

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
Кафедра інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської кваліфікаційної роботи  
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА MUSIC-BOT ДЛЯ СОРТУВАННЯ КОМПОЗИЦІЙ  
МОВОЮ JAVA»**

Виконав: студент 4 курсу, групи ПД-44

---

спеціальності 121 Інженерія програмного  
забезпечення

---

(шифр і назва спеціальності)

Горкуша В.С.

---

(прізвище та ініціали)

Керівник

Шевченко С.М.

---

(прізвище та ініціали)

Рецензент

---

(прізвище та ініціали)

Нормоконтроль

---

(прізвище та ініціали)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного

забезпечення

\_\_\_\_\_ О.В. Негоденко

« \_\_\_\_ » \_\_\_\_\_ 2021 року

**З А В Д А Н Н Я**  
**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**  
**ГОРКУШІ ВІКТОРУ СЕРГІЙОВИЧУ**

1. Тема роботи: «Розробка music-bot для сортування композицій мовою java»

Керівник роботи Шевченко Світлана Миколаївна, доцент, кандидат технічних наук

затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки IntelliJ IDEA.

3.2. Мова програмування Java.

3.3. База даних H2.

3.4. Документація по розробці.

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

- 4.1. Бот повинен бути безкоштовним;
- 4.2. Бот повинен сортувати композиції за виконавцем та жанром;
- 4.3. Повинна бути функція створення власного списку відтворення;
- 4.4. Повинна бути функція завантаження композицій;
- 4.5. Повинна бути функція відтворення музики у фоновому режимі.

5. Перелік графічного матеріалу:

- 5.1. Аналоги.
- 5.2. Програмні засоби реалізації.
- 5.3. Демонстрація інтерфейсу.
- 5.4. Діаграма користувача.
- 5.5. Опис тригерів.

6. Дата видачі завдання 19.04.2021

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз існуючих прототипів	22.04.21 – 23.04.21	
3	Дослідження програмних засобів	23.04.21 – 01.05.21	
4	Моделювання об'єкту проектування	01.05.21 – 05.05.21	
5	Розробка функціоналу бота	05.05.21 – 07.05.21	
6	Вступ, висновки, реферат	07.05.21 – 08.05.21	
7	Розробка презентації застосунку	08.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент

Керівник роботи





## РЕФЕРАТ

Текстова частина бакалаврської роботи .

Ключеві слова: Telegram, music-bot, композиції, сортування, Java, месенджер, розробка, структура, аналіз, Bot API

*Об'єкт дослідження* – особливості розробки ботів на основі месенджера Telegram.

*Предмет дослідження* – розробка music-bot для сортування композицій мовою Java.

*Мета роботи* – доповнити можливості месенджера Telegram за допомогою music-bot для сортування композицій.

*Методи дослідження* – методи теорії інформації, методи структурного аналізу і проектування, методи розробки програмного забезпечення, методи тестування, валідації та верифікації програмного забезпечення.

Наукова новизна даної роботи полягає в наступному:

В роботі виконано аналіз існуючих аналогів. Встановлено переваги та недоліки існуючих аналогів. В результаті аналізу було визначено основні недоліки аналогів та визначено методи їх ліквідації. Проаналізовано можливості середовища розробки IntelliJ IDEA. Розроблено алгоритм роботи бота, концепцію взаємодії з користувачем та методи представлення інформації.

Галузь використання – бота може використовувати будь який користувач месенджера, який знайде його по імені або логіну.

## ЗМІСТ

ВСТУП.....	8
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ .....	10
1.1 Огляд аналогів .....	10
1.1.1 Spotify.....	10
1.1.2 YouTube Music.....	10
1.1.3 Apple Music .....	11
1.1.4 Порівняльна таблиця аналогів .....	12
1.2 Обґрунтування вибору месенджера Telegram у якості платформи для прослуховування та сортування музики .....	14
1.2.1 Огляд месенджера Telegram.....	14
1.2.2 Музика Telegram: основні функції .....	17
1.2.3 Як знайти музику в Telegram.....	18
2 ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ .....	19
2.1 Порівняння мов програмування Java, Python та C++ .....	19
2.1.1 Мова програмування Java .....	19
2.1.2 Мова програмування Python .....	24
2.2 Інтегроване середовище розробки IntelliJ IDEA .....	26
2.3 База даних H2 .....	32
2.4 Документація до бібліотек для створення Telegram бота .....	33
3 РОЗРОБКА MUSIC-БОТ .....	38
3.1 Реєстрація бота в системі Telegram Bot API .....	38
3.2 Розробка структури та алгоритму бота .....	39
3.3 Структура та реалізація бази даних.....	43
3.4 Тестування розробленого бота .....	44
ВИСНОВКИ .....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	52

## ВСТУП

У наш час, велику кількість задач по всьому світу, виконують люди всього за один день. Вирішення деяких задач потребують саме людського фактору присутності: навичок та вмінь. Але деяку частину задач можна розв'язувати не використовуючи людських можливостей. Так як ми проживаємо у вік технологій, розвиток яких відбувається дуже швидко, було створено деяку одиницю під назвою – бот. Саме тому людство хоче все більше автоматизувати процеси, винайти безпілотні автомобілі, удосконалити голосових помічників та багато іншого. Що саме являє собою бот? Поняття бот, скорочено від «робот» – це програма, що імітує діяльність людини, виконує автоматично та/або за розкладом якісь дії, що задані користувачем або алгоритмом. Зазвичай ботів використовують для виконання якихось рутинних завдань з досить великою швидкістю. Також їх можна застосувати тоді, коли потрібна реакція, яка буде набагато швидша ніж у людини. З даним видом програми ми стикаємось майже кожен день. Боти досить тісно увійшли в наше життя, а саме, чат-боти при замовленні товарів або послуг, боти в месенджерах для повідомлення про події, або про відображення будь-якої інформації, у іграх, де бот виступає іншим гравцем. Ще за допомогою ботів можна розширити, або вдосконалити функціонал месенджера. На даний час вже існує безліч платформ для створення даних програм, наприклад: «Aimylogic», «BotKits», «Botmother» та інші.

**Актуальність обраної теми.** Актуальність дипломної роботи полягає в тому, що на сьогоднішній день месенджери набувають все більшої популярності. В значній мірі цьому сприяє розвиток ботів, адже автоматизація та використання ботів є одним із способів спрощення та покращення умов для життя людей. Завдяки цьому можна зберегти не тільки час, тому що для ботів немає обідніх перерв і неробочих годин. Користувачі можуть отримати допомогу або результат певної роботи в будь-який час доби, тижня і року — саме тоді, коли вони того потребують, поки їхні питання ще актуальні та інтерес не зник. Також за допомогою ботів розробники можуть значно покращити, або навіть розширити функціонал



месенджера. На разі все більше користувачів почали використовувати месенджер Telegram для прослуховування музики. Але робити це за допомогою базових функцій досить незручно. Враховуючи все вище сказане дана тема є актуальною, і буде актуальна ще досить довго, тому варто розглядати її в більш глибокому аспекті, оскільки тут є досить великий спектр можливостей починаючи від простого інформування за заданим алгоритмом до використання штучного інтелекту чи нейронних мереж.

**Мета написання роботи:** доповнити можливості месенджера Telegram за допомогою music-bot для сортування композицій.

**Об'єкт наукового дослідження:** особливості розробки ботів на основі месенджера Telegram.

**Предмет наукового дослідження:** розробка music-bot для сортування композицій мовою Java.

Задля досягнення поставленої мети, слід виконати низку **завдань**, а саме:

1. Проаналізувати обрану предметну область;
2. Порівняти аналоги;
3. Обрати технології і середовища розробки;
4. Розробити music-bot для сортування композицій;
5. Протестувати розробленого бота.

Під час написання роботи було використано такі **методи дослідження**:

- теоретичні: аналіз відповідної літератури, яка містить відомості про розробку та галузі використання ботів;
- емпіричні: проведення експерименту, а саме апробація створеного music – bot серед певних користувачів та винесення переваг і недоліків на основі аналізу проведеної апробації створеного продукту.

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд аналогів

### 1.1.1 Spotify

Spotify — це сервіс для прослуховування цифрової музики, подкастів та відео [17]. Він дає доступ до контенту від композиторів всього світу. Основні функції Spotify, такі як відтворення музики, отримання рекомендацій та створення колекцій безкоштовні. На рисунку 1 зображено інтерфейс Spotify.

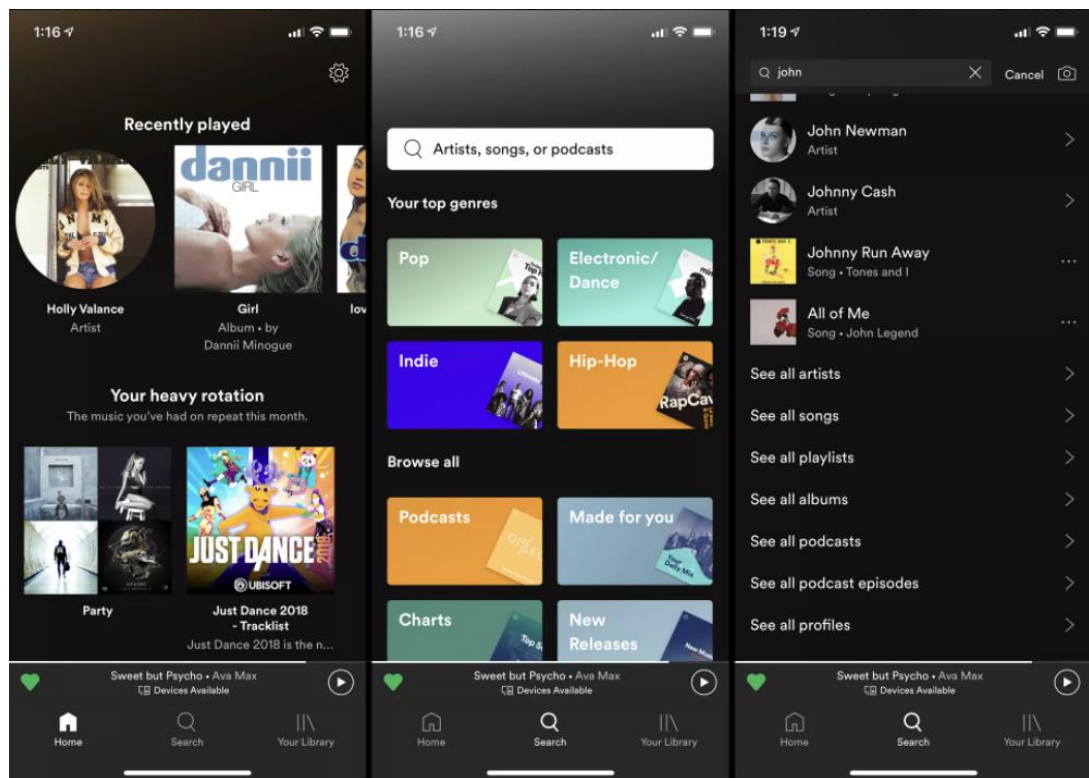


Рисунок 1 – Інтерфейс Spotify

### 1.1.2 YouTube Music

YouTube Music – це сервіс від компанії Google для прослуховування музики та відтворення музичних кліпів, орієнтований на користувачів які слухають музику у YouTube [18]. Він дозволяє відсортувати пісні за альбомами, жанрами, виконавцями та створеними списками відтворення. Безкоштовна версія дозволяє знаходити, сортувати, зберігати та відтворювати композиції, але лише на

ввімкненому пристрої. Для програвання музики в фоновому режимі, або офлайн потрібно купувати підписку. На рисунку 2 зображено інтерфейс YouTube Music.

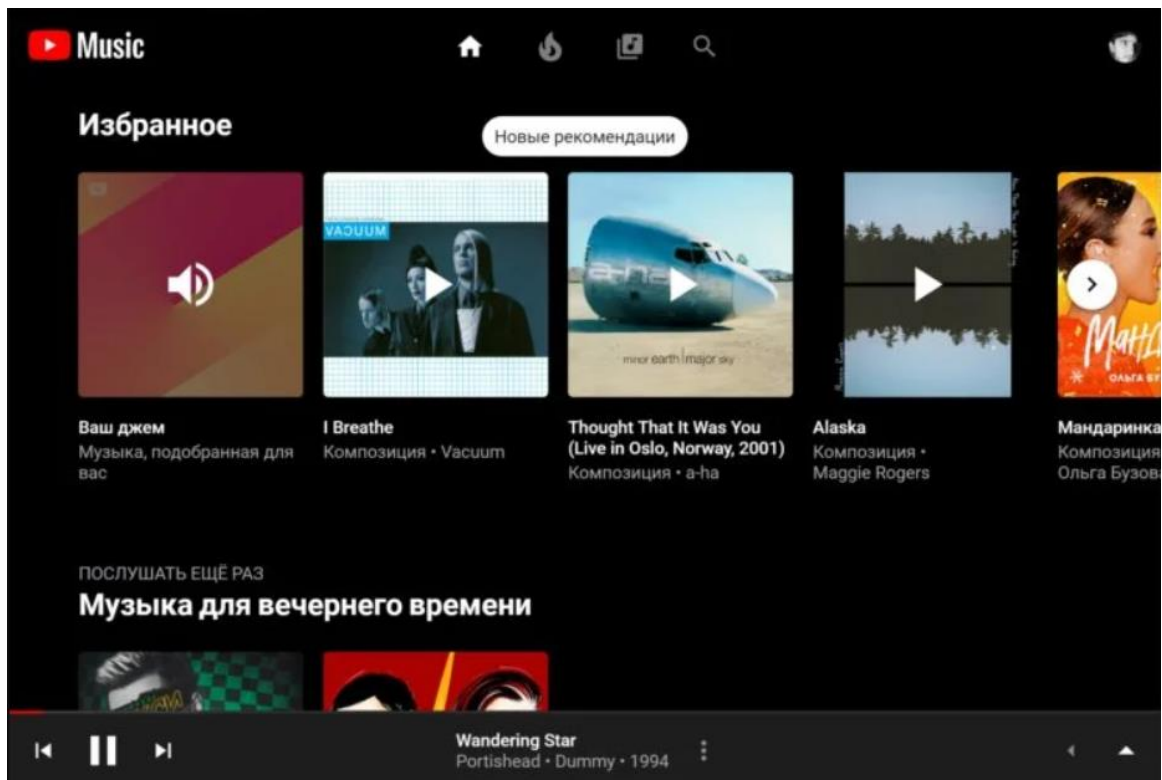


Рисунок 2 – Інтерфейс YouTube Music

### 1.1.3 Apple Music

Apple Music – це сервіс для прослуховування потокової музики, розроблений компанією Apple тісно інтегрований в їхню екосистему [19]. Він дозволяє завантажувати улюблені композиції та відтворювати їх у режимі офлайн, переглядати тексти пісень, прослуховувати кураторські списки відтворення від редакторів та багато іншого. Все це на додаток до ексклюзивного та оригінального контенту. На рисунку 3 зображено інтерфейс Apple Music.

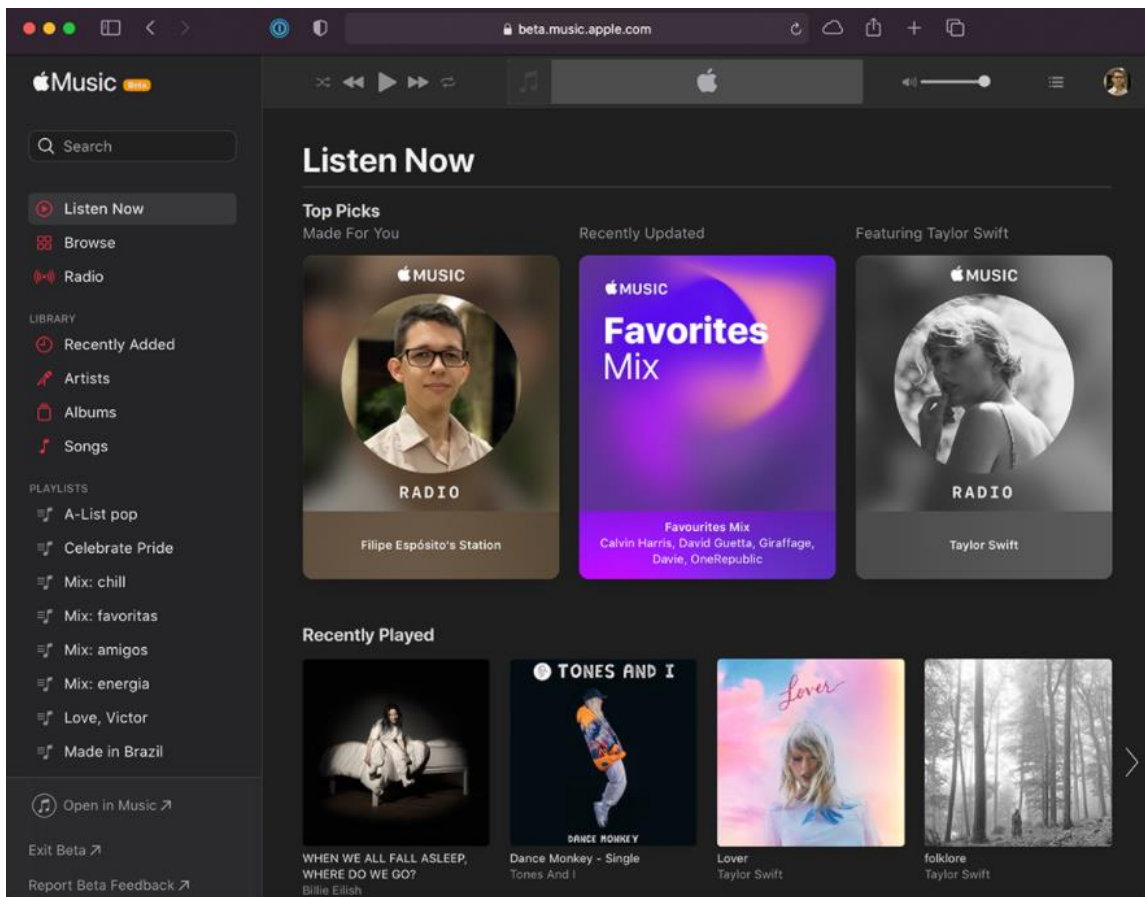


Рисунок 3 – Інтерфейс Apple Music

### 1.1.4 Порівняльна таблиця аналогів

Розглянемо основні переваги та недоліки сервісів Spotify, YouTube Music та Apple Music.

Таблиця 1 – Порівняння сервісів [16]

Назва сервісу	Переваги	Недоліки
Spotify	<ol style="list-style-type: none"> <li>1. Пробна версія повністю безкоштовна;</li> <li>2. Автоматично сформовані списки відтворення на основі ваших вподобань;</li> <li>3. Збір та відображення статистики.</li> </ol>	<ol style="list-style-type: none"> <li>1. Для прослуховування музики офлайн потрібно купувати підписку;</li> <li>2. Реклама;</li> <li>3. Неможливо створювати власні плейлісти.</li> </ol>

Назва сервісу	Переваги	Недоліки
YouTube Music	<ol style="list-style-type: none"> <li>1. Інтелектуальний пошук пісень, не тільки по назві, але й за частиною тексту;</li> <li>2. Рекомендації на основі попередніх пошукових запитів і переглядів;</li> <li>3. Можливість створювати власні списки відтворення.</li> </ol>	<ol style="list-style-type: none"> <li>1. Не завжди висока якість звуку;</li> <li>2. Реклама;</li> <li>3. Без підписки неможливо відтворювати композиції офлайн, або у фоновому режимі.</li> </ol>
Apple Music	<ol style="list-style-type: none"> <li>1. Приємний та зрозумілий інтерфейс;</li> <li>2. Тривалість пробного періоду три місяці (безкоштовно);</li> <li>3. Сервіс інтегрований в екосистему Apple.</li> </ol>	<ol style="list-style-type: none"> <li>1. Не завжди коректна робота рекомендацій;</li> <li>2. Сервісом не можливо користуватися без підписки;</li> <li>3. Не можливо додавати власні списки відтворення.</li> </ol>

Проаналізувавши переваги та недоліки аналогів для бота були створені наступні вимоги:

1. Бот повинен бути безкоштовним;
2. Бот повинен сортувати композиції за виконавцем та жанром;
3. Повинна бути функція створення власного списку відтворення;
4. Повинна бути функція завантаження композицій (щоб прослуховувати їх офлайн);
5. Повинна бути функція відтворення музики у фоновому режимі.

Вирішення ряду поставлених задач було здійснено за допомогою месенджера Telegram.

## 1.2 Обґрунтування вибору месенджера Telegram у якості платформи для прослуховування та сортування музики

### 1.2.1 Огляд месенджера Telegram

Telegram – це програма обміну повідомленнями, орієнтована на швидкість та безпеку, надзвичайно швидка, проста та безкоштовна. Telegram можна використовувати на всіх пристроях користувача одночасно - повідомлення будуть синхронізуватися на будь-якій кількості телефонів, планшетів або комп'ютерів. Щомісяця Telegram має понад 500 мільйонів активних користувачів і є однією з 10 найбільш завантажуваних програм у світі [1].

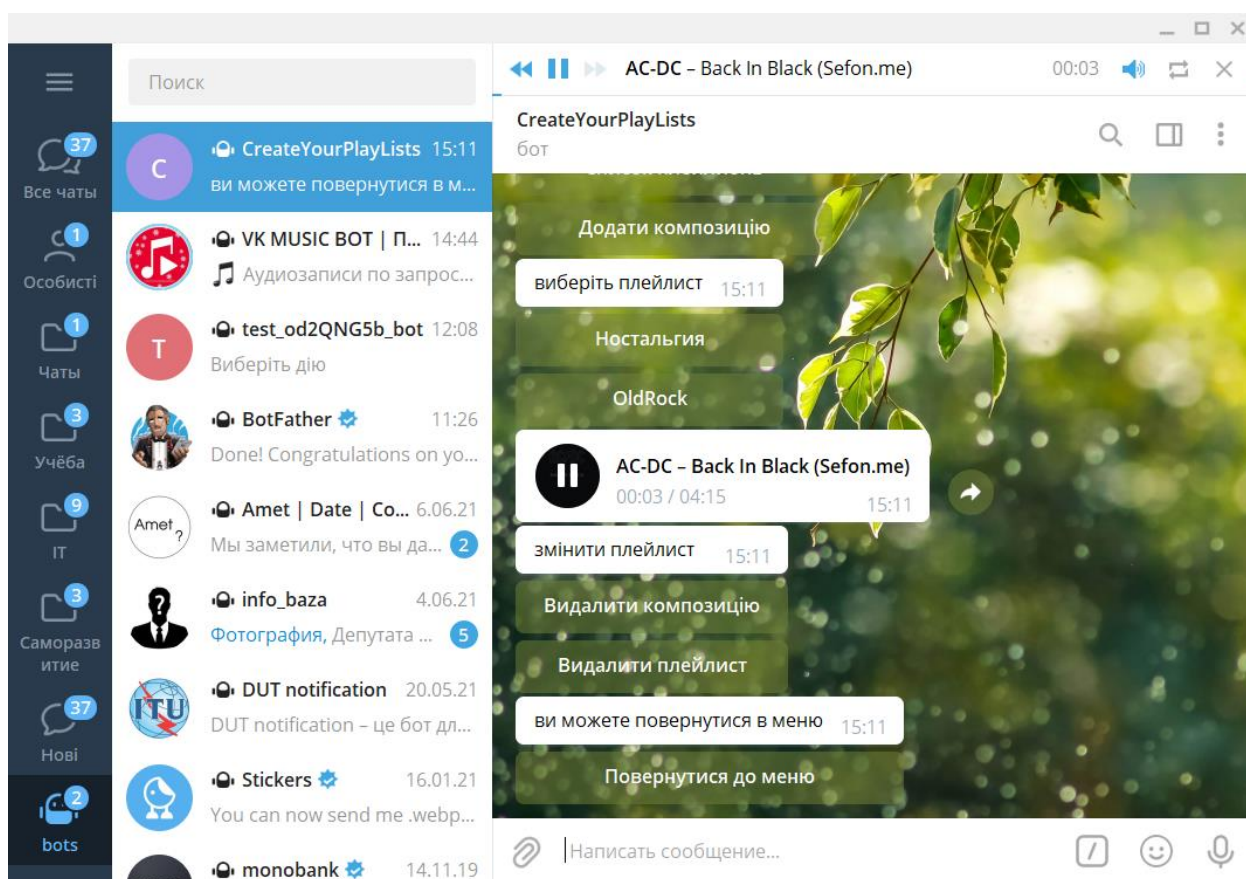


Рисунок 4 – Інтерфейс Telegram

За допомогою Telegram можна надсилати повідомлення, фотографії, відео та файли будь-якого типу (doc, zip, mp3 тощо), а також створювати групи до 200 000 людей або канали для трансляції для необмеженої аудиторії. Також знаходити

людей за їхніми іменами користувачів. Як результат, Telegram схожий на SMS та електронну пошту - він може подбати про особисті або ділові потреби щодо обміну повідомленнями. На додаток до цього, месенджер підтримує наскрізні зашифровані відеодзвінки, а також голосові чати в групах для тисяч учасників.

Telegram використовують для швидкого та надійного надсилання повідомлень, або здійснення аудіо- та відеодзвінків. Обмін файлами з різними розширеннями можна здійснювати повністю конфіденційно як між двома людьми так і між групами користувачів.

Оскільки групи Telegram можуть налічувати до 200 000 учасників, наявна підтримка згадувань та хештегів, які допомагають підтримувати порядок та ефективність спілкування у великих громадах. У каналах та групах можна призначати адміністраторів із спеціальними інструментами, які допоможуть власним чатам зберігати порядок, або вирішувати різні технічні проблеми. До громадських груп може приєднуватися кожен бажаючий, адже це потужна платформа для дискусій та збору відгуків.

Обмін фотографіями. Telegram має анімований пошук у GIF-файлах, сучасний редактор фотографій та відкриту платформу. Більше того, збережені медіафайли не будуть займати місце на пристрої. Завдяки хмарній підтримці та опціям управління кешем Telegram займає найменшу кількість простору на пристрої.

Політика конфіденційності. Для безпеки передачі файлів слід ознайомитися з розширеними налаштуваннями та політикою конфіденційності. Також Telegram містить функцію секретного чату для пристроїв із самознищувальними повідомленнями, фотографіями та відео. Та для більшої впевненості можна заблокувати програму додатковим паролем.

Але найприємнішою “фішкою” Telegram для більшості користувачів є перегляд відео та прослуховування музики. Адже на сьогоднішній день однією з потреб людини є саме музика. Кожна людина зникає з самого народження чути музику. У кожного є улюблений стиль музики, музика, яка розслабляє і та, яка напружує. Дивлячись на різні ситуації можна помітити, що роль музики в житті



людини досить велика, музика може впливати на настрій та поведінку.

Виявляється, кожна людина безперервно, на генетичному рівні пов'язана з музикою, вірніше зі звуками. Ці звуки можуть виходити від птахів, тварин та самої природи. Сама людина, не помічаючи за собою, розслабляється і оздоровлюється за рахунок цих звуків. Адже голоси тварин і птахів зафіксовані вже давно в підсвідомості, як певний подразник. Він може бути пов'язаний, як з позитивними, так і з негативними емоціями.

Як виявили психологи, мелодія з ритмом 60 ударів на хвилину, діє на людину як медитація, вона цілком здатна відвернути будь-яку людину від будь-якої проблеми, сповільнюючи мозкову діяльність. Якщо слухати таку музику, активно поліпшується пам'ять, працездатність, спокій і впевненість у собі [2].

Все пояснюється тим, що людський мозок просто припиняє зайве вироблення енергії, яка йде на ті самі негативні емоції, які заважають правильно думати, збивають з пантелику і змушують зайвий раз нервувати. Так само, прослуховування класичної музики сприяє запам'ятовуванню інформації. У боротьбі з безсонними ночами може врятувати «Сумний вальс» Сібеліуса, а п'єси Чайковського просто творять чудеса.

Але так як усі люди різні, кожен має свій індивідуальний музичний смак. Хтось любить поп-музику, а хтось реп. Інші ж люди можуть захоплюватися класикою, або джазом. Саме тому для зручного пошуку вподобаного треку, композиції можна сортувати за: авторами, жанрами, або створювати власні списки відтворення. Реалізувати таке сортування можна за допомогою music-боту в Telegram. На основі вищевказаних тверджень буде розроблено music-бот для сортування композицій.

Так як соціальні мережі вводять обмеження на безкоштовне прослуховування музики з мобільних пристроїв, також можна підхопити вірус на сайтах для безкоштовного завантаження треків, а підписки на Apple Music, Google Music та схожі сервіси, в основному, платні, Telegram - надає своїм користувачам можливість слухати музику без будь-яких обмежень.



## 1.2.2 Музика Telegram: основні функції

Для того щоб продемонструвати зручність використання Telegram для прослуховування та пошуку музики, буде розглянуто програвач Telegram для прослуховування музики та його можливості. Для цього потрібно перейти на спеціально створений канал [@cctracks](https://t.me/cctracks).

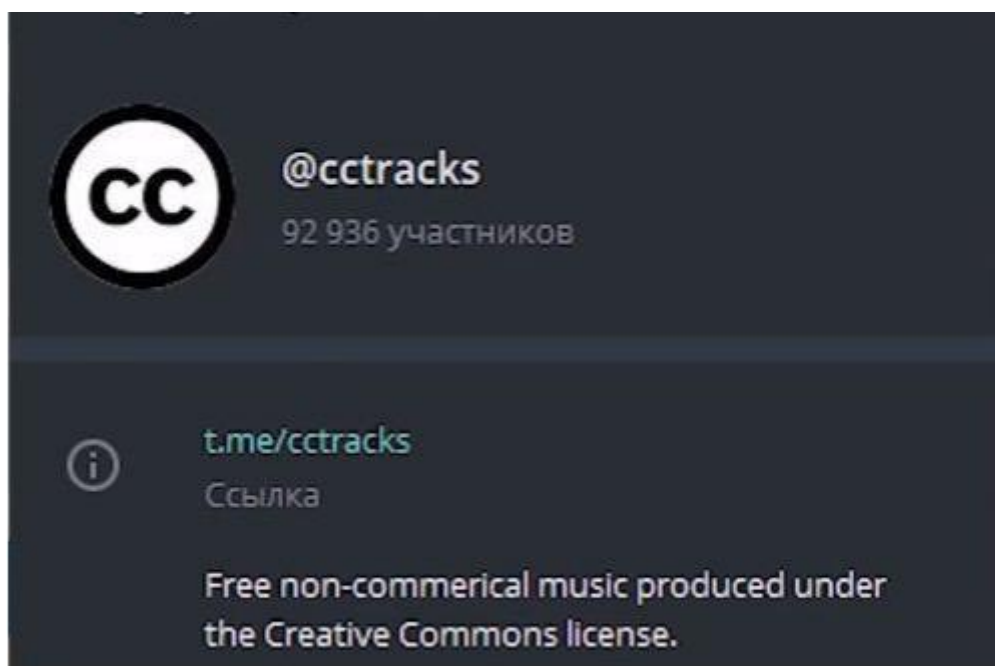
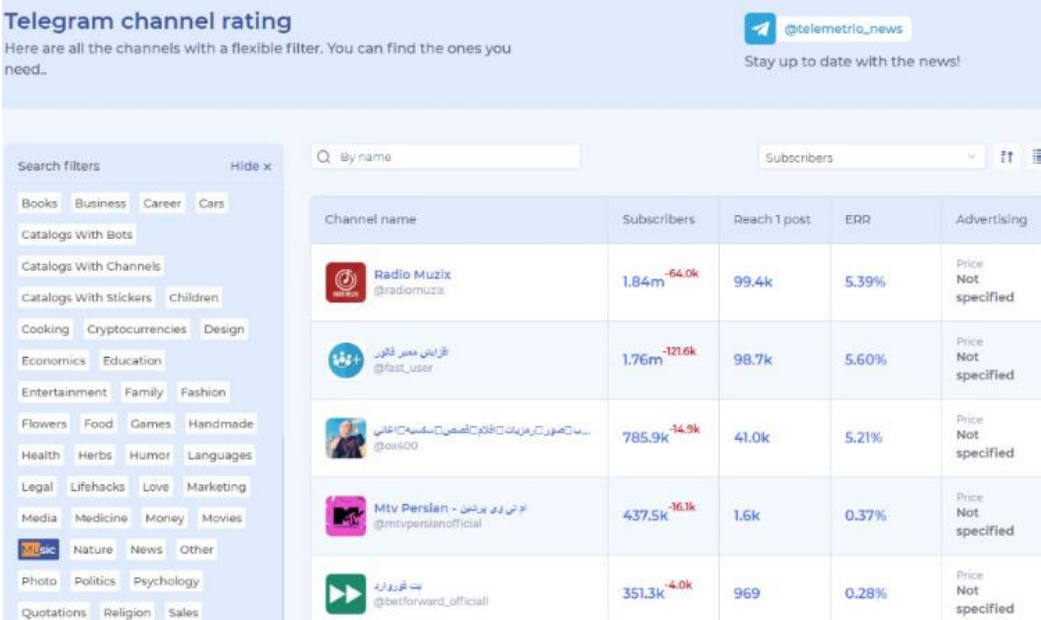


Рисунок 5 – Музичний канал @Cctracks [1]

Після натиснення на відповідну композицію можна побачити не тільки назву треку, а і виконавця, який грає зараз. Під час відтворення пісні доріжка автоматично завантажується в кеш, тобто під час наступного відтворення Інтернет не потрібен. Пісня буде збережена в пам'яті телефону. У цьому величезний плюс месенджера. Використання пам'яті телефону також можна контролювати, адже завдяки спеціальним налаштуванням можна зазначити скільки файлів буде зберігатися в месенджері, а також є можливість очистити кеш. Якщо час від часу користувач сам очищає кеш Telegram, і потрібно уникнути очищення завантажених композицій, є можливість ВИДАЛИТИ галочку для очищення Музика. Приємним бонусом для власників Android є те, що можна додати завантажену доріжку до стандартного додатку «Музика», чого, на жаль, не можна зробити в iOS [3].

### 1.2.3 Як знайти музику в Telegram

Знайти канал або бота в Telegram для прослуховування музики дуже просто. Потрібно знайти сайт "telemetr.io" в музичному розділі і вибрати канали, які подобаються, наприклад за кількістю передплатників.



The screenshot shows the 'Telegram channel rating' website. It features a search bar and a list of channels with the following data:

Channel name	Subscribers	Reach 1 post	ERR	Advertising
Radio Muzix (@radiomuzix)	1.84m <sup>-64.0k</sup>	99.4k	5.39%	Price Not specified
آر ای بی موزیک (@fast_user)	1.76m <sup>-121.6k</sup>	98.7k	5.60%	Price Not specified
بیت فورورڈ (@bex600)	785.9k <sup>-34.9k</sup>	41.0k	5.21%	Price Not specified
Mtv Persian - آهنگ روز پرشین (@mtvpersianofficial)	437.5k <sup>-16.1k</sup>	1.6k	0.37%	Price Not specified
بیت فورورڈ (@bitforward_official)	351.3k <sup>-4.0k</sup>	969	0.28%	Price Not specified

Рисунок 6 – Сайт Telegram channel rating [1]

Кілька разів на день адміністратори таких каналів завантажують нові композиції та музику до Telegram, тому пропустити музичні новини стало майже неможливо.

А якщо потребою є знайти конкретного виконавця чи пісню, можна натиснути на назву каналу та вибрати кнопку “Пошук”.

Окрім музичних каналів, існують також боти для пошуку музики. Але потрібно звернути увагу на те, що більшість із них не працюють для пристроїв iOS та MacOS. Але Android та версія для ПК легко підтримуються.

Для отримання бажаного результату потрібно ввести трек або виконавця в пошук.

## 2 ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ

### 2.1 Порівняння мов програмування Java, Python та C++

#### 2.1.1 Мова програмування Java

Мова програмування Java була розроблена Sun Microsystems на початку 1990-х років [4]. Хоча в основному вона використовується для програм підключених до Інтернету, Java є простою, ефективною мовою загального призначення. Спочатку Java була розроблена для вбудованих мережевих програм, що працюють на декількох платформах. Це портативна, об'єктно-орієнтована та інтерпретована мова.

Розроблений додаток буде однаково працювати на будь-якому комп'ютері, незалежно від апаратних можливостей або операційної системи, якщо у нього є інтерпретатор Java. Окрім портативності, ще однією з ключових переваг Java є набір функцій безпеки, які захищають ПК, на якому запущена програма, не тільки від проблем, спричинених помилковим кодом, але і від шкідливих програм (таких як віруси). Можна сміливо запускати аплет Java, завантажений з Інтернету, оскільки функції безпеки Java заважають цим типам аплетів отримувати доступ до жорсткого диска ПК або мережевих з'єднань. *Аплет*, як правило, невелика програма Java, яка вбудовується в HTML – сторінки [4].

Java можна вважати компільованою *та* інтерпретованою мовою, оскільки її вихідний код спочатку компілюється у двійковий байт-код. Цей байт-код працює на віртуальній машині Java (JVM), яка, як правило, є програмним інтерпретатором. Використання скомпільованого байт-коду дозволяє інтерпретатору (віртуальній машині) бути ефективним (і майже таким же швидким, як центральний процесор, що працює). Крім того, цей байт-код надає Java портативності: він працює на будь-якій JVM, яка правильно реалізована, незалежно від конфігурації комп'ютера або програмного забезпечення. Більшість веб-браузерів (наприклад, Microsoft Internet Explorer або Netscape Communicator) містять JVM для запуску аплетів Java.

Порівняно з C ++ (іншою об'єктно-орієнтованою мовою), код Java працює

трохи повільніше (через JVM), але він є більш портативним і має набагато кращі функції безпеки. Віртуальна машина забезпечує ізоляцію між ненадійною програмою Java та ПК, на якому запущено програмне забезпечення. Синтаксис Java схожий на C ++, але мови досить різні. Наприклад, Java не дозволяє програмістам переважувати оператор, так як це робить C ++. Крім того, Java – це *динамічна* мова, можна безпечно модифікувати програму під час її роботи, тоді як C ++ не дозволяє цього. Це особливо важливо для мережевих додатків, які не повинні зупинятися при роботі. Крім того, всі основні типи даних Java визначені заздалегідь і не залежать від платформи, тоді як деякі типи даних можуть змінюватися залежно від платформи, що використовується в C або C ++ (наприклад, тип int).

Програми Java більш структуровані, ніж еквіваленти C ++. Всі функції (або *методи* Java ) та виконувані оператори повинні знаходитись у межах класу, тоді як C ++ дозволяє визначенням функцій та рядкам коду існувати поза класами (як у програмах у стилі C). Ці обмеження, хоча часом і громіздкі, допомагають підтримувати цілісність і безпеку програм Java і змушують їх бути повністю об'єктно-орієнтованими.

Ще одна ключова особливість Java - це відкритий стандарт із загальнодоступним вихідним кодом. Sun Microsystems контролює мову Java та пов'язані з нею продукти, проте ліберальна ліцензійна політика Sun сприяла тому, що Інтернет-спільнота прийняла Java як стандарт [4].

Фреймворки додатків. Android Runtime. Хоча програми для Android написані мовою програмування Java, середовище виконання не базується на віртуальній машині Java. Відмовляючись від відповідних байт-кодів та JVM, платформа Android позбавляється вимог ліцензування Java і може адаптувати середовище виконання до обмежень та вимог мобільних платформ.

У випадку з Android віртуальна машина Dalvik - це середовище. Платформа Android включає набір основних бібліотек в Dalvik, які забезпечують більшість функціональних можливостей, доступних в основних бібліотеках мови програмування Java [5].

Як уже згадувалося раніше, кожна програма для Android існує у своєму

власному процесі Linux. Крім того, кожен процес включає власний екземпляр віртуальної машини Dalvik. Віртуальна машина Dalvik визначає власний формат файлу програми та має модель виконання на основі реєстру, на відміну від набору команд на основі стеку JVM. Реалізація системних програмних функцій, таких як доступ до пристроїв вводу-виводу, дозволи, потокові роботи та управління пам'яттю, базується на підтримці, наданій у ядрі Linux.

Для чого використовується Java? За даними Oracle, на Java працює більше 3 мільярдів пристроїв, і більшість великих компаній так чи інакше використовують Java. Java використовується у великій кількості галузей, від наукових додатків, таких як фізичні симулятори, до фінансових додатків, таких як веб-сайти електронної комерції [6].

Такі великі компанії, як Google, Microsoft та багато інших, також використовують Java для ряду цілей.

- Веб-розробка.

Хоча HTML і Javascript зазвичай використовуються для написання веб-сайтів та інших веб-додатків, Java має ряд пакетів та бібліотек, корисних для веб-розробки.

Java також добре підходить для веб-розробки на стороні сервера, тобто програм, які взаємодіють безпосередньо із сервером (тип пристрою, який надає певні функціональні можливості, наприклад, управління ресурсами, клієнтським пристроям).

- Android.

Програми Google для Android, а також операційна система Android написані з використанням Java та API Google Android API, який дуже схожий на Java Development Kit (JDK).

Хоча додатки Android використовують інший JVM, код все ще пишеться на Java.

- Фінанси та електронна комерція

Багато веб-сайтів електронної комерції, а також веб-додатки уряду, охорони здоров'я, страхування, освіти та оборони побудовані на Java, що робить

програмістів Java цінними у великій кількості галузей.

Впровадження Java робить її захищеною мовою, що важливо для веб-сайтів електронної комерції, які обробляють великі обсяги захищених даних, таких як номери кредитних карток та адреси користувачів.

У Java є ряд функцій, які роблять її безпечною мовою програмування.

Наприклад, Java обмежує програму Java середовищем виконання Java і не дозволяє їй отримувати доступ до системних ресурсів, що допомагає запобігти порушенням безпеки.

- Великі дані.

Технології великих даних, які набирають величезних темпів, також часто пишуться на Java.

Java є кращим вибором для відносно нового поля великих даних через свої громіздкі, добре керовані бібліотеки з відкритим кодом, а також масивну спільноту досвідчених програмістів Java.

Java також має ряд пакетів, корисних для обробки даних та наукових обчислень, що робить її дуже доречною для застосування в галузі науки про дані.

- Ігри.

Популярні ігри, такі як Minecraft, також створені за допомогою Java!

Ігри використовують основні концепції Java, такі як ООР (об'єктно-орієнтоване програмування) та багатопотоковість. Багатопотоковість - це одночасне виконання двох або більше процесів з найбільш ефективним використанням центрального процесора [6].

Плюси та мінуси Java. Як і всі мови кодування, Java також має свої сильні та слабкі сторони. Коли програмісти вирішують, яку мову використовувати для програми чи проєкту, вони повинні враховувати обмеження кожної мови кодування, а також унікальні переваги, які кожен може запропонувати.

Переваги Java:

- Простота. Java проста у вивченні та використанні. Мова створена для легкого розуміння, вивчення та використання для кодерів різного віку. Вона має

простіший синтаксис, ніж такі мови, як C та C++ і не вимагає знання систем пам'яті та інших обчислювальних деталей низького рівня!

- Універсальність та потужність. Якість роботи на Java походить від різних його компонентів. Java Development Kit (JDK), Java Virtual Machine (JVM) та Java Runtime Environment (JRE) утворюють трійку компонентів платформи Java для розробки та запуску програм Java. JRE, або більш конкретно, середовище виконання Java SE (Standard Edition) - це частина програмного забезпечення, призначеного для запуску коду, написаного на Java [6]. JRE містить JVM та бібліотеки класів Java, які є важливими для запуску коду Java. JDK дозволяє програмістам створювати програми Java, які можуть виконуватися та запускатися JVM та JRE.

- Об'єктно-орієнтоване програмування. Об'єктно-орієнтовані мови програмування добре підходять для написання модульного та багаторазового коду.

- Незалежність від платформи. Код Java працює на будь-якій машині, на якій є JVM, без необхідності будь-якого спеціального програмного забезпечення. Це дозволяє розробникам Java створювати програми "будь-де і будь-коли", що полегшує співпрацю і розповсюдження ідей та додатків.

- Безпека. Деталі реалізації та компіляції Java роблять її придатною для додатків, де важлива безпека [6].

Недоліки Java:

Мінуси:

- Продуктивність. У порівнянні з такими мовами як C та C++, Java вимагає великої кількості пам'яті і повільна.

- Управління пам'яттю. На Java управління пам'яттю здійснюється за допомогою збирача сміття, який обробляє відновлення пам'яті, зайнятої невикористаними даними. Цей метод збору сміття в Java впливає на продуктивність програми.

- Синтаксис. Незважаючи на те, що Java є мовою програмування високого рівня, її синтаксис все одно може здатися складним у вивченні. Хоча Java

простіша, ніж мова програмування C++, вона має більш конкретний та менш інтуїтивний синтаксис, ніж мови, такі як Python або Scratch [6].

### **2.1.2 Мова програмування Python**

Python - це широко використовувана, інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою, що використовується для програмування загального призначення. Мова була створена Гідо ван Россумом і вперше випущена 20 лютого 1991 року [7].

Однією з дивовижних особливостей Python є той факт, що розробка основи цієї мови відбувалася однією людиною. Зазвичай нові мови програмування розробляються та публікуються великими компаніями, в яких працює безліч професіоналів, і через правила авторських прав дуже важко назвати когось із людей, які беруть участь у проекті. Python - виняток.

Звичайно, ван Россум не розробляв і не покращував усі компоненти Python сам. Швидкість поширення Python по всьому світу є результатом постійної роботи тисяч (дуже часто анонімних) програмістів, тестувальників, користувачів (багато з яких не є IT-спеціалістами) та ентузіастів, але слід сказати, що саме перша ідея прийшла до однієї голови - до Гвідо.

Цілі Python. У 1999 році Гвідо ван Россум визначив свої цілі для Python:

- простота та інтуїтивність як потужний засіб для вивчення мови;
- відкритий код, де кожен може зробити свій внесок у його розвиток;
- зрозумілість коду на рівні англійської мови;
- підходить для повсякденних завдань, витрачаючи короткий термін часу

для розробки [7].

Приблизно через 20 років стає очевидно, що всі ці наміри були здійснені. Деякі джерела стверджують, що Python є третьою за популярністю мовою програмування у світі.

У будь-якому випадку, Python все ще займає високе місце в першій десятці спільнот програмування ТЮВЕ та популярності індексів мов програмування РУРЛ:



Nov 2019	Nov 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.246%	-0.50%
2	2		C	16.037%	+1.64%
3	4	▲	Python	9.842%	+2.16%
4	3	▼	C++	5.605%	-2.68%
5	6	▲	C#	4.316%	+0.36%
6	5	▼	Visual Basic .NET	4.229%	-2.28%
7	7		JavaScript	1.929%	-0.73%
8	8		PHP	1.720%	-0.66%
9	9		SQL	1.690%	-0.15%
10	12	▲	Swift	1.653%	+0.20%
11	16	▲	Ruby	1.261%	+0.17%
12	11	▼	Objective-C	1.195%	-0.28%
13	13		Delphi/Object Pascal	1.142%	-0.28%
14	25	▲	Groovy	1.099%	+0.50%
15	15		Assembly language	1.022%	-0.09%

Рисунок 7 – Рейтинг використання мов програмування [7]

Чому саме Python? Що робить Python таким особливим? Причин багато - деякі з них були вказані вище, але для кращого порівняння з мовою програмування Java, було розглянуто детальний опис переваг:

- легкий у вивченні – період часу, необхідний для вивчення Python, коротший, ніж для багатьох інших мов; це означає, що можна швидше розпочати кодування на практиці;
- легкий у навчанні – об’єм відомостей для вивчення набагато менший, ніж у інших мовах; це означає, що вчитель може більше акцентувати увагу на загальних (незалежних від мови) техніках програмування, не витрачаючи сили на різні трюки, дивні винятки та незрозумілі правила;
- легкий у використанні для написання нового програмного забезпечення - часто можна писати код швидше, використовуючи Python;
- доступність у отриманні та встановленні - Python безкоштовний, відкритий та мультиплатформенний; не всі мови можуть цим похвалитися [7].

Python в дії. Ця мова програмування широко використовується для реалізації складних Інтернет-послуг, таких як пошукові системи, хмарне сховище та

соціальні медіа тощо. Кожного разу, при використанні будь-яких з цих служб, справа стосується роботи з Python.

У Python реалізовано багато інструментів для розробки. Все більше і більше програм для повсякденного використання пишеться на Python. Багато вчених відмовилися від дорогих індивідуальних інструментів і перейшли на Python. Також більшість тестувальників IT-проектів почали використовувати Python для проведення повторюваних процедур тестування.

Приклади Python. Python - чудовий вибір для:

- розробка веб (наприклад, фреймворки Django та Pyramid, мікрофреймворки Flask та Bottle);
- наукові та числові обчислення (наприклад, SciPy - колекція пакетів для математики, науки та техніки; Ipython - інтерактивна оболонка, що включає редагування та запис робочих сесій);
- освіта (це чудова мова для викладання програмування!);
- настільні графічні інтерфейси (наприклад, wxWidgets, Kivy, Qt);
- розробка програмного забезпечення (контроль збірки, управління та тестування - Scons, Buildbot, Apache Gump, Roundup, Trac);
- бізнес-додатки (ERP та системи електронної комерції - Odoo, Tryton);
- ігри (наприклад, серія Battlefield, Sid Meier \ 's Civilization IV), веб-сайти та сервіси (наприклад, Dropbox, UBER, Pinterest, BuzzFeed) [7].

Базуючись на описах трьох представлених вище мов програмування, обрано Java для розробки даного дипломного проєкту. За допомогою доступного синтаксису та досить великого об'єму як базових, так і написаних спільнотою бібліотек, навіть складні технічні AI або NLP задачі можна вирішувати ефективніше та швидше, ніж у Python або C++.

## **2.2 Інтегроване середовище розробки IntelliJ IDEA**

IntelliJ IDEA - це інтегроване середовище розробки (IDE) для мов JVM, розроблене для максимізації продуктивності розробників [8]. Середовище може

виконувати рутинні та повторювані завдання, забезпечуючи розумне завершення коду, статичний аналіз коду та рефакторинг, а також дозволяє зосередитися на яскравій стороні розробки програмного забезпечення, роблячи виконання роботи продуктивним процесом.

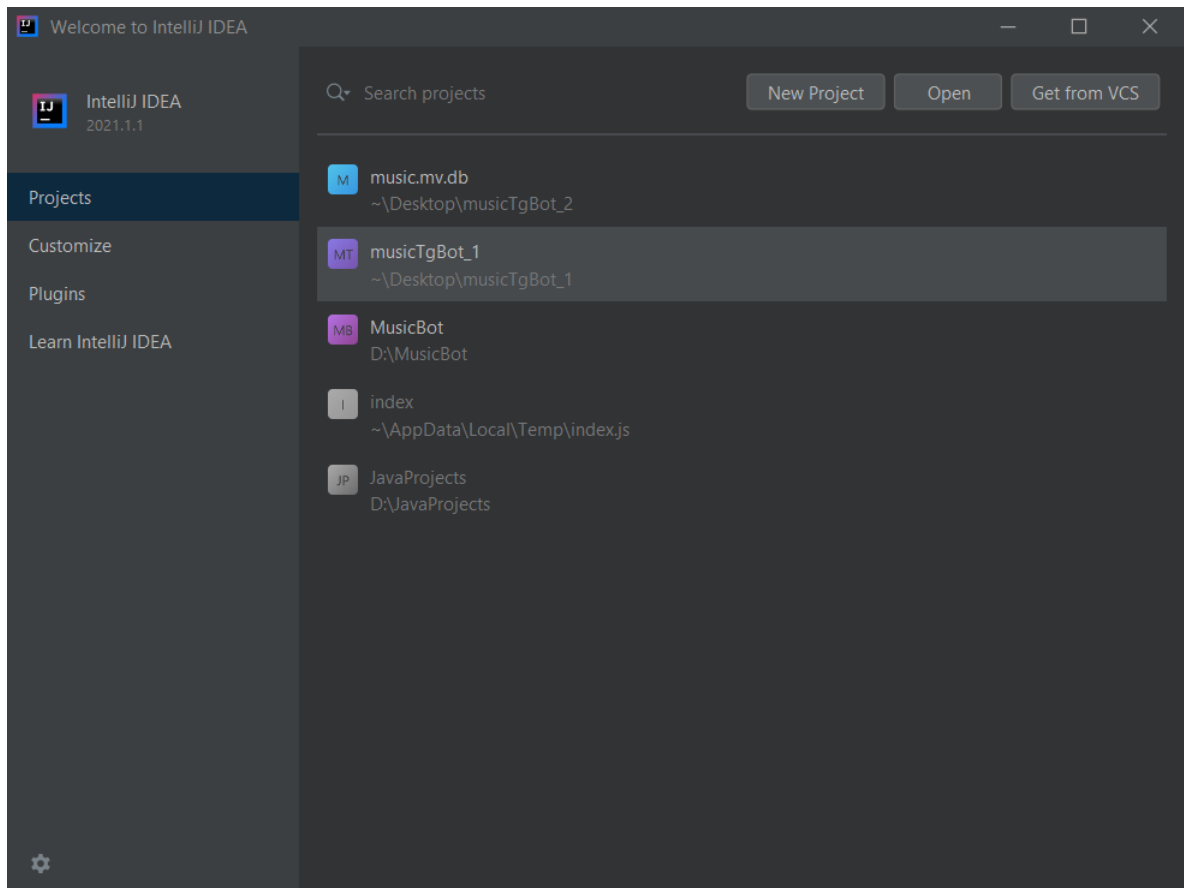


Рисунок 8 – Інтерфейс стартового вікна IntelliJ IDEA

Переваги використання середовища:

1. Підтримка великої кількості мов програмування. Використання IntelliJ IDEA для розробки програм наступними мовами, які можна скомпілювати в байт-код JVM, а саме:

- Java;
- Kotlin;
- Scala;
- Groovy.

Плагіни, що постачаються в комплекті з IntelliJ IDEA - є легко доступними,

вони підтримують додаткові мови програмування, а саме:

- Python (повна функціональність PyCharm );
- Ruby (повна функціональність RubyMine );
- PHP (повна функціональність PhpStorm );
- SQL (повна функціональність DataGrip );
- Go (повна функціональність GoLand );
- JavaScript (повна функціональність WebStorm );
- TypeScript (повна функціональність WebStorm );
- CoffeeScript (повна функціональність WebStorm ).

2. Мовний плагін для IntelliJ IDEA. IntelliJ IDEA Ultimate - це набір більшості середовищ розроблених на основі платформи IntelliJ, де включені мовні плагіни, що дають підтримку всіх технологій, доступних у більш конкретних середовищах розробки, таких як PyCharm, WebStorm , PhpStorm тощо. Так, наприклад, якщо код програми в основному на Java, але використовуються сценарії коду Python, використання IntelliJ IDEA у поєднанні з додатковим плагіном Python буде найзручнішим та найбезпечнішим засобом створення програми.

3. IntelliJ IDEA. IntelliJ IDEA випускається у трьох видах:

- IntelliJ IDEA Ultimate: комерційний продукт для розвитку JVM, веб-мереж та підприємств. Він включає всі функції видання Community, а також додає підтримку мов, на яких зосереджені інші середовища розробки створені на базі платформи IntelliJ; підтримка різноманітних серверних та інтерфейсних платформ, серверів додатків, інтеграції з базами даних та профілювання інструментів.

- IntelliJ IDEA Community Edition: безкоштовний продукт з відкритим кодом для розробки JVM та Android.

- IntelliJ IDEA Edu: безкоштовний продукт із вбудованими уроками для вивчення Java, Kotlin та Scala, інтерактивними завданнями та спеціальними функціями для вчителів для створення власних курсів та управління навчальним процесом [8].

4. Програма раннього доступу. IntelliJ IDEA Ultimate доступна безкоштовно в рамках Програми раннього доступу (EAP). Збірки EAP

публікуються до випуску стабільної версії продукту, тим самим можна завантажити їх, щоб спробувати нові функції, перш ніж вони стануть загальнодоступними. Збірки EAP налаштовані на збір статистичних даних використання функцій і є цінним джерелом інформації для розробників.

5. Інтерфейс користувача. IntelliJ IDEA - середовище орієнтоване на користувача. Програма слідує за внесеними змінами і автоматично пропонує необхідні інструменти, які допоможуть мінімізувати ризик переривання потоку роботи [9].

- Ергономічний дизайн та настроюваний зовнішній вигляд. Одним з найкращих доповнень IntelliJ IDEA є налаштованість. Можна змінити практично будь-що: зовнішній вигляд IDE, макет вікон та панелей інструментів, підсвічування коду тощо. Існує також безліч способів налаштування редактору та його роботи, щоб пришвидшити навігацію та позбутися будь-яких додаткових функцій, які відволікають від роботи.

1. Налаштування кольорів та шрифтів для вихідного коду, виводу консолі, інформації про налагодження, результатів пошуку тощо. Визначення або налаштування колірних схем, можна здійснити заздалегідь, щоб створити унікальне робоче середовище.

2. Налаштування меню та панелі інструментів, для позбавлення зайвих інструментів, які можуть заважати при роботі.

6. Навігація та пошук. IntelliJ IDEA забезпечує швидку навігацію не тільки всередині файлів вихідного коду, але і у всьому проєкті. Одним з найкорисніших елементів є “double Shift” - відкриття діалогового вікна «Search Everywhere», де можна знайти потрібний рядок серед усіх файлів, класів та символів, що належать до проєкту, і навіть серед дій IDE [10].

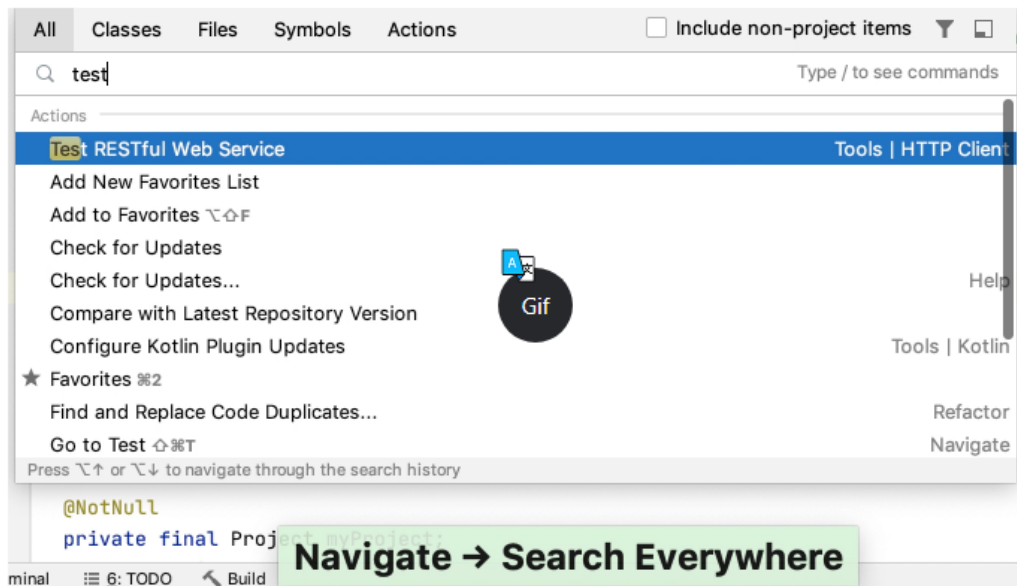


Рисунок 9 – Вікно пошуку середовища IntelliJ IDEA [10]

7. Допомога при написанні коду:

- Доповнення коду. IntelliJ IDEA допомагає прискорити процес кодування, забезпечуючи завершення контекстного коду.

1. Основне заповнення допомагає додати імена класів, методів, полів та ключових слів у межах видимості:



Рисунок 10 – Основне заповнення [8]

2. Інтуїтивне завершення підставляє найбільш відповідні символи, які застосовуються в поточному контексті, коли IntelliJ IDEA може визначити відповідний тип:

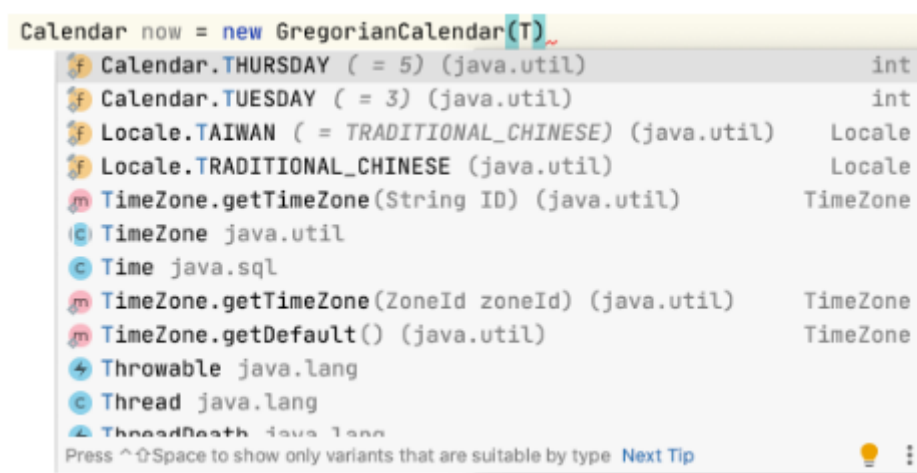


Рисунок 11 – Інтуїтивне завершення [8]

- Рефакторинг. IntelliJ IDEA пропонує повний набір автоматизованих рефакторингів коду, що призводить до значного підвищення продуктивності. Наприклад, коментування класу, IDE оновить усі посилання на цей клас у проєкті [11].

- Аналіз статичного коду. IntelliJ IDEA пропонує набір перевірок, які є вбудованими засобами статичного аналізу коду. Вони допомагають знаходити потенційні помилки, зайвий код, виявляти проблеми з продуктивністю та покращувати загальну структуру коду. Служби не тільки повідомляють де проблема, але й надають швидкі виправлення помилок. Окрім швидких виправлень, IntelliJ IDEA також надає дії з подіями, які допомагають застосовувати автоматичні зміни до правильного коду.

- Генерація коду. IntelliJ IDEA пропонує безліч способів генерації загальних конструкцій коду та повторюваних елементів, що допомагає підвищити продуктивність, виконуючи рутинні завдання IDE. Сюди входить генерація коду із заздалегідь визначених або спеціальних шаблонів коду, генерація обгортки, гетери та сетери, автоматичне сполучення символів тощо [8].

8. Інтеграція з інструментами розробника. Окрім надання інтелектуальної допомоги в навігації та кодуванні, IntelliJ IDEA інтегрує основні інструменти розробника та дозволяє налагоджувати, аналізувати та редагувати базу коду програм у середовищі IDE.

- IntelliJ IDEA представляє вбудований налагоджувач JVM. Це дозволяє отримувати та аналізувати інформацію про час роботи, що корисно для діагностики проблем та розуміння того, як працює програма.

- Профайлер. IntelliJ IDEA пропонує наступні вбудовані інструменти профайлера, які дозволяють дослідити, що використовує пам'ять процесора. Інструменти роботи:

1. Асинхронний Profiler: інструмент для Linux і MacOS, який дозволяє побачити, як саме пам'ять і ресурси процесора виділяються в додатку.

2. Java Flight Recorder: інструмент мульти-платформа, яка збирає інформацію про події в певний момент часу на віртуальній машині Java при виконанні програми.

- Термінал. IntelliJ IDEA включає вбудований термінал для роботи з оболонкою командного рядка всередині IDE. Наприклад, якщо зазвичай виконання команди Git відбувається із командного рядка, можна запустити її із терміналу, а не викликати цю дію з меню.

Отже, аналізуючи вищевказані переваги та недоліки, для створення практичної частини роботи було обрано інтегроване середовище розробки IntelliJ IDEA, так як цей продукт має безліч переваг та додаткових функцій у порівнянні з його аналогами.

## **2.3 База даних H2**

H2 – це полегшена база даних Java з відкритим кодом. Вона може бути вбудована в додатки Java або працювати в режимі клієнт-сервер. В основному базу даних H2 використовують для розробки і тестування.

Нижче наведені основні особливості бази даних H2.

- Надзвичайно швидкий, відкритий вихідний код, JDBC API;
- Доступний у вбудованому і серверному режимах;
- Консольний додаток на основі браузера;
- Компактність – розмір файлу jar близько 1,5 МБ;



- Підтримує стандартний SQL і JDBC API. Вона також може використовувати драйвер PostgreSQL ODBC.
- Підтримує кластеризацію і багатоверсійний паралелізм

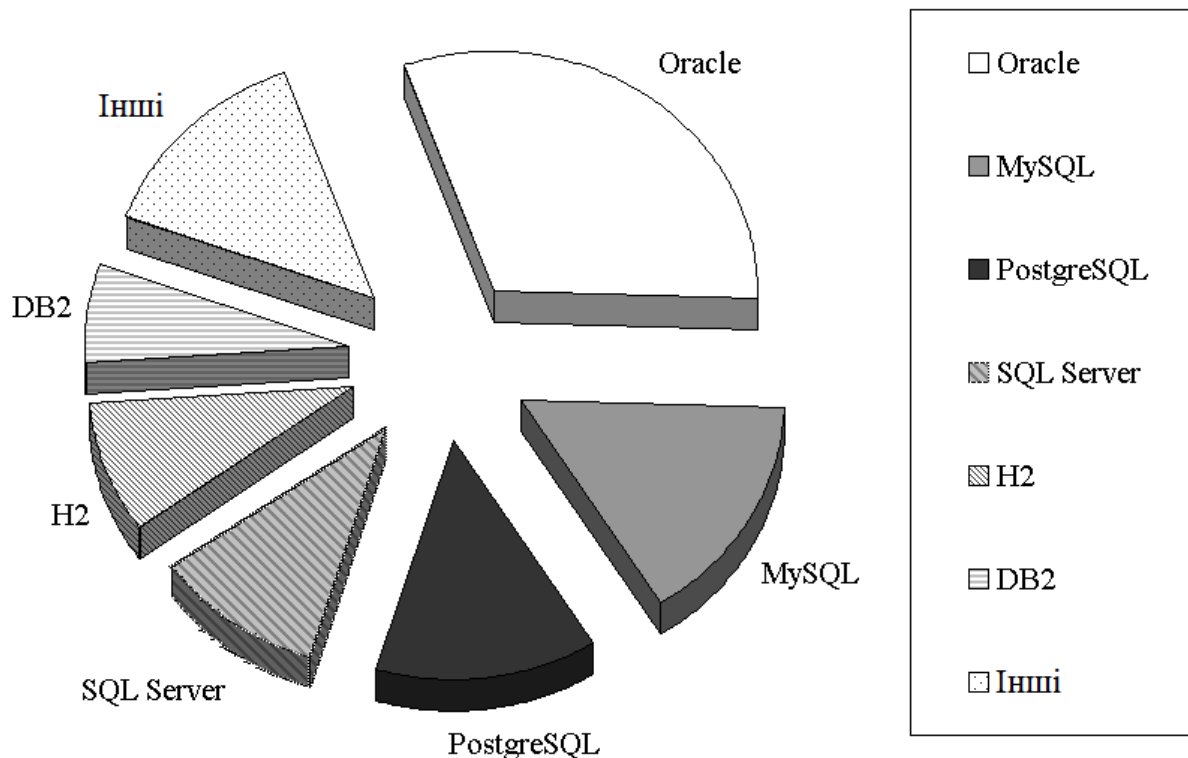


Рисунок 12 – Порівняльна таблиця СУБД [12]

Як видно на рисунку 12, H2 користується великою популярністю серед розробників, завдяки своїм особливостям.

## 2.4 Документація до бібліотек для створення Telegram бота

Перед розробкою практичного проєкту необхідно ознайомитися з офіційною документацією до бібліотек по розробці ботів на Telegram. Відомості цієї теми будуть висвітлювати детальний опис кроків роботи, які необхідні перед початком розробки music – боту, а саме:

1. Авторизація бота. Кожному боту при створенні присвоюється унікальний токен виду “123456: ABC-DEF1234ghlkl-zyx57W2v1u123ew11”. У документації для простоти замість нього можна використовувати <token>.

- Відправлення запитів. Всі запити до Telegram Bot API повинні здійснюватися через HTTPS в наступному вигляді:  
[https://api.telegram.org/bot<token>/НАЗВА\\_МЕТОДУ](https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ). Наприклад:

<https://api.telegram.org/bot123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11/getMe>.

Допускаються GET і POST запити. Для передачі параметрів в Bot API доступні 4 способи:

- 1) Запит в URL;
- 2) application / x-www-form-urlencoded;
- 3) application / json (не підходить для завантаження файлів);
- 4) multipart / form-data (для завантаження файлів).

Відповідь прийде у вигляді JSON-об'єкта, в якому завжди буде булеве поле ok і рядкове поле description, що містить опис результату. Якщо поле ok істинно, то запит пройшов успішно і результат його виконання можна побачити в полі result. У разі помилки поле ok дорівнюватиме false, а причини помилки будуть описані в полі description.

- Відправлення запитів при отриманні оновлення. Якщо бот працює через вебхуки (метод збільшення або розширення функціональності веб-сторінки або веб-застосунку за допомогою користувацьких зворотних викликів (callbacks)) [13] то можна здійснити запит до Bot API одночасно з відправкою відповіді на вебхук. Для передачі параметрів потрібно використовувати один з типів:

- 1) application / json;
- 2) application / x-www-form-urlencoded;
- 3) multipart / form-data.

Метод для виклику повинен бути визначений у запиті в полі method.

2. Отримання оновлення. Існує два діаметрально протилежних за логікою способу отримання оновлення від бота: getUpdates і вебхуки. Вхідні оновлення будуть зберігатися на сервері до тих пір, поки не відбудеться їх обробка, але не довше 24 годин. Незалежно від способу отримання оновлень, у відповідь можна отримати об'єкт Update, серіалізований в JSON.

1) Update. Цей об'єкт представляє із себе вхідне оновлення. Під оновленням мається на увазі дію, виконану з ботом - наприклад, отримання повідомлення від користувача. Тільки один з необов'язкових параметрів може бути присутнім у кожному оновленні.

Таблиця 2 – Параметри оновлення та їх опис [14]

Поле	Тип	Опис
update_id	Integer	Унікальний ідентифікатор оновлення. Ідентифікатори оновлення починаються з певного додатного числа і послідовно збільшуються. Цей ідентифікатор стає особливо зручним, якщо використовується Webhooks, оскільки він дозволяє ігнорувати повторні оновлення або відновлювати правильну послідовність оновлень, якщо вони виходять з ладу.
message	Message	Нове вхідне повідомлення будь-якого типу - текст, фотографія, наклейка тощо.
inline_query	InlineQuery	Новий вхідний вбудований запит.
chosen_inline_result	ChosenInlineResult	Результат вбудованого запиту, який обрав користувач і надіслав своєму партнеру по чату.
callback_query	CallbackQuery	Новий запит зворотного дзвінка.

2) GetUpdates. Метод використовується для отримання оновлень через long polling. Відповідь повертається у вигляді масиву об'єктів Update.

3) SetWebhook. Метод необхідний для завдання URL вебхука, на який бот буде відправляти оновлення. Кожен раз при отриманні оновлення на цю адресу буде відправлений HTTPS POST з серіалізованим в JSON об'єктом Update. При невдалому запиті до сервера спроба буде повторена. Для більшої безпеки рекомендується включити токен в URL вебхука, наприклад, так: [https://yourwebhookserver.com/ <token>](https://yourwebhookserver.com/<token>). Так як ніхто сторонній не знає токена, який буде використовуватися, можна бути впевненим, що запити до вебхуку шле саме Telegram [14].

3. Використання типів. Всі типи, що використовуються в Bot API, є JSON-об'єктами. Для зберігання всіх полів типу Integer безпечно використовувати 32-бітові знакові цілі числа.

- 1) User. Об'єкт являється ботом або користувачем Telegram.
- 2) Chat. Чат - об'єкт.
- 3) Message. Об'єкт - повідомлення.
- 4) MessageEntity. Об'єкт представляє одну з особливих сутностей в текстовому повідомленні. Наприклад: хештеги, імена користувачів, посилання ітд.
- 5) PhotoSize. Об'єкт представляє зображення певного розміру або прев'ю файлу / стікера.
- 6) Audio. Об'єкт - аудіозапис, який клієнти Telegram сприймають як музичний трек.
- 7) Document. Об'єкт - файл, який не є фотографією, голосовим повідомленням або аудіозаписом.
- 8) Sticker. Об'єкт - стікер.
- 9) Video. Об'єкт – відеозапис.
- 10) Voice. Об'єкт – голосове повідомлення.
- 11) Contact. Об'єкт - контакт з номером телефону.
- 12) Location. Об'єкт - точка на карті.
- 13) Venue. Об'єкт на карті.
- 14) UserProfilePhotos. Об'єкт містить фотографії профілю користувача.

15) File. Об'єкт являє файл, готовий до завантаження. Він може бути завантажений за посиланням виду `https://api.telegram.org/file/bot <token> / <file_path>`. Посилання буде дійсним як мінімум протягом 1 години. Після закінчення цього терміну URL може бути запрошено ще раз за допомогою методу `getFile` [14].

4. Використання методів. Усі методи в Bot API не враховують регістр. Підтримуються методи GET та POST HTTP.

1) GetMe. Простий метод тестування автентичності бота. Не вимагає додаткових параметрів. Повертає основну інформацію про бота у вигляді об'єкта User.

2) SendMessage. Метод для надсилання текстових повідомлень. Якщо робота завершилася успіхом надіслане повідомлення повертається.

3) Formatting options. Бот API підтримує базове форматування повідомлень. Можна використовувати жирний та курсивний стилі тексту, а також вбудовані посилання та попередньо відформатований код у повідомленнях ботів. Також можна використовувати форматування в стилі розмітки або HTML.

4) ForwardMessage. Метод пересилання повідомлень будь-якого типу.

5) SendPhoto. Спосіб надсилання фотографій.

6) SendAudio. Метод надсилання аудіофайлів. Наразі боти можуть надсилати аудіофайли розміром до 50 МБ, але це обмеження може бути змінено.

7) SendDocument. Метод надсилання загальних файлів. Наразі боти можуть надсилати файли будь-якого типу розміром до 50 МБ, але це обмеження може бути змінено [14].

8) GetFile. За допомогою цього методу можна отримати основну інформацію про файл та підготувати його до завантаження. На даний момент боти можуть завантажувати файли розміром до 20 МБ.

9) KickChatMember. Метод для видалення користувача з групи або супергрупи [14].

5. Inline-режим. Методи та об'єкти, які дозволяють боту працювати у вбудованому режимі.

## 3 РОЗРОБКА MUSIC-BOT

### 3.1 Реєстрація бота в системі Telegram Bot API

Для початку розробки будь-якого бота, перш за все, його потрібно зареєструвати у спеціального чат-бота під назвою «BotFather». BotFather – це бот (інтерфейс) розроблений компанією Telegram, за допомогою якого кожен користувач може, легко та зручно, створити та налаштувати свого бота для месенджеру телеграм [15]. Щоб зареєструвати бота потрібно в пошуку знайти @BotFather та запустити його командою /start. Потім за допомогою команди /newbot ми створюємо власного бота. Під час створення ми повинні придумати, будь-яке, ім'я (воно буде відображатися в контактах) для нашого бота та username (використовується для легкого пошуку). Username обов'язково повинен бути унікальним і закінчуватися на «bot». На рисунку 13 зображено процес реєстрації бота.

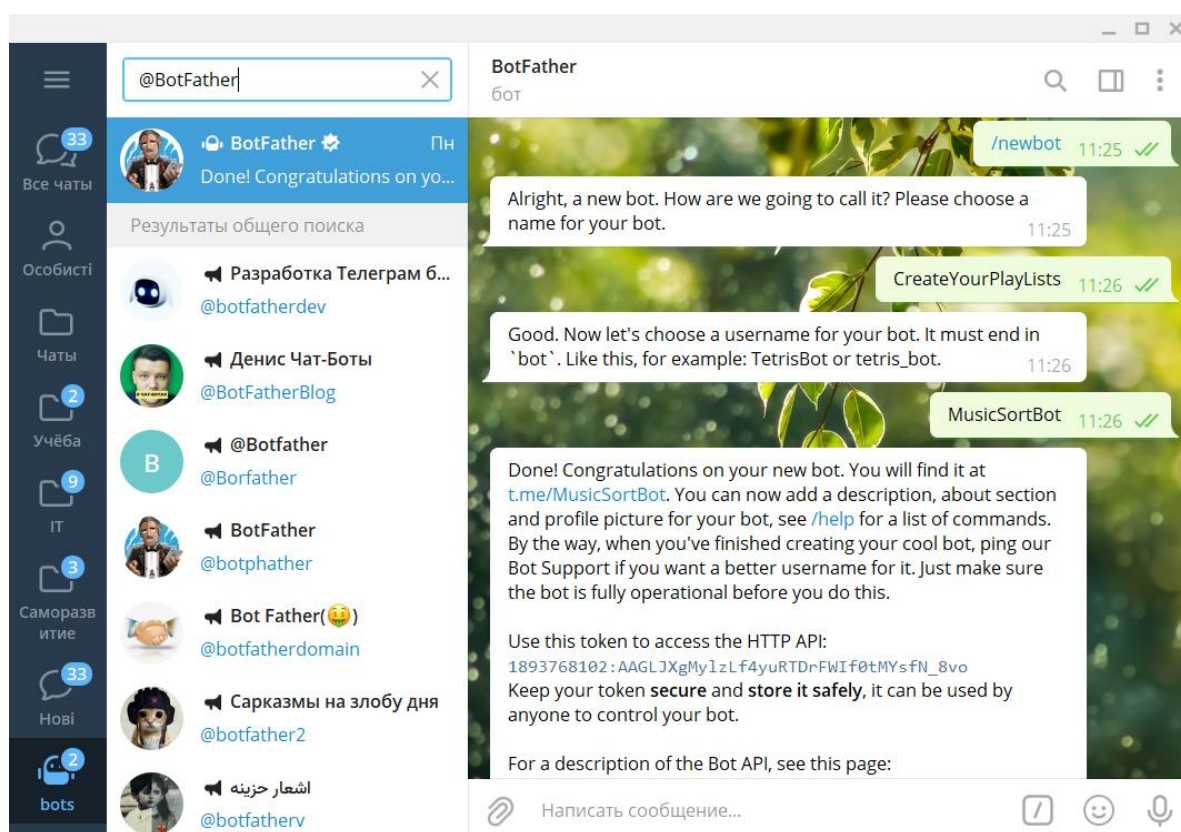


Рисунок 13 – Реєстрація бота

Після реєстрації BotFather генерує та надсилає токен. Токен – це унікальний набір символів, який являє собою ключ доступу до HTTP API протоколу управління ботом. Також за допомогою BotFather можна здійснити базові налаштування бота. В таблиці 3 наведено список команд для зміни ботів.

Таблиця 3 – Список команд для налаштування ботів

Команда	Призначення
/setname	змінити ім'я бота
/setdescription	змінити опис бота
/setabouttext	змінити інформацію про бота
/setuserpic	змінити фото профілю бота
/setcommands	змінити список команд бота
/deletebot	видалити бота

Коли отримано токен і здійснено необхідні налаштування в BotFather можна почати розробку програмної частини бота.

### 3.2 Розробка структури та алгоритму бота

Бот розроблявся по принципу long polling, тобто при натисканні кнопки відбувається запит до сервера який в свою чергу відправляє у відповідь масив об'єктів Update. Наповнення цього об'єкта залежить від того яка callback.data була вказана у натиснутої кнопки (простіше яку команду ми надіслали боту).

У кожній команді є свій обробник і кожна команда виводить своє меню. Всі обробники команд реалізують інтерфейс CommandButtonProcessor. На рисунку 14 зображено реалізацію цього інтерфейса.

```

public interface CommandButtonProcessor {
    String description();
    String callbackData();

    // process update on command
    void processCommand(TelegramBot bot, Update update);
    // process update on button press
    void processButton(TelegramBot bot, Update update);
}

```

Рисунок 14 – Реалізація класу CommandButtonProcessor

Потім вони реєструються в класі CommandButtonRegistry

```

public class CommandButtonRegistry {
    private final static Logger logger = Logger.getLogger(CommandButtonRegistry.class.getName());

    private LinkedHashMap<String, CommandButtonProcessor> buttons = new LinkedHashMap<>();

    public void registerCommand(CommandButtonProcessor commandButton) {
        if (buttons.containsKey(commandButton.callbackData())) {
            logger.warning(msg: "Already has button with name " + commandButton.callbackData() + ". Ignore " + commandButton.getClass());
        } else {
            buttons.put(commandButton.callbackData(), commandButton);
        }
    }

    public Set<String> getCommands() { return buttons.keySet(); }

    public boolean processUpdate(TelegramBot bot, Update update) {
        // process callbackQuery (result of button push)
        if (update.callbackQuery() != null) {
            String buttonName = update.callbackQuery().data().split( regex: ":", limit: 2)[0];
            var processor : Optional<CommandButtonProcessor> = Optional.ofNullable(buttons.get(buttonName));
            processor.ifPresent(b -> b.processButton(bot, update));
            if(processor.isEmpty()) {
                logger.warning(msg: "Cant find button processor for callback " + buttonName);
            }
            return processor.isPresent();
        }

        // process command (result of /command)
        var message : Message = update.message();
        if (message != null && message.text() != null && message.text().startsWith("/")) {
            String commandName = message.text().substring(1); // drop first '/'
            var processor : Optional<CommandButtonProcessor> = Optional.ofNullable(buttons.get(commandName));
            processor.ifPresent(c -> c.processCommand(bot, update));
            if(processor.isEmpty()) {
                logger.warning(msg: "Cant find command processor for callback " + commandName);
            }
            return processor.isPresent();
        }

        return false;
    }
}

```

Рисунок 15 – Реалізація класу CommandButtonRegistry



І в класі `CommandButtonRegistry` вже викликається потрібний обробник.

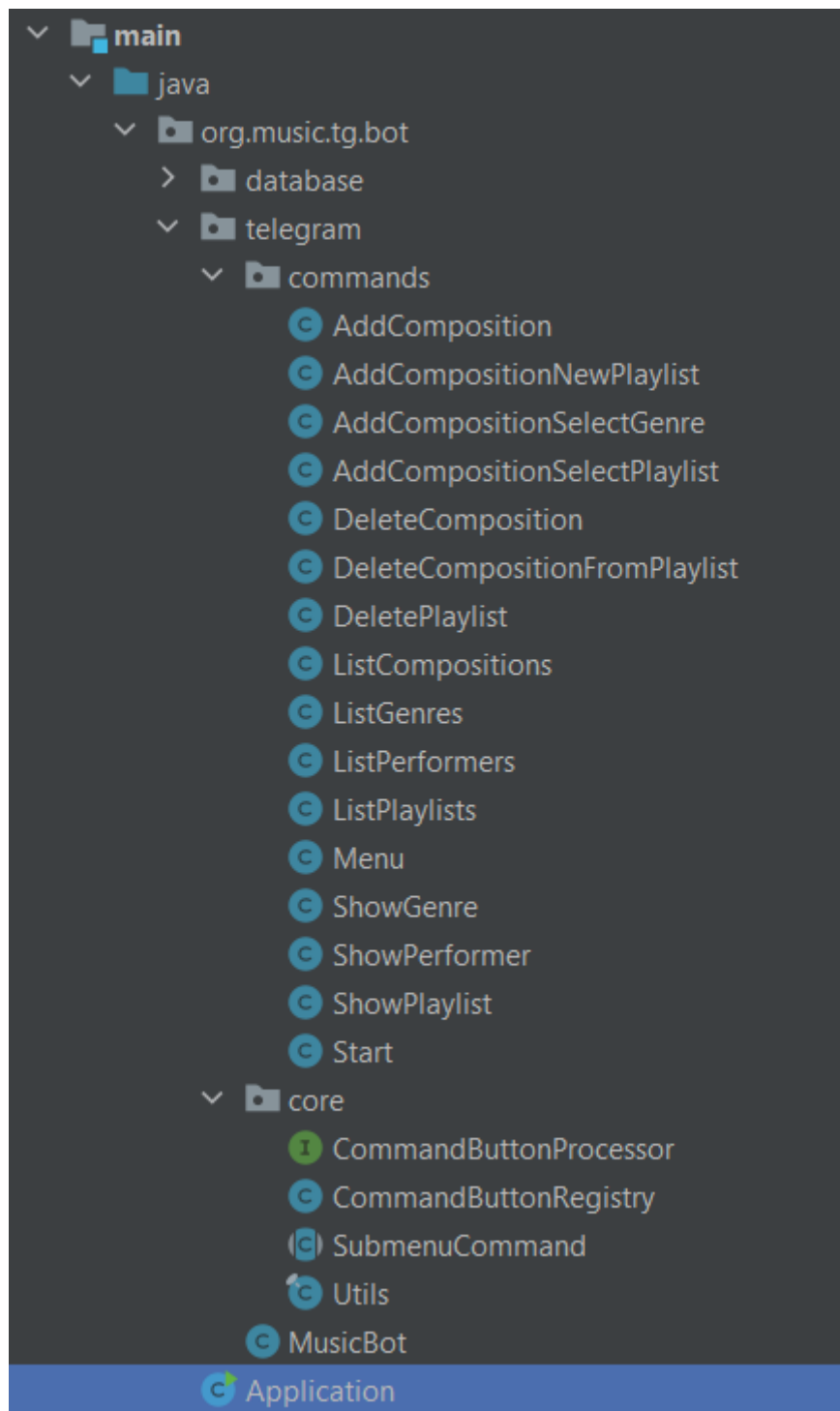


Рисунок 16 – Структура проекту

На рисунку 16 продемонстровано структуру проекту в середовищі розробки IntelliJ IDEA. Пакет `telegram.core` містить логіку роботи телеграм-інтерфейса.

В пакеті `telegram.commands` реалізовані самі обробники команд. Клас

MusicBot створює обробники кнопок і містить метод startBot() в якому реалізовано основний цикл обробки і окремо обробку аудіо файлів і введеного тексту.

```
private void initProcessors() {
    // init button processors and its submenus
    var menu = new Menu();
    var start = new Start(menu);
    addComposition = new AddComposition(manager);
    var addCompositionGenre = new AddCompositionSelectGenre(manager);
    addCompositionNewPlaylist = new AddCompositionNewPlaylist(manager);
    var addCompositionSelectPlaylist = new AddCompositionSelectPlaylist(manager);
    var deleteCompositionFromPlaylist = new DeleteCompositionFromPlaylist(manager);
    var deletePlaylist = new DeletePlaylist(manager);
    var deleteComposition = new DeleteComposition(manager);
    var listGenres = new ListGenres(manager);
    var listPerformers = new ListPerformers(manager);
    var listCompositions = new ListCompositions(manager);
    var listPlaylists = new ListPlaylists(manager);
    var showPerformer = new ShowPerformer(manager);
    var showGenre = new ShowGenre(manager);
    var showPlaylist = new ShowPlaylist(manager);

    menu.setSubmenu(listGenres, listPerformers, listCompositions, listPlaylists, addComposition);

    // register command processors
    Arrays.asList(menu, start, addComposition, listGenres, listPerformers, listCompositions, listPlaylists
        , showPerformer, showGenre, showPlaylist, addCompositionGenre, addCompositionNewPlaylist, addCompositionSelectPlaylist
        , deleteCompositionFromPlaylist, deletePlaylist, deleteComposition
    ).forEach(CommandButtonRegistry::registerCommand);
}
```

Рисунок 17 – Створення обробників

```
public void startBot() {
    // Create your bot passing the token received from @BotFather
    TelegramBot bot = new TelegramBot(token);
    // setting available commands for autocompletion
    List<BotCommand> commandsList = commandButtonRegistry.getCommands().stream()
        .map(c -> new BotCommand(c, c)).toList();
    bot.execute(new SetMyCommands(commandsList.toArray(new BotCommand[0])));
    // Register update listener
    bot.setUpdatesListener(updates -> {
        updates.forEach(update -> {
            boolean processed = commandButtonRegistry.processUpdate(bot, update);
            if (!processed) {
                var message : Message = update.message();
                if (message != null && message.audio() != null) {
                    // add uploaded audio
                    addComposition.processAudio(bot, update);
                } else if (message != null && message.text() != null) {
                    // create new playlist for composition
                    addCompositionNewPlaylist.addCompositionNewPlaylist(bot, chatId(update), message.text());
                } else {
                    logger.warning(msg: "Cant process message " + new Gson().toJson(update));
                }
            }
        });
    });

    return UpdatesListener.CONFIRMED_UPDATES_ALL;
};
```

Рисунок 18 – Реалізація метода startBot()

В класі Application міститься метод main() з якого починається робота програми, в ньому вказується токен, отриманий при реєстрації, та викликається

метод startBot() для запуску головного циклу. На рисунку 19 зображено його реалізацію.

```
package org.music.tg.bot;

import org.music.tg.bot.telegram.MusicBot;

public class Application {

    public static void main(String[] args) {
        MusicBot bot = new MusicBot(token: "1893768102:AAGLJXgMyLzLf4yuRTDrFWIf0tMYsfN_8vo");
        bot.startBot();
    }
}
```

Рисунок 19 – Реалізація класу Application

### 3.3 Структура та реалізація бази даних

В даному проєкті використовується база даних H2 вбудована в програму тож під неї було створено окремий пакет database де зберігається все що її стосується. На рисунку 20 продемонстровано структуру бази даних в середовищі розробки IntelliJ IDEA.

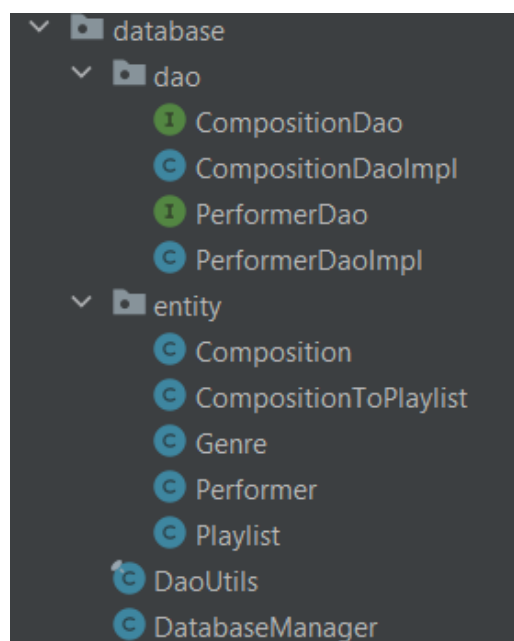


Рисунок 20 – Структура пакету database

Пакет `database.dao` зберігає класи `data access objects`. З допомогою цих об'єктів відбувається доступ до даних у вигляді CRUD - `create, read, update, delete`.

Пакет `database.entity` містить сутності таблиць з яких напряму створюються таблиці вперше. В них задаються колонки, ID та відношення між таблицями. А в класі `DatabaseManager` реалізовано бізнес-логіку роботи з сутностями, їх створення та пошук. На рисунку 21 зображено реалізацію класу `DatabaseManager`.

```
public class DatabaseManager {
    private static final Logger logger = LoggerFactory.getLogger(DatabaseManager.class.getName());
    private final String databaseUrl;

    // table entities. if change, drop database for update
    private static final List<Class?>> tables = List.of(
        Genre.class,
        Performer.class,
        Composition.class,
        Playlist.class,
        CompositionToPlaylist.class
    );

    // genres list. if change, drop database for update
    private static final List<String> genres = List.of(
        "Pop", "Rock", "Classic", "Techno", "Alternative"
    );

    public DatabaseManager(String databaseUrl) { this.databaseUrl = databaseUrl; }
```

Рисунок 21- Реалізація класу `DatabaseManager`

### 3.4 Тестування розробленого бота

Під час тестування було перевірено на працездатність всі функції бота.

Інтерфейс бота реалізовано за допомогою `Callback`-кнопок. Після запуску бота командою `старт` користувач одразу потрапляє до головного меню, яке зображено на рисунку 22.

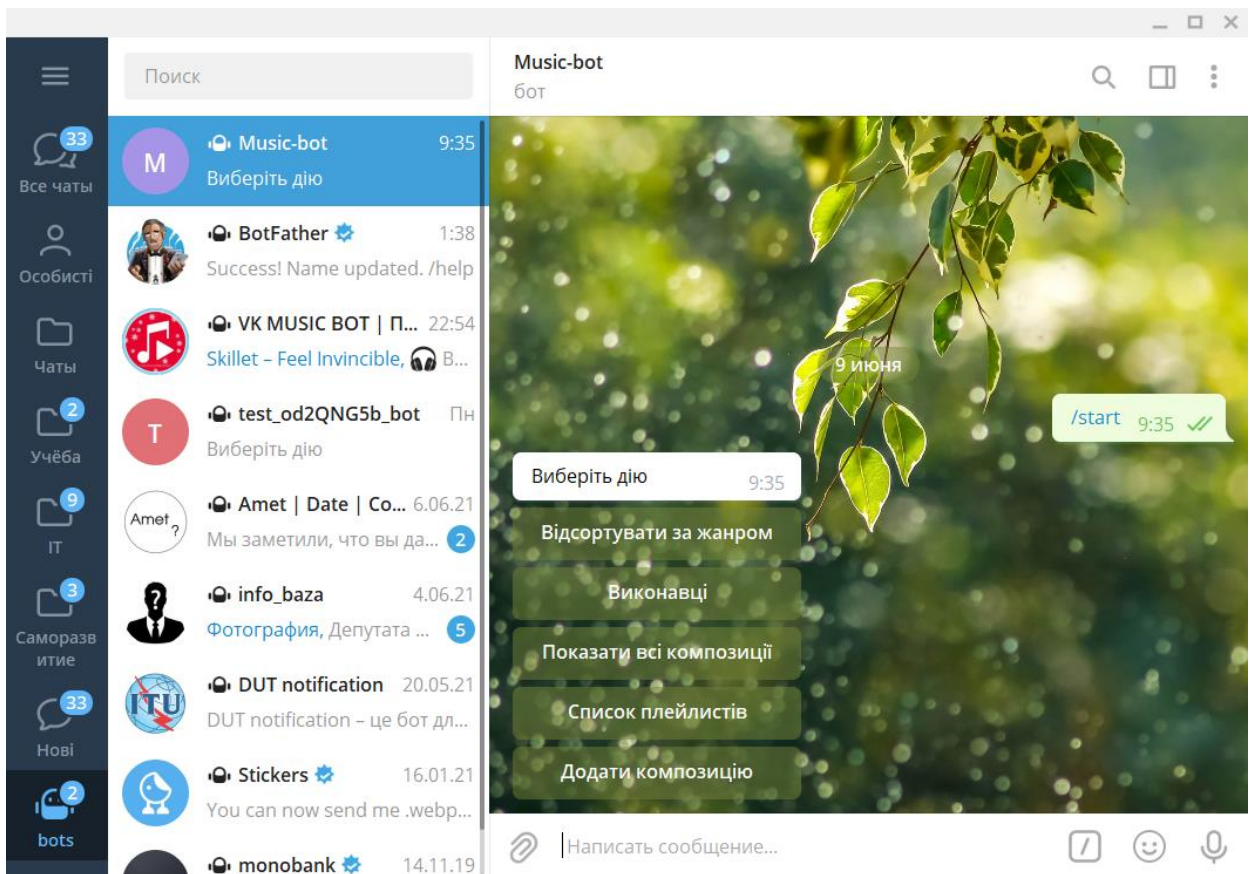


Рисунок 22 – Головне меню бота

В головному меню є 5 пунктів які зображено на рисунку 22. Щоб почати використання бота йому потрібно додати хоча б 1 композицію, це зробити дуже легко. Для початку натиснемо кнопку «Додати композицію» після чого бот запропонує надіслати йому файл з композицією. Це можна зробити двома шляхами надіслати файл с комп'ютера або переслати боту повідомлення з композицією. Коли композицію надіслано потрібно вказати жанр цієї композиції та обрати плейлист до якого вона буде додана, або створити новий. На рисунку 24 продемонстровано процес додання нової композиції та створення нового плейлиста. Коли користувач створює новий плейлист для нього потрібно придумати назву на вказати її. Після збереження композиції з'являється кнопка повернутися до головного меню. Якщо користувач захоче додати композицію вже збережену композицію то з'явиться відповідне повідомлення яке зображено на рисунку 23.

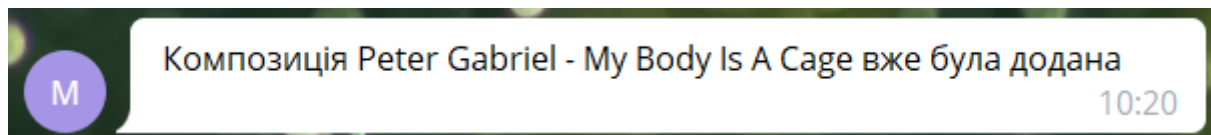


Рисунок 23 – Повідомлення про те що така композиція вже є

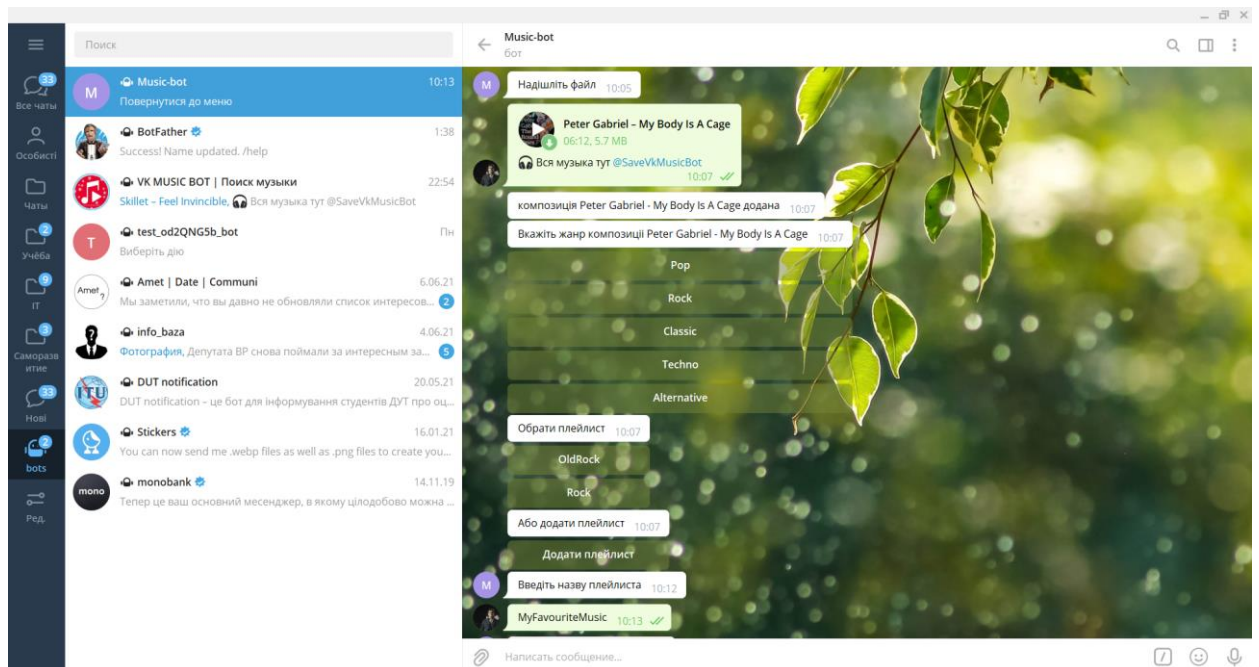


Рисунок 24 – Процес зберігання композиції

Коли користувач вже додав певну кількість композицій, їх можна відсортувати за жанрами та виконавцями. Для цього в головному меню є відповідні кнопки. При сортуванні за жанром користувачу пропонується обрати один із 5 жанрів, цей список статичний і формується в кодї програми під час створення бази даних. Після вибору жанра бот надсилає список із всіх доданих пісень, для яких при збереженні було обрано відповідний жанр. При сортуванні за виконавцем користувачеві також пропонується обрати зі списку виконавців, але цей список формується динамічно і залежить від виконавців композицій які було збережено. Якщо композицій обраного жанра не додано то бот відправить відповідне повідомлення, яке зображено на рисунку 25 і запропонує повернутися до меню.



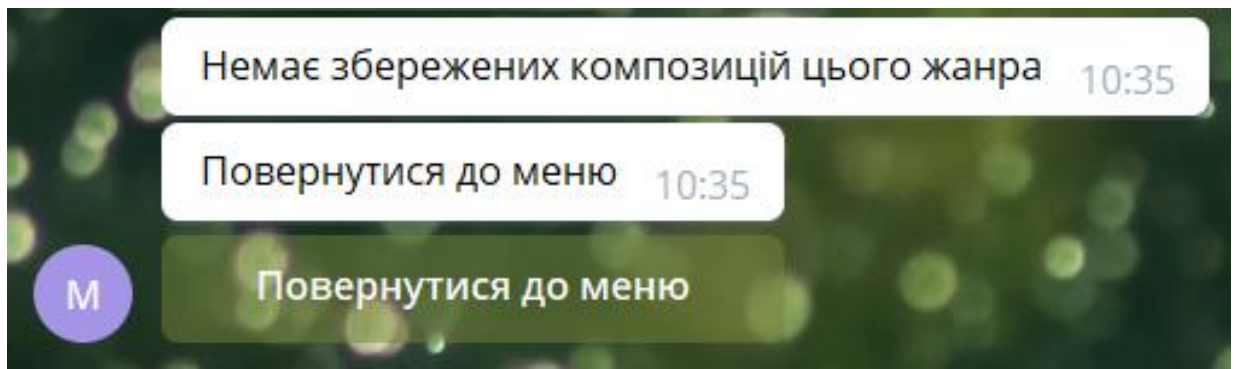


Рисунок 25 – Повідомлення про відсутність композицій

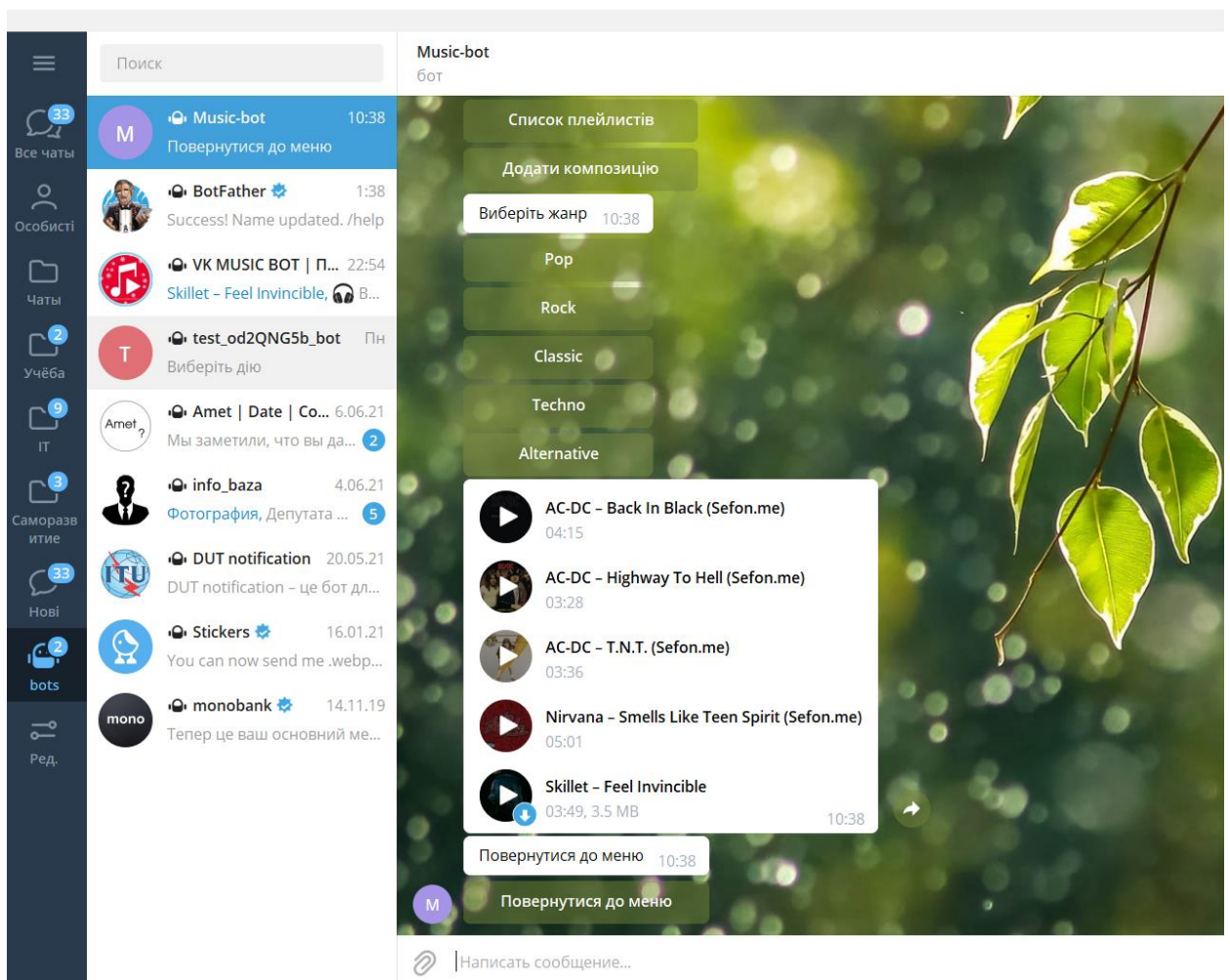


Рисунок 26 – Сортування за обраним жанром

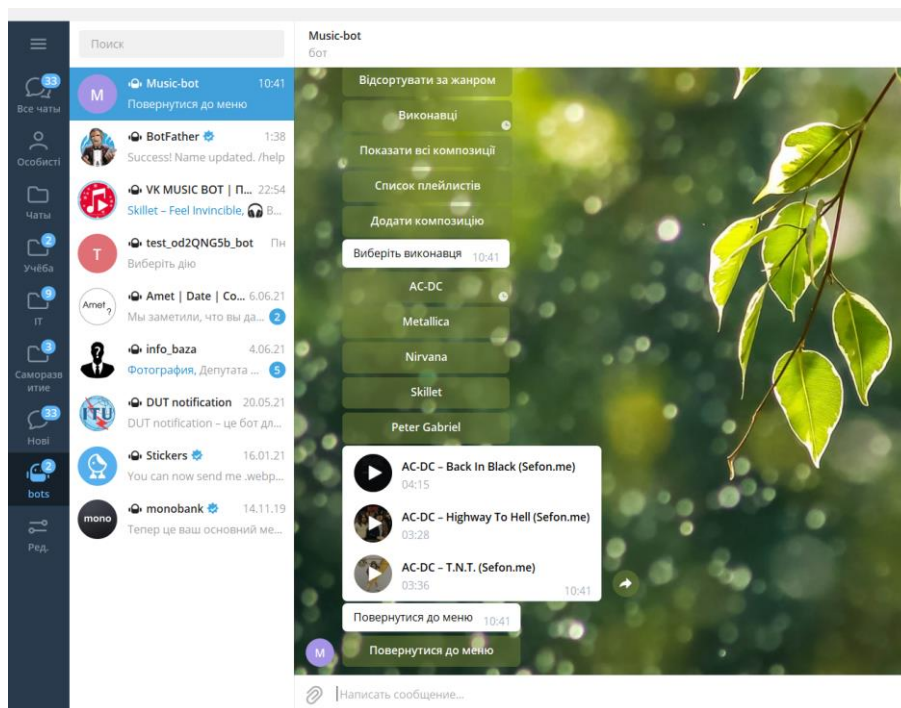


Рисунок 27 – Сортування за обраним виконавцем

Також в меню є кнопка «Переглянути всі композиції» при натисканні на яку бот відправить список всіх збережених композицій. В цьому підменю у користувача є можливість видалити композицію. Для цього потрібно натиснути відповідну кнопку та обрати пісні які буде видалено. На рисунку 28 продемонстровано процес видалення композиції.

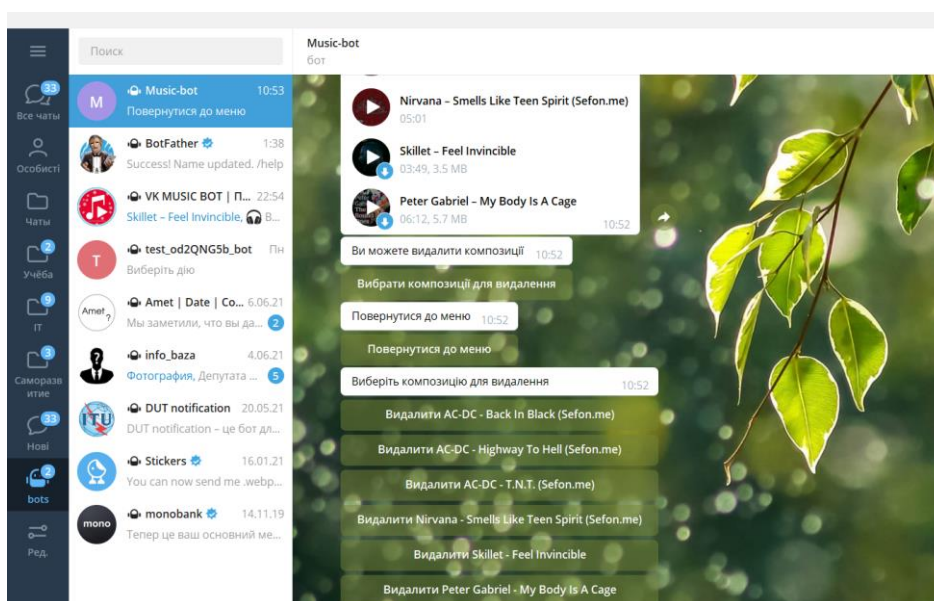


Рисунок 28 – Процес видалення композицій



В пункті меню «плейлисти» клієнту відображаються створені ним списки відтворення та пропонується обрати один із них. Коли користувач обере плейлист, бот надсилає йому список композицій які були додані до цього плейлиста і є можливість видалити весь плейлист, або окремі пісні із нього, цей функціонал продемонстровано на рисунках 30 та 31. Якщо обраний список відтворення буде порожній виведеться відповідне повідомлення та бот запропонує видалити його або повернутися до головного меню. Це зображено на рисунку 29.

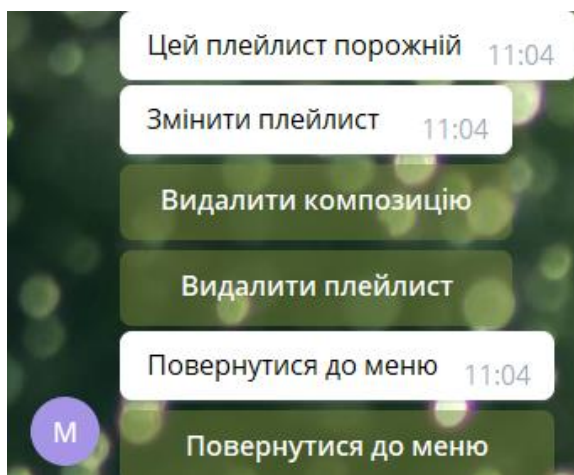


Рисунок 29 – Видалення порожнього плейлиста

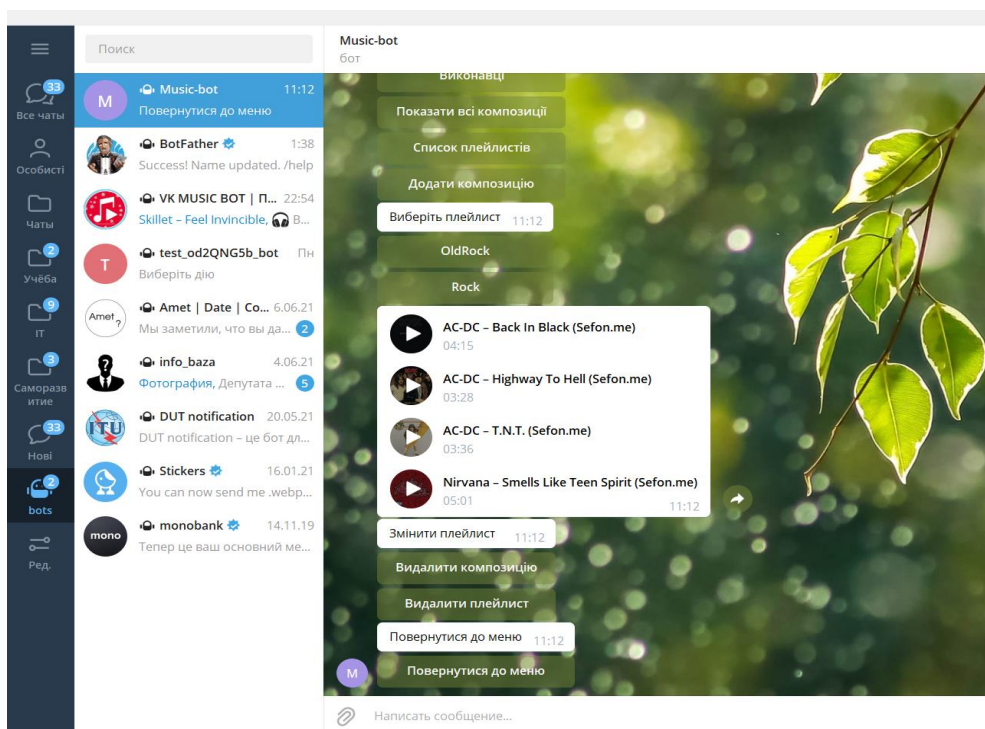


Рисунок 30 – Виведення списку відтворення

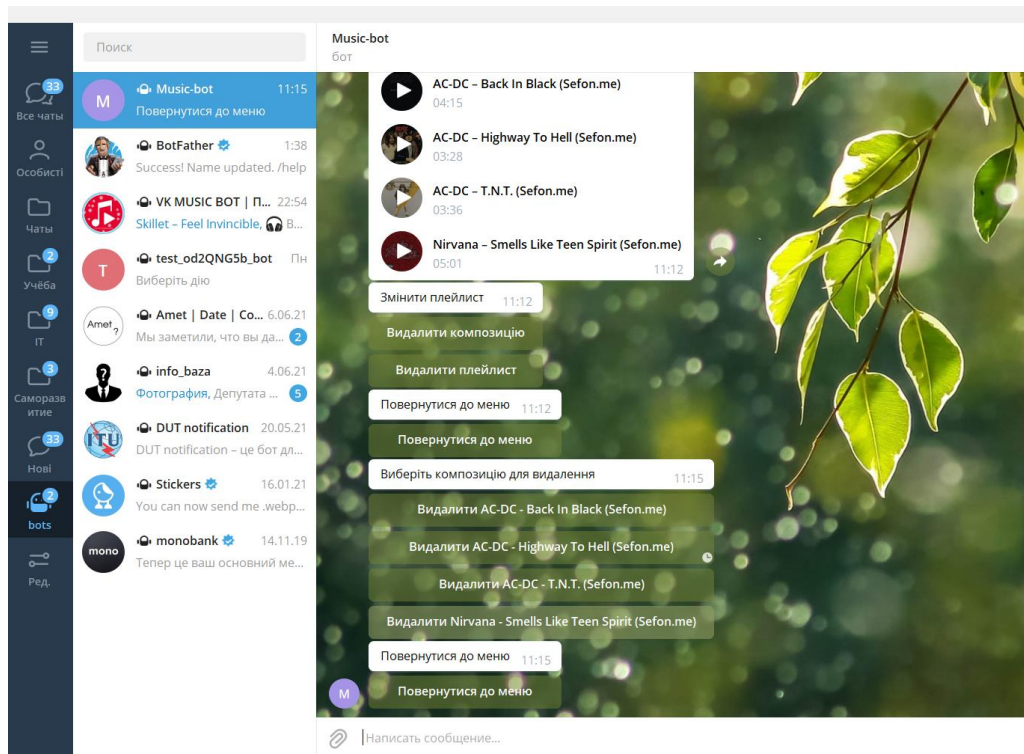


Рисунок 31 – Видалення композицій зі списку відтворення

Тестування відбувалося за схемою користувача зображеною на Рисунку 32. Під час тестування було виявлено 1 баг інтерфейса. Його суть в тому що бот пропонує видалити композицію із пустого списку відтворення. Інших багів інтерфейсу та функціоналу знайдено не було. Бот повністю виконує заявлені функції.

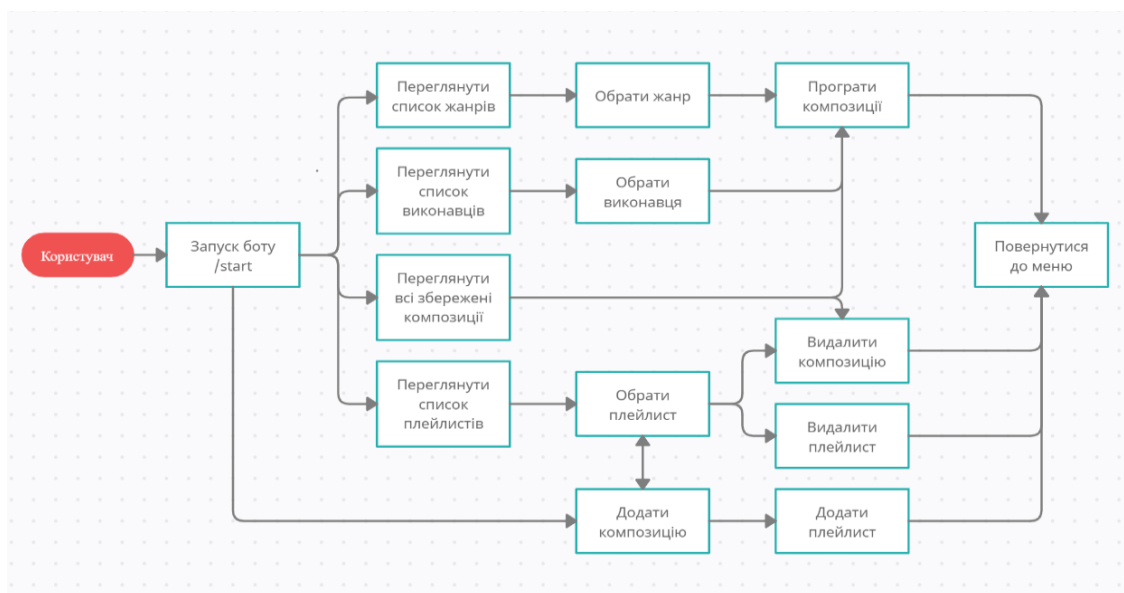


Рисунок 32 – Діаграма користувача

## ВИСНОВКИ

В ході виконання дипломної було розглянуто такі теми: обґрунтування вибору месенджера Telegram у якості платформи для прослуховування та сортування музики, порівняння мов програмування Java, Python та C++, інтегроване середовище розробки IntelliJ IDEA та документація до бібліотек для створення Telegram бота.

Було розглянуто та проаналізовано додатки аналоги та на основі їх недоліків сформовано список вимог до розроблюваного боту. На основі відповідних завдань можемо отримати такі результати дослідження: Telegram бот для сортування композицій є дуже зручним засобом для пошуку вподобаного треку за певними параметрами, адже використання Telegram ботів, для пошуку або сортування відомостей, у наш час, є дуже актуальною подією. Після аналізу різних технологій та середовищ для розробки практичної частини, було обрано мову програмування Java та інтегроване середовище IntelliJ IDEA, через їхні переваги над іншими програмними засобами.

Було розглянуто офіційну документацію до бібліотек для створення Telegram ботів, а саме: методи та об'єкти, які будуть використовуватися під час роботи.

Також було розроблено повнофункціонального Music-bot для сортування композицій, та проведено його тестування.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. T9GRAM [Електронний ресурс] – Telegram Music. How to Make a Music Player From the Messenger - Режим доступу: <https://t9gram.com/p/telegram-music>
2. SOVFARFOR [Електронний ресурс] – Музика в житті людини - Режим доступу: <https://ukr.sovfarfor.com/muzichne-mistectvo/135-muzika-v-zhitt-ljudini.html>
3. Комп'ютерна інженерія та кібербезпека [Електронний ресурс] – Автоматизація за допомогою використання ботів - Режим доступу: <https://conf.ztu.edu.ua/wp-content/uploads/2019/12/80.pdf>
4. ScienceDirect [Електронний ресурс] – Java Programming Language - Режим доступу: <https://www.sciencedirect.com/topics/computer-science/java-programming-language>
5. AndroidUkraine [Електронний ресурс] – Що таке Dalvik? - Режим доступу: [http://android.com.ua/dalvik\\_vm\\_machine.html](http://android.com.ua/dalvik_vm_machine.html)
6. JUNI [Електронний ресурс] – What is Java Coding? - Режим доступу: <https://junilearning.com/blog/guide/what-is-java-coding/>
7. PythonInstitute [Електронний ресурс] – What is Python? - Режим доступу: <https://pythoninstitute.org/what-is-python/>
8. JetBrains [Електронний ресурс] – IntelliJ IDEA overview - Режим доступу: <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>
9. SysSoft [Електронний ресурс] – JetBrains IntelliJ IDEA - Режим доступу: <https://www.syssoft.ru/JetBrains/JetBrains-IntelliJ-IDEA/>
10. JetBrains [Електронний ресурс] – Чому IntelliJ IDEA? - Режим доступу: <https://www.jetbrains.com/ru-ru/idea/>
11. Refactoring [Електронний ресурс] – Коли рефакторити? - Режим доступу: <https://refactoring.guru/ru/refactoring/when>
12. Drach [Електронний ресурс] – Порівняння сучасних СУБД - Режим доступу: <https://drach.pro/blog/hi-tech/item/145-db-comparison>
13. Wikipedia [Електронний ресурс] – Webhook - Режим доступу: <https://uk.wikipedia.org/wiki/Webhook>

14. TLGRM [Електронний ресурс] – Авторизація бота - Режим доступу:  
<https://tlgrm.ru/docs/bots/api#authorizing-your-bot>

15. BotCreators [Електронний ресурс] – BotFather - Режим доступу:  
<https://botcreators.ru/blog/botfather-instrukciya/>

16. Maximum [Електронний ресурс] – Spotify, Apple Music і YouTube Music -  
- Режим доступу: [https://maximum.fm/spotify-apple-music-i-youtube-music-porivnyannya-oglyad-ta-cini-strimingovih-servisiv\\_n181501](https://maximum.fm/spotify-apple-music-i-youtube-music-porivnyannya-oglyad-ta-cini-strimingovih-servisiv_n181501)

17. Spotify [Електронний ресурс] – About Us - Режим доступу:  
<https://www.spotify.com/us/about-us/contact/>

18. Справка Google [Електронний ресурс] – Про додаток YouTube Music -  
Режим доступу: <https://support.google.com/youtubemusic/answer/6313529?hl=ru>

19. Apple [Електронний ресурс] – Apple-music - Режим доступу:  
<https://www.apple.com/ua/apple-music/>

## ДОДАТОК



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## РОЗРОБКА MUSIC-BOT ДЛЯ СОРТУВАННЯ КОМПОЗИЦІЙ МОВОЮ JAVA

Виконав студент 4 курсу  
Групи ПД-44  
Горкуша Віктор Сергійович  
Керівник роботи  
Доцент Шевченко Світлана Миколаївна

Київ – 2021

## АНАЛОГИ

Назва	Переваги	Недоліки
Spotify	<ol style="list-style-type: none"><li>1. Пробна версія повністю безкоштовна;</li><li>2. Автоматично сформовані списки відтворення на основі ваших вподобань;</li><li>3. Збір та відображення статистики.</li></ol>	<ol style="list-style-type: none"><li>1. Для прослуховування музики <u>офлайн</u> потрібно купувати підписку;</li><li>2. Реклама;</li><li>3. Неможливо створювати власні <u>плейлісти</u>.</li></ol>
YouTube Music	<ol style="list-style-type: none"><li>1. Інтелектуальний пошук пісень, не тільки по назві, але й за частиною тексту;</li><li>2. Рекомендації на основі попередніх пошукових запитів і переглядів;</li><li>3. Можливість створювати власні списки відтворення.</li></ol>	<ol style="list-style-type: none"><li>1. Не завжди висока якість звуку;</li><li>2. Реклама;</li><li>3. Без підписки неможливо відтворювати композиції <u>офлайн</u>, або у фоновому режимі.</li></ol>
Apple Music	<ol style="list-style-type: none"><li>1. <u>Приємний та зрозумілий інтерфейс</u>;</li><li>2. Тривалість пробного періоду три місяці (безкоштовно);</li><li>3. Сервіс інтегрований в екосистему <u>Apple</u>.</li></ol>	<ol style="list-style-type: none"><li>1. Не завжди коректна робота рекомендацій;</li><li>2. Сервісом не можливо користуватися без підписки;</li><li>3. Не можливо додавати власні списки відтворення.</li></ol>

## МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** доповнити можливості месенджера Telegram за допомогою music-bot для сортування композицій.
- **Об'єкт дослідження:** особливості розробки ботів на основі месенджера Telegram.
- **Предмет дослідження:** розробка music-bot для сортування композицій мовою Java.

3

## ТЕХНІЧНІ ЗАВДАННЯ

- Проаналізувати аналоги;
- Розробити функціонал бота враховуючи недоліки аналогів;
- Обрати технології і середовища розробки;
- Розробити music-бот для сортування композицій;
- Протестувати розробленого бота.

4



# ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Telegram Bot API



Середовище розробки (IDE: [IntelliJ IDEA](#))



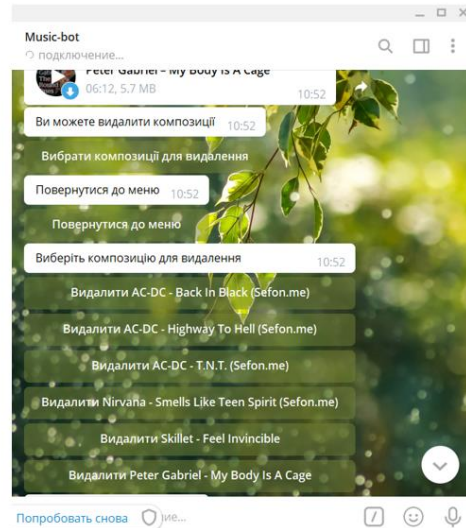
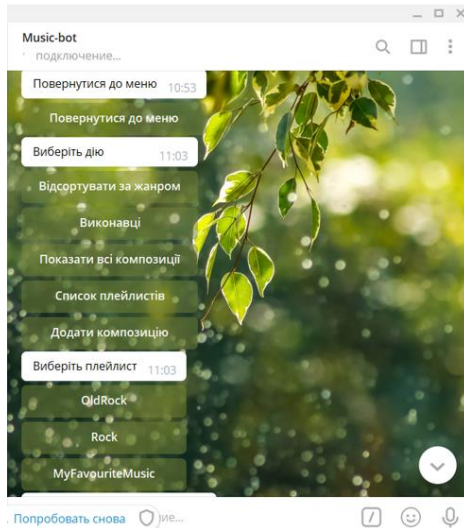
Система Управління Базою Даних



[Gradle](#)

5

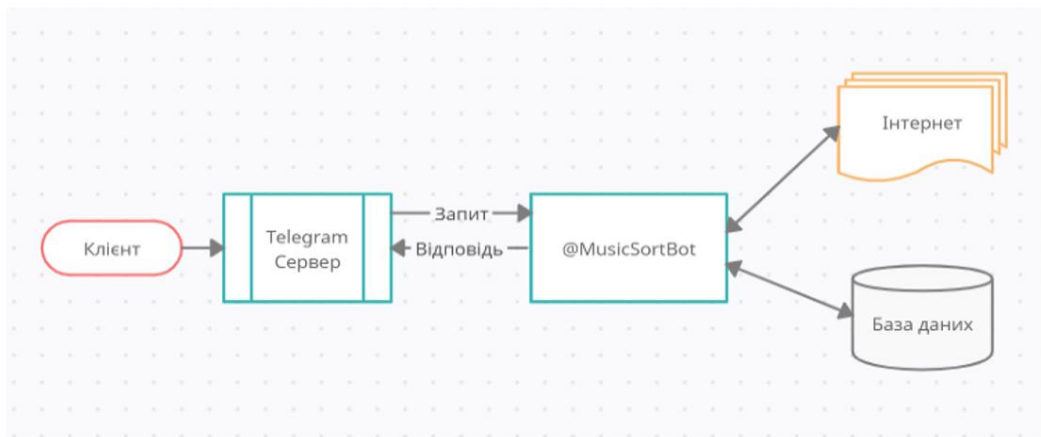
# ДЕМОНСТРАЦІЯ ІНТЕРФЕЙСА



6

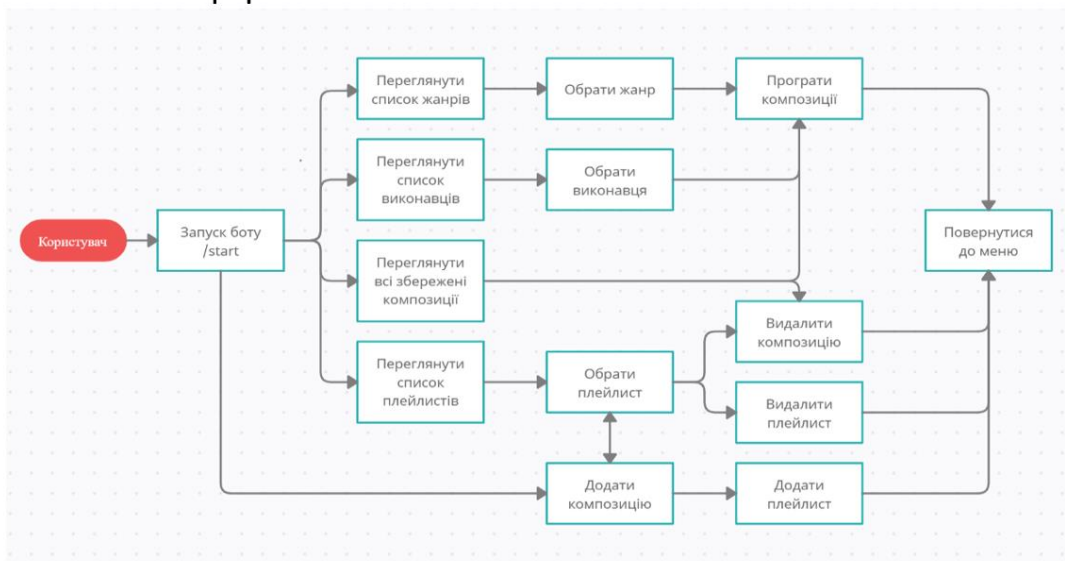


## ДІАГРАМА ВЗАЄМОДІЇ КЛІЄНТ-СЕРВЕР



7

## ДІАГРАМА КОРИСТУВАЧА



8

## ОПИС ТРИГЕРІВ

Назва	Опис
/start	Запуск бота
<a href="#">/list_compositions</a>	Переглянути список всіх композицій
<a href="#">/list_performers</a>	Переглянути список виконавців
<a href="#">/list_genres</a>	Переглянути список жанрів
<a href="#">/list_playlists</a>	Переглянути список <a href="#">плейлистів</a>
<a href="#">/add_composition</a>	Додати композицію
<a href="#">/delete_composition</a>	Видалити композицію
<a href="#">/add_composition_new_playlist</a>	Створити <a href="#">плейлист</a>
<a href="#">/delete_playlist</a>	Видалити <a href="#">плейлист</a>
/menu	Повернутися до головного меню

9

## ВИСНОВКИ

1. Проаналізовано додатки аналоги.
2. [Розглянуто особливості розробки ботів на основі месенджера телеграм.](#)
3. Обрано технології та середовища розробки.
4. Розроблено music-bot для сортування композицій.
5. [Доповнено можливості месенджеру телеграм за допомогою music-боту для сортування композицій.](#)

10

# АПРОБАЦІЯ

1. Горкуша В.С.// Розробка чат-бота на основі месенджера Telegram: Матеріали першої всеукраїнської науково-технічної конференції. Збірник тез. 12.02.2021, ДУТ, м. Київ — К.: ДУТ, 2021. — С. 91.

**ДЯКУЮ ЗА УВАГУ!**