

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

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему: «Розроблення застосунку формування музичних рекомендацій з використанням технологій ШІ»

здобувача Устименка Олександра Олександровича

(ПІБ)

_____ (Підпис)

Науковий керівник:

д.е.н., професор

_____ Лондар С.Л.

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

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

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

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)
освітньо-професійної програми

Помазун О.М.
“ ____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

Тішков Б.О.

“ ____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти **Устименка Олександра Олександровича**

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи

на тему: «Розроблення застосунку формування музичних рекомендацій з використанням технологій ШІ»

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

Кваліфікаційна бакалаврська робота виконується на матеріалах отриманих під час навчання, навчальних посібників, матеріалів з мережі Інтернет, а також зібраного фактичного матеріалу з обраної теми дослідження під час виконання курсових робіт.

План кваліфікаційної бакалаврської роботи

Розділ I. Характеристика та аналіз предметної галузі

Розділ II. Розробка вимог і моделювання інформаційної системи

Розділ III. Проєктування та реалізація компонентів системи

Кваліфікаційна бакалаврська робота присвячена розробці сучасного застосунку для формування музичних рекомендацій на основі технологій штучного інтелекту. Метою проєкту є створення інноваційного інструменту, який дозволяє користувачам отримувати персоналізовані музичні добірки з урахуванням їхніх індивідуальних вподобань, настрою та музичної активності.

Робота складається з трьох взаємопов'язаних розділів. У першому розділі детально проаналізовано предметну область, досліджено підходи до побудови рекомендаційних систем у музичній індустрії, а також проведено огляд існуючих рішень та визначено об'єкт дослідження — інтелектуальну систему музичних рекомендацій.

У другому розділі сформульовано вимоги до системи, описано архітектуру застосунку, обґрунтовано вибір алгоритмів машинного навчання, що використовуються для аналізу музичних уподобань користувача.

Третій розділ містить опис інформаційного, технічного та програмного забезпечення застосунку. Подано результати реалізації MVP-продукту, включно з інтерфейсом користувача, основною функціональністю та інтеграцією із зовнішніми API для обробки музичних даних.

Особливістю проекту є використання нейромереж для аналізу прослуханих треків та побудови персоналізованих рекомендацій у реальному часі. Такий підхід дозволяє значно підвищити точність системи та рівень задоволеності користувачів.

У висновках подано рекомендації щодо подальшого розвитку та впровадження застосунку, а також оцінено ефективність використання ШІ в системах музичних рекомендацій.

Конкретні завдання, які потрібно виконати для досягнення поставленої мети:

1. У розділі I навести характеристику предметної галузі та об'єкта дослідження, проаналізувати сучасний стан та тенденції розвитку музичних рекомендаційних систем, визначити ключові технології штучного інтелекту, необхідні для реалізації нової системи, а також проаналізувати літературні джерела та практичний досвід розробки таких систем.
2. У розділі II визначити основні вимоги до інформаційної системи та обґрунтувати методи її розробки. При моделюванні системи використовувати сучасні підходи до автоматизованого машинного навчання та аналізу даних. Розробити постановку задачі, алгоритм та інформаційну модель системи, враховуючи вимоги системного підходу та забезпечення повної функціональності.
3. У розділі III виконати проектування та розробку інформаційної системи, зокрема її інформаційного, технічного та програмного забезпечення. Реалізувати систему рекомендацій для музичної платформи, визначивши можливості та доцільність її застосування в реальних умовах музичних сервісів.

**Завдання підготував
науковий керівник**

Лондар Сергій Леонідович

« 10 » березня 2025 р.

**Завдання одержав
здобувач**

Устименко Олександр

Олександрович

« 10 » березня 2025 р..

ВІДГУК

на кваліфікаційну бакалаврську роботу здобувача навчально-наукового інституту «Інститут інформаційних технологій в економіці» освітньо-професійної програми «Комп'ютерні науки» **Устименка Олександра Олександровича** на тему «Розроблення застосунку формування музичних рекомендацій з використанням технологій ШІ»

Тематика дипломної роботи є досить актуальною в умовах розвитку цифрових технологій у сфері музичних платформ. Зростання попиту на персоналізовані музичні рекомендації потребує впровадження інноваційних, зручних та безпечних інформаційних рішень, що відповідають сучасним вимогам користувачів. Обрана тема повністю відповідає поточним трендам розвитку ІТ у музичних сервісах.

Кваліфікаційна бакалаврська робота вирізняється логічною структурованістю, достатньо повним охопленням усіх етапів проєктування інформаційної системи від постановки задачі до побудови ефективної моделі інформаційної системи для формування музичних рекомендацій. Автор продемонстрував розуміння предметної галузі, вміло поєднав теоретичні знання з практичними інструментами проєктування. У проєкті враховано не лише функціональні, але й нефункціональні вимоги, що свідчить про комплексний підхід до розробки.

Автор самостійно спроектував архітектуру системи, розробив інформаційну модель, реалізував ключові компоненти програмного забезпечення, включаючи модуль надання музичних рекомендацій, генерацію персоналізованих списків для користувачів, інтеграцію з API для аналізу музичних смаків. Робота містить авторські діаграми, моделі та програмні реалізації.

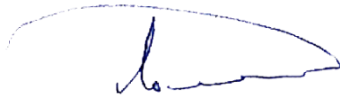
Теоретичні висновки є обґрунтованими, базуються на ґрунтовному аналізі джерел та сучасних підходів до створення рекомендаційних систем. Практичні рекомендації щодо впровадження системи можуть бути корисними для музичних платформ, що свідчить про прикладну спрямованість дослідження.

Як і при виконанні будь якого проекту є й певні недоліки. Зокрема автором не надано розрахунків щодо економічного аналізу впровадження ІС у реальних умовах. Реалізовано лише фрагмент БД у середовищі Postgres Server. Не приведено результати модульного, системного та інтеграційного тестування серверної частини додатку.

Проте ці зауваження не зменшують теоретичної та практичної цінності роботи.

Вважаю, що бакалаврська робота Устименка Олександра Олександровича є завершеним, самостійним дослідженням, яке відповідає вимогам до бакалаврських проєктів у галузі комп'ютерних наук щодо обсягу, структури та змісту, демонструє достатній рівень професійної підготовки здобувача. Робота може бути допущена до захисту.

**Науковий керівник,
д.е.н., професор**



Сергій ЛОНДАР

РЕЦЕНЗІЯ

на кваліфікаційну бакалаврську роботу
здобувача вищої освіти

Устименка Олександра Олександрович

Тема «Розроблення застосунку формування музичних рекомендацій з використанням технологій ШІ»

Обрана тема є надзвичайно актуальною, зважаючи на швидкий розвиток технологій штучного інтелекту та персоналізованих сервісів у сфері цифрових медіа. В умовах зростаючого попиту на інтелектуальні системи рекомендацій у музичній індустрії, створення застосунку, який здатен аналізувати уподобання користувачів і формувати персоналізовані музичні добірки, має високу прикладну цінність. Такий продукт є особливо доречним як для користувачів, так і для музичних платформ, адже сприяє підвищенню залученості аудиторії та оптимізації взаємодії з контентом.

Робота характеризується якісною дослідницькою базою та має досить чітку структуру. Автором проведено ґрунтовний огляд сучасних підходів до реалізації подібних систем — як класичних (на основі контенту та колаборативної фільтрації), так і новітніх, побудованих на основі нейронних мереж. Продемонстровано вміння інтегрувати моделі машинного навчання в реальний застосунок, зокрема використано алгоритми обробки ознак, кластеризації, векторизації та аналізу поведінки користувачів.

До позитивних особливостей роботи слід віднести створення повноцінного прототипу музичного застосунку, який демонструє інтеграцію користувацького інтерфейсу з алгоритмами ШІ. Робота містить якісне технічне обґрунтування вибору архітектури системи, використання актуального стеку технологій: Python, FastAPI, PostgreSQL, React Native, TensorFlow/Scikit-learn. Також варто відзначити реалізацію функціоналу рекомендацій у реальному часі, інтерфейс користувача з можливістю оцінки композицій, а також дотримання принципів безпеки та захисту даних.

Робота виконана на високому рівні, однак в подальшому доцільно розширити тестування моделей з використанням більш об'ємних датасетів і впровадити автоматизовані тести як на рівні бекенду, так і фронтенду. Крім того, архітектурна частина потребує більш детального опису сервісної взаємодії компонентів.

Загалом практична цінність роботи безсумнівна — застосунок може бути використаний як стартовий продукт для подальшого розвитку персоналізованих музичних сервісів або як проєкт у портфоліо розробника. Запропоновані автором напрями розвитку — інтеграція з музичними платформами, вдосконалення моделей на основі deep learning, масштабування застосунку — є обґрунтованими та демонструють розуміння ринкових вимог.

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Розроблення застосунку формування музичних
рекомендацій з використанням технологій ШІ»
Київ: кафедра інформаційних систем в економіці, 2025 р.

Кваліфікаційна бакалаврська робота присвячена актуальній темі розробки застосунку для формування музичних рекомендацій з використанням технологій штучного інтелекту, який покликаний полегшити процес пошуку та отримання персоналізованих музичних рекомендацій для користувачів. Застосунок надає доступ до актуальної інформації про музичні композиції, дозволяючи користувачам швидко отримувати рекомендації відповідно до їх музичних переваг в реальному часі.

Кваліфікаційна бакалаврська робота складається з трьох розділів, логічно пов'язаних між собою.

У ході виконання першого розділу було детально описано предметну область та визначено об'єкт дослідження – процес автоматизованої розробки та впровадження системи формування музичних рекомендацій, сфокусований на покращенні персоналізованого досвіду користувачів і оптимізації процесів керування музичним контентом у межах платформи.

У розділі 2 сформульовано конкретні вимоги до системи, поставлено задачу та створено модель інформаційної системи.

У третьому розділі описано інформаційне, технічне та програмне забезпечення для системи формування музичних рекомендацій. Також, представлено результати реалізації програмного продукту.

Унікальною особливістю проєкту є інтеграція з зовнішніми API для аналізу музичних вподобань та контенту. Це дозволяє користувачам отримувати музичні рекомендації без необхідності покидати застосунок, отримуючи всі необхідні дані в одному місці.

У висновках надано рекомендації щодо доцільності розробки та впровадження системи музичних рекомендацій на платформах для прослуховування музики.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота присвячена розробці програмного застосунку для формування персоналізованих музичних рекомендацій з використанням технологій штучного інтелекту. Система покликана забезпечити користувачів інтуїтивно зрозумілим та ефективним інтерфейсом для отримання рекомендацій на основі їхніх музичних переваг. Завдяки впровадженню методів машинного навчання, система формує музичні добірки, які відповідають смакам користувачів, полегшуючи процес відкриття нової музики в умовах великого обсягу контенту.

Ключові слова: музичні рекомендації, штучний інтелект, машинне навчання, персоналізація, музичний контент, Python, рекомендаційна система.

Об'єкт дослідження – розробка застосунку музичних рекомендацій з використанням технологій ШІ.

Предмет дослідження – є методи і алгоритми розробки застосунку формування музичних рекомендацій

Мета роботи полягає в розробці програмного застосунку для формування музичних рекомендацій з використанням технологій штучного інтелекту, здатного аналізувати історію прослуховувань, жанрові вподобання та інші релевантні параметри для надання точних і персоналізованих музичних рекомендацій.

Структура роботи:

1. **Вступ:** Огляд тенденцій розвитку музичної індустрії та необхідності впровадження персоналізованих музичних рекомендацій.
2. **Аналіз предметної області:** Опис еволюції рекомендаційних систем, огляд технологій ШІ, які використовуються для створення музичних рекомендацій, а також аналіз існуючих аналогів.
3. **Проектування системи рекомендацій:** Визначення вимог до системи, постановка завдання, моделі та алгоритми надання рекомендацій.
4. **Реалізація інформаційної системи:** Вибір технологій для реалізації, архітектура системи, програмна реалізація основних компонентів, приклади використання системи.
5. **Висновки:** Оцінка результатів роботи, можливі напрямки подальшого розвитку системи, впровадження складніших моделей ШІ для покращення рекомендацій.

Обрані технології для реалізації системи включають мову програмування Python, фреймворк Flet для побудови графічного інтерфейсу, базу даних SQLite для зберігання даних користувачів і історії прослуховувань. Моделі рекомендацій були побудовані на основі алгоритмів K-ближчих сусідів та контентної фільтрації, що дозволяють забезпечити високу точність персоналізованих рекомендацій.

Практичне значення роботи полягає у створенні системи, яка дозволяє користувачам музичних платформ отримувати якісні персоналізовані рекомендації, що сприяють відкриттю нових композицій і підвищенню рівня задоволення від користування платформою.

Рік виконання дипломної роботи – 2025.

Рік захисту дипломної роботи – 2025.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- **API** – Application Programming Interface
- **IT** – Інформаційні технології
- **PDF** – Portable Document Format
- **QR** – Quick Response code
- **UI** – User Interface
- **АРМ** – Автоматизоване робоче місце
- **БД** – База даних
- **ІЗ** – Інформаційне забезпечення
- **ІС** – Інформаційна система
- **ПЗ** – Програмне забезпечення
- **СУБД** – Система управління базами даних

ВСТУП

У сучасному цифровому світі музична індустрія переживає справжню революцію. За останні два десятиліття спосіб споживання музичного контенту кардинально змінився – від фізичних носіїв до потокових сервісів, що надають миттєвий доступ до мільйонів композицій. Згідно з даними Spotify, їхня платформа містить понад 100 мільйонів треків, а щодня додається близько 100 тисяч нових композицій. Такий величезний обсяг музичного контенту створює парадокс вибору для користувачів – маючи доступ до практично необмеженої кількості музики, людям стає все складніше знаходити композиції, які відповідають їхнім смакам та настрою.

Традиційні методи пошуку музики, засновані на жанрах, виконавцях або популярності, виявляються недостатніми для ефективної навігації у цьому музичному океані. Користувачі часто втрачаються серед безлічі варіантів, що призводить до зниження задоволення від музичного досвіду та обмеження їхніх музичних горизонтів. Саме тому розробка інтелектуальних систем музичних рекомендацій стала одним з найважливіших напрямків розвитку музичних платформ.

Вибір даної теми дослідження обумовлений кількома ключовими факторами. По-перше, це практична необхідність вирішення реальної проблеми музичного перенасичення, з якою стикаються мільйони користувачів щодня. По-друге, це можливість застосування сучасних технологій штучного інтелекту для створення персоналізованого музичного досвіду. По-третє, це перспективність даного напрямку як з комерційної, так і з наукової точки зору, оскільки якісні рекомендаційні системи є ключовим конкурентним фактором для музичних платформ.

Предметна галузь дослідження охоплює сферу інформаційних технологій, зокрема розробку рекомендаційних систем з використанням методів машинного навчання та штучного інтелекту. Дослідження перебуває на перетині кількох дисциплін: інформатики, математичного моделювання, музикознавства та психології сприйняття музики.

Мета дипломного проекту полягає у розробці програмного застосунку для формування персоналізованих музичних рекомендацій з використанням сучасних технологій штучного інтелекту, який здатен аналізувати музичні переваги користувачів та надавати релевантні рекомендації на основі комплексного аналізу музичних характеристик та поведінкових патернів.

Актуальність теми дослідження обумовлена кількома ключовими тенденціями сучасного цифрового світу:

Зростання обсягів музичного контенту. Щорічно у світі випускається понад 40 мільйонів нових музичних треків, що створює критичну потребу в інструментах для навігації у цьому контенті. Без ефективних рекомендаційних систем користувачі ризикують залишитися в межах вузького кола знайомої музики, втрачаючи можливості для музичних відкриттів.

Персоналізація цифрового досвіду. Сучасні користувачі очікують персоналізованого підходу в усіх сферах цифрового споживання. Музичні платформи, які не можуть запропонувати релевантні рекомендації, втрачають конкурентні позиції на ринку.

Розвиток технологій ШІ. Стрімкий прогрес у сфері машинного навчання, особливо в областях обробки природної мови та аналізу часових рядів, відкриває нові можливості для створення більш точних та інтелектуальних рекомендаційних систем.

Економічне значення. Якісні рекомендаційні системи прямо впливають на економічні показники музичних платформ: збільшують час, проведений користувачами в застосунку, підвищують рівень утримання користувачів та сприяють виявленню нових талантів.

На сьогоднішній день рекомендаційні системи є невід'ємною частиною всіх провідних музичних платформ. Spotify використовує алгоритм Discover Weekly, який щотижня генерує персональні плейлисти для понад 200 мільйонів користувачів. Apple Music застосовує гібридний підхід, поєднуючи машинне навчання з експертними рекомендаціями музичних кураторів.

YouTube Music використовує дані про перегляди відео для формування музичних рекомендацій.

Розроблюваний застосунок призначений для широкого кола користувачів цифрових музичних сервісів, які прагнуть розширити свої музичні горизонти та отримувати персоналізовані рекомендації. Цільова аудиторія включає як звичайних слухачів музики, так і музичних ентузіастів, які активно шукають нові композиції та виконавців.

Практичне значення проекту:

1. Для користувачів: забезпечення персоналізованого музичного досвіду, економія часу на пошук нової музики, розширення музичних горизонтів
2. Для музичної індустрії: підвищення споживання музичного контенту, сприяння відкриттю нових талантів, оптимізація промоційних стратегій
3. Для розробників: демонстрація практичного застосування сучасних технологій ШІ, створення основи для подальших досліджень у сфері рекомендаційних систем
4. Для наукової спільноти: внесок у розвиток методів персоналізації та рекомендаційних алгоритмів

Реалізація даного проекту дозволить не лише вирішити практичні завдання створення ефективної системи музичних рекомендацій, але й продемонструє можливості інтеграції різних технологій штучного інтелекту для створення комплексного рішення з високою практичною цінністю. Розроблений застосунок може служити основою для подальших досліджень та розробок у сфері персоналізаційних технологій.

Зміст	
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	10
ВСТУП	11
РОЗДІЛ 1	16
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	16
1.1 Еволюція рекомендаційних систем	18
1.2 Огляд технологій ШІ у рекомендаційних сервісах	26
1.3 Аналіз аналогів	31
РОЗДІЛ 2	36
ПРОЕКТУВАННЯ СИСТЕМИ	36
2.1 Визначення вимог до інформаційної системи	37
2.2 Постановка та алгоритм розв’язання задачі	46
2.3 Моделі та алгоритми надання рекомендацій	53
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	58
3.1 Інформаційне забезпечення	59
3.1.1 Загальна характеристика інформаційного забезпечення (ІЗ)	59
3.1.2 Організація збору і передання первинної інформації	61
3.1.3 Побудова системи класифікації та кодування	63
3.1.4 Проектування форм первинних документів, машинограм та відеокадрів	64
.....	66
3.1.5 Структура інформаційних масивів	66
3.1.6 Вибір СУБД	68
3.1.8 Даталогічна модель бази (сховища) даних	70

3.2 Архітектура системи.....	73
3.3 Опис програмної реалізації.....	76
3.4 Результати реалізації інформаційної системи.....	81
ВИСНОВКИ.....	87
ДОДАТКИ.....	92

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Створення ефективного застосунку для формування музичних рекомендацій з використанням технологій штучного інтелекту вимагає глибокого розуміння предметної області та сучасного стану розвитку рекомендаційних систем. Рекомендаційні системи є однією з найдинамічніших галузей інформаційних технологій, що постійно еволюціонує під впливом нових алгоритмічних підходів, зростаючих обсягів даних та змінних потреб користувачів.

У контексті музичних сервісів рекомендаційні системи відіграють особливо важливу роль, оскільки музичні переваги є надзвичайно суб'єктивними та можуть залежати від безлічі факторів: настрою, часу доби, активності, соціального контексту та навіть погодних умов. Це створює унікальні виклики для розробників, які повинні враховувати не лише технічні аспекти алгоритмів, але й психологічні та соціокультурні особливості музичного сприйняття.

Перший розділ дипломної роботи присвячений комплексному аналізу предметної області з метою формування теоретичного підґрунтя для подальшої розробки застосунку. Дослідження охоплює три ключові напрямки, які є критично важливими для успішної реалізації проекту.

Спочатку розглядається еволюція рекомендаційних систем від найпростіших методів статистичної фільтрації до сучасних гібридних підходів, що використовують глибоке навчання та нейронні мережі. Такий історичний огляд дозволяє зрозуміти логіку розвитку галузі, виявити основні віхи та тенденції, а також усвідомити переваги та обмеження різних поколінь рекомендаційних алгоритмів.

Далі проводиться детальний аналіз сучасних технологій штучного інтелекту, що застосовуються у рекомендаційних сервісах. Особлива увага приділяється методам машинного навчання, які довели свою ефективність у задачах персоналізації: від класичних підходів колаборативної та контентної

фільтрації до складних архітектур глибокого навчання, включаючи рекурентні нейронні мережі, автоенкодери та трансформери. Розглядаються також специфічні методи обробки аудіоданих, алгоритми виявлення музичних патернів та підходи до аналізу часових послідовностей поведінки користувачів.

Завершує розділ всебічний аналіз існуючих аналогів – провідних музичних платформ та рекомендаційних сервісів. Цей аналіз включає не лише технічні аспекти реалізації рекомендаційних алгоритмів, але й архітектурні рішення, підходи до збору та обробки даних, методи оцінки якості рекомендацій та стратегії взаємодії з користувачами. Порівняльний аналіз дозволяє виявити сильні та слабкі сторони існуючих рішень, а також сформулювати вимоги до власного застосунку з урахуванням найкращих практик галузі.

Результати аналізу предметної області формують теоретичне підґрунтя для обґрунтованого вибору методів та технологій, що будуть використані при розробці застосунку, а також дозволяють сформулювати технічні вимоги та критерії оцінки ефективності майбутньої системи.

1.1 Еволюція рекомендаційних систем

Концепція автоматичних рекомендацій виникла задовго до появи цифрових технологій. У традиційній торгівлі продавці завжди намагалися рекомендувати товари, що найкраще відповідають потребам покупців, базуючись на власному досвіді та знанні клієнтів. Однак формалізація цього процесу та його автоматизація стали можливими лише з розвитком обчислювальної техніки та накопиченням цифрових даних про поведінку користувачів.

Перші спроби створення автоматичних рекомендаційних систем датуються початком 1990-х років, коли дослідники почали експериментувати з методами фільтрації інформації. Термін "колаборативна фільтрація" був вперше введений в 1992 році Дугласом Терлі та його колегами з Xerox PARC у контексті системи Tapestry, призначеної для фільтрації електронної пошти. Ця система дозволяла користувачам створювати запити типу "показати мені документи, які Сьюзен прочитала та оцінила позитивно", що стало прототипом сучасних колаборативних підходів.

Справжнім проривом у розвитку рекомендаційних систем стала поява системи GroupLens у 1994 році, розробленої в Університеті Міннесоти. Ця система була призначена для рекомендації статей у групах новин Usenet і вперше продемонструвала ефективність автоматичної колаборативної фільтрації на великих масивах даних. GroupLens використовувала алгоритм найближчих сусідів для знаходження користувачів зі схожими інтересами та генерації рекомендацій на основі їхніх оцінок.

Паралельно розвивалися підходи, засновані на аналізі вмісту документів. У 1995 році була представлена система Fab, яка поєднувала колаборативну та контентну фільтрацію для рекомендації веб-сторінок. Цей гібридний підхід дозволив подолати деякі обмеження чисто колаборативних методів, зокрема проблему холодного старту для нових користувачів та елементів без рейтингів.

Розвиток колаборативної фільтрації у другій половині 1990-х років характеризувався активними дослідженнями алгоритмів прогнозування рейтингів. Основна ідея полягала у тому, що користувачі зі схожими смаками в минулому матимуть схожі переваги і в майбутньому. Це припущення дозволило розробити ефективні алгоритми, засновані на обчисленні схожості між користувачами або між елементами.

Методи колаборативної фільтрації поділяються на два основні підтипи: memory-based (засновані на пам'яті) та model-based (засновані на моделях). Memory-based підходи включають алгоритми найближчих сусідів як для користувачів (user-based), так і для елементів (item-based). User-based колаборативна фільтрація знаходить користувачів зі схожими історіями рейтингів та рекомендує елементи, які сподобалися схожим користувачам. Item-based підхід, запропонований Amazon у 2001 році, аналізує схожість між елементами на основі рейтингів, які вони отримали від користувачів.

Model-based підходи використовують машинне навчання для створення моделей користувацьких переваг. До них належать методи кластеризації, байєсівські класифікатори, методи зменшення розмірності та латентні факторні моделі. Особливе місце серед цих методів займає матричне розкладання (matrix factorization), яке дозволяє виявляти приховані фактори, що впливають на користувацькі переваги.

Контентна фільтрація базується на аналізі характеристик елементів та побудові профілів користувацьких переваг. Цей підхід особливо ефективний у доменах з багатими описовими характеристиками, таких як текстові документи, фільми або музика. Основна перевага контентної фільтрації полягає у здатності рекомендувати нові елементи, навіть якщо вони ще не отримали жодного рейтингу від користувачів.

У музичній сфері контентна фільтрація використовує різноманітні характеристики композицій: метадані (жанр, виконавець, рік випуску), текстові описи, та найважливіше – аудіохарактеристики, отримані шляхом аналізу самого аудіосигналу. Розвиток методів обробки аудіосигналів

дозволив виділяти такі характеристики, як тембр, ритм, гармонія, що значно підвищило якість музичних рекомендацій.

Профіль користувача в контентній фільтрації представляється як вектор ваг характеристик, що відображає переваги користувача. Цей профіль будується на основі аналізу елементів, які користувач оцінив позитивно, та постійно оновлюється з надходженням нових даних. Для побудови та оновлення профілів використовуються різноманітні методи машинного навчання, від простих лінійних моделей до складних нейронних мереж.

Окремим напрямком розвитку стали знання-орієнтовані (knowledge-based) рекомендаційні системи, які використовують явні знання про домен для генерації рекомендацій. Ці системи особливо корисні для складних товарів або послуг, де рішення про покупку вимагає врахування багатьох факторів та обмежень.

У музичній сфері знання-орієнтовані підходи можуть використовувати музикознавчі знання про жанри, стилі, історичні періоди, впливи між виконавцями тощо. Такі системи здатні пояснювати свої рекомендації, що підвищує довіру користувачів та дозволяє їм краще зрозуміти логіку системи.

Початок 2000-х років ознаменувався масовим впровадженням рекомендаційних систем в електронній комерції. Компанії Amazon, Netflix та eBay стали піонерами в застосуванні рекомендацій для підвищення продажів та покращення користувацького досвіду. Цей період характеризувався переходом від академічних досліджень до промислових реалізацій, що поставило нові виклики щодо масштабованості, швидкодії та надійності систем.

Amazon революціонізувала сферу рекомендацій, запровадивши item-based колаборативну фільтрацію та алгоритм "Customers who bought this item also bought". Ця система стала настільки успішною, що за оцінками компанії, рекомендації генерують близько 35% від загального доходу Amazon. Ключовими факторами успіху стали не лише алгоритмічні інновації, але й

ефективна інтеграція рекомендацій у користувацький інтерфейс та бізнес-процеси.

Netflix запустила свій онлайн-сервіс у 1999 році та швидко усвідомила важливість персоналізації для утримання користувачів. Система рекомендацій Netflix базувалася на рейтингах фільмів від користувачів та використовувала складні алгоритми колаборативної фільтрації. У 2006 році компанія оголосила конкурс Netflix Prize з призовим фондом у 1 мільйон доларів за покращення точності рекомендацій на 10%. Цей конкурс став каталізатором розвитку нових алгоритмічних підходів та привернув увагу наукової спільноти до проблем рекомендаційних систем.

Музична індустрія також почала активно використовувати рекомендаційні технології. Last.fm, запущена у 2003 році, стала однією з перших платформ, що застосувала колаборативну фільтрацію для музичних рекомендацій. Система аналізувала історію прослуховування користувачів та рекомендувала нові композиції на основі схожості музичних смаків. Pandora, запущена у 2005 році, революціонізувала підхід до музичних рекомендацій, використовуючи Music Genome Project – детальну систему класифікації музики за сотнями музичних характеристик.

Розвиток рекомендаційних систем супроводжувався виявленням та поступовим вирішенням ряду фундаментальних проблем, які обмежували ефективність ранніх реалізацій.

Матриці рейтингів у реальних системах зазвичай є надзвичайно розрідженими – користувачі оцінюють лише невелику частку доступних елементів. Це створює труднощі для алгоритмів колаборативної фільтрації, які потребують достатньої кількості спільних рейтингів для обчислення схожості між користувачами або елементами.

Для вирішення цієї проблеми були розроблені різноманітні підходи. Методи зменшення розмірності, такі як сингулярне розкладання (SVD) та негативне матричне розкладання (NMF), дозволили виявляти приховані паттерни в розріджених даних. Регуляризовані моделі допомогли уникнути

перенавчання на обмежених даних. Інтеграція неявного зворотного зв'язку, такого як час перегляду, кліки, покупки, значно збагатила дані про користувацькі переваги.

Холодний старт виникає для нових користувачів або нових елементів, які не мають достатньої історії взаємодій для генерації якісних рекомендацій. Ця проблема особливо актуальна для швидко зростаючих платформ, де постійно з'являються нові користувачі та контент.

Для нових користувачів були розроблені стратегії активного навчання, які дозволяють швидко зібрати інформацію про переваги через цілеспрямовані запитання або рекомендації популярних елементів з різних категорій. Демографічна інформація та дані з соціальних мереж також використовуються для створення початкових профілів користувачів.

Для нових елементів ефективним рішенням стало використання контентної інформації. У музичній сфері це означає аналіз аудіохарактеристик нових композицій для їх порівняння з існуючими треками. Гібридні підходи, що поєднують колаборативну та контентну фільтрацію, показали найкращі результати у вирішенні проблеми холодного старту.

Зі зростанням кількості користувачів та елементів традиційні алгоритми стали неефективними через високу обчислювальну складність. Алгоритми найближчих сусідів мають квадратичну складність за кількістю користувачів або елементів, що робить їх непрактичними для великих систем.

Вирішення проблеми масштабованості потребувало розробки нових алгоритмічних підходів та архітектурних рішень. Model-based методи, особливо матричне розкладання, виявилися значно більш масштабованими. Розподілені обчислення та використання кластерів серверів дозволили обробляти великі обсяги даних. Техніки кешування та попереднього обчислення рекомендацій також суттєво покращили продуктивність систем.

Середина 2010-х років ознаменувалася революційними змінами в рекомендаційних системах завдяки успіхам глибокого навчання. Нейронні

мережі продемонстрували здатність моделювати складні нелінійні залежності в даних, що дозволило суттєво підвищити якість рекомендацій.

Початкові спроби застосування нейронних мереж у рекомендаційних системах зосереджувалися на заміні традиційних методів матричного розкладання нейронними еквівалентами. Автоенкодери показали ефективність у навчанні компактних представлень користувачів та елементів. Обмежені машини Больцмана (Restricted Boltzmann Machines) були успішно застосовані Netflix для моделювання користувацьких переваг.

Прорив стався з появою глибоких нейронних мереж, здатних навчати ієрархічні представлення даних. Deep Neural Networks for YouTube Recommendations, представлена у 2016 році, продемонструвала можливості глибокого навчання для рекомендацій у масштабах мільярдів користувачів та відеороликів. Система використовувала двоетапний підхід: спочатку candidate generation мережа відбирала сотні релевантних відео з мільйонів варіантів, а потім оцінювала мережа впорядковувала їх за релевантністю.

Важливим кроком стало усвідомлення того, що користувацькі переваги змінюються з часом, і рекомендаційні системи повинні враховувати темпоральну динаміку. Рекурентні нейронні мережі (RNN), особливо LSTM та GRU, виявилися ідеальними для моделювання послідовностей користувацьких дій та прогнозування наступних елементів інтересу.

Згорткові нейронні мережі (CNN) знайшли застосування в рекомендаційних системах для обробки структурованих даних, таких як зображення, текст та аудіо. У музичних рекомендаціях CNN використовуються для аналізу спектрограм аудіосигналів, дозволяючи автоматично виділяти релевантні акустичні характеристики.

Мультимодальні підходи, що поєднують різні типи даних (аудіо, текст, зображення, метадані), показали значні покращення якості рекомендацій. Наприклад, системи можуть аналізувати обкладинки альбомів, тексти пісень, аудіохарактеристики та метадані для формування комплексного уявлення про музичний контент.

Революційною подією стала поява механізмів уваги та архітектури трансформерів. Моделі, що базуються на увазі, дозволили системам фокусуватися на найбільш релевантних частинах вхідних даних, значно покращивши якість рекомендацій. BERT4Rec та SASRec продемонстрували ефективність трансформерів для послідовних рекомендацій.

Механізми уваги також дозволили підвищити інтерпретованість рекомендацій – система може показувати, на які елементи історії користувача вона звертала найбільшу увагу при генерації конкретної рекомендації.

Сучасні рекомендаційні системи все більше враховують контекстуальну інформацію: час, місцезнаходження, пристрій, соціальне оточення, настрої користувача. Контекстно-залежні рекомендації дозволяють адаптувати пропозиції до поточної ситуації користувача, що особливо важливо для музичних сервісів.

Зростаюча увага приділяється питанням справедливості рекомендацій та уникнення алгоритмічних упереджень. Системи можуть несправедливо дискримінувати певні групи користувачів або створювати "фільтр-бульбашки", обмежуючи різноманітність рекомендацій.

Розробляються методи для забезпечення справедливості рекомендацій відносно різних демографічних груп та для підтримки різноманітності пропозицій. Це включає техніки пост-обробки даних для коригування результатів та інтеграцію обмежень справедливості безпосередньо в процес оптимізації.

Важливою тенденцією стає розробка пояснюваних рекомендаційних систем, які можуть не лише генерувати рекомендації, але й пояснювати причини своїх рішень. Це підвищує довіру користувачів та дозволяє їм краще контролювати процес рекомендацій.

Зростаючі вимоги до приватності даних стимулюють розвиток федеративного навчання для рекомендаційних систем. Цей підхід дозволяє навчати моделі, не передаючи персональні дані на централізовані сервери, що забезпечує кращий захист приватності користувачів.

Сучасні рекомендаційні системи представляють собою складні багатокомпонентні рішення, що поєднують різноманітні алгоритмічні підходи, враховують множину факторів та балансують між точністю, різноманітністю, справедливістю та приватністю. Еволюція від простих статистичних методів до складних систем штучного інтелекту відображає не лише технологічний прогрес, але й зростаюче розуміння складності людських переваг та потреб персоналізації в цифровому світі.

1.2 Огляд технологій ШІ у рекомендаційних сервісах

Методи кластеризації та класифікації

Кластеризація користувачів та елементів дозволяє виявляти групи зі схожими характеристиками, що спрощує генерацію рекомендацій та підвищує масштабованість системи. K-means кластеризація часто використовується для групування користувачів за схожими музичними смаками. Ієрархічна кластеризація дозволяє створювати таксономії жанрів або стилів, що корисно для організації великих музичних каталогів.

Байєсівські методи, зокрема наївний байєсівський класифікатор, ефективно використовуються для моделювання ймовірностей користувацьких преференцій. Ці методи особливо корисні для інкорпорації попередніх знань про домен та для роботи з невизначеністю в даних.

Дерева рішень та випадкові ліси (Random Forest) знайшли застосування в рекомендаційних системах для моделювання складних взаємодій між характеристиками користувачів та елементів. Ці методи забезпечують високу інтерпретованість результатів, що важливо для розуміння причин конкретних рекомендацій.

Методи опорних векторів та ансамблі

Машини опорних векторів (Support Vector Machines, SVM) використовуються для класифікації користувацьких переваг та рангування елементів. SVM особливо ефективні при роботі з високорозмірними даними, такими як векторні представлення музичних треків, отримані з аудіоаналізу.

Ансамблеві методи, такі як AdaBoost, Gradient Boosting та XGBoost, дозволяють комбінувати прогнози кількох слабких моделей для отримання більш точних результатів. Ці методи часто використовуються на фінальних етапах рекомендаційних пайплайнів для тонкого налаштування рангування.

Багатошарові перцептрони та автоенкодера

Багатошарові перцептрони (Multi-Layer Perceptrons, MLP) стали основою для багатьох нейронних рекомендаційних систем. Вони дозволяють

моделювати складні нелінійні взаємодії між користувачами та елементами, що неможливо зробити за допомогою традиційних лінійних моделей.

Neural Collaborative Filtering (NCF) замінює скалярний добуток у традиційному матричному розкладанні багатопаровим перцептроном, який може навчати довільні функції від векторів користувачів та елементів. Це дозволяє моделювати складні паттерни взаємодій, які не можуть бути представлені простим лінійним розкладанням.

Автоенкодери показали високу ефективність у рекомендаційних системах, особливо для навчання компактних представлень користувачів та елементів. Denoising автоенкодери можуть ефективно обробляти зашумлені та неповні дані, що часто зустрічається в реальних системах.

Варіаційні автоенкодери (Variational Autoencoders, VAE) дозволяють генерувати нові рекомендації шляхом семплювання з навченого латентного простору. Це особливо корисно для створення різноманітних плейлистів або для генерації рекомендацій у слабо досліджених областях музичного простору.

Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) та їх удосконалені варіанти, такі як LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit), революціонізували підходи до моделювання послідовностей в рекомендаційних системах.

Рекомендації використовують RNN для моделювання послідовностей дій користувача в рамках однієї сесії. GRU4Rec стала піонерською роботою в цій області, демонструючи можливість прогнозування наступного елемента інтересу на основі послідовності попередніх дій.

Ієрархічні RNN дозволяють моделювати поведінку користувачів на різних часових масштабах: короткотермінові сесії та довготермінові переваги. Це особливо актуально для музичних рекомендацій, де користувачі можуть мати сталі жанрові переваги, але змінювати настрій протягом дня.

RNN засновані на увазі використовують механізми уваги для фокусування на найбільш релевантних частинах послідовності. Це дозволяє моделі виділяти ключові моменти в історії користувача, які найбільше впливають на поточні рекомендації.

Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) знайшли широке застосування в рекомендаційних системах, особливо для обробки структурованих даних, таких як зображення, аудіо та тексти.

У музичних рекомендаційних системах CNN використовуються для аналізу спектрограм аудіосигналів. Спектрограми представляють аудіо як двовимірні зображення, де одна вісь відповідає часу, а інша - частоті. CNN можуть автоматично виділяти релевантні акустичні паттерни, такі як мелодичні лінії, ритмічні структури та тембральні характеристики.

Текстові CNN аналізують тексти пісень, описи альбомів та інші текстові метадані для виділення семантичних характеристик музичного контенту. Це дозволяє враховувати не лише звучання музики, але й її смислове навантаження.

Трансформери та механізми уваги

Архітектура трансформерів, запропонована у роботі "Attention Is All You Need", стала революційною для багатьох задач обробки природної мови та знайшла застосування в рекомендаційних системах.

Self-attention механізми дозволяють моделі визначати важливість кожного елемента в послідовності відносно всіх інших елементів. У контексті музичних рекомендацій це означає можливість врахування складних взаємозв'язків між різними композиціями в історії прослуховування користувача.

BERT4Rec адаптує архітектуру BERT для задач послідовних рекомендацій, використовуючи двонаправлений контекст для кращого розуміння користувацьких переваг. SASRec (Self-Attentive Sequential

Recommendation) демонструє ефективність самоуваги для моделювання довгих послідовностей користувацьких дій.

Обробка аудіосигналів

Музичні рекомендаційні системи унікальні тим, що вони можуть аналізувати сам аудіоконтент, а не покладатися виключно на метадані або користувацьку поведінку. Це відкриває можливості для рекомендацій навіть для абсолютно нових композицій без жодної історії прослуховувань.

Мел-спектрограми є стандартним представленням аудіо для машинного навчання, оскільки вони наближено відтворюють частотне сприйняття людського слуху. CNN, що працюють з мел-спектрограмами, можуть виділяти такі характеристики, як тембр, ритм, гармонія та динаміка.

Мультимодальні архітектури

Сучасні музичні рекомендаційні системи інтегрують різні типи даних для формування комплексного уявлення про музичний контент. Мультимодальні архітектури дозволяють поєднувати аудіоаналіз, текстову інформацію, візуальні дані та метадані.

Fusion архітектури використовують різні стратегії для об'єднання інформації з різних модальностей. Early fusion об'єднує дані на рівні входу, middle fusion - на рівні проміжних представлень, а late fusion - на рівні фінальних прогнозів. Кожен підхід має свої переваги залежно від специфіки даних та задачі.

Cross-modal attention дозволяє моделі фокусуватися на релевантних частинах одної модальності на основі інформації з іншої. Наприклад, система може виділяти ключові слова в тексті пісні, які корелюють з певними акустичними характеристиками.

Графові нейронні мережі

Графові нейронні мережі (Graph Neural Networks, GNN) представляють новий перспективний напрямок для рекомендаційних систем. Вони дозволяють моделювати складні взаємозв'язки між користувачами, елементами та їх характеристиками у вигляді графових структур.

У музичних рекомендаціях графи можуть представляти різноманітні зв'язки: схожість між композиціями, зв'язки між виконавцями, жанрові таксономії, соціальні зв'язки між користувачами. Graph Convolutional Networks (GCN) можуть ефективно поширювати інформацію через ці зв'язки для генерації рекомендацій.

LightGCN спрощує традиційні GCN для рекомендаційних систем, зосереджуючись на найбільш важливих компонентах - neighborhood aggregation та layer combination. Це дозволяє досягти високої продуктивності при менших обчислювальних витратах.

Основи навчання з підкріпленням в рекомендаціях

Навчання з підкріпленням (Reinforcement Learning, RL) розглядає рекомендації як послідовність дій, які впливають на довгострокову задоволеність користувача. На відміну від традиційних підходів, які максимізують миттєву релевантність, RL оптимізує кумулятивну винагороду протягом часу.

Користувач розглядається як середовище, стан якого змінюється в залежності від показаних рекомендацій. Дії агента - це конкретні рекомендації, а винагорода визначається через клікві, час прослуховування, оцінки або інші індикатори задоволеності користувача.

Multi-Armed Bandit (MAB) алгоритми представляють спрощену версію RL, де кожна рекомендація розглядається як незалежна дія. Epsilon-greedy, UCB (Upper Confidence Bound) та Thompson Sampling є популярними MAB алгоритмами, які балансують між exploitation (використанням відомих гарних рекомендацій) та exploration (дослідженням нових можливостей).

Глибоке навчання з підкріпленням

Deep Q-Networks (DQN) поєднують глибокі нейронні мережі з Q-learning для обробки високорозмірних станів користувачів. Це дозволяє моделювати складні користувацькі профілі та генерувати рекомендації, що враховують довгострокові наслідки.

Actor-Critic методи використовують дві нейронні мережі: actor для генерації дій (рекомендацій) та critic для оцінки якості цих дій. Це дозволяє більш стабільне та ефективне навчання порівняно з чисто value-based або policy-based методами.

Policy Gradient методи безпосередньо оптимізують політику рекомендацій, що може призвести до більш природних та різноманітних рекомендацій. REINFORCE з baseline є популярним алгоритмом цього класу.

1.3 Аналіз аналогів

Системи рекомендацій сьогодні стали основою користувацького досвіду у сфері потокового прослуховування музики. Найпопулярніші сервіси, такі як Spotify, YouTube Music, Apple Music, Deezer та інші, використовують складні алгоритми персоналізації, щоб утримувати увагу слухачів та збільшувати їх залучення. У цьому підрозділі буде здійснено порівняльний аналіз сучасних музичних сервісів, які мають інтегровані системи рекомендацій, переваги, недоліки та особливості реалізації.

1. Spotify

Загальна характеристика:

- Один із найпопулярніших музичних сервісів у світі.
- Має понад 500 млн активних користувачів (дані на 2024 рік).
- Пропонує щоденні мікси, тижневі рекомендації ("Discover Weekly"), персоналізовані добірки ("Made for You").

Технології:

- Гібридна система рекомендацій:
 - Колаборативна фільтрація.
 - Аналіз аудіофіч (темп, жанр, тональність).
 - Deep learning-моделі для сесійного прогнозування.
 - Spotify DJ — голосовий персоналізований ШІ на базі LLM.

Переваги:

- Висока точність персоналізації.
- Сильна контентна аналітика.
- Наявність AI-інструментів (включно з генеративними).

Недоліки:

- Потребує багато даних про користувача.
- Схильність до "бульбашки вподобань".
- Менш ефективний для нових акаунтів (проблема холодного старту).

2. YouTube Music

Загальна характеристика:

- Сервіс від Google, який інтегрує музику з відеоконтентом.
- Враховує дані з пошукової системи, YouTube-історії тощо.

Технології:

- Використовує глибинні нейронні мережі (DNN) для аналізу поведінки.
- Контентна фільтрація через семантичний аналіз назв, описів, метаданих.
- Reinforcement learning для оптимізації фіда в реальному часі.

Переваги:

- Враховує весь цифровий слід користувача у системах Google.
- Пропонує відео-версії треків.
- Висока адаптивність до запитів.

Недоліки:

- Слабка курація плейлистів (у порівнянні з Spotify).
- Може плутати музичні смаки з переглядами не-музичного контенту.

3. Apple Music

Загальна характеристика:

- Платформа, орієнтована на високу якість звучання і преміальний контент.
- Сильна інтеграція з екосистемою Apple.

Технології:

- Контентна + експертна фільтрація (куровані добірки).
- Використання поведінкової аналітики Apple ID.
- Поступовий перехід до нейромережових моделей.

Переваги:

- Дуже якісні, людськи-куровані рекомендації.
- Інтеграція з Siri (можна голосом просити нову музику).
- Гармонійна інтеграція з iOS-пристроями.

Недоліки:

- Менш гнучкі моделі самонавчання.
- Мало відкритості щодо алгоритмів (black-box).
- Вузьке коло використання (здебільшого Apple-користувачі).

4. Deezer

Загальна характеристика:

- Європейський музичний сервіс з широким спектром функцій.
- Пропонує функцію Flow — безперервний персоналізований стрім.

Технології:

- Гібридні рекомендації (з 2017 року застосовує deep learning).
- Використання моделі Session-based Recommendations.
- Додатково враховується соціальний граф (друзі, уподобання).

Переваги:

- Інтерфейс, орієнтований на відкриття нової музики.

- Достатньо гнучкі моделі рекомендацій.

Недоліки:

- Менша база користувачів — менше даних для глибокого навчання.
- Рекомендації часто повторюються.

5. Tidal

Загальна характеристика:

- Орієнтований на якість звуку (Hi-Fi, Master).
- Позиціонується як сервіс для меломанів і артистів.

Технології:

- Основний акцент — на куровані списки.
- Моделі рекомендацій — контентно-орієнтовані.
- Високий рівень редакційної участі.

Переваги:

- Висока якість стрімінгу.
- Підтримка незалежних артистів.
- Прозорість монетизації для виконавців.

Недоліки:

- Рекомендації менш "розумні" порівняно зі Spotify/YouTube.
- Вузький фокус — аудіофіли та профільні слухачі.

6. SoundCloud

Загальна характеристика:

- Сервіс, орієнтований на незалежних артистів і молоді релізи.
- Містить треки, які відсутні в інших сервісах.

Технології:

- Колаборативна фільтрація та базовий content-based аналіз.

- Система лайків і переглядів формує базу рекомендацій.

Переваги:

- Відкриття нового контенту, інді-сцена.
- Соціальні функції (коментарі, взаємодія між артистами і слухачами).

Недоліки:

- Відсутність глибокої персоналізації.
- Мала контрольована якість контенту.

Аналіз сучасних музичних сервісів демонструє широке різноманіття підходів до реалізації систем рекомендацій. Spotify та YouTube Music — лідери у впровадженні технологій ШІ, тоді як Apple Music і Tidal більше орієнтуються на редакторську курацію та екосистему. SoundCloud — унікальний своєю соціальною орієнтацією та відкритістю до інді-артистів.

Цей аналіз дозволяє зрозуміти найкращі практики реалізації рекомендаційних систем та визначити, які підходи можуть бути найбільш доцільними для реалізації власного застосунку, зокрема у сфері персоналізованих музичних рекомендацій.

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИСТЕМИ

Розробка інформаційної системи формування музичних рекомендацій потребує детального опрацювання її архітектури, функціональних та нефункціональних вимог, а також проектування ключових компонентів і взаємозв'язків між ними. Проектування є важливим етапом створення системи, оскільки саме на цій стадії визначаються структура даних, логіка роботи модулів, технологічна основа програмної реалізації та способи взаємодії користувача із системою.

У цьому розділі представлено загальну архітектуру програмного продукту, схеми його основних компонентів та модулів, обґрунтування вибору інструментальних засобів реалізації, а також моделі даних і опис інтерфейсів користувача. Особливу увагу приділено проектуванню рекомендаційного модуля, який реалізує алгоритми штучного інтелекту для формування персоналізованих музичних добірок.

При проектуванні враховано сучасні принципи розробки програмного забезпечення, такі як масштабованість, модульність, безпека даних та зручність користування системою.

2.1 Визначення вимог до інформаційної системи

Бізнес-вимоги до системи

Інформаційна система формування музичних рекомендацій з використанням технологій штучного інтелекту є складним інтерактивним застосунком, успішне функціонування якого залежить від досягнення бізнес-цілей усіх зацікавлених сторін. Вона поєднує в собі інструменти машинного навчання, бази даних, модулі взаємодії з користувачем, а також механізми забезпечення безпеки й конфіденційності. Кожен компонент системи має враховувати інтереси ключових стейкхолдерів, які мають свої унікальні цілі, очікування та вимоги.

Внутрішні стейкхолдери, зокрема розробники, націлені на створення не лише функціонального, а й стабільного, масштабованого та технічно гнучкого програмного продукту. Їхня мета включає реалізацію якісної архітектури системи, що дозволить легко впроваджувати оновлення, вдосконалювати функціонал та усувати вразливості. Розробники зацікавлені також у наявності стабільного джерела фінансування, яке забезпечить підтримку та розвиток системи в довгостроковій перспективі. Надійна технічна база для реалізації ШІ-модуля є критично важливою умовою ефективної роботи системи.

Власник продукту (керівництво компанії) розглядає систему як інструмент для досягнення стратегічних бізнес-цілей: збільшення прибутку, зростання аудиторії користувачів, підвищення впізнаваності бренду, а також оптимізації витрат на підтримку та розвиток платформи. Особливої уваги заслуговує економічна ефективність системи — вона повинна забезпечувати значний рівень автоматизації процесів, що дозволить мінімізувати втрати ресурсів та скоротити витрати на ручне обслуговування.

Зовнішні стейкхолдери, у тому числі користувачі, очікують від системи високого рівня зручності, доступності й безпеки. Для них важливим є інтуїтивно зрозумілий інтерфейс, персоналізований контент, якісні рекомендації та можливість швидкого зворотного зв'язку. Важливо, щоб користувач мав змогу швидко знаходити бажані музичні треки, формувати

власні плейлисти, ділитися враженнями та взаємодіяти з рекомендаційною системою через зручні елементи управління. Окрім функціоналу, очікується, що система забезпечить надійний захист персональних даних та відповідатиме всім чинним нормам щодо приватності.

Окрему роль відіграють регуляторні органи, які контролюють дотримання системою вимог у сфері безпеки, захисту інформації, прозорості фінансових операцій і відповідності стандартам шифрування та обробки даних. Дотримання вимог нормативно-правової бази (зокрема, GDPR або Закону України «Про захист персональних даних») є обов'язковою умовою допуску системи до використання на ринку. Важливо також забезпечити прозорість усіх фінансових операцій, що відбуваються в межах функціоналу застосунку (наприклад, у разі підписки чи преміум-доступу), а також зберігання логів дій відповідно до вимог аудиту.

Таким чином, для гармонійного співіснування всіх груп стейкхолдерів система має відповідати узгодженим бізнес-цілям і вимогам. Їх реалізація забезпечить успішне функціонування платформи, її конкурентоздатність та сталість розвитку на ринку цифрових сервісів. Своєчасна реакція на потреби користувачів, стабільна підтримка з боку розробників, стратегічне бачення керівництва та дотримання правових норм дозволять створити продукт, який користуватиметься довірою та популярністю.

Далі наведено діаграму бізнес-вимог та бізнес-цілей кожної особи, яка взаємодіє з системою (див. рис. 2.1). Основними стейкхолдерами системи є власник продукту, розробник, користувач та регуляторні органи.

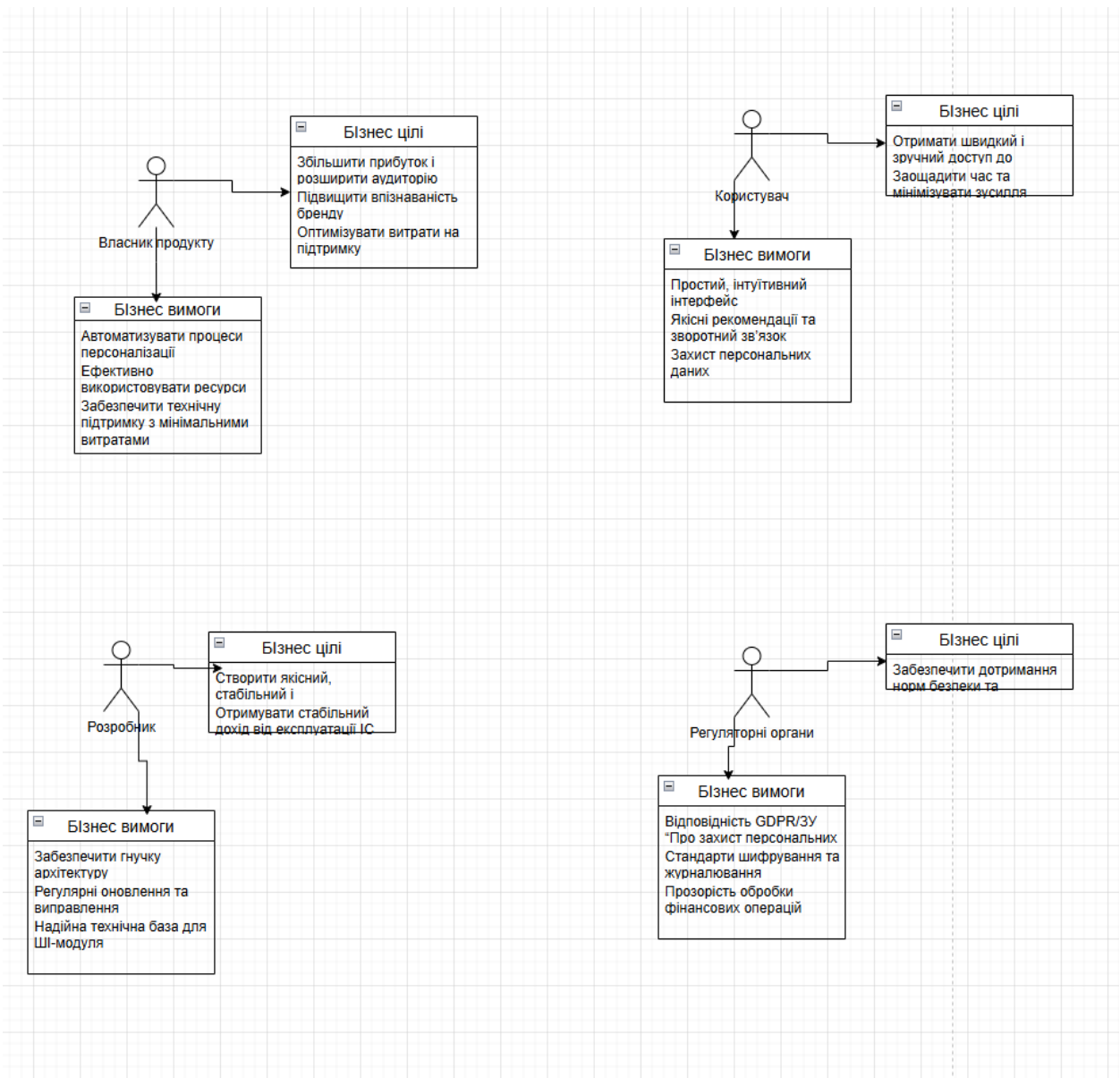


Рисунок 2.1 – Діаграма бізнес-вимог

Функціональні вимоги до системи

Функціональні вимоги до інформаційної системи формування музичних рекомендацій визначають перелік ключових функцій, які забезпечують персоналізовану взаємодію з користувачем, ефективну роботу системи штучного інтелекту та безпеку даних.

Однією з центральних підсистем є підсистема обробки вподобань користувача. Вона включає збір, зчитування та фіксацію історії прослуховувань, жанрових уподобань, настроїв і темпу. Зібрана інформація використовується для формування профілю смаків користувача, що

зберігається в системі для подальшої аналітики. На основі цих даних запускається модель ШІ, яка генерує персоналізовані музичні рекомендації. Результатом роботи цієї підсистеми є вивід списку треків, що найбільше відповідають інтересам користувача.

Важливою є також функціональність збереження історії прослуховувань. Кожен трек, який прослуховує користувач, фіксується в системі разом із датою та оцінкою. Система має підтримувати можливість редагування або видалення записів із історії прослуховувань. Це дозволяє користувачу коригувати рекомендації та впливати на точність моделі.

Підсистема ШІ-рекомендацій має забезпечувати не лише генерацію рекомендацій, але й аналіз профілю, обробку даних та гнучке масштабування моделей. Важливою є можливість регулярного оновлення моделі на основі нових даних для покращення точності результатів.

Інтерфейс користувача (Web/Mobile) забезпечує відображення рекомендацій у зручному форматі. Він повинен підтримувати перегляд персонального списку треків, можливість прослуховування треків прямо з додатку, а також оцінювання кожного треку. Інтерфейс має бути простим, адаптивним та підтримувати мультимовність.

Система автентифікації користувача включає можливість реєстрації, авторизації через email або соцмережі, а також доступ до особистого кабінету. Особистий кабінет повинен надавати доступ до історії прослуховувань, оцінок, персональних даних та можливість керування профілем. Усі дії повинні здійснюватися з дотриманням вимог до захисту персональних даних.

Окремо передбачено можливість інтеграції з мобільними платформами та адаптації функціоналу до різних типів пристроїв. Це дозволяє охопити широку аудиторію та забезпечити доступ до сервісу у будь-якому середовищі.

Таким чином, представлена система забезпечує комплексну взаємодію між користувачем, ШІ-модулем та інтерфейсом доступу, забезпечуючи при цьому збереження конфіденційності та високу якість персоналізованого контенту. Деталізація функціональних вимог представлена на діаграмі (рис.

2.2), а також у специфікаціях вимог, створених за допомогою Specification Manager в середовищі Enterprise Architect (рис. 2.3).

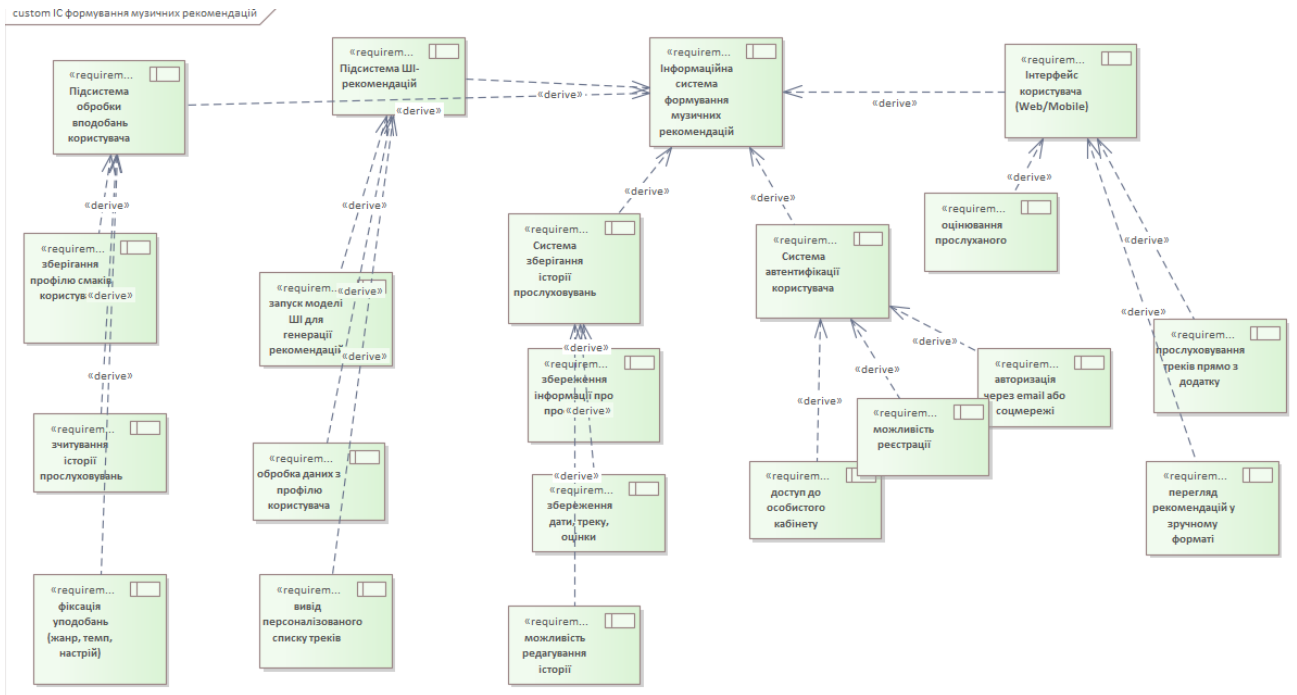


Рисунок 2.2 – Діаграма функціональних вимог

<input checked="" type="checkbox"/> Підсистема обробки вподобань користувача	requirement	Proposed	Medium
<input checked="" type="checkbox"/> зберігання профілю смаків користувача	requirement	Proposed	Low
<input checked="" type="checkbox"/> фіксація уподобань (жанр, темп, настрої)	requirement	Proposed	Medium

Рисунок 2.3 – Специфікація функціональних вимог «Підсистеми обробка вподобань користувача»

<input checked="" type="checkbox"/> Підсистема ШІ-рекомендацій	requirement	Proposed	Medium
<input checked="" type="checkbox"/> вивід персоналізованого списку треків	requirement	Proposed	High
<input checked="" type="checkbox"/> запуск моделі ШІ для генерації рекомендацій	requirement	Proposed	High
<input checked="" type="checkbox"/> обробка даних з профілю користувача	requirement	Proposed	High

Рисунок 2.4 – Специфікація функціональних вимог «Підсистема ШІ рекомендацій»

<input checked="" type="checkbox"/> Система зберігання історії прослуховувань	requirement	Proposed	Medium
<input checked="" type="checkbox"/> збереження дати, треку, оцінки	requirement	Proposed	Medium
<input checked="" type="checkbox"/> збереження інформації про прослухане	requirement	Proposed	Medium
<input checked="" type="checkbox"/> можливість редагування історії	requirement	Proposed	Medium
<input checked="" type="checkbox"/> зчитування історії прослуховувань	requirement	Proposed	Medium

Рисунок 2.5 – Специфікація функціональних вимог «Система зберігання історії прослуховувань»

<input checked="" type="checkbox"/> Система автентифікації користувача	requirement	Proposed	Medium
<input checked="" type="checkbox"/> авторизація через email або соцмережі	requirement	Proposed	Medium
<input checked="" type="checkbox"/> доступ до особистого кабінету	requirement	Proposed	Medium
<input checked="" type="checkbox"/> можливість реєстрації	requirement	Proposed	Medium

Рисунок 2.6 – Специфікація функціональних вимог «Система автентифікації користувача»

<input checked="" type="checkbox"/> Інтерфейс користувача (Web/Mobile)	requirement	Proposed	Medium
<input checked="" type="checkbox"/> оцінювання прослуханого	requirement	Proposed	Medium
<input checked="" type="checkbox"/> перегляд рекомендацій у зручному форматі	requirement	Proposed	Low
<input checked="" type="checkbox"/> прослуховування треків прямо з додатку	requirement	Proposed	Low

Рисунок 2.7 – Специфікація функціональних вимог «Інтерфейс користувача»

Нефункціональні вимоги до системи

Нефункціональні вимоги визначають якісні характеристики інформаційної системи, які є не менш важливими за її функціональність. У контексті системи формування музичних рекомендацій з використанням технологій штучного інтелекту, особливу увагу слід приділити інтерфейсу користувача, безпеці, надійності, масштабованості та швидкодії.

Інтерфейс користувача є одним із ключових аспектів взаємодії користувача із системою. Інтуїтивно зрозуміле та просте у використанні графічне середовище дозволяє підвищити ефективність користування та зменшити навантаження на користувача. Сумісність з різними типами пристроїв (ПК, смартфони, планшети) та адаптивний дизайн, що забезпечує правильне масштабування інтерфейсу під різну роздільну здатність екрану, є критично важливими для сучасного сервісу. Додатковою перевагою є

можливість персоналізації зовнішнього вигляду – зміна темної/світлої теми оформлення.

Безпека інформації – базова вимога до будь-якої системи, що працює з персональними даними користувача. Система повинна підтримувати шифрування паролів у базі даних, а також чітко визначене розмежування прав доступу між різними типами користувачів, включаючи адміністративний доступ. Дотримання стандартів безпеки, зокрема GDPR або Закону України «Про захист персональних даних», є обов’язковою умовою.

Швидкодія системи впливає на задоволення користувацьких очікувань. Особливо важливо забезпечити генерацію списку рекомендацій менш ніж за 1 секунду, швидку обробку запитів до моделі ШІ, а також автоматичну синхронізацію даних між компонентами системи – фронтендом, бекендом та базою даних. Це дозволяє досягти високої продуктивності та запобігти затримкам у роботі інтерфейсу.

Надійність системи передбачає її стабільну та безперебійну роботу впродовж тривалого часу. Система повинна забезпечувати цілодобову доступність сервісу, стійкість до помилок користувача та внутрішніх збоїв, а також мати реалізовані механізми автоматичного резервного копіювання критичних даних для запобігання їх втраті.

Масштабованість є необхідною умовою довготривалого розвитку. Система повинна підтримувати збільшення кількості одночасних користувачів без зниження продуктивності, а також мати гнучку архітектуру, яка дозволить інтегруватися з зовнішніми сервісами, наприклад, API музичних платформ або систем авторизації.

Загальні нефункціональні вимоги до інформаційної системи формування музичних рекомендацій представлено у вигляді діаграми на рис. 2.8. Додатково, усі вимоги структуровано у вигляді специфікацій у менеджері вимог (Specification Manager), що зображені на рис. 2.8 – 2.13.

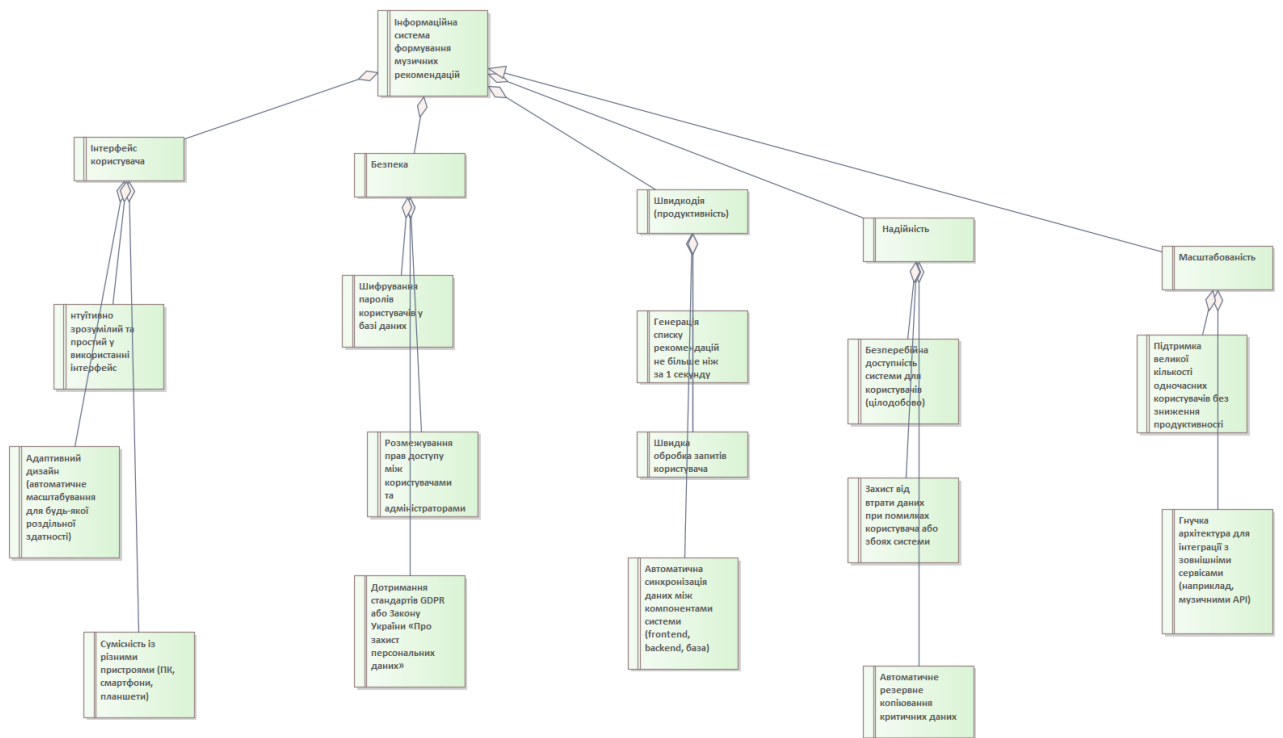


Рисунок 2.8 – Діаграма нефункціональних вимог

Надійність	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Автоматичне резервне копіювання критичних даних	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Безперерйна доступність системи для користувачів (цілодобово)	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Захист від втрати даних при помилках користувача або збоїв системи	Proposed	Medium	Medium

Рисунок 2.9 – Специфікація функціональних вимог «Надійність»

Безпека	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Дотримання стандартів GDPR або Закону України «Про захист персональних даних»	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Розмежування прав доступу між користувачами та адміністраторами	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Шифрування паролів користувачів у базі даних	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Інтерфейс користувача	Proposed	Medium	Medium

Рисунок 2.10 – Специфікація функціональних вимог «Безпека»

Інтерфейс користувача	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Адаптивний дизайн (автоматичне масштабування для будь-якої роздільної здатності)	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Інтуїтивно зрозумілий та простий у використанні інтерфейс	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Сумісність із різними пристроями (ПК, смартфони, планшети)	Proposed	Medium	Medium

Рисунок 2.11 – Специфікація функціональних вимог «Інтерфейс користувача»

Швидкодія (продуктивність)	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Автоматична синхронізація даних між компонентами системи (frontend, backend, база)	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Генерація списку рекомендацій не більше ніж за 1 секунду	Proposed	Medium	Medium
<input checked="" type="checkbox"/> Швидка обробка запитів користувача	Proposed	Medium	Medium

Рисунок 2.12 – Специфікація функціональних вимог «Швидкодія»

Масштабованість

Гнучка архітектура для інтеграції з зовнішніми сервісами (наприклад, музичними API)

Підтримка великої кількості одночасних користувачів без зниження продуктивності

Рисунок 2.13 – Специфікація функціональних вимог «Масштабованість»

2.2 Постановка та алгоритм розв'язання задачі

Задача, визначена для автоматизації процесу пошуку, бронювання, оплати та видачі музичних рекомендацій, має на меті забезпечення високої ефективності взаємодії між користувачами, системою та адміністраторами. Вона забезпечує оптимізацію процесу надання персоналізованих музичних рекомендацій, що базуються на перевагах користувачів, і зменшує потребу в ручному втручанні в цей процес. Економічна доцільність полягає в зменшенні витрат на фізичні інтерфейси взаємодії з користувачами, зниженні людського фактору та підвищенні оперативності обробки запитів, а також у покращенні користувацького досвіду. Рішення цієї задачі за допомогою сучасних обчислювальних засобів дозволяє забезпечити високу швидкість, масштабованість, доступність і інтеграцію з різними платіжними системами та базами даних для збору інформації про музичні вподобання користувачів.

Об'єктами цієї системи є користувачі (слухачі), адміністратори системи, музичні треки, музичні категорії та рекомендаційні алгоритми.

Вихідною інформацією системи є персоналізовані музичні рекомендації, повідомлення про стан взаємодії з системою (наприклад, підтвердження створення плейлиста або успішного виконання рекомендацій), історія взаємодії з платформою. Ці дані використовуються користувачами для налаштування власних музичних вподобань, а адміністраторами для моніторингу та управління системою.

Задача вирішується в реальному часі, на запит користувача. Вихідна інформація (наприклад, рекомендація, пісня) надається миттєво після виконання операції.

Умови, за яких припиняється автоматизоване розв'язання задачі, включають:

- Відсутність доступу до Інтернету або серверів системи.
- Збій або атака на інформаційну систему, що загрожує безпеці даних.
- Помилки у платіжних системах (якщо вони використовуються для надання платних послуг).

Розподіл дій між персоналом та технічними засобами за різних ситуацій:

1. У штатному режимі роботи:
 - Технічні засоби (ІС) автоматично виконують основні функції: пошук треків, фільтрацію, надання рекомендацій, генерацію плейлистів, автентифікацію користувачів та формування звітності.
 - Персонал (адміністратор) контролює коректність даних у системі, оновлює базу треків, ціни (для платних функцій), доступність музичних жанрів і т.д.
2. У випадку часткових збоїв:

- ІС фіксує помилки, повідомляє користувачів про проблему та логує події для подальшого аналізу.
- Адміністратор аналізує причини збоїв і надає користувачам інформацію про вирішення проблеми (наприклад, через повідомлення або підтримку).

Вихідна інформація. Вихідна інформація системи формування музичних рекомендацій використовується для надання користувачам оперативних результатів взаємодії з системою, а саме, для генерації персоналізованих музичних рекомендацій. Це може включати список рекомендованих треків, що відповідають уподобанням користувача, або ж автоматично сформовані плейлисти, які враховують попередні вибори та вподобання. Крім того, система може надавати рекомендації, що базуються на настрої користувача або на його музичних звичках. Окрім того, система формує звіти про активність користувача, такі як статистика прослуховувань, кількість взаємодій з рекомендаціями, оцінки пісень, що дозволяє аналізувати ефективність запропонованих рекомендацій і покращувати їх в майбутньому.

Ця інформація є необхідною для всіх учасників процесу. Для користувачів важливо, щоб система надавала актуальні та точні рекомендації, які відображають їхні вподобання і дозволяють здійснити вибір музики з мінімальними зусиллями. Крім того, для адміністраторів системи важливо мати сповіщення та звіти про взаємодію користувачів з платформою, включаючи статистику успішності рекомендацій, проблеми з доступністю контенту або оновленням бази даних. Це дає змогу оперативно вирішувати проблеми, коригувати алгоритми рекомендацій і забезпечувати належну роботу системи в режимі реального часу.

Вихідна інформація є важливою для підтримки безперервної роботи системи, її надійності та ефективності, а також для забезпечення прозорості взаємодії з користувачами. Вона забезпечує швидке надання результатів і дозволяє операторам та адміністраторам системи слідкувати за її ефективністю, допомагаючи коригувати й адаптувати систему до потреб користувачів. Дана інформація має бути коректною, своєчасною та зрозумілою, щоб максимально полегшити користування платформою та зберегти високий рівень довіри з боку користувачів.

Нижче наводиться таблиця 2.14 із переліком і описом вихідних повідомлень, що формуються під час автоматизованого розв'язання задачі, а також описом вимог до них.

Назва	Ідент	Фор	Періо	Те	Кори
вихідного повідомлен ня	ифікатор	ма подання і вимоги до неї	дичність видання	рмін видання і допусти мий час затримк и	стувачі інформації

Згенеровані рекомендації на основі вподобань користувача	VR-001	Файл або список з рекомендаціями	Після кожного оновлення даних	Миттєво, затримка – до 5 секунд	Користувач
Згенерований плейлист в форматі PDF	VR-002	Файл (PDF) з плейлистом	Після створення плейлисту	Миттєво, затримка – до 5 секунд	Користувач
Підтвердження зміни музичних вподобань користувача	VR-003	Діалогове вікно або сповіщення	Після змін вподобань користувача	Миттєво, без затримки	Користувач
Підтвердження вибору музичного жанру	VR-004	Діалогове вікно або підтвердження в системі	Після вибору жанру	Миттєво, без затримки	Користувач
Сповіщення про успішну зміну даних в базі	VR-005	Сповіщення у правому верхньому кутку екрану	Після успішної зміни даних в БД	Миттєво, без затримки	Адміністратор

Виведення прогнозу погоди	VP-006	Вивідення на екран прогнози погоди для місця користувача	На запит користувача	Миттєво, без затримки	Користувач
---------------------------	--------	----------------------------------------------------------	----------------------	-----------------------	------------

Таблиця 2.14-Опис вихідних повідомлень

Кожне з вихідних повідомлень, що формуються в системі музичних рекомендацій, містить структурні одиниці інформації, які мають самостійне значення.

1. Згенеровані рекомендації на основі вподобань користувача (VP-001)
 - Структурні одиниці: Ім'я користувача, список рекомендованих пісень, автори пісень, жанри, час тривалості кожної пісні.
 - Вимоги до точності: Уся інформація має бути повністю коректною та відповідати даним у базі. Помилка в пісні чи жанрі є критичною. Допустимий рівень похибки – 0%.
2. Згенерований плейлист в форматі PDF (VP-002)
 - Структурні одиниці: Назва плейлисту, список пісень, виконавці, жанри, тривалість, дата створення.
 - Вимоги до точності: Плейлист має бути сформований без помилок, відповідати вимогам формату PDF і містити точні дані щодо треків.
3. Підтвердження зміни музичних вподобань користувача (VP-003)
 - Структурні одиниці: Ім'я користувача, старі і нові вподобання, зміни в жанрах або піснях.
 - Вимоги до точності: Зміни мають бути відображені точно, без помилок або втрати інформації. Відповідність даним у базі – обов'язкова.
4. Підтвердження вибору музичного жанру (VP-004)
 - Структурні одиниці: Обраний жанр, нові рекомендації, інформація про те, чи є цей жанр у бібліотеці користувача.

- Вимоги до точності: Необхідна точна відповідність між вибраним жанром і наданими рекомендаціями.
5. Сповіщення про успішну зміну даних у базі (VP-005)
- Структурні одиниці: Ідентифікатор зміни, тип змінених даних, дата і час зміни, статус операції (успішно).
 - Вимоги до точності та надійності: Дані мають точно відображати результат дії, без затримки. Важлива синхронізація з базою даних і відсутність дублювання.
6. Виведення прогнозу погоди (VP-006)
- Структурні одиниці: Місто, температура, зображення погодних умов на найближчий час.
 - Вимоги до точності та надійності: Дані мають бути відображені точно і відповідати реальному прогнозу для вказаного міста.

Вхідна інформація

Вхідна інформація в системі музичних рекомендацій використовується для здійснення всіх основних операцій: пошуку музичних треків, реєстрації користувачів та формування рекомендацій. Дані надходять через інтерфейси користувача (веб-форми або додатки) та з бази даних. Частина інформації є постійною, а інша формується динамічно під час сесії користувача в процесі взаємодії з платформою. Вхідна інформація є основою для роботи рекомендаційної системи та забезпечує коректність та точність сформованих рекомендацій.

Нижче наведено таблицю 2.15 із переліком та описом основних вхідних повідомлень, що використовуються для формування бази даних, підтримки процесу взаємодії з користувачем та генерації вихідних рекомендацій.

з/п	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
	Параметри пошуку музичних треків	IN-001	Веб-запит	На запит користувача	Користувач
	Інформація про наявність музичних треків	IN-002	Файл БД / Запит	Постійно доступний	Внутрішня база даних

з/п	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
	Персональні дані користувачів	IN-003	Веб-форма	Під час реєстрації/авторизації	Користувач
	Дані для формування рекомендацій	IN-004	Веб-форма	Під час взаємодії з платформою	Користувач
	Дані про музичні жанри	IN-005	Файл БД / Запит	Постійно доступний	Внутрішня база даних

Таблиця 2.15 – Перелік і опис вхідних повідомлень

Опис вхідних повідомлень:

1. Параметри пошуку музичних треків (IN-001)
 - Ідентифікатор: IN-001
 - Форма подання: Веб-запит, через який користувач може вказати свої уподобання для пошуку треків (жанр, виконавець, настрій).
 - Термін і частота надходження: Данні надходять на запит користувача, коли він хоче знайти нові пісні або плейлисти.
 - Джерело: Користувач, який взаємодіє з платформою.
2. Інформація про наявність музичних треків (IN-002)
 - Ідентифікатор: IN-002
 - Форма подання: Дані з бази даних або запити до неї, що містять інформацію про доступні треки, їх метадані (назва, виконавець, жанр тощо).
 - Термін і частота надходження: Інформація доступна постійно та оновлюється в реальному часі.
 - Джерело: Внутрішня база даних, що містить інформацію про треки.
3. Персональні дані користувачів (IN-003)
 - Ідентифікатор: IN-003
 - Форма подання: Веб-форма, де користувач реєструється або авторизується на платформі, надаючи свої дані (ім'я, email, пароль).
 - Термін і частота надходження: Ці дані вводяться під час реєстрації або авторизації користувача.

- Джерело: Користувач, який надає свої особисті дані для створення акаунту.
4. Дані для формування рекомендацій (IN-004)
- Ідентифікатор: IN-004
 - Форма подання: Веб-форма або інтерактивні елементи на платформі для вибору музичних вподобань, жанрів або оцінки пісень.
 - Термін і частота надходження: Під час взаємодії з платформою користувач може вводити свої вподобання, що сприятиме створенню персоналізованих рекомендацій.
 - Джерело: Користувач, який взаємодіє з системою, оцінює або надає свої музичні переваги.
5. Дані про музичні жанри (IN-005)
- Ідентифікатор: IN-005
 - Форма подання: Дані з бази даних про різні музичні жанри, які користувач може вибирати для персоналізації рекомендацій.
 - Термін і частота надходження: Постійно доступні дані, що регулярно оновлюються в базі даних.
 - Джерело: Внутрішня база даних платформи, яка містить категорії музики, жанри, виконавців тощо.

2.3 Моделі та алгоритми надання рекомендацій

Інформаційна система формування музичних рекомендацій базується на використанні різних алгоритмів персоналізації, які дозволяють пропонувати користувачу музичний контент, максимально відповідний його смакам та вподобанням. Для досягнення високої точності рекомендацій у системі передбачається застосування кількох типів алгоритмів — від найпростіших у реалізації до складних моделей, що використовують сучасні технології машинного навчання й штучного інтелекту. У цьому підрозділі докладно розглядаються ці підходи.

1. Алгоритм К-ближчих сусідів (KNN)

Загальна ідея

К-ближчих сусідів (K-Nearest Neighbors) — це один із найпростіших алгоритмів рекомендацій. Його суть полягає у знаходженні K користувачів або об'єктів, найбільш подібних до заданого користувача або треку, та використанні їх уподобань для формування списку рекомендацій.

Принцип роботи

У варіанті user-based KNN система обчислює схожість між користувачами на основі їхніх оцінок треків або історії прослуховувань. Користувачу рекомендуються треки, які оцінили або прослухали його найближчі сусіди.

У варіанті item-based KNN обчислюється схожість між треками, а рекомендації будуються на основі схожих до вже вподобаних треків.

Метрики схожості

Для обчислення близькості зазвичай застосовуються:

- Косинусна схожість
- Кореляція Пірсона
- Евклідова відстань

Переваги:

- Простота реалізації та пояснюваність.

- Не потребує навчання моделі (алгоритм «лінійний»).

Недоліки:

- Не масштабований для великих даних (складність $O(n)O(n)$ при пошуку сусідів).
- Схильний до проблеми холодного старту для нових користувачів або треків.

2. Контентно-орієнтовані рекомендації

Загальна ідея

Контентно-орієнтовані моделі аналізують властивості треків (жанр, темп, ритм, мова, інструментальність) і формують профіль користувача на основі вподобаного контенту. Далі система шукає інші треки з подібними характеристиками.

Принцип роботи

- Кожен трек описується вектором ознак.
- Профіль користувача формується як усереднений вектор прослуханих або оцінених треків.
- Рекомендації генеруються на основі близькості між профілем користувача і треками в базі.

Переваги:

- Працює навіть для нових користувачів (якщо є базова взаємодія).
- Можна рекомендувати нові треки без історії прослуховувань інших користувачів.

Недоліки:

- Низька різноманітність — рекомендації зазвичай подібні до вже відомого контенту.
- Не враховує соціальних зв'язків або вподобань інших слухачів.

3. Матрична факторизація (SVD)

Загальна ідея

Матрична факторизація — це техніка зменшення розмірності, яка дозволяє виявляти приховані закономірності у вподобаннях користувачів. Найпоширеніший метод — сингулярне розкладання матриці (SVD).

Принцип роботи

- Маємо матрицю RR розміром $m \times n$, де m — кількість користувачів, n — кількість треків, а елементи — оцінки або кількість прослуховувань.
- Факторизуємо матрицю:
- Оцінки невідомих пар користувач-трек прогнозуються як добуток латентних векторів.

Переваги:

- Виявляє приховані патерни вподобань.
- Масштабується для великих даних при використанні оптимізованих реалізацій (ALS, stochastic gradient descent).

Недоліки:

- Проблема холодного старту для нових об'єктів.
- Потребує оновлення при надходженні нових даних.

4. Нейронні мережі для рекомендацій

Autoencoder

Автокодуювальні мережі навчаються стискати інформацію про вподобання користувача до компактного латентного вектора і відновлювати її.

Модель складається з:

- Кодуювальника, що стискає вхідні дані до латентного простору.
- Декодуювальника, що реконструює оцінки або історію прослуховувань.

Feedforward Neural Networks

Такі мережі приймають вхідні ознаки користувача й треку (жанри, країна, час доби, пристрій) та видають прогноз імовірності того, що трек буде вподобано.

RNN / LSTM / GRU

Використовуються для послідовних рекомендацій, коли важливий порядок прослуховувань:

- Прогнозують наступний трек на основі історії сесії.

Переваги:

- Висока точність завдяки гнучкості моделей.
- Можливість роботи з різними типами даних (числові, категоріальні, текстові).

Недоліки:

- Велика обчислювальна складність.
- Потреба в обширних навчальних даних.

5. Гібридні системи

Загальна ідея

Гібридні системи поєднують кілька підходів для усунення обмежень окремих методів. Наприклад:

- KNN + контентна фільтрація: для користувача спочатку шукаються сусіди, потім їх рекомендації фільтруються за контентом.
- SVD + деревоподібні моделі: латентні ознаки передаються на вхід моделей XGBoost для підвищення точності.

Приклади реалізації:

- Netflix: поєднання матричної факторизації й правил.
- Spotify: гібрид колаборативного й контентного підходу + сесійне моделювання.

Переваги:

- Висока точність.
- Гнучкість і стійкість до холодного старту.

Недоліки:

- Висока складність реалізації та налаштування.
- Необхідність A/B тестування для підбору ваг моделей.

Підсумок

Для проєктованої системи рекомендується використовувати багаторівневий підхід:

- KNN / контентна фільтрація — як базова реалізація для старту.
- SVD або Autoencoder — для підвищення якості рекомендацій із накопиченням даних.
- Гібридна модель — як перспективний напрям розвитку системи після запуску MVP (minimum viable product).

Такий підхід дозволяє створити гнучку й ефективну систему, що забезпечить користувачам якісні музичні рекомендації.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Розділ присвячено опису практичної реалізації інформаційної системи формування музичних рекомендацій із використанням технологій штучного інтелекту. На цьому етапі проектування переходить у стадію створення робочого програмного продукту, що поєднує в собі окремі модулі й компоненти, описані в попередніх розділах.

Основна увага приділяється програмній реалізації ключових підсистем:

- модуля автентифікації й керування обліковими записами користувачів;
- підсистеми збору, обробки та зберігання даних про вподобання й історію прослуховувань;
- рекомендаційного модуля на базі обраних алгоритмів машинного навчання;
- інтерфейсу користувача, що забезпечує інтерактивну взаємодію з системою;
- засобів захисту персональних даних та інформаційної безпеки.

У цьому розділі розглянуто використані інструменти та середовища розробки, особливості реалізації програмної архітектури, а також окремі фрагменти коду й приклади взаємодії з системою. Реалізація системи відповідає функціональним і нефункціональним вимогам, зазначеним на етапі проектування, та враховує сучасні стандарти розробки програмного забезпечення й інформаційної безпеки.

3.1 Інформаційне забезпечення

3.1.1 Загальна характеристика інформаційного забезпечення (ІЗ)

Інформаційне забезпечення є важливою складовою функціонування інформаційної системи формування музичних рекомендацій. Воно включає сукупність інформаційної бази, нормативних матеріалів, правил обробки, зберігання, структурування і кодування даних. Головною метою інформаційного забезпечення є забезпечення надійної, достовірної та структурованої інформаційної підтримки всіх процесів системи. Це охоплює етапи від введення користувацьких даних до формування персоналізованих музичних рекомендацій.

Структура інформаційного забезпечення в системі формування музичних рекомендацій включає кілька важливих елементів:

- **Методичні та інструктивні матеріали:** документи, що описують порядок роботи системи, процедури взаємодії користувачів та адміністраторів, а також специфікації алгоритмів для генерації рекомендацій.
- **Схеми класифікації та кодування:** визначення стандартів для категоризації музичних жанрів, виконавців та інших характеристик треків.
- **Нормативно-довідкова інформація:** надання даних про треки, виконавців, альбоми та жанри для побудови рекомендаційних алгоритмів.
- **Інформаційні масиви:** набори даних, що зберігаються в системі, включаючи інформацію про уподобання користувачів, їхню історію прослуховувань та результативність рекомендацій.
- **Вихідні інформаційні повідомлення:** персоналізовані рекомендації, звіти про активність користувача, а також повідомлення про зміну особистих даних.
- **Вхідні інформаційні повідомлення:** дані користувачів, що вводяться через веб-інтерфейси або програми (наприклад, інформація про пісні, оцінки, вибір жанрів).

Уся інформація зберігається в централізованій реляційній базі даних, яка розміщується на сервері з можливістю резервного копіювання для забезпечення безпеки даних. Основним типом носія є електронна пам'ять серверів (HDD/SSD) з доступом через локальну мережу або Інтернет. Для забезпечення високої доступності та масштабованості використовуються технології резервного копіювання та відновлення даних.

Система керування базами даних (СКБД)

Для управління даними в системі використовується **Microsoft SQL Server**, який є потужною і надійною системою управління базами даних, спеціально розробленою для використання в корпоративних середовищах. Ось основні переваги використання цієї СУБД для музичної рекомендаційної системи:

- **Високий рівень безпеки** з можливістю шифрування даних та керування доступом.
- **Масштабованість** для роботи з великими обсягами даних та складними запитам.
- **Інтуїтивно зрозумілий інтерфейс**, що спрощує роботу з базою даних.
- **Сумісність із .NET-орієнтованими веб-технологіями**, що дозволяє ефективно інтегрувати систему з веб-додатками та іншими онлайн-сервісами.
- **Надійність і стабільність**, що забезпечує постійну доступність даних і запобігає їх втраті.

Надійність і безпека інформаційного забезпечення

Для забезпечення надійності та безпеки інформації в системі було впроваджено кілька важливих заходів:

- **Перевірка введених даних на рівні інтерфейсу** для забезпечення коректності даних користувачів та їхніх уподобань.
- **Обмеження цілісності на рівні СУБД**, включаючи первинні та зовнішні ключі для правильного зберігання і взаємодії даних.
- **Контроль доступу за ролями** (користувач, адміністратор), що дозволяє забезпечити безпеку та правильний доступ до даних.

Структура інформаційного забезпечення

Інформаційне забезпечення системи формується на основі чіткої структури, що включає:

- Методичні матеріали, що описують інтерфейси та алгоритми взаємодії з користувачами.
- Нормативні документи, що визначають стандарти класифікації музичних жанрів, виконавців та рекомендацій.
- Таблиці, які містять інформацію про треки, користувачів, їхні вподобання і плейлисти.
- Механізми для генерації рекомендацій і їх подальше збереження.

Інформаційне забезпечення гарантує ефективну підтримку всіх функцій системи, від введення даних користувачами до надання персоналізованих музичних рекомендацій.

3.1.2 Організація збору і передавання первинної інформації

Первинна інформація, необхідна для функціонування інформаційної системи формування музичних рекомендацій, надходить з різних джерел і використовується для формування вхідних документів у пам'яті обчислювального пристрою. Організація збору інформації передбачає дотримання вимог до своєчасності, достовірності, формату подання та автоматизації обробки даних. Джерелами можуть бути як внутрішні модулі системи (веб-інтерфейси, бази даних), так і зовнішні підрозділи та сервіси (API для музичних треків, музичні платформи) (див. таблиця 3.1).

Загальні вимоги до збору та передавання інформації:

- **Забезпечення актуальності та повноти даних:** Збір актуальної інформації про користувачів та їх вподобання для створення персоналізованих рекомендацій.
- **Стандартизований формат подання:** Використання JSON, SQL-запитів, API для обміну даними між модулями.
- **Автоматична перевірка структури й обов'язкових полів:** Перевірка введених даних користувачами на коректність та повноту.
- **Шифрування або захист каналів передавання:** Використання HTTPS, токен-авторизації для захищеного обміну даними.
- **Мінімізація ручного введення:** Автоматизація збору та обробки даних без потреби вручну вводити інформацію.
- **Фіксація часу взаємодії:** Час запиту рекомендацій або обробки оцінок користувача для моніторингу ефективності системи.

Назва вхідного повідомлення	Джерело інформації	Формат подання	Періодичність	Місце зберігання даних
Базові дані користувача	Користувач (через веб-форму)	Інтерфейс (HTTPS)	Регістрації / входу	Активна пам'ять
Метри пошуку музичних треків	Користувач	Інтерфейс	Запит користувача	Часовий кеш
Дані для формування рекомендацій	Користувач	Інтерфейс	Взаємодії з платформою	БД (табл. Users)
Інформація про наявність музичних треків	Внутрішня БД системи	Запит	В режимі реального часу	Таблиця Tracks

Назва вхідного повідомлення	Джерело інформації	Спосіб подання	Періодичність	Тип даних
Про музичні жанри	Вішня БД (табл. Genres)	Запит	Завжди доступна	Список Genres
Про пільги / типи квитків (при наявності платних функцій)	Вішня довідкова таблиця	Запит	Вибір функцій	TicketTypes
Про взаємодію з платіжною системою	Зовнішня система (зовнішній сервіс)	Запит (JSON)	Підтвердження	Сервйонний запит
Сервіс прогнозу погоди на основі API для місць користувачів	Зовнішній погодні API	Запит (JSON)	Запит користувача	Список пам'ять / кеш

Таблиця 3.1 – Джерела і характеристика вхідних повідомлень

Опис вхідних повідомлень:

1. Персональні дані користувача
 - Джерело: Користувач надає свої дані при реєстрації або авторизації на платформі.
 - Спосіб подання: Інформація передається через веб-інтерфейс за допомогою HTTPS.
 - Періодичність: Ці дані надходять при першому вході або при зміні персональних даних користувача.
2. Параметри пошуку музичних треків
 - Джерело: Користувач вводить параметри пошуку (наприклад, жанр, виконавець, настрій).
 - Спосіб подання: Веб-інтерфейс передає запит на сервер для пошуку відповідних треків.
 - Періодичність: Ці дані надходять кожного разу, коли користувач хоче знайти нові треки.
3. Дані для формування рекомендацій
 - Джерело: Користувач надає дані про свої музичні вподобання, оцінки пісень, звички прослуховування.
 - Спосіб подання: Інформація надходить через веб-форму.

- Періодичність: Після взаємодії з платформою або на основі історії прослуховувань.
- 4. Інформація про наявність музичних треків
 - Джерело: Внутрішня база даних системи, яка містить інформацію про доступні треки.
 - Спосіб подання: Дані запитуються за допомогою SQL-запиту.
 - Періодичність: В режимі реального часу, коли користувач взаємодіє з системою.
- 5. Дані про музичні жанри
 - Джерело: Внутрішня база даних, яка зберігає список музичних жанрів.
 - Спосіб подання: SQL-запит до бази даних для отримання доступних жанрів.
 - Періодичність: Постійно доступні для системи, коли користувач хоче вибрати жанр.
- 6. Дані про пільги / типи квитків (для платних функцій)
 - Джерело: Довідкова таблиця, що містить інформацію про типи підписок.
 - Спосіб подання: SQL-запит.
 - Періодичність: Після вибору функції платної підписки.
- 7. Дані про оплату
 - Джерело: Платіжна система або сервіс.
 - Спосіб подання: API-запит для обробки платежів і підтвердження оплати.
 - Періодичність: Після підтвердження платежу.
- 8. Прогноз погоди
 - Джерело: Зовнішній API, що надає прогноз погоди для місць маршруту користувача.
 - Спосіб подання: REST-запит через API.
 - Періодичність: На запит користувача для конкретного місця.

3.1.3 Побудова системи класифікації та кодування

Система класифікації та кодування в інформаційній системі формування музичних рекомендацій забезпечує унікальну ідентифікацію всіх об'єктів, які беруть участь у процесах пошуку музичних треків, генерації рекомендацій, оцінки пісень і взаємодії користувачів з платформою. Використання системи кодування дозволяє зменшити обсяг даних, прискорити обробку запитів, знизити ризик помилок при зберіганні та передачі інформації.

Використовувані класифікатори та методи кодування

1. UTF-8 (Unicode Transformation Format 8-bit)
 - Призначення: Текстова інформація (імена користувачів, назви пісень, виконавці, жанри, повідомлення).

- Метод: Кодування символів змінної довжини (1–4 байти).
 - Структура коду: Кожен символ має свій унікальний код у таблиці Unicode.
 - Приклад: «Kiev» → [0x4B 0x69 0x65 0x76].
2. ISO 8601 (Міжнародний стандарт представлення дати й часу)
- Призначення: Представлення часу для взаємодії користувачів з платформою, створення плейлистів, взаємодія з рекомендаціями.
 - Метод: Формат YYYY-MM-DDTHH:MM:SSZ або з вказівкою часового поясу (наприклад, +03:00).
 - Довжина: 20 символів.
 - Приклад: 2025-05-22T16:00:00+03:00.
3. QR Code (Quick Response Code)
- Призначення: Унікальне кодування рекомендацій або музичних плейлистів для швидкого доступу або обміну.
 - Метод: Двовимірне матричне кодування інформації.
 - Структура коду: Статичний/динамічний текстовий запис (до 4 296 символів).
 - Зміст коду: Ідентифікатор треку, ім'я виконавця, жанр, дата додавання.
 - Приклад: TID:00093412;Name:Coldplay;Track:Fix You;Genre:Rock;Date:2025-06-01 → QR.

Використання стандартизованих форматів (UTF-8, ISO 8601, QR Code)

Забезпечує сумісність між підсистемами, зручність обробки даних на етапі програмування, а також мінімізує ймовірність помилок при передачі та відображенні інформації. Таке використання стандартів сприяє надійності, масштабованості та безпеці інформаційної системи, забезпечуючи ефективне управління даними та їх безпечне збереження.

Це дозволяє системі формувати точні рекомендації та зберігати відповідну інформацію для подальшого аналізу і коригування алгоритмів, що підвищує ефективність платформи.

3.1.4 Проктування форм первинних документів, машинограм та відеокадрів

У процесі функціонування інформаційної системи формування музичних рекомендацій створюється низка первинних електронних документів і повідомлень, які відображають результати обробки даних і виконують як інформаційно-довідкову, так і юридичну функцію. Ці форми документів спроектовані з урахуванням вимог зручності для користувача, автоматизованої обробки даних і забезпечення цифрової ідентифікації. Вони можуть бути реалізовані у вигляді PDF-документів, діалогових вікон (відеокадрів) або електронних сповіщень, що надаються користувачеві через інтерфейс платформи.

Назва функції	Ідентифікатор	Користувачі	Формат подання	Періодичність	Тривалість виконання
Сформовані рекомендації на основі вподобань користувача	01	Користувач	PDF	Оновлення даних	Щодня (до 1 години)
Сформований плейлист у форматі PDF	02	Користувач	PDF	Створення плейлисту	Щодня
Здійснення зміни музичних вподобань користувача	03	Користувач	Модальне вікно	Внесення змін	Щодня
Здійснення вибору музичного жанру	04	Користувач	Модальне вікно	Вибір жанру	Щодня
Здійснення про успішну зміну даних в базі	05	Адміністратор	Інкасація/спливаюче вікно	Дії	Щодня
Вибір погоди для місць користувачів	06	Користувач	Інфо-блок	Вибір рекомендацій	Щодня

Таблиця 3.2 – Основні документи та форми, що генеруються системою

Вимоги до форм документів:

- **Зміст:** Має містити лише перевірену, актуальну та уніфіковано структуровану інформацію, яка відповідає вимогам користувача.
- **Форма подання:** Форма подання документів повинна бути зрозумілою користувачеві, не перевантаженою зайвими деталями та забезпечувати зручний доступ до інформації.
- **Мобільна адаптивність:** Усі форми повинні коректно відображатися на мобільних пристроях для забезпечення зручності користування на різних платформах.
- **Безпека:** Електронні документи мають містити унікальні коди (QR), бути захищеними від підробки і гарантувати цілісність даних.
- **Мовна локалізація:** Підтримка принаймні української мови для користувачів, що забезпечить доступність для більш широкої аудиторії.

Таким чином, система формує набір стандартних машинограм і відеокадрів, що дозволяють ефективно інформувати користувача про результат обробки запитів (рекомендації, плейлисти, зміни вподобань) і водночас виконують функції юридичного підтвердження їх дій. Надсилання основних документів на електронну пошту підвищує зручність і забезпечує збереження проїзного документа поза межами системи для подальшого використання або перевірки.

3.1.5 Структура інформаційних масивів

Інформаційні масиви є основою функціонування інформаційної системи формування музичних рекомендацій. Кожен масив відповідає окремому об'єкту предметної області – користувачам, трекам, жанрам, рекомендаціям тощо – і має чітко визначену структуру. Масиви забезпечують ефективне зберігання, обробку і доступ до даних, що критично важливо для швидкого та точного формування рекомендацій. Таблиця 3.3 надає характеристику основних інформаційних масивів, що використовуються в системі, включаючи носій, обсяг, спосіб організації та ключі індексування.

енуван ня масив у	ї інформації	кс. об'єм	кина запи су	од організації	Ключі упорядкува ння	ексний масив
стувачі	sers	000 запис ів	айтів	еризований індекс за UserId	Id (PK), Email (UNIQ)	sers_Email
ні треки	racks	0 000 запис ів	024 байті в	еризований індекс за TrackId	Id (PK), ArtistId (FK), GenreId (FK)	racks_ArtistId, IX_Tracks_Gen reId
и	genres	запис ів	айтів	ий індекс за GenreId	Id (PK), Name	genres_Name
ендаці ї	recommendati ons	000 запис ів	512 байті в	еризований індекс за Recommendati onId	mmendationI d (PK), UserId (FK)	ecs_UserId
и	atings	0 000 запис ів	128 байті в	ий індекс за RatingId	Id (PK), TrackId (FK), UserId (FK)	atings_UserId
щення про зміну даних	otifications	000 запис ів	айтів	ий індекс за NotificationId	cationId (PK), UserId (FK)	otifications_Use rId

Таблиця 3.3 – Структура інформаційних масивів

Джерело: сформовано автором на основі виконаного дослідження

Як видно з таблиці, структура кожного масиву оптимізована для специфіки збереження даних, що відповідають функціям музичної рекомендаційної системи. Використовуються як кластеризовані індекси для швидкого доступу до основних записів (наприклад, до треків або рекомендацій), так і прості індекси для підтримки цілісності даних (наприклад, для жанрів або сповіщень). Це дозволяє забезпечити високу швидкість доступу до інформації, підтримувати її цілісність і забезпечити масштабованість системи при зростанні обсягу даних.

3.1.6 Вибір СУБД

Для реалізації інформаційної системи формування музичних рекомендацій була обрана система керування базами даних SQLite3. SQLite3 є потужною, вбудованою реляційною системою керування базами даних, що дозволяє ефективно зберігати дані локально на пристрої користувача або сервері без потреби в окремому сервері бази даних. Вибір цієї СУБД обумовлений вимогами до надійності, зручності адміністрування та високої продуктивності при роботі з невеликими і середніми обсягами даних.

Обґрунтування вибору:

- Простота налаштування та використання: SQLite3 є вбудованою СУБД, що не вимагає окремого сервера або складної конфігурації, що значно полегшує процес розгортання та адміністрування.
- Низькі вимоги до ресурсів: Вона ідеально підходить для систем з помірними навантаженнями, таких як музичні рекомендаційні системи, що працюють із даними про користувачів, їхні вподобання та історію прослуховувань.
- Масштабованість: Незважаючи на свою простоту, SQLite3 підтримує великі об'єми даних і дозволяє ефективно працювати з базами, які містять десятки тисяч записів.
- Швидкість роботи: SQLite3 забезпечує високу продуктивність при роботі з базами даних невеликого та середнього розміру, що є важливим для своєчасного формування рекомендацій.
- Сумісність з різними платформами: SQLite3 сумісна з більшістю операційних систем, таких як Windows, macOS, Linux, а також з мобільними платформами (Android, iOS).
- Інтеграція з мовами програмування: Легка інтеграція з популярними мовами програмування, такими як Python, C++, Java, і підтримка SQL-запитів через ADO.NET, JDBC, та інші бібліотеки для зручної роботи з базами даних.

Короткий опис обраної СУБД:

- Тип: Реляційна СУБД.
- Платформа: Windows, macOS, Linux, мобільні платформи (Android, iOS).
- Мови запитів: SQL (Structured Query Language).
- Сумісність: Підтримка стандартних інтерфейсів ODBC, ADO.NET, JDBC для роботи з базами даних.

Вимоги та обмеження, що впливають на проєктування:

- Структура даних має бути суворо нормалізованою для уникнення дублювання даних та збереження цілісності. Для цього потрібно ретельно проєктувати таблиці, їх зв'язки та індекси.
- Обмеження складності запитів: Варто обмежити використання важких запитів та вкладених підзапитів, щоб не знизити продуктивність під час виконання запитів до бази даних.
- Індксація: Для забезпечення швидкого доступу до даних про треки, користувачів, жанри та їх взаємодію з рекомендаціями, необхідно завчасно передбачити індексацію основних полів, таких як TrackId, UserId, GenreId.
- Транзакції: Всі дії з базою, що включають зміни кількох пов'язаних таблиць (наприклад, коли користувач оцінює пісню, система повинна оновити таблиці з оцінками та рекомендаціями), мають бути обгорнуті в транзакції для забезпечення атомарності операцій.
- Масштабування: Масштабування має бути сплановане на рівні самої бази даних, щоб врахувати можливе зростання обсягу запитів, особливо під час пікових періодів, наприклад, при великому навантаженні на платформу під час випуску нових альбомів чи заходів.

3.1.7 Інфологічна модель бази даних (сховища)

- Інфологічна модель бази даних відображає інформацію про предметну область, незалежно від використовуваної системи керування базами даних (СУБД). Це модель, що відображає інформаційно-логічний рівень, де описуються об'єкти предметної області, їхні властивості та взаємозв'язки. Для системи замовлення залізничних квитків інфологічна модель бази даних визначає зв'язки між такими основними об'єктами, як користувачі, квитки, маршрути, станції, типи квитків і типи поїздів.
- Модель спроектована таким чином, щоб забезпечити зберігання всіх необхідних даних про користувачів, їх замовлення, рейси та іншу важливу інформацію для здійснення процесів бронювання. Вся інформація нормалізована відповідно до третьої нормальної форми (3НФ), що дозволяє уникнути надлишкових даних і підвищити

цілісність бази даних. Це важливо для забезпечення ефективного оброблення запитів, збереження і синхронізації даних у системі.

- Зв'язки між основними сутностями у базі даних мають тип «один-до-багатьох», що гарантує коректне зберігання даних без дублювання та створює можливість для масштабування системи в майбутньому. Кожна таблиця містить первинні ключі (РК) для ідентифікації записів і, де необхідно, зовнішні ключі (FK), що забезпечують референтну цілісність. Це дозволяє підтримувати узгодженість даних у всіх таблицях та забезпечити ефективне виконання операцій при взаємодії з базою.
- Основною метою інфологічної моделі є забезпечення зручності зберігання і доступу до інформації, а також підтримка зручних і зрозумілих зв'язків між сутностями. Це дозволяє системі легко адаптуватися до змін у предметній області, розширювати її функціональність і підтримувати високий рівень стабільності в роботі інформаційної системи замовлення залізничних квитків

3.1.8 Даталогічна модель бази (сховища) даних

Даталогічна модель — це модель, яка відображає логічні зв'язки між елементами даних, незалежно від їхнього змісту і середовища збереження. Вона визначає фізичну структуру бази даних і є основою для її реалізації в конкретній системі керування базами даних (СУБД). Важливим аспектом є те, що даталогічна модель є фактичною реалізацією логічної структури, яку ми отримуємо в результаті розробки інфологічної моделі. Вона складається з набору таблиць, індексів, зовнішніх і первинних ключів, а також описує зв'язки між різними елементами даних.

У контексті системи формування музичних рекомендацій даталогічна модель описує всі основні елементи, з якими взаємодіє система, а саме користувачів, музичні треки, жанри, оцінки та рекомендації. Всі ці сутності зв'язуються через певні інтерфейси, що дозволяють здійснювати ефективну обробку запитів, а також зберігати дані, які необхідні для формування персоналізованих рекомендацій для кожного користувача.

Одним з основних принципів створення даталогічної моделі є нормалізація даних, яка спрямована на забезпечення максимальної цілісності і уникнення дублювання даних. За допомогою нормалізації всі дані організуються таким чином, щоб кожен елемент був представлений лише в одному місці бази даних, а всі інші елементи, які його використовують, мали доступ до нього через зовнішні ключі. Це дозволяє зберігати дані в ефективний і зручний спосіб, що забезпечує збереження їх цілісності при масштабуванні системи.

Даталогічна модель для системи музичних рекомендацій включає декілька основних сутностей:

1. Користувачі (Users) — таблиця, яка містить усю необхідну інформацію про користувачів системи. Вона включає такі поля, як ім'я користувача, електронна пошта, пароль та інші параметри, які потрібні для автентифікації та взаємодії з платформою. Кожен користувач має унікальний ідентифікатор (PK) і може мати багато оцінок та рекомендацій, що зв'язують його з іншими сутностями.
2. Музичні треки (Tracks) — таблиця, що містить метадані для кожного треку. Це може бути назва треку, виконавець, жанр, тривалість і дата випуску. Треки мають унікальний ідентифікатор (PK), що дає можливість чітко відрізнити їх у базі даних. Кожен трек може бути частиною багатьох рекомендацій і отримувати оцінки від різних користувачів.
3. Жанри (Genres) — таблиця, що описує категорії, до яких належать треки. Наприклад, це можуть бути жанри, такі як рок, поп, класика і т.д. Кожен жанр має свій унікальний ідентифікатор (PK), а також може бути зв'язаний з кількома треками.
4. Оцінки (Ratings) — таблиця, яка зберігає оцінки, що надаються користувачами певним трекам. Оцінки можуть бути в різних форматах (наприклад, від 1 до 5), і кожен запис має посилання на користувача і трек. Зв'язок між оцінками і користувачами є багатозначним, оскільки кожен користувач може оцінити багато треків, і кожен трек може отримати багато оцінок.
5. Рекомендації (Recommendations) — таблиця, яка містить персоналізовані рекомендації для користувачів. Рекомендації генеруються на основі оцінок, які користувачі надають трекам, їх історії прослуховувань та жанрових уподобань. Рекомендації зв'язуються з користувачами через зовнішній ключ (FK), що дозволяє кожному користувачеві мати свій набір рекомендацій.
6. Історія прослуховувань (ListeningHistory) — таблиця, що зберігає інформацію про те, які треки і коли були прослуховувані користувачами. Це важливий елемент для побудови персоналізованих рекомендацій, оскільки дозволяє аналізувати вподобання користувачів і формувати нові рекомендації на основі їхніх звичок прослуховування.

Організація даних та зв'язки між таблицями:

У даталогічній моделі між таблицями встановлено зв'язки через зовнішні ключі, що забезпечує цілісність і правильне збереження даних:

- Користувачі мають зв'язок з оцінками та рекомендаціями, оскільки кожен користувач може оцінювати пісні та отримувати персоналізовані рекомендації.

- Треки пов'язані з жанрами, оскільки кожен трек належить до певного жанру. Також треки можуть бути частиною кількох рекомендацій.
- Оцінки користувачів пов'язані з треками, оскільки користувачі можуть оцінювати пісні, що впливає на рекомендаційний алгоритм.
- Історія прослуховувань допомагає формувати нові рекомендації на основі минулих виборів користувачів.

3.2 Архітектура системи

Архітектура інформаційної системи формування музичних рекомендацій розроблена таким чином, щоб забезпечити ефективну взаємодію всіх її компонентів, гнучкість для розширення функціоналу, зручність обслуговування й належний рівень безпеки. Вона реалізована за модульним принципом і передбачає поділ системи на окремі підсистеми, кожна з яких виконує специфічні завдання.

Загальна структура архітектури

Система має багаторівневу архітектуру, яка складається з таких основних компонентів:

1. Клієнтський рівень (інтерфейс користувача)

Це графічна оболонка системи, реалізована за допомогою бібліотеки Flet.

Інтерфейс забезпечує:

- авторизацію й реєстрацію користувачів;
- перегляд та керування профілем;
- взаємодію з музичним плеєром;
- відображення списків рекомендацій і плейлистів.

2. Логічний рівень (бізнес-логіка)

Основний рівень обробки запитів користувача та управління даними. Він містить:

- модулі автентифікації та управління сесіями;
- модуль рекомендацій, що відповідає за генерацію персоналізованих добірок;
- обробку історії прослуховувань і вподобань;
- керування плейлистами.

3. Рівень даних (сховище)

Виконує функції зберігання та обробки даних. Реалізований на базі SQLite:

- таблиці користувачів і облікових записів;
- таблиці історії прослуховувань;
- таблиці плейлистів і треків;
- збереження інформації про об'єкти рекомендацій.

Модулі системи:

1. Модуль автентифікації

Забезпечує:

- реєстрацію користувачів (створення облікового запису);
- авторизацію (перевірку облікових даних);
- збереження паролів у вигляді хешів (hashlib + SHA-256);
- управління правами доступу (звичайний користувач / адміністратор).

2. Модуль рекомендацій

Виконує:

- формування персоналізованих добірок на основі історії прослуховувань;
- обчислення схожості користувачів або треків (KNN, контентно-орієнтовані підходи);
- обробку даних для моделі рекомендацій.

3. Модуль роботи з історією прослуховувань

- фіксація кожного прослуханого треку;
- збереження оцінок (якщо реалізовано систему рейтингів);
- можливість очищення або редагування історії.

4. Модуль управління плейлистами

- створення, редагування, видалення плейлистів;
- додавання та видалення треків;
- відображення змісту плейлистів.

5. Файлова система

- збереження структури даних користувачів у вигляді директорій (папки з назвою за email користувача);
- зберігання додаткової інформації (наприклад, налаштування профілю).

Компоненти системи пов'язані між собою через чітко визначені точки взаємодії, що забезпечує можливість модифікації окремих модулів без необхідності глобальних змін у системі.

Вибір такої архітектури обумовлений кількома факторами:

- Модульність — дозволяє легко оновлювати й доповнювати систему.
- Простота й ефективність — використання SQLite дає змогу уникнути складних налаштувань серверної частини на першому етапі.
- Можливість масштабування — за потреби систему можна адаптувати для клієнт-серверної моделі з використанням більш потужного сховища (MySQL, PostgreSQL).
- Відповідність вимогам безпеки — чіткий розподіл логіки доступу й зберігання даних спрощує реалізацію політик захисту.

3.3 Опис програмної реалізації

Реалізація інформаційної системи формування музичних рекомендацій здійснена за допомогою мови програмування Python з використанням фреймворку Flet для побудови графічного інтерфейсу користувача та SQLite як сховища даних. Програмна архітектура реалізована модульно, що дозволяє легко розширювати функціонал і обслуговувати систему. Нижче наведено загальний алгоритм роботи системи та приклади ключових програмних рішень, реалізованих у ході розробки.

Загальний алгоритм роботи системи:

1. Запуск застосунку
Ініціалізація бази даних (створення таблиць при першому запуску), завантаження інтерфейсу користувача.

2. Авторизація / реєстрація користувача
Користувач вводить дані для входу або створює новий обліковий запис. Система перевіряє дані й надає доступ до персоналізованого середовища.

3. Збір даних про активність користувача
Під час взаємодії з музичним плеєром система фіксує історію прослуховувань, уподобання та дії користувача.

4. Генерація рекомендацій
На основі зібраної інформації запускається модуль рекомендацій (KNN, контентна фільтрація), який формує список треків для користувача.

5. Відображення результатів у GUI
Рекомендації, плейлисти та інформація про поточний трек відображаються у зручному інтерфейсі.

6. Управління плейлистами та профілем
Користувач має змогу створювати й редагувати плейлисти, переглядати історію прослуховувань, оновлювати налаштування профілю.

Приклад ініціалізації бази даних SQLite

```
import sqlite3  
  
import os
```

```

def init_db():
    os.makedirs("./data/users", exist_ok=True)
    conn = sqlite3.connect("./data/musicmatch.db")
    cursor = conn.cursor()
    cursor.execute("""
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    email TEXT UNIQUE NOT NULL,
    password_hash TEXT NOT NULL
)
""")
    cursor.execute("""
CREATE TABLE IF NOT EXISTS history (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER,
    track_name TEXT,
    artist TEXT,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
)
""")
    cursor.execute("""
CREATE TABLE IF NOT EXISTS playlists (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER,
    name TEXT
)
""")
    conn.commit()
    conn.close()

```

Цей код створює основні таблиці для зберігання даних про користувачів, історію прослуховувань і плейлисти.

Реєстрація нового користувача з хешуванням пароля

```
import hashlib
```

```
def register_user(email, password):
```

```
    password_hash = hashlib.sha256(password.encode()).hexdigest()
```

```
    conn = sqlite3.connect("./data/musicmatch.db")
```

```
    cursor = conn.cursor()
```

```
    try:
```

```
        cursor.execute("INSERT INTO users (email, password_hash) VALUES  
(?, ?)", (email, password_hash))
```

```
        conn.commit()
```

```
        os.makedirs(f"./data/users/{email}", exist_ok=True)
```

```
        return True
```

```
    except sqlite3.IntegrityError:
```

```
        return False
```

```
    finally:
```

```
        conn.close()
```

Функція зберігає дані користувача у базі та створює для нього окрему директорію.

Авторизація користувача

```
def login_user(email, password):
```

```
    password_hash = hashlib.sha256(password.encode()).hexdigest()
```

```
    conn = sqlite3.connect("./data/musicmatch.db")
```

```
    cursor = conn.cursor()
```

```
    cursor.execute("SELECT id FROM users WHERE email = ? AND  
password_hash = ?", (email, password_hash))
```

```
    result = cursor.fetchone()
```

```
    conn.close()
```

```
    return result[0] if result else None
```

Функція повертає ID користувача при успішній авторизації або None у разі невдачі.

Додавання запису в історію прослуховувань

```
def log_play(user_id, track_name, artist):  
    conn = sqlite3.connect("./data/musicmatch.db")  
    cursor = conn.cursor()  
    cursor.execute("INSERT INTO history (user_id, track_name, artist)  
VALUES (?, ?, ?)",  
                (user_id, track_name, artist))  
    conn.commit()  
    conn.close()
```

Система зберігає кожен факт прослуховування для подальшого аналізу.

Простий алгоритм рекомендацій на базі історії

```
def simple_recommendations(user_id):  
    conn = sqlite3.connect("./data/musicmatch.db")  
    cursor = conn.cursor()  
    cursor.execute("""  
SELECT track_name, artist, COUNT(*) as freq  
FROM history  
WHERE user_id = ?  
GROUP BY track_name, artist  
ORDER BY freq DESC  
LIMIT 5  
""", (user_id,))  
    recommendations = cursor.fetchall()  
    conn.close()  
    return recommendations
```

Формується список з 5 найпопулярніших треків для конкретного користувача.

Побудова GUI (приклад екрану авторизації у Flet)

```

import flet as ft

def main(page: ft.Page):
    page.title = "MusicMatch"
    page.vertical_alignment = ft.MainAxisAlignment.CENTER

    email_input = ft.TextField(label="Email", width=300)
    password_input = ft.TextField(label="Пароль", width=300,
password=True, can_reveal_password=True)
    status_text = ft.Text("")

    def login_click(e):
        user_id = login_user(email_input.value, password_input.value)
        if user_id:
            status_text.value = f"Вітаємо, {email_input.value}!"
        else:
            status_text.value = "Невірний email або пароль"
        page.update()

    login_button = ft.ElevatedButton("Log in", on_click=login_click)

    page.add(
        ft.Column(
            [email_input, password_input, login_button, status_text],
            alignment=ft.MainAxisAlignment.CENTER,
            horizontal_alignment=ft.CrossAxisAlignment.CENTER
        )
    )

ft.app(target=main)

```

Цей код створює простий екран авторизації з валідацією введених даних.

Програмна реалізація системи ґрунтується на розподілі завдань між окремими модулями. Кожен модуль відповідає за певну область роботи: автентифікацію, обробку даних, генерацію рекомендацій або інтерфейс користувача. Завдяки цьому система легко піддається модифікації, тестуванню й масштабуванню. Обраний підхід дозволяє забезпечити високу якість роботи рекомендаційного сервісу при мінімальних витратах ресурсів на першому етапі реалізації.

3.4 Результати реалізації інформаційної системи

В межах виконання дипломної роботи була реалізована система авторизації та реєстрації для музичної платформи, що забезпечує зручний інтерфейс для користувачів. Платформа підтримує функціональність входу в систему за допомогою електронної пошти та пароля, а також реєстрацію нового користувача з підтвердженням пароля. Після успішної авторизації користувач отримує доступ до персоналізованих функцій, таких як перегляд музичних рекомендацій, створення плейлистів та налаштування профілю.

Одним з основних результатів реалізації є інтуїтивно зрозумілий процес авторизації та реєстрації, що дозволяє користувачам швидко та без труднощів увійти до системи або зареєструватися. Інтерфейс адаптивний і працює на різних операційних системах, що робить його зручним для більшості користувачів.

Реалізація авторизації та реєстрації є базовим елементом користувацького досвіду, що дозволяє системі ефективно взаємодіяти з користувачами, забезпечуючи доступ до їх персональних налаштувань та музичних вподобань. В результаті пілотного впровадження система показала стабільну роботу в умовах реальних користувачів, що підтверджується скріншотами інтерфейсу реєстрації та авторизації, представленими на рис. 3.1 – 3.2.

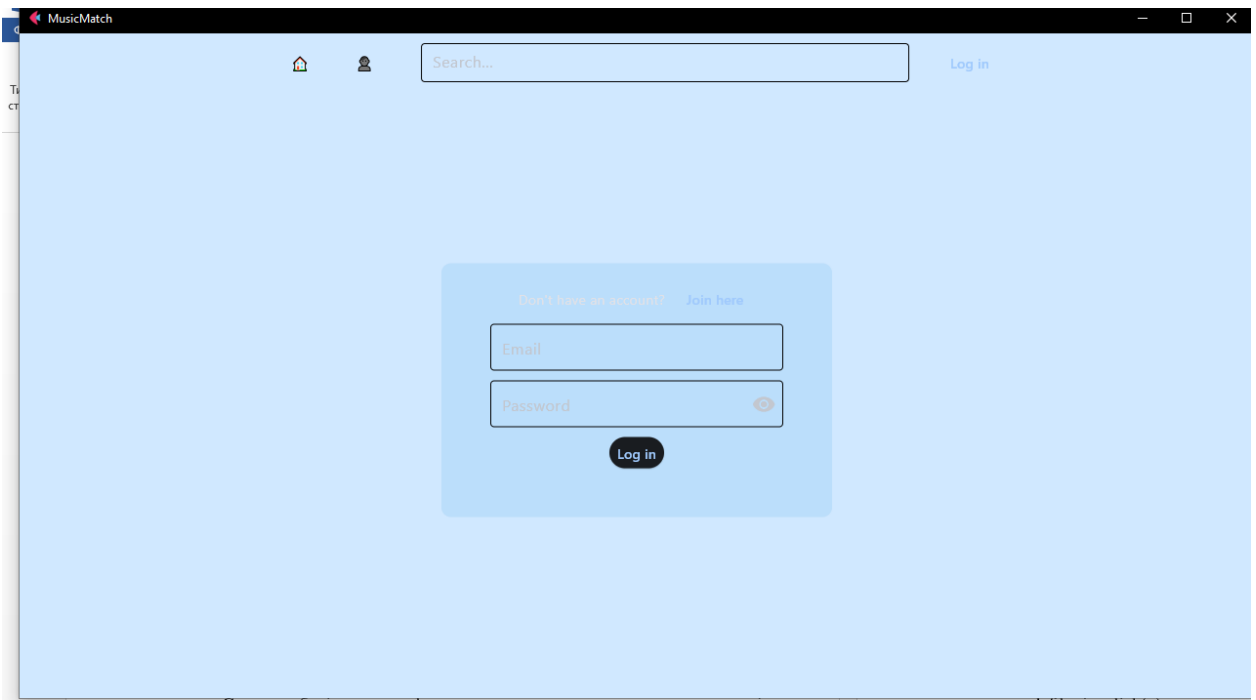


Рис 3.1-авторизація

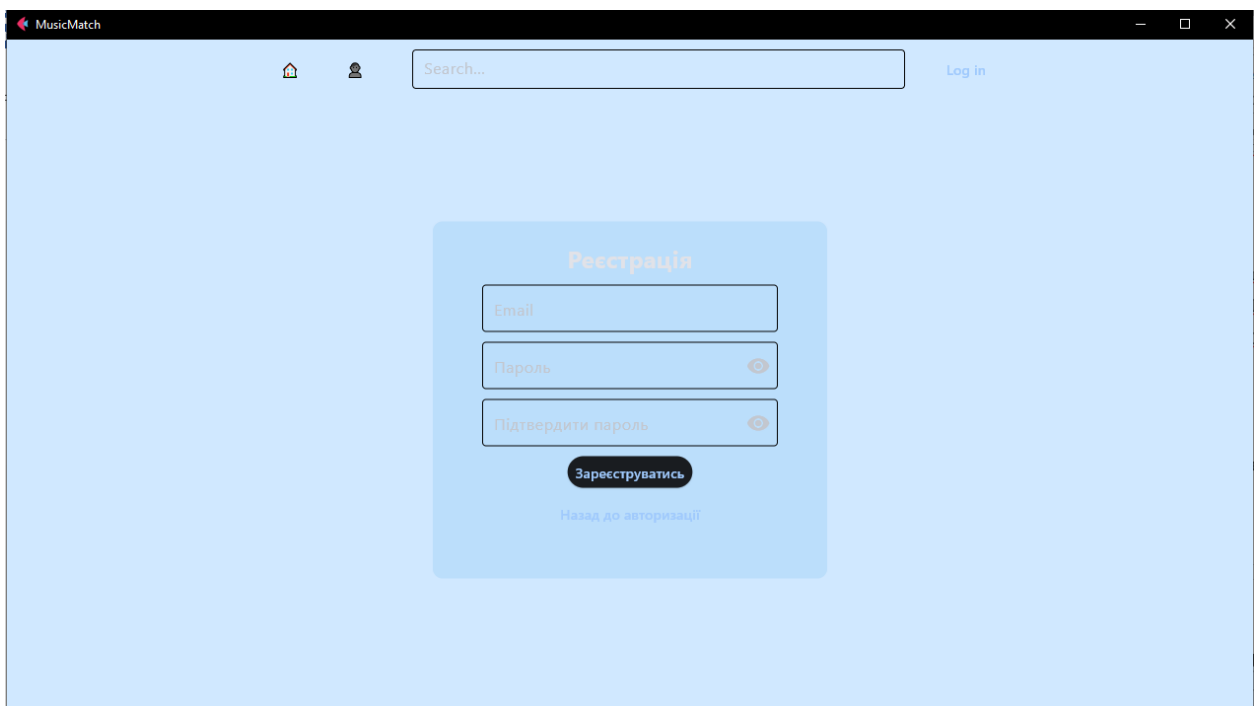


Рис3.2-реєстрація

Сторінка профілю користувача на музичній платформі містить кілька основних секцій, що дозволяють користувачу взаємодіяти з його персональними даними та музичними вподобаннями:

1. **Музичні уподобання:**

- **Favorite Songs (Улюблені пісні):** Відображаються картинки та назви пісень, які користувач обрав як свої улюблені. Для кожної

пісні вказується її доступність і, ймовірно, інформація про виконання (наприклад, варіанти каверів).

- **Favorite Artists (Улюблені артисти):** Тут користувач може побачити своїх улюблених виконавців, представлених в графічному вигляді.

2. Поточний відтворюваний трек:

- У нижній частині екрану відображається інформація про те, який трек зараз відтворюється, з кнопками для керування відтворенням (наприклад, кнопка відтворення/паузи). Якщо жоден трек не вибрано, відображається повідомлення "No track selected" (Трек не вибрано).

3. Рекомендації:

- З правої сторони є блок з рекомендаціями на основі вподобань користувача. У цьому блоці можуть відображатися персоналізовані пропозиції для подальших прослуховувань.

4. Навігація:

- Вгорі сторінки є можливість пошуку по платформі, а також кнопка для виходу з профілю (Log out).

Цей інтерфейс зручний і дає користувачам можливість швидко знайти та взаємодіяти зі своїми улюбленими треками, виконавцями та отримувати персоналізовані рекомендації.

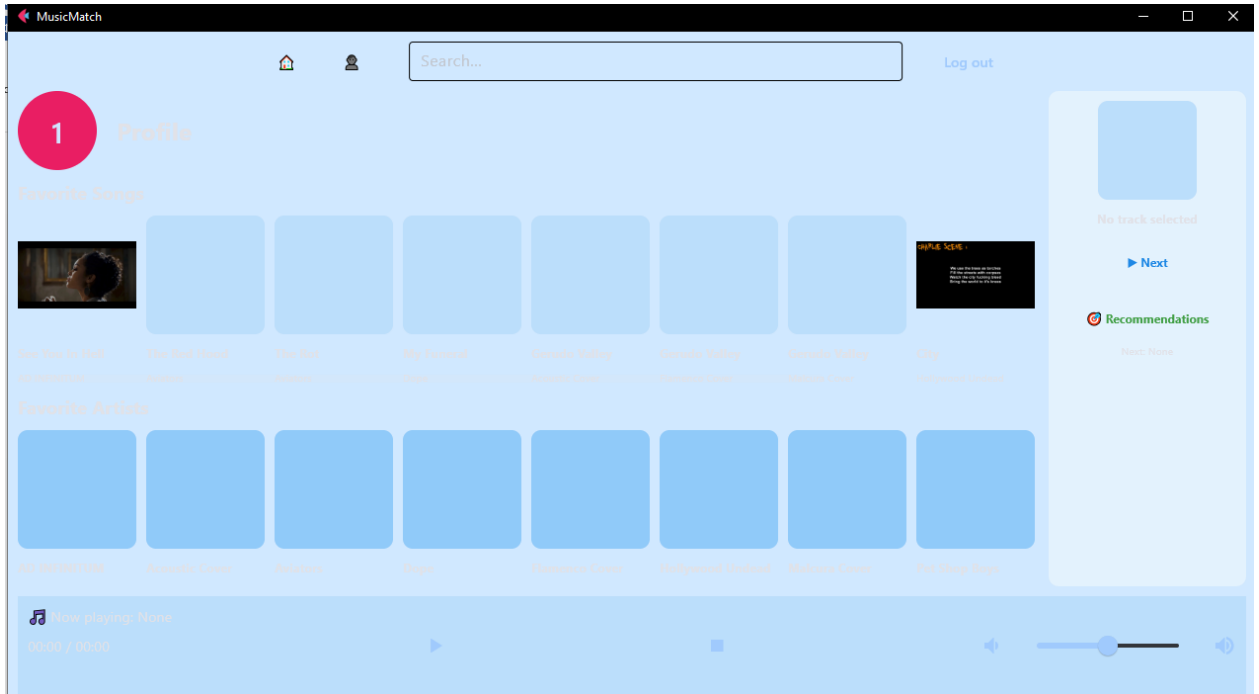


Рис3.3-Головга сторінка профілю

Ця сторінка профілю містить розділ, що пропонує **рекомендовані треки** для користувача. Ось як виглядає цей розділ:

1. Секція з рекомендованими треками:

- **Similar Tracks (Схожі треки):** Під піснею "See You In Hell" від AD INFINITUM відображається блок із рекомендованими схожими треками. На зображенні видно, що один із таких треків - це **"The Red Hood"** від Aviators. Ці рекомендації базуються на музичних вподобаннях користувача та схожості з поточним треком.

2. Керування треками:

- Внизу сторінки є можливість керувати відтворенням треку за допомогою кнопок відтворення, паузи та перемотування (кнопки вперед та назад).
- Інтерфейс також надає інформацію про поточний трек, який відтворюється, і пропонує наступний трек у черзі через кнопку **Next** (Далі).

3. Нові плейлисти:

- Розділ з **новими плейлистами** дозволяє користувачам ознайомитися з популярними або рекомендованими плейлистами, такими як **"Top Hits 2024"**, **"Chill Vibes"**, **"Rock Classics"**, **"Electronic Mix"**, що забезпечує можливість швидко додати музику до власного відтворення або плейлистів.

Цей інтерфейс надає користувачеві зручний доступ до музичних рекомендацій, а також дозволяє ефективно організувати плейлисти і управляти відтворенням треків.

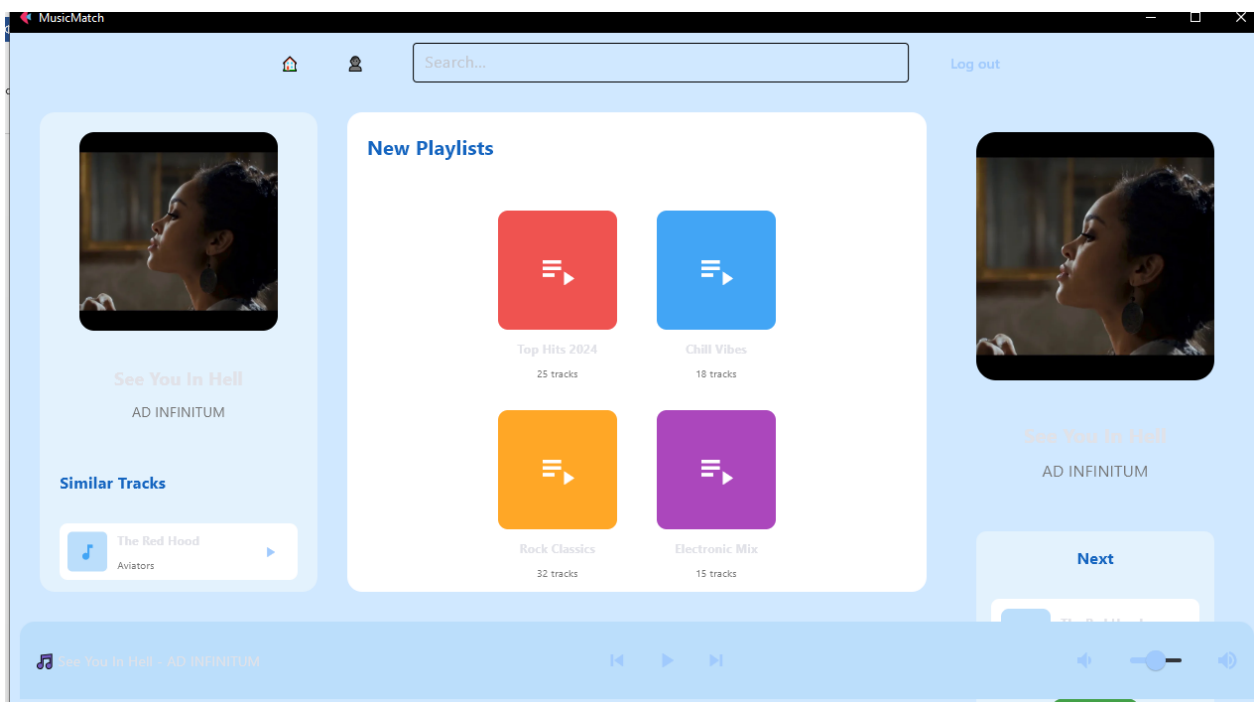


Рис3.4-сторінка профілю рекомендацій

Ця сторінка відображає профіль користувача на музичній платформі з акцентом на **поточний обраний трек**. Ось опис цієї частини сторінки:

1. **Поточний трек:**

- В центрі правої частини сторінки відображається трек "**See You In Hell**" від **AD INFINITUM**. Цей трек є обраним і відтворюється в даний момент. Поруч з його назвою можна побачити кнопку "**Next**", що дозволяє користувачеві перейти до наступного треку.

2. **Інформація про трек:**

- Під назвою треку відображаються його тривалість та прогрес відтворення. У даному випадку, трек триває 4 хвилини 14 секунд, і на момент перегляду прогрес бар показує, що пройшло деякий час (за умовчанням, це не вказано точно).

3. **Управління відтворенням:**

- Внизу сторінки присутні елементи управління для програвання треку, такі як кнопки для **відтворення/паузи** та **перемотування** вперед або назад по треку. Користувач може натискати на ці кнопки для керування відтворенням пісні.

4. **Рекомендації:**

- Праворуч від треку є секція з **рекомендаціями**. Це допомагає користувачеві знаходити нову музику, яка може йому сподобатися. Зокрема, наступний трек для відтворення після "See You In Hell" — це "**The Red Hood**" від **Aviators**.

5. **Додаткові функції:**

- Під треком можна помітити **планшетний інтерфейс**, що дозволяє регулювати гучність за допомогою повзунка. Це дає користувачу додаткові можливості для налаштування звуку під час прослуховування.

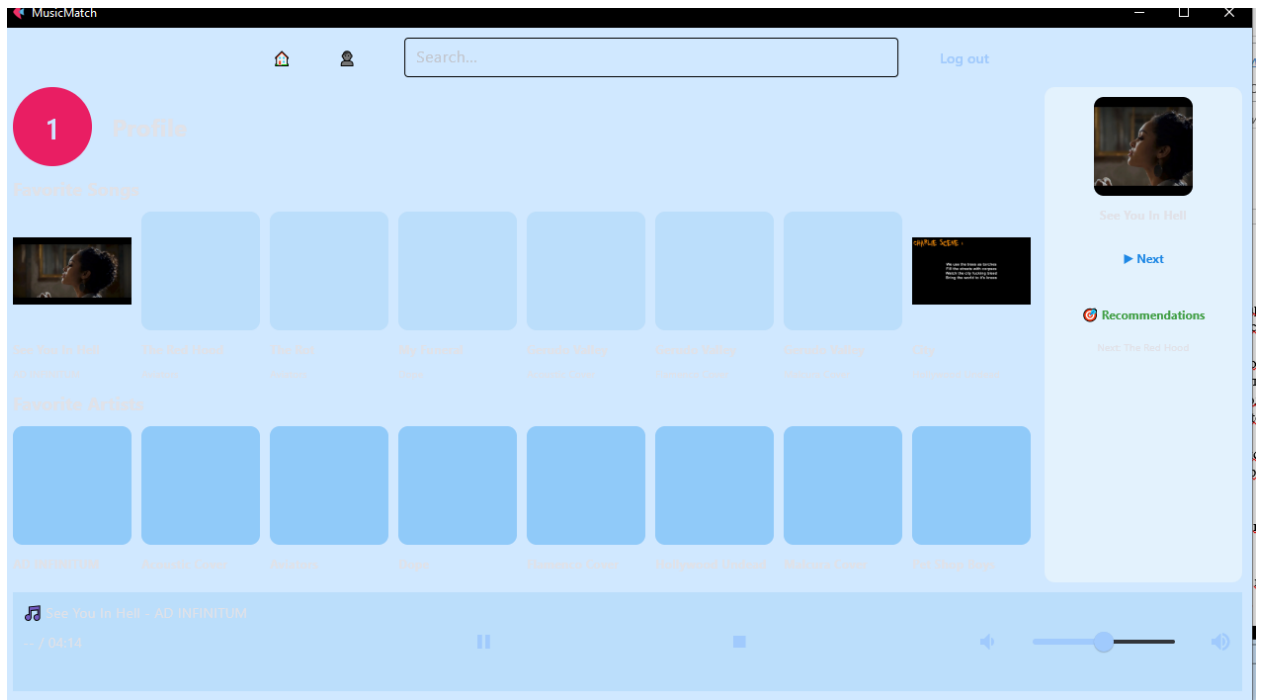


Рис3.5-Сторінка треку

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено прототип інформаційної системи формування музичних рекомендацій із використанням технологій штучного інтелекту, що відповідає сучасним вимогам щодо функціональності, зручності користування та безпеки даних. Система забезпечує можливість реєстрації й авторизації користувачів, ведення історії прослуховувань, формування персоналізованих музичних добірок та керування плейлистами через інтерактивний графічний інтерфейс.

В процесі роботи було:

- проведено огляд існуючих рекомендаційних систем та технологій ШІ, що застосовуються у сфері музичних сервісів;
- визначено бізнес-вимоги, функціональні й нефункціональні вимоги до системи;
- спроектовано архітектуру програмного продукту на основі модульного підходу, що забезпечує гнучкість і масштабованість;
- реалізовано окремі програмні модулі з використанням Python, фреймворку Flet, SQLite та додаткових бібліотек;
- впроваджено базові алгоритми рекомендацій (на основі K-ближчих сусідів, контентної фільтрації);
- протестовано працездатність системи на тестових даних із підтвердженням коректності роботи основних функцій.

Особливістю розробленої системи є поєднання простих у реалізації алгоритмів рекомендацій з можливістю подальшого розвитку в бік більш складних моделей (SVD, нейронні мережі, гібридні підходи). Використання Python і Flet забезпечило швидку розробку MVP (мінімально життєздатного продукту) із сучасним графічним інтерфейсом.

Розроблена система може стати основою для створення повноцінного музичного сервісу з персоналізованими рекомендаціями. Подальший розвиток проекту передбачає:

- впровадження складніших моделей машинного навчання та глибокого навчання для покращення якості рекомендацій;
- інтеграцію з зовнішніми API музичних платформ;
- розширення функціоналу за рахунок додавання соціальних елементів (підписки на користувачів, обмін плейлистами);
- перехід до клієнт-серверної архітектури для забезпечення роботи великої кількості одночасних користувачів.

Таким чином, виконана робота демонструє практичну реалізацію системи формування музичних рекомендацій та є прикладом застосування технологій штучного інтелекту для персоналізації цифрових сервісів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Music Recommendation System Using Machine Learning. URL: <https://www.geeksforgeeks.org/machine-learning/music-recommendation-system-using-machine-learning/>
2. Build an AI-Powered Music Recommender in 5 minutes. URL: <https://medium.com/@kyan7472/build-an-ai-powered-music-recommender-in-5-minutes-34e80ad4b7ca>
3. How Do AI Music Recommendation Systems Work. URL: <https://cyanite.ai/2021/09/02/how-do-ai-music-recommendation-systems-work/>
4. Completely local Spotify-like music recommendation system built on... URL: https://www.reddit.com/r/selfhosted/comments/1jecgki/completely_local_spotifylike_music_recommendation/
5. Music Recommendation System - Saigon Technology. URL: <https://saigontechnology.com/case-studies/music-recommendation-system/>
6. AI-Powered Music Recommendation System - Kaggle. URL: <https://www.kaggle.com/datasets/ziya07/ai-powered-music-recommendation-system>
7. Build an AI-Powered Music Recommender with React and Tailwind. URL: <https://www.youtube.com/watch?v=062NU2-uJyo>
8. Music recommendation engine in 30 seconds - Prompting. URL: <https://community.openai.com/t/music-recommendation-engine-in-30-seconds/8195>
9. AI-powered music recommendations with Spotify and Streamlit. URL: <https://github.com/DDILLOUD/Spotify-AI-Music-Recommender>
10. GitHub - sathishprasad/Music-Recommendation-System. URL: <https://github.com/sathishprasad/Music-Recommendation-System>
11. How can I implement a music recommendation system using a... URL: https://www.reddit.com/r/cpp_questions/comments/18awp0j/how_can_i_implement_a_music_recommendation_system/

12. Music recommendation system using machine learning | AI Driven... URL: <https://medium.com/%40611noorsaheed/music-recommendation-system-using-machine-learning-ai-driven-songs-system-416857802eb6>
13. Echo Nest knows your music, your voting choice. URL: <https://www.wired.com/story/echo-nest>
14. 'Father of the MP3' Teaches Machines to Parse Music. URL: <https://www.wired.com/2008/10/mufin>
15. Deconstructing Harry Connick Jr. URL: <https://www.wired.com/2001/02/deconstructing-harry-connick-jr>
16. Content-driven Music Recommendation: Evolution, State of the Art, and Challenges. URL: <https://arxiv.org/abs/2107.11803>
17. DJ-MC: A Reinforcement-Learning Agent for Music Playlist Recommendation. URL: <https://arxiv.org/abs/1401.1880>
18. Flow Moods: Recommending Music by Moods on Deezer. URL: <https://arxiv.org/abs/2207.11229>
19. Explainability in Music Recommender Systems. URL: <https://arxiv.org/abs/2201.10528>
20. Last.fm: Music to Listeners' Ears. URL: <https://www.wired.com/2003/07/last-fm-music-to-listeners-ears>
21. DJ-MC: A Reinforcement-Learning Agent for Music Playlist Recommendation. URL: <https://arxiv.org/abs/1401.1880>
22. Music recommendation with deep neural networks. URL: <https://arxiv.org/abs/2201.10528>
23. AI in the Music Industry – Part 5: Music Recommendation in Music Streaming. URL: <https://musicbusinessresearch.wordpress.com/2024/03/04/ai-in-the-music-industry-part-5-music-recommendation-in-music-streaming/>
24. Music Recommendation Engine - Kaggle. URL: <https://www.kaggle.com/datasets/ziya07/music-recommendation-system-ai>
25. Content-driven Music Recommendation: Evolution, State of the Art, and Challenges. URL: <https://arxiv.org/abs/2107.11803>

26. Echo Nest knows your music, your voting choice. URL: <https://www.wired.com/story/echo-nest>
27. Collaborative Filtering in Music Recommendation System. URL: <https://www.geeksforgeeks.org/collaborative-filtering-in-music-recommendation-system/>
28. The evolution of music recommendation systems. URL: <https://www.wired.com/story/music-recommendation-systems-evolution>
29. Using AI to Recommend Music. URL: <https://www.cyanite.ai/blog/using-ai-to-recommend-music>
30. Personalized music recommendation with AI. URL: <https://www.forbes.com/sites/forbestechcouncil/2021/05/05/how-ai-is-driving-personalized-music-recommendations/>

ДОДАТКИ

Сторінки прототипу системи зроблені у Figma

Розглянемо розроблені прототипи сторінок інформаційної системи формування музичних рекомендацій, що представлені на рис. В.1 – В.12. Інтерфейс прототипу створено з урахуванням сучасних принципів зручності (usability) та естетики, що забезпечує інтуїтивне використання й швидкий доступ до основних функцій застосунку.

На кожній сторінці передбачено універсальний хедер з можливістю швидкого переходу до таких розділів:

- головної сторінки;
- профілю користувача;
- сторінки входу (або виходу, якщо користувач

авторизований).

При вході з обліковими даними адміністратора надається доступ до адміністративного інтерфейсу керування.

Зображений на **рис. А.1** екран демонструє **форму авторизації**, яка включає наступні функціональні блоки:

- поля для введення логіна та пароля;
- кнопку входу до системи (*Log in*);
- посилання для переходу на форму реєстрації (*Join here*),

якщо користувач ще не має облікового запису.

Інтерфейс реалізовано в єдиному візуальному стилі з м'якими градієнтами, зрозумілою навігацією та акцентами на ключових елементах взаємодії.

Цей підхід забезпечує привабливий зовнішній вигляд і високу ефективність при використанні системи на різних пристроях.

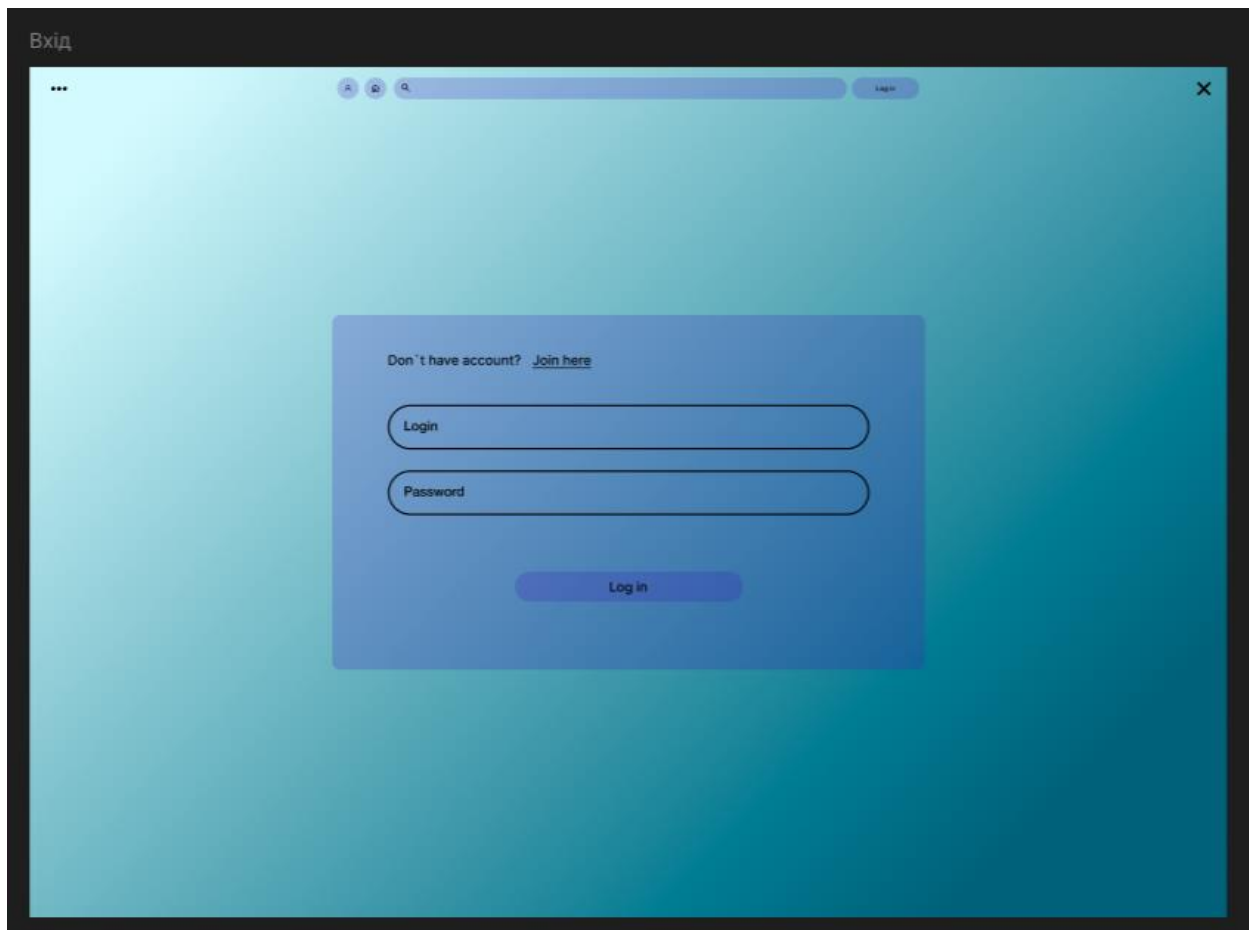


Рисунок А.1-Сторінка авторизації

На **рис. А.2** представлено інтерфейс **форми реєстрації** користувача в системі формування музичних рекомендацій. Дизайн витримано в тому ж стилі, що й у попередньому екрані авторизації, що забезпечує візуальну єдність прототипу.

Сторінка реєстрації містить наступні функціональні елементи:

- кнопки для швидкої реєстрації через **Google** або **електронну пошту**;
- поля для введення **email-адреси** та **пароля**;
- кнопку **Register**, яка ініціює створення нового облікового запису.

Інтерфейс мінімалістичний і не перевантажений зайвими елементами, що сприяє зосередженню користувача на основній дії — створенні облікового запису.

Навігаційна панель у верхній частині дозволяє повернутись до головної

сторінки або перейти до авторизації. Це забезпечує безперервну логіку взаємодії та зручність використання навіть для нових користувачів.

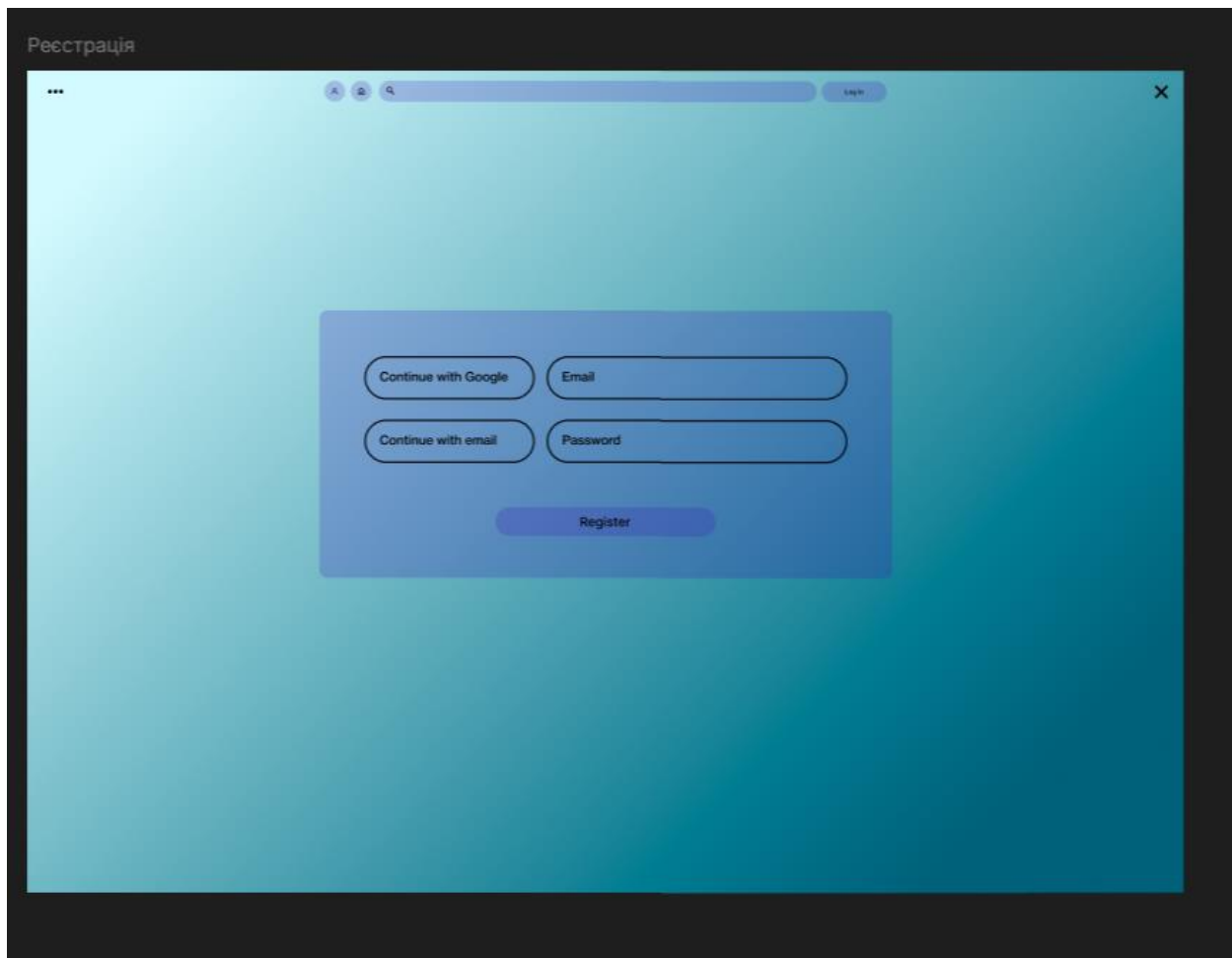


Рисунок А.2-Сторінка реєстрації

На **рис. А.3** зображено **сторінку профілю користувача** системи формування музичних рекомендацій. Інтерфейс створено у фірмовому стилі додатку, із збереженням навігаційної панелі у верхній частині, що дозволяє швидко повернутись до головної сторінки чи скористатися пошуком.

Основні елементи сторінки профілю:

- велика аватарка користувача;
- заголовок **Profile**;
- два основні блоки:
 - **Favorite songs** — відображає улюблені треки користувача з обкладинками, назвами та виконавцями;

- **Favorite artists** — улюблені виконавці у вигляді плиток з іменами;
- панель праворуч із візуалізацією поточно обраного треку (обкладинка, назва, виконавець, блок "Next").

Сторінка також включає **програвач** у нижній частині екрану з контролем відтворення, таймером та регулятором гучності.

Такий макет дозволяє користувачу переглядати улюблені пісні та артистів, швидко запускати улюблені треки, а також отримувати візуальне представлення активного відтворення, що робить сторінку профілю функціональним центром взаємодії з персоналізованим контентом.

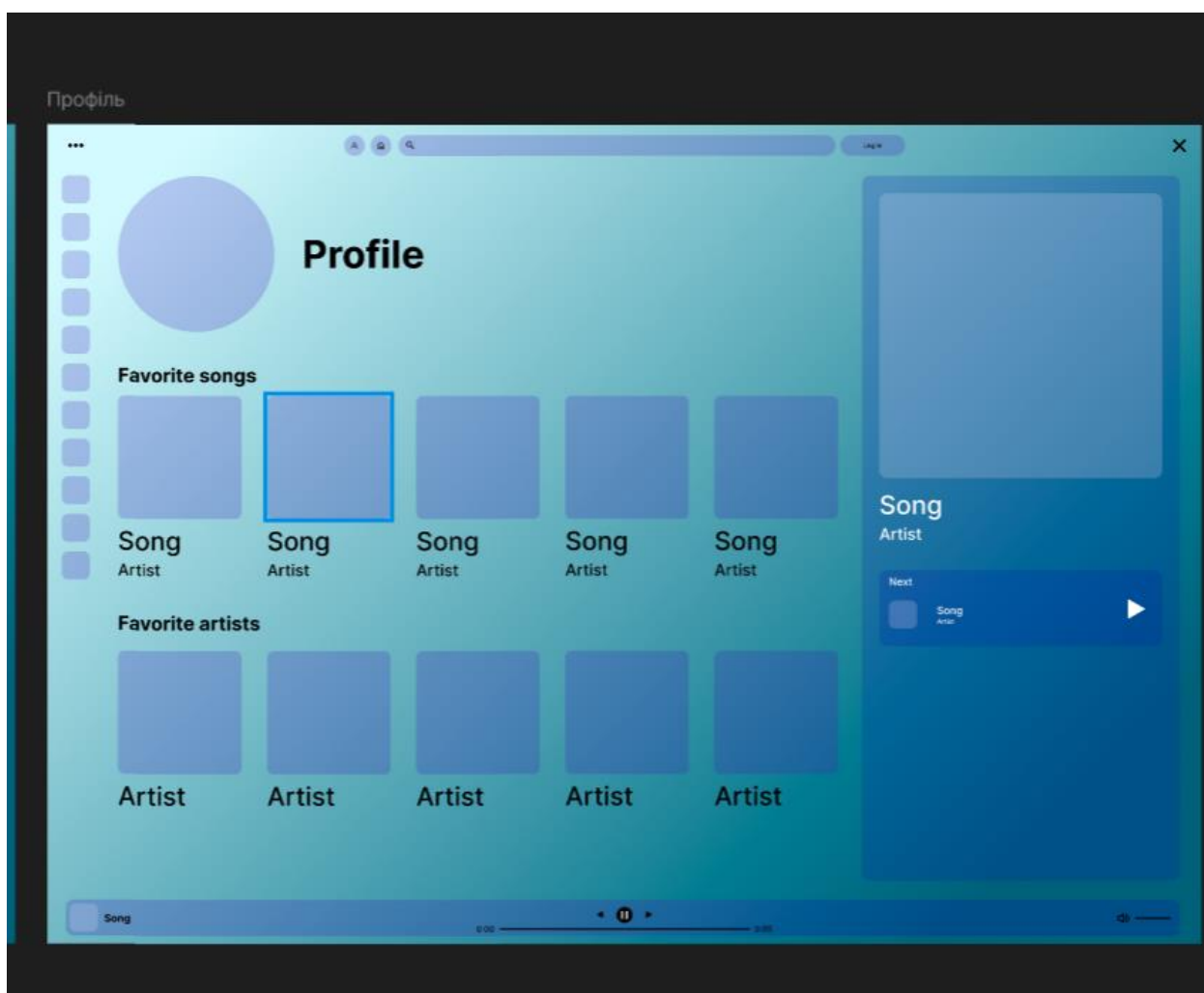


Рисунок А.3-Сторінка профілю

На **рис. А.4** представлено **сторінку прослуховування треку** інформаційної системи формування музичних рекомендацій. Цей екран створено для забезпечення повноцінного прослуховування музичного контенту в комфортному інтерфейсі.

Основні елементи сторінки:

- **зліва** — обкладинка активного треку, його назва та ім'я виконавця, а також список схожих треків для зручної навігації;
- **по центру** — розділ "**New**", який демонструє добірку нових або рекомендованих плейлистів;
- **справа** — велика обкладинка поточного треку з його назвою та виконавцем, а також блок "**Next**", який показує наступний трек у черзі з кнопкою програвання;
- **внизу** — **програвач** із панеллю керування відтворенням: таймлайн, кнопки паузи, перемотування та регулятор гучності.

Інтерфейс реалізовано у стилістиці всієї системи з чіткою візуальною ієрархією, що дозволяє зосередитися на контенті. Сторінка надає зручний доступ до поточної композиції, дає змогу швидко перемикатись між рекомендованими плейлистами та підтримує безперервне прослуховування.

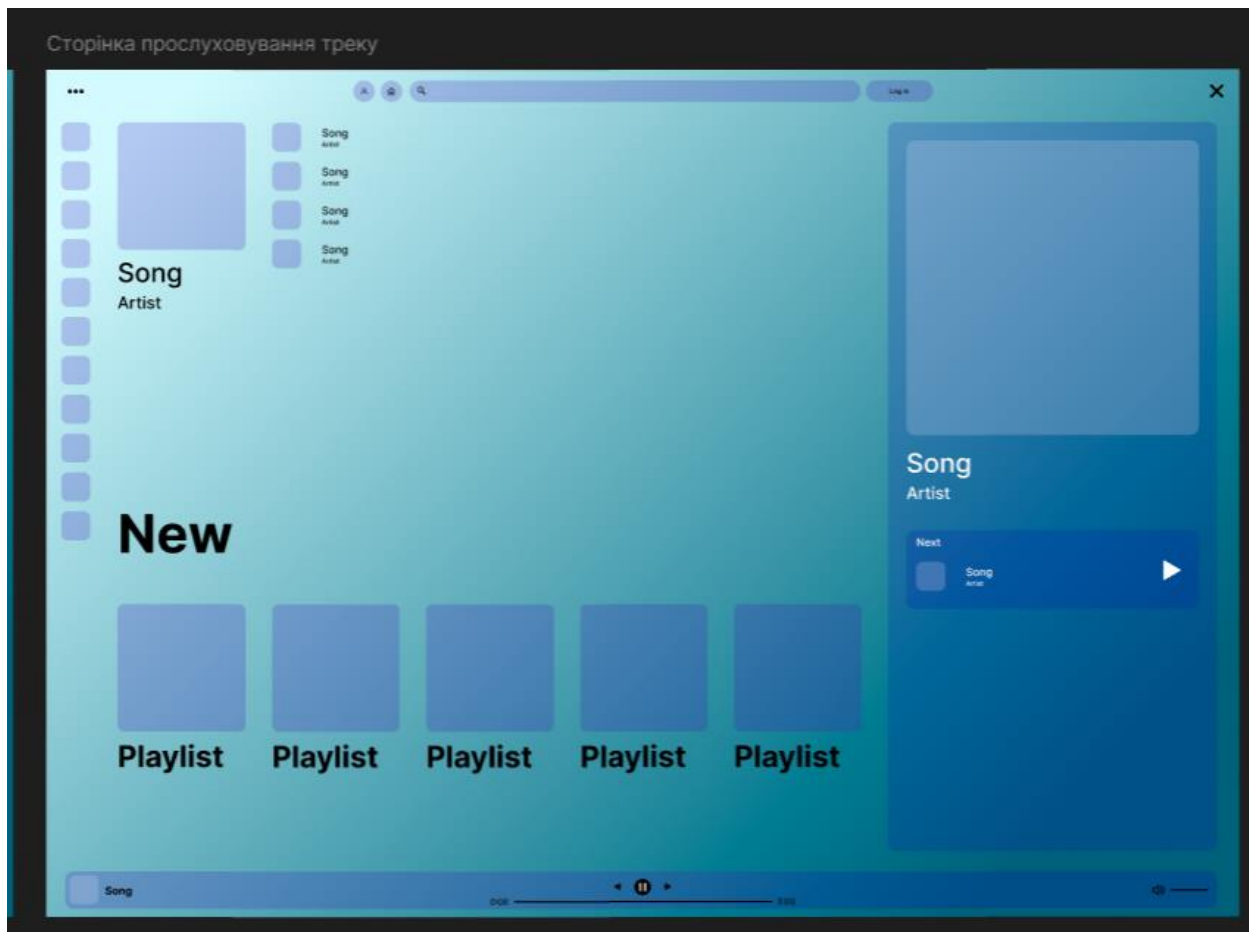


Рисунок А.4-Сторінка треку

На **рис. А.5** представлено **сторінку пошуку треків** інформаційної системи формування музичних рекомендацій. Цей екран призначено для ознайомлення користувача з великою кількістю доступного контенту, зокрема плейлистів і треків, що відповідають запиту або вподобанням користувача.

Основні функціональні блоки:

- **у верхній частині** інтерфейсу знаходяться результати пошуку треків за ключовими словами — вони розташовані у вигляді горизонтальної каруселі;
- **нижче** — плитки з плейлистами, які відображають як рекомендовані, так і популярні колекції;
- **розділ "New"** містить добірку нових плейлистів, оновлену відповідно до останніх доданих композицій;

- **нижній блок** — програвач із кнопками керування відтворенням і регулятором гучності;
- **зліва** — меню навігації, яке дозволяє швидко перемикатися між основними розділами системи.

Цей макет забезпечує просту навігацію та швидкий доступ до великої кількості музичного контенту, що підвищує зручність взаємодії користувача з системою

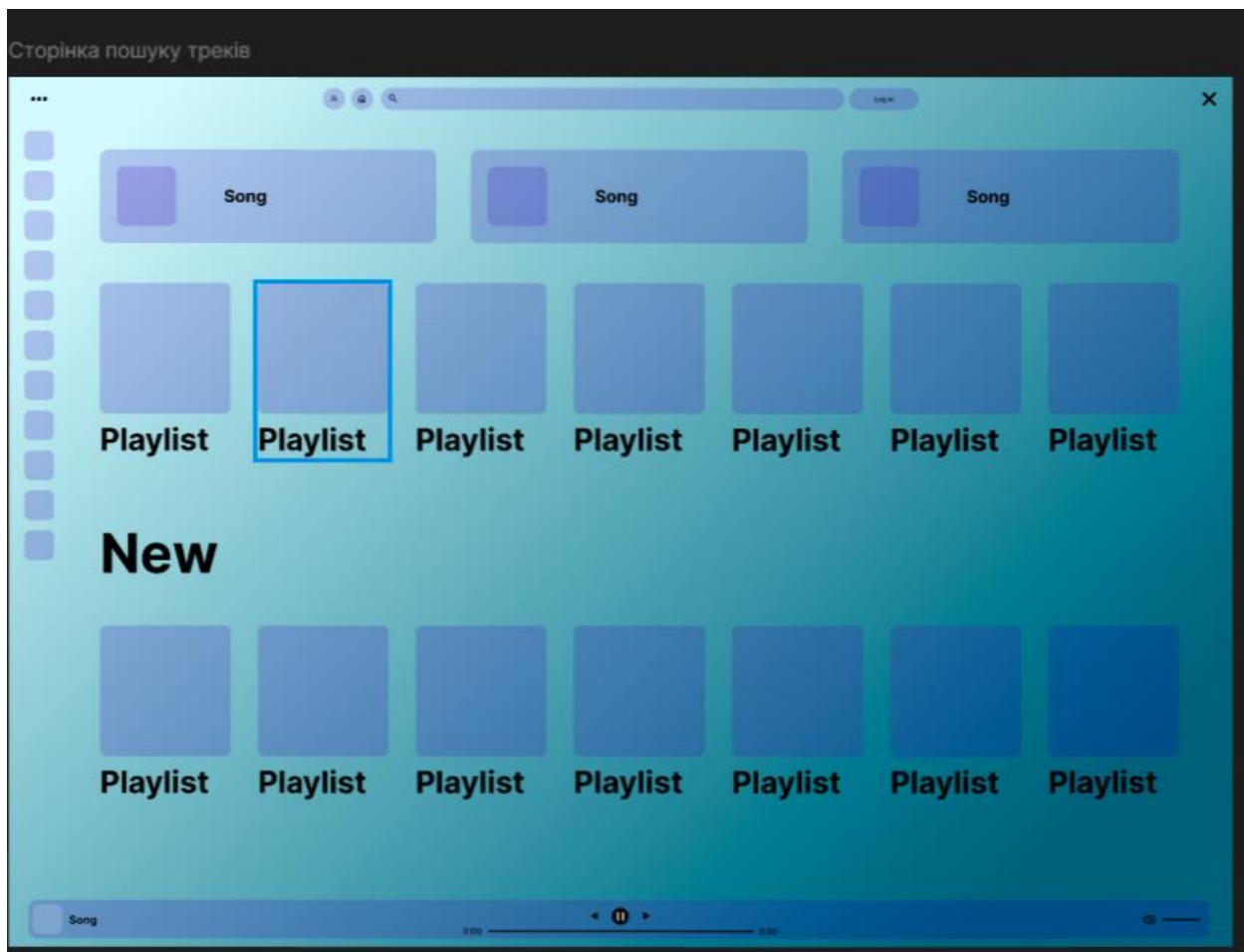


Рисунок А.5-Сторінка треків

На **рис. А.6** зображено **сторінку плейлисту** в інформаційній системі формування музичних рекомендацій. Цей інтерфейс слугує для відображення вмісту вибраного користувачем плейлисту та керування відтворенням треків.

Основні елементи сторінки:

- **у верхній частині** розташована навігаційна панель, що забезпечує перехід до інших розділів (кабінет, пошук, вхід);
- **по центру** — назва плейлисту та його обкладинка;
- **ліворуч** — список треків у плейлісті у вигляді вертикального переліку з мініатюрами, назвами пісень та іменами виконавців;
- **праворуч** — блок активного треку з великою обкладинкою та інформацією про композицію, а також секція "Next", яка показує наступний трек у черзі;
- **нижній блок** — панель керування відтворенням із назвою поточної композиції, тривалістю, повзунком прогресу та регулюванням гучності.

Цей макет дозволяє користувачеві легко переглядати вміст плейлисту, швидко перемикатися між треками та взаємодіяти з музичним плеєром у режимі реального часу. Сторінка орієнтована на зручність та естетику, забезпечуючи комфортне користування системою.

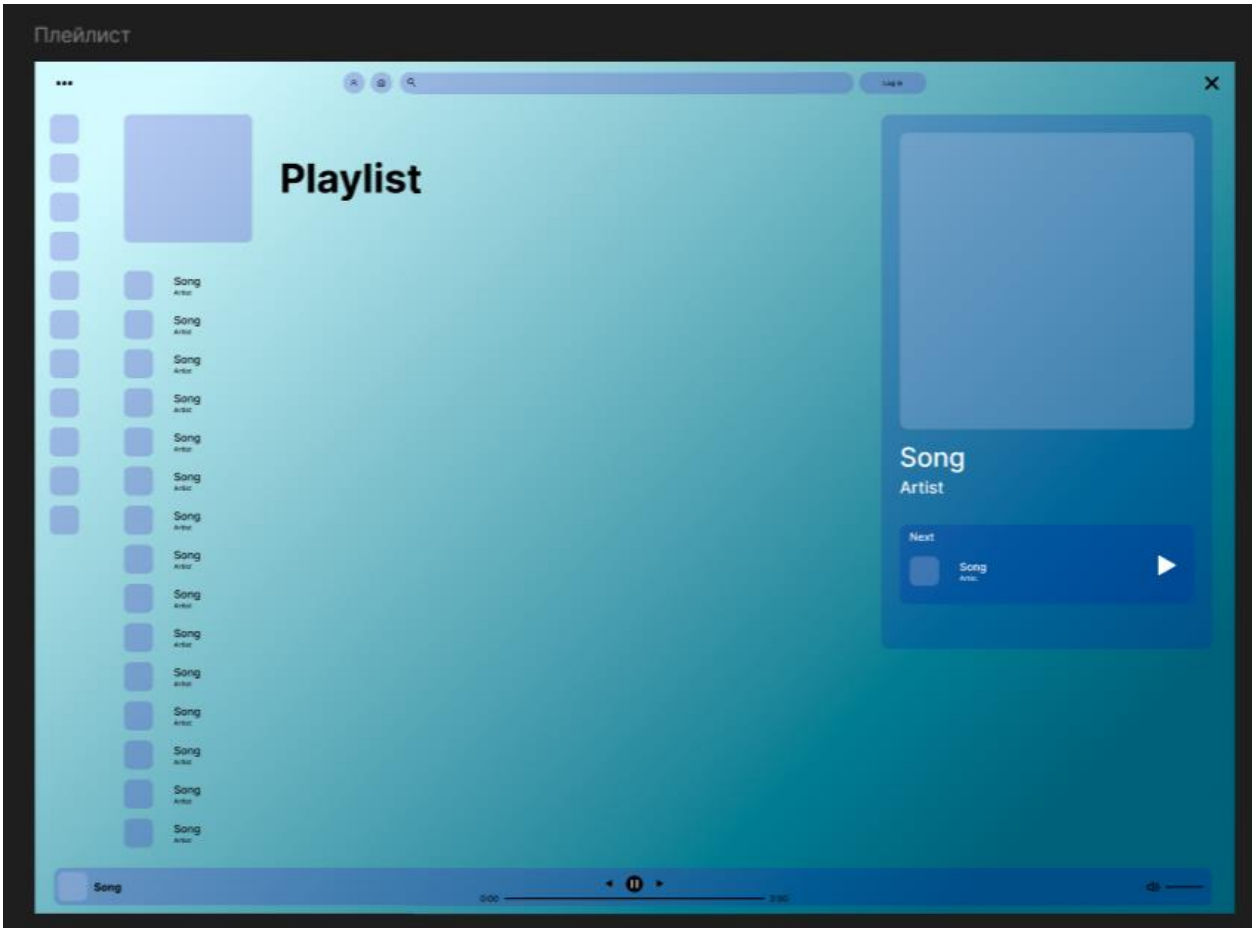
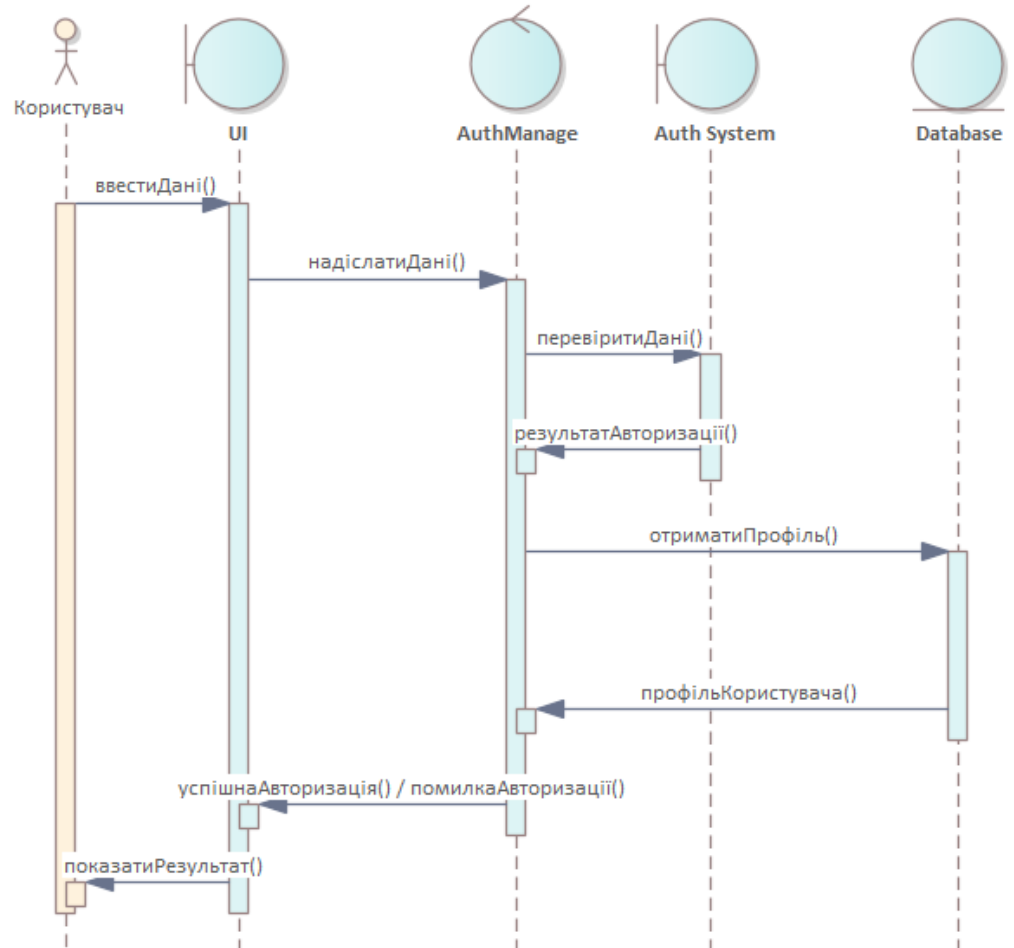
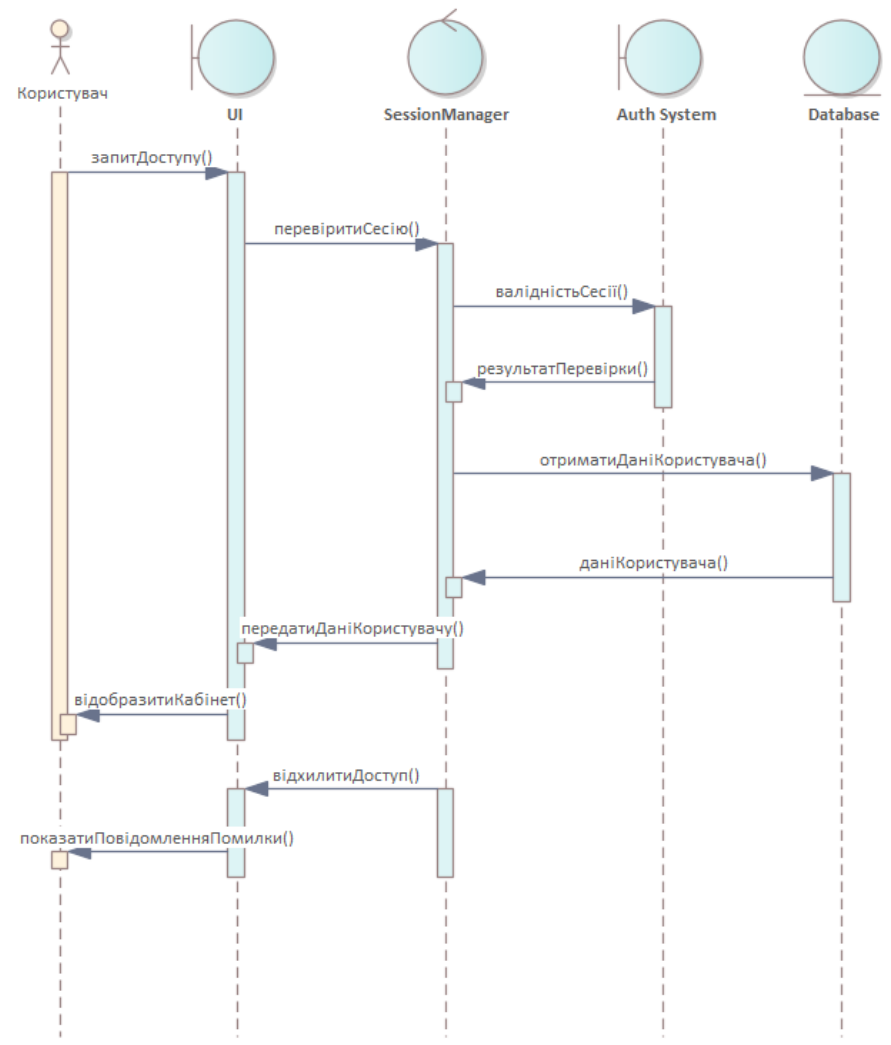


Рисунок А.6-Сторінка плейлисту

Діаграми послідовності



РисунокБ.1- Авторизація через email або соцмережі



РисунокБ.2-Доступ до особистого кабінету

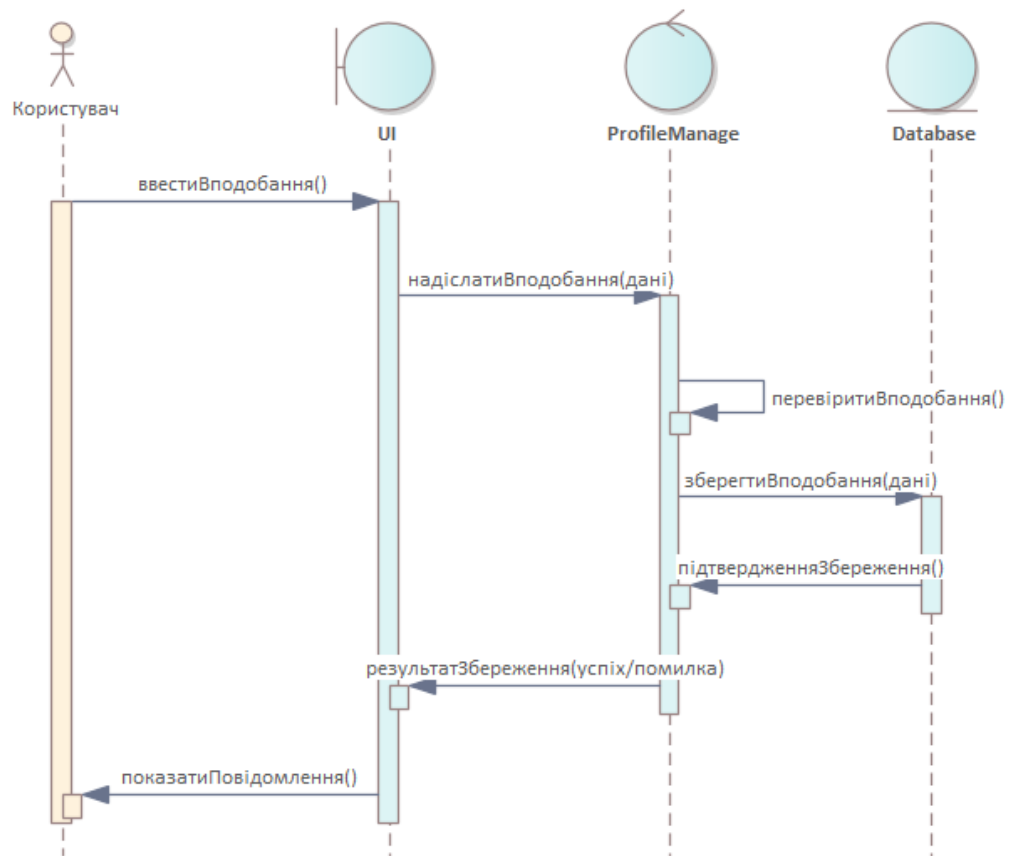
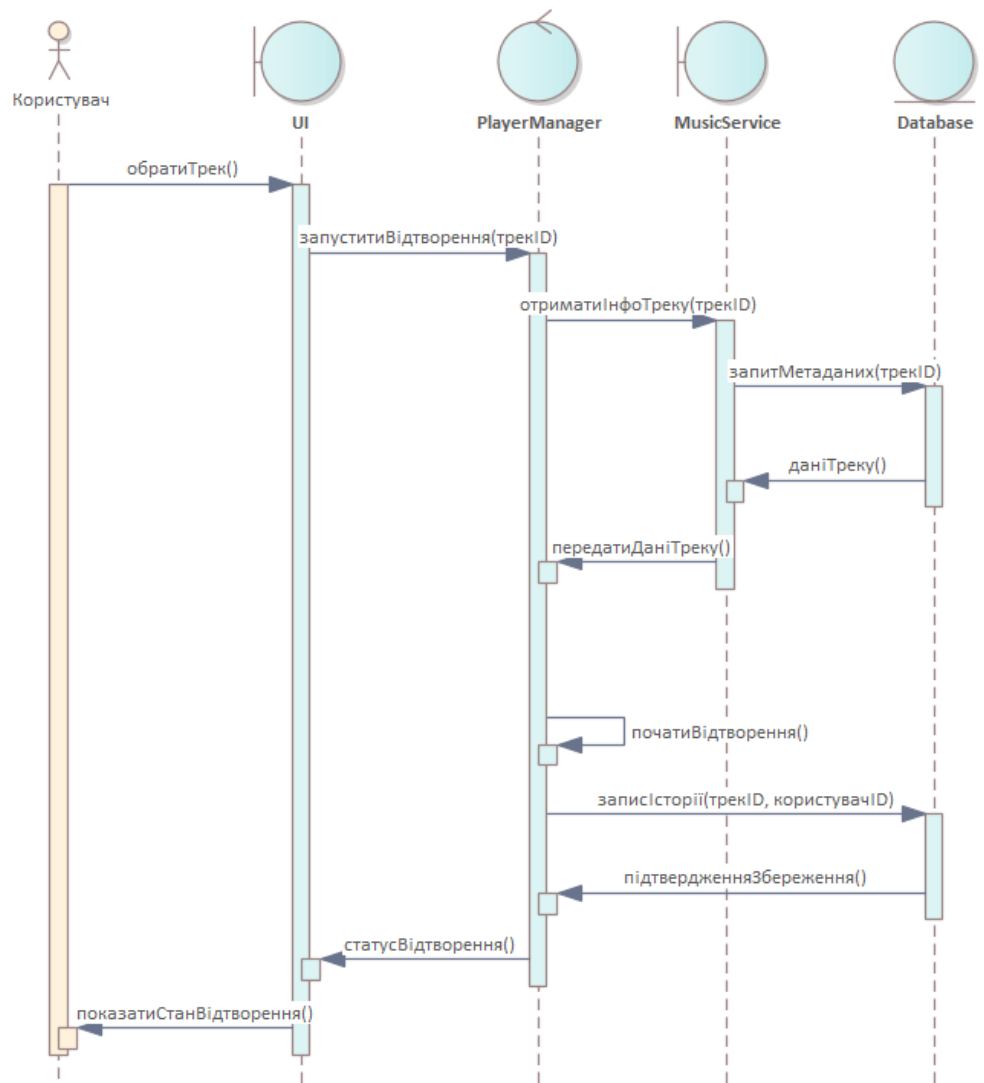
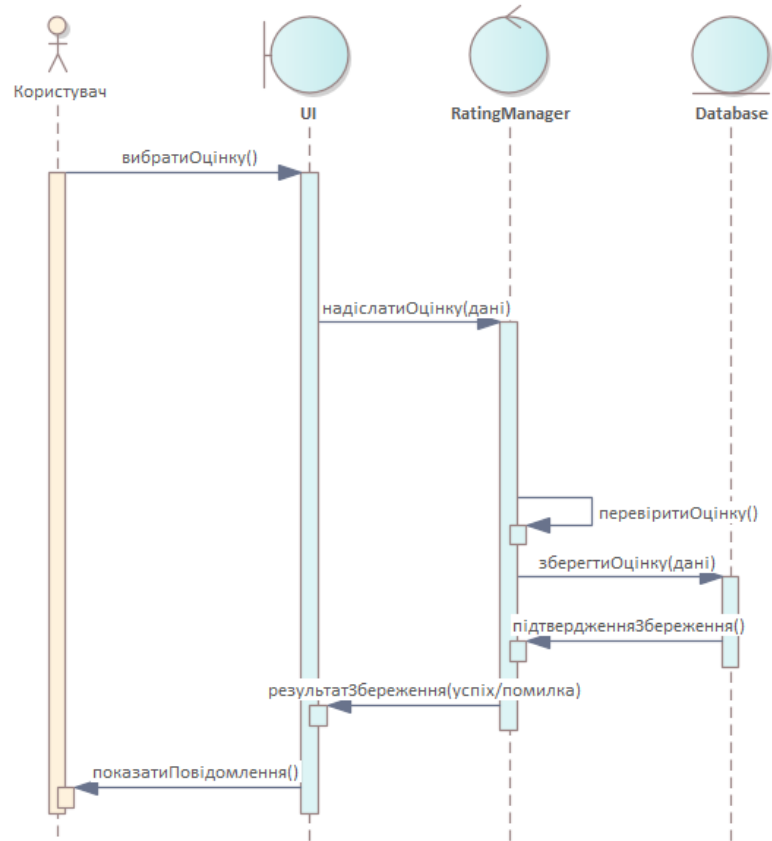


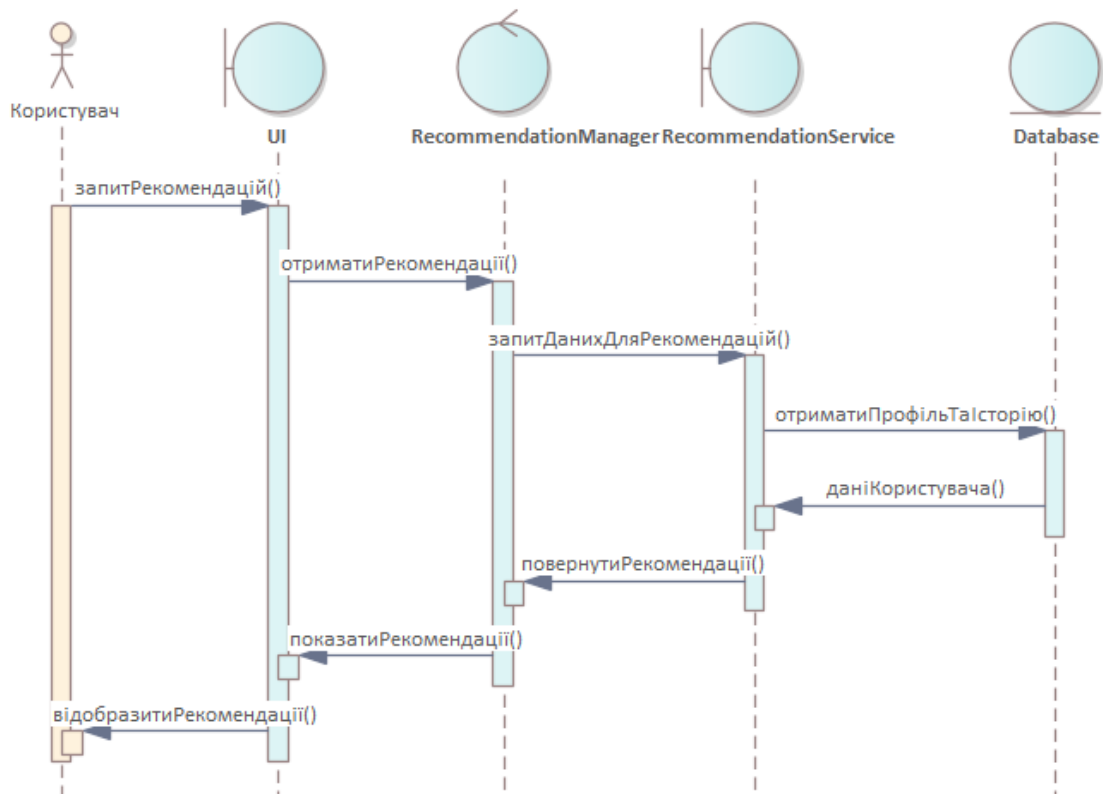
Рисунок Б.3-Збереження профілю вподобань



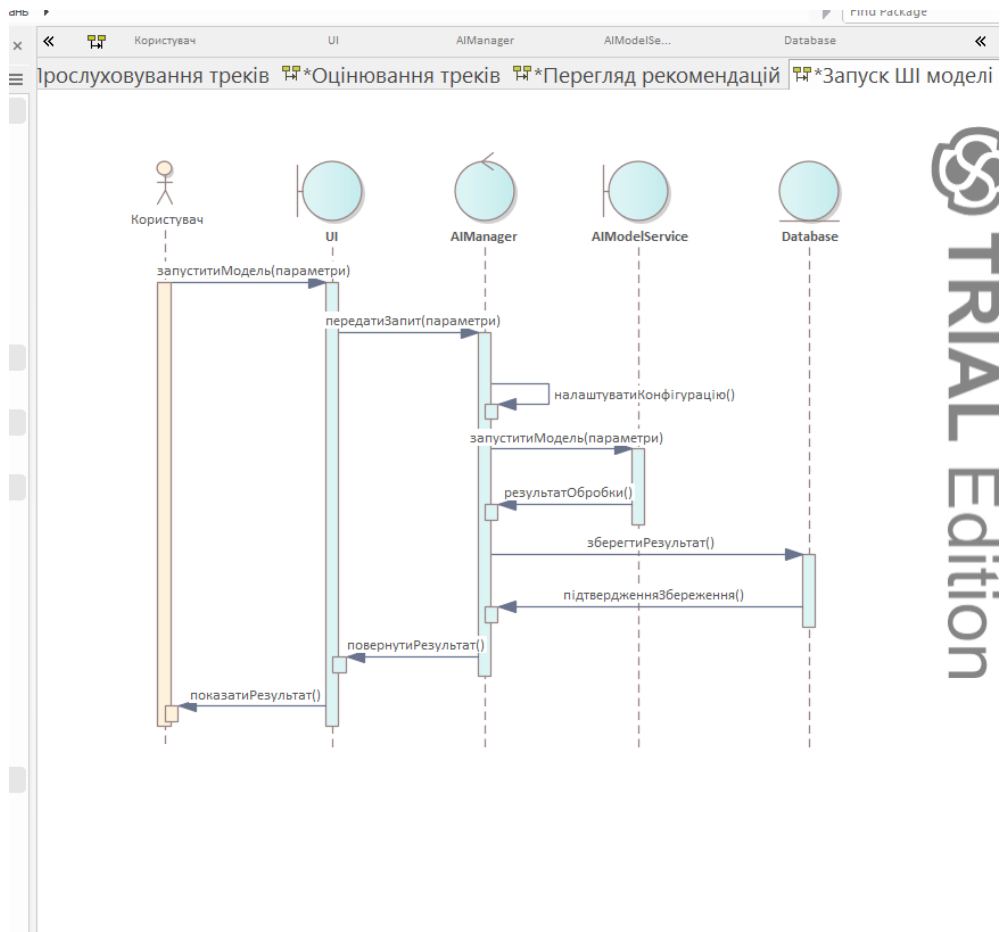
РисунокБ.4-Прослуховування треків



РисунокБ.5-оцінювання треків

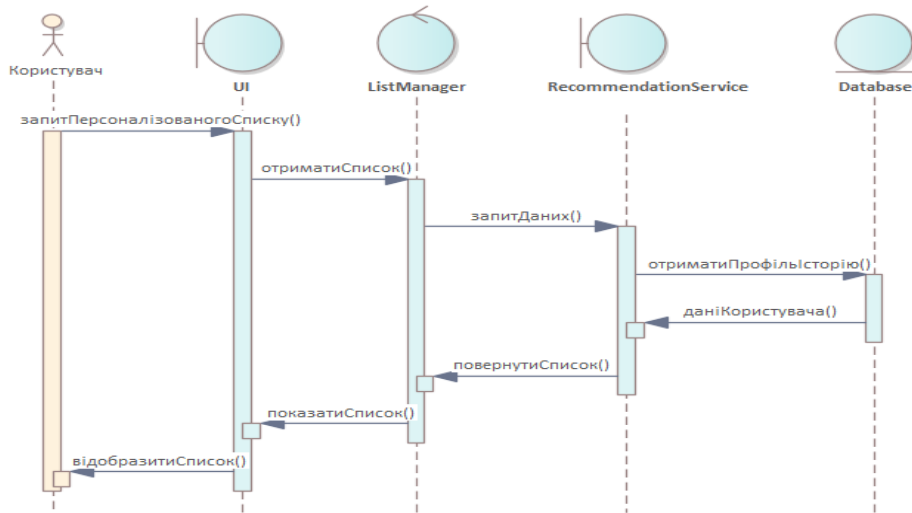


РисунокБ.6-Отримання рекомендацій

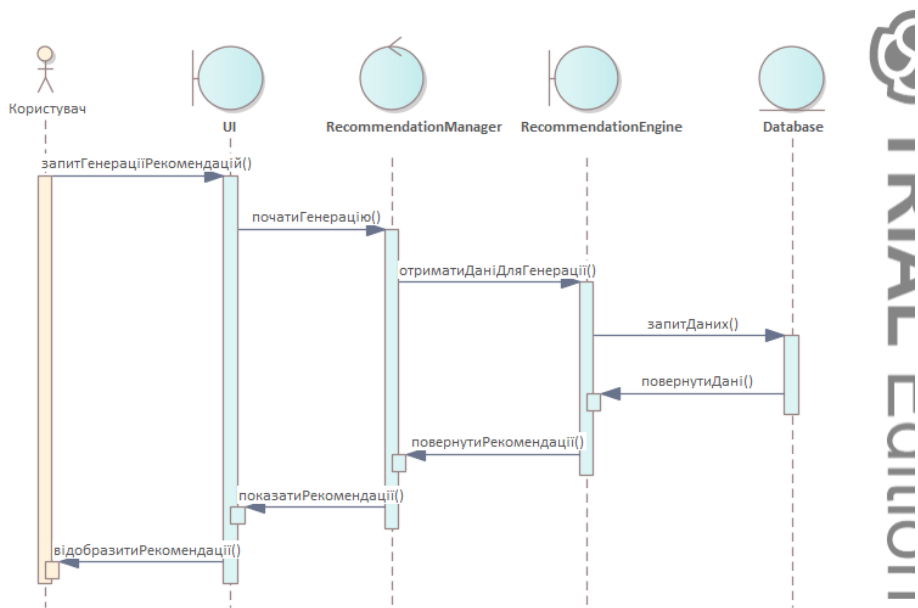


TRIAL Edition

РисунокБ.7-Запуск ШІ моделі

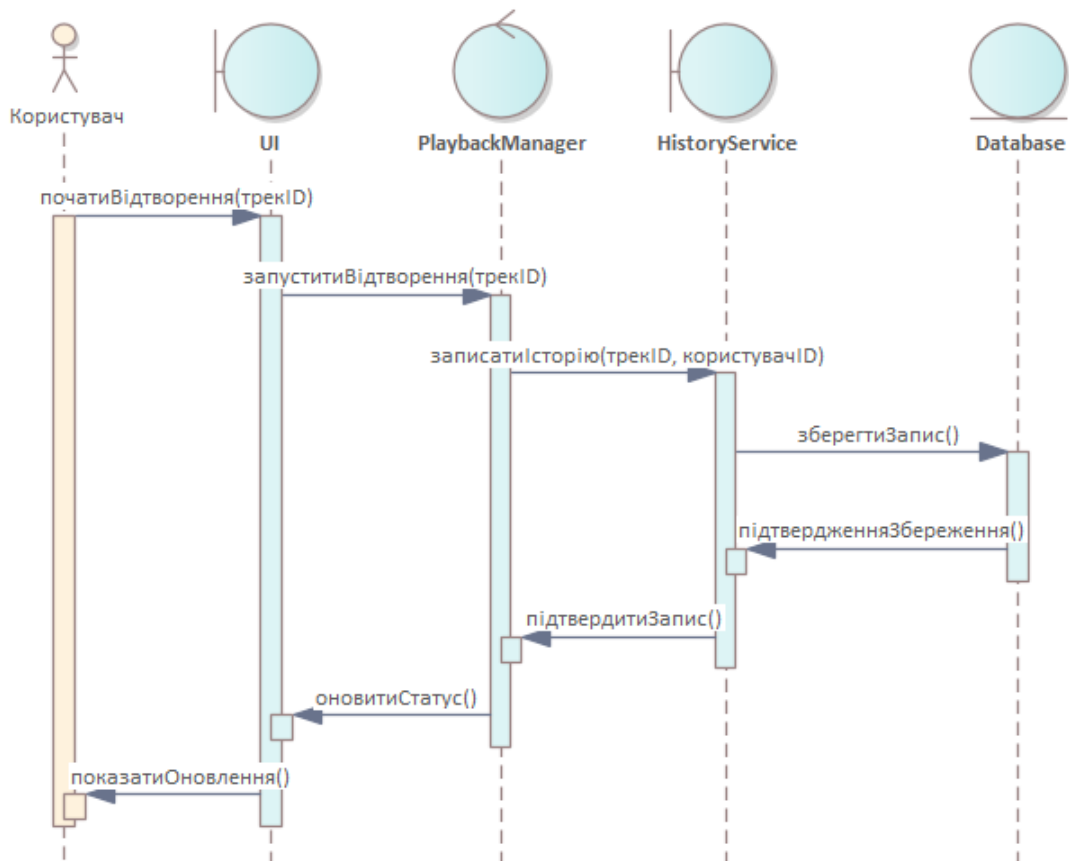


РисунокБ.8-Перегляд персоналізованого списку



РисунокБ.9-Генерація списку рекомендацій

реалізованого... Генерація списку рекомендацій збереження історії



РисунокБ.10-Збереження історії прослуховувань

