



## ДИПЛОМНА РОБОТА МАГІСТРА

на тему Інформаційна технологія завантаження слів пісень в реальному часі  
для мобільних пристроїв

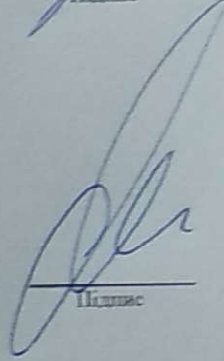
Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності

Виконав: студент 2 курсу, група КНМ-19-1  П.О. Грицишин  
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ  О.А. Пасічник  
Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ  Р.О. Багрії  
Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри КНІТ, д.т.н., професор  О.В. Бармак  
Ініціали, прізвище  
7 12 2020 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра комп'ютерних наук та інформаційних технологій

Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних  
наук та інформаційних технологій

(підпис)

д.т.н., професор О.В. Бурлак

« 7 » 9 2020 року

ЗАВДАННЯ  
НА ДИПЛОМНУ РОБОТУ МАГІСТРА

1. Тема дипломної роботи магістра: «Інформаційна технологія зв'язування слів пісень в реальному часі для мобільних пристроїв»

2. Завдання видано студенту Грицишину Павлу Олександровичу

(прізвище, ім'я, по батькові)

3. Керівник роботи к.т.н., доцент Пасічник Олександр Анатолійович

(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від « 9 » 9 2020 р. № 22

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи полягає у реалізації інформаційної технології зв'язування слів пісень в реальному часі для мобільних пристроїв з використанням загальнодоступних мережевих аудіо-хостингів.

## Реферат

Дипломна робота магістра присвячена розробці інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв із використанням загальнодоступних мережевих аудіо-хостингів.

**Актуальність теми.** Доведено, що будь-яке навчання є ефективнішим в процесі гри чи квесту. Тоді новий матеріал сприймається легше та засвоюється краще. Наприклад, можна вивчити чи покращити рівень володіння англійською, слухаючи пісні, перекладаючи незнайомі слова, або, дивлячись фільми англійською з субтитрами. Починаючи вивчати нову мову, мозок людини має справу з новими йому звуками, які невластиві нашому мовленнєвому апарату. Прослуховуючи пісні англійською, можна мимоволі, не помічаючи цього, звикнути до нових слів та їх правильної вимови. Саме у такий спосіб людина ефективно вчиться розставляти смислові акценти та вловлює ритм нової мови. Слова з пісень засвоюються набагато легше, ніж, наприклад, зі словника, бо приспіву можуть повторюватись декілька разів, до того ж, можна власноруч повернутись до важкого для розуміння моменту у пісні та послухати його ще раз.

Сучасний розвиток техніки та технологій виводить на якісно новий рівень практично всі сфери життєдіяльності людини. На заміну та допомогу традиційним підручникам приходять технічні пристрої та сучасні технології. Тобто, вивчення іноземної мови може відбуватися за допомогою смартфона або планшета, які є під рукою практично завжди.

Одним із методологічних підходів вивчення іноземних мов, який довів свою високу ефективність, є спілкування з носіями мови. Певним віртуальним замінником такого носія може слугувати пісенний виконавець. Музика дуже корисна при вивченні іноземних мов, а особливо в освоєнні граматики, збільшенні словникового запасу і постановці вимови. Крім того, відомо, що прослуховування класичної музики стимулює мозкову активність, покращує уважність, зосередженість, пам'ять. Більшість пісень містить стійкі вирази і

фрази, а їх тексти розраховані на слухачів-носіїв мови і включають в себе сучасні розмовні обороти [35].

Інформаційний простір сучасності характеризується наявністю великої кількості аудіохостингів з приблизно однаковим функціоналом. Ці ресурси надають доступ до аудіо продукції, здебільшого у mp3 форматі. Їхнє головне призначення полягає у наданні користувачу можливості прослуховування відповідних файлів.

З точки зору вивчення іноземної мови доцільним та необхідним додатком до аудіотреку є текст відповідного пісенного твору. Така необхідність полягає у можливості перевірки правильності результатів аудіювання.

Таким чином, питання отримання тексту відповідного пісенного твору є актуальним на сучасному етапі розвитку людства.

**Мета і задачі роботи.** Мета роботи полягає у реалізації інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв з використанням загальнодоступних мережевих аудіо-хостингів.

Для досягнення поставленої мети визначені наступні задачі досліджень:

- провести аналіз існуючих методів, технологій та рішень завантаження слів пісень в реальному часі для мобільних пристроїв;
- удосконалення методики завантаження слів пісень у напрямку отримання максимально повних і достовірних даних;
- розробити інформаційну технологію завантаження слів пісень в реальному часі для мобільних пристроїв;
- виконати експериментальну перевірку інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

**Об'єкт дослідження** – процес завантаження слів пісень в реальному часі для мобільних пристроїв.

**Предмет дослідження** – моделі, методи, підходи та засоби інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

**Методи дослідження.** Для розв'язання поставлених задач використовуються основні положення методів аналізу даних, кодування та

декодування аудіо файлів; для удосконалення методики завантаження слів пісень протоколи обміну даними, організація і керування базами даних; для реалізації інформаційної технології - технології проектування інформаційних систем та об'єктно орієнтований підхід.

**Наукова новизна одержаних результатів.** В результаті проведеної роботи були отримані такі результати:

- Удосконалено існуючі методи завантаження слів пісень в реальному часі для мобільних пристроїв у напрямку отримання максимально повних і достовірних даних.

**Практичне значення одержаних результатів.** В результаті виконання дипломної роботи магістра розроблено відповідне програмне забезпечення, яке підтвердило вірність запропонованих положень. Застосування інформаційної технології дає можливість завантаження слів пісень в реальному часі для мобільних пристроїв у напрямку отримання максимально повних і достовірних даних.

**Апробація результатів дипломної роботи.** Основні наукові та практичні результати опубліковані в наукових виданнях МОН України:

- публікація на тему “Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв” в науковому журналі “Вісник Хмельницького національного університету”.

За темою дипломної роботи магістра автором виконано одну наукову публікацію [36].

**Структура та обсяг роботи.** Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 35 найменувань та 5 додатків. Загальний обсяг дипломної роботи магістра становить 72 сторінки, з них 65 сторінок основного тексту та 7 сторінок додатків. У роботі наведено 51 рисуноків та 1 таблиця.

**Ключові слова:** інформаційна система, аудіотвори, вивчення мови.

## Зміст

Перелік скорочень	4
Вступ	5
Розділ 1	
Аналіз сучасного стану проблеми автоматизації формування тестових завдань	8
1.1 Аналіз предметної області	8
1.2 Аналіз сучасного програмного забезпечення предметної області	9
1.2.1 Аналіз існуючих аудіохостингів	9
1.2.2 Аналіз існуючих сервісів пошуку музики та слів для мобільних пристроїв	11
1.3 Огляд засобів реалізації	16
1.4 Висновки до розділу та постановка задачі	16
Розділ 2	
Розробка інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв	17
2.1 Основні етапи і процедури	17
2.2 Опис сервісів для пошуку медіаданих	19
2.3 Висновки до розділу	22
Розділ 3	
Проектування структури інформаційної технології	23
3.1 Аналіз та автоматизація обробки інформаційних потоків	23
3.2 Розробка структури інформаційної технології	27
3.3 Вибір засобів розробки інформаційної технології	29
3.4 Структура і функціональне призначення модулів системи, їх взаємозв'язок	32
3.5 Розробка інтерфейсу користувача	38
3.6 Висновки до розділу	46
Розділ 4	

Апробація програмної інформаційної технології	47
4.1 Висновки до розділу	58
Загальні висновки	59
Перелік посилань	60
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
БД	База даних
СКБД	Система керування базою даних
ПЗ	Програмне забезпечення
ІТ	Інформаційні технології
ДРМ	Дипломна робота магістра
SDK	Software development kit
API	Application program interface
DAO	Data access objects
ІС	Інформаційна система
ОС	Операційна система
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language

## Вступ

**Актуальність теми.** Доведено, що будь-яке навчання є ефективнішим в процесі гри чи квесту. Тоді новий матеріал сприймається легше та засвоюється краще. Наприклад, можна вивчити чи покращити рівень володіння англійською, слухаючи пісні, перекладаючи незнайомі слова, або, дивлячись фільми англійською з субтитрами. Починаючи вивчати нову мову, мозок людини має справу з новими йому звуками, які невластиві нашому мовленнєвому апарату. Прослуховуючи пісні англійською, можна мимоволі, не помічаючи цього, звикнути до нових слів та їх правильної вимови. Саме у такий спосіб людина ефективно вчиться розставляти смислові акценти та вловлює ритм нової мови. Слова з пісень засвоюються набагато легше, ніж, наприклад, зі словника, бо приспівки можуть повторюватись декілька разів, до того ж, можна власноруч повернутись до важкого для розуміння моменту у пісні та послухати його ще раз.

Сучасний розвиток техніки та технологій виводить на якісно новий рівень практично всі сфери життєдіяльності людини. На заміну та допомогу традиційним підручникам приходять технічні пристрої та сучасні технології. Тобто, вивчення іноземної мови може відбуватися за допомогою смартфона або планшета, які є під рукою практично завжди.

Одним із методологічних підходів вивчення іноземних мов, який довів свою високу ефективність, є спілкування з носіями мови. Певним віртуальним заміником такого носія може слугувати пісенний виконавець. Музика дуже корисна при вивченні іноземних мов, а особливо в освоєнні граматики, збільшенні словникового запасу і постановці вимови. Крім того, відомо, що прослуховування класичної музики стимулює мозкову активність, покращує уважність, зосередженість, пам'ять. Більшість пісень містить стійкі вирази і фрази, а їх тексти розраховані на слухачів-носіїв мови і включають в себе сучасні розмовні обороти [35].

Інформаційний простір сучасності характеризується наявністю великої кількості аудіохостингів з приблизно однаковим функціоналом. Ці ресурси надають доступ до аудіо продукції, здебільшого у mp3 форматі. Їхнє головне призначення полягає у наданні користувачу можливості прослуховування відповідних файлів.

З точки зору вивчення іноземної мови доцільним та необхідним додатком до аудіотреку є текст відповідного пісенного твору. Така необхідність полягає у можливості перевірки правильності результатів аудіювання.

Таким чином, питання отримання тексту відповідного пісенного твору є актуальним на сучасному етапі розвитку людства.

**Мета і задачі роботи.** Мета роботи полягає у реалізації інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв з використанням загальнодоступних мережевих аудіо-хостингів.

Для досягнення поставленої мети визначені наступні задачі досліджень:

- провести аналіз існуючих методів, технологій та рішень завантаження слів пісень в реальному часі для мобільних пристроїв;
- удосконалення методики завантаження слів пісень у напрямку отримання максимально повних і достовірних даних;
- розробити інформаційну технологію завантаження слів пісень в реальному часі для мобільних пристроїв;
- виконати експериментальну перевірку інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

**Об'єкт дослідження** – процес завантаження слів пісень в реальному часі для мобільних пристроїв.

**Предмет дослідження** – моделі, методи, підходи та засоби інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

**Методи дослідження.** Для розв'язання поставлених задач використовуються основні положення методів аналізу даних, кодування та декодування аудіо файлів; для удосконалення методики завантаження слів пісень

протоколи обміну даними, організація і керування базами даних; для реалізації інформаційної технології - технології проектування інформаційних систем та об'єктно орієнтований підхід.

**Наукова новизна одержаних результатів.** В результаті проведеної роботи були отримані такі результати:

- Удосконалено існуючі методи завантаження слів пісень в реальному часі для мобільних пристроїв у напрямку отримання максимально повних і достовірних даних.

**Практичне значення одержаних результатів.** В результаті виконання дипломної роботи магістра розроблено відповідне програмне забезпечення, яке підтвердило вірність запропонованих положень. Застосування інформаційної технології дає можливість завантаження слів пісень в реальному часі для мобільних пристроїв у напрямку отримання максимально повних і достовірних даних.

**Апробація результатів дипломної роботи.** Основні наукові та практичні результати опубліковані в наукових виданнях МОН України:

- публікація на тему “Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв” в науковому журналі “Вісник Хмельницького національного університету”.

За темою дипломної роботи магістра автором виконано одну наукову публікацію [36].

**Структура та обсяг роботи.** Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 35 найменувань та 5 додатків. Загальний обсяг дипломної роботи магістра становить 72 сторінки, з них 65 сторінок основного тексту та 7 сторінок додатків. У роботі наведено 51 рисуноків та 1 таблиця.

## **Розділ 1**

### **Аналіз сучасного стану проблеми завантаження слів пісень в реальному часі для мобільних пристроїв**

#### **1.1 Аналіз предметної області**

Сучасний розвиток техніки та технологій виводить на якісно новий рівень практично всі сфери життєдіяльності людини. На заміну та допомогу традиційним підручникам приходять технічні пристрої та сучасні технології. Тобто, вивчення іноземної мови може відбуватися за допомогою смартфона або планшета, які є під рукою практично завжди.

Одним із методологічних підходів вивчення іноземних мов, який довів свою високу ефективність, є спілкування з носіями мови. Певним віртуальним заміником такого носія може слугувати пісенний виконавець. Музика дуже корисна при вивченні іноземних мов, а особливо в освоєнні граматики, збільшенні словникового запасу і постановці вимови. Крім того, відомо, що прослуховування класичної музики стимулює мозкову активність, покращує уважність, зосередженість, пам'ять. Більшість пісень містить стійкі вирази і фрази, а їх тексти розраховані на слухачів-носіїв мови і включають в себе сучасні розмовні обороти [35].

Інформаційний простір сучасності характеризується наявністю великої кількості аудіохостингів з приблизно однаковим функціоналом. Ці ресурси надають доступ до аудіо продукції, здебільшого у mp3 форматі. Їхнє головне призначення полягає у наданні користувачу можливості прослуховування відповідних файлів.

З точки зору вивчення іноземної мови доцільним та необхідним додатком до аудіотреку є текст відповідного пісенного твору. Така необхідність полягає у можливості перевірки правильності результатів аудіювання.

Таким чином, питання отримання тексту відповідного пісенного твору є актуальним на сучасному етапі розвитку людства.

## 1.2 Аналіз сучасного програмного забезпечення предметної області

### 1.2.1 Аналіз існуючих аудіохостингів

Зараз з прогресом людства та розвитком сучасних технологій індустрія запису та відтворення музики переходить в онлайн. Створюється багато сервісів для роботи з потоковим аудіо, які дають змогу легко та безкоштовно прослуховувати музичні композиції, надають послуги легального потокового стримінгу аудіо найбільших світових лейблів, таких як Sony [1], Warner Music Group [2], Universal [3] та інші. Деякі сервіси надають обширну інформацію про композиції, включаючи її слова, інформацію про виконавця і його творів.

Одним з прикладів сервісів, які дозволяють перенести сферу музичної індустрії в онлайн є сервіс “Spotify” [4] (рисунок 1.1).

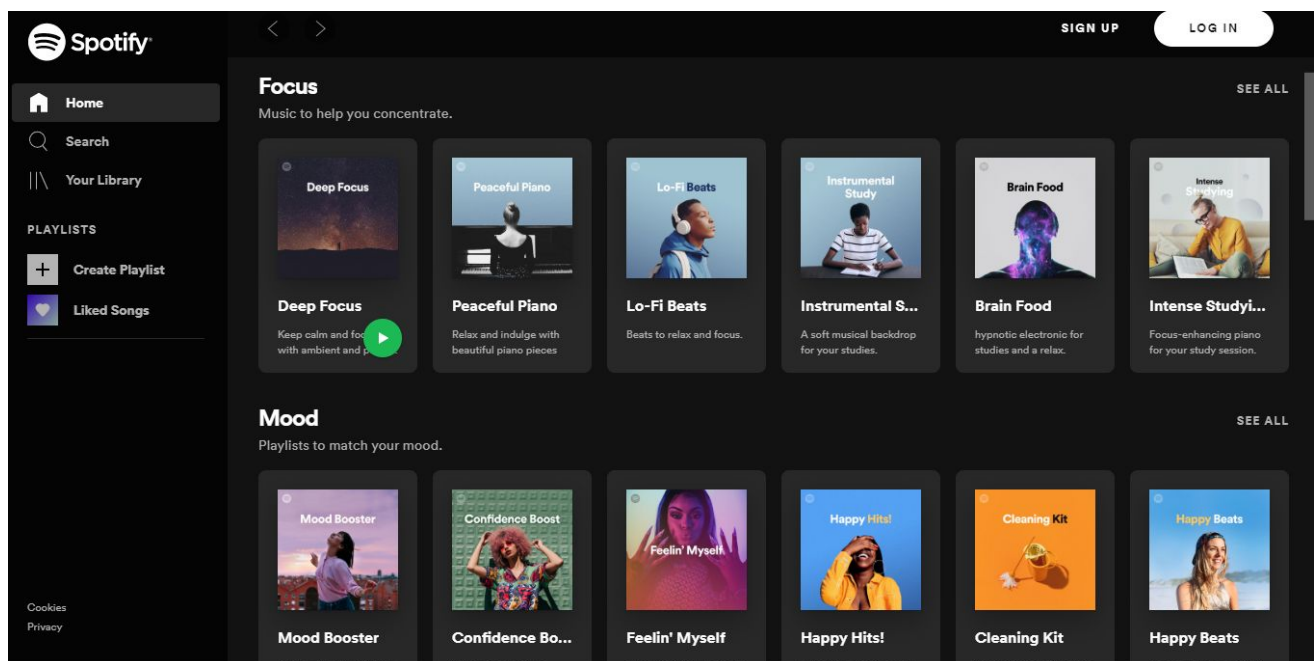


Рисунок 1.1 – Spotify - багатofункціональний сервіс прослуховування музики [4]

Головна сторінка сайту організована в стилі landing-page, що одразу акцентує увагу користувача на головних перевагах різних видів підписок. Можна перейти безпосередньо до сторінки з композиціями (див. рис 1.1). Також на сайті

є досить зручний і інтерактивний пошук композиції за різними фільтрами [5] (рис 1.2). Зручним є те, що з головної сторінки можна перейти на сторінку музичного плеєра [5] (рис 1.3). Одним з недоліків даного сервісу, є те, що він є поки недоступним в Україні (доступ можна отримати через послуги VPN), а також те, що сервіс є платним.

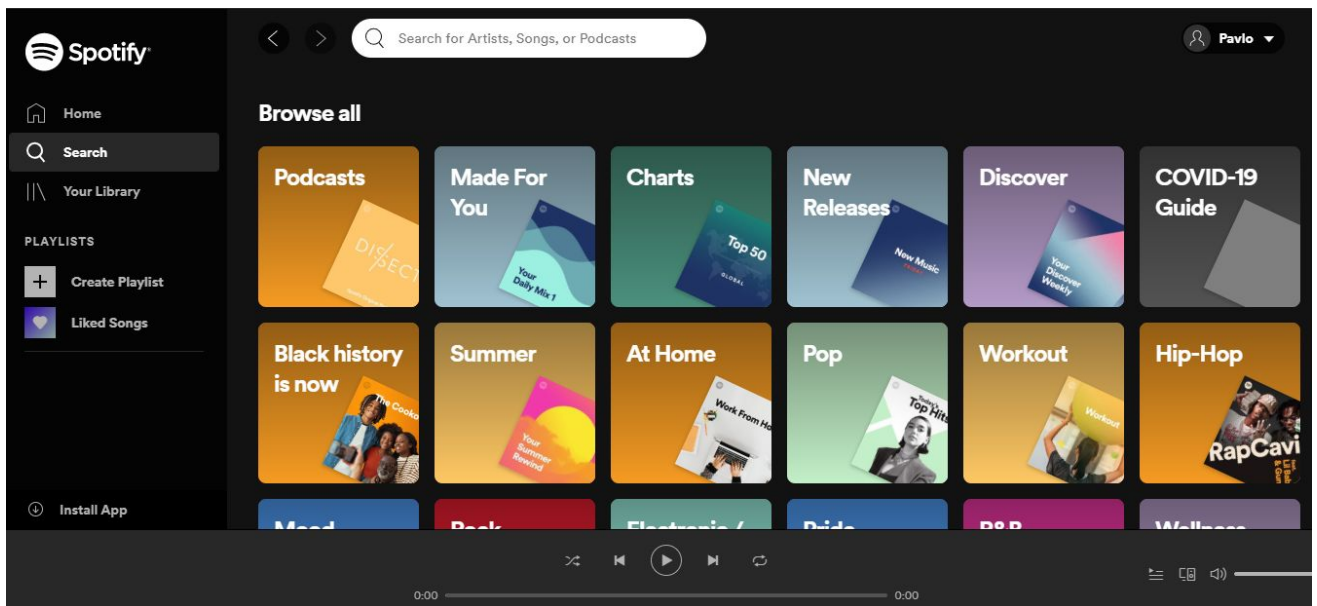


Рисунок 1.2 – Сторінка пошуку композицій [5]

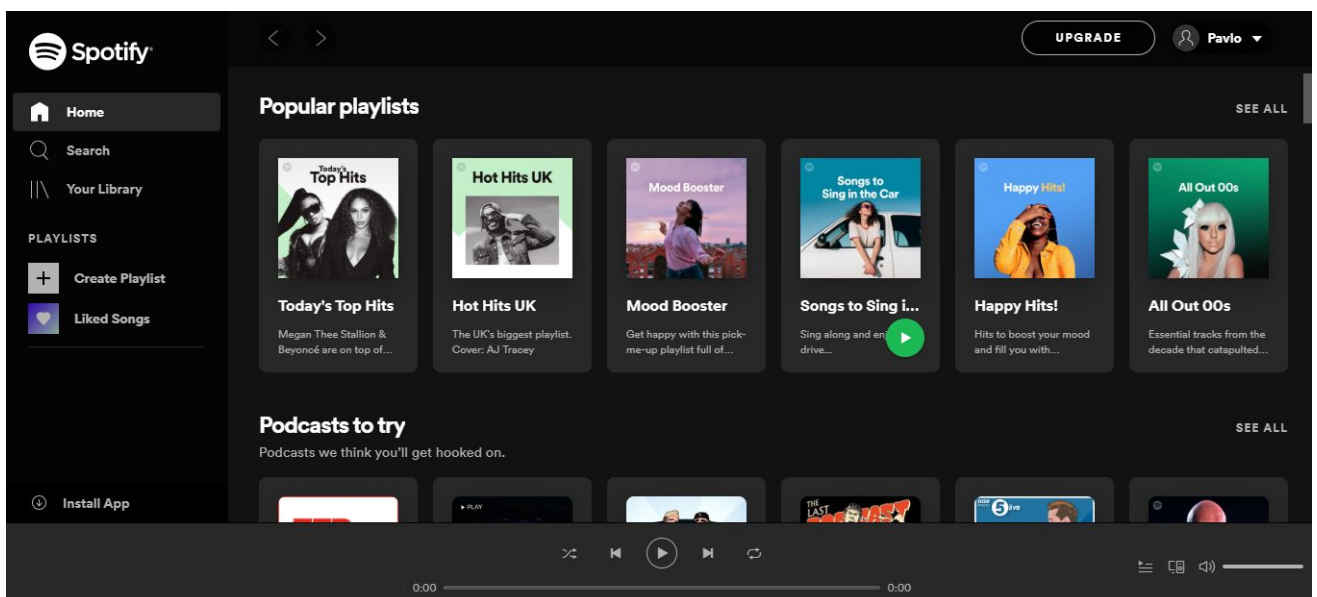
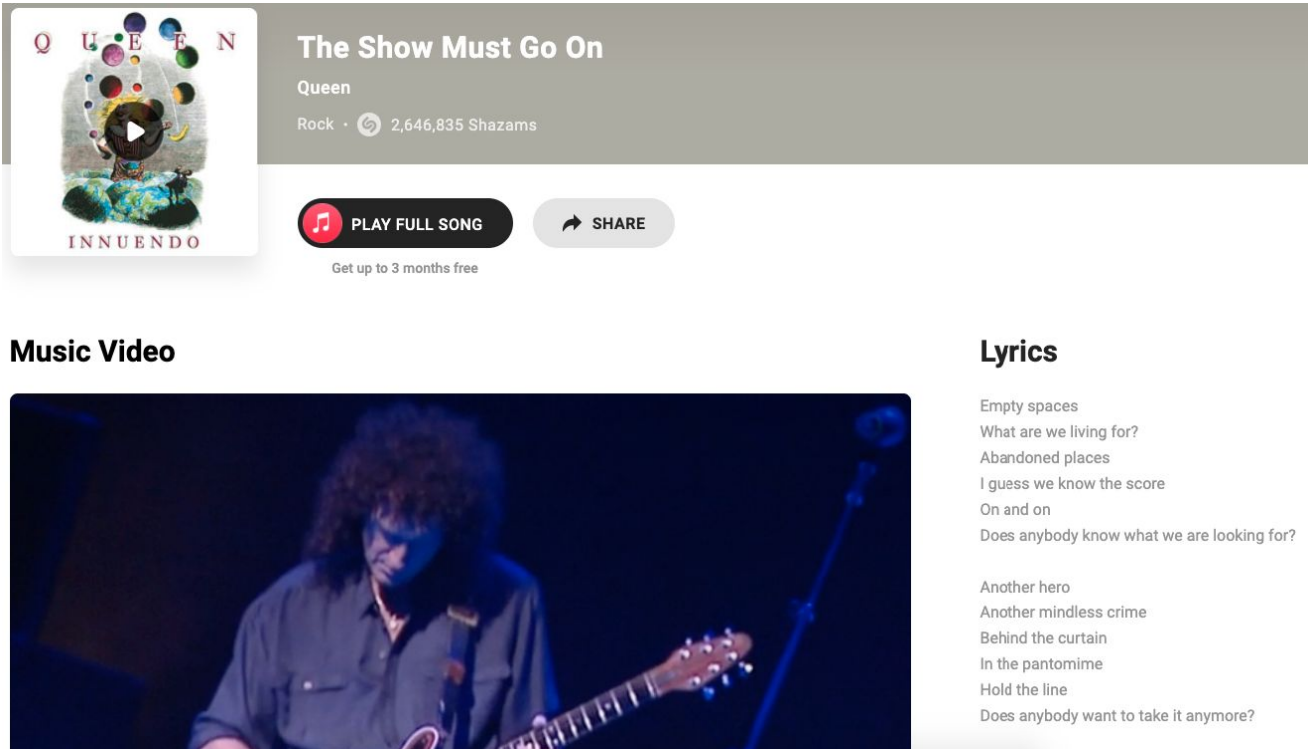


Рисунок 1.3 – Сторінка музичного плеєра [5]

Shazam [6] (Рис. 1.4) - популярний сервіс для пошуку музичних композицій. Користується популярністю за функцію пошуку пісень, прослуховуючи саму пісню. Тобто, користувач слухає пісню та хоче знайти її виконавця або слова. Він запускає Shazam та “дає послухати” пісню самому сервісу. Як результат, сервіс показує хто виконавець, слова, рейтинг та має зв'язок з популярним відеохостингом Youtube, для того, щоб швидко переглянути відеокліп пісні.



**The Show Must Go On**  
Queen  
Rock · 2,646,835 Shazams

PLAY FULL SONG  
SHARE

Get up to 3 months free

**Music Video**

**Lyrics**

Empty spaces  
What are we living for?  
Abandoned places  
I guess we know the score  
On and on  
Does anybody know what we are looking for?

Another hero  
Another mindless crime  
Behind the curtain  
In the pantomime  
Hold the line  
Does anybody want to take it anymore?

Рисунок 1.4 - Shazam - популярний сервіс пошуку музики [6]

## 1.2.2 Аналіз існуючих сервісів пошуку музики та слів для мобільних пристроїв

Прикладом сервісу для пошуку слів в реальному часі для мобільних пристроїв є сервіс “Musixmatch” [7]. Мобільні програми “Musixmatch” відображають тексти пісень, синхронізовані з музикою, що відтворюється (рис 1.5). У своїх власних програмах він може сканувати всі пісні в музичній

бібліотеці користувача, знаходити тексти пісень та використовувати їх як музичний плеєр. На Android він також підтримує сервіси потокового передавання музики, такі як Spotify [4], Google Play Music [8], Deezer [9]. MusixMatch також дозволяє користувачам вносити тексти пісень та синхронізації до слів в обмін на бали, які показують рівень та можна обміняти на значки.

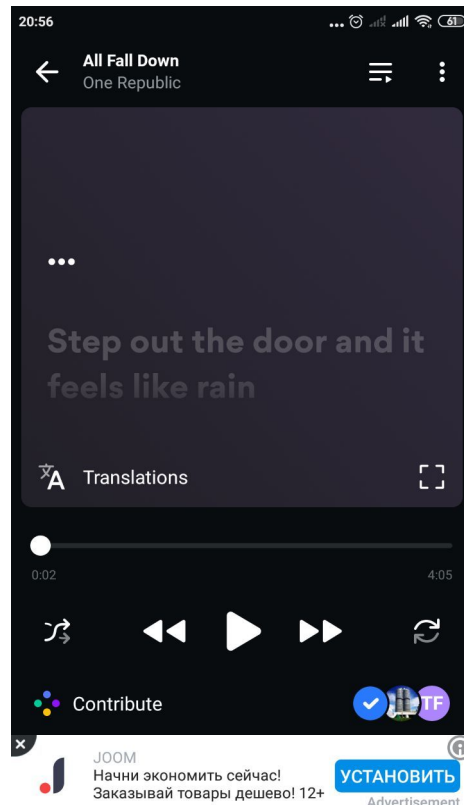


Рисунок 1.5 – Сторінка з аудіоплеєром Musixmatch [8]

Основна функція є надання синхронізованих слів пісні користувачу.

Shazam (Android app) [10] - мобільний додаток для пошуку музичної композиції, слів та її виконавця за самою композицією. Розширює функціонал веб версії більш зручним пошуком (рис 1.6).

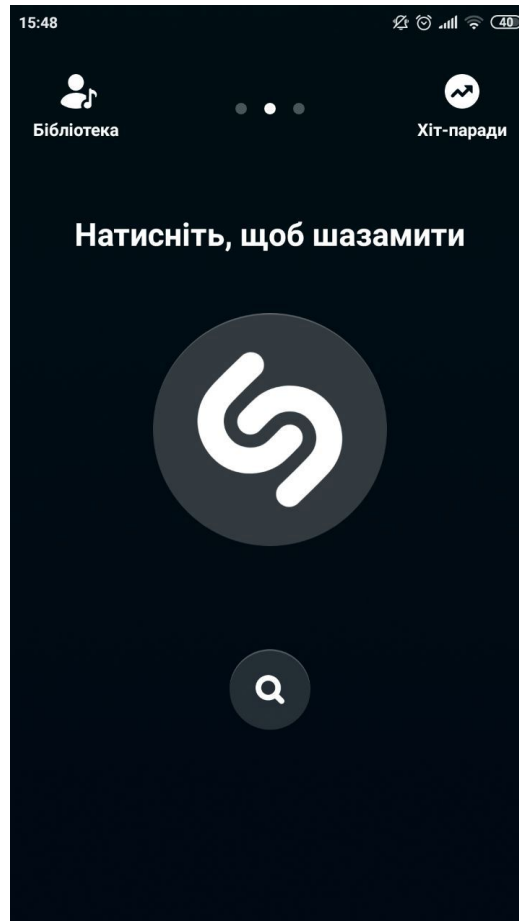


Рисунок 1.6 - Shazam - мобільний додаток [10]

Також користувачі мають можливість переглянути недавні пошуки, згенеровані плейлисти, на основі недавніх пошуків та виконавців (рис. 1.7). Виконавців можна відсортувати за різними параметрами для зручності пошуку. До того ж, з цього екрану можна зайти на екран з налаштуваннями.

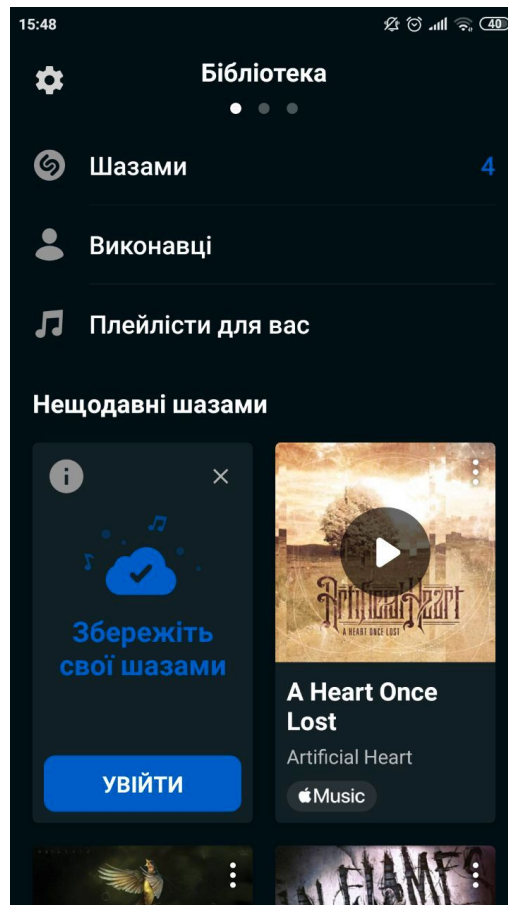


Рисунок 1.7 - Бібліотека Shazam

Ще одна з популярних сторінок у програмі - "Хіт-парад" (рис. 1.8). Сервіс надає користувачу список останніх найбільш популярних виконавців та їх композицій. Можна переглядати хіт-парад для окремої країни, міста чи загалом світу. Можна також поділитися популярним треком із друзями у соціальних мережах або послуг передачі інформації, таких як Bluetooth [11], Android Beam [12] та ін.

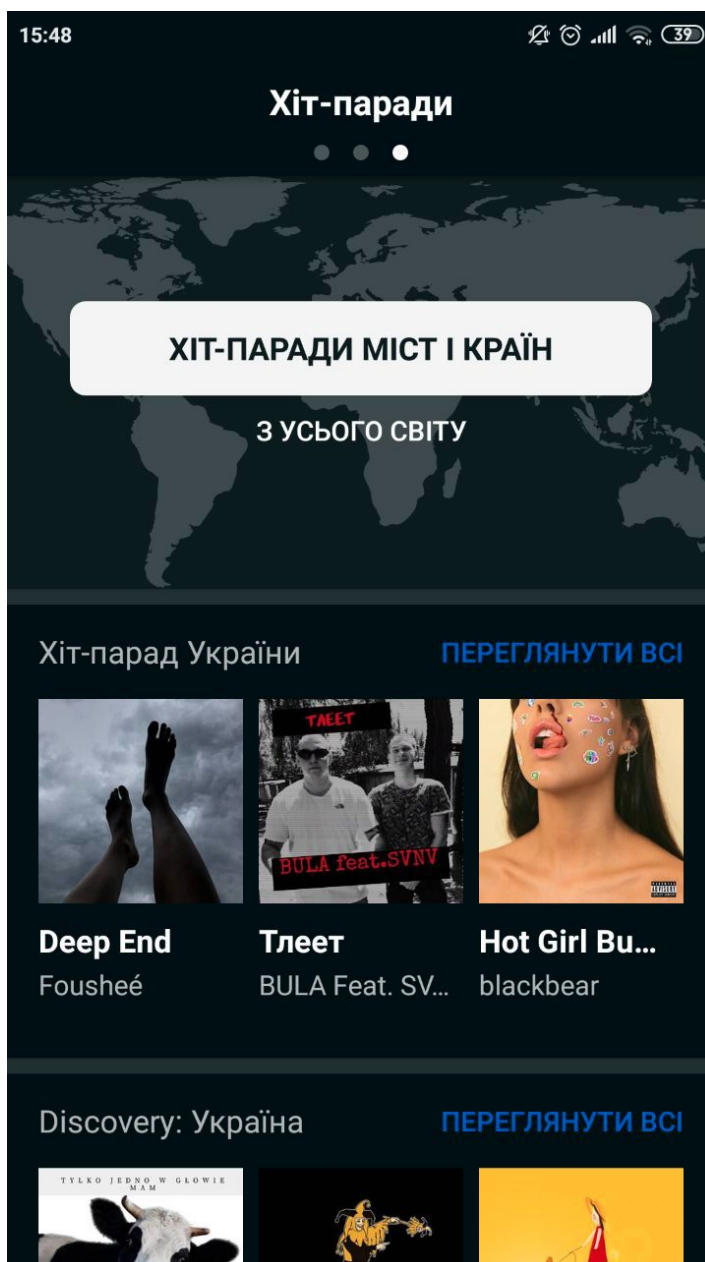


Рисунок 1.8 - сторінка “Хіт-парад”

### 1.3 Огляд засобів реалізації

Для реалізації програмних застосунків для мобільних пристроїв найбільш ефективним є використання мови програмування Kotlin [23], локальної бази даних Room [24], Koin Framework [28] як інструмент внесення залежностей, середовище розробки застосунків для ОС Android Android Studio [26] за допомогою зручного набору інструментів Android SDK [27].

### 1.4 Висновки до розділу та постановка задачі

Проведений аналіз літературних джерел засвідчує суттєву та зростаючу потребу у завантаженні слів пісень на мобільні пристрої. Це обумовлено загальною тенденцією на глобалізацію, що визначає нагальну необхідність у знанні іноземних мов, а розвиток інформаційних технологій та відповідних технічних засобів формує потребу у відповідних мобільних додатках.

В результаті проведеного аналізу існуючих підходів сформульовані такі завдання дослідження метою якого є реалізація інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв з використанням загальнодоступних мережевих аудіо-хостингів.

Для досягнення поставленої мети визначені наступні задачі досліджень:

- провести аналіз існуючих методів, технологій та рішень завантаження слів пісень в реальному часі для мобільних пристроїв;
- удосконалення методики завантаження слів пісень у напрямку отримання максимально повних і достовірних даних;
- розробити інформаційну технологію завантаження слів пісень в реальному часі для мобільних пристроїв;
- виконати експериментальну перевірку інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

## Розділ 2

### Розробка інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв

#### 2.1 Основні етапи і процедури

Починаючи з версії Android 6.0, при першому запуску програми система може вимагати від користувача дозвіл на отримання різного роду інформації. Дозволи можуть бути на різних рівнях системи: керування внутрішніми функціями системи, обробка та збір інформації про систему, обробка та збір інформації про користувача. Такі дозволи зазвичай використовуються в межах програми та можуть бути відкликані користувачем у будь-який час. Приклад такого runtime-дозволу можна побачити на рисунку 2.1.

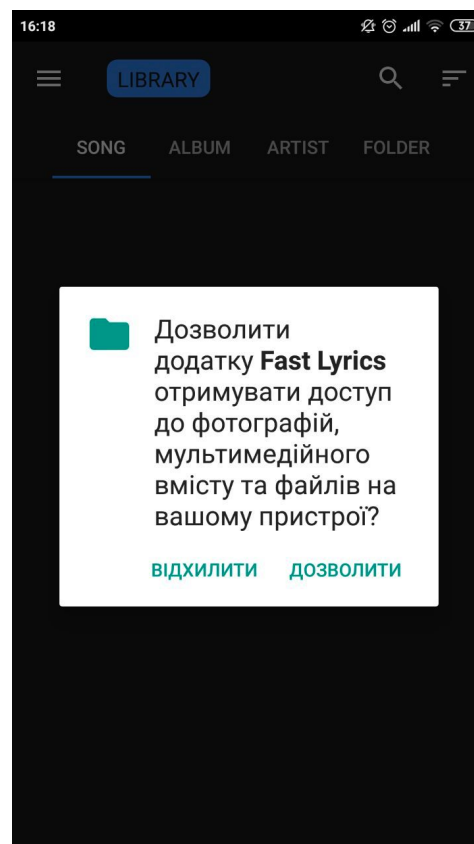


Рисунок 2.1 - Дозвіл на отримання файлів з пристрою

Для того, щоб зчитати інформацію з файлу, необхідно визначити метадані композиції. Метадата може складатися з таких параметрів (рис. 2.2):

- Назва пісні;
- виконавець;
- альбом;
- номер пісні в альбомі;
- номер диску;
- рік;
- жанр;
- рейтинг.

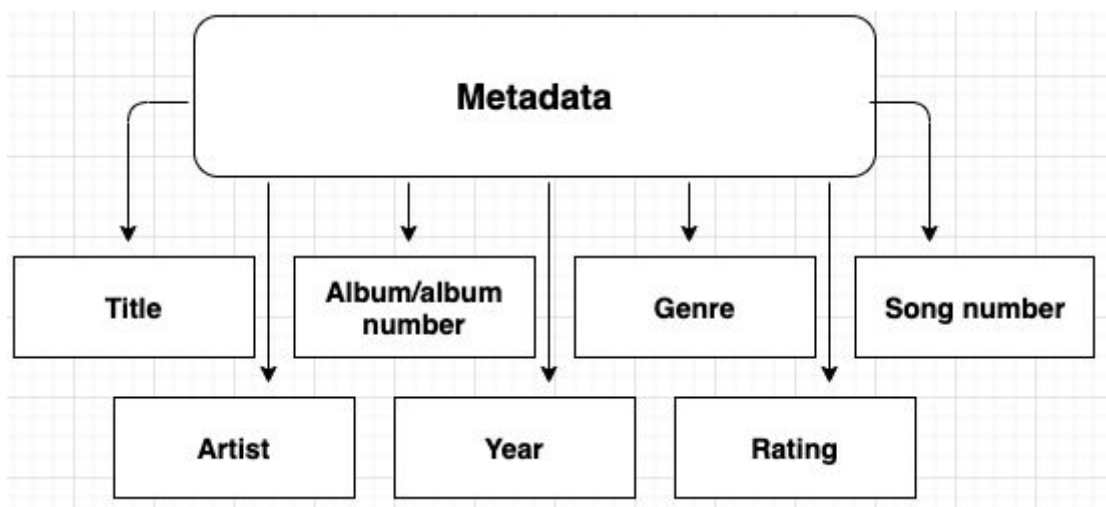


Рисунок 2.2 - Параметри метадати

Зазвичай метадата заповнюється на етапі звукозапису у студії, але часто буває, що пісня випускається з стислими параметрами метадати.

Для розробки інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв достатньо небагатьох параметрів метадати. Такі параметри зображені на рисунку 2.3.

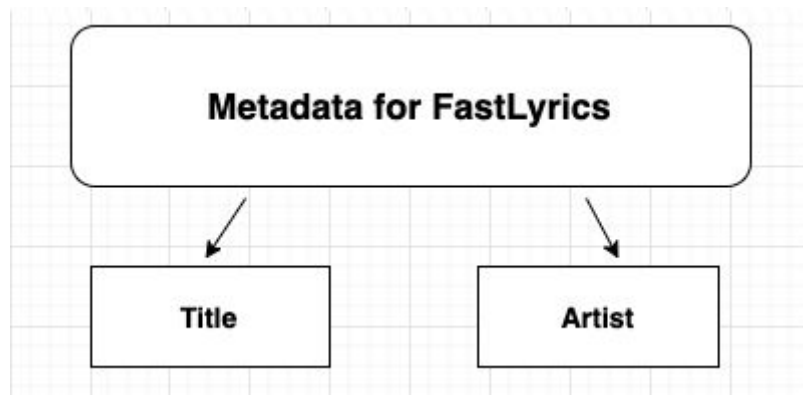


Рисунок 2.3 - Метадані для FastLyrics додатку

Для завантаження слів достатньо знати назву виконавця та назву композиції.

## 2.2 Опис сервісів для пошуку медіаданих

За основу, як відкриті джерела пошуку музики, були взяті такі сервіси:

- genius API для пошуку пісень;
- iTunes API для ручного пошуку пісень;
- lastFM API для пошуку інформації про виконавця.

Кожен раз при відправленні запиту, сервіс потребує спеціальний токен доступу, який можна отримати, зареєструвавшись у кожному з вище названих сервісах, та зайшовши у кабінет користувача.

Далі, отримавши токен, потрібно підставити його у запит. Загальна структура запиту:

`<base url> + <service params> + <query params>`, де

`base url` - базовий URL сервісу;

`service params` - це токен або ключ api;

`query params` - це додаткові параметри запиту, наприклад, назва виконавця.

Як архітектурний стиль обробки і передачу стану ресурсів по HTTP протоколу можна вибрати REST. Взаємодію між програмою, базою даних та REST сервісом можна побачити на рисунку 2.4.

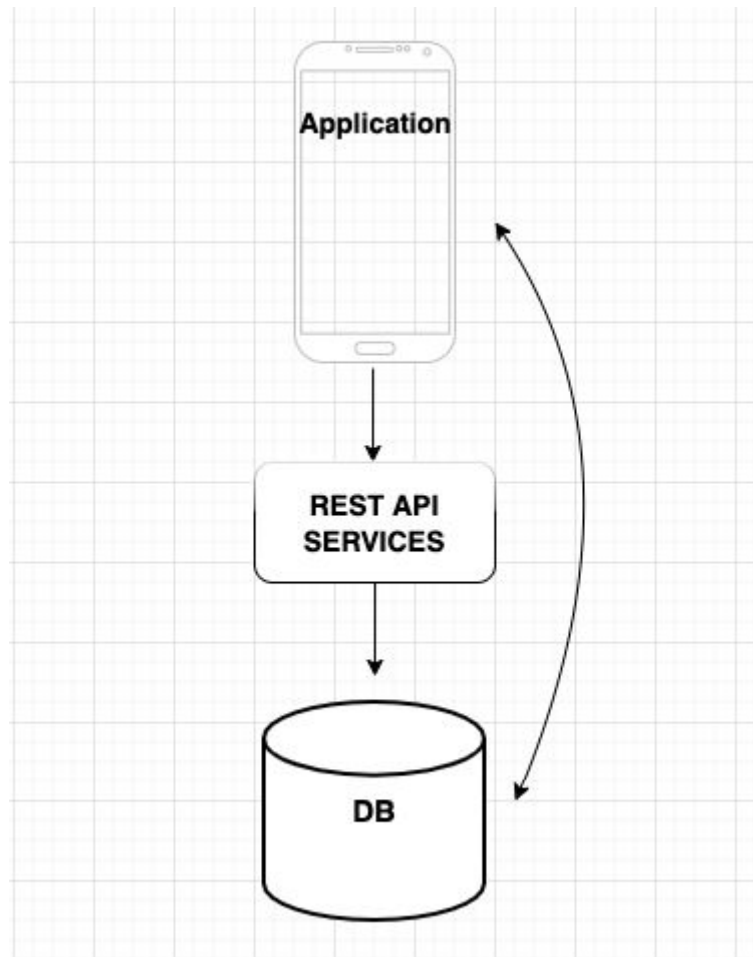


Рисунок 2.4. - Взаємодія між програмою, базою даних та REST сервісом.

Програма може напряму звернутися до бази даних за інформацією, але записати інформацію в базу даних може тільки через REST сервіс, тобто, спочатку отримати дані з віддалених сервісів і тільки потім записати їх у локальну базу даних.

Для зв'язування стану отриманих даних і об'єктів з життєвим циклом активності можна використати інструмент LiveData. Такі дані зазвичай поміщаються в ViewModel клас. Такий механізм підходить, якщо потрібно зберегти дані і відновити їх через деякий час, наприклад, після зміни конфігурації.

Для того, щоб було легше міняти залежності, можна використати процес Dependency Injection. Це дасть можливість легко замінити модуль, який

використовується в програмі. Узагальнену схему роботи DI можна побачити на рисунку 2.5.

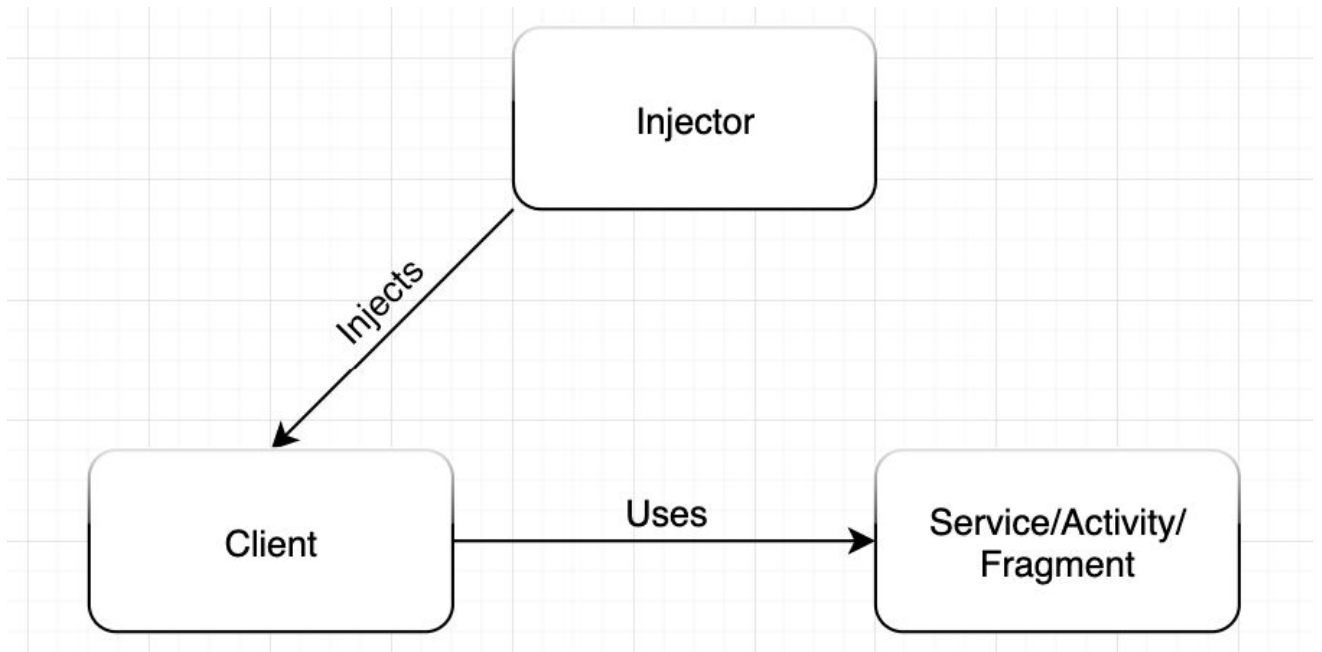


Рисунок 2.5 - Узагальнена схема DI

Також для надійної роботи програми потрібно використати механізм Shared Preferences. Він представляє собою сховище key-value і зберігається у внутрішній пам'яті мобільного пристрою. Загальну схему можна побачити на рисунку 2.6.

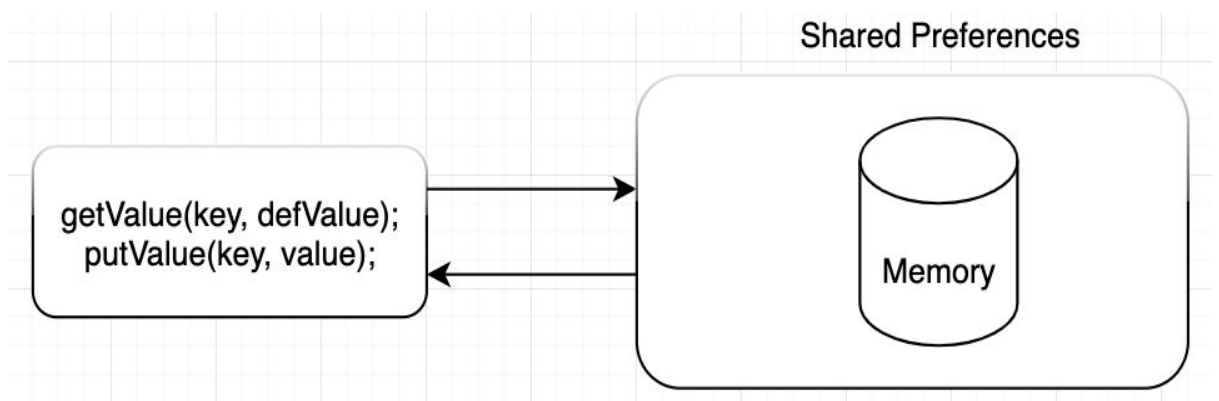


Рисунок 2.6 - Представлення механізму Shared Preferences

### **2.3 Висновки до розділу**

Було визначено структуру інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв. Описано процедури забезпечення функціональності інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв, потоки даних та інформації між модулями системи з представленням інформаційних діаграм.

## Розділ 3

### Проектування структури інформаційної технології

#### 3.1 Аналіз та автоматизація обробки інформаційних потоків

Сьогодні розвиток інформаційних технологій відіграє важливу роль для бізнесу та людства в цілому. Різні технології взаємодіють між собою утворюючи одну систему, яка підтримує бізнес, формуючи спосіб життя в цілому.

Для будь-якої інформаційної технології важливими є бізнес-процеси, які складають основний бізнес тої чи іншої компанії, забезпечують її основними потоками доходів. Для автоматизації деяких процесів і зберігання даних використовуються бази даних (табл. 3.1).

Таблиця 3.1 - Представлення бази даних

Сутність	Значення
AlbumStat	Представлення статистики альбому
ArtistStat	Представлення статистики виконавця
Bio	Представлення біографії та тегів
EQPreset	Представлення еквайзера
Favorite	Представлення списку улюблених
Lyric	Представлення слів
SongStat	Представлення статистики композиції

#### *Бізнес-процес «Завантаження слів музичної композиції в реальному часі»*

Даний бізнес-процес передбачає опрацювання інформації, що стосується завантаження слів, й включає в себе наступні функції:

- збір інформації по композиції, що відтворюється;
- пошук слів у віддалених базах даних;

- отримання слів;
- збереження слів у локальній базі даних.

*Бізнес-процес «видалення завантажених слів композицій»*

Даний бізнес-процес передбачає видалення вибраного файлу з внутрішнього сховища мобільного пристрою назавжди й включає в себе наступні функції.

*Бізнес-процес «медіа-плеєр»*

Даний бізнес-процес передбачає функціональну частину інтерфейсу користувача й включає в себе наступні функції:

- початок відтворення;
- пауза;
- вперед на 1 композицію;
- назад на 1 композицію;
- показ завантаження слів (показ вибору слів, якщо доступно);

*Бізнес-процес «інформація про виконавця»*

Даний бізнес-процес передбачає показ користувачу детальну інформацію про виконавця, список тегів та альбоми з композиціями, доступних локально на мобільному пристрої.

*Бізнес-процес «пошук композицій»*

Даний бізнес-процес передбачає пошук композиції по списку знайдених та відфільтрованих композицій у локальному сховищі мобільного пристрою.

*Бізнес-процес «сортування»*

Даний бізнес-процес передбачає набір критерій сортування, таких як:

- сортування за назвою композиції;

- сортування за номером композиції;
- сортування за тривалістю;
- сортування за датою додавання у сховище мобільного пристрою;
- сортування за роком;
- сортування за назвою виконавця;
- сортування за спаданням/зростанням;

#### *Бізнес-процес “улюблені”*

Даний бізнес-процес передбачає додавання композиції в список улюблених і включає наступні функції:

- відтворення списку улюблених;
- додавання в чергу для відтворення;
- видалення зі списку улюблених;

#### *Бізнес-процес “еквалайзер”*

Даний бізнес-процес передбачає використання пресетів еквалайзера для забезпечення комфортного звучання композицій та включає наступні функції:

- увімкнення, вимкнення;
- вибір пресету.

#### *Бізнес-процес “шаринг композиції”*

Даний бізнес-процес передбачає використання внутрішніх процесів -системи Android для відправлення композиції за межі додатку наступним чином:

- пошта;
- bluetooth;
- месенджери;
- відправка у віддалене сховище (якщо доступно)

### *Бізнес-процес “завантаження усіх доступних слів”*

Даний бізнес-процес передбачає сканування музичних композицій та завантаження їх слів в локальну базу даних для перегляду оффлайн.

Для того, щоб автоматизувати роботу застосунку, дані потрібно організувати у таблиці, створити базу даних і провести нормалізацію БД.

Коротке представлення бази даних можна побачити на рисунку 3.1. Даталогічна модель БД створюється відповідно до бізнес-процесів та їх функцій (додаток А).

Table Name	Column Name	Data Type
album_stats	albumid	INTEGER
	playcount	INTEGER
	playtime	INTEGER
android_metadata	locale	TEXT
artist_stats	artistid	INTEGER
	playcount	INTEGER
	playtime	INTEGER
bio	artistid	INTEGER
	bio	TEXT
	tags	TEXT
equalizer	eqid	INTEGER
	eqname	TEXT
favorites	fav	INTEGER
lyrics	songid	INTEGER
	lyric	TEXT
room_master_table		
song_stats	songid	INTEGER
	playcount	INTEGER
	playtime	INTEGER
sqlite_sequence	name	
	seq	
Indices (0)		
Views (0)		
Triggers (0)		

Рисунок 3.1 – Коротке представлення бази даних

База даних «fl.db» вміщує в собі таблиці:

1. album\_stats:
  - albumid - ідентифікатор альбому (integer);
  - playcount - к-сть відтворень альбому (integer);
  - playtime - загальний час відтворення альбому (integer).
2. artist\_stats:
  - artistid - ідентифікатор виконавця (integer);
  - playcount - к-сть відтворень виконавця (integer);
  - playtime - загальний час відтворення виконавця (integer).
3. android\_metadata:
  - locale - локаль додатку (text).
4. bio:
  - artistid - ідентифікатор виконавця (integer);
  - bio - текст біографії (text);
  - tags - пов'язані з виконавцем теги (text).
5. equalizer:
  - eqid - ідентифікатор пресету (integer);
  - eqname - назва пресету (text).
6. favorites:
  - fav - список улюблених композицій (integer).
7. lyrics:
  - songid - ідентифікатор композиції (integer);
  - lyric - слова композиції (text).
8. song\_stats:
  - songid - ідентифікатор композиції(integer);
  - playcount - к-сть відтворень композиції (integer);
  - playtime - загальний час відтворення композиції (integer).

### 3.2 Розробка структури інформаційної технології

Розроблений застосунок буде корисним у використанні для людей які, часто слухають музику та бажають переглядати слова композиції.

Користувач буде мати змогу завантажити слова композиції та вирішити чи зберегти її локально для того, щоб мати змогу відкрити слова знову без

підключення до мережі. Схема доступних функцій для користувача зображено на рисунку 3.2.

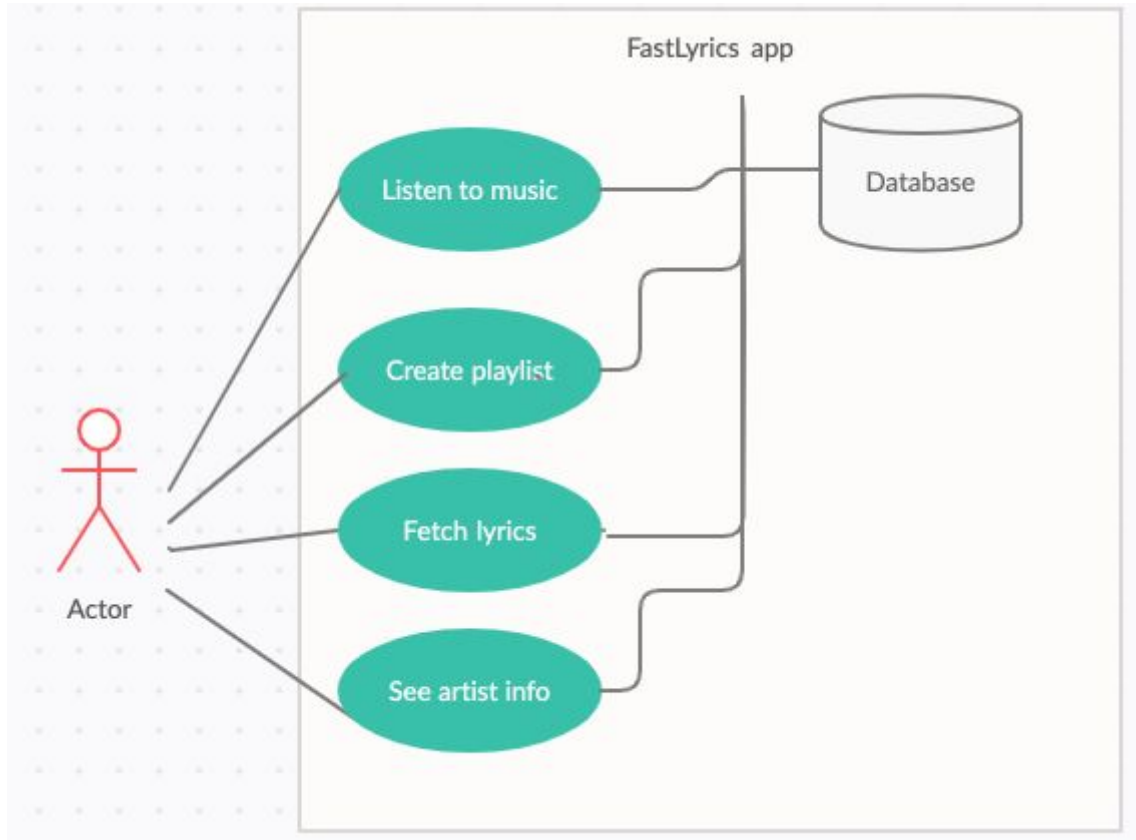


Рисунок 3.2 – Роль користувача

Загальну схему застосунку представлено на рисунку 3.3.

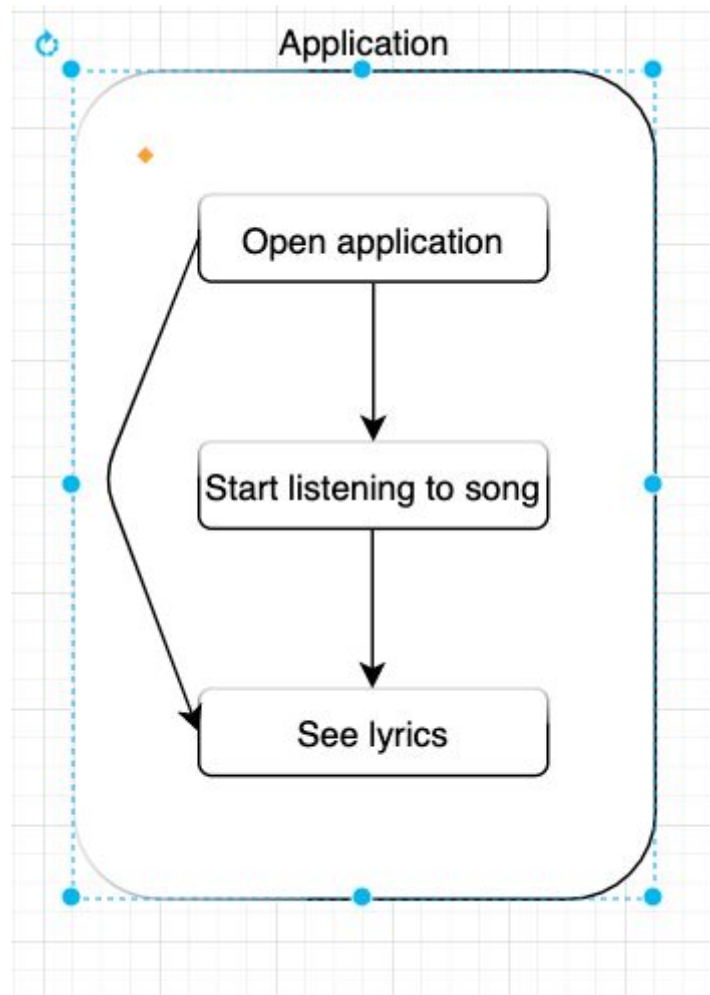


Рисунок 3.3 – Схема користування застосунком

З рисунку 3.3 можна зробити висновок, що користувач має можливість не тільки переглядати слова, слухаючи пісню, але й без прослуховування, відкривши застосунок і завантаживши слова.

### 3.3 Вибір засобів розробки інформаційної технології

Kotlin - статично типізована мова програмування, що працює поверх JVM. Kotlin - мова програмування, яка була вибрана як основна для розробки інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

Room - це бібліотека ORM (Object Relational Mapping). Іншими словами, Room буде відображати об'єкти нашої бази даних до об'єктів Java. Room забезпечує абстракційний рівень над SQLite, щоб забезпечити вільний доступ до бази даних, одночасно використовуючи всю потужність SQLite.

Основні можливості Room:

- Перевірка SQL-запитів під час компіляції. Кожен `@Query` та `@Entity` перевіряється під час компіляції, що захищає вашу програму від збоїв під час виконання, і не тільки перевіряє єдиний синтаксис, але й відсутність таблиці;
- забезпечує генерування шаблонного коду;
- легко інтегрується з іншими архітектурними компонентами (наприклад, LiveData).

Компоненти Room Database представлені на рисунку 3.4.

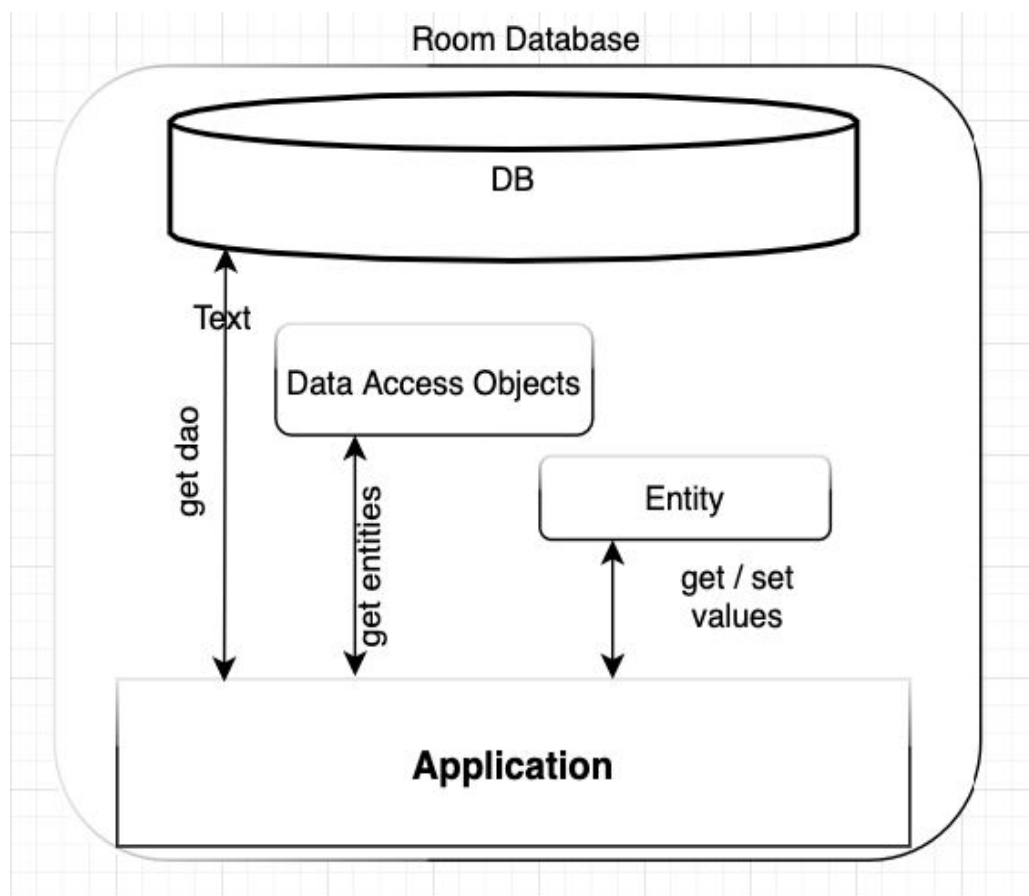


Рисунок 3.4 - Компоненти Room Database

Room має три основні компоненти RoomDB:

- Entity;
- dao;
- database.

Entity представляє таблицю в базі даних. Room створює таблицю для кожного класу, яка має анотацію @Entity, поля в класі відповідають стовпцям таблиці. Тому класи сутності, як правило, є невеликими модельними класами, які не містять жодної логіки.

DAO відповідають за визначення методів доступу до бази даних. У початковому SQLite ми використовуємо об'єкти Cursor. З Room нам не потрібен весь код, пов'язаний з курсором, і ми можемо просто визначити наші запити, використовуючи анотації в класі Dao.

Database містить базу даних і служить основною точкою доступу для базового підключення до постійних реляційних даних вашого додатка.

Для створення бази даних необхідно визначити абстрактний клас, який розширює RoomDatabase. Цей клас анотується @Database, перелічує сутності, що містяться в базі даних, і DAO, які мають до них доступ.

Клас, котрий анотується як @Database, повинен відповідати наступним умовам:

- бути абстрактним класом, який розширює RoomDatabase;
- включати список об'єктів, пов'язаних із базою даних;
- містити абстрактний метод, який має 0 аргументів, і повертає клас, котрий анотується як @Dao.

Room було обрано як основне локальне сховище інформації.

Android Studio - інтегроване середовище розробки для платформи Android, було обрано як основний інструмент розробки.

Android SDK - універсальний засіб розробки мобільних додатків для операційної системи Android. Включає в себе всі основні нативні функції та надає зручне API для розробки ПЗ для ОС Android.

Koin - фреймворк для внесення залежностей.

### **3.4 Структура і функціональне призначення модулів системи, їх взаємозв'язок**

Реалізований застосунок має свою структуру, яка відповідає функціям, які має виконувати сервіс для завантаження слів композицій.

Програмний проект були поділено за пакети за призначенням:

- data (рис. 3.5);
- di (рис 3.6);
- extensions (рис 3.7);
- interfaces (рис 3.8);
- recyclerview (рис 3.9);
- service (рис 3.10);
- ui (рис 3.11);
- utils (рис 3.12);
- views (рис 3.13).

Клас App представляє головний клас, з якого починається зв'язування Dependency Injection та ініціалізація змінної instance - контексту застосунку.

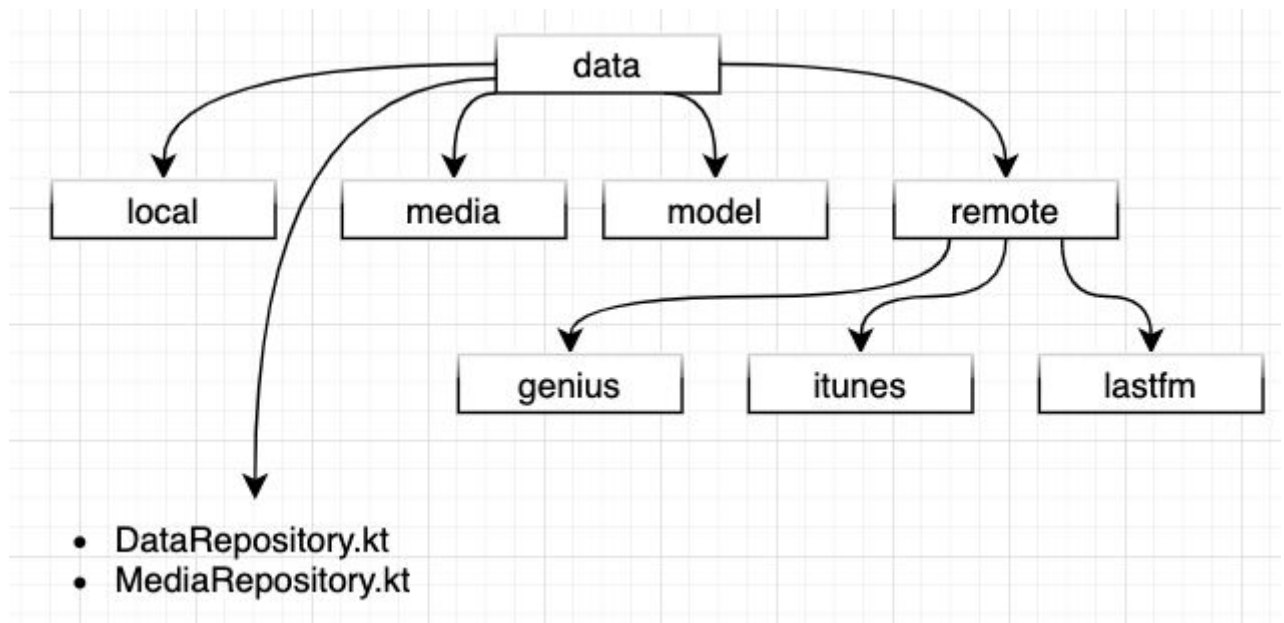


Рисунок 3.5 - Пакет data

Пакет data включає в себе класи, які призначені для опису даних, які використовуються в застосунку. Також в цьому пакеті описані основні методи для роботи з базою даних (інтерфейс DataDao.kt та клас LocalDataSource.kt). У класі LocalDatabase.kt описано процес створення бази даних.

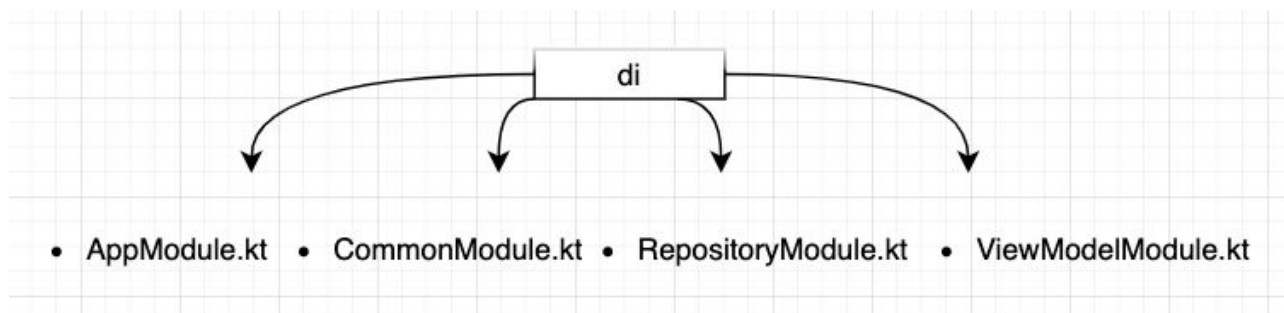


Рисунок 3.6 - Пакет di

Пакет di включає в себе програмні модулі для процесу dependency injection. Можна сказати, що це основний “будівельник” застосунку. Наприклад, в модулі CommonModule.kt можна легко додати новий або видалити існуючий фрагмент чи активність.

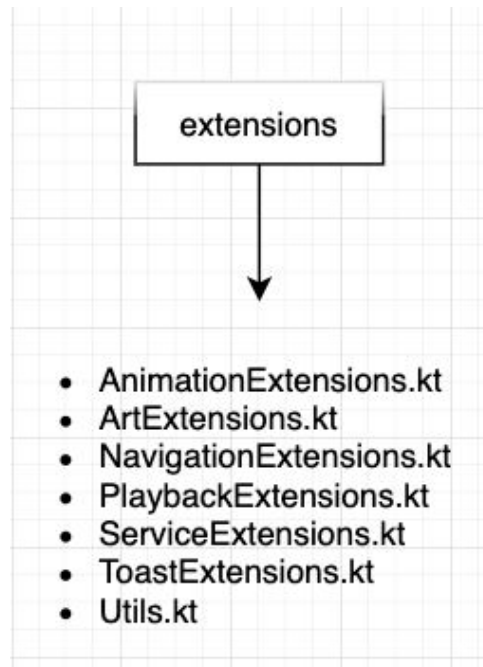


Рисунок 3.7 - Пакет extensions

Пакет extensions включає в себе набір функцій для роботи з анімаціями, зображеннями виконавців, чергами відтворення треків, спливаючих сповіщень, перетворення текстів.

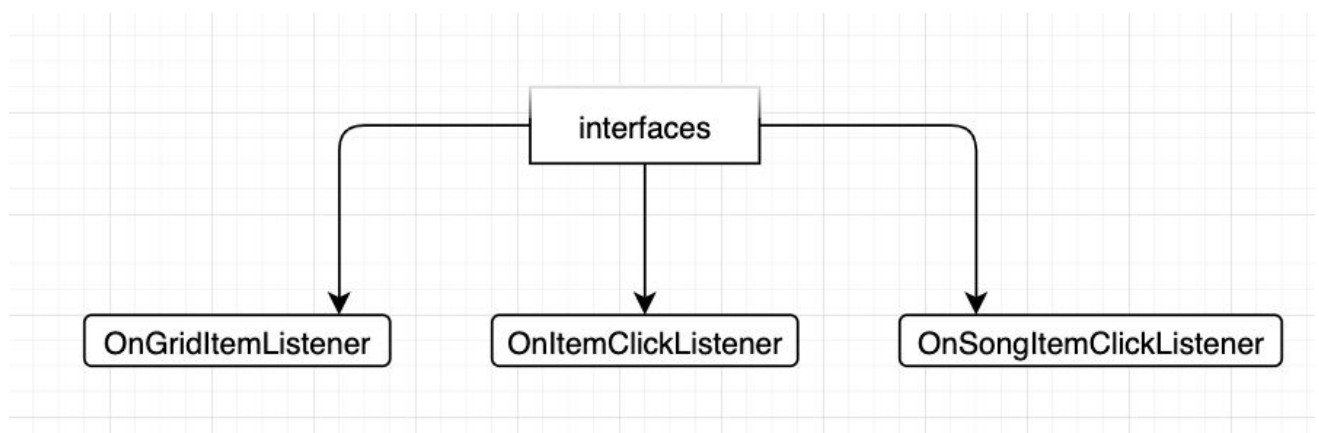


Рисунок 3.8 - Пакет interfaces

Пакет interfaces включає в себе набір методів у відповідних інтерфейсах для роботи зі списками та таблицями. У випадку з FastLyrics, наприклад,

списком може бути список виконавців, а таблицею - список альбомів, який представлено у вигляді таблиці.

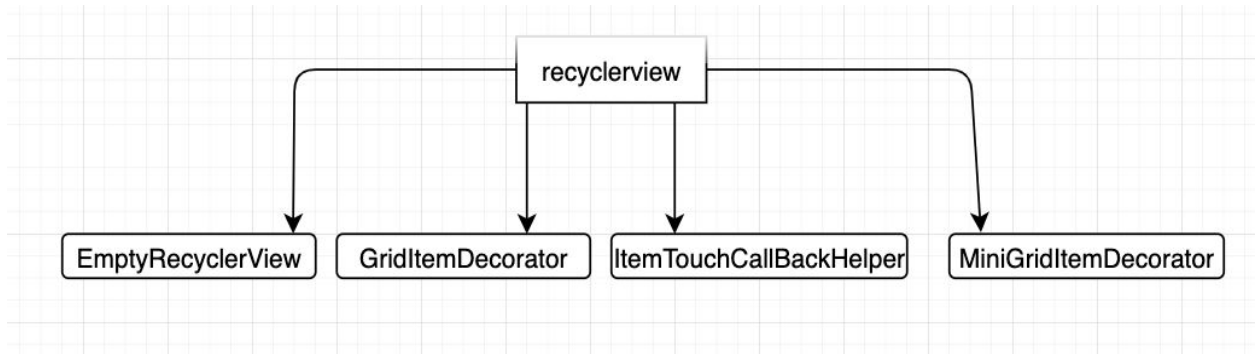


Рисунок 3.9 - Пакет recyclerview

Пакет recyclerview включає в себе класи, кожен з яких розширює клас RecyclerView. За допомогою цього класу можна легко створювати організувати великі сети інформації на екрані у вигляді списків, таблиць, динамічних списків, груп і т.д.

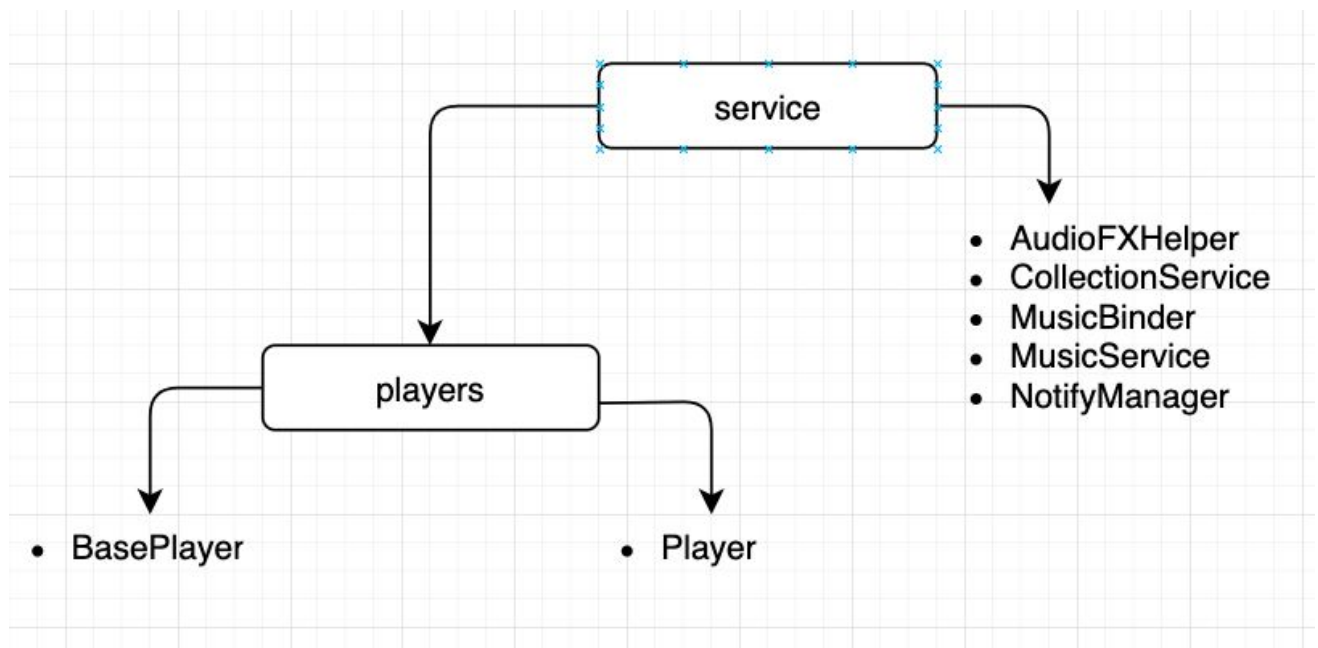


Рисунок 3.10 - Пакет service

Пакет `service` включає в себе ще один пакет `players`, який містить абстрактний клас `BasePlayer`. `BasePlayer` містить в собі деякі основні поля та методи для коректної роботи музичного плеєра. Клас `Player` описує та реалізує методи базового класу `BasePlayer` та ще деякі з основного функціонального класу музичного плеєра `MediaPlayer`. Також у пакет `service` включено класи для опису ефектів еквайзера, сповіщень в рядку стану системи, синхронізованого завантаження слів.

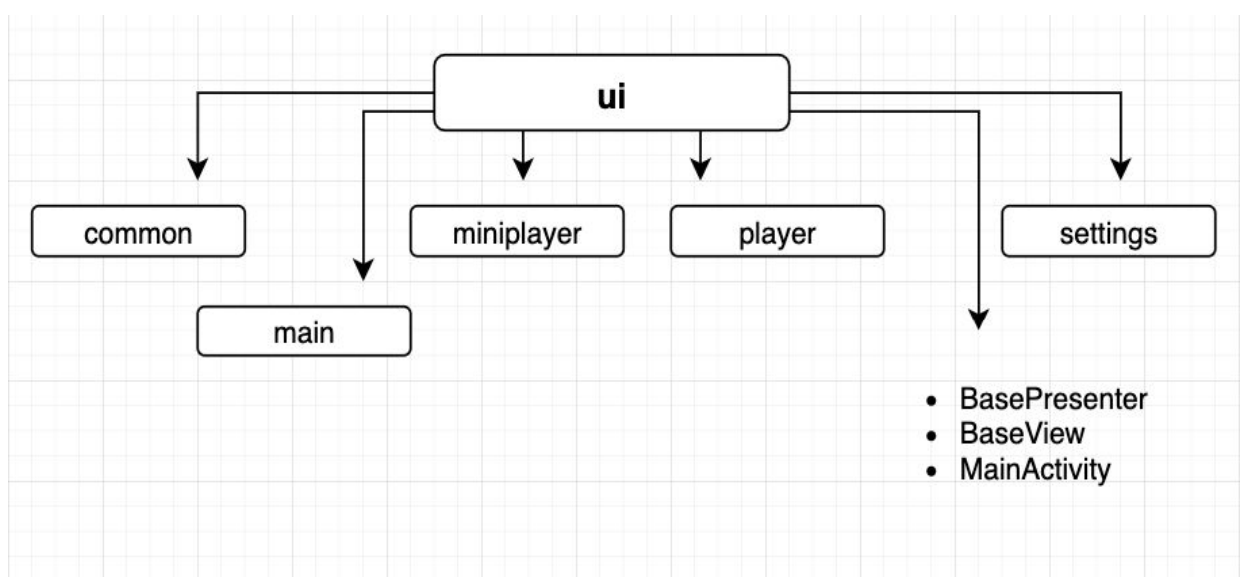


Рисунок 3.11 - Пакет `ui`

Пакет `ui` - основний пакет, в якому описано інтерфейс користувача та його поведінку. За основу відображення користувацького інтерфейсу було взято механізм `Fragment`. Кожен з фрагментів повинен розширяти базовий клас `Fragment` та бути “учасником” життєвого циклу. Також, в пакет `ui` включено клас `MainActivity`, що являє собою вхідну точку програми та відкривається в першу чергу, коли відкривається програма. Має в собі механізми зв'язку з іншими частинами програми, наприклад, медіа плеєрами.

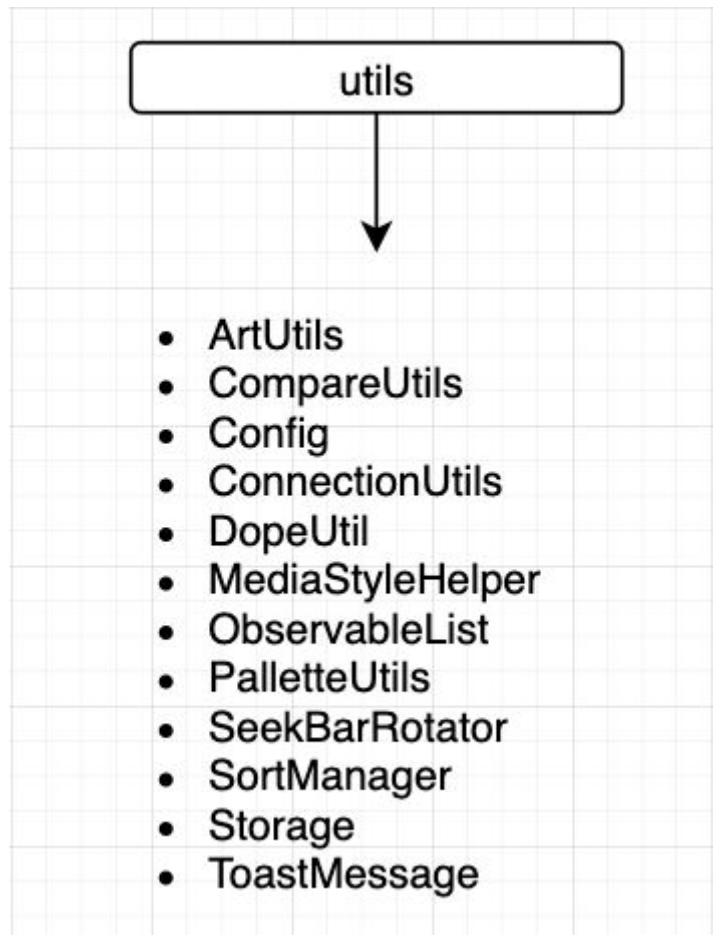


Рисунок 3.12 - Пакет utils

Пакет `utils` включає в себе функціональні набори об'єктів, методів для легшої роботи з колекціями, сортуванням, пам'яттю мобільного пристрою, спливаючих сповіщень, підключенням до мережі.

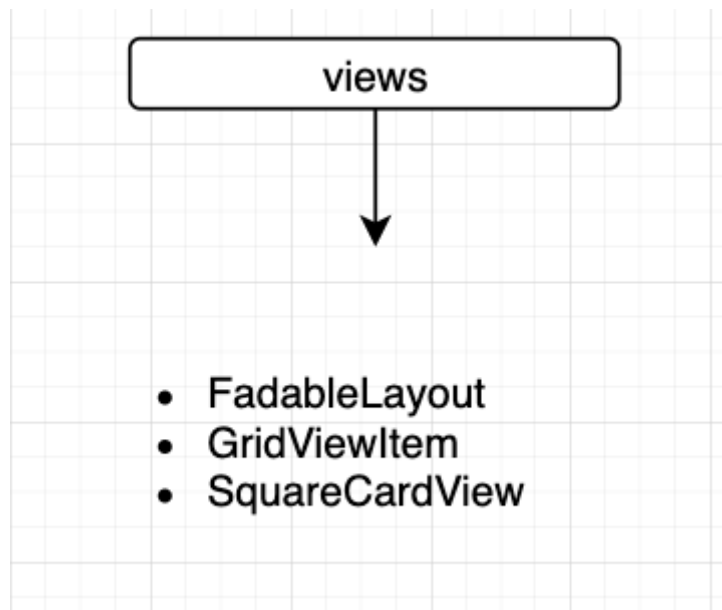


Рисунок 3.13 - Пакет views

Пакет views включає в себе кастомні реалізації деяких об'єктів інтерфейсу. Це зроблено для того, щоб їх можна було використовувати на макетах користувацького інтерфейсу з додатковими функціями.

### 3.5 Розробка інтерфейсу користувача

Для реалізації застосунку потрібно розробити макети інтерфейсу користувача. Реалізований застосунок містить в собі 2 активності (Activity):

- MainActivity;
- SettingsActivity.

Макет користувача описується за допомогою мови розмітки XML. Всі елементи, що можна відобразити на екрані смартфона наслідуються від класу View - базового класу який відображає макет у вигляді простого блоку.

Загальну ієрархію об'єктів, які визначають інтерфейс користувача можна побачити на рисунку 3.14.

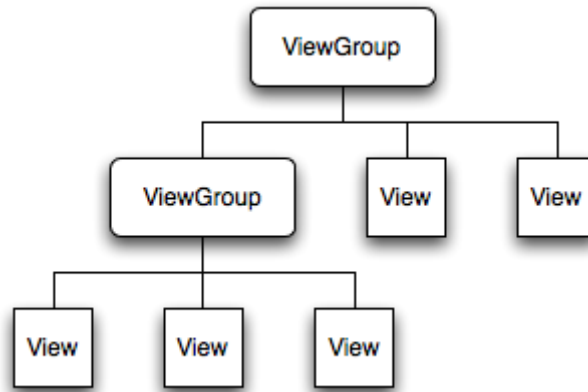


Рисунок 3.14 - загальна ієрархія об'єктів

Всім елементам, до яких потрібно звернутися з класу може бути присвоєний унікальний ідентифікатор.

Кожна з активностей повинна мати відповідний собі макет. Це робиться для того, щоб було зручно описувати інтерфейси користувача незалежно один від одного.

Макет `activity_main.xml` (рис. 3.15) включає в себе 2 основних об'єкти:

- `SlidingUpPanelLayout` (основний об'єкт головного екрану, на якому реалізовано функції анімації та ефект “їзди” інших об'єктів поверх нього);
- `NavigationView` (нативний об'єкт ОС Android, предствляє собою бокову панель з переліком доступних функцій(екранів) для переходу на них).

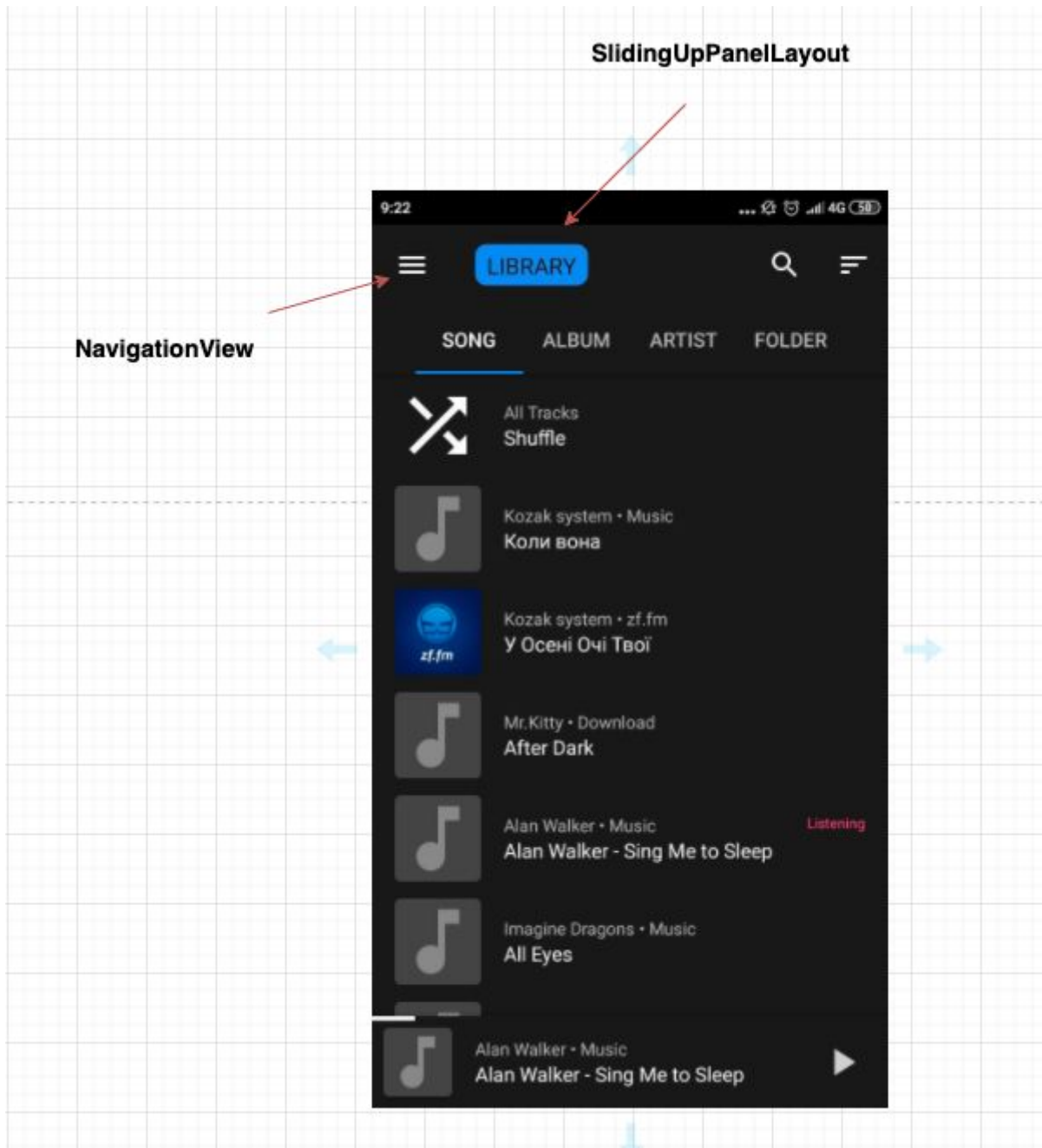


Рисунок 3.15 - main\_activity.xml макет

Макет settings\_activity.xml (рис. 3.16) включає в себе такі об'єкти:

- AppBarLayout (включає в себе Toolbar та TextView об'єкти. Toolbar представляє собою нативний елемент системи для відображення невеликої панелі у верхній частині екрану з текстом TextView в середині);

- `FrameLayout` (нативний об'єкт ОС Android, представляє собою розмітку макету).

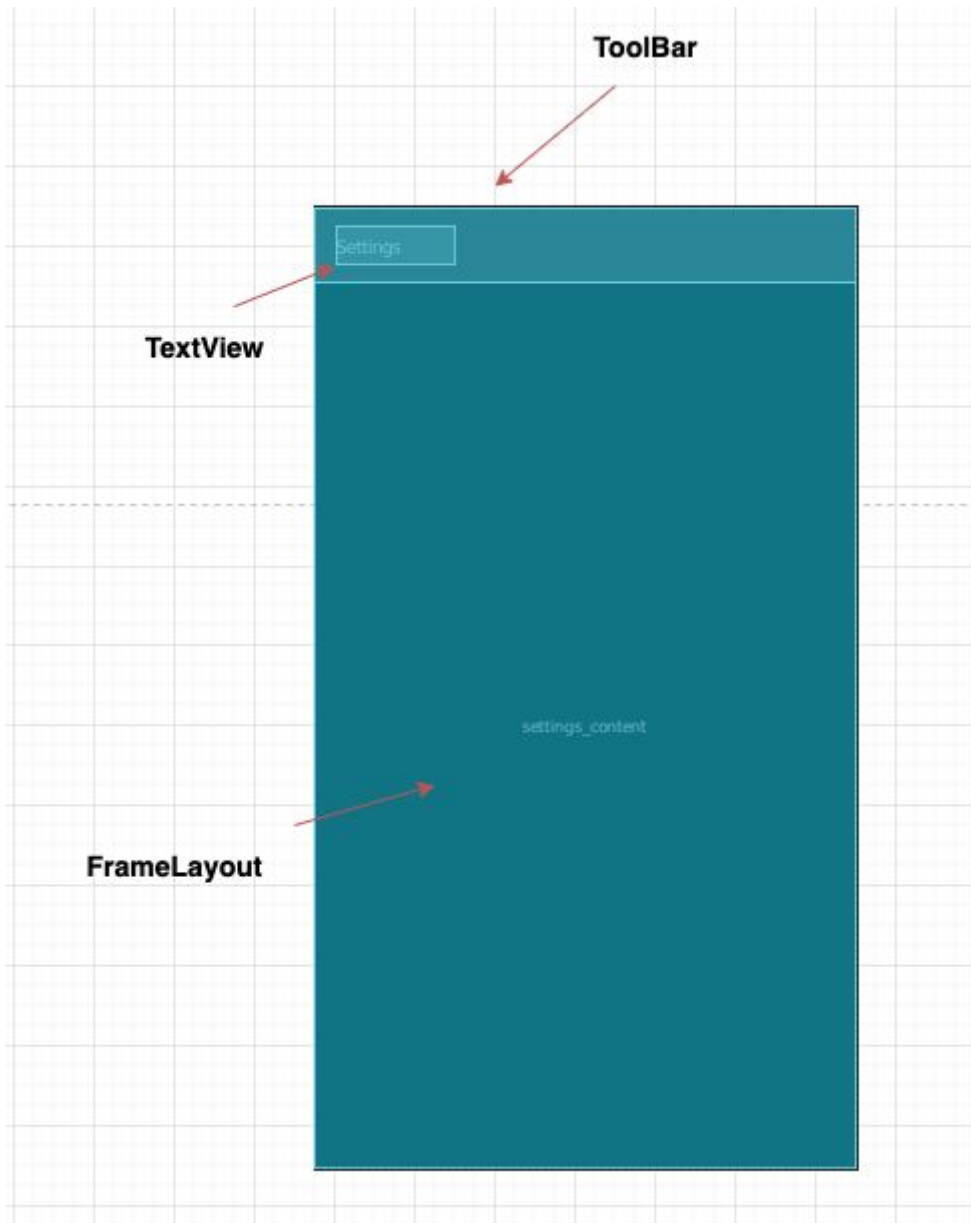


Рисунок 3.16 - `settings_activity.xml` макет

Для кожного з фрагментів також потрібно описати макет. Проте, не кожен макет потребує наявності зв'язуючого класу. Макет може бути частиною іншого макету.

Для опису макету музичного плеєру необхідно описати більшу кількість елементів. Макет може включати в себе такі елементи:

- SquareCardView;
- ImageView;
- FrameLayout;
- TextView;
- ConstraintLayout;
- ImageButton;
- AppCompatActivity;
- LinearLayout;
- AppCompatActivity;
- RelativeLayout;
- ConstraintLayout;

Загальну структуру макету frag\_player.xml можна побачити на рисунку 3.17.

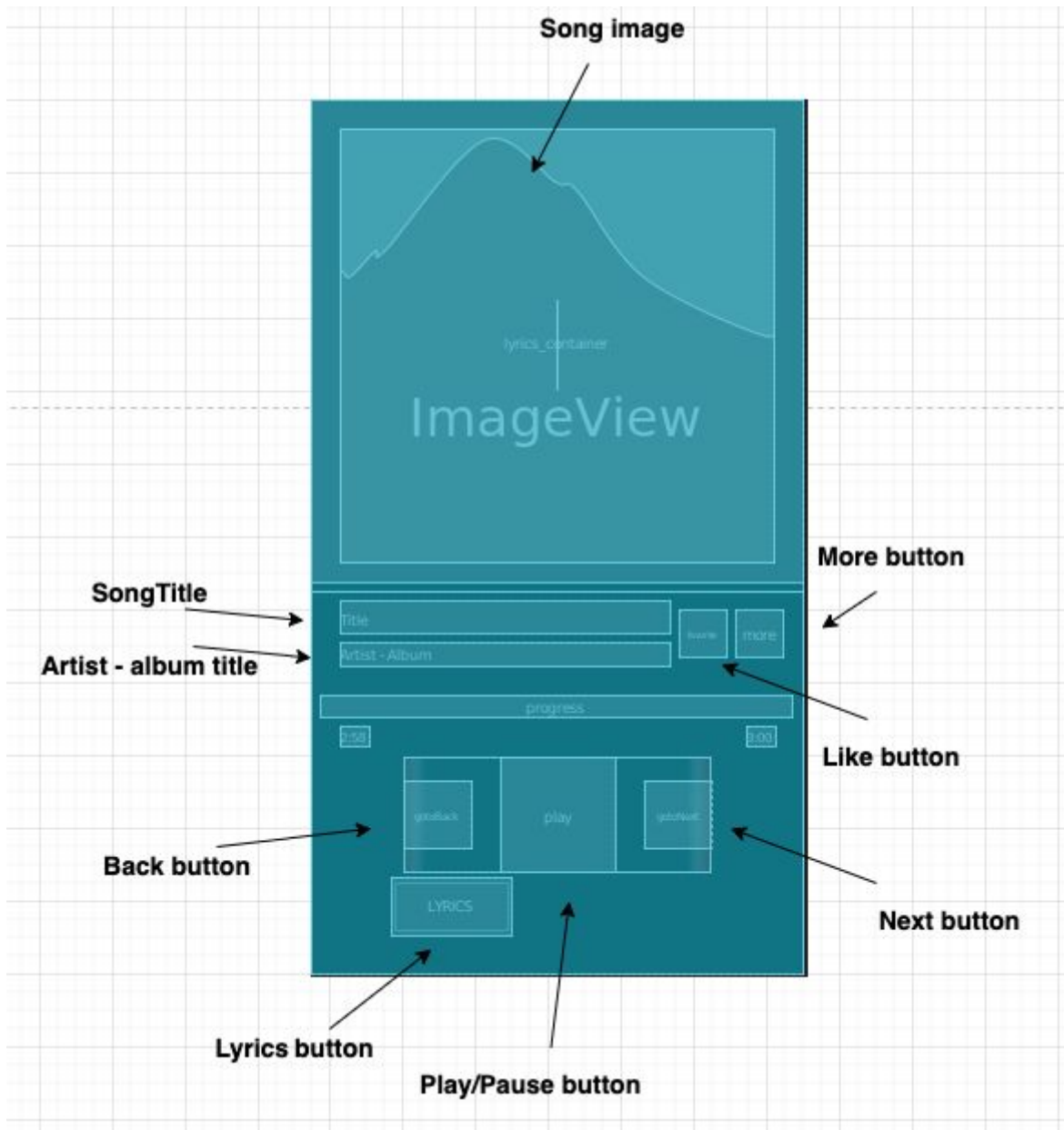


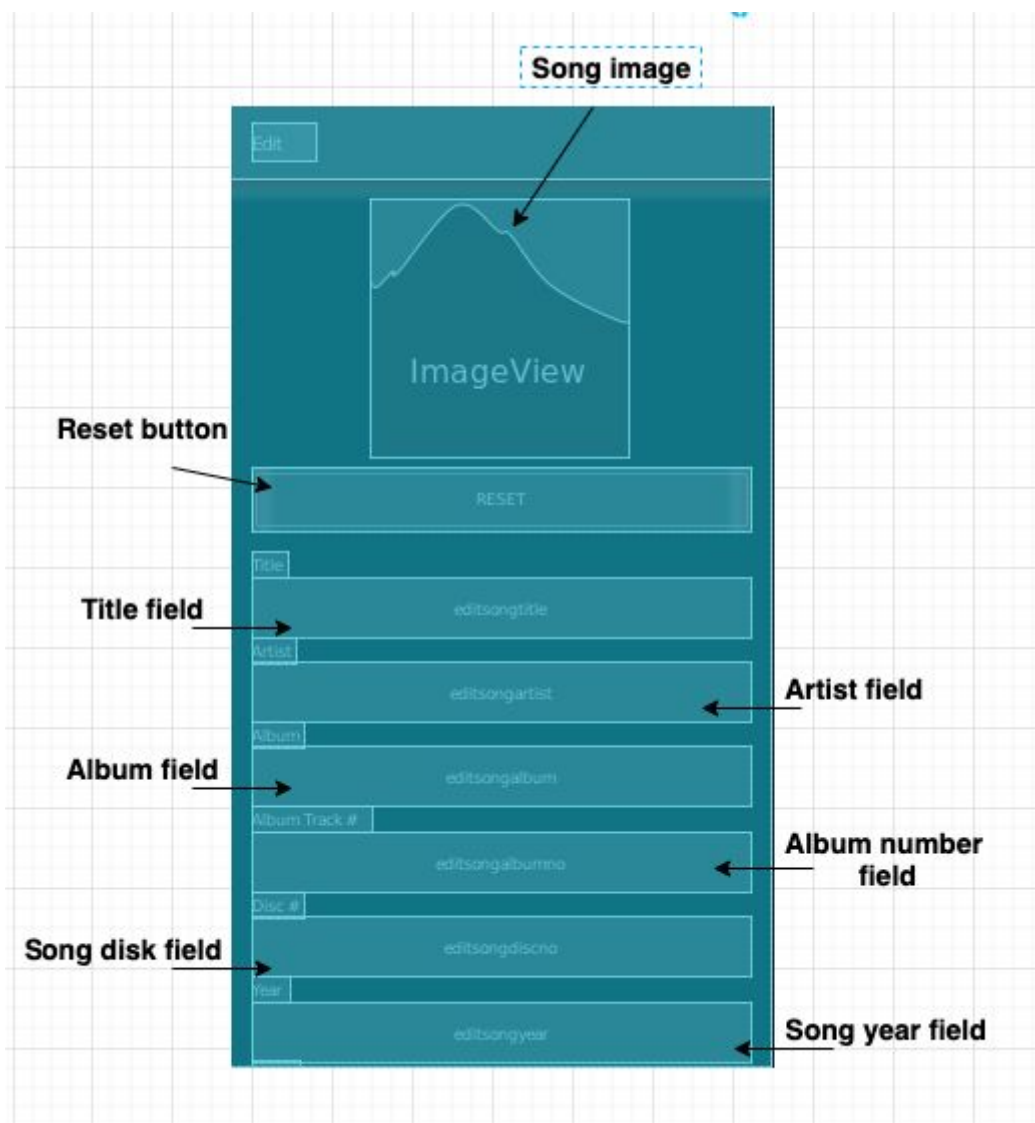
Рисунок 3.17 - frag\_player макет.

Макет `frag_edit_song.xml` представляє собою екран для редагування тегів пісні. Включає в себе наступні елементи:

- `ImageView`;
- `TextView`;
- `ScrollView`;
- `LinearLayout`;
- `Button`;

- EditText;
- RelativeLayout.

Загальну структуру макету frag\_edit\_song.xml можна побачити на рисунку 3.18.



Рисунк 3.18 - frag\_edit\_song.xml макет.

Також, для макету можна застосувати свій стиль/тему. Це можна зробити за допомогою файлу стилів styles.xml. Стиль задає набір різних кольорів, які макет може використовувати для текстів, фонів і т.д. На рисунку 3.19. можна побачити

приклад організації стилю.

```
<style name="AppTheme" parent="Theme.AppCompat.NoActionBar">
  <item name="colorPrimary">@color/colorPrimary</item>
  <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
  <item name="colorAccent">@color/colorAccent</item>
  <item name="artBackground">@color/artBackground</item>
  <item name="android:colorBackground">@color/colorBackground</item>
</style>
```

Рисунок 3.19 - Приклад організації стилю

Задля зручного перекладу застосунку в майбутньому, передбачено файл `strings.xml`, який містить в собі інформацію про текстові дані в наступному вигляді:

```
<string name = "stringName">String Value</string>, де
```

`stringName` - ідентифікатор текстових даних,  
`String Value` - значення.

Можна відображати цілий застосунок на різних мовах, додавши новий файл текстових даних з потрібним параметром значення в папку `values-uk`. У випадку, якщо у системі мова Українська - для застосунку буде вибрана папка `values-uk` для перекладу мови застосунку на українську. Приклад такого файлу на рисунку 3.20.

Key	Path	Enabled	Value
app_name	app/src/main/res	<input type="checkbox"/>	Fast Lyrics
action_play	app/src/main/res	<input type="checkbox"/>	Play
action_pause	app/src/main/res	<input type="checkbox"/>	Pause
info_library	app/src/main/res	<input type="checkbox"/>	Library
info_all_tracks	app/src/main/res	<input type="checkbox"/>	All Tracks
action_shuffle	app/src/main/res	<input type="checkbox"/>	Shuffle
empty_view_title	app/src/main/res	<input type="checkbox"/>	Nothing here
meta_album_art	app/src/main/res	<input type="checkbox"/>	Album Art
action_sort	app/src/main/res	<input type="checkbox"/>	Sort
action_search	app/src/main/res	<input type="checkbox"/>	Search
action_sort_default	app/src/main/res	<input type="checkbox"/>	Default
action_sort_title	app/src/main/res	<input type="checkbox"/>	Title
action_sort_artist_name	app/src/main/res	<input type="checkbox"/>	Artist Name
action_sort_album_name	app/src/main/res	<input type="checkbox"/>	Album Name
action_sort_date_added	app/src/main/res	<input type="checkbox"/>	Date Added
action_sort_year	app/src/main/res	<input type="checkbox"/>	Year
action_sort_duration	app/src/main/res	<input type="checkbox"/>	Duration
action_sort_track	app/src/main/res	<input type="checkbox"/>	Track #

Рисунок 3.20 - Редактор локалізації

### 3.6 Висновки до розділу

У розділі 3 були визначені та описані основні бізнес-процеси, представлено та описано структуру бази даних. Вибрані засоби розробки інформаційної технології, представлено її структуру, класи, взаємодію між ними. Описано макети користувацького інтерфейсу та принцип їх побудови. Представлено схему функцій користувача.

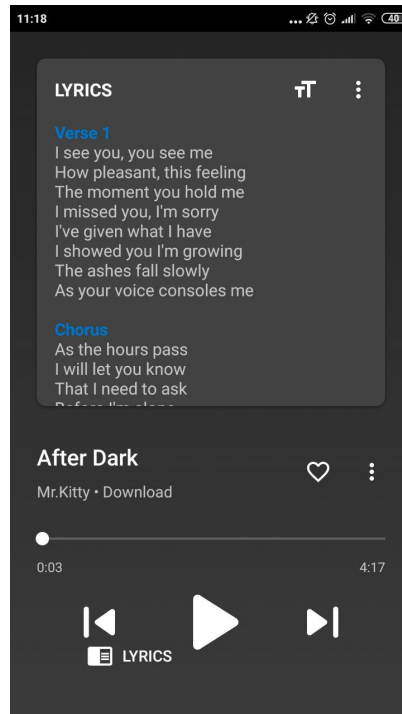
## Розділ 4

### Апробація програмної інформаційної технології

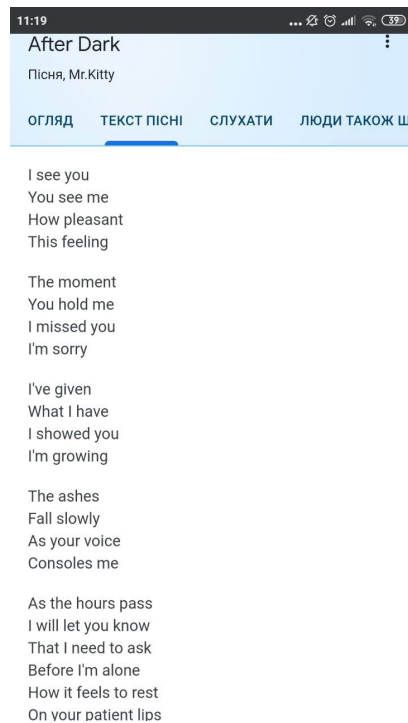
Для перевірки правильності роботи інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв, візьмемо як вхідні дані наступні пісні, а саме виконавця та назву пісні:

- Mr.Kitty - After Dark (рис. 4.1);
- Alan Walker - Sing Me to Sleep (рис. 4.2);
- Queen - We Will Rock You (рис 4.3);
- Linkin Park - Breaking The Habit (рис. 4.4)
- Bring Me The Horizon - Can You Feel My Heart (рис. 4.5);
- Twenty One Pilots - Chlorine (рис. 4.6);
- Linkin Park - Heavy (рис. 4.7);
- Black Veil Brides - In The End (рис. 4.8);
- Linkin Park - Castle Of Glass (рис 4.9);
- Queen - Don't Stop Me Now (рис. 4.10).

Як сервіс для перевірки правильності завантажених слів було взято пошук Google.



a)

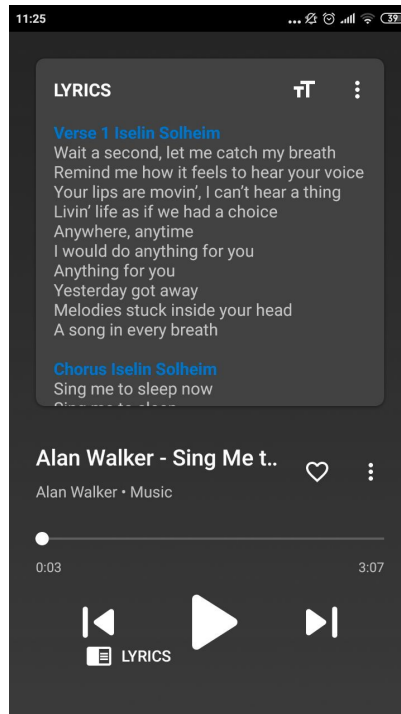


б)

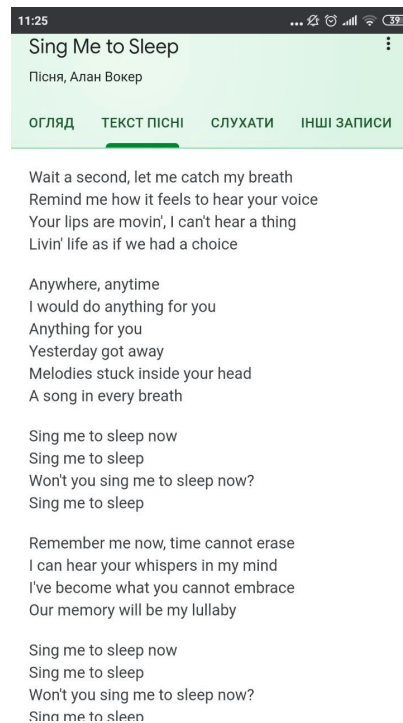
Рисунок 4.1 - Слова пісні Mr.Kitty - After Dark:

а) результат роботи реалізованої інформаційної системи пошуку слів;

б) результати пошуку Google.



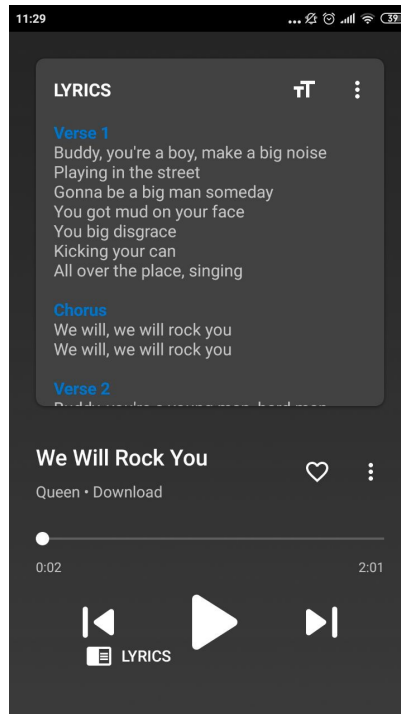
a)



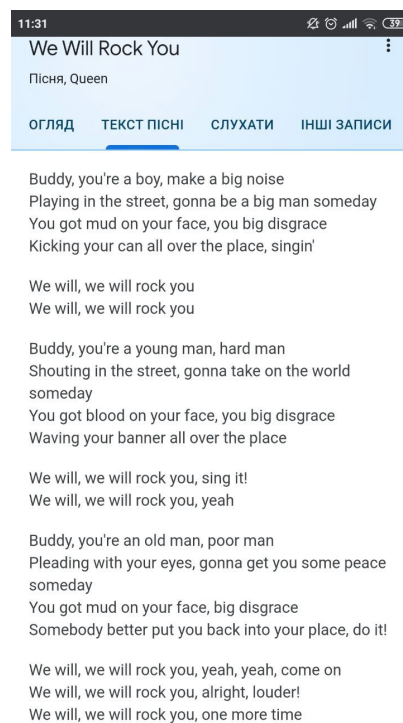
б)

Рисунок 4.2 - Слова пісні Alan Walker - Sing Me to Sleep:

- а) результат роботи реалізованої інформаційної системи пошуку слів;  
 б) результати пошуку Google.



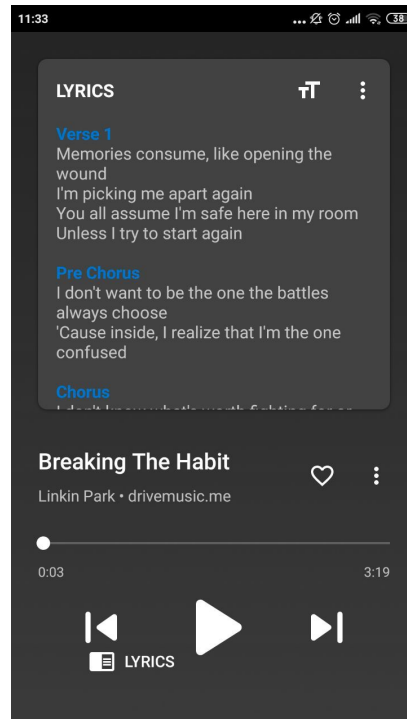
a)



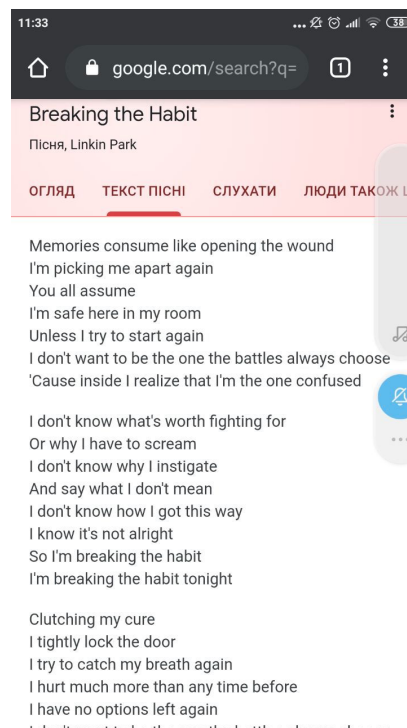
б)

Рисунок 4.3 - Слова пісні Queen - We Will Rock You:

- а) результат роботи реалізованої інформаційної системи пошуку слів;  
 б) результати пошуку Google.



a)

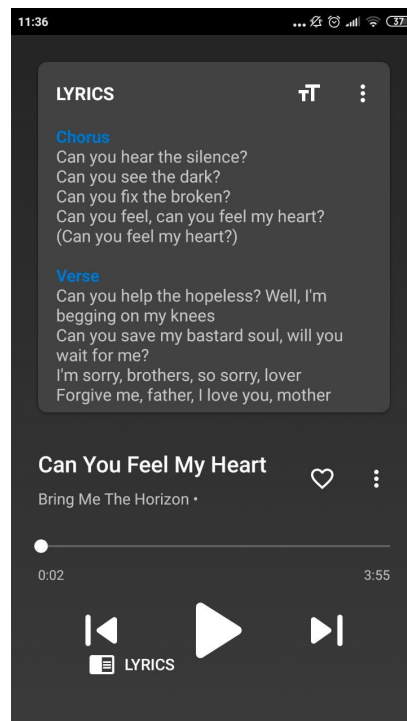


б)

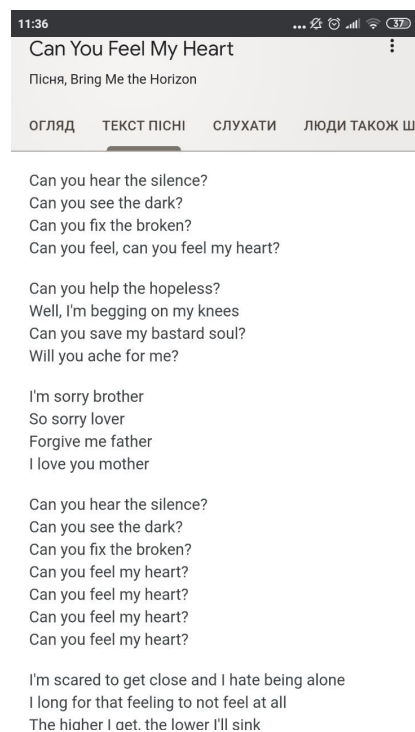
Рисунок 4.4 - Слова пісні Linkin Park - Breaking The Habit:

а) результат роботи реалізованої інформаційної системи пошуку слів;

б) результати пошуку Google.



a)

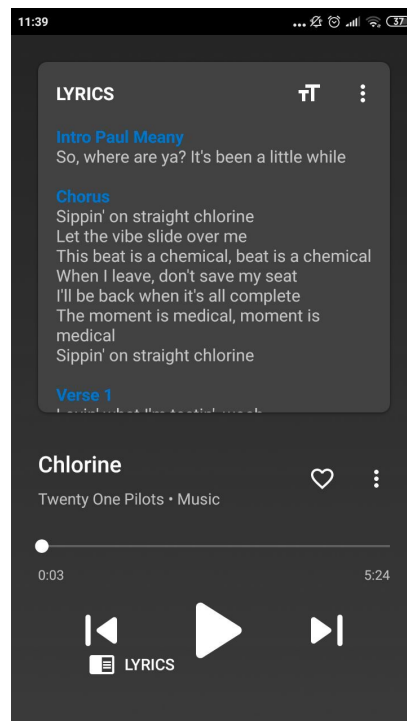


б)

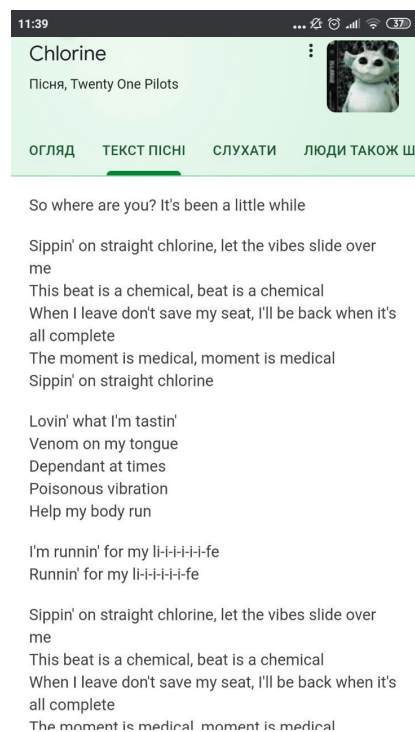
Рисунок 4.5 - Слова пісні Bring Me The Horizon - Can You Feel My Heart:

а) результат роботи реалізованої інформаційної системи пошуку слів;

б) результати пошуку Google.



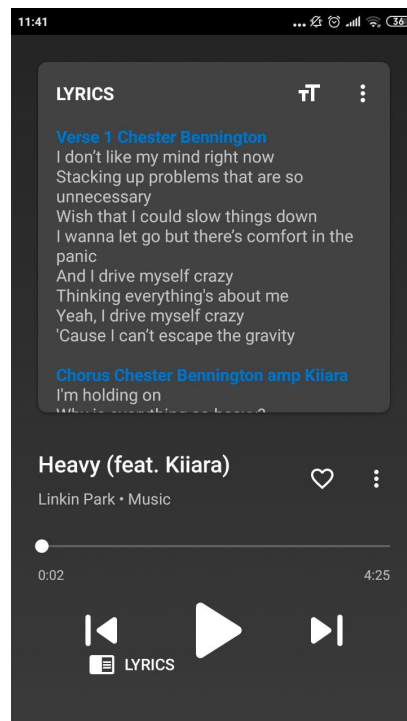
a)



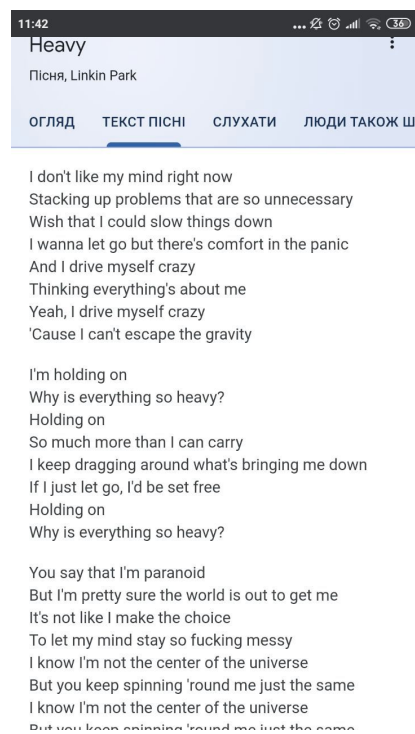
б)

Рисунок 4.6 - Слова пісні Twenty One Pilots - Chlorine:

- а) результат роботи реалізованої інформаційної системи пошуку слів;  
б) результати пошуку Google.



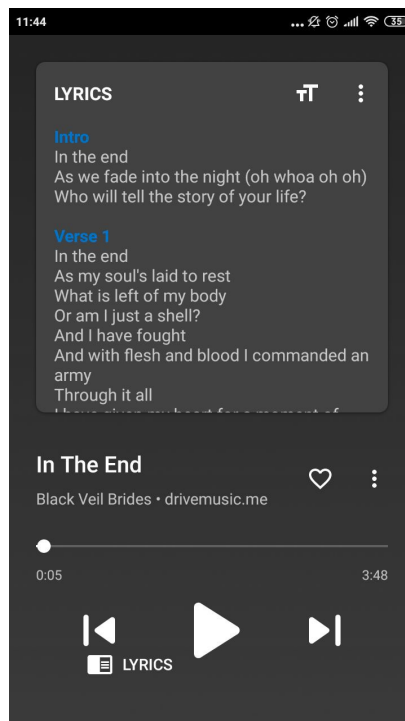
a)



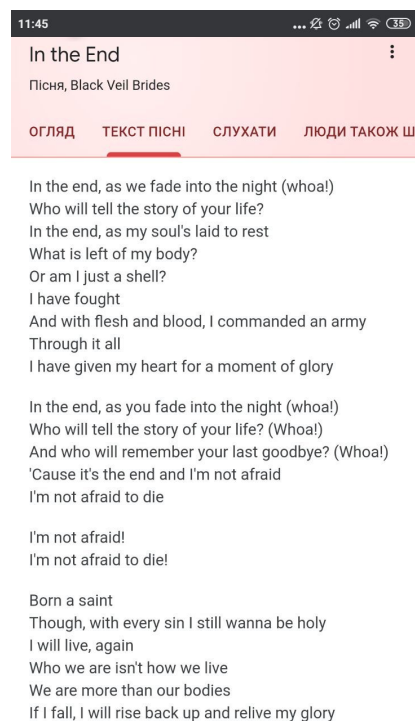
б)

Рисунок 4.7 - Слова пісні Linkin Park - Heavy:

- а) результат роботи реалізованої інформаційної системи пошуку слів;  
 б) результати пошуку Google.



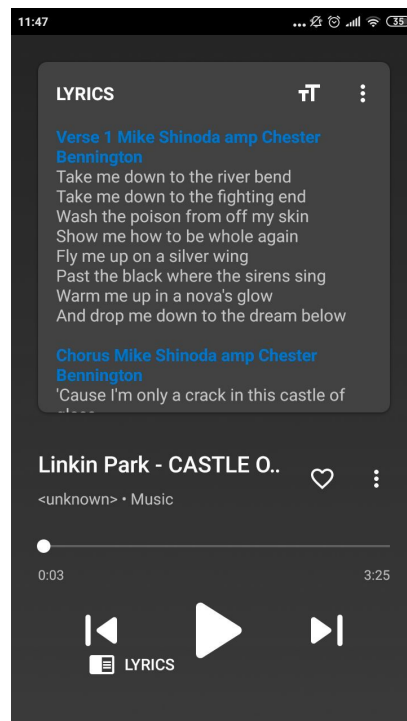
a)



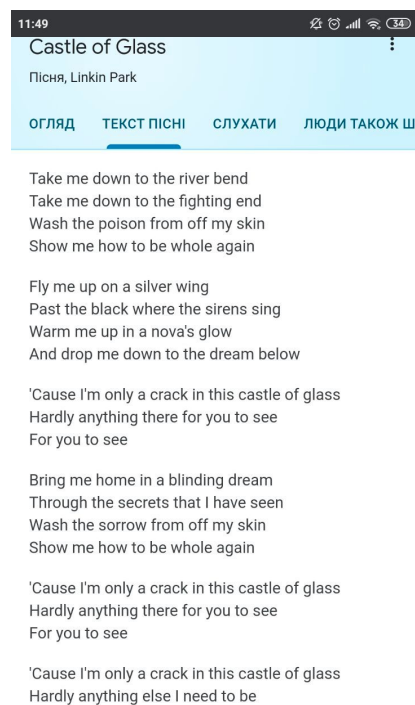
б)

Рисунок 4.8 - Слова пісні Black Veil Brides - In The End:

- а) результат роботи реалізованої інформаційної системи пошуку слів;  
б) результати пошуку Google.



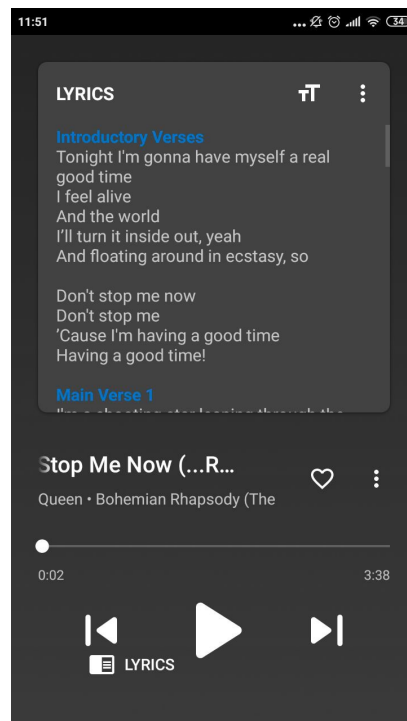
a)



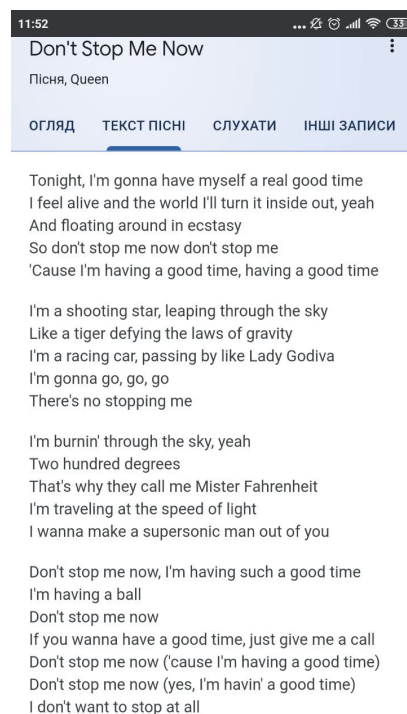
б)

Рисунок 4.9 - Слова пісні Linkin Park- Castle Of Glass:

- a) результат роботи реалізованої інформаційної системи пошуку слів;  
 б) результати пошуку Google.



a)



б)

Рисунок 4.10 - Слова пісні Queen - Don't Stop Me Now:

- а) результат роботи реалізованої інформаційної системи пошуку слів;  
 б) результати пошуку Google.

Проведене моделювання для якого, як вхідні дані було взято 10 пісень, а саме назва виконавця та назва пісні. Як результат було отримано слова цих пісень. Задля перевірки правильності отриманих даних, слова порівняли зі словами з надійного джерела. Відмінностей у результатах не було знайдено, що засвідчує про коректність роботи програми.

#### **4.1 Висновки до розділу**

Проведена експериментальна апробація результатів роботи реалізованої інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв підтвердила її адекватність. Як приклад був виконаний аналіз 10 музичних композицій різних виконавців та жанрів. В усіх проаналізованих модельних ситуаціях було отримано повне співпадіння.

## Загальні висновки

В результаті виконаної дипломної роботи магістра було реалізовано інформаційну технологію завантаження слів пісень в реальному часі для мобільних пристроїв з використанням загальнодоступних мережевих аудіо-хостингів.

При досягненні поставленої мети були вирішені такі задачі дослідження:

- проведено аналіз існуючих методів, технологій та рішень завантаження слів пісень в реальному часі для мобільних пристроїв;
- удосконалено методику завантаження слів пісень у напрямку отримання максимально повних і достовірних даних;
- розроблено інформаційну технологію завантаження слів пісень в реальному часі для мобільних пристроїв;
- виконано експериментальну перевірку інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

Було визначено структуру інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв. Описано процедури забезпечення функціональності інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв, потоки даних та інформації між модулями системи з представленням інформаційних діаграм.

Визначені основні бізнес-процеси, представлено та описано структуру бази даних. Вибрані засоби розробки інформаційної системи, представлено її структуру, класи, взаємодію між ними. Описано макети користувацького інтерфейсу та принцип їх побудови. Представлено схему функцій користувача.

Проведена експериментальна апробація результатів роботи реалізованої інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв підтвердила її адекватність.

### Перелік посилань

1. Sony Music [Електронний ресурс]. - Режим доступу: <https://www.sonymusic.com/>
2. Warner Music Group [Електронний ресурс]. - Режим доступу: <https://www.wmg.com/>
3. Universal Music Group [Електронний ресурс]. - Режим доступу: <https://www.universalmusic.com/>
4. Spotify [Електронний ресурс].- Режим доступу: <https://www.spotify.com/us/>
5. Spotify Music Player [Електронний ресурс]. - Режим доступу: <https://open.spotify.com/search>
6. Shazam [Електронний ресурс]. - Режим доступу: <https://www.shazam.com/gb>
7. Musixmatch [Електронний ресурс]. - Режим доступу: <https://play.google.com/store/apps/details?id=com.musixmatch.android.lyrify&hl=en&gl=US>
8. Google Play Music [Електронний ресурс]. - Режим доступу: <https://music.youtube.com/>
9. Deezer [Електронний ресурс]. - Режим доступу: <https://www.deezer.com/en/>
10. Shazam app [Електронний ресурс]. - Режим доступу: <https://play.google.com/store/apps/details?id=com.shazam.android&hl=en&gl=US>
11. Bluetooth [Електронний ресурс]. - Режим доступу: <https://en.wikipedia.org/wiki/Bluetooth>
12. Android Beam [Електронний ресурс]. - Режим доступу: [https://en.wikipedia.org/wiki/Android\\_Beam](https://en.wikipedia.org/wiki/Android_Beam)
13. Guide to music metadata [Електронний ресурс]. - Режим доступу: <https://soundcharts.com/blog/music-metadata>

14. Genius API [Электронный ресурс]. - Режим доступа: <https://docs.genius.com/>
15. iTunes API [Электронный ресурс]. - Режим доступа: <https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/index.html>
16. LastFM API [Электронный ресурс]. - Режим доступа: <https://www.last.fm/api>
17. HTTP [Электронный ресурс]. - Режим доступа: [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
18. REST [Электронный ресурс]. - Режим доступа: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)
19. LiveData [Электронный ресурс]. - Режим доступа: <https://developer.android.com/topic/libraries/architecture/livedata>
20. ViewModel [Электронный ресурс]. - Режим доступа: <https://developer.android.com/topic/libraries/architecture/viewmodel>
21. Dependency Injection [Электронный ресурс]. - Режим доступа: <https://developer.android.com/training/dependency-injection>
22. Shared Preferences [Электронный ресурс]. - Режим доступа: <https://developer.android.com/reference/android/content/SharedPreferences>
23. Kotlin [Электронный ресурс]. - Режим доступа: <https://kotlinlang.org/docs/reference/android-overview.html>
24. Room Library [Электронный ресурс]. - Режим доступа: <https://developer.android.com/jetpack/androidx/releases/room>
25. Sqlite for android [Электронный ресурс]. - Режим доступа: <http://developer.alexanderklimov.ru/android/sqlite/android-sqlite.php>
26. Android Studio [Электронный ресурс]. - Режим доступа: <https://developer.android.com/studio>
27. Android SDK [Электронный ресурс]. - Режим доступа: <https://developer.android.com/studio/releases/platform-tools>

28. Koin [Електронний ресурс]. - Режим доступу: <https://insert-koin.io/>
29. RecyclerView [Електронний ресурс]. - Режим доступу: <https://developer.android.com/guide/topics/ui/layout/recyclerview>
30. MediaPlayer [Електронний ресурс]. - Режим доступу: <https://developer.android.com/reference/android/media/MediaPlayer>
31. Fragment [Електронний ресурс]. - Режим доступу : <https://developer.android.com/guide/components/fragments>
32. Життєвий цикл фрагмента [Електронний ресурс]. - Режим доступу: <https://www.journaldev.com/9266/android-fragment-lifecycle>
33. Activity [Електронний ресурс]. - Режим доступу: <https://developer.android.com/reference/android/app/Activity>
34. XML [Електронний ресурс]. - Режим доступу: <https://developer.android.com/guide/topics/ui/declaring-layout>
35. Як вчити англійську мову за допомогою пісень? - Режим доступу: <https://englishstudent.net/blog/2017/11/23/yak-vchiti-anglijsku-movu-za-dopomogoyu-pisen/>
36. П.О. Грицишин, О.А. Пасічник, Т.К. Скрипник. Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв // Вісник Хмельницького національного університету. 2020 (подано до редакції).

# ДОДАТКИ

## Додаток А

### Даталогічна модель БД

Name	Type	Schema
▼ Tables (10)		
▼ album_stats		CREATE TABLE `album_stats` (`albumid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL, PRIMARY KEY(`albumid`))
albumid	INTEGER	"albumid" INTEGER NOT NULL
playcount	INTEGER	"playcount" INTEGER NOT NULL
playtime	INTEGER	"playtime" INTEGER NOT NULL
▼ android_metadata		CREATE TABLE android_metadata (locale TEXT)
locale	TEXT	"locale" TEXT
▼ artist_stats		CREATE TABLE `artist_stats` (`artistid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL, PRIMARY KEY(`artistid`))
artistid	INTEGER	"artistid" INTEGER NOT NULL
playcount	INTEGER	"playcount" INTEGER NOT NULL
playtime	INTEGER	"playtime" INTEGER NOT NULL
▼ bio		CREATE TABLE `bio` (`artistid` INTEGER NOT NULL, `bio` TEXT, `tags` TEXT, PRIMARY KEY(`artistid`))
artistid	INTEGER	"artistid" INTEGER NOT NULL
bio	TEXT	"bio" TEXT
tags	TEXT	"tags" TEXT
▼ equalizer		CREATE TABLE `equalizer` (`eqid` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `eqname` TEXT NOT NULL)
eqid	INTEGER	"eqid" INTEGER NOT NULL
eqname	TEXT	"eqname" TEXT NOT NULL
▼ favorites		CREATE TABLE `favorites` (`fav` INTEGER NOT NULL, PRIMARY KEY(`fav`))
fav	INTEGER	"fav" INTEGER NOT NULL
▼ lyrics		CREATE TABLE `lyrics` (`songid` INTEGER NOT NULL, `lyric` TEXT NOT NULL, PRIMARY KEY(`songid`))
songid	INTEGER	"songid" INTEGER NOT NULL
lyric	TEXT	"lyric" TEXT NOT NULL
▶ room_master_table		CREATE TABLE room_master_table (id INTEGER PRIMARY KEY,identity_hash TEXT)
▼ song_stats		CREATE TABLE `song_stats` (`songid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL, PRIMARY KEY(`songid`))
songid	INTEGER	"songid" INTEGER NOT NULL
playcount	INTEGER	"playcount" INTEGER NOT NULL
playtime	INTEGER	"playtime" INTEGER NOT NULL
▼ sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
Indices (0)		
Views (0)		
Triggers (0)		

## Додаток Б

### Файл AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.sandman.android"
    android:installLocation="auto">

    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
    <permission android:name="android.permission.MEDIA_CONTENT_CONTROL" />
    <application
        android:name="com.sandman.android.App"
        android:allowBackup="true"
        android:fullBackupContent=""
        android:hardwareAccelerated="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:largeHeap="true"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:usesCleartextTraffic="true"
        tools:targetApi="m">
        <activity
            android:name="com.sandman.android.ui.MainActivity"
            android:launchMode="singleInstance"
            android:screenOrientation="portrait"
            android:windowSoftInputMode="adjustPan">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <action android:name="android.intent.action.MUSIC_PLAYER" />
                <category android:name="android.intent.category.LAUNCHER" />
                <category android:name="android.intent.category.APP_MUSIC" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
                <data android:scheme="content" />
                <data android:mimeType="audio/*" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
                <data android:scheme="file" />
                <data android:mimeType="audio/*" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.sandman.android.ui.settings.SettingsActivity"
            android:parentActivityName="com.sandman.android.ui.MainActivity" />
        <service
            android:name="com.sandman.android.service.MusicService"
            android:exported="false">
            <intent-filter>
                <action android:name="android.intent.action.MEDIA_BUTTON" />
                <action android:name="com.sandman.android.widgetintents.previous" />
                <action android:name="com.sandman.android.widgetintents.play_pause" />
                <action android:name="com.sandman.android.widgetintents.next" />
            </intent-filter>
        </service>
        <service
            android:name="com.sandman.android.service.CollectionService"
            android:exported="false">
            <intent-filter>

```

```
        <action android:name="com.sandman.android.collectionservice.ACTION_COLLECT_LYRICS" />
        <action android:name="com.sandman.android.collectionservice.ACTION_STOP" />
    </intent-filter>
</service>
<receiver android:name="androidx.media.session.MediaButtonReceiver">
    <intent-filter>
        <action android:name="android.intent.action.MEDIA_BUTTON" />
    </intent-filter>
</receiver>
<meta-data
    android:name="preloaded_fonts"
    android:resource="@array/preloaded_fonts" />
</application>
</manifest>
```

## Додаток В

### Файл App

```

package com.sandman.android
import android.content.pm.PackageManager
import androidx.core.content.ContextCompat
import androidx.multidex.MultiDexApplication
import com.bumptech.glide.Glide
import com.sandman.android.data.MediaRepository
import com.sandman.android.di.appModule
import org.jaudiotagger.tag.TagOptionSingleton
import org.koin.android.ext.android.inject
import org.koin.android.ext.android.startKoin
class App : MultiDexApplication() {
    val mediaRepository by inject<MediaRepository>()

    override fun onCreate() {
        super.onCreate()
        startKoin(this, appModule)
        instance = this
        TagOptionSingleton.getInstance().isAndroid = true
        cleanGenres()
    }
    override fun onLowMemory() {
        super.onLowMemory()
        Glide.get(this).clearMemory()
    }
    private fun cleanGenres() {
        if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.READ_EXTERNAL_STORAGE) !=
        PackageManager.PERMISSION_GRANTED)
            return
        mediaRepository.cleanGenres()
    }
    companion object {
        @get: Synchronized
        lateinit var instance: App
    }
}

```

## Додаток Г

### Лістинги модулів DI

```
val viewModelModule = module {

    viewModel { AlbumDetailViewModel(get(), get(), get()) }
    viewModel { ArtistDetailViewModel(get(), get(), get()) }

    viewModel { TrackViewModel(get(), get()) }
    viewModel { FolderViewModel(get(), get(), get()) }

    viewModel { FavoritesViewModel(get(), get(), get()) }
    viewModel { SearchViewModel(get(), get(), get()) }
    viewModel { EditorViewModel(get(), get(), get()) }

    viewModel { MiniPlayerViewModel(get()) }
    viewModel { PlayerViewModel(get(), get(), get()) }
    viewModel { QueueViewModel(get(), get()) }
}

val repositoryModule = module {

    factory { Room.databaseBuilder(get(), LocalDatabase::class.java, "fl.db")
        .addMigrations(LocalDatabase.Migration_5_6)
        .allowMainThreadQueries()
        .build().dataDao() }

    factory { RemoteDataSource }

    single { LocalDataSource(get()) }

    single { DataRepository(get(), get()) }

    single { MediaRepository(get(), get()) }
}

val commonModule = module {
    factory { PreferenceManager.getDefaultSharedPreferences(get()) }

    factory { MiniPlayerFragment() }

    factory { PlayerFragment() }

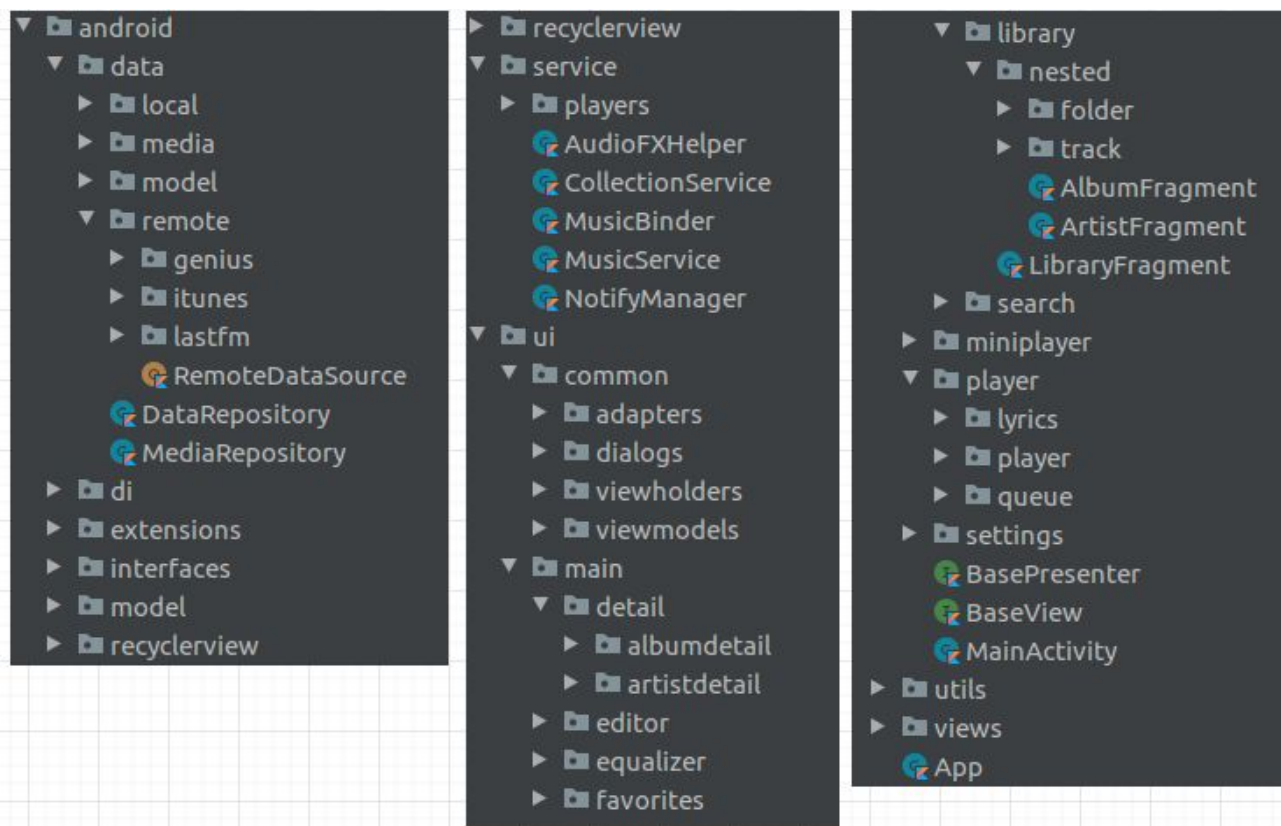
    factory { LyricsFragment() }

    factory { LyricsPresenter(get(), get()) as LyricsContract.Presenter }

    factory { EqualizerPresenter() as EqualizerContract.Presenter }
}
```

## Додаток Д

### Загальна структура пакетів



## ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗАВАНТАЖЕННЯ СЛІВ ПІСЕНЬ В РЕАЛЬНОМУ ЧАСІ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

Реалізована інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв. Програмний застосунок забезпечує завантаження слів музичної композиції на основі її метаданих з подальшим відображенням тексту на екрані мобільного пристрою. Результати пошуку та завантаження зберігаються у локальній базі даних задля забезпечення оффлайн режиму доступу. Реалізовано інтеграцію з популярним відео-хостингом YouTube, що забезпечує розширення функціональних можливостей реалізованого програмного застосунку з прослуховування аудіо творів на перегляд мультимедійного контенту. Наявний функціонал інформаційної системи дозволяє використовувати її для вивчення іноземних мов.

Ключові слова: інформаційна технологія, виконавець, слова пісні, навчання.

P.O. HRYTSYSHYN, O.A. PASICHNYK, T.K. SKRYPNYK

Khmelnitskyi National University

### REAL-TIME LYRICS DOWNLOADING INFORMATION TECHNOLOGY FOR MOBILE DEVICES

Any learning is more effective when we have fun when new material is learned during the game or quest. For example, many admit that they have learned or improved their English by listening to songs and translating them on their own. English songs are not only a great source of "living" language, but also allow you to train several language skills at once. To be successful, learning English should be fun. Learning a language by doing what you love is both easier and more fun. Do you like to sing? Then learn English with songs. Everything you need - a smartphone or tablet! This is a very simple but effective way that requires almost no effort. Songs acquaint us with how carriers express their emotions, feelings, thoughts. Besides, you will have topics to discuss with your English-speaking friends - this can apply to both the content of the songs and the artists. Language and music are two ways of human communication and self-expression. These concepts are inextricably linked by certain basic characteristics, such as rhythm, dynamics, pitch, timbre, and so on. That is why the methods of teaching language and music are very similar. Moreover, scientists claim that people who have good musical abilities can achieve great success in learning the language. Today, with the development of technology, the music industry is moving online, there are many services for streaming audio, which allow you to legally and for free listen to music, provide legal online streaming audio of major world and independent labels, including BBC, Sony, EMI, Warner Music Group, Universal and others. Some services provide extensive information about the song, including its words. The development of new information technologies allows us to develop a society in certain directions. They are actively transforming other technologies of tangible and intangible production, eventually forming a new style of work, a way of life in general. The essence of information technology are methods and means of forming and maintaining information flows in object management systems.

Keywords: information technology, lyrics, performer, studying.

### Аналіз предметної області

Кожен день людина слухає музику і не замислюється над тим, що різні пісні можна слухати не тільки заради свого задоволення, але і з їх допомогою також можна поліпшити свої навички англійської. Адже в наш час багато відомих хітів написано саме на цій мові, оскільки вона одна з найбільш популярних.

Але чому прослуховування музики може допомагати вивчати іноземну мову? Відповідь полягає в тому, що Ви самі себе навчаєте і при цьому отримуєте задоволення. І, напевно, це один з найбільш простих і ефективних способів розширити свій словниковий запас. Так як під час прослуховування пісні, яка нам подобається, ми автоматично намагаємося запам'ятати слова. Тому чим більше музики Ви слухаєте, тим краще. В наш час популярність додатків, що відтворюють музику, є надвисокою, проте, просто прослуховування музики можна розширити та надати додаткової користі — опанування іноземної мови.

### Аналіз існуючого програмного забезпечення предметної області

Сьогодні з розвитком технологій музична індустрія переходить в он-лайн, з'являється багато сервісів для потокового аудіо, які дозволяють легально та безкоштовно прослуховувати музичні композиції, надають послуги легального онлайнного стримінгу аудіозаписів основних світових і незалежних лейблів, включаючи BBC, Sony, EMI, Warner Music Group, Universal тощо. Деякі сервіси надають розширену інформацію про композиції, включно зі словами.

Одним з прикладів сервісів, які дозволяють перенести сферу музичної індустрії в онлайн, є сервіс "Spotify" [1]. Головна сторінка сайту організована в стилі landing-page, що одразу акцентує увагу користувача на головних перевагах різних видів підписок. Можна перейти безпосередньо до сторінки з композиціями. Також на сайті є досить зручний і інтерактивний пошук композиції за різними фільтрами. Зручним є те, що з головної сторінки можна перейти на сторінку музичного плеєра.

Ще одним прикладом сервісу в сфері аудіо є "Musixmatch" [2]. Мобільні програми "Musixmatch" відображають тексти пісень, синхронізовані з музикою, що відтворюється. У своїх власних програмах він може сканувати всі пісні в музичній бібліотеці користувача, знаходити тексти пісень та використовувати їх як музичний плеєр. На Android він також підтримує сервіси потокового передавання музики, такі як Spotify, Google Play Music [3], Deezer [4]. MusixMatch також дозволяє користувачам вносити тексти пісень та

синхронізації до лірики в обмін на бали, які відповідають рівням та значкам.

### Аналіз та автоматизація обробки інформаційних потоків

Можна виділити основний бізнес-процес застосунку для автоматизації його потреб – завантаження слів музичної композиції.

#### Бізнес-процес «Завантаження слів музичної композиції»

Даний бізнес-процес передбачає опрацювання інформації, що стосується завантаження слів, й включає такі функції:

- збір інформації по композиції, що відтворюється;
- пошук слів у віддалених базах даних, таких як LastFM [5], Genius [6], iTunes [7].
- отримання слів, при першому відтворенні композиції з метадати береться інформація про виконавця та назву композиції. За цими параметрами здійснюється запит на вище згадані сервіси. Знайдені слова відображаються на екрані (рис. 1) та зберігаються у локальній базі даних (рис. 2).

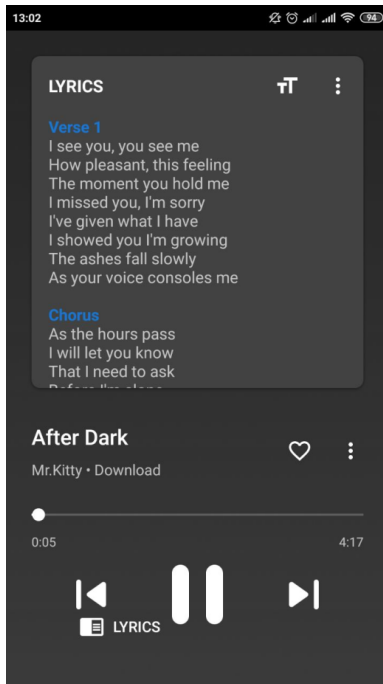


Рис. 1. Завантажені слова на екрані.

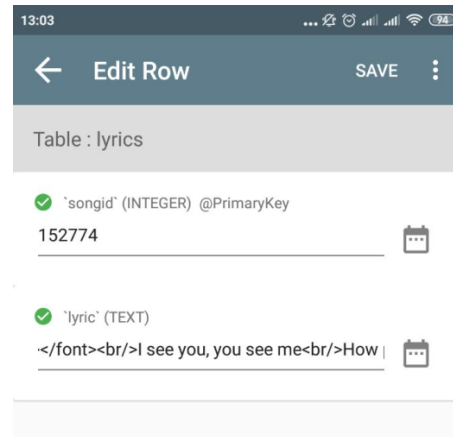


Рис. 2. Завантажені слова у базі даних

Table Name	Fields	Table Definition
album_stats	albumid, playcount, playtime	CREATE TABLE `album_stats` (`albumid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL)
android_metadata	locale	CREATE TABLE android_metadata (locale TEXT)
artist_stats	artistid, playcount, playtime	CREATE TABLE `artist_stats` (`artistid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL)
bio	artistid, bio, tags	CREATE TABLE `bio` (`artistid` INTEGER NOT NULL, `bio` TEXT, `tags` TEXT)
equalizer	eqid, eqname	CREATE TABLE `equalizer` (`eqid` INTEGER PRIMARY KEY AUTOINCREMENT, `eqname` TEXT NOT NULL)
favorites	fav	CREATE TABLE `favorites` (`fav` INTEGER NOT NULL, PRIMARY KEY(`fav`))
lyrics	songid, lyric	CREATE TABLE `lyrics` (`songid` INTEGER NOT NULL, `lyric` TEXT NOT NULL)
room_master_table	id, identity_hash	CREATE TABLE room_master_table (id INTEGER PRIMARY KEY, identity_hash TEXT)
song_stats	songid	CREATE TABLE `song_stats` (`songid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL)

Рис. 3. Представлення бази даних

Для того, щоб автоматизувати роботу веб-сервісу, дані потрібно організувати у таблиці бази даних і провести нормалізацію.

Для збереження слів у локальній базі даних використовується мобільна база даних Room [8]. БД складається з таких таблиць:

- 1) album\_stats (кількість разів відтворення альбому та час відтворення);
- 2) android\_metadata (локаль для завантаження);
- 3) artist\_stats (кількість разів відтворення виконавця та час відтворення);
- 4) bio (інформація про виконавця);
- 5) equalizer (збережені набори еквалайзера);
- 6) favorites (вподобані композиції);
- 7) lyrics (слова композиції);
- 8) song\_stats (кількість разів відтворення композиції та час відтворення).

Представлення бази даних можна побачити на рис. 3.

Розроблений застосунок буде корисним у використанні для людей які, часто слухають музику та бажають переглядати слова композиції. Користувач буде мати змогу завантажити слова композиції та вирішити чи зберегти її локально для того, щоб мати змогу відкрити слова знову без підключення до мережі. Схему доступних функцій для користувача зображено на рис. 4.



Рис. 4. Базові функції користувача

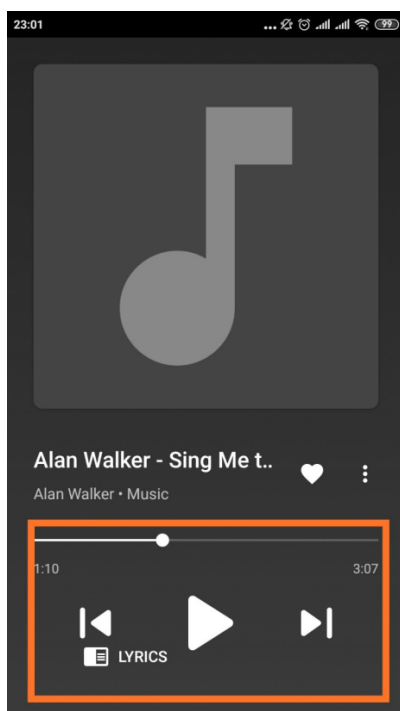


Рис. 5. Міні-плеєр

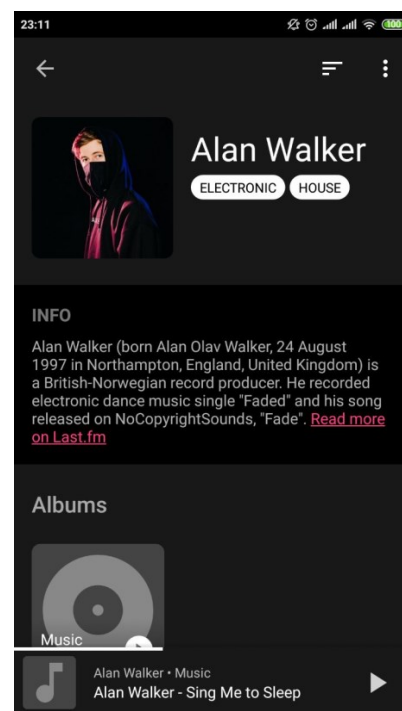


Рис. 6. Show Artist екран

Бізнес-процес «медіа-плеєр»/ Даний бізнес-процес передбачає елементи управління музичним відтворенням і включає в себе наступні функції:

- управління музикою(пауза, перемотування, наступний, попередній трек);
- додавання композиції в список вподобаних;
- перегляд інформації про виконавця;
- редагування тегів;
- шаринг.

Управління музикою – це медіа-плеєр (міні-плеєр) (рис. 5).

За допомогою функції “Додати в список вподобаних” користувач має можливість швидко знайти улюблену композицію. На екрані композиції через меню швидких дій можна вибрати наступні функції:

Show Artist (рис. 6) – для перегляду інформації про виконавця, його зображення. Також можна ознайомитись з його біографією, локально збережені альбоми та композиції.

Edit Tags (рис. 7) – для редагування метаданих композиції. Метадані можуть включати назву композиції, тривалість, назву альбому, номер диску, рік випуску, жанр, дані про виконавці і т.д. Редагування буде корисним, якщо метадані композиції пошкоджено і пошук слів був невдалим. В такому випадку можна власноруч змінити метадані та успішно знайти потрібні слова.

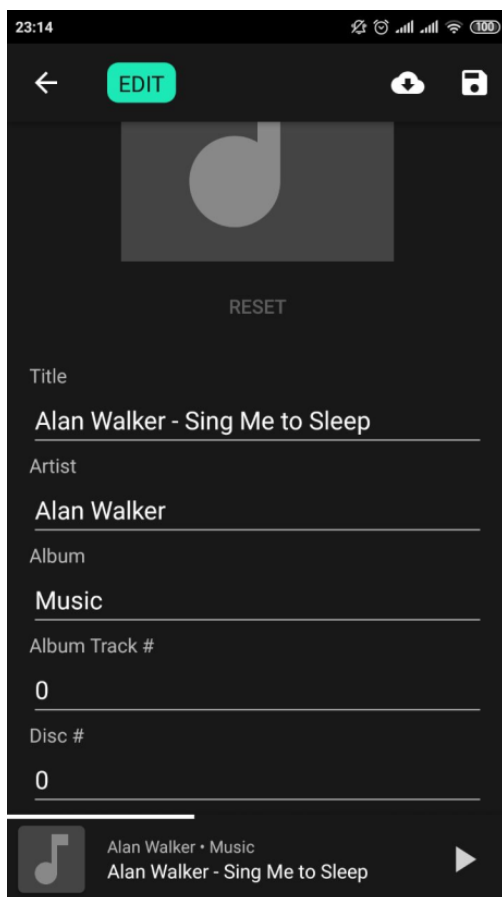


Рис. 7. Екран з метаданими

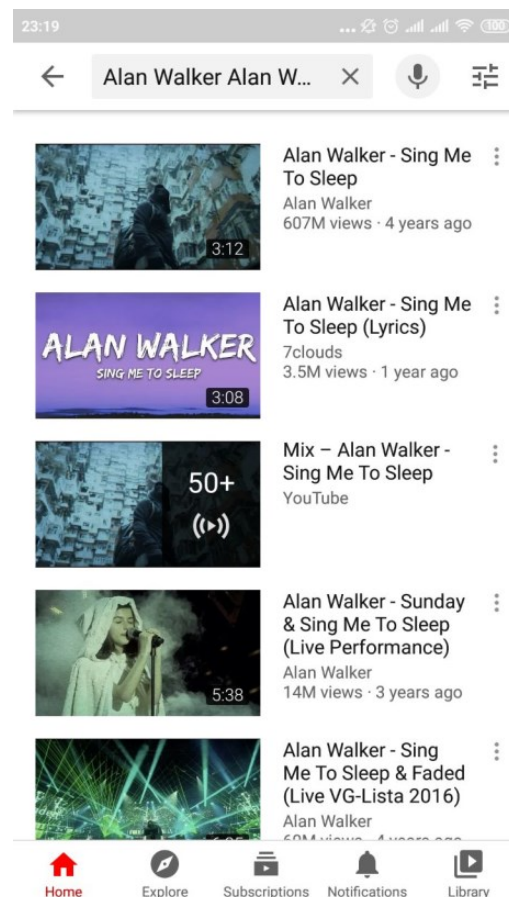


Рис. 8. Сервіс YouTube з відеокліпом композиції

View on YouTube (рис. 8) – для швидкого пошуку оригінального відеокліпу композиції. Відбувається перенаправлення на сервіс відео-хостингу YouTube.

Share – для того, щоб поділитись композицією у соц. мережах через технологію безпроводового зв'язку Bluetooth, поштовий клієнт і т.д.

### Висновок

Реалізована інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв. Програмний застосунок забезпечує завантаження слів музичної композиції на основі її метаданих з подальшим відображенням тексту на екрані мобільного пристрою. Результати пошуку та завантаження зберігаються у локальній базі даних задля забезпечення оффлайн режиму доступу. Реалізовано інтеграцію з популярним відео-хостингом YouTube, що забезпечує розширення функціональних можливостей реалізованого програмного застосунку з прослуховування аудіо творів на перегляд мультимедійного контенту. Наявний функціонал інформаційної системи дозволяє використовувати її для вивчення іноземних мов.

---

**Література**

1. Spotify [Електронний ресурс]. – Режим доступу : <https://www.spotify.com/ua/>
2. MusixMatch [Електронний ресурс]. – Режим доступу : <https://www.musixmatch.com/>
3. Google Play Music [Електронний ресурс]. – Режим доступу : <https://play.google.com/music/listen>
4. Deezer [Електронний ресурс]. – Режим доступу : <https://www.deezer.com/ru/>
5. LastFM [Електронний ресурс]. – Режим доступу : <https://www.last.fm/ru/home>
6. Genius [Електронний ресурс]. – Режим доступу : <https://genius.com/>
7. iTunes [Електронний ресурс]. – Режим доступу : <https://www.apple.com/ru/itunes/>
8. Room DB [Електронний ресурс]. – Режим доступу : <https://developer.android.com/training/data-storage/room/>

**References**

1. Spotify. URL: <https://www.spotify.com/ua/>
2. MusixMatch. URL: <https://www.musixmatch.com/>
3. Google Play Music. URL: <https://play.google.com/music/listen>
4. Deezer. URL: <https://www.deezer.com/ru/>
5. LastFM. URL: <https://www.last.fm/ru/home>
6. Genius. URL: <https://genius.com/>
7. iTunes. URL: <https://www.apple.com/ru/itunes/>
8. Room DB. URL: <https://developer.android.com/training/data-storage/room/>

Рецензія/Peer review : 19.09.2020 р.

Надрукована/Printed :06.11.2020 р.

—  
**Інформаційна технологія  
завантаження слів пісень в  
реальному часі для  
мобільних пристроїв**

# Актуальність теми



Доведено, що будь-яке навчання є ефективнішим в процесі гри чи квесту. Тоді новий матеріал сприймається легше та засвоюється краще. Наприклад, можна вивчити чи покращити рівень володіння англійською, слухаючи пісні, перекладаючи незнайомі слова, або, дивлячись фільми англійською з субтитрами. Починаючи вивчати нову мову, мозок людини має справу з новими йому звуками, які невластиві нашому мовленнєвому апарату. Прослуховуючи пісні англійською, можна мимоволі, не помічаючи цього, звикнути до нових слів та їх правильної вимови. Саме у такий спосіб людина ефективно вчиться розставляти смислові акценти та вловлює ритм нової мови.

# Мета і завдання



Для досягнення поставленої мети визначені наступні задачі досліджень:

- провести аналіз існуючих методів, технологій та рішень завантаження слів пісень в реальному часі для мобільних пристроїв;
- удосконалення методики завантаження слів пісень у напрямку отримання максимально повних і достовірних даних;
- розробити інформаційну технологію завантаження слів пісень в реальному часі для мобільних пристроїв;
- виконати експериментальну перевірку інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв.

# Наукова новизна одержаних результатів



В результаті проведеної роботи були отримані такі результати:

- удосконалено існуючі методи завантаження слів пісень в реальному часі для мобільних пристроїв у напрямку отримання максимально повних і достовірних даних.

# Практична цінність одержаних результатів



В результаті виконання дипломної роботи магістра розроблено відповідне програмне забезпечення, яке підтвердило вірність запропонованих положень. Застосування інформаційної технології дає можливість завантаження слів пісень в реальному часі для мобільних пристроїв у напрямку отримання максимально повних і достовірних даних.

# Публікація



Основні наукові та практичні результати опубліковані в наукових виданнях МОН України:

- публікація на тему “Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв” в науковому журналі “Вісник Хмельницького національного університету”.

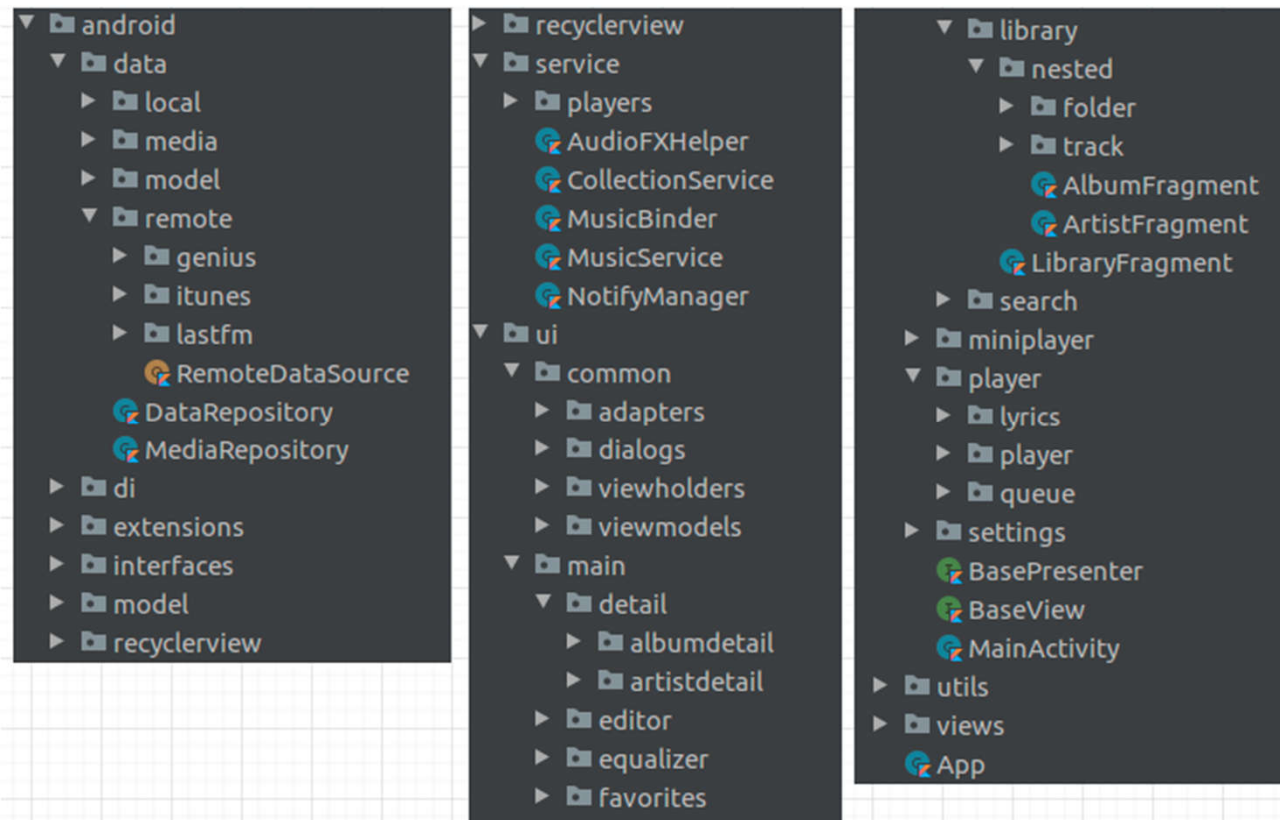
За темою дипломної роботи магістра автором виконано одну наукову публікацію:

*П.О. Грицишин, О.А. Пасічник, Т.К. Скрипник. Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв // Вісник Хмельницького національного університету. 2020*

# Структура БД

Name	Type	Schema
▼ Tables (10)		
▼ album_stats		CREATE TABLE `album_stats` (`albumid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL, PRIMARY KEY(`albumid`))
albumid	INTEGER	"albumid" INTEGER NOT NULL
playcount	INTEGER	"playcount" INTEGER NOT NULL
playtime	INTEGER	"playtime" INTEGER NOT NULL
▼ android_metadata		CREATE TABLE android_metadata (locale TEXT)
locale	TEXT	"locale" TEXT
▼ artist_stats		CREATE TABLE `artist_stats` (`artistid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL, PRIMARY KEY(`artistid`))
artistid	INTEGER	"artistid" INTEGER NOT NULL
playcount	INTEGER	"playcount" INTEGER NOT NULL
playtime	INTEGER	"playtime" INTEGER NOT NULL
▼ bio		CREATE TABLE `bio` (`artistid` INTEGER NOT NULL, `bio` TEXT, `tags` TEXT, PRIMARY KEY(`artistid`))
artistid	INTEGER	"artistid" INTEGER NOT NULL
bio	TEXT	"bio" TEXT
tags	TEXT	"tags" TEXT
▼ equalizer		CREATE TABLE `equalizer` (`eqid` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `eqname` TEXT NOT NULL)
eqid	INTEGER	"eqid" INTEGER NOT NULL
eqname	TEXT	"eqname" TEXT NOT NULL
▼ favorites		CREATE TABLE `favorites` (`fav` INTEGER NOT NULL, PRIMARY KEY(`fav`))
fav	INTEGER	"fav" INTEGER NOT NULL
▼ lyrics		CREATE TABLE `lyrics` (`songid` INTEGER NOT NULL, `lyric` TEXT NOT NULL, PRIMARY KEY(`songid`))
songid	INTEGER	"songid" INTEGER NOT NULL
lyric	TEXT	"lyric" TEXT NOT NULL
▶ room_master_table		CREATE TABLE room_master_table (id INTEGER PRIMARY KEY,identity_hash TEXT)
▼ song_stats		CREATE TABLE `song_stats` (`songid` INTEGER NOT NULL, `playcount` INTEGER NOT NULL, `playtime` INTEGER NOT NULL, PRIMARY KEY(`songid`))
songid	INTEGER	"songid" INTEGER NOT NULL
playcount	INTEGER	"playcount" INTEGER NOT NULL
playtime	INTEGER	"playtime" INTEGER NOT NULL
▼ sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
Indices (0)		
Views (0)		
Triggers (0)		

# Загальна ієрархія пакетів

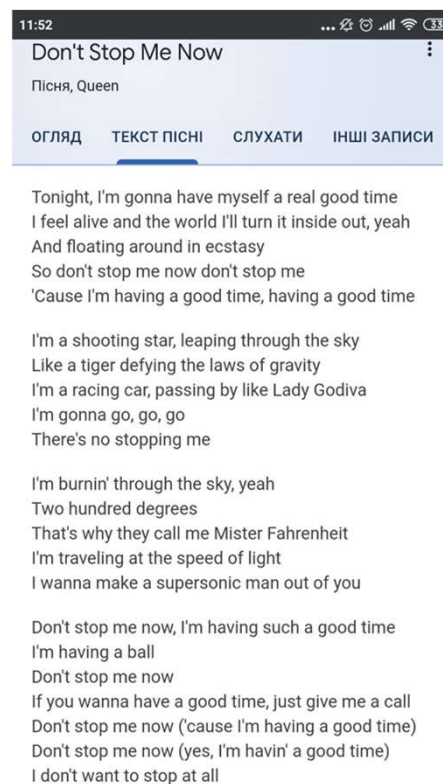
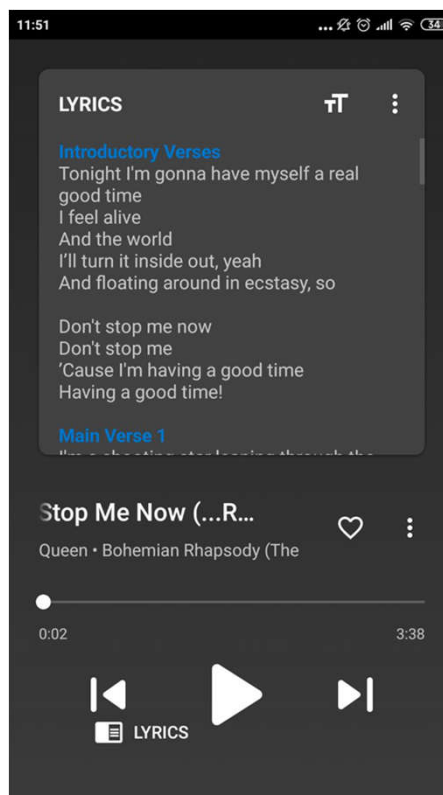


## Загальні доступні функції



- прослуховування пісень з пам'яті пристрою;
- автоматичний пошук слів пісні;
- ручний пошук слів пісні;
- ручне завантаження слів;
- завантаження усіх доступних слів у локальну бд;
- виправлення тегів;
- створення черг/списку улюблених;
- перегляд інформації про виконавця, його пісень на пристрої.

# Результати роботи



## Загальні висновки



В результаті виконаної дипломної роботи магістра було реалізовано інформаційну технологію завантаження слів пісень в реальному часі для мобільних пристроїв з використанням загальнодоступних мережевих аудіо-хостингів. Було визначено структуру інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв. Описано процедури забезпечення функціональності інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв, потоки даних та інформації між модулями системи з представленням інформаційних діаграм. Проведена експериментальна апробація результатів роботи реалізованої інформаційної технології завантаження слів пісень в реальному часі для мобільних пристроїв підтвердила її адекватність.

# Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 3.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилоч в документах: 8%**

ID: 81736 Назва: Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв Додано в БД: 2020-11-30 Автора: Грицишин Павло Олегович Керівники: Пасічник О.А. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	31053	293	1410 (5%)	19 (6%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РІШЕННЯ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв

Автор: Грицишин П.О.

Спеціальність: 122 Комп'ютерні науки

Науковий керівник: к.т.н., доцент Пасічник О.А.


Після аналізу звіту подібності зроблено такий висновок:

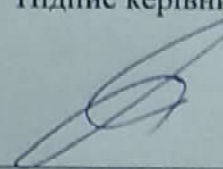
№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	-
3	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-
4	Інше:	-

Підтвердження: Виявленні запозичення не є плагіатом т.я. розміщені в розділах, які не описують безпосередньо авторське дослідження, складають 2,41% та мають посилання на приведений список літературних джерел

02.12.2020

Дата

  
Підпис керівника

  
Підпис завідувача кафедри

**ВІДГУК ОПОНЕНТА**  
**на дипломну роботу магістра**

Магістра гр. КНМ-19-1 Грицишина Павла Олеговича

На тему: Інформаційна технологія завантаження слів пісень в реальному часі для мобільних пристроїв.

1. Актуальність і значення теми

Сучасний розвиток техніки та технологій виводить на якісно новий рівень практично всі сфери життєдіяльності людини. На заміну та допомогу традиційним підручникам приходять технічні пристрої та сучасні технології. Тобто, вивчення іноземної мови може відбуватися за допомогою смартфона або планшета, які є під рукою практично завжди. Одним із методологічних підходів вивчення іноземних мов, який довів свою високу ефективність, є спілкування з носіями мови. Певним віртуальним заміником такого носія може слугувати пісенний виконавець. Музика дуже корисна при вивченні іноземних мов, а особливо в освоєнні граматики, збільшенні словникового запасу і постановці вимови. Саме у такий спосіб людина ефективно вчиться розставляти смислові акценти та вловлює ритм нової мови.

2. Оцінка якості та достовірності проведених досліджень.

Отримані результати інформаційної технології є достовірними та співпадають з результатами з надійних інтернет-сервісів.

3. Оцінка запропонованих заходів та пропозицій, практичної цінності та ефективності.

Проведені дослідження представляють науково-технічну цінність, є ефективним дослідженням в галузі лінгвістики, що можна використати з метою підвищення рівня знань іноземної мови.

4. Загальний висновок та оцінка

Робота виконана в повному обсязі. Пояснювальна записка оформлена в відповідності з нормами. Відмічені недоліки не знижують цінності дипломної роботи. Виконана робота відповідає вимогам вищої школи і вимогам, що пред'являються до освітньо-кваліфікаційного рівня «магістр», а її автор Грицишин П.О. заслуговує присвоєння кваліфікації магістра з комп'ютерних наук та інформаційних технологій.

Робота заслуговує на оцінку «добре».

Опонент Магдальона В.В., д.т.н., проф.