vision) - комп'ютерний зір;

DFD (Data Flow Diagram) – діаграма потоків даних;

CRUD (Create, Read, Update, Delete) – акронім, що позначає чотири базові функції, які використовуються при роботі з базами даних;

## ВСТУП

Із зростанням таких сервісів, як Youtube, Amazon, Netflix, та багатьох інших веб-сервісів, рекомендаційні системи займають дедалі більше місця в нашому житті. Від електронної комерції (пропонування покупцям статті, які могли б їх зацікавити) до реклами в Інтернеті (пропонування користувачам правильний вміст, що відповідає їхнім уподобанням), рекомендаційні системи сьогодні неминучі в нашому щоденному житті з використанням Інтернету.

Загалом, рекомендаційні системи, - це алгоритми, спрямовані на те, щоб пропонувати користувачам відповідні предмети (фільми, відео, серіали, музику, статті, товари або щось інше, залежно від галузей), які можуть їх заінтересувати [1].

Рекомендаційні системи, дійсно є критично важливими в деяких галузях, оскільки вони можуть приносити величезний дохід, якщо вони являються ефективними, або також можуть бути способом виокремити конкурентів.

Швидке зростання кількості доступної цифрової інформації та кількості відвідувачів Інтернету створило потенційну проблему перевантаження інформації, яка перешкоджає своєчасному доступу до предметів, що цікавлять кожного користувача. Системи пошуку інформації, такі як Google, Yahoo, Yandex частково вирішили цю проблему, але пріоритети та персоналізація (де система відображає наявний вміст за інтересами та уподобаннями користувача) інформації відсутні. Це збільшило попит на рекомендовані системи як ніколи раніше.

Рекомендаційні системи являються і системами фільтрації інформації, які вирішують проблему перевантаження інформації, фільтруючи життєво важливий фрагмент інформації з великої кількості динамічно генерованої інформації відповідно до уподобань користувача, інтересу або спостережуваної поведінки щодо товару. Рекомендаційна система має можливість передбачити, чи віддасть певний користувач перевагу елементу чи ні, на основі профілю користувача.

Рекомендовані системи вигідні як постачальникам послуг, так і користувачам. Вони зменшують час на пошук та відбір предметів в середовищі інтернет-покупок. Рекомендаційні системи також довели, що покращують процес та якість прийняття рішень.

Для того, щоб надати клієнтам рекомендації щодо послуг чи продуктів, механізми рекомендацій використовують алгоритми. Останнім часом ці двигуни почали використовувати алгоритми машинного навчання, що робить процес

прогнозування елементів більш точним. На основі даних, отриманих із рекомендаційних систем, алгоритми змінюються. Користувачі зазвичай оцінюють лише невелику частину товарів, що є в каталозі, і завдання рекомендаційної системи - узагальнити цю інформацію і передбачити ставлення клієнта до інших товарів, про які нічого не відомо.

Тому метою атестаційної роботи є дослідити особливості роботи рекомендаційних систем при застосуванні штучного інтелекту.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Опис предметної області

#### 1.1.1 Опис рекомендуючих систем

Рекомендуючі системи підтримують користувачів у процесі пошуку та вибору продуктів (предметів) із заданого асортименту. Прикладами таких предметів є фільми, книги, пісні, фінансові послуги, квартири, цифрові камери тощо.

Система рекомендацій визначається як стратегія прийняття рішень для користувачів у складних інформаційних середовищах. Крім того, система рекомендацій була визначена з точки зору електронної комерції як інструмент, який допомагає користувачам здійснювати пошук у записах знань, які пов'язані з інтересами та уподобаннями користувачів. Рекомендаційні системи вирішують проблему перевантаження інформації, з якою зазвичай стикаються користувачі, надаючи їм персоналізований, ексклюзивні рекомендації щодо вмісту та послуг. Нещодавно були розроблені різні підходи до побудови систем рекомендацій, які можуть використовувати колаборативну фільтрацію, фільтрацію на основі вмісту або гібридну фільтрацію [2].

Рекомендаційні системи в основному працюють одним із двох способів: пропонуючи предмети, схожі на ті, що подобаються людині, або пропонуючи предмети, які подобаються людям, схожим на користувача. Вони можуть переглядати всі елементи, які користувач оцінив, а потім шукати елементи, схожі на те, що подобається користувачеві. Так працює багато онлайн-сервісів потокової передачі музики. Користувач починає з пісні або виконавця, і система створює музичний профіль. Потім система відбирає пісні, схожі за профілем, і відтворює їх. Якщо користувачу подобається нова пісня, профіль цієї пісні поєднується з існуючим профілем, щоб створити новий набір атрибутів, які подобаються користувачеві. Якщо користувачу не подобається нова пісня, атрибути цієї пісні підкреслюються у профілі. Ця тактика пошуку предметів, подібних до того, що, як відомо, подобається користувачеві, називається рекомендацією на основі елементів або моделей.

Алгоритми машинного навчання для рекомендаційних систем, як правило, поділяються на дві категорії; колаборативна фільтрація та фільтрація на основі вмісту. Однак сучасні системи рекомендацій поєднують їх обидві [3].

Фільтрація на основі вмісту враховує схожість атрибутів товару, а методи колаборативної роботи враховують подібність взаємодії клієнтів.

Як правило, ядром машинного навчання є розробка функції, що передбачає корисність предметів один для одного.

Оскільки стільки інформації в Інтернеті і так багато людей, які її використовують, для організацій стало життєво важливим пошук та надання даних своїм клієнтам відповідно до їх потреб та смаку.

### 1.1.2 Опис відеоігрової індустрії

В сучасному світі створення відеоігор є одним з найбільш великих сегментів індустрії розваг. Масштаби ігрової індустрії можна порівняти, наприклад, з кіноіндустрією. А по швидкості росту за останні п'ять років індустрія відеоігор істотно її випереджала.

За ступенем впливу на споживачів і залучення їх в інтерактивне оточення, пропоноване відеоіграми, цей сегмент уже давно виділяється серед інших видів розваг.

Геймдев або розробку ігор неможливо розглядати відокремлено від індустрії комп'ютерних ігор в цілому. Безпосередньо створення ігор - це лише частина комплексної «екосистеми», що забезпечує повний життєвий цикл виробництва, розповсюдження та споживання таких складних продуктів, як комп'ютерні ігри.

В рамках найбільш популярних ігрових напрямків, навколо конкретних ігор або серій утворюються ігрові спільноти, виникають нові сайти, форуми і групи в соціальних мережах. Найчастіше подібні групи генерують навіть більший потік інформації, ніж можуть собі дозволити розробники і видавці через власні ресурси і рекламу.

Але відеоігри являються не лише розвагою, а і є корисними в нашому житті.

За допомогою ігор, які потребують уваги до об'єктів, несподівано виникають і зникають на периферії поля зору, значно покращують так зване «низькорівневе сприйняття» і дійсно прискорюють прийняття рішень.

Відеоігри сприяють соціальної адаптації. Середній геймери дуже активно взаємодіють із суспільством, та загальна любов до ігор може стати основою довгих і щасливих відносин.

Люди, які добре грають в швидкі ігри, що вимагають постійного залучення і роздумів, здатні досягти успіху і в навчанні.

Одною з найважливіших частиною ігри є музикальне супроводження. Під час короткого проміжку часу ігрова ситуація може змінюватись декілька разів, тому не під всі жанри відеоігор можливо правильно і точно підібрати музичне супроводження.

Було обрано дослідити особливості роботи рекомендаційних систем при застосуванні штучного інтелекту на основі системи рекомендацій музичного супроводу в різних станах користувача в залежності від ігрових ситуацій.

### 1.1.3 Опис штучного інтелекту

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

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

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

Штучний інтелект найчастіше класифікується на такі поняття:

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

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

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

Деякими типами технологіями штучного інтелекту є:

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

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

- нейронні мережі (Neural Network) включають в себе глибоке навчання. Як уже згадувалося, нейронні мережі - це комп'ютерні системи, створені за зразком нейронних зв'язків у мозку людини. Штучним еквівалентом нейрона людини є перцептрон. Подібно до того, як пучки нейронів створюють нейронні мережі в мозку, стеки перцептронів створюють штучні нейронні мережі в комп'ютерних системах. Нейронні мережі навчаються, обробляючи навчальні приклади. Цей процес аналізує дані багато разів, щоб знайти асоціації та надати значення раніше невизначеним даним. За допомогою різних моделей навчання, як позитивне підкріплення, машину навчають, що вона успішно ідентифікувала об'єкт;

- когнітивне обчислення (Cognitive Computing) - ще один важливий компонент ШІ. Його мета - імітувати та покращити взаємодію між людьми та машинами. Когнітивні обчислення прагнуть відтворити процес мислення людини за допомогою комп'ютерної моделі, в даному випадку, розуміючи людську мову та значення

зображень. Разом когнітивні обчислення та штучний інтелект прагнуть наділити машини людською поведінкою та здібностями до обробки інформації;

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

- комп'ютерний зір (Computer Vision) - це техніка, що реалізує глибоке навчання та ідентифікацію шаблону для інтерпретації змісту зображення; включаючи графіки, таблиці та зображення у документах PDF, а також інший текст та відео. Комп'ютерний зір - це невід'ємне поле ШІ, що дозволяє комп'ютерам ідентифікувати, обробляти та інтерпретувати візуальні дані. Застосування цієї технології вже почало революцію в таких галузях, як дослідження та розробка та охорона здоров'я. Комп'ютерний зір використовується для швидшої діагностики пацієнтів за допомогою комп'ютерного зору та машинного навчання для оцінки рентгенівських сканів пацієнтів.

## 1.2 Опис сучасних проблем

Дослідження в рекомендаційних системах музики (Music Recommendation System), суттєво посилили інтерес як в наукових колах, так і в промисловості. Завдяки послугам потокового передавання музики, таким як Spotify, Pandora або Apple Music, сьогодні кожен має змогу отримати доступ до мільйонів музичних творів. MRS часто дуже успішно пропонують пісні, які відповідають уподобанням їх користувачів. Однак такі системи все ще далекі від досконалості і часто дають незадовільні рекомендації. Частково це пов'язано з тим, що смаки користувачів та музичні потреби в значній мірі залежать від безлічі факторів, які не враховуються в достатній мірі в сучасних підходах MRS, які, як правило, зосереджені на основній концепції взаємодії між елементами користувача, або на основі вмісту. На відміну від цього, для задоволення потреб користувачів у музичній розвазі потрібно враховувати внутрішні, зовнішні та контекстуальні аспекти слухачів, а також більш гідну інформацію про взаємодію. Наприклад, як відомо, особистість та емоційний стан слухачів (внутрішній), а також їх діяльність (зовнішній) впливають на музичні смаки та потреби. Так само як і контекстуальні фактори користувачів, включаючи погодні

умови, соціальне оточення тощо. Також композиція та анотація музичного списку відтворення або сеансу прослуховування розкривають інформацію про те, які пісні добре поєднуються або підходять для певного випадку. Тому дослідники та дизайнери MRS повинні переглянути своїх користувачів цілісно, щоб побудувати системи з урахуванням особливостей кожного користувача.

### 1.3 Аналіз існуючих методів добутку знань

Мета системи рекомендацій - пропонувати користувачам відповідні пункти. Для досягнення цього завдання існує дві основні категорії методів: методи колаборативної фільтрації та методи на основі вмісту.

Методи колаборативної роботи для рекомендаційних систем - це методи, які базуються виключно на минулих взаємодіях, зафіксованих між користувачами та елементами, з метою вироблення нових рекомендацій. Ці взаємодії зберігаються у так званій «матриці взаємодій між елементами користувача».

Головна ідея, яка керує методами колаборативної фільтрації, полягає в тому, що цих минулих взаємодій між елементами та користувачами достатньо для виявлення подібних користувачів або подібних елементів та прогнозування на основі цього [4].

Клас алгоритмів колаборативної фільтрації поділяється на дві підкатегорії, які зазвичай називаються підходами на основі пам'яті та моделями. Підходи, засновані на пам'яті, безпосередньо працюють зі значеннями записаних взаємодій, припускаючи відсутність моделі, і по суті засновані на пошуку найближчих сусідів (наприклад, знайти найближчих користувачів у зацікавленого користувача та запропонувати найбільш популярні предмети серед цих сусідів).

Головною перевагою колаборативної фільтрації є те, що вони не вимагають інформації про користувачів чи предмети, тому їх можна використовувати в багатьох ситуаціях. Більше того, чим більше користувачів взаємодіє з елементами, тим більше нових рекомендацій стає точнішим: для фіксованого набору користувачів та елементів нові взаємодії, зафіксовані з часом, приносять нову інформацію та роблять систему все більш ефективною [5].

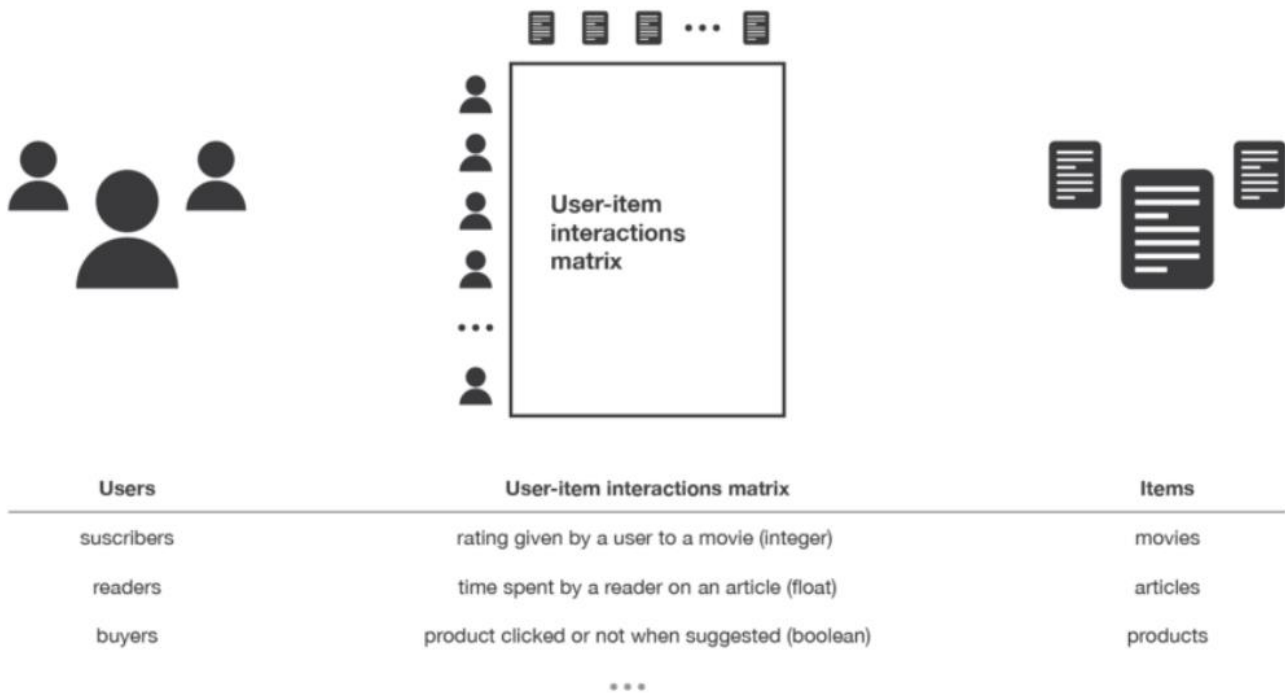


Рисунок 1.1 – Матриця взаємодій між елементами користувача.

Однак, оскільки він враховує лише попередні взаємодії для вироблення рекомендацій, колаборативна фільтрація страждає від “проблеми холодного старту”: неможливо рекомендувати що-небудь новим користувачам або рекомендувати новий елемент будь-яким користувачам, і багато користувачів або елементи мають занадто мало взаємодій ефективно обробляти. Цей недолік можна вирішити по-різному: рекомендувати випадкові предмети новим користувачам або нові предмети випадковим користувачам (випадкова стратегія), рекомендувати популярні предмети новим користувачам або нові предмети найбільш активним користувачам (стратегія максимального очікування), рекомендувати набір різних предмети для нових користувачів або нові предмети для набору різних користувачів (дослідницька стратегія) або, нарешті, за допомогою методу, що не є спільним для раннього життя користувача або предмета.

На відміну від методів колаборативної фільтрації, які покладаються лише на взаємодію між елементами користувача, підходи на основі вмісту використовують додаткову інформацію про користувачів та/або елементи. Якщо ми розглянемо приклад системи рекомендування фільмів, цією додатковою інформацією може бути, наприклад, вік, стать, робота чи будь-яка інша особиста інформація для користувачів,

а також категорія, основні актори, тривалість чи інші характеристики для фільмів (предметів) [6].

Потім ідея методів, що базуються на вмісті, полягає в спробі побудувати модель на основі доступних “функцій”, що пояснюють спостережувані взаємодії між елементами користувача.

Методи, що базуються на вмісті, страждають набагато менше від проблеми холодного старту, ніж метод колаборативної фільтрації: нових користувачів або елементи можна описати за їх характеристиками (вмістом), і тому для цих нових сутностей можна зробити відповідні пропозиції.

У методах колаборативної фільтрації на основі пам'яті не передбачається прихованої моделі. Алгоритми безпосередньо працюють із взаємодіями елементів користувача: наприклад, користувачі представлені їх взаємодією з елементами, а пошук найближчих сусідів за цими поданнями використовується для створення пропозицій. Оскільки не передбачається жодної прихованої моделі, ці методи теоретично мають низьке упередження, але велику дисперсію.

У методах колаборативної фільтрації на основі моделі передбачається деяка модель прихованої взаємодії. Модель навчена реконструювати значення взаємодії між елементами користувача на основі власного представлення користувачів та елементів. Потім на основі цієї моделі можна робити нові пропозиції. Користувачі та предмети, приховані в уявленні, отримані моделлю, мають математичне значення, яке для людини може бути важко інтерпретувати. Оскільки передбачається модель взаємодії між елементами користувача, ці методи теоретично мають більше упередження, але меншу дисперсію, ніж методи, які не передбачають прихованої моделі [7].

У методах фільтрації, заснованих на вмісті, також передбачається деяка модель прихованої взаємодії. Однак тут модель забезпечена вмістом, який визначає представлення користувачів та / або елементів: наприклад, користувачі представлені заданими функціями, і ми намагаємось змодельювати для кожного елемента тип профілю користувача, якому цей елемент подобається чи ні. Тут, як і для методів спільної роботи на основі моделі, передбачається модель взаємодії між елементами користувача. Однак ця модель є більш обмеженою (оскільки подано представлення користувачів та / або елементів) і, отже, метод, як правило, має найвищу упередженість, але найменшу дисперсію.

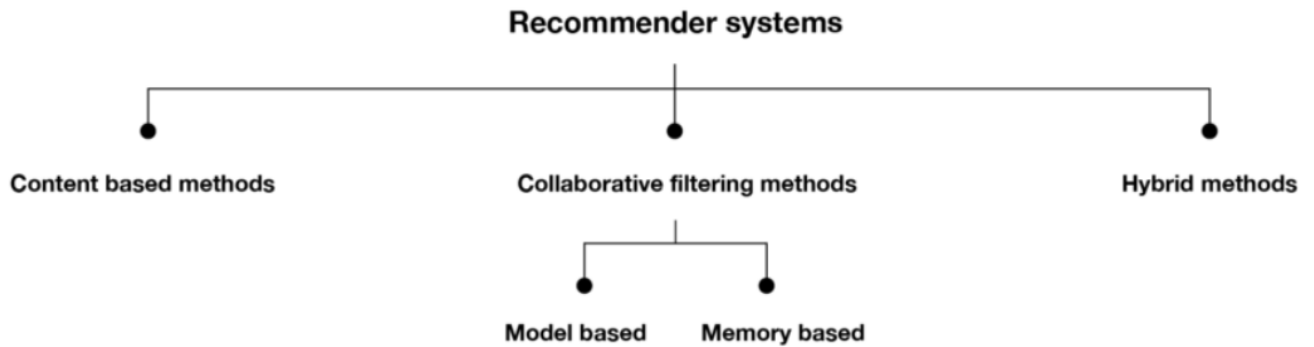


Рисунок 1.2 – Методи рекомендаційної системи

#### 1.4 Аналіз існуючих алгоритмів

Існує безліч варіацій завдання рекомендацій і в кожній є свої нюанси, але всі вони зводяться до декількох базових підходів. До найбільш класичним відносяться алгоритми Summary-based (неперсоналізована), Content-based (моделі на основі опису товару), Collaborative Filtering (колаборативна фільтрація), Matrix Factorization (методи засновані на матричному розкладанні) і деякі інші.

У центрі будь-якої рекомендаційної системи знаходиться так звана матриця переваг. Це матриця, по одній з знаходяться всі клієнти сервісу (Users), а по іншій - об'єкти рекомендації (Items). На перетині деяких пар (user, item) дана матриця заповнена оцінками (Ratings) - це відомий нам показник зацікавленості користувача в даному товарі, виражений за заданою шкалою (наприклад від 1 до 5) [8].

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1		3		5	
User 2	4		1		1
User 3		2			4
User 4				5	
User 5		2	3		4

Рисунок 1.3 – Матриця переваг

Користувачі зазвичай оцінюють лише невелику частину товарів, що є в каталозі, і завдання рекомендаційної системи - узагальнити цю інформацію і передбачити ставлення клієнта до інших товарів, про які нічого не відомо.

Шаблони споживання у людей різні, і не обов'язково повинні рекомендуватися нові товари. Можна показувати повторні позиції, наприклад, для поповнення запасу. За цим принципом виділяють дві групи товарів.

- Повторні. Наприклад, їжа, мило або техніка, тобто товари, які необхідно покупати циклічно.

- Неповторні. Наприклад, фільми або книги, тобто товари, які покупаються одноразово.

Поняття «інтерес» теж суб'єктивне. Деяким користувачам потрібні речі тільки з їхньої улюбленої категорії (*conservative recommendations*), а хтось, навпаки, більше відгукується на нестандартні товари або групи товарів (*risky recommendations*). Наприклад, відеохостинги може рекомендувати користувачеві тільки нові відео улюблених каналів, а може періодично рекомендувати йому нові відео з різних каналів або взагалі нові жанри. Тому варто вибирати стратегію показу рекомендацій під кожного клієнта окремо, за допомогою моделювання категорії клієнта.

Оцінки користувача товару можна отримати двома способами:

- явно (*explicit ratings*) - користувач сам ставить рейтинг товару, залишає відгук тощо.

- неявно (*implicit ratings*) - можна зробити непрямий висновок з його дій: купив товар - значить він йому подобається, довго читав опис - значить є інтерес, пропустив пісню, значить вона йому не подобається тощо [9].

Явні вподобання можуть здатися кращими за неявні, однак далеко не всі сайти та системи мають змогу надати цю можливість, та деякі користувачі не мають нагоду або бажання робити це. Тому найбільш універсальним підходом являється об'єднання цих способів отримання оцінки.

Неперсоналізовані рекомендації самі прості в реалізації. У них потенційний інтерес користувача визначається просто середнім рейтингом товару: «Усім подобається - значить сподобається і вам».

Із-за проблеми холодного старту такі оцінки являються недостовірними, тому їх часто редагують.

Перший спосіб - показувати не середнє значення, а згладжені середнє (*Damped Mean*). Сенс такий: при малій кількості оцінок відображається рейтинг більше тяжіє

до нікому безпечного «середньому» показнику, а як тільки набирається достатня кількість нових оцінок, «усереднюються» коригування перестає діяти.

Другий підхід - розраховувати по кожному рейтингу інтервали достовірності (Confidence Intervals). Математично, чим більше оцінок, тим менше варіація середнього і, отже, більше впевненість в його правильності. А в якості рейтингу можна виводити, наприклад, нижню межу інтервалу (Low CI Bound). При цьому зрозуміло, що така система буде досить консервативною, з тенденцією до заниження оцінок до нового товару [10].

Проблема холодного старту так само актуальна і для неперсоналізованих рекомендацій. Загальний підхід тут - замінювати те, що в даний момент не може бути пораховано, різними евристичними (наприклад, замінювати середнім рейтингом, використовувати алгоритм простіше, або взагалі не використовувати товар, поки не будуть зігнані дані).

У деяких випадках також важливо враховувати «свіжість» рекомендації. Це особливо актуально для статей або постів на форумах. Свіжі записи повинні частіше потрапляти в топ. Для цього використовуються коригувальні коефіцієнти (damping factors). Універсальної формули не існує, і кожен сервіс винаходить ту формулу, яка найкраще вирішує його завдання - перевіряється це емпірично.

Персональні рекомендації передбачають максимальне використання інформації про самого користувача, в першу чергу про його попередні покупки. Одним з перших з'явився підхід content-based filtering. В рамках даного підходу опис товару (content) зіставляється з інтересами користувача, отриманими з його попередніх оцінок. Чим більше товар цим інтересам відповідає, тим вище оцінюється потенційна зацікавленість користувача. Очевидна вимога тут - у всіх товарів в каталозі має бути опис.

Предметом персональних рекомендацій частіше були товари з неструктурованим описом: фільми, книги, статті. Такими ознаками можуть бути, наприклад, текстові описи, рецензії, склад акторів та інше. Однак ніщо не заважає використовувати і звичайні числові або категоріальні ознаки.

Неструктуровані ознаки описуються типовим для тексту способом - векторами в просторі слів (Vector-Space model). Кожен елемент такого вектора - ознака, потенційно що характеризує інтерес користувача. Аналогічно, продукт - вектор в тому ж просторі [10].

У міру взаємодії користувача з системою, векторні опису придбаних ним товарів об'єднуються (підсумовуються і нормалізуються) в єдиний вектор і, таким чином, формується вектор його інтересів. Далі досить знайти товар, опис якого найближче до вектору інтересів, тобто вирішити задачу пошуку  $n$  найближчих сусідів.

Не всі елементи однаково значущі: наприклад, союзні слова, очевидно, не несуть ніякої корисної навантаження. Тому при визначенні числа співпадаючих елементів в двох векторах все вимірювання потрібно попередньо зважувати по їх значимості. Дану задачу вирішує добре відоме в Text Mining перетворення TF-IDF, яке призначає більшу вагу більш рідкісним інтересам. Збіг таких інтересів має більше значення при визначенні близькості двох векторів, ніж збіг популярних.

$$w_{x,y} = tf_{x,y} * \log\left(\frac{N}{df_x}\right), \quad (2.1)$$

де  $tf_{x,y}$ - частота слова  $x$  в описі товару  $y$ ,  $df_x$  – кількість товару, яке містить слово  $x$ ,  $N$  – загальна кількість товару.

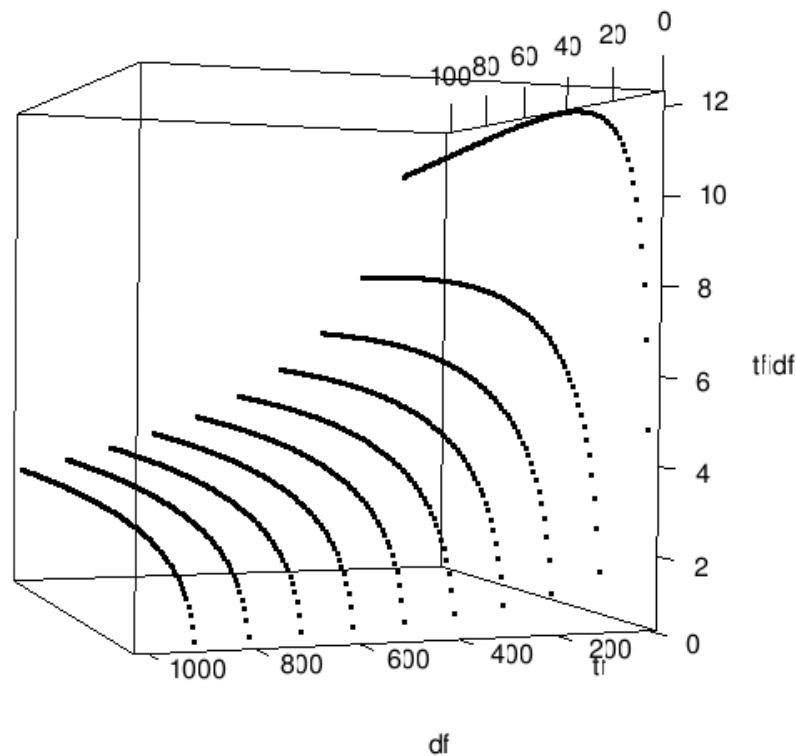


Рисунок 1.4- Приклад векторів за принципом TF-IDF

В рамках підходу рекомендації генеруються на підставі інтересів інших схожих користувачів. Такі рекомендації є результатом «колаборації» безлічі користувачів.

Класична реалізація алгоритму заснована на принципі  $k$  найближчих сусідів. Тобто для кожного користувача шукаємо  $k$  найбільш схожих на нього і доповнюємо інформацію про користувача відомими даними по його сусідам. Так, наприклад, якщо відомо, що ваші сусіди по інтересам в захваті від нового альбому музичної групи, а ви його ще не слухали, це гарний привід запропонувати вам даний музичний альбом.

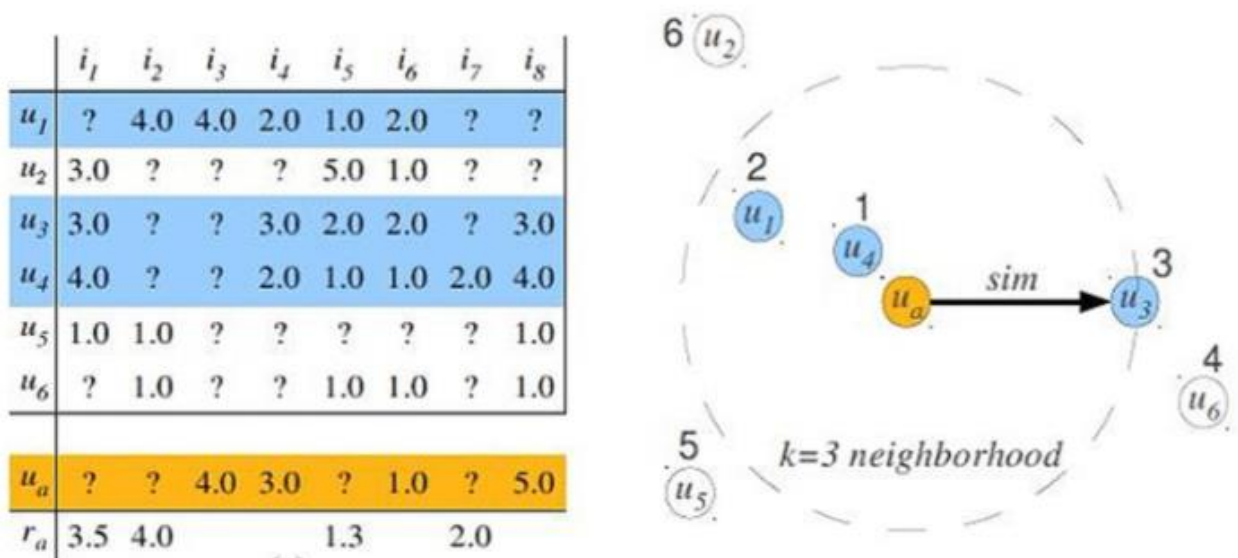


Рисунок 1.5 – Принцип роботи  $k$ -найближчих сусідів

У матриці переваг жовтим кольором виділено користувача, для якого ми хочемо визначити оцінки з нових товарів (знаки питання). Синім кольором виділені три його найближчих сусіда.

«Схожість» - в даному випадку синонім «кореляції» інтересів і може вважатися безліччю способів.

У класичній реалізації алгоритму є один явний мінус - він погано застосуємо на практиці через квадратичної складності. Дійсно, як будь-який метод найближчого сусіда, він вимагає розрахунку всіх попарних відстаней між користувачами (а користувачів можуть бути мільйони). Тобто складність розрахунку матриці відстаней

буде,  $O(n^2m)$  де  $n$  - число користувачів, а  $m$  - число товарів. При мільйон користувачів для зберігання матриці відстаней в сирому вигляді, потрібно багато ресурсів.

Важливий етап підготовки даних – нормалізація оцінок.

Оскільки всі користувачі по різному оцінюють товари, то потрібно їх привести до єдиної шкали, щоб алгоритм міг правильно порівнювати їх між собою.

Нормалізувати можна декількома способами:

- центруванням (mean-centering) - з оцінок користувача просто віднімаємо його середню оцінку;

- стандартизацією (z-score) - на додаток до центрування ділимо оцінку її на стандартне відхилення у користувача;

- подвійний стандартизацією - перший раз нормуємо оцінками користувача, другий раз - оцінками товару.

«Схожість» або кореляцію переваг двох користувачів можна вважати різними способами.

## 1.5 Постановка задачі

Об'єктом дослідження є підбір найбільш ефективного алгоритму рекомендаційної системи в рамках програмної реалізації.

Метою дослідження є отримання найбільш достовірних та точних рекомендацій за допомогою розробленої системи.

Задачею дослідження є проектування системи, яка за допомогою обробки різних вхідних даних під час різних ігрових ситуацій буде надавати рекомендації музичного супроводження.

## 2 РОЗРОБКА МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

### 2.1 Розробка вимог до системи

Метою системи є надавання рекомендацій музичного супроводу в різних станах користувача в залежності від ігрових ситуацій.

Система буде реалізована у вигляді сервісно-орієнтованої архітектури.

Сервісно-орієнтована архітектура являє собою модульний підхід до розробки програмного забезпечення, заснований на використанні розподілених, слабо пов'язаних замінних компонентів, оснащених стандартизованими інтерфейсами для взаємодії за стандартизованими протоколами.

Визначені наступні сервіси системи:

- рекомендаційний сервіс – сервіс який буде рекомендувати предмет на основі даних, які буде надавати інші сервіси системи;
- сервіс бізнес-логіки – сервіс, в якому будуть реалізовані вся бізнес логіка системи, наприклад обчислення оцінки та обчислення стану користувача в залежності від даних, які будуть надходити з сервісу введення даних;
- сервіс введення даних – сервіс, який буде надавати системі інформацію про користувача та його стан;
- сервіс введення оцінки – сервіс, який буде надавати системі інформацію про оцінки предмета користувачем.

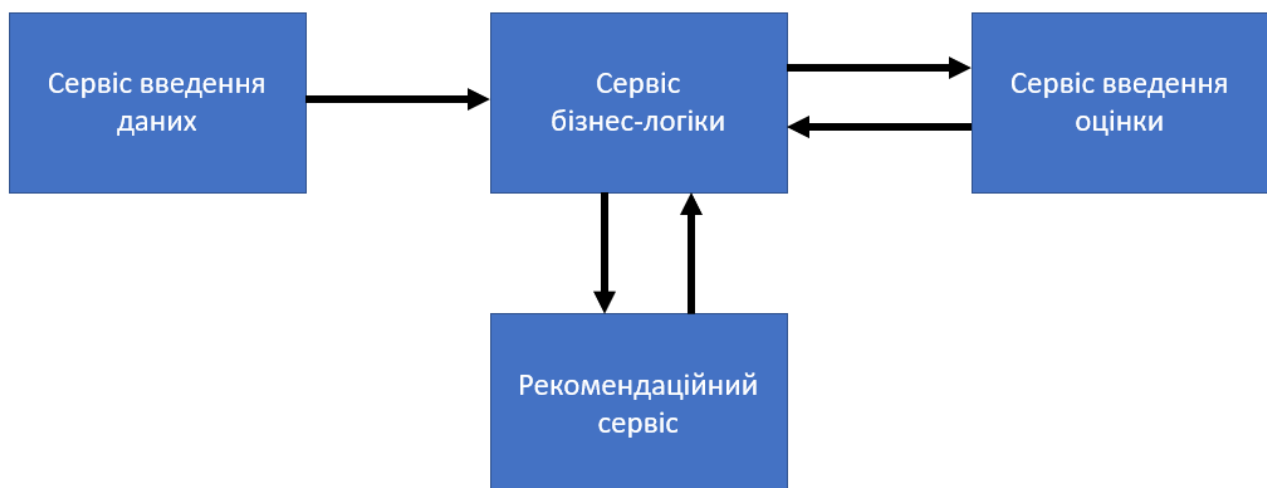


Рисунок 2.1 – Взаємодія сервісів в системі

## 2.2 Визначення функціональних вимог до системи

Для визначення функціональних вимог інформаційної системи розроблена її функціональна модель з використанням стандарту IDEF0 [12].

Стандарт IDEF0 використовується для створення функціональної моделі, що відображає структуру і функції системи, а також потоки інформації і матеріальних об'єктів, що зв'язують ці функції. Функціональна модель призначена для опису існуючих бізнес-процесів системи [13].

За допомогою методології IDEF0 будується ієрархічна система діаграм - одиничних описів фрагментів системи. Спочатку проводиться опис системи в цілому і її взаємодії з навколишнім світом (контекстна діаграма), після чого проводиться функціональна декомпозиція - система розбивається на підсистеми і кожна підсистема описується окремо (діаграми декомпозиції). Потім кожна підсистема розбивається на більш дрібні і так далі до досягнення потрібного ступеня деталізації [14].

Кожна IDEF0-діаграма містить блоки і дуги. Блоки зображують функції модельованої системи. Дуги пов'язують блоки разом і відображають взаємодії і взаємозв'язку між ними [15].

На контекстній діаграмі системи представлені наступні складові:

- вхідні дані:

- «Дані стану користувача» - вхідні дані системи, за допомогою яких система буде робити рекомендації;

- «Оцінка користувача» - вхідні дані системи, за допомогою яких система буде коригувати вибір рекомендацій, користувачу надходять рекомендаційне супроводження для даної ігрової ситуації, і він повинен надати дані, як точно підходить рекомендоване музичне супроводження під ситуацію;

- вихідні дані:

- «Рекомендоване музичне супроводження» - музичне супроводження, яке рекомендує система, виходячи із вхідних даних, які були надані системі;

- механізм:

- «Користувач» - користувач системи, які надає дані;

- «ПЗ» - програмне забезпечення, завдяки якому система працює;

- управління:

- «Умови користування ресурсом» - умови користування, які потрібен притримуватись користувач системи.

Також представлена основна функція системи, а саме «Функціонування системи надавання рекомендацій музичного супроводження».

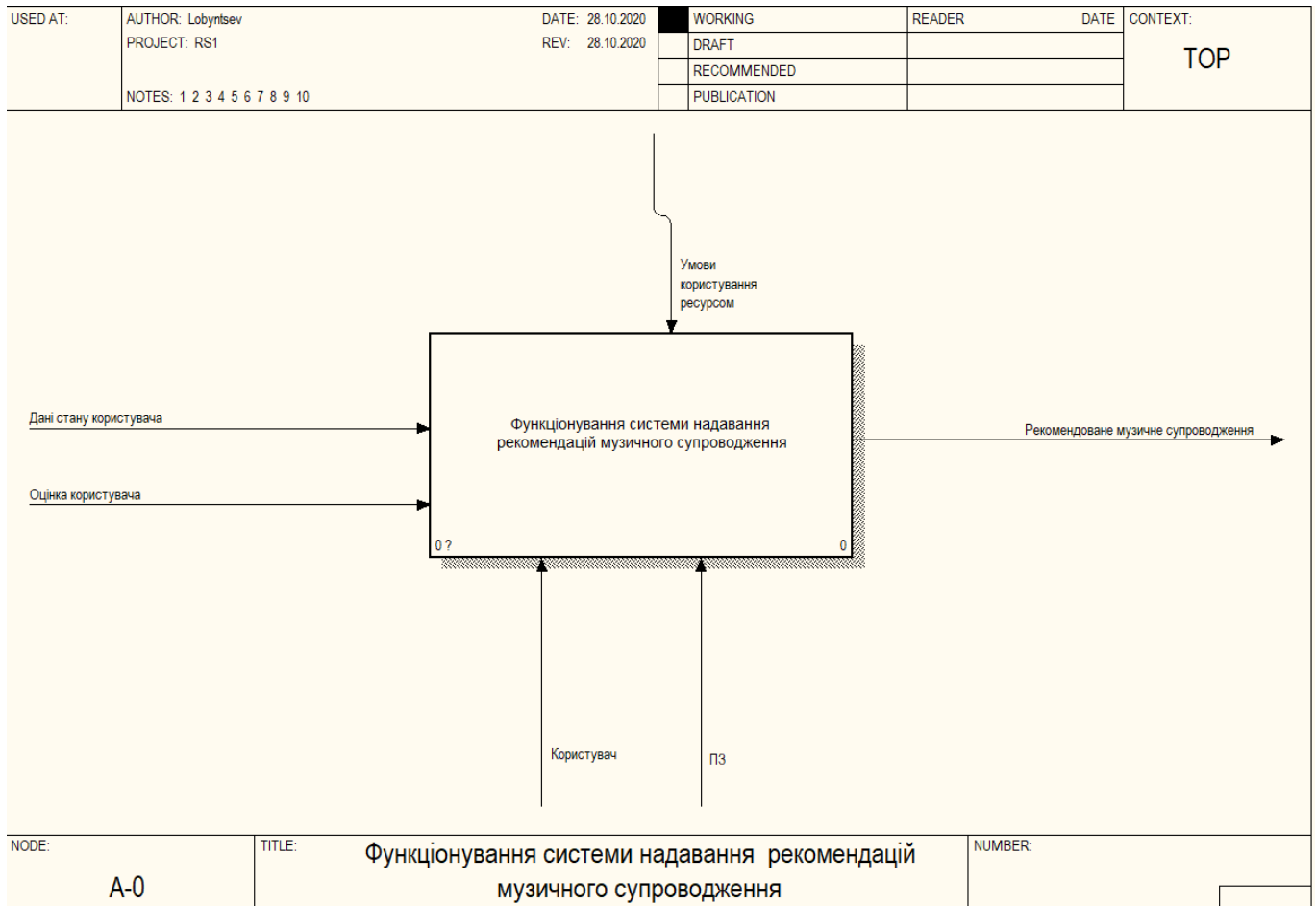


Рисунок 2.2 – Контекста діаграма

На рисунку 2.3 зображена діаграма декомпозиції контекстної діаграми. Вона дозволяє уточнити функцію системи «Функціонування системи надавання рекомендацій музичного супроводження».

Виділенні наступні функції:

- «Обробка даних стану користувача» - функція системи, що оброблює вхідні дані стану користувача та трансформує їх у вигляд необхідній для системи та подальшого оброблення. Вхідними даними функції є «Дані стану користувача». Вихідними даними є «Оброблені дані стану користувача». Механізмом функції є «Користувач» та «ПЗ». Управління є «Умови користування ресурсом»;



## 2.3 Розробка моделі потоків даних системи

Наступним етапом, при визначенні вимог до інформаційної системи, є створення діаграми потоків даних DFD. Інструменти методології DFD дозволяють відображати джерела і адресати даних, ідентифікувати процеси і групи даних, що зв'язують в потоки одну функцію з іншого, і ефективно використовуються для опису процесів при впровадженні процесного підходу до управління організацією, так як дозволяє максимально знизити суб'єктивність опису бізнес процесів. Крім того, нотація DFD дозволяє описувати потоки документів (документообіг) і потоки ресурсів (наприклад, рух матеріалів від однієї роботи до іншої)[16].

На рис. 2.3 зображена концептуальна діаграма DFD.

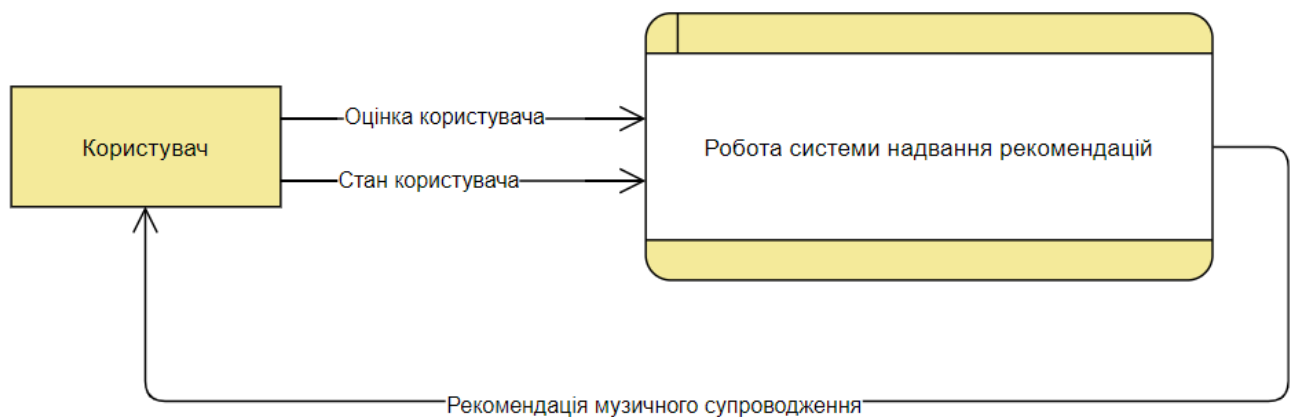


Рисунок 2.4 – Концептуальна DFD системи надання рекомендацій

На рисунку 2.5 зображена діаграма декомпозиції концептуальної діаграми DFD, в якій присутні основні процеси:

- «Обробка даних оцінки користувача» - оброблює вхідні дані оцінки користувача та трансформує їх у вигляд необхідній для системи та подальшого оброблення;
- «Обробка даних стану користувача» - оброблює вхідні дані стану користувача та трансформує їх у вигляд необхідній для системи та подальшого оброблення;
- «Зіставлення оцінки та стану користувача» - зіставлення даних оцінки та стану користувача для процесу вибору музичного супроводження;
- «Вибір музичного супроводження» - вибір рекомендованого музичного супроводження в залежності від вхідних даних.

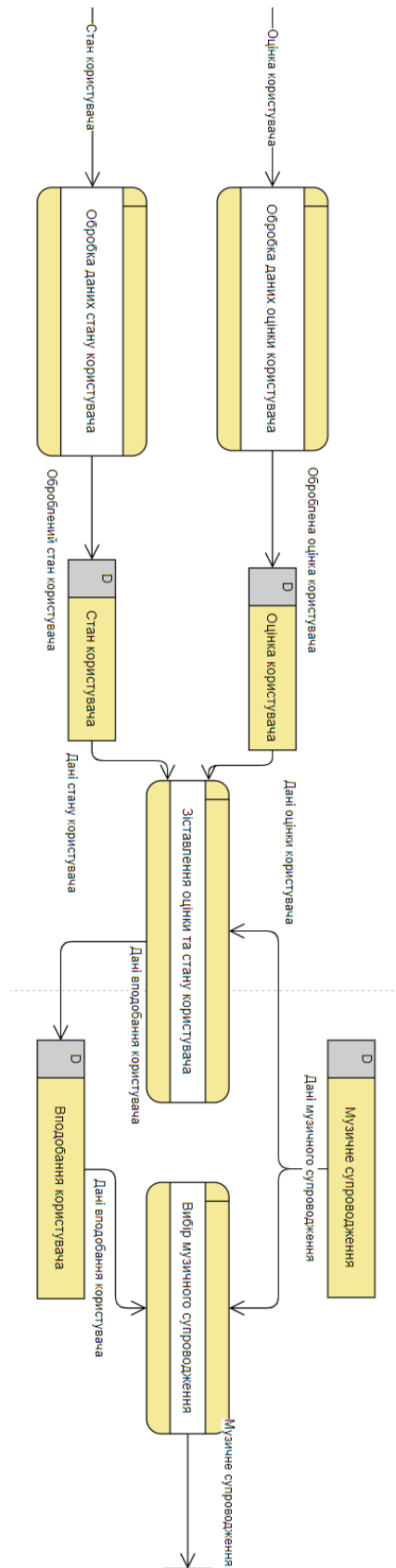


Рисунок 2.5 – Діаграма декомпозиції процесу «Робота системи надавання рекомендацій»

## 2.4 Діаграма варіантів використання

Діаграма варіантів використання (Use case diagram) описує найбільш загальне уявлення функціонального призначення системи.

Суть даної діаграми складається в наступному: проектована система представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою так званих варіантів використання. При цьому актором (actor) або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на моделювану систему так, як визначить сам розробник. У свою чергу, варіант використання (use case) служить для опису сервісів, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який чинять системою при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів з системою [17].

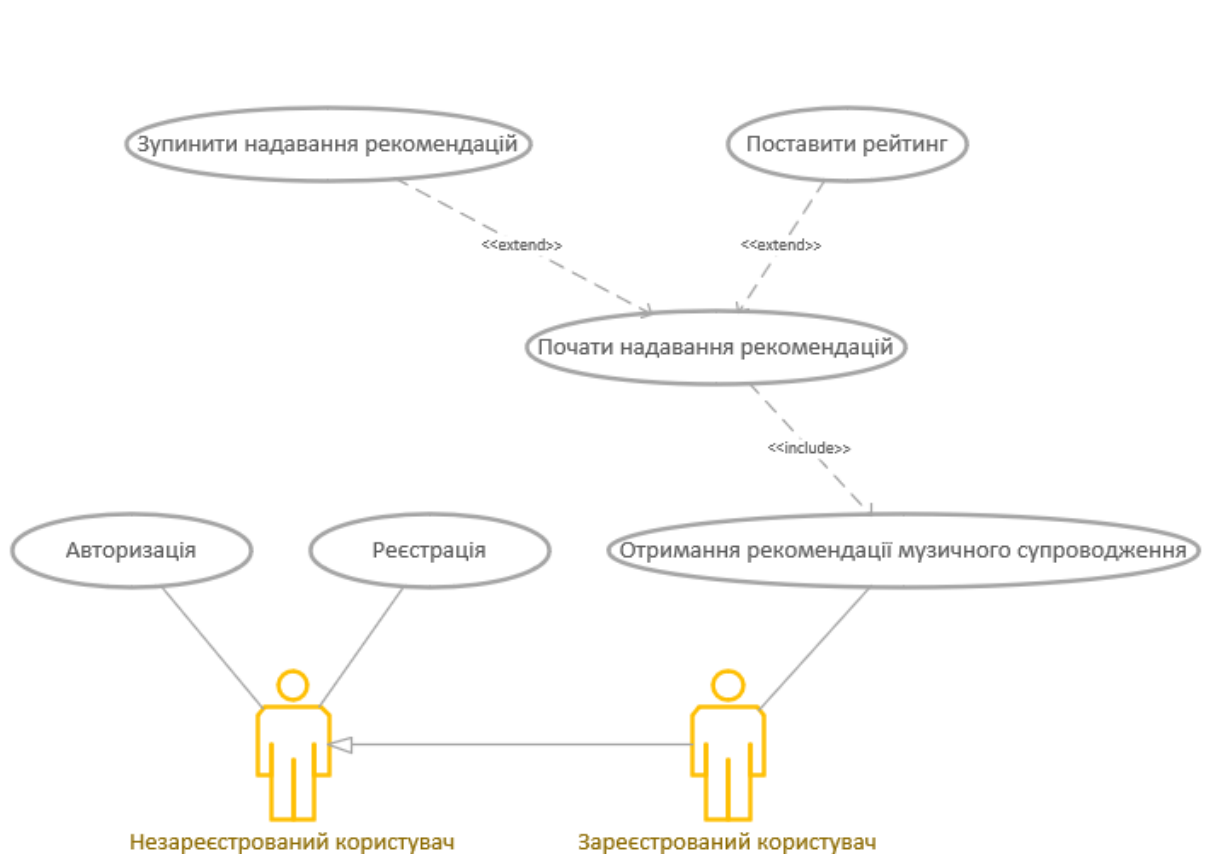


Рисунок 2.6 – Діаграма варіантів використання системи

На рис. 2.6 зображена діаграма варіантів використання, яка дозволяє сформулювати вимоги до функціональної поведінки системи, що проектується.

У даній діаграмі зовнішніми сутностями являються актори, а саме: зареєстрований та незареєстрований користувач.

В свою чергу, варіант використання слугує для опису описання функцій, які система надає акторам. При цьому на діаграмі ніяк не описується, яким саме чином буде реалізована взаємодія акторів з системою і власне виконання варіантів використання.

На даній діаграмі відношення асоціації між актором та варіантом використання вказує на той факт, що актор є ініціатором відповідного варіанта використання.

Відношення включення між двома варіантами використання вказує на те, що деяка поведінка одного варіанта використання включається в якості зіставного фрагмента в послідовності поведінки іншого.

Відношення розширення означає, що властивості варіанта використання одного варіанта в деяких випадках можуть бути виконані за рахунок функціональності варіанта використання іншого варіанта.

Таким чином, розроблена діаграма використання відображає функціональне призначення системи.

## 2.5 Діаграма класів системи

Діаграма класів слугує для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. Вона описує структуру системи, показуючи її класи, їх атрибути і оператори, і навіть взаємозв'язку цих класів.

На рисунку 2.7 зображено взаємодію класів між собою.

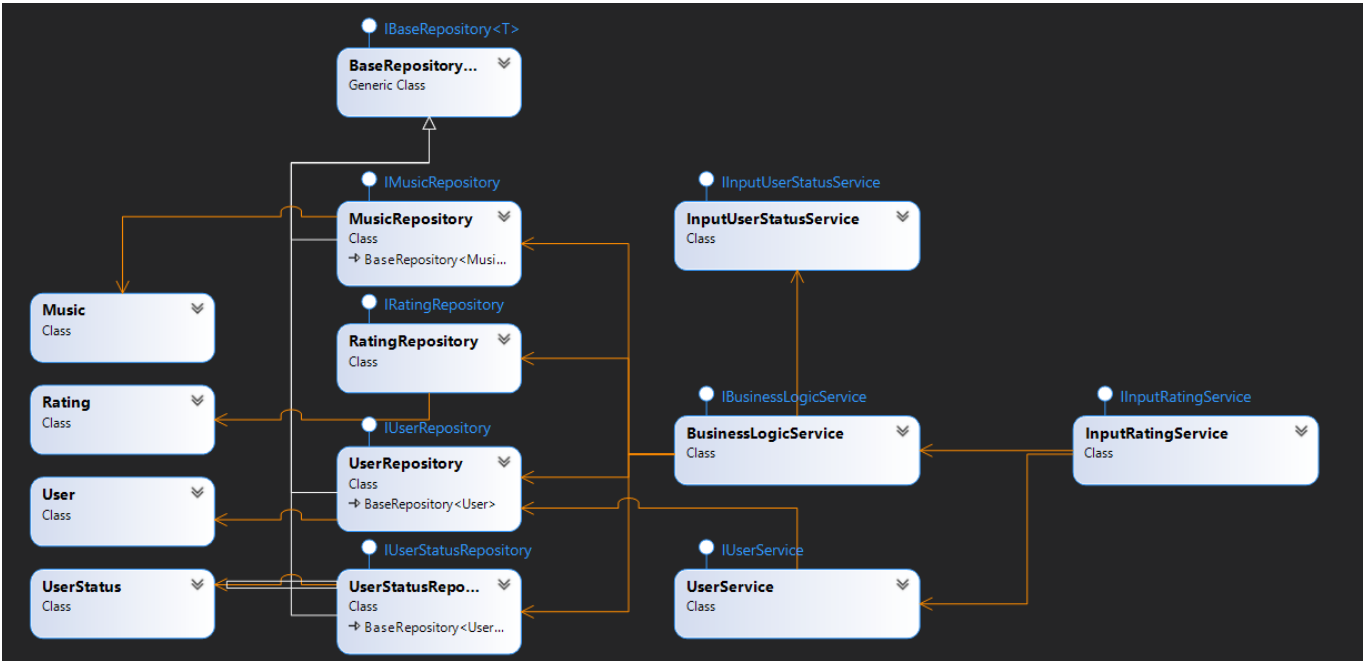


Рисунок 2.7 – Діаграма класів

На рисунку 2.8 зображено діаграму класів – сервісів. Вони забезпечують основний функціонал системи.

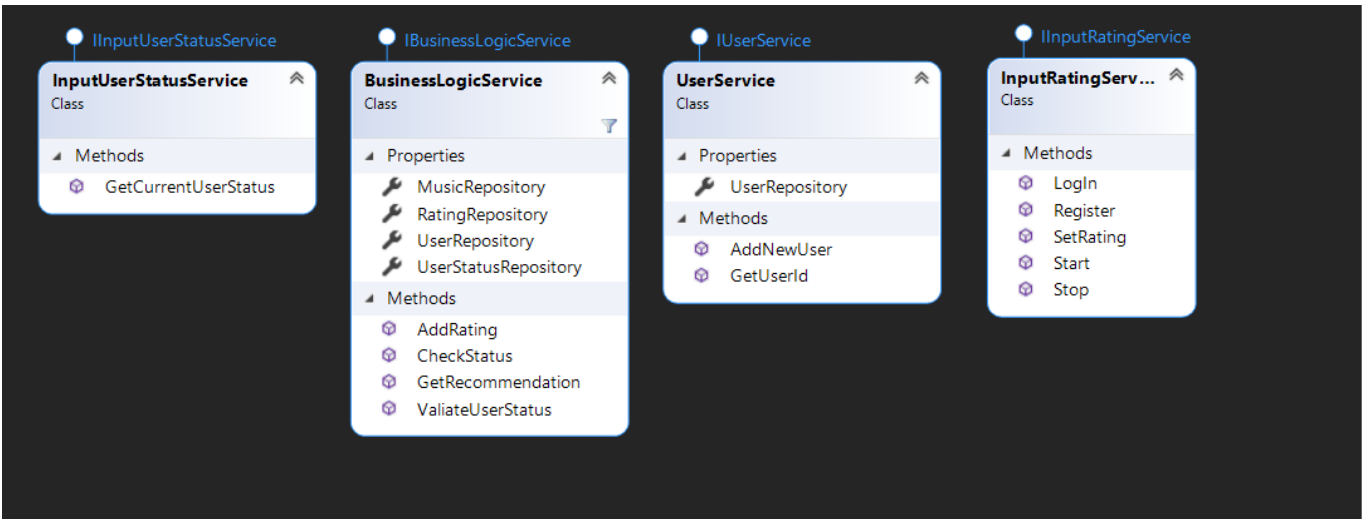


Рисунок 2.8 – Діаграма класів – сервісів

Клас «InputUserStatusService» - являється сервісом, який відповідає за надання системі даних стану користувача у даний період часу.

Він має метод «GetCurrentUserStatus», який повертає модель стану користувача.

Клас «BusinessLogicService» - являється сервісом, який відповідає за основний функціонал системи, та являється «мозком» системи.

Він використовує класи – репозиторії:

- «MusicRepository» - клас - репозиторій, який відповідає за роботу з базою даних сутності «Music»;
- «RatingRepository» - клас - репозиторій, який відповідає за роботу з базою даних сутності «Rating»;
- «UserRepository» - клас - репозиторій, який відповідає за роботу з базою даних сутності «User»;
- «UserStatusRepository» - клас - репозиторій, який відповідає за роботу з базою даних сутності «UserStatus».

Та має такі методи, як:

- «AddRating» - ставить оцінку користувача музичного супроводження у знаходженні в певному статусі користувача;
- «CheckStatus» - перевіряє стан користувача;
- «GetRecommendation» - надає користувачу рекомендацію щодо музичного супроводження;
- «ValidateUserStatus» - співвідносить стан користувача, який був отриманий від сервісу «InputUserStatusService» до стану, який знаходиться в базі даних.

Клас «UserService» - являється сервісом, який відповідає за логіку роботи з користувачами, та має такі методи, як:

- «AddNewUser» - перевіряє та додає нового користувача до бази даних;
- «GetUserId» - надає ідентифікатор користувача, за його логіном та паролем.

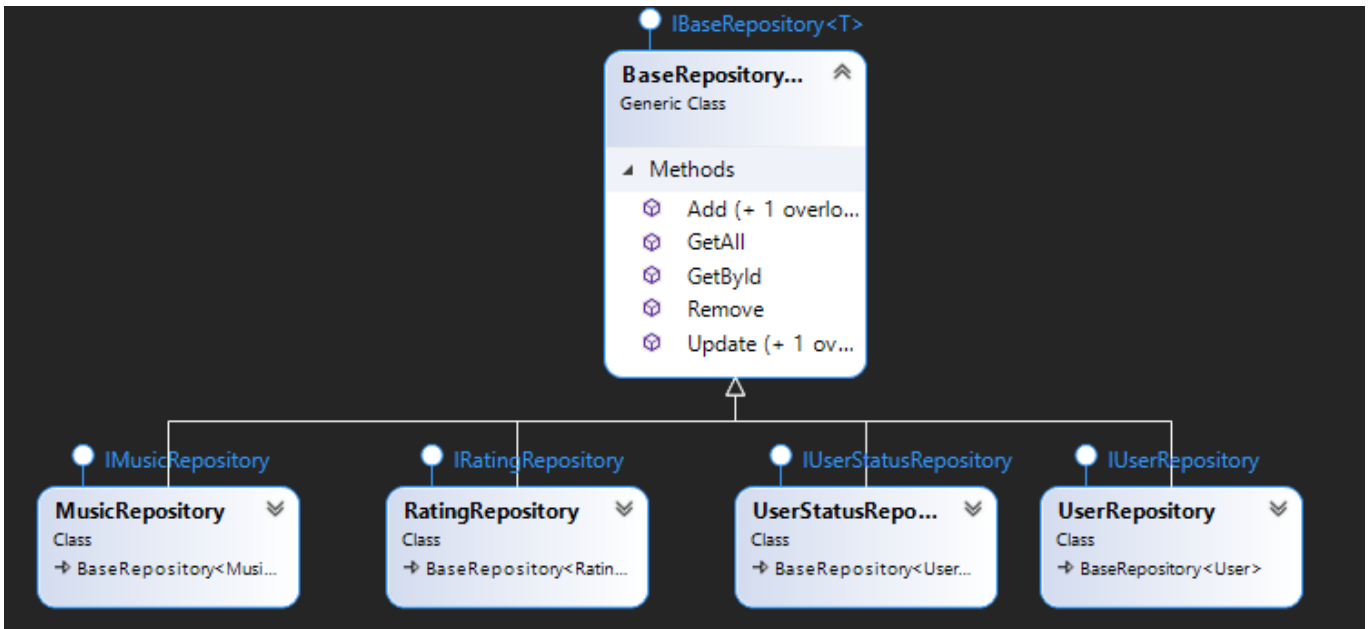
Клас «InputRatingService» - являється основним класом для роботи з користувачем, та являється інтерфейсом користувача (User Interface).

Він має такі методи, як:

- «LogIn» - метод, який дозволяє користувачу авторизуватися в системі;
- «Register» - метод, що дозволяє зареєструватися новому користувачу системи;
- «SetRating» - надає користувачу змогу поставити оцінку, яка показує наскільки надане рекомендоване музичне супроводження було влучним;
- «Start» - метод, який починає надавати користувачу рекомендації музичного супроводження, залежно від його стану;
- «Stop» - метод, який припиняє надавати користувачу рекомендації музичного супроводження

Також клас використовує класи - сервіси «BusinessLogicService» та «UserLogicService» для виконання своїх функціональних можливостей.

На рисунку 2.9 зображено класи-репозиторії. Класи «MusicRepository», «RatingRepository», «UserRepository», «UserStatusRepository» унаслідують клас «BaseRepository», який має реалізацію основних CRUD методів.



Рисунко 2.9 – Діаграма класів-репозиторієв

На рисунку 2.10 зображено класи-сутності.

Клас «Rating» - являється сутністю, яка має властивості оцінки певного музичного супроводження певного користувача у певний стан користувача. Клас має наступні поля:

- «Id» - ідентифікатор сутності;
- «MusicId» - ідентифікатор музичного супроводження, яке було рекомендоване;
- «RatingValue» - оцінка музичного супроводження користувачем, являє собою число від 1 до 10, та вказує наскільки вдало було надане музичне супроводження;
- «UserId» - ідентифікатор користувача;
- «UserStatus» - ідентифікатор статусу користувача.

Клас «Music» - являється сутністю, яка має властивості музичного супроводження. Клас має наступні поля:

- «Id» - ідентифікатор сутності;
- «Name» - назва музичного супроводження;

- «Path» - шлях до музичного файлу.

Клас «User» - являється сутністю, яка має властивості користувача, та має наступні поля:

- «Id» - ідентифікатор користувача;
- «UserName» - ім'я користувача;
- «Password» - пароль користувача.

Клас «UserStatus» - являється сутністю, яка має властивості стану користувача. Було вирішено, що всі можливі стани користувача будуть задані заздалегідь. Кожен стан буде матиме свої порогові межі. Клас має наступні поля:

- «Id» - ідентифікатор сутності;
- «MaxAmountClicksButton» - верхня межа кількості натиснутих клавiш клавіатури за визначений час;
- «MinAmountClicksButton» - нижня межа кількості натиснутих клавiш клавіатури за визначений час;
- «MaxAmountClicksMouse» - верхня межа кількості натиснутих клавiш миші за визначений час;
- «MinAmountClicksMouse» - нижня межа кількості натиснутих клавiш миші за визначений час;

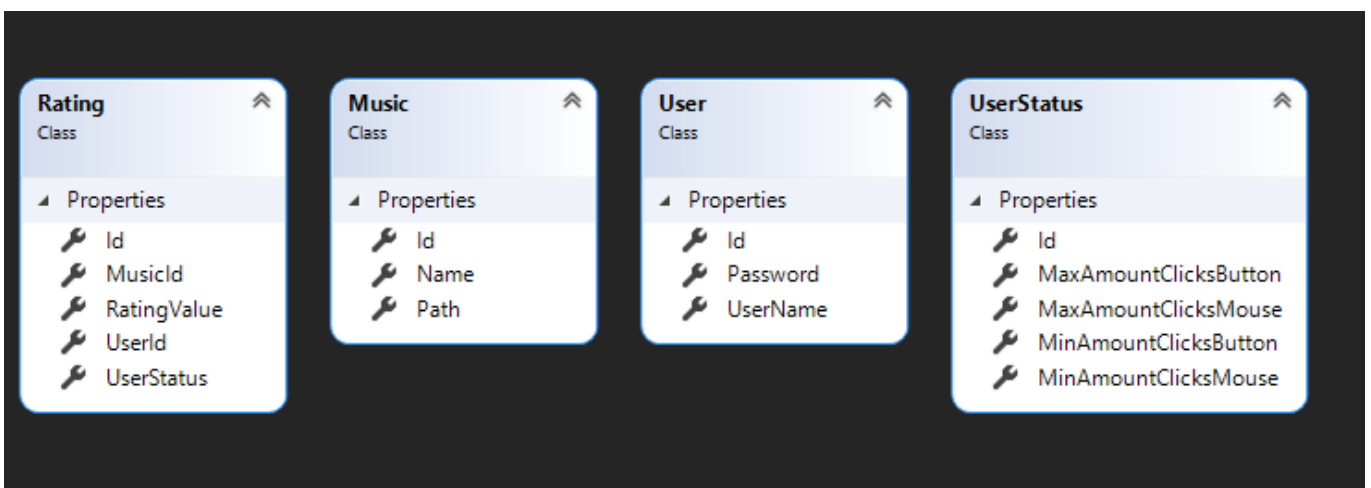


Рисунок 2.10 – Діаграма класів-сутностей

## 2.6 Діаграма послідовності дій

Діаграма послідовності дій описує поведінку системи на основі вказівки послідовності переданих повідомлень, що впорядковані за часом їх прояву. На

діаграмі об'єкти в основному представляють екземпляри класу або сутності, які володіють поведінкою. Як об'єкти можуть виступати користувачі, які ініціюють взаємодію, класи, що володіють поведінкою в системі або програмні компоненти, а іноді і системи в цілому. Діаграми послідовності дій зазвичай відповідають реалізаціям прецедентів в логічному представленні системи [18].

На рисунку 2.11 зображена діаграма послідовності дій для варіанта використання системи «Почати надавання рекомендацій».

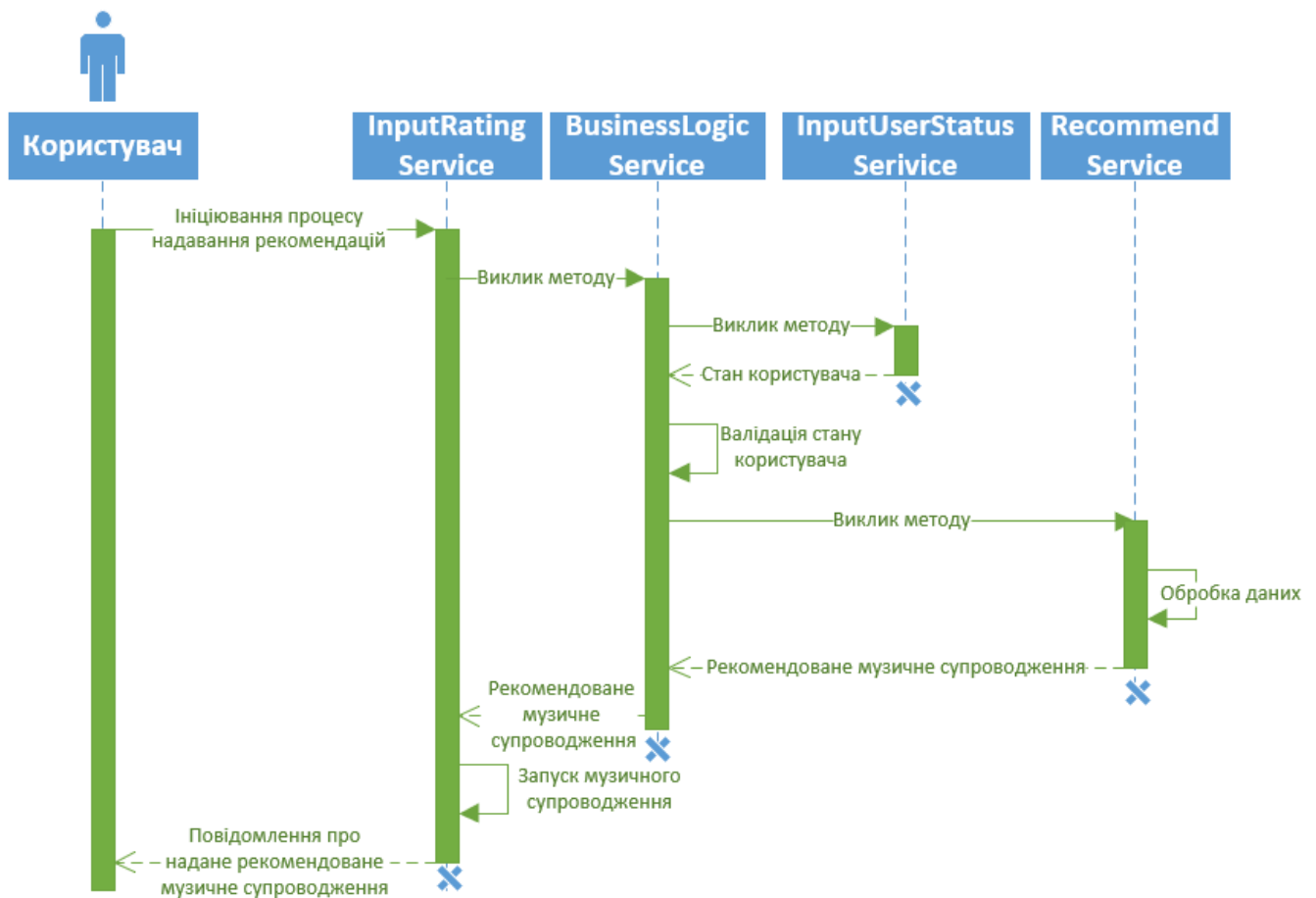


Рисунок 2.11 – Діаграма послідовності дій для надавання рекомендацій

Користувач системи є ініціатором цієї функції в системі. Після ініціювання процесу надавання рекомендацій, в «InputRatingService» викликається метод початку надання рекомендації в «BusinessLogicService». Наступним кроком є виклик методу з «InputUserStatusService», який повертає стан користувача в даний момент часу. Після чого йде валідація стану користувача, а саме з бази даних беруться дані верхніх та нижніх меж стану користувача та визначається в якому стані перебуває користувач.

Надалі йде виклик методу з рекомендуючого сервісу. На основі даних, які зберігаються в системі та які надійшли до системи, сервіс робить висновок та надає рекомендацію музичного супроводження. Після чого вся інформація повертається до «InputRatingService» та в цьому сервісі запускається музичне супроводження.

Встановив час, по закінченні якого йде перевірка стану користувача. Якщо стан користувача змінився, то надається нова рекомендація.

## 3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Обґрунтування вибору мови програмування

Для виконання роботи було обрано мову C#.

Головна причина вивчення C# криється у величезній кількості інструментів і фреймворків, які підтримує цю мову. Visual Studio є однією з найбільш багатофункціональних і потужних середовищ розробки на ринку. Фреймворк .Net надає сотні бібліотек для створення веб-додатків, забезпечення безпеки, роботи з файловими системами.

C# – один з найпопулярніших мов програмування. Це важливо для розробників, оскільки популярність мови прямо пропорційна тому, наскільки для нього будуть доступні онлайн-матеріали.

Гнучкість мови C# є величезною перевагою, в порівнянні з деякими мовами програмування. Різноманітність додатків, які можуть бути розроблені за допомогою #, .Net і Visual Studio практично безмежно. [19]

В якості рекомендуючої системи було обрано бібліотеку MyMediaLite. Ця бібліотека легка в користуванні та розглядає два найпоширеніші сценарії спільної фільтрації:

- прогнозування рейтингу;
- прогнозування товару за позитивними відгуками (наприклад, за кліками, вподобаннями чи діями покупки).

### 3.2 Обґрунтування вибору СУБД

SQL Server є однією з найбільш популярних систем управління базами даних в світі. Дана СУБД підходить для самих різних проектів: від невеликих додатків до великих високонавантажених проектів. SQL Server був створений компанією Microsoft. Перша версія вийшла в 1987 році.

SQL Server характеризується такими особливостями як:

- продуктивність. SQL Server працює дуже швидко;
- надійність і безпека. SQL Server надає шифрування даних;
- простота. З даної СУБД відносно легко працювати і вести адміністрування.

Центральним аспектом в MS SQL Server, як і в будь-якій СУБД, є база даних. База даних являє сховище даних, організованих певним способом. Нерідко фізично база даних представляє файл на жорсткому диску, хоча таке відповідність необов'язково. Для зберігання і адміністрування баз даних застосовуються системи управління базами даних (database management system) або СУБД (DBMS). І якраз MS SQL Server є однією з такою СУБД.

Для організації баз даних MS SQL Server використовує реляційну модель. Ця модель баз даних була розроблена ще в 1970 році Едгаром Коддом. А на сьогоднішній день вона фактично є стандартом для організації баз даних.

Реляційна модель передбачає зберігання даних у вигляді таблиць, кожна з яких складається з рядків і стовпців. Кожен рядок зберігає окремий об'єкт, а в стовпчиках розміщуються атрибути цього об'єкта.

Для ідентифікації кожного рядка в рамках таблиці застосовується первинний ключ (primary key). В якості первинного ключа може виступати один або декілька стовпців. Використовуючи первинний ключ, ми можемо посилатися на певну рядок в таблиці. Відповідно два рядки не можуть мати один і той же первинний ключ.

Через ключі одна таблиця може бути пов'язана з іншого, тобто між двома таблицями можуть бути організовані зв'язку. А сама таблиця може бути представлена у вигляді відносини ("relation").

Для взаємодії з базою даних застосовується мова SQL (Structured Query Language). Клієнт (наприклад, зовнішня програма) відправляє запит на мові SQL за допомогою спеціального API. СУБД належним чином інтерпретує і виконує запит, а потім посилає клієнту результат виконання. [20]

### 3.3 Моделювання бази даних системи

Логічне та фізичне моделювання даних інформаційної системи проводилося з використанням засобу редагування діаграм «Visio».

Логічна модель даних інформаційної системи зображена на рисунку 3.1.

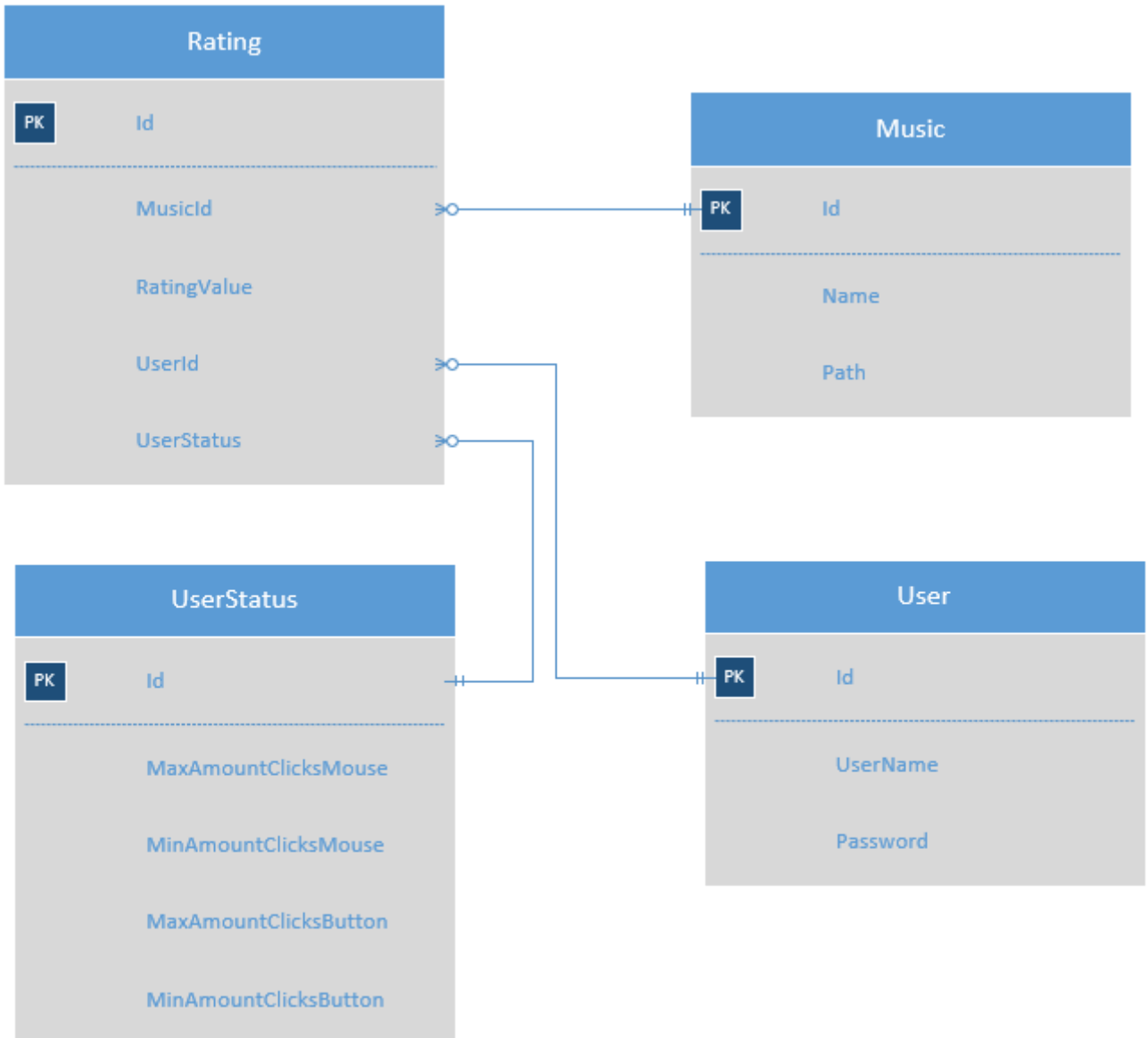


Рисунок 3.1 – Логічна модель даних системи

Суті предметної області, які увійшли в логічну модель представлені в табл.3.1.

Таблиця 3.1 – Сутності логічної моделі даних системи

№	Найменування сутності	Найменування атрибута	Тип даних	Назначення
1.	User	Id	Текст	Первинний ключ
		Username	Текст	Ім'я користувача
		Password	Текст	Хеш паролю

Продовження таблиці 3.1

№	Найменування сутності	Найменування атрибута	Тип даних	Назначення
2.	Music	Id	Текст	Первинний ключ
		Name	Текст	Найменування музичного супроводження
		Path	Текст	Шлях до файлу музичного супроводження
3.	UserStatus	Id	Текст	Первинний ключ
		MaxAmountClicksMouse	Ціле число	Верхня межа кількості натиснутих клавiш миші за визначений час
		MinAmountClicksMouse	Ціле число	Нижня межа кількості натиснутих клавiш миші за визначений час
		MaxAmountClicksButton	Ціле число	Верхня межа кількості натиснутих клавiш клавіатури за визначений час
		MinAmountClicksButton	Ціле число	Нижня межа кількості натиснутих клавiш клавіатури за визначений час
4.	Rating	Id	Текст	Первинний ключ
		MusicId	Текст	Зовнішній ключ
		RatingValue	Ціле число	Оцінка рекомендованого музичного супроводження
		UserId	Текст	Зовнішній ключ
		UserStatus	Текст	Зовнішній ключ

БД була створенна за допомогою Entity Framework Core.

Entity Framework Core (EF Core) являє собою об'єктно-орієнтовану, легковажну і розширяемую технологію від компанії Microsoft для доступу до даних. EF Core є ORM-інструментом (object-relational mapping - відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати базами даних, але є більш високий рівень

абстракції: EF Core дозволяє абстрагуватися від самої бази даних і її таблиць і працювати з даними незалежно від типу сховища. Якщо на фізичному рівні ми оперуємо таблицями, індексами, первинними і зовнішніми ключами, але на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами[21].

Було обрано підхід code first. Тобто за допомогою створення класів сутностей генерується сама БД.

### 3.4 Проведення досліджень

Дослідження точності та коректності системи буде проведене за допомогою генерації оцінки користувачів та імітації вхідних даних стану користувача.

Було згенеровано 2 стани користувача, також були згенеровані оцінки користувачів до певних музичних супроводжень. Так як в системі ідентифікатором є GUID, для наочності він був замінений на числовий еквівалент ідентифікатора. Було згенеровано 2 групи користувачів по 100 користувачів в кожній з протилежними музичними вподобаннями в різних станах. Оцінки першої групи користувачів надано в таблиці 3.2.

Таблиця 3.2 – Оцінки першої групи користувачів

Ідентифікатор стану	Ідентифікатор музичного супроводження	Оцінка користувачів (1–10)
1	1	10
1	2	8
1	3	4
1	4	1
2	1	1
2	2	4
2	3	8
2	4	10

Оцінки другої групи користувачів надано в таблиці 3.3.

Таблиця 3.3 – Оцінки другої групи користувачів

Ідентифікатор стану	Ідентифікатор музичного супроводження	Оцінка користувачів (1–10)
1	1	1
1	2	4
1	3	8
1	4	10
2	1	10
2	2	8
2	3	4
2	4	1

Було згенеровано користувачів, для яких буде надаватися музичне супроводження. У користувача «А» оцінки схожі до оцінок першої групи користувачів, а у користувача «Б» - до другої.

Таблиця 3.4 – Оцінки користувача «А»

Ідентифікатор стану	Ідентифікатор музичного супроводження	Оцінка
1	2	9
1	3	3
2	2	3
2	3	9

Таблиця 3.5 – Оцінки користувача «Б»

Ідентифікатор стану	Ідентифікатор музичного супроводження	Оцінка
1	2	3
1	3	9
2	2	9
2	3	3

Після чого, було земульовано використання системи користувачем «А». Було виявлено, що система надала наступні рекомендації в залежності від стану:

- стан «1» - рекомендоване музичне супроводження «1» та «2»;

- стан «2» - рекомендоване музичне супроводження «3» та «4».

Також було земульовано використання системи користувачем «Б». Було виявлено, що система надала наступні рекомендації в залежності від стану:

- стан «1» - рекомендоване музичне супроводження «4» та «3»;

- стан «2» - рекомендоване музичне супроводження «2» та «1».

## ВИСНОВКИ

Алгоритми рекомендаційних систем можна розділити на дві великі парадигми: колабораційні підходи які базуються лише на матриці взаємодії між елементом-користувачем та підходах на основі вмісту, які використовують попередня інформація про користувачів та / або предмети.

Методи спільної роботи на основі пам'яті не передбачають жодної прихованої моделі і мають тоді низьку упередженість, але велику дисперсію; підходи, що базуються на моделях, передбачають приховану модель взаємодії, яка потребує вивчення уявлень як користувачів, так і предметів з нуля і, отже, має більший ухил, але меншу дисперсію; методи, що базуються на вмісті, передбачають приховану модель побудови навколо користувачів та / або елементів, явно заданих, і, таким чином, мають найвищу упередженість і найменшу дисперсію.

Рекомендуючі системи стають все більш важливими у всіх аспектах людського життя.

Проведене дослідження створеної системи було досить поверхневим. Для більш точних рекомендацій система потребує вдосконалення, а саме:

- надати додаткові дані користувача для більш точного визначення його стану, наприклад пульс користувача та місце розташування погляду на моніторі;
- в різних відеоіграх користувач надає різні вхідні дані, тому надати можливість штучному інтелекту самому визначати стан користувача, а не визначати його заздалегідь. Також ця властивість вказує на те, що в системі не вистачає залежності від відеоігри.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. D. Jannach, M. Zanker, A. Felfernig, G. Friedrich “Recommender systems – an introduction” Cambridge University Press (2010)
2. J.A. Konstan, J. Riedl “Recommender systems: from algorithms to user experience”.
3. C. Pan, W. Li “Research paper recommendation with topic analysis”.
4. Pu P, Chen L, Hu R. “A user-centric evaluation framework for recommender systems”.
5. Schedl M, Zamani H, Chen CW, Deldjoo Y, Elahi M. Current challenges and visions in music recommender systems research. *Int J Multi Inform Ret.* (2018)
6. Zhernova P., Deyneko A., Deyneko Z., Pliss I., Ahafonov V. “Data Stream Clustering in Conditions of an Unknown Amount of Classes”. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) *Advances in Computer Science for Engineering and Education. ICCSEEA 2018. Advances in Intelligent Systems and Computing*, vol 754. Springer, Cham.
7. Zhernova P., Deyneko A., Bodyanskiy Ye., Riepin V. Adaptive Kernel Data Streams Clustering Based on Neural Networks Ensembles in Conditions of Uncertainty About Amount and Shapes of Clusters. *IEEE Second International Conference on Data Stream Mining & Processing*, 21-25 August, 2018, Lviv, Ukraine. P. 7-12.
8. Batmaz Z, Yurekli A, Bilge A, Kaleli C. A review on deep learning for recommender systems: challenges and remedies. *Artif Intell Rev.* (2018)
9. Oramas S, Nieto O, Sordo M, Serra X. A deep multimodal approach for cold-start music recommendation. In: *Proceedings of the 2Nd Workshop on Deep Learning for Recommender Systems. DLRS 2017.* New York, NY: ACM (2017)
10. Sachdeva N, Gupta K, Pudi V. Attentive neural architecture incorporating song features for music recommendation. In: *Proceedings of the 12th ACM Conference on Recommender Systems. RecSys '18.* New York, NY: ACM (2018)
11. Goodfellow I, Bengio Y, Courville A. *Deep Learning.* Cambridge, MA: MIT Press (2016).
12. Похилько, А. Ф. CASE-технология моделирования процессов с использованием средств BPWin и ERWin: учебное пособие / А. Ф. Похилько, И. В. Горбачев. – Ульяновск: УлГТУ, 2008. – 120 с.

13. Столбовський Д.Н. Основи розробки Інтернет додатків і Web сервісів на основі ASP.NET - ГТУ, 2008. – 256 с.
14. Бахтизин В.В., Глухова Л.А. Структурный анализ и моделирование в среде CASE-средства VPwin: Учебное пособие по курсу “Технология проектирования программ” для студ. спец. 40 01 01 “Программное обеспечение информационных технологий”/ В.В. Бахтизин, Л.А. Глухова. – Мн.: БГУИР, 2002. – 44 с.: ил. ISBN 985-444-401-5.
15. Долженкова М.Л. Использование CASE-средств для проектирования информационных систем: Учебное пособие / М.Л. Долженкова, О.В. Караева – Киров: Изд-во ВятГУ, 2002. – 73 с.
16. Моделирование потоков данных. Диаграммы DFD - Режим доступа URL: [http://b-c-group.ru/?page\\_id=103](http://b-c-group.ru/?page_id=103)
17. Розенберг Д.; Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов / Пер. с англ. – М.: ДМК Пресс. – 160 с.
18. Хассан Гома UML. Проектирование систем реального времени, параллельных и распределенных приложений / Хассан Гома ; пер. А. А. Слинкин. - Москва : ДМК Пресс, 2016. – 698 с.
19. С# для начинающих - Режим доступа URL: <https://itcloud.academy/blog/programming/c-dlya-nachinayushikh-pochemu-stoit-vybrat-imenno-etot-yazyk/>
20. MS SQL Server и T-SQL | Введение - Режим доступа URL: <https://metanit.com/sql/sqlserver/1.1.php>
21. Entity Framework Core | Введение - Режим доступа URL: <https://metanit.com/sharp/entityframeworkcore/1.1.php>