

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**До захисту допущено:**

**Завідувач кафедри**

\_\_\_\_\_ Сергій СТИРЕНКО

(підпис)

“ \_\_\_ ” \_\_\_\_\_ 2021 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою “Комп’ютерні системи та мережі”**

**спеціальності 123 “Комп’ютерна інженерія”**

на тему: Бібліотека музичних творів

Виконав: студент 4 курсу, групи ІО-72

Клецун Микита Сергійович

(прізвище, ім’я, по батькові)

(підпис)

Керівник ст. викладач Сімоненко А. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) д. т. н., проф. Сімоненко В. П.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент

(підпис)

Київ – 2021 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальність 123 “Комп’ютерна інженерія”

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри**

\_\_\_\_\_ Сергій СТИРЕНКО

(підпис)

“ \_\_\_ ” \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**

на бакалаврський дипломний проєкт студента

Клецуна Микити Сергійовича

1. Тема проєкту Бібліотека музичних творів

керівник проєкту \_\_\_\_\_ ст. викладач Сімоненко А.В.,  
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» травня 2021 р. №1139-с

2. Термін здачі студентом закінченого проєкту \_\_\_\_\_ 20 травня 2021 р.

3. Вихідні дані до проєкту: технічна документація. теоретичні дані

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)

1.Визначення та аналіз роботи музичних сервісів. 2.Обґрунтування, вибір архітектури, структури та програмного забезпечення. 3.Опис розробленої системи. 4.Тестування розробленої системи.

5. Перелік графічного матеріалу (з точним позначенням обов’язкових креслень) структурна схема системи, узагальнена схема роботи системи, схема алгоритму модуля редактора, схема \_\_\_\_\_ інформаційних вікон редактора

6. Консультанта проекту, з вказівкою розділів проекту, які до них вносяться

| Розділ        | Консультант    | Підпис, дата   |                  |
|---------------|----------------|----------------|------------------|
|               |                | Завдання видав | Завдання прийняв |
| Нормоконтроль | Сімоненко В.П. |                |                  |

7. Дата видачі завдання 7 вересня 2020

#### Календарний план

| № П/П | Найменування етапів дипломного проекту                     | Терміни виконання етапів проекту | Примітки |
|-------|--|----------------------------------|----------|
| 1.    | <i>Затвердження теми проекту</i>                           | <i>10.12.2020-15.12.2020</i>     |          |
| 2.    | <i>Вивчення та аналіз завдання</i>                         | <i>15.12.2020-15.03.2021</i>     |          |
| 3.    | <i>Розробка архітектури та загальної структури системи</i> | <i>15.03.2021-25.03.2021</i>     |          |
| 4.    | <i>Розробка структур окремих підсистем</i>                 | <i>25.03.2021-5.04.2021</i>      |          |
| 5.    | <i>Програмна реалізація системи</i>                        | <i>5.04.2021-15.04.2021</i>      |          |
| 6.    | <i>Оформлення пояснювальної записки</i>                    | <i>15.04.2021-20.05.2021</i>     |          |
| 7.    | <i>Захист програмного продукту</i>                         | <i>25.04.2021</i>                |          |
| 8.    | <i>Передзахист</i>   | <i>23.05.2021</i>                |          |
| 9.    | <i>Захист</i>  | <i>17.06.2021</i>                |          |

Студент-дипломник

(підпис)

Керівник проекту

(підпис)

## **Анотація**

Дана дипломна робота присвячена розробці бібліотеки музичних творів. Даний сервіс дає змогу користувачу безкоштовно прослуховувати музику, робити свої плейлисти з композиціями, ділитися улюбленими піснями в соціальних мережах, додавати до вподобаних та коментувати будь-яку музику. Користувач також може користуватися зручним пошуком, фільтрами або спеціальними тегами для швидкого знаходження бажаної композиції. В той же час, сервіс дає змогу виконавцю завантажувати та поширювати свою музику, отримувати відгуки, критику або похвалу. Адміністратор системи може видаляти будь-яку музику, альбом чи коментар. Отже, сервіс присвячений для безкоштовного прослуховування та поширення будь-якої музичної композиції.

## **Annotation**

This diploma thesis is devoted to the development of a digital music library. This service allows the user to listen to music for free, make their own playlists with songs, share favorite songs on social networks, add to favorites and comment on any music. The user can also use a convenient search, filters or special tags to quickly find the desired composition. At the same time, the service allows the artist to download and distribute their music, receive feedback, criticism or praise. The system administrator can delete any music, album or comment. Therefore, the service is dedicated to free listening and distribution of any musical composition.

**Опис альбому  
до дипломного проєкту  
на тему: «Бібліотека музичних творів»**

Київ – 2021 року

| № рядка | Формат | Позначення         | Найменування                 | Кількість | Примітка |
|---------|--------|--------------------|------------------------------|-----------|----------|
| 1       |        |                    | Завдання на дипломний проєкт | 2         |          |
| 2       |        |                    |                              |           |          |
| 3       | A4     | ІАЛЦ.467200.001 ОА | Опис альбому                 | 1         |          |
| 4       |        |                    |                              |           |          |
| 5       | A4     | ІАЛЦ.467200.002 ТЗ | Технічне завдання            | 3         |          |
| 6       |        |                    |                              |           |          |
| 7       | A4     | ІАЛЦ.467200.003 ПЗ | Пояснювальна записка         | 58        |          |
| 8       |        |                    |                              |           |          |
| 9       | A4     | ІАЛЦ.467200.004 Д1 | Додаток А                    | 1         |          |
| 10      |        |                    |                              |           |          |
| 11      | A4     | ІАЛЦ.467200.005 Д2 | Додаток Б                    | 1         |          |
| 12      |        |                    |                              |           |          |
| 13      | A4     | ІАЛЦ.467200.006 Д3 | Додаток С                    | 1         |          |
| 14      |        |                    |                              |           |          |
| 15      |        |                    |                              |           |          |
| 16      |        |                    |                              |           |          |
| 17      |        |                    |                              |           |          |
| 18      |        |                    |                              |           |          |
| 19      |        |                    |                              |           |          |
| 20      |        |                    |                              |           |          |
| 21      |        |                    |                              |           |          |
| 22      |        |                    |                              |           |          |
| 23      |        |                    |                              |           |          |
| 24      |        |                    |                              |           |          |

|           |      |                |        |      |  |                                  |      |         |
|-----------|------|----------------|--------|------|--|----------------------------------|------|---------|
|           |      |                |        |      | <i>ІАЛЦ.467200.001 ОА</i>                      |                                  |      |         |
| Змн.      | Арк. | № докум.       | Підпис | Дата |  |                                  |      |         |
| Розроб.   |      | Клецюн М.С.    |        |      | Бібліотека музичних творів<br><br>Опис альбому | Літ.                             | Арк. | Аркушів |
| Перевір.  |      | Сімоненко А.В. |        |      |  |                                  | 1    | 2       |
|           |      |                |        |      |  | <i>НТУУ «КПІ», ФІОТ<br/>Ю-72</i> |      |         |
| Н. Контр. |      | Сімоненко В.П. |        |      |  |                                  |      |         |
| Затверд.  |      |                |        |      |  |                                  |      |         |

**Технічне завдання  
до дипломного проєкту  
за темою: «Бібліотека музичних творів»**

Київ – 2021 року

# Технічне завдання до дипломної роботи

## ЗМІСТ

|   |   |
|---|---|
| 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ ..... | 2 |
| 2. ПІДСТАВИ ДЛЯ РОЗРОБКИ .....                | 2 |
| 3. МЕТА РОЗРОБКИ .....                        | 2 |
| 4. ДЖЕРЕЛА РОЗРОБКИ .....                     | 2 |
| 5. ТЕХНІЧНІ ВИМОГИ .....                      | 3 |
| 5.1 Вимоги до розроблюваної системи .....     | 3 |
| 5.2 Вимоги до програмного забезпечення .....  | 3 |
| 5.3 Вимоги до апаратного забезпечення .....   | 3 |
| 6. ЕТАПИ РОЗРОБКИ .....                       | 3 |

|                  |             |                       |               |             |   |                         |             |                |
|------------------|-------------|-----------------------|---------------|-------------|---|-------------------------|-------------|----------------|
|                  |             |                       |               |             | <i>ІАЛЦ.467200.002 ТЗ</i>   |                         |             |                |
| <i>Змн.</i>      | <i>Арк.</i> | <i>№ докум.</i>       | <i>Підпис</i> | <i>Дата</i> |   |                         |             |                |
| <i>Розроб.</i>   |             | <i>Клецюн М.С.</i>    |               |             | <i>Бібліотека музичних творів</i><br><br><i>Технічне завдання</i> | <i>Літ.</i>             | <i>Арк.</i> | <i>Акрушів</i> |
| <i>Перевір.</i>  |             | <i>Сімоненко А.В.</i> |               |             |   |                         | <i>1</i>    | <i>3</i>       |
| <i>Н. Контр.</i> |             | <i>Сімоненко В.П.</i> |               |             |   | <i>НТУУ «КПІ», ФІОТ</i> |             |                |
| <i>Затверд.</i>  |             |                       |               |             |   | <i>ІО-72</i>            |             |                |

## **1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ**

Дане технічне завдання розповсюджується на розробку бібліотеки музичних творів.

Область застосування: безкоштовне прослуховування та поширення музики онлайн.

## **2. ПІДСТАВИ ДЛЯ РОЗРОБКИ**

Підставою для розробки є завдання на створення онлайн системи для прослуховування та поширення музики.

## **3. МЕТА РОЗРОБКИ**

Метою даної розробки є створення бібліотеки для безкоштовного прослуховування та поширення музики.

## **4. ДЖЕРЕЛА РОЗРОБКИ**

Основними джерелами для розробки слугують публікації на дану тему в Інтернеті та документації існуючих програм з функціоналом для рішення цього питання.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.002 ТЗ</i> | <i>Арк.</i> |
|             |             |                 |               |             |                           | 0           |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           |             |

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1 Вимоги до розроблюваної системи

- Безкоштовне прослуховування музики
- Безкоштовне завантаження та поширення музики
- Можливість коментувати, писати відгук
- Можливість сортування та створення плейлистів

### 5.2 Вимоги до програмного забезпечення

- Google Chrome, Opera, або інший аналогічний браузер

### 5.3 Вимоги до апаратного забезпечення

- Комп'ютер з підключенням до мережі Інтернет

## 6. ЕТАПИ РОЗРОБКИ

|  |            |
|--|------------|
| Вивчення літератури та інших матеріалів      | 20.01.2021 |
| Складання та узгодження технічних матеріалів | 07.02.2021 |
| Аналіз доступних технологій                  | 14.02.2021 |
| Розробка модулів розроблюваної системи       | 01.03.2021 |
| Тестування системи                           | 15.04.2021 |
| Виправлення помилок                          | 14.05.2021 |
| Оформлення документації дипломної роботи     | 28.05.2021 |

|      |      |          |        |      |                    |      |
|------|------|----------|--------|------|--------------------|------|
|      |      |          |        |      | ІАЛЦ.467200.002 ТЗ | Арк. |
|      |      |          |        |      |                    | 3    |
| Змн. | Арк. | № докум. | Підпис | Дата |                    |      |

**Пояснювальна записка  
до дипломного проєкту  
за темою: «Бібліотека музичних творів»**

Київ – 2021 року

# ЗМІСТ

|  |           |
|--|-----------|
| Перелік позначень.....   | 3         |
| Вступ.....   | 4         |
| <b>РОЗДІЛ 1 ВИЗНАЧЕННЯ ТА АНАЛІЗ РОБОТИ МУЗИЧНИХ СЕРВІСІВ .....</b>                          | <b>6</b>  |
| 1.1 Що таке музичний сервіс.....   | 6         |
| 1.2 Характеристика музичних сервісів .....   | 6         |
| 1.3 Огляд та порівняння наявних рішень .....   | 7         |
| 1.3.1 YouTube Music .....  | 7         |
| 1.3.2 Spotify.....   | 11        |
| 1.3.3 SoundCloud .....   | 14        |
| 1.3.4 Порівняння наявних рішень.....   | 15        |
| 1.4 Вимоги до сервісу, що розробляється.....   | 17        |
| Висновок до розділу 1 .....  | 19        |
| <b>РОЗДІЛ 2 ОБҐРУНТУВАННЯ, ВИБІР АРХІТЕКТУРИ, СТРУКТУРИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b> | <b>20</b> |
| 2.1 Аналіз і характеристика об’єкта проєктування.....  | 20        |
| 2.2 Вибір програмного забезпечення .....   | 23        |
| 2.2.1 Python.....  | 24        |
| 2.2.2 Java.....  | 24        |
| 2.2.3 Microsoft .NET (ASP.NET).....  | 26        |
| 2.2.4 Node.JS.....   | 27        |
| 2.2.5 PHP.....   | 27        |
| 2.2.6 MySQL .....  | 29        |
| 2.2.7 PostgreSQL.....  | 30        |
| 2.3 Вибір архітектури.....   | 30        |
| 2.4 Вибір середовища розробки.....   | 32        |
| 2.5 Вибір контейнеру сервлетів .....   | 34        |
| Висновок до розділу 2 .....  | 35        |
| <b>РОЗДІЛ 3 ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ .....</b>   | <b>36</b> |
| 3.1 Опис бази даних.....   | 36        |

|                  |             |                       |               |             |                                   |                         |             |                |
|------------------|-------------|-----------------------|---------------|-------------|-----------------------------------|-------------------------|-------------|----------------|
|                  |             |                       |               |             | <i>ІАЛЦ.467200.003 ПЗ</i>         |                         |             |                |
| <i>Змн.</i>      | <i>Арк.</i> | <i>№ докум.</i>       | <i>Підпис</i> | <i>Дата</i> |                                   |                         |             |                |
| <i>Розроб.</i>   |             | <i>Клецун М.С.</i>    |               |             | <i>Бібліотека музичних творів</i> | <i>Літ.</i>             | <i>Арк.</i> | <i>Акрушів</i> |
| <i>Перевір.</i>  |             | <i>Сімоненко А.В.</i> |               |             |                                   |                         | <i>1</i>    | <i>58</i>      |
| <i>Н. Контр.</i> |             | <i>Сімоненко В.П.</i> |               |             | <i>Пояснювальна записка</i>       | <i>НТУУ «КПІ», ФІОТ</i> |             |                |
| <i>Затверд.</i>  |             |                       |               |             |                                   | <i>ІО-72</i>            |             |                |

|   |   |           |
|---|---|-----------|
| 3.2   | Опис архітектури системи.....                               | 38        |
| 3.2.1                                       | RegisterCommand.....  | 40        |
| 3.2.2                                       | LogInCommand .....  | 41        |
| 3.2.3                                       | UserAddFavouriteCommand .....                               | 41        |
| 3.2.4                                       | ArtistAddMusicCommand, ArtistChangeMusicAlbumCommand .....  | 42        |
| 3.3   | Опис алгоритмів.....  | 42        |
| 3.3.1                                       | Алгоритм реєстрації.....                                    | 42        |
| 3.3.2                                       | Алгоритми автентифікації та авторизації.....                | 43        |
| 3.3.3                                       | Алгоритм додавання музики до альбому улюблених пісень ..... | 44        |
|   | Висновок до розділу 3 .....                                 | 45        |
| <b>РОЗДІЛ 4 ТЕСТУВАННЯ ПРОГРАМИ .....</b>   |   | <b>46</b> |
| 4.1   | Тестування головної сторінки.....                           | 46        |
| 4.2   | Тестування сторінки автентифікації та реєстрації .....      | 47        |
| 4.3   | Тестування сторінки особистого кабінету.....                | 49        |
| 4.4   | Тестування сторінок пісні, виконавця та альбому.....        | 52        |
|   | Висновки до розділу 4.....                                  | 55        |
| <b>ВИСНОВКИ .....</b>                       |   | <b>56</b> |
| <b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....</b> |   | <b>58</b> |

## Перелік позначень

Стрімінг (англ. streaming) – спосіб передачі мультимедіа, яке користувач безперервно отримує від провайдера потокового мовлення.

Чарт – список найпопулярніших у визначений період музичних композицій.

Тренд – збільшення кількості пошукових запитів, тем, статей з певної тематики

MP3 – формат файлу, створений командою MPEG-3 для збереження аудіоінформації з суттєвими втратами якості звучання.

OGG – формат, створений для зберігання потоку даних, таких як відео або аудіо

FLAC (англ. Free Lossless Audio Codec) – вільний аудіокодек призначений для стиснення аудіоданих без втрат

Кодек – програма, яка здатна виконувати перетворення потоку даних.

Бітрейт – кількість бітів, які використовуються для передачі або обробки даних за секунду

Плейлист – підбірка аудіоконтенту, створеного для відтворення на радіо або за допомогою медіаплеєра

БД – база даних

СУБД – система управління базами даних

MVC – Model View Controler

MVP – Model View Presenter

MVVM – Model View ViewModel

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 3           |

## Вступ

На сьогоднішній день, музика є невід'ємною частиною людського життя. Ми слухаємо музику для концентрації, коли працюємо або вчимося, енергійну музику, коли займаємось в спортзалі або йдемо кудись, чи спокійну, щоб відпочити від швидкоплинного життя. Музика сильно впливає на людину, вона здатна підняти настрій, розбудити або навіть приспати.

Вчені встановили певні закономірності впливу різної музики на психічний та фізичний стан людини. Наприклад, європеєць не завжди буде відчувати себе комфортно, слухаючи східні мотиви, а тривалий вплив такої музики може привести до психічного розладу. У той час як жителям Азії навряд чи припаде до душі західна музика. У людини прослуховування музики може супроводжуватися почуттям ейфорії, викликаним вивільненням дофаміну в області смугастого тіла головного мозку.

Передбачається, що музика існує вже 50.000 років, а найдавнішим зафіксованим вченими музикальним інструментом є флейта. Тобто, спочатку музика серед людей поширювалась, передавалась лише інструментами або співом. Але 150 років назад Томас Едісон все змінив, створивши фонограф. Цей апарат міг записувати звуки, в тому числі та музику. Після цього був створений магнітофон. Люди могли слухати та передавати музику на касетах. Через невеликий час було створено телевізори, на яких також можна було слухати музику. За останні 40 років індустрія програвачів музики дуже сильно змінилась, від плеєрів з CD-дисками, до стрімінгових сервісів музичних композицій, які є в кожному телефоні та в інтернеті. Саме про них і піде далі мова.

Метою цього дипломного проекту є розробка власного клієнт-серверного веб-сервісу для прослуховування, збереження та поширення музики у мережі Інтернет. Перед початком роботи було досліджено і порівняно основні найвідоміші сервіси потокового аудіо у світі. В результаті дослідження було виявлено такі риси гарного програмного продукту: доволі

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 4           |

простий, а саме зрозумілий у користуванні інтерфейс, гарна якість звучання самої музики, різноманітність жанрів, безкоштовне прослуховування музики, наявність підбору наступної музики за вашим вподобанням, збереження улюблених композицій та багато всього іншого.

Призначення дипломного проєкту – розробити веб-сервіс потокового аудіо, для прослуховування музики.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 5           |





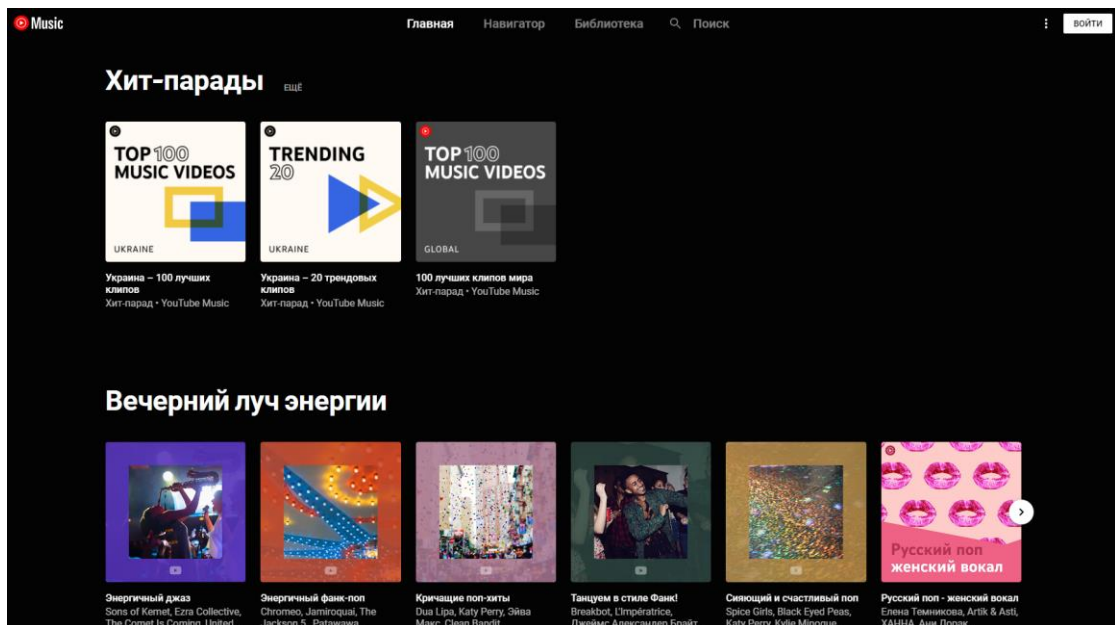


Рисунок 1.1 – интерфейс «YouTube Music»

Тут же на підставі лайків формується плейлист «Рекомендації для вас» і «Мій джем» (рисунок 1.2). Ще є плейлист з новими релізами, куди потрапляють свіжі композиції.

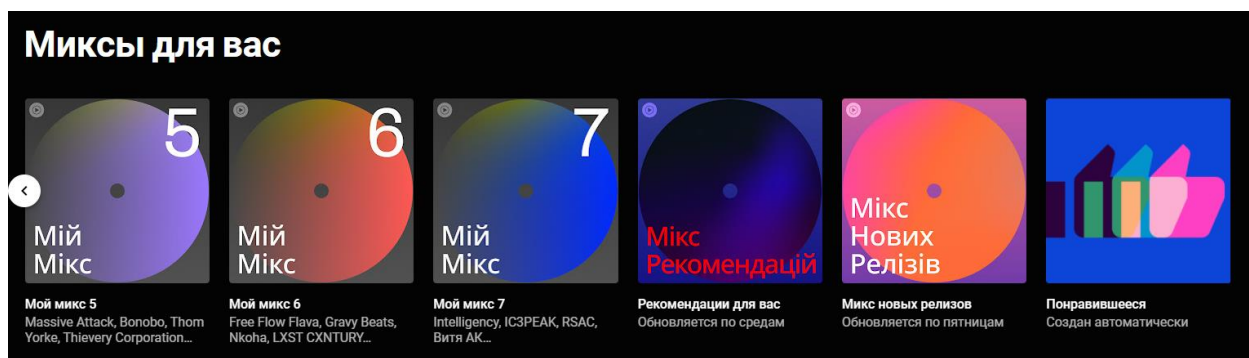


Рисунок 1.2 – Плейлисты в YouTube Music

В плейлист улюблених треків додаються не тільки офіційні композиції, а й отлайкані кліпи і музичні відео. Причому Google перевіряє інформацію з облікового запису користувача і іноді додає до списку відтворення не пов'язані з музикою. Сервіс має чудову пошукову систему, яка пропонує користувачу альбоми, треки та виконавців за ключовими словами (рисунок 1.3). Є початківці або досить відомі музиканти і групи, які з певних причин не потрапляють в стрімінгові сервіси. Але якщо їх треки або відео завантажені на YouTube, то композиції можна спокійно слухати в додатку YouTube Music.

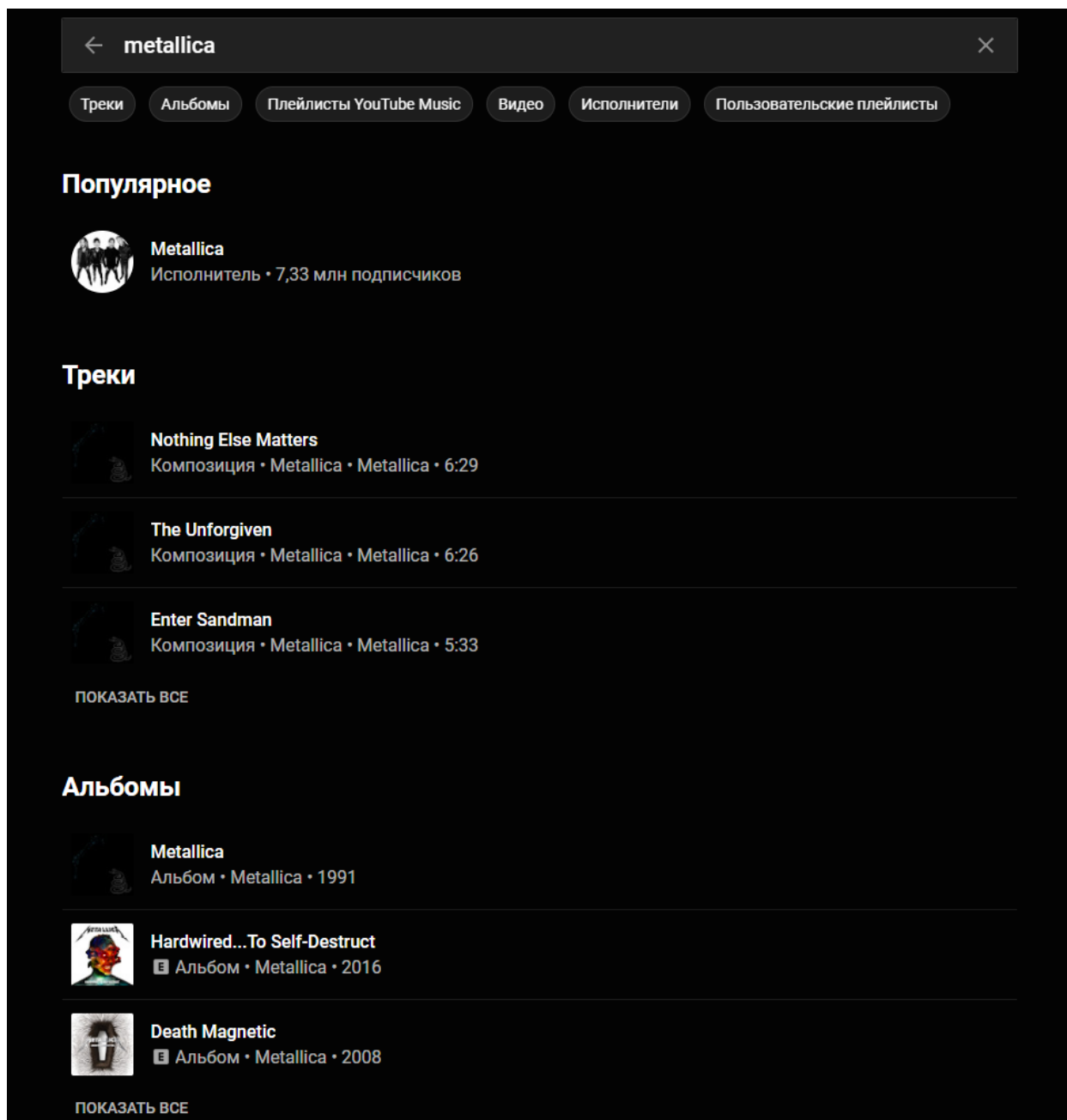


Рисунок 1.3 – Пошукова система YouTube Music

Також, на головному меню «YouTube Music» пропонує вам композиції зі схожим звучанням, на ті, котрі ви слухали раніше, або авторів зі схожим стилем та жанром пісень, які вам сподобались. Після вибору плейлисту, можна робити будь-які дії з кожними музичними композиціями, а саме: включити радіостанцію (створюється новий плейлист, на початку якого знаходиться обрана пісня), включити наступним, додати в чергу (пісня додається в кінець списку), додати до вподобаних, додати в плейлист (створюється новий або

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | <i>ІАЛЦ.467200.003 ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 9    |

додається в уже наявний плейлист), видалити з черги, відкрити альбом, перейти на сторінку виконавця або поділитися композицією (рисунок 1.4).

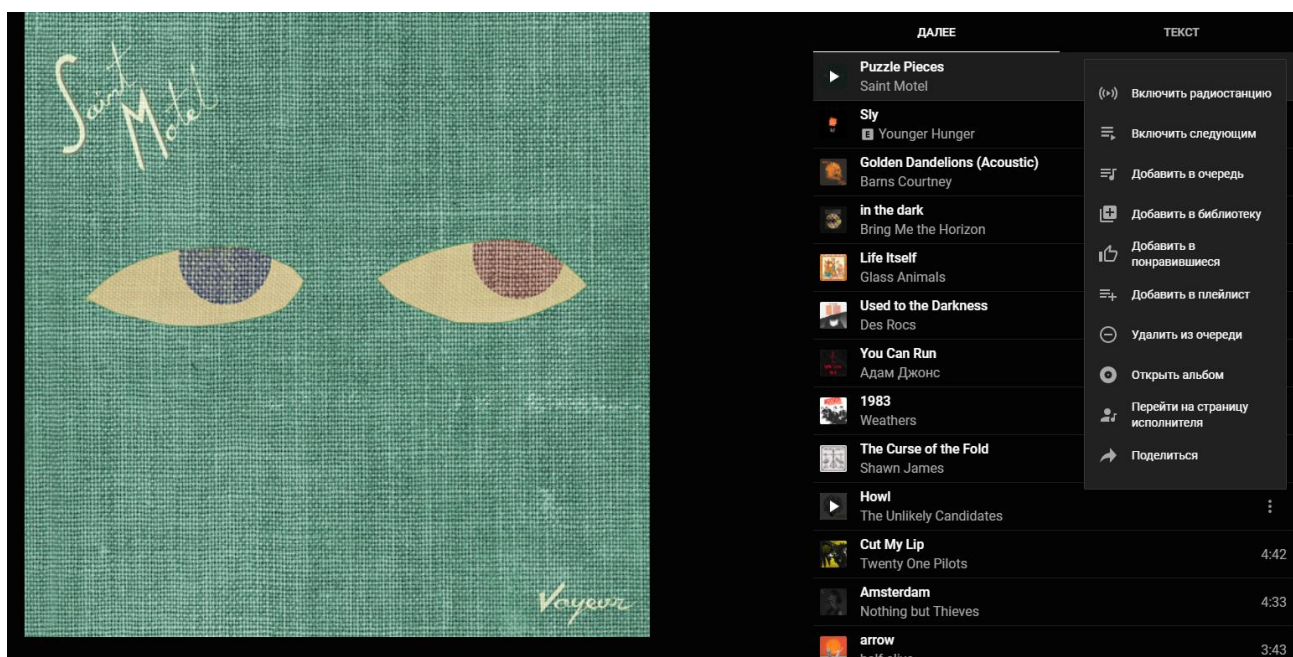


Рисунок 1.4 – Можливі дії користувача в обраному плейлисті

Для деяких композицій є кліпи, які можна переглянути прямо на сайті. Взагалі, на цьому сервісі існують, мабуть, усі пісні та композиції, від всесвітньо відомих до пісень, котрі знають невеличка купа людей. Також, ще одною перевагою «YouTube Music» є те, що тут можна слухати музику не проходячи реєстрацію на сайті. В мобільному додатку можна завантажувати безмежну кількість пісень, щоб потім їх слухати оффлайн.

Але цей сервіс має не тільки переваги, а й недоліки. Головним з них є реклама, яка може з'являтися майже перед кожною композицією, що досить швидко набридає слухачу. Також, якість звуку в безкоштовній версії «YouTube Music» майже в два рази гірша за платну, якою можна виправити два попередні недоліки. Але такий недолік, як відсутність коментарів, не можна виправити платною підпискою. Це не дає користувачу повністю висловити свою точку зору щодо музики, яку він слухає.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | <i>ІАЛЦ.467200.003 ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 10   |





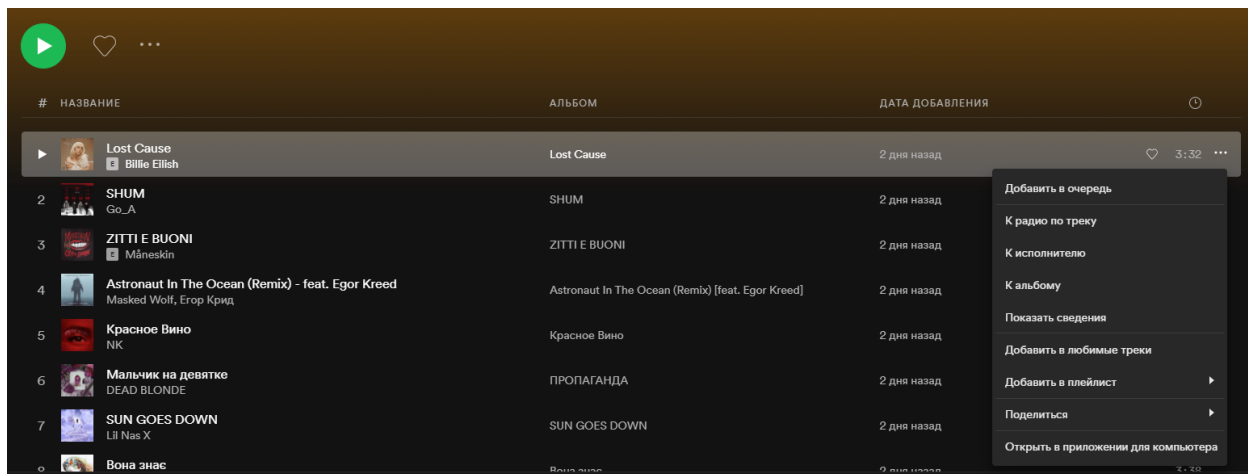


Рисунок 1.7 – Можливості користувача Spotify

Тобто «Spotify» має майже все те, що й YouTube Music, але кількість реклами в рази менша, ніж у YouTube Music. Якщо ви використовуєте план Spotify Premium, доступна функція прослуховування в автономному режимі, і ви можете завантажувати і зберігати музику Spotify на своєму телефоні для автономного відтворення. Таким чином, ви можете слухати музику Spotify і зберігати багато мобільних даних без підключення до інтернету. Також, цей сервіс надає змогу користувачу завантажувати будь-яку кількість композицій на персональній комп'ютер, але тільки за наявності преміумної підписки.

Недоліком «Spotify» є те, що не можна почати слухати музику без реєстрації на сайті. Маючи безкоштовну обліковий запис, користувачі Spotify повинні слухати звукову рекламу між своїми улюбленими піснями і епізодами подкастів. Крім того, ви не можете транслювати певні пісні за запитом, і вам дозволено пропускати лише 6 треків на годину. Це дуже неприємний для користувача досвід, особливо коли ви чуєте щось, що вам не подобається. Варто додати, що цей сервіс не має кліпів. Також, як і YouTube Music, Spotify не дає можливості користувачу залишати коментарі під композиціями або подкастами. Відсутність повної підтримки текстів в Spotify і його мінімальна увага до голосу починають ставати проблемами для потокового сервісу.



музика маловідомих композиторів, діджеїв чи бітмейкерів. Інтерфейс цього сайту кардинально відрізняється від двох попередніх і є трохи складним у використанні. Проте на ньому можна слухати музику без реєстрації та без реклами. Також, користувачі цього сайту можуть коментувати музику, яку вони слухають, навіть з поміткою моментів, котрі вплинули на них найбільше.

### **1.3.4 Порівняння наявних рішень.**

Після аналізу наявних популярних стрімінгових музичних сервісів було визначено такі риси хорошого продукту:

- Простий та зручний у користуванні інтерфейс. Користувач повинен без допомоги інших засобів легко користуватися будь-якими функціями сервісу.
- Баланс між якістю звучання та розміром потокового аудіо-файлу. Тобто, якість звуку не повинна бути поганою (96-128 Кбіт/с), але й розмір самої композиції не повинен перевищувати 10МБ (Розмір залежить від якості), бо, при наявності поганого інтернет з'єднання, користування таким сервісом буде жахливим.
- Велика бібліотека музичних творів. Також, на популярність сильно вплине наявність кліпів або текстів пісень.

Якщо порівнювати саме три сервіси з попередніх підрозділів, то до переваг “YouTube Music” над іншими я би відніс зручний у користуванні інтерфейс та наявність контенту з сервісу “YouTube”, а саме кліпів. На мою думку, компанія Google розробила найбільш приємний та інтуїтивно-зрозумілий інтерфейс своєї програми, де користувачу не потрібно шукати інформацію в мережі Інтернет, щоб скористатися всіма функціями сервісу. Також, наявність синхронізації з ще одним сервісом компанії Google – “Youtube”, дає можливість користувачу переглядати кліпи своїх улюблених пісень. Також, дешевизна спеціальною підписки є ще одною перевагою серед інших сервісів.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 15          |

До переваг стрімінгово сервісу музики “Spotify” можна віднести наявність найбільшої бібліотеки музичних творів та блискучі алгоритми підбору музики. Так як сервіс працює з 2006 року, за 15 років він зміг зібрати величезну базу виконавців усіх років та поколінь і зараз може задовольнити будь-які примхи користувачів. А блискучі алгоритми підбору музики, на розробку яких було витрачено багато років та грошей, не дають користувачу нудьгувати та весь час пропонують йому щось нове або давно забуте старе, утримуючи його на платформі.

Щодо переваг “SoundCloud”, то до них можна віднести найбільш велику бібліотеку музичних композицій інді-виконавців та можливість повністю висловити свою точку зору. Взагалі, сервіс створювався як платформа для виконавців, тому кожен може викласти свою композицію та отримати коментарі, відгуки або рекомендації. В цьому звичайному користувачу допоможе хмарне сховище для своїх аудіо-файлів та гарно розроблена система оцінки. Адже сервіс “SoundCloud”, майже як соціальна мережа, надає можливість користувачу створювати спільноти, підписуватись на інших людей та контактувати з улюбленими виконавцями або кумирами.

В таблиці 1.1, наведено опис переваг та недоліків розглянутих наявних рішень.

Таблиця 1.1 – Переваги та недоліки аналогів

| Назва         | Переваги   | Недоліки  |
|---------------|--|---|
| YouTube Music | <ul style="list-style-type: none"> <li>- привабливий та зрозумілий дизайн</li> <li>- зручний пошук</li> <li>- гарні підбірки музики</li> <li>- наявність кліпів</li> <li>- прослуховування музики без реєстрації</li> <li>- дешева підписка</li> </ul> | <ul style="list-style-type: none"> <li>- багато реклами та якість звучання в безкоштовній версії</li> <li>- відсутність коментарів</li> </ul> |

## Продовження таблиці 1.1

| Назва      | Переваги  | Недоліки   |
|------------|---|--|
| Spotify    | <ul style="list-style-type: none"> <li>- привабливий дизайн</li> <li>- зручний пошук</li> <li>- чудові підбірки музики</li> <li>- відсутність реклами</li> <li>- хороша якість звучання</li> </ul>                  | <ul style="list-style-type: none"> <li>- прослуховування музики тільки після реєстрації</li> <li>- відсутність кліпів</li> <li>- відсутність коментарів</li> </ul> |
| SoundCloud | <ul style="list-style-type: none"> <li>- зручний пошук</li> <li>- відсутність реклами</li> <li>- наявність коментарів</li> <li>- хороша якість звучання</li> <li>- прослуховування музики без реєстрації</li> </ul> | <ul style="list-style-type: none"> <li>- складний інтерфейс</li> <li>- підбірка музики гірша ніж у конкурентів</li> <li>- відсутність кліпів</li> </ul>            |

### 1.4 Вимоги до сервісу, що розробляється

Після аналізу та порівняння наявних стрімінгових музичних сервісів було визначено вимоги майбутнього сервісу «Бібліотека музичних творів». Така програма повинна мати гарний та зручний у користуванні інтерфейс, за основу якого буде взятий та дороблений інтерфейс «YouTube Music». До кожної музичної композиції буде додана можливість залишати та лайкати вподобані коментарі. Сервіс повинен мати гарні алгоритми підбору музики та хорошу якість звучання. При спеціальній підписці, користувач має змогу насолоджуватися музикою не тільки в форматі OGG або MP3, а ще й у форматі FLAC, який стискає музику без втрат якості. Кожен власник такої підписки має можливість завантажувати будь-яку музичну композицію з сайту. Також, кожен користувач матиме можливість підтримати свого улюбленого виконавця, пожертвувавши невеликою сумою грошей, та контактувати з ним

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | <i>ІАЛЦ.467200.003 ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 17   |

у особистому чаті. Сервіс не буде мати рекламу, бо основний дохід буде з підписок та невеликого відсотку з пожертвування виконавцям.

Усі ці властивості повинні зробити сервіс «Бібліотека музичних творів» популярним і зручним для користувачів, та прибутковим для власників.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 18          |

## Висновок до розділу 1

У цьому розділі було проведено огляд поняття музичного сервісу та наявних в мережі Інтернет найбільш популярних стрімінгових сервісів. Після детального аналізу та порівнянні наявних аналогів було визначено основні риси успішного та популярного продукту. Також, було встановлено вимоги до майбутнього сервісу «Бібліотека музичних творів» та поліпшено переваги наявних аналогів.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 19          |

## РОЗДІЛ 2

# ОБҐРУНТУВАННЯ, ВИБІР АРХІТЕКТУРИ, СТРУКТУРИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Аналіз і характеристика об'єкта проєктування

Основою на першому розділі, проєкт буде містити вхідні дані формату .mp3 (для музичних творів, пісень, аудіокнижок та інших матеріалів, що можуть зберігатися в бібліотеці) та .jpg (для зберігання титульного зображення композицій, що присутні в бібліотеці). Також інформаційні дані про авторів, композиції чи альбоми, які мають заповнювати та відтворювати інформацію, котра надходить до сервісу. Веб-додаток повинен надати користувачу можливість реєстрації (процес створення нового облікового запису користувача, присвоєння йому ключа у вигляді логіна та пароля і надання прав доступу до ресурсу), автентифікації (цей процес перевіряє наявність користувача в базі даних та надає доступ до неї, якщо він там є, або забороняє доступ, за його відсутності в ній); авторизації (процес, що слідує за автентифікацією та надає права доступу (адміністратор, виконавець або звичайний користувач) до системи за наявності правильного ключа (в нашому випадку логін та пароль)), прослуховування музики яка присутня в бібліотеці, коментування та додавання композиції до вподобаних, завантажування аудіо файлів у вигляді .mp3, котрі присутні в бібліотеці.

Більш детальні умови до функціоналу наведено в таблиці 2.1

Таблиця 2.1 – Вимоги до функціоналу додатку

| Функція    | Вимоги   |
|------------|--|
| Реєстрація | Потрібно надати можливість користувачу зареєструватися в системі, вказавши свій email, повне ім'я та пароль. |

Продовження таблиці 2.1

| Функція                             | Вимоги  |
|-------------------------------------|---|
| Атентифікація                       | Потрібно надати можливість користувача аутентифікуватись в системі, використовуючи вже створений профіль. Аутентифікація – це процес перевірки відповідності введеного логіну і паролю до логіну і паролю, що збережені у базі даних. Якщо така пара є в БД системи, то користувачу надається доступ до внутрішніх даних та сервісів. |
| Редагування профілю                 | Необхідно додати функціонал для редагування персональної інформації користувача. Змінити можна лише певні дані профілю, відповідно наданій користувачу інформації.  |
| Надання користувачу прав виконавця  | Потрібно забезпечити можливість адміністратору надати конкретному користувачу права менеджера.  |
| Перегляд сторінки профілю           | Необхідно надати можливість переглядати інформацію у своєму профілі.  |
| Перегляд бібліотеки музики          | Потрібно надати можливість переглядати користувачу всю музику, котра є на сайті. Дана функція повинна підтримувати розбиття усієї музики за жанрами та виконавцями.   |
| Прослуховування музики або подкасту | Потрібно надати можливість прослуховувати онлайн музику, котра є на сайті.  |
| Оцінка музичних композицій          | Потрібно надати можливість користувачу ставити «лайк» на ті пісні, котрі йому подобаються та додавати їх у спеціальну підбірку з улюбленими музичними творами.  |

Продовження таблиці 2.1

| Функція   | Вимоги   |
|---|--|
| Коментування музичних композицій                        | Потрібно надати можливість користувачу залишати та видаляти свої коментарі під кожною музикою, що існує на сайті.                              |
| Видалення коментарів                                    | Потрібно надати можливість адміністратору видалити будь-який коментар, якщо в ньому є погрози або негативна поведінка                          |
| Завантаження музичних композицій або подкастів          | Потрібно надати можливість усім користувачам завантажувати будь-яку музику або подкаст   |
| Додавання музичних композицій або подкастів             | Потрібно надати можливість додавати музичні композиції або подкасти виконавцям та адміністратору   |
| Додавання та видалення тегів                            | Потрібно надати можливість виконавцям та адміністратору додавати чи видаляти теги до своїх композицій (адміністратору до будь-яких композицій) |
| Додавання та видалення музичних композицій з плейлистів | Потрібно надати можливість усім користувачам додавання та видалення музичних творів з плейлисту, який вони зараз слухають                      |
| Додавання та видалення музичних творів з альбому        | Потрібно надати можливість виконавцям додавати або видаляти музичні твори зі свого альбому   |

## Продовження таблиці 2.1

| Функція                                   | Вимоги  |
|---|---|
| Вихід з облікового запису                 | У користувача має бути можливість виходу з облікового запису  |
| Забезпечення вводу коректності всіх даних | Потрібно перевіряти коректність всіх введених даних, та, при помилках під час введення даних, повідомляти про це користувача. |

## 2.2 Вибір програмного забезпечення

На сьогоднішній день існує велика кількість мов програмування, що допомагають розробнику у вирішенні поставлених завдань та реалізують потреби програмістів, що працюють на різних платформах або операційних системах.

Для розробки дипломного проєкту потрібно розглянути насамперед веб-орієнтовані мови програмування, тому що мій сервіс повинен буде запускатися на сервері і користувачі зможуть оперувати ним тільки через Інтернет ресурс, а саме веб-сайт. Також, треба реалізувати надійну та гнучку серверну частину, яка за допомогою інтерфейсу буде обробляти запити користувачів та надсилати оброблену інформацію у відповідь. Тобто, для того, щоб дипломний проєкт працював належним чином, потрібно створити веб-додаток, який має можливість аналізувати запити від клієнта на стороні серверній частині, обробляти їх, формалізувати та відправляти у відповідь інформацію у вигляді HTML, CSS та JS сторінки. З цим впорається протокол HTTP, через який будуть здійснюватися усі дії.

Для створення такого сервісу потрібно обрати актуальну мову програмування, яка легка та продуктивна у використанні, має достатню кількість бібліотек для полегшення розробки програми, має незалежність від платформи та доступну для налагодження серверну частину з подальшою

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 23          |

можливістю розширення, має можливість зміни фрейм ворку у зв'язку з розвитком технологій програмування та створення новітніх, покращених засобів. Також, така мова програмування повинна мати зручне та перевірене роками середовище розробки.

Серед популярних мов для веб-програмування є такі: Python, Java, Microsoft .NET(ASP.NET), Node.JS, PHP.

### 2.2.1 Python

Python [5] став дуже популярною за останні роки мовою програмування для реалізації задуманих веб-проектів. Хоча синтаксис є доволі простим і деякі використовують його лише для написання скриптів чи невеликих програм, він має сучасні і популярні фрейм ворки: FastAPI – високопродуктивний, доступний та швидкий фрейм ворк Python; Django [6] – найпопулярніший фрейм ворк написаний на Python, який зручно використовувати для створення сайтів, що працюють за базами даних. Найпопулярнішими прикладами реалізації є: YouTube, Instagram, Pinterest, NASA. Django – вільний фреймворк для веб-додатків на мові Python, що використовує шаблон проектування MVC. Проект підтримується організацією Django Software Foundation. Сайт на Django будується з одного або декількох додатків. Це одне з істотних архітектурних відмінностей цього фреймворка від деяких інших (наприклад, Ruby on Rails). Один з основних принципів фреймворка - DRY (англ. Do not repeat yourself). Також, на відміну від інших фреймворків, обробники URL в Django конфігуруються явно за допомогою регулярних виразів. Для роботи з базою даних Django використовує власну ORM (об'єктно-реляційна проекція, або «віртуальна об'єктна база даних»), в якому модель даних описується класами Python, і по ній генерується схема бази даних.

### 2.2.2 Java

Мову програмування Java [7] створили, щоб забезпечити функціонування приладів, котрі нас оточують (монітори, в основному

|      |      |          |        |      |  |  |  |  |  |      |
|------|------|----------|--------|------|--|--|--|--|--|------|
|      |      |          |        |      |  |  |  |  |  | Арк. |
|      |      |          |        |      |  |  |  |  |  | 24   |
| Змн. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |      |

телефони, кондиціонери та інші) та вирішити проблему ресурсних затрат. Саме на цій мові була створена легендарна гра Minecraft. Також, Java застосовується у веб-розробці, де так і не втрачає свого девізу «Один раз напиши код і всюди його використовуй».

Servlet являє собою спеціальний тип класів Java, який виконується на веб-сервері і котрий обробляє запити та повертає результат обробки. Бібліотека Java містить широкий вибір засобів, що дає змогу розробнику повністю реалізувати усі можливості Java Servlets. Також є можливість обробляти низько-рівневі запити, що надає програмі гнучкості.

Також було створено такі технології як JSP, JSF та Spring за увесь час існування сервлетів.

JSP – зручний але вже застарілий механізм розробки, завдяки якому можна компонувати Java-код і HTML-код. Java Server Pages представляє технологію, яка дозволяє створювати динамічні веб-сторінки. Спочатку JSP (разом з Сервлетами) на зорі розвитку Java EE були домінуючим підходом до веб-розробки на мові Java. І хоча в даний час вони поступилося своїм місцем іншій технології - JSF, проте JSP продовжують широко використовуватися. По суті Java Server Page або JSP є html-код з вкрапленнями коду Java. У той же час станиці jsp - це не стандартні html-сторінки. Коли приходить запит до певної сторінці JSP, то сервер обробляє її, генерує з неї код html і відправляє його клієнту. В результаті користувач після звернення до сторінці JSP бачить в своєму браузері звичайну html-сторінку.

JSF [8] являє собою більш сучасну і складнішу технологію для взаємодії користувача через інтерфейс з серверною частиною. Цю технологію зручно використовувати разом з високопродуктивними та доступними компонентами PrimeFaces, які розробник може повністю налаштувати під свої нужди і створити інтерфейс користувача, використовуючи велику бібліотеку тегів. Під час розробки програмного забезпечення можна зіткнутися з проблемами в

налагоджені веб-додатку, але в їх вирішенні допоможе велика кількість форумів створених для обговорень і JSF, і PrimeFaces. Пару слів про те, що взагалі з себе представляє JSF. JSF - веб-MVC фреймворк, який заснований на використанні компонентів і є частиною JAVA EE. Для базового розуміння про те, як виглядає додаток на JSF потрібно знати лише кілька речей. Перше - за подання до JSF відповідають JSP сторінки або Facelets шаблони. Взагалі, Facelet'и були створені для заміни JSP, але девізом jsp можна назвати наступне: "Те, що мертво, померти не може", тому в якості розуміння все ще часом ним користуються. По суті Facelets, це xml документи, що описують сторінки і вміють в jsf компоненти.

Spring Framework [9] – один з найпопулярніших фреймворків для J2EE. Розробники по всьому світу використовують Spring для створення надійних і якісних додатків. Він був розроблений в червні 2003 року Родом Джонсоном. За допомогою Spring можна розробити будь-який Java програма. Spring Framework забезпечує комплексну модель розробки і конфігурації для сучасних бізнес-додатків на Java - на будь-яких платформах. Ключовий елемент Spring - підтримка інфраструктури на рівні програми: основна увага приділяється "водопроводу" бізнес-додатків, тому розробники можуть зосередитися на бізнес-логіці без зайвих налаштувань в залежності від середовища виконання.

### 2.2.3 Microsoft .NET (ASP.NET)

.NET Framework - це програмна платформа, випущена компанією Microsoft, яка підходить для різних мов програмування. ASP.NET - технологія створення веб-додатків і веб-сервісів від компанії Microsoft. Вона є складовою частиною платформи Microsoft .NET і розвитком старшої технології Microsoft ASP. Вважається, що платформа .NET Framework стала відповіддю компанії Microsoft на платформу Java. ASP.NET ґрунтується на Common Language Runtime: розробники можуть писати код для ASP.NET, використовуючи практично будь-які мови програмування, деякі з яких входять в комплект .NET

|      |      |          |        |      |  |  |  |  |      |
|------|------|----------|--------|------|--|--|--|--|------|
|      |      |          |        |      |  |  |  |  | Арк. |
|      |      |          |        |      |  |  |  |  | 26   |
| Змн. | Арк. | № докум. | Підпис | Дата |  |  |  |  |      |

Framework (C #, Visual Basic.NET і JScript .NET), а інші можуть бути встановлені додатково (IronRuby, IronPython, PHP, Perl, Smalltalk, Haskell і ін.). Проте, нерідко використовується не цілком коректний термін «.NET-программист», який, мабуть, характеризує розробника на будь-якій мові програмування, який може бути використаний в .NET-проект.

### 2.2.4 Node.JS

Node.JS являє собою серверну реалізацію мови програмування JavaScript. Перевагами над іншими мовами програмування є: можливість одночасного виконання декількох операцій, які між собою не пов'язані; використання на сервері і на клієнті однакової мови програмування, що спрощує розробку; гнучка та лаконічна мова. Основними недоліками Node.JS є проблеми з налагодженням можливих помилок та проблеми з контролем пам'яті. Також до недоліків можна віднести відсутність повноцінної IDE. Програмна платформа заснована на движку V8 (здійснює трансляцію JavaScript в машинний код), що перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення. Node.js додає можливість JavaScript взаємодіяти з пристроями вводу-виводу через свій API, написаний на C ++, підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики до них з JavaScript-коду. Node.js застосовується переважно на сервері, виконуючи роль веб-сервера, але є можливість розробляти на Node.js і десктопні віконні додатки (за допомогою NW.js, AppJS або Electron для Linux, Windows і macOS) і навіть програмувати мікроконтролери. В основі Node.js лежить подієво-орієнтоване і асинхронне (або реактивне) програмування з неблокуючим введенням / виводом.

### 2.2.5 PHP

PHP - скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що

|      |      |          |        |      |  |  |  |  |  |      |
|------|------|----------|--------|------|--|--|--|--|--|------|
|      |      |          |        |      |  |  |  |  |  | Арк. |
|      |      |          |        |      |  |  |  |  |  | 27   |
| Змн. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |      |



(останнє оновлення версії 2.3 в 2013 році). Для виконання завдання потрібна програма яку можна буде легко доповнювати, розширювати можливості і в цьому нам допоможе JSF, а у разі потреби можна без особливих затрат змінити технологію на Spring Framework. Інтегровані середовища розробки (IntelliJ IDEA, NetBeans, Eclipse), які вже декілька десятків років підтримують середовище інструментів, налагодження, компілювання і редагування коду, а також, мають офіційні сайти з технічною підтримкою користувачів, надають впевненості та полегшують розробку програмного продукту.

Тепер треба обрати базу даних для зберігання інформації. Мені потрібна гнучка та проста в використанні реляційна база даних, яка має можливість створювати різні типи об'єктів, в котрих дані мають логічну послідовність та зв'язок за певними критеріями (id, name, username). Щоб я міг за просто взяти потрібні мені дані за допомогою запитів і так само легко доповнити їх, редагувати, видалити, фільтрувати та сортувати. В цьому мені допоможуть такі бази даних: MySQL, PostgreSQL.

## 2.2.6 MySQL

MySQL[11] - це реляційна система управління базами даних з відкритим вихідним кодом. В даний час ця СУБД одна з найбільш популярних в веб-додатках - переважна більшість CMS використовує саме MySQL (часто тільки її, без альтернатив), а майже всі веб-фреймворки підтримують MySQL вже на рівні базової конфігурації (без додаткових модулів). З переваг СУБД MySQL варто відзначити простоту використання, гнучкість, низьку вартість володіння (щодо платних СУБД), а також масштабованість і продуктивність. Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Є й інші типи таблиць, розроблені спільнотою.

|      |      |          |        |      |  |  |  |  |  |  |      |
|------|------|----------|--------|------|--|--|--|--|--|--|------|
|      |      |          |        |      |  |  |  |  |  |  | Арк. |
|      |      |          |        |      |  |  |  |  |  |  | 29   |
| Змн. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |  |      |

## 2.2.7 PostgreSQL

PostgreSQL - це система управління базами даних з відкритим вихідним кодом корпоративного класу. Вона підтримує як SQL для реляційних, так і JSON для нереляційних запитів. Система підтримується досвідченим спільнотою розробників, які внесли величезний внесок у створення високонадійної системи СУБД. Розробники прагнуть виконувати майже всі стандарти запитів SQL. Виконує складні SQL-запити, забезпечує передову цілісність даних.

Проаналізувавши передові бази даних, я зупинився на виборі MySQL, тому, що з двох ринкових гігантів потрібно вибрати щось одне і більше за все, для не самого масштабного проєкту, розробкою якого займається одна людина, підійде саме MySQL, яка гарантує швидке читання даних, що навіть покращить нашу продуктивність і швидкість отримання даних. А запити в БД будуть не настільки складні, як для використання PostgreSQL.

## 2.3 Вибір архітектури

Архітектура програми – це логічна структура її компонентів, в якій всі компоненти взаємодіють для вирішення поставленого завдання, їх властивості і зв'язки у вигляді однієї продуктивної системи.

Архітектурні патерни проєктування – це і є вирішення архітектури нашого коду, вони включають схеми наших компонентів, їх зв'язки і взаємодію одного з іншими. Вони незалежні від мови програмування, чи структури програми.

Існує декілька видів архітектурних патернів для веб-програмування. Такі, як: MVC, MVP, MVVM та інші.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 30          |

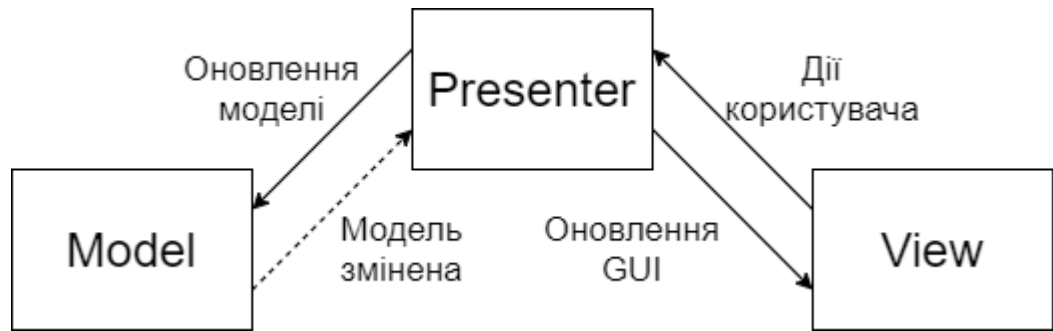


Рисунок 2.1 – Модель MVP

MVP (рисунок 2.1) – схожа до MVC, де модель отримує дані від View тільки через Presenter і не може отримувати відразу з View, як у MVC.

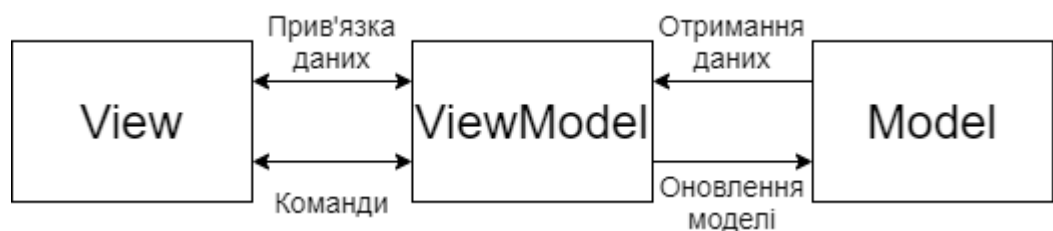


Рисунок 2.2 – Модель MVVM

MVVM (рисунок 2.2) – зв'язаність між Model і View проходить через ViewModel, така концепція дуже зручна для тестування веб-додатку, коли нам не потрібно змінювати Model, а потім перевіряти View, ми можемо зразу перевірити у ViewModel, перед тим, як дані будуть відображені у View.

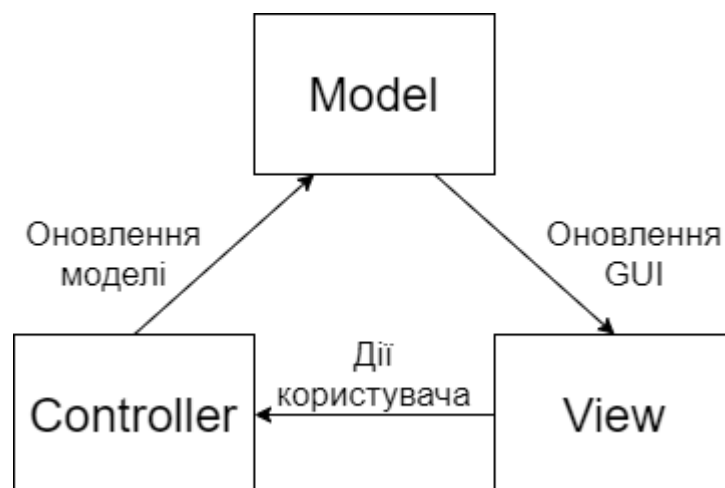


Рисунок 2.3 – Модель MVC

Для вирішення проблеми проєктування додатку, використовуємо архітектурну модель проєктування MVC [10] (рисунок 2.3). Вона допомагає розділити програмний код на модель, представлення і контролер, що мають певну логічну роль у розділенні програмного коду інтерфейсу користувача від управляючої логіки програми. При чому, кожен компонент може існувати окремо. Модель служить для маніпуляцій з даними (у нашому випадку з базою даних), представлення – це відображення інтерфейсу користувача у вигляді сторінки HTML, контролер – для управління всіма даними (через нього проходять всі операції маніпулювання, виконання функцій, зміни інтерфейсу).

Процес виконання запиту починається за допомогою інтерфейсу користувача, контролер аналізує дані, видає помилку у разі некоректності, або направляє дані для обробки бізнес-логікою (функціями). Йому повертається результат, який він інтерпретує для представлення. Користувач отримує результат виконання запиту. Це архітектурне рішення допоможе відділити створення графічного інтерфейсу від бізнес-логіки та класів, що відповідають за отримання даних з БД. Тобто класи будуть поділені на Model, View, Controller.

## 2.4 Вибір середовища розробки

При виборі інтегрованого середовища розробки для реалізації програмного забезпечення я вибрав IntelliJ IDEA 2021.1.1 [12] (рисунок 2.4), тому що вона вже багато років займає одну з перших позицій інтегрованих середовищ для програмування на Java. Щоб студенти могли користуватися повною версією програмного продукту, компанія-розробник (JetBrains) надає ліцензію для студентів на використання повної версії продукту на рік, а саме, для розробки веб-додатків.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 32          |





## Висновок до розділу 2

Проаналізувавши завдання, вибрано мову програмування Java для серверної частини, HTML і JS для інтерфейсу користувача; архітектурний шаблон MVC; програмне забезпечення, яке створить комфортні умови для взаємодії технологій і розробці програмного продукту.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 35          |

## РОЗДІЛ 3

### ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ

#### 3.1 Опис бази даних

База даних грає важливу роль в розробці вебзастосунку. В ній дуже зручно тримати та оперувати всією потрібною нам інформацією для виконання роботи, а саме інформацію про користувача, музику. В попередньому розділі я детально описав, чому я обрав реляційну базу даних MySQL та систему керування БД MySQL Workbench. Під час розробки було створено такі таблиці: User, Music, Album, Genre та Commentary. На рисунку 3.1 наведено структуру БД.

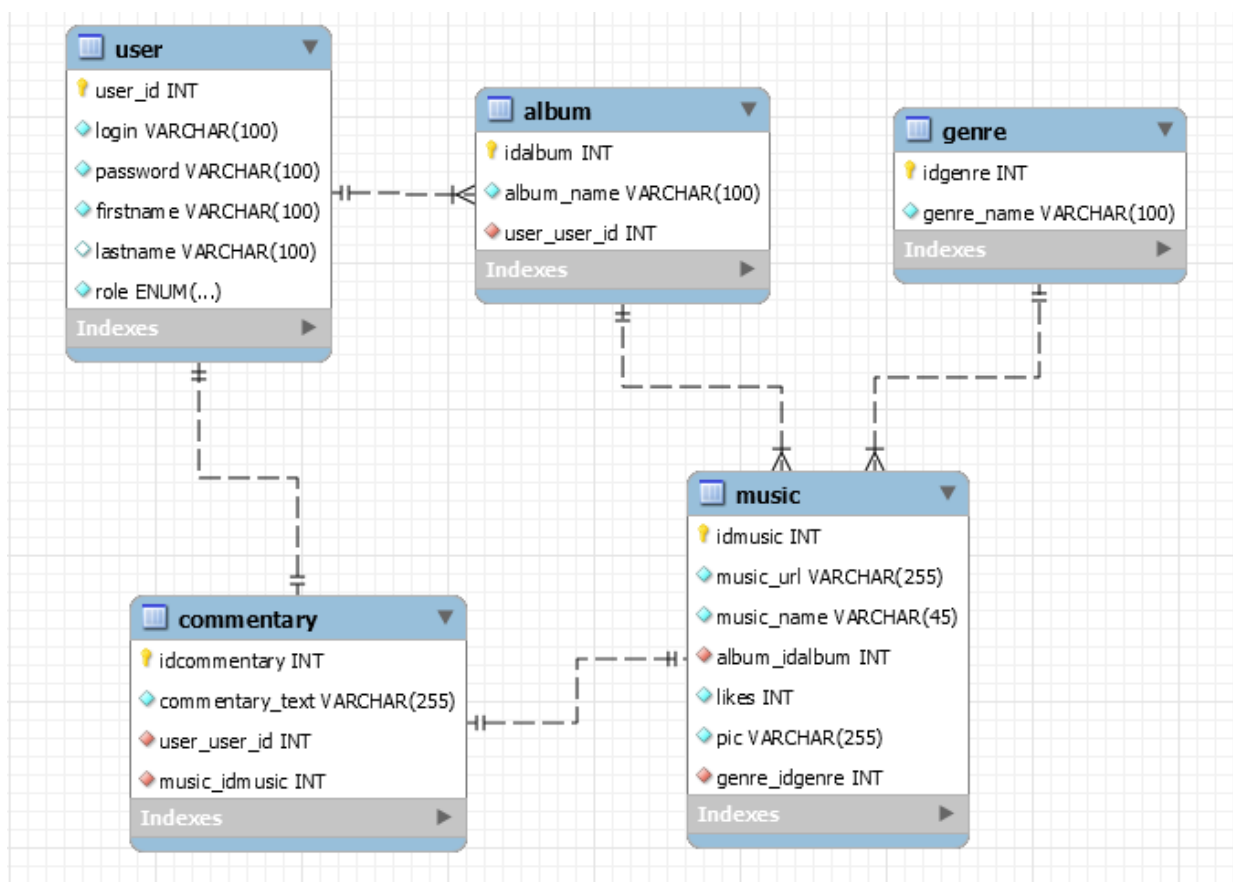


Рисунок 3.1 – Структура бази даних

Нижче, в таблиці 3.1 можна побачити усі поля розробленої бази даних та їх детальний опис.

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

Таблиця 3.1 – Опис таблиць бази даних

| Назва таблиці | Назва поля     | Тип даних                       | Детальний опис   |
|---------------|----------------|---------------------------------|--|
| User          | user_id        | int()                           | Ідентифікаційний номер користувача                         |
|               | login          | varchar(100)                    | Логін користувача  |
|               | password       | varchar(100)                    | Пароль користувача   |
|               | first_name     | varchar(100)                    | Ім'я користувача   |
|               | last_name      | varchar(100)                    | Прізвище користувача                                       |
|               | role           | ENUM("admin", "user", "artist") | Роль користувача (адмін, звичайний користувач, виконавець) |
| Music         | music_id       | int()                           | Ідентифікаційний номер музичної композиції                 |
|               | music_url      | varchar(255)                    | Ссилка на прослуховування та скачування музики             |
|               | music_name     | varchar(100)                    | Назва музики   |
|               | music_album_id | int()                           | Ідентифікатор музичного альбому                            |
|               | music_genre_id | int()                           | Ідентифікатор музичного жанру                              |
|               | music_likes    | int()                           | Кількість користувачів, котрим сподобалась ця музика       |
|               | music_pic      | varchar(255)                    | Посилання на картинку для музики                           |

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Змн. | Арк. | № докум. | Підпис | Дата |

### Продовження таблиці 3.1

| Назва таблиці | Назва поля | Тип даних    | Детальний опис  |
|---------------|------------|--------------|---|
| Album         | album_id   | int()        | Ідентифікаційний номер альбому                              |
|               | album_name | varchar(100) | Назва альбому   |
|               | artist_id  | int()        | Ідентифікаційний номер виконавця, котрому належить альбом   |
| Genre         | genre_id   | int()        | Ідентифікаційний номер жанру                                |
|               | genre_name | varchar(100) | Назва жанру   |
| Commentary    | comm_id    | int()        | Ідентифікаційний номер коментарю                            |
|               | music_id   | int()        | Ідентифікаційний номер музики                               |
|               | user_id    | int()        | Ідентифікаційний номер користувача, котрий залишив коментар |
|               | comm_text  | varchar(255) | Текст коментарю   |

### 3.2 Опис архітектури системи

Як я вже писав в попередньому розділі, я буду використовувати архітектурку модель проектування MVC. До Model віднесемо класи User, Music, Album, Genre, Commentary. User – клас який буде відповідати за користувача та містити його логін, пароль, ім'я, прізвище та роль. Music – вся інформація про музику (автор, назва, ідентифікатор в базі даних, жанр, альбом,

шлях місцезнаходження, кількість лайків). Album – це клас, який несе в собі назву альбому та ідентифікатор виконавця, котрому належить цей альбом. Commentary – клас, який відповідає за інформацію про текст коментарю ідентифікатори користувача та музики, де був залишен цей коментар. На рисунку 3.2 показано діаграми цих класів.

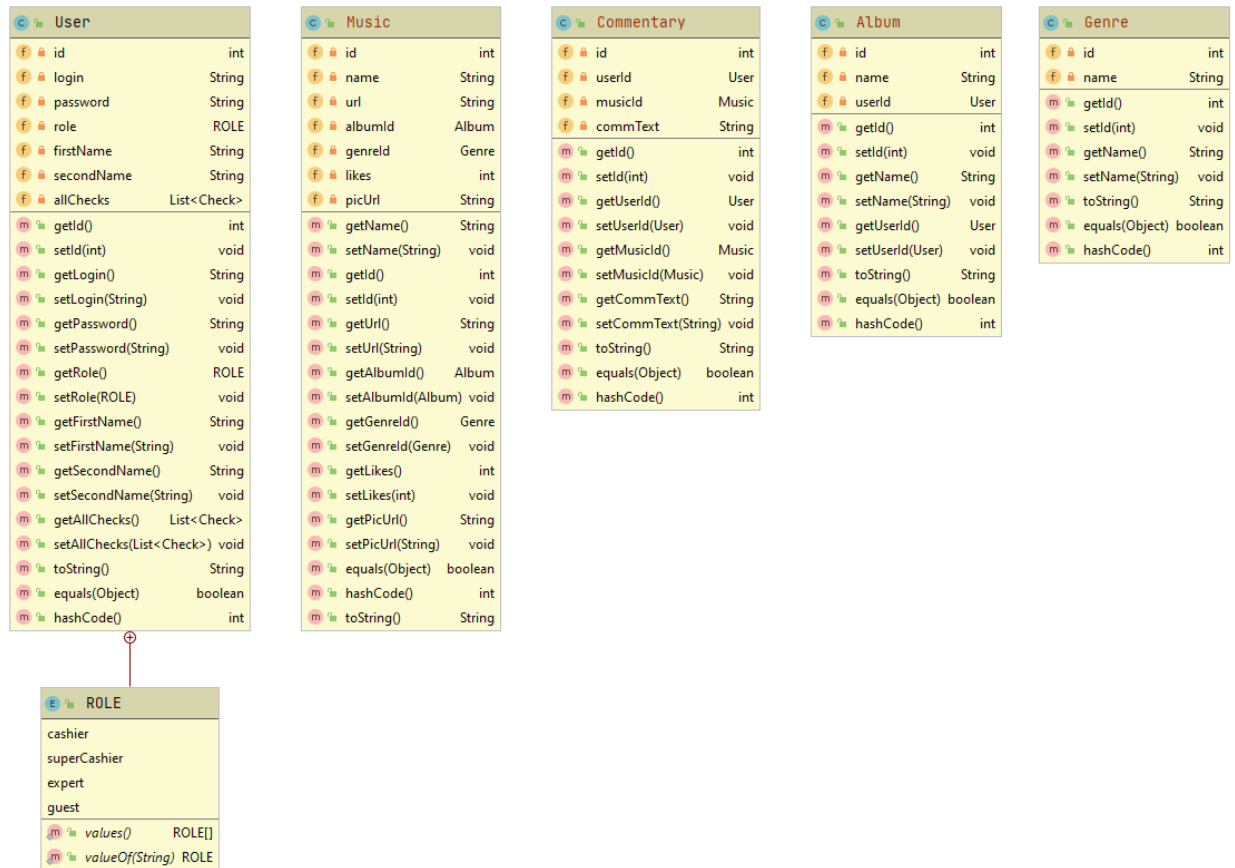


Рисунок 3.2 – Діаграма класів моделі

В цьому проєкті я вирішив зробити контроллер, який складається з одного сервлету [14] та багатьох команд. Функція сервлету заключається в тому, що він отримує команду через методи POST або GET та делегує її іншим контролерам, які вже виконують певні дії з моделю. Так можна облегшити розуміння коду та зробити його більш гнучким, тобто можна додавати будь яку кількість функціоналу не турбуючись про те, що цей код погано вплине на працездатність програми. Код класу Servlet наведено на рисунку 3.3. Далі я наведу декілька прикладів головних команд та опишу їх функції.

```

@Override
public void init(ServletConfig servletConfig) {
    context = servletConfig.getServletContext();
    commands = new ConcurrentHashSet<>();

    commands.put("index", new IndexCommand());
    commands.put("login", new LoginCommand());
    commands.put("register", new RegisterCommand());
    commands.put("logout", new LogoutCommand());
    commands.put("expert", new ArtistCommand());
    commands.put("expert/newProduct", new ArtistAddMusicCommand());
    commands.put("expert/changeAmount", new ArtistChangeMusicAlbumCommand());
    commands.put("cashier", new UserCommand());
    commands.put("cashier/addProductToCheck", new UserAddFavouriteMusicCommand());
    commands.put("superCashier/deleteProduct", new UserChangeProfileCommand());

    log.info("commands initialised");
}

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    process(req, resp);
}

@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    process(req, resp);
}

private void process(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    Command command = getCommand(req);
    String page = null;
    try {
        page = command.execute(req, resp);
    } catch (Exception e) {
        System.err.println("Exception in Servlet");
        e.printStackTrace();
    }
    if (page != null) {
        if (page.contains("redirect: ")) {
            resp.sendRedirect(location: req.getContextPath() + page.replaceAll(regex: "redirect: ", replacement: ""));
            log.info("redirect : " + page);
        } else {
            req.getRequestDispatcher(page).forward(req, resp);
            log.info("forward : " + page);
        }
    }
}
}

```

Рисунок 3.3 – Код класу Servlet

### 3.2.1 RegisterCommand

RegisterCommand відповідає за реєстрацію користувача у системі. В цій команді описано функції для перевірки правильності вводу усіх полів, створення нового користувача та підключення його до сесії. Діаграма класу показана на рисунку 3.4.

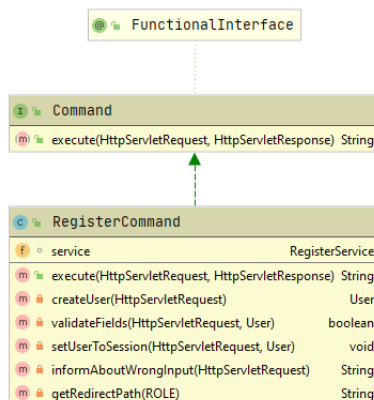


Рисунок 3.4 – Діаграма класу RegisterCommand

### 3.2.2 LogInCommand

LogInCommand відповідає за вхід до аккаунту користувача у системі. Метод logout завершує попередню сесію, якщо така взагалі існувала. Потім, команда виконує метод checkLoginAndPassword, в яку входить відразу декілька функцій, а саме перевірка правильності вводу даних та автентифікація. Далі, команда створює нову сесію та додає до неї користувача. Діаграма класу показана на рисунку 3.5.

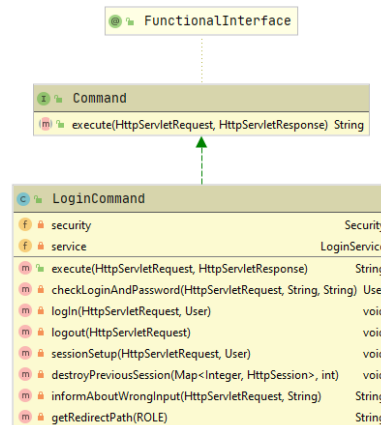


Рисунок 3.5 – Діаграма класу LogInCommand

### 3.2.3 UserAddFavouriteCommand

UserAddFavouriteMusicCommand (рисунок 3.6) виконує функції додавання музики до альбому вподобаних. Спочатку, команда перевіряє наявність музики в БД та додає їй один лайк. Потім, виконується функція зміни персонального альбому вподобаних пісень користувача, де додається обрана музична композиція.

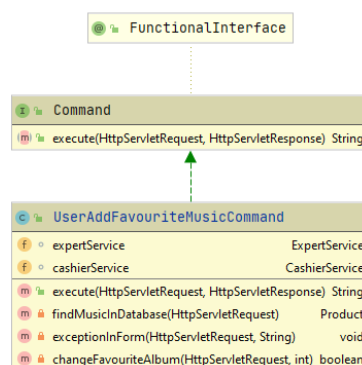


Рисунок 3.6 – Діаграма класу UserAddFavouriteMusicCommand



при реєстрації не заповнене поле, або пароль та підтвердження пароля співпадають, то виконується переправлення на сторінку з помилкою наповненості полів. При успішній реєстрації, користувач додається в базу даних всіх користувачів, йому видаються права (user). Це означає, що користувач з такими правами може слухати музику, додавати до вподобаних, коментувати, знаходити бажані композиції, змінювати профіль, завантажувати пісні. Після завершення бажаних операцій, користувач може вийти зі свого профілю і завершити сесію. Для повторного входу до власного кабінету потрібно пройти наступні етапи: автентифікація і авторизація.

### 3.3.2 Алгоритми автентифікації та авторизації

Автентифікація – це процес перевірки доступу користувача до інформації за пред’явленим ідентифікатором (в моєму випадку – за логіном та паролем). Система перевіряє введені логін та пароль з даними з БД та якщо такі дані відсутні – видає помилку, де написано, що такого користувача не існує, а при успішній перевірці – виконує алгоритм авторизації. Авторизація – це процедура отримання доступу користувачу у відповідності до його прав доступу. У системі «Бібліотека музичних творів» існує чотири види прав доступу:

- Unsigned – користувач, котрий не зареєструвався або не пройшов автентифікацію. Він має тільки можливість шукати та прослуховувати музику
- User – користувач, котрий пройшов автентифікацію та має можливість змінювати свій профіль, «лайкати» улюблені композиції, коментувати та завантажувати їх.
- Artist – користувач, котрий, крім прав звичайного користувача, має можливість створювати альбоми, додавати в них свої пісні, редагувати та видаляти їх.
- Admin – адміністратор веб-сервісу, який має можливості видаляти пісні, альбоми, користувачів та коментарі.

|      |      |          |        |      |                    |      |
|------|------|----------|--------|------|--------------------|------|
|      |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                    | 43   |

Після успішної авторизації, користувачу надаються права та можливості користування веб-сервісом згідно до наданої йому ролі.

### **3.3.3 Алгоритм додавання музики до альбому улюблених пісень**

При реєстрації у користувача немає пісень, які йому подобаються. Ця функція відтворена, щоб користувач міг легко та швидко знаходити улюблені пісні. Коли користувач вперше «лайкає» композицію, яка йому сподобалась, на сервер передається ця інформація та створюється новий альбом з ідентифікатором користувача. Після цього, та кожні наступні рази, коли користувач «лайкає» композицію, ідентифікатор музичного твору додається до цього альбому. Далі, коли користувач натискає на новостворений альбом, сервер надсилає базі даних запит на отримання ідентифікатора альбому і в кінці користувач отримує список його улюблених пісень.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 44          |

### Висновок до розділу 3

У цьому розділі було створено базу даних та проаналізовано усі її поля. Також було детально описано архітектурну модель MVC та класи, що виконують роль Model та Controller. До Model відносяться такі класи як User, Music, Album, Genre та Commentary, а до Controller – багато класів-команд, роботу яким делегує клас один Servlet. Також було проведено детальний опис основних алгоритмів програми.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 45          |

# РОЗДІЛ 4

## ТЕСТУВАННЯ ПРОГРАМИ

### 4.1 Тестування головної сторінки.

При вході на сайт, вас зустрічає початкова сторінка (рисунок 4.1), яка дозволяє користувачу, який не пройшов реєстрацію або автентифікацію, користуватися основними функціями стрімінгово аудіосервісу, а саме прослуховування музики.

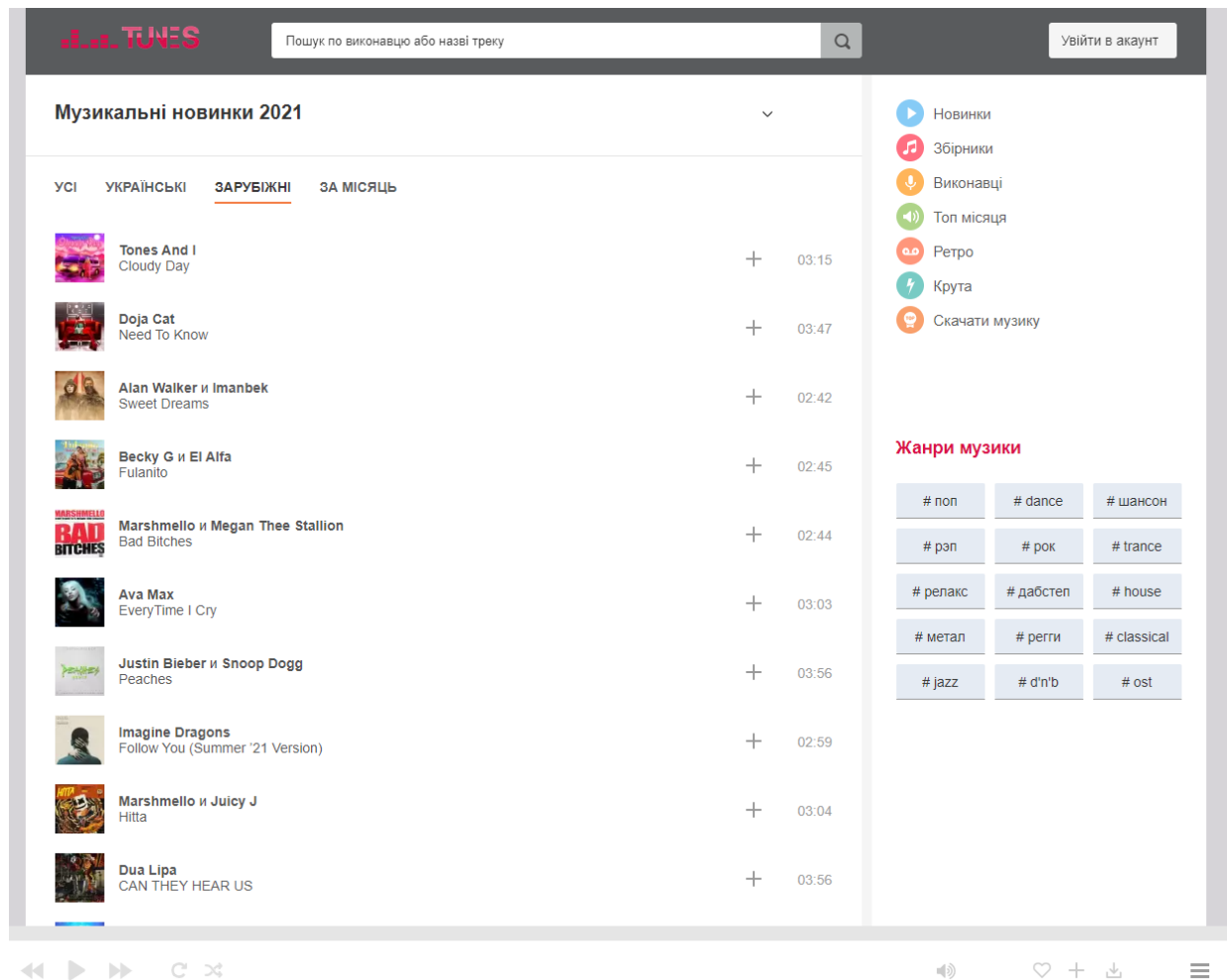


Рисунок 4.1 – Головна сторінка сервісу

Користувач може обрати будь-яку пісню зі списку та натиснути на її. Після цього внизу головної сторінки з'явиться плеєр з обраною композицією (рисунок 4.2).



Рисунок 4.2 – Плеєр обраної композиції

Також, користувач може скористатися пошуком або підбірками, що розташовані справа на головній сторінці, щоб з легкістю знайти потрібні йому пісні.

В будь-який час, користувач може увійти в свій обліковий запис або зареєструвати новий. Для цього йому потрібно натиснути на кнопку, що знаходиться зверху у правому кутку. Вона перенаправить його на сторінку автентифікації (рисунок 4.3).

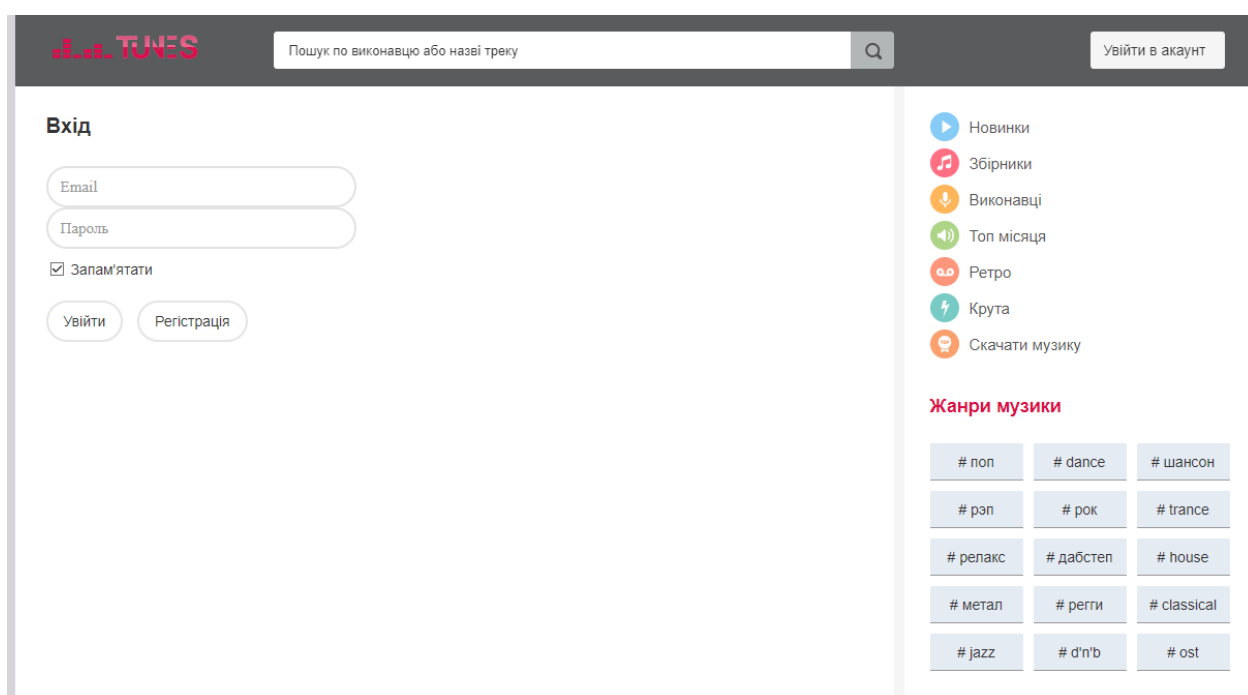


Рисунок 4.3 – Сторінка автентифікації

## 4.2 Тестування сторінки автентифікації та реєстрації

На цій сторінці користувач повинен ввести свій email та пароль, щоб увійти у свій обліковий запис. Якщо в нього нема такого облікового запису, він повинен зареєструвати новий, натиснувши кнопку «Регістрація» (рисунок 4.4).

## Вхід

Email

Пароль

Запам'ятати

Увійти    Регістрація

Рисунок 4.4 – Вікно входу в обліковий запис

Після цього, користувача перенаправить на сторінку реєстрації. Задля успішного проходження процесу реєстрації, він потрібен ввести свій email, ім'я, пароль та підтвердити свій пароль (рисунок 4.5).

## Регістрація

Email

Пароль

Пароль ще раз

Ім'я

Зареєструватися

Рисунок 4.5 – Сторінка реєстрації

Після успішної реєстрації або входу в обліковий запис, на головній сторінці зверху справа з'явиться новий віджет з випадаючим списком, через який можна потрапити до особистого кабінету, натиснувши кнопку «Профіль» (рисунок 4.6).

|      |      |          |        |      |                    |      |
|------|------|----------|--------|------|--------------------|------|
|      |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                    | 48   |

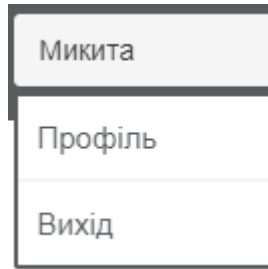


Рисунок 4.6 – Випадаючий список

### 4.3 Тестування сторінки особистого кабінету

На цій сторінці (рисунок 4.7) користувач може змінити своє ім'я, пароль або зовсім вийти зі свого облікового запису. Також, він має можливість переглядати список улюблених треків, виконавців або плейлистів.

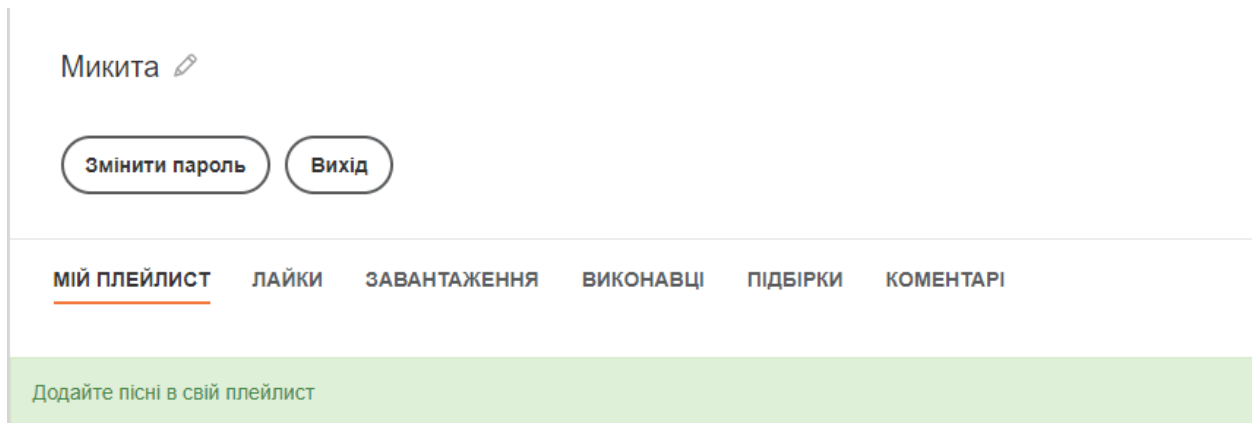


Рисунок 4.7 – Сторінка особистого кабінету

Натиснувши на олівець біля свого ім'я, користувач може змінити його і для підтвердження цієї дії, він повинен натиснути кнопку «Зберегти» (рисунок 4.8).

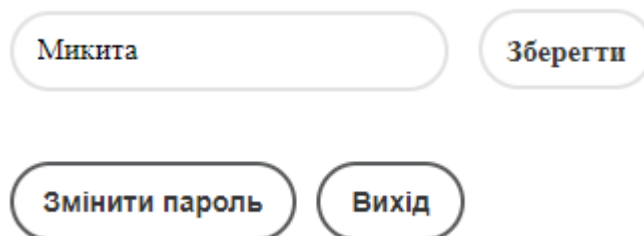


Рисунок 4.8 – Можливість користувача змінити ім'я

У вкладинці «Мій плейлист» користувач має змогу переглядати та слухати пісні, який він додав до спеціального плейлисту (рисунок 4.9).

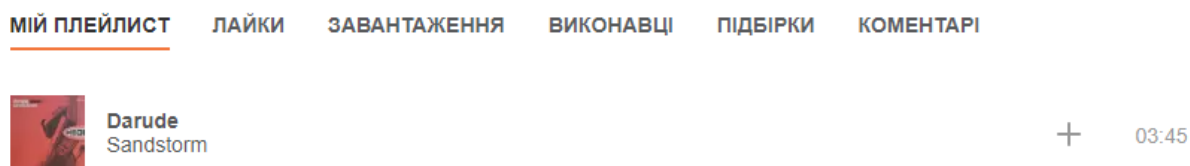


Рисунок 4.9 – вкладинка «Мій плейлист»

У вкладинці «Лайки» користувач має можливість переглядати та слухати пісні, що сподобались йому (рисунок 4.10).

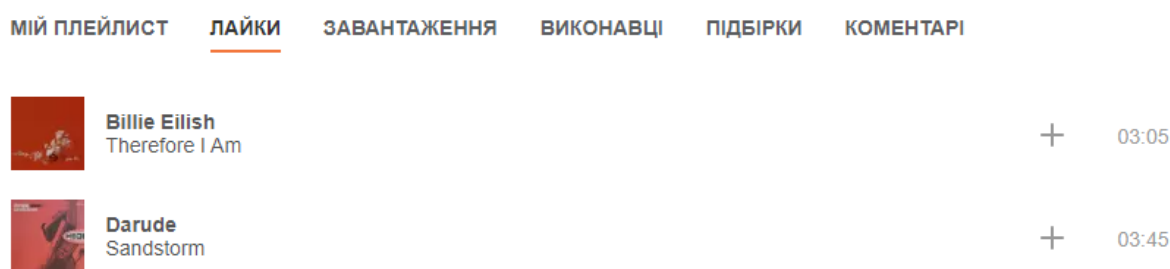


Рисунок 4.10 – вкладинка «Лайки»

У вкладинці «Завантаження» користувач має змогу обрати та завантажити будь-який трек у форматі .mp3 (рисунок 4.11).

## Завантажити

Завантажено: 0 треків

Оберіть mp3 файл

Выберите файл | Файл не выбран

Отправить

Рисунок 4.11 – вкладинка «Завантаження»

У вкладинці «Виконавці» та «Підбірки» користувач може переглянути улюблених виконавців (рисунок 4.12) та плейлисти (рисунок 4.13), що йому сподобались, відповідно.

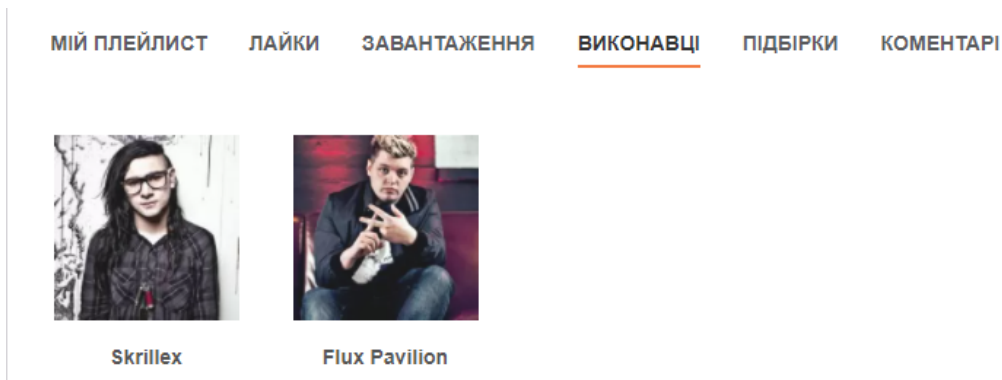


Рисунок 4.12 – вкладинка «Виконавці»

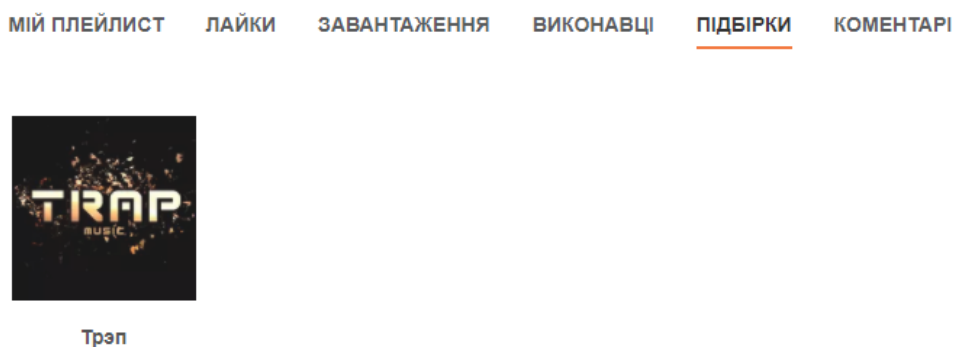


Рисунок 4.13 – вкладинка «Підбірки»

У вкладинці «Коментарі» користувач може переглянути коментарі, які він залишав, з вказаними піснями та часом (рисунок 4.14).

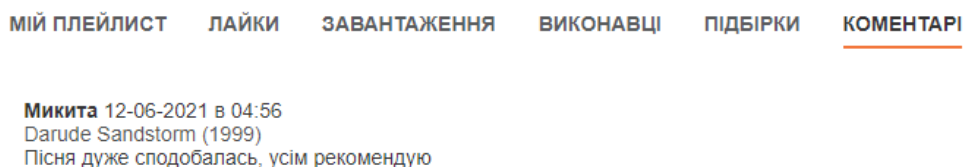


Рисунок 4.14 – вкладинка «Коментарі»

## 4.4 Тестування сторінок пісні, виконавця та альбому

З головної сторінки або сторінки особистого кабінету користувач може потрапити на сторінку музичної композиції (рисунок 4.15). Щоб зробити це, він повинен натиснути на назву пісні.

### Darude — Sandstorm

Бітрейт: 320 kbps  
Розмір: 8.69 Mb  
Тривалість: 03:45  
Прослуховувань: 9794



# edm хіти # trance

### Darude – Sandstorm слухати онлайн

|  |  |   |       |
|--|--|---|-------|
|  | Darude<br>Sandstorm                                | + | 03:45 |
|  | Darude и Robert Miles<br>Children of the Sandstorm | + | 07:02 |
|  | Darude<br>Feel The Beat                            | + | 08:39 |

Усі пісні Darude

Рисунок 4.15 – Сторінка музичної композиції

На цій сторінці показана основна інформація щодо цієї пісні, а саме бітрейт, розмір файлу, тривалість музичної композиції, кількість прослуховувань та спеціальні теги, що покращують пошук схожих композицій. Також, внизу є підбірка інших музичних творів виконавця. На цій сторінці користувач може прослуховувати, скачати пісню або додати її до свого плейлісту.

|      |      |          |        |      |                    |      |
|------|------|----------|--------|------|--------------------|------|
|      |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                    | 52   |



## Darude

75 треків. 11102 прослуховувань

# trance

Даруд – фінський продюсер і DJ

▶ Слухати



D / Darude

### Усі пісні Darude

|  |   |   |       |
|--|---|---|-------|
|  | <b>Darude и Robert Miles</b><br>Children of the Sandstorm               | + | 07:02 |
|  | <b>Darude</b><br>Feel The Beat  | + | 08:39 |
|  | <b>Darude</b><br>Sandstorm  | + | 03:45 |
|  | <b>Darude feat. Sebastian Rejman</b><br>Look Away                       |   | 02:59 |
|  | <b>Darude, Ashley Wallbridge feat. Foux</b><br>Surrender                |   | 03:20 |
|  | <b>Darude, Ashley Wallbridge feat. Foux</b><br>Surrender (Extended Mix) |   | 05:28 |
|  | <b>Darude</b><br>Tell Me (Original Radio Edit)                          |   | 03:10 |

Рисунок 4.16 – Сторінка виконавця

На сторінці виконавця (див. рисунок 4.16) користувач отримає головну інформацію про творця музики (а саме, кількість його треків, кількість усіх прослуховувань на сайті, жанр, в якому він працює та невеличку інформацію про самого виконавця, хто він та звідки) та список усіх його пісень. Також, користувач може слухати тут всі його композиції або, натиснувши спеціальну кнопку (рисунок 4.17), додати його до улюблених виконавців.

|      |      |          |        |      |                    |      |
|------|------|----------|--------|------|--------------------|------|
|      |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                    | 53   |



Рисунок 4.17 – Кнопка додавання виконавця до улюблених

На сторінці альбому або підбірці (рисунок 4.18) показано дату, коли він був створений, кількість треків у ньому та рейтинг користувачів, що слухали цей альбом. Головне, що відрізняє цю сторінку від інших, так це наявність трьох видів сортування музики: за датою випуску, за популярністю та за алфавітом. Альбом також, як і на попередніх твоїх сторінках, можна додати до улюблених.

**Трап**  
08.06.2021 в 00:44 22 треку  
★★★★☆ 4.04 (голосів: 39)  
▶ Слухати × ↗

ЗА ДАТОЮ ЗА ПОПУЛЯРНІСТЮ ЗА АЛФАВІТОМ

|  |  |   |       |
|--|--|---|-------|
|  | <b>CryJaxx</b><br>Beautiful Liar                               | + | 03:09 |
|  | <b>Delta Heavy</b><br>Kaleidoscope (Rene LaVice Remix)         | + | 04:27 |
|  | <b>Ekali</b><br>Braids   | + | 02:46 |
|  | <b>Fabian Mazur</b><br>The Life                                | + | 03:18 |
|  | <b>Gucci Mane</b><br>Richer Than Errbody (Juelz & Bishu Remix) | + | 03:39 |

Рисунок 4.18 – Сторінка альбому

|      |      |          |        |      |                    |      |
|------|------|----------|--------|------|--------------------|------|
|      |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                    | 54   |

## Висновки до розділу 4

Було розроблено та протестовано стрімінговий аудіо сервіс «Бібліотека музичних творів». Усі операції перевірені та працюють без помилок. Виправлені помилки, що могли виникнути при використанні даного сервісу. Надано детально інструкцію з використання веб додатку та наведено декілька прикладів функціонування програми.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 55          |



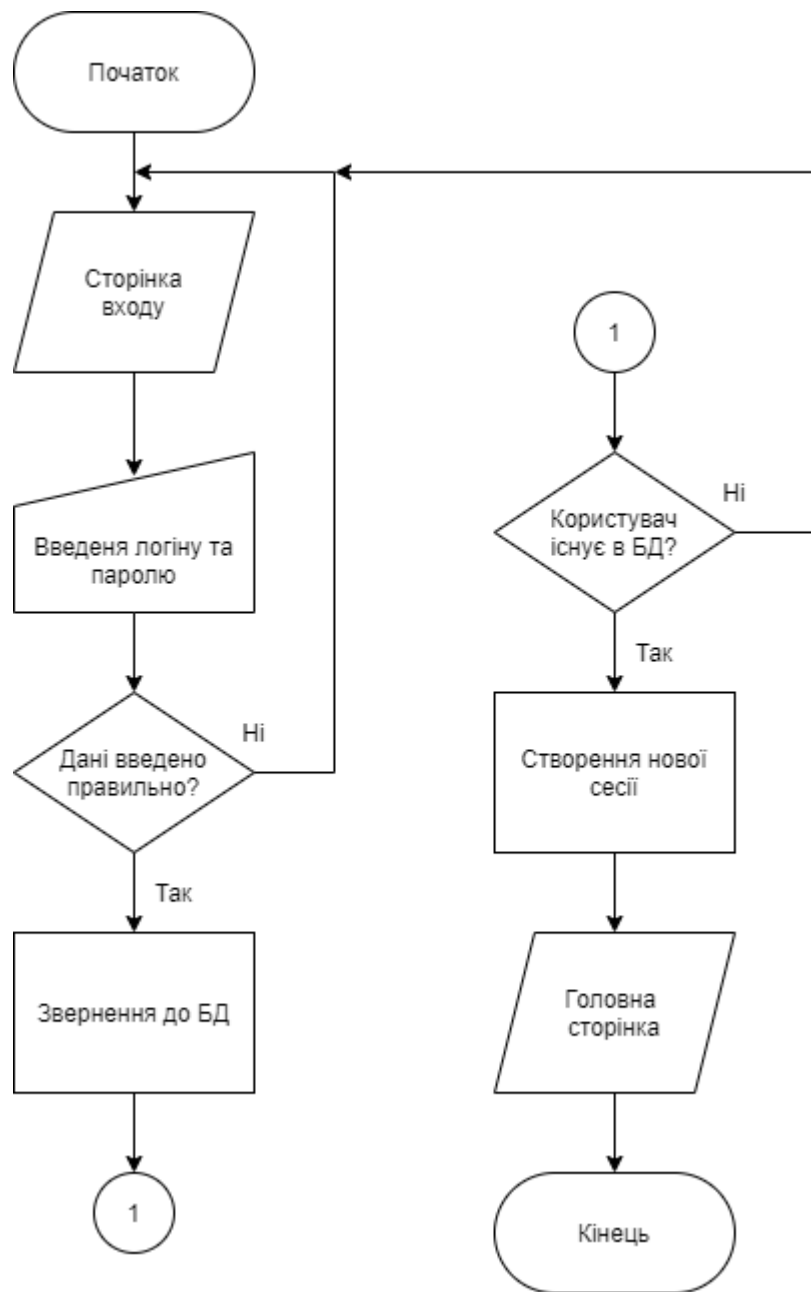
Було протестовано основні функції програми та надано детальну інструкцію користування веб додатком, в якому можна побачити, що усі завдання та задачі бакалаврського проєкту були виконані та реалізовані. Веб сервіс «Бібліотека музичних творів» («Digital music library») готова до використання в цілях отримання адміністратором проєкту грошової вигоди.

|             |             |                 |               |             |                           |             |
|-------------|-------------|-----------------|---------------|-------------|---------------------------|-------------|
|             |             |                 |               |             | <i>ІАЛЦ.467200.003 ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                           | 57          |



**Додаток А**  
**Структурна схема алгоритму автентифікації в системі**  
**до дипломного проєкту**  
**на тему: «Бібліотека музичних творів»**

Київ – 2021 року



|                  |             |                       |               |             |
|------------------|-------------|-----------------------|---------------|-------------|
|                  |             |                       |               |             |
|                  |             |                       |               |             |
|                  |             |                       |               |             |
|                  |             |                       |               |             |
| <i>Змн.</i>      | <i>Арк.</i> | <i>№ докум.</i>       | <i>Підпис</i> | <i>Дата</i> |
| <i>Розроб.</i>   |             | <i>Клецун М.С.</i>    |               |             |
| <i>Перевір.</i>  |             | <i>Сімоненко А.В.</i> |               |             |
| <i>Т. Контр.</i> |             |                       |               |             |
| <i>Н. Контр.</i> |             | <i>Сімоненко В.П.</i> |               |             |
| <i>Затверд.</i>  |             |                       |               |             |

ІАЛЦ.467200.004 Д1

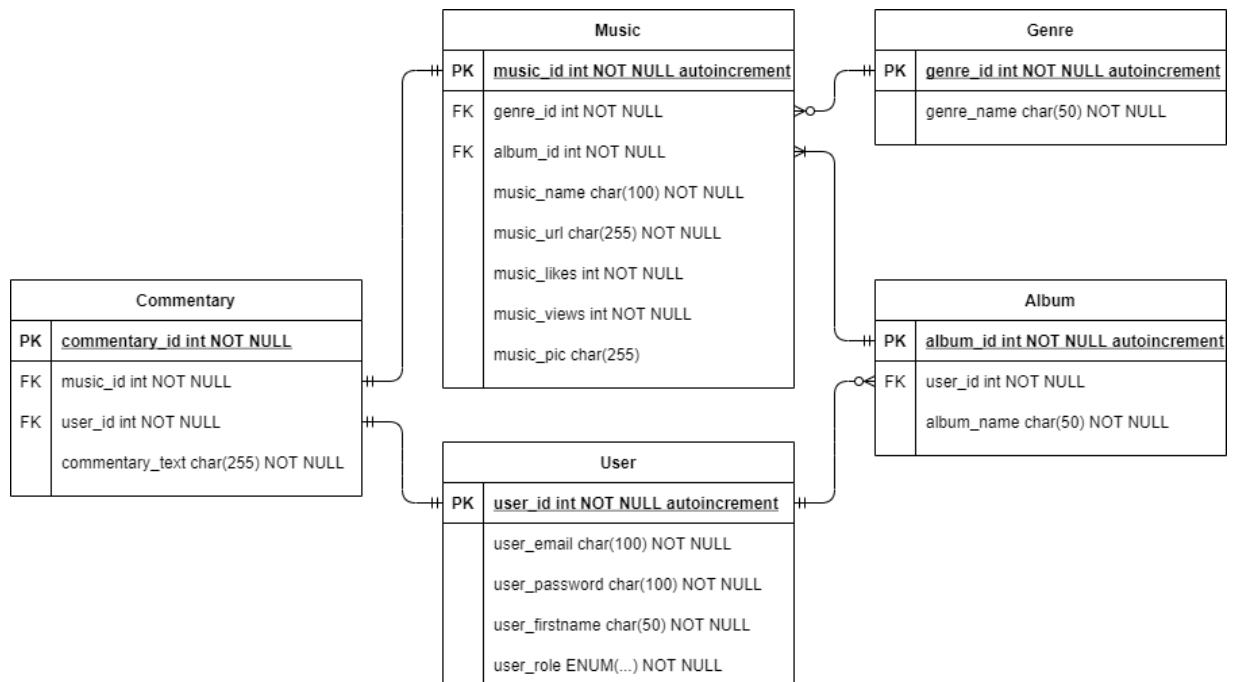
Бібліотека музичних творів  
 Структурна схема алгоритму  
 автентифікації в системі

|             |             |                  |
|-------------|-------------|------------------|
| <i>Літ.</i> | <i>Маса</i> | <i>Масштаб</i>   |
|             |             |                  |
| <i>Арк.</i> | <i>1</i>    | <i>Аркушів 1</i> |

НТУУ «КПІ», ФІОТ  
 ІО-72

**Додаток Б**  
**Функціональна схема бази даних бібліотеки музичних творів**  
**до дипломного проєкту**  
**на тему: «Бібліотека музичних творів»**

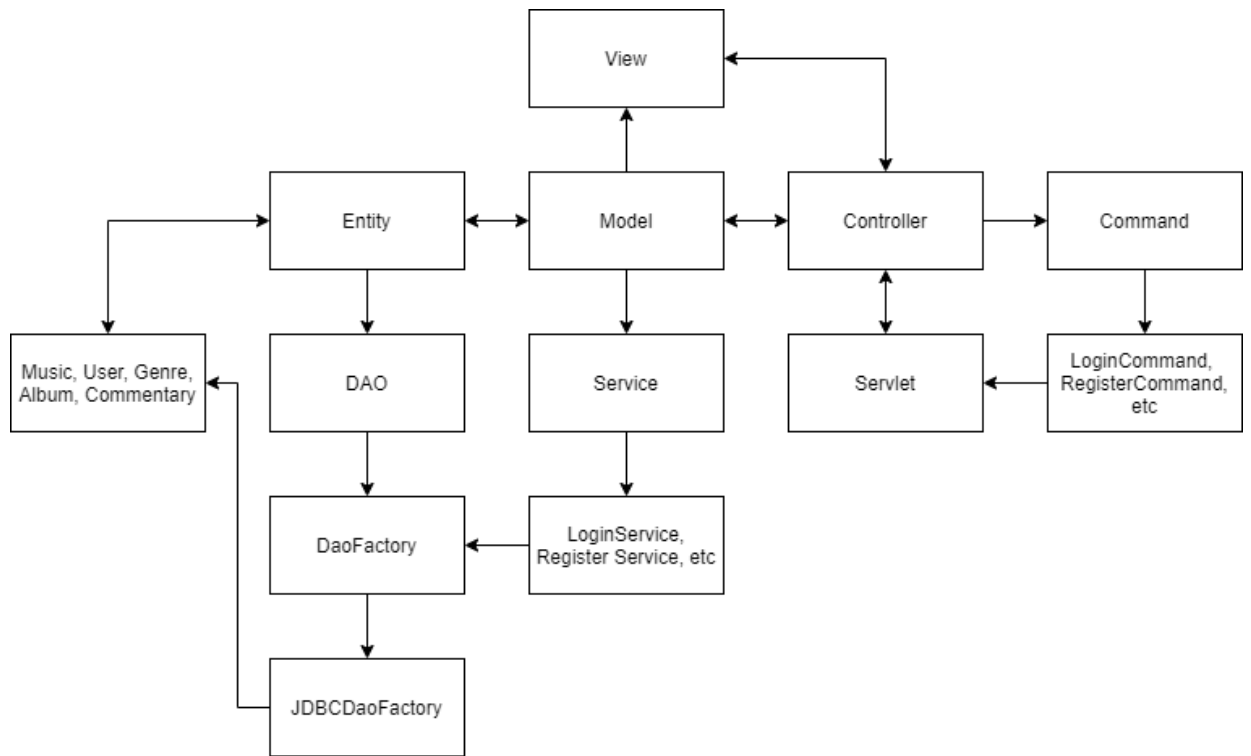
Київ – 2021 року



|           |                |          |        |      |  |  |  |      |      |         |
|-----------|----------------|----------|--------|------|--|--|--|------|------|---------|
|           |                |          |        |      | <i>ІАЛЦ.467200.005 Д2</i>  |  |  |      |      |         |
|           |                |          |        |      | Бібліотека музичних творів<br>Функціональна схема бази даних<br>бібліотеки музичних творів |  |  | Лім. | Маса | Масштаб |
| Змн.      | Арк.           | № докум. | Підпис | Дата |  |  |  |      |      |         |
|           |                |          |        |      | Арк. 1      Аркушів 1<br><br>НТУУ «КПІ», ФІОТ<br>ІО-72                                     |  |  |      |      |         |
|           |                |          |        |      |  |  |  |      |      |         |
| Розроб.   | Клецюн М.С.    |          |        |      |  |  |  |      |      |         |
| Перевір.  | Сімоненко А.В. |          |        |      |  |  |  |      |      |         |
| Т. Контр. |                |          |        |      |  |  |  |      |      |         |
| Н. Контр. | Сімоненко В.П. |          |        |      |  |  |  |      |      |         |
| Затверд.  |                |          |        |      |  |  |  |      |      |         |

**Додаток В**  
**Принципова схема системи бібліотеки музичних творів**  
**до дипломного проєкту**  
**на тему: «Бібліотека музичних творів»**

Київ – 2021 року



|                  |             |                       |               |             |  |               |                  |                |
|------------------|-------------|-----------------------|---------------|-------------|--|---------------|------------------|----------------|
|                  |             |                       |               |             | <i>ІАЛЦ.467200.006 ДЗ</i>                                  |               |                  |                |
|                  |             |                       |               |             | <i>Бібліотека музичних творів</i>                          | <i>Літ.</i>   | <i>Маса</i>      | <i>Масштаб</i> |
| <i>Змн.</i>      | <i>Арк.</i> | <i>№ докум.</i>       | <i>Підпис</i> | <i>Дата</i> |  |               |                  |                |
| <i>Розроб.</i>   |             | <i>Клецун М.С.</i>    |               |             | <i>Принципова схема системи бібліотеки музичних творів</i> |               |                  |                |
| <i>Перевір.</i>  |             | <i>Сімоненко А.В.</i> |               |             |  |               |                  |                |
| <i>Т. Контр.</i> |             |                       |               |             |  | <i>Арк. 1</i> | <i>Аркушів 1</i> |                |
| <i>Н. Контр.</i> |             | <i>Сімоненко В.П.</i> |               |             | <i>НТУУ «КПІ», ФІОТ</i>                                    |               |                  |                |
| <i>Затверд.</i>  |             |                       |               |             | <i>ІО-72</i>   |               |                  |                |