

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
(КПІ ім. Ігоря Сікорського)

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

«До захисту допущено»
В.о. завідувач кафедри БМК

Євген НАСТЕНКО

“ ___ ” _____ 2023р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Комп'ютерні технології в біології та медицині»
спеціальності 122 «Комп'ютерні науки»

на тему: Додаток для визначення психічного стану на основі
музичних вподобань

Виконала: студентка IV курсу, групи БС-91

ПАВЛУСЬ ДІАНА СВЯТОСЛАВІВНА

(прізвище, ім'я, по батькові)



(підпис)

Керівник: ст. викл. каф. БМК Корнієнко Галина Альбертівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

Консультант з розділів дипломної роботи:

доц. каф. БМК, к.т.н., Городецька Олена Костянтинівна

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ім'я, по батькові)



(підпис)

Рецензент: доцент. кафедри біомедичної інженерії, к.т.н.

Тарасова Лариса Дмитрівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студентка
(підпис)



Київ – 2023 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет біомедичної інженерії

Кафедра біомедичної кібернетики

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп’ютерні науки»

Освітньо-професійна програма «Комп’ютерні технології в біології та медицині»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри БМК

 **Євген НАСТЕНКО**
« 30 » травня 2023 р.

**ЗАВДАННЯ
на дипломну роботу студентці**

ПАВЛУСЬ ДІАНА СВЯТОСЛАВІВНА

(прізвище, ім’я, по батькові)

1. Тема роботи: **Додаток для визначення психічного стану на основі музичних вподобань**

Керівник роботи:

Корнієнко Галина Альбертівна

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені засіданням кафедри БМК від «31» травня 2023 р. № 2106-с

2. Термін подання студентом роботи **06-08 червня 2023р.**
3. Вихідні дані до роботи: *набір даних “Music and Mental Health Data*
4. Зміст роботи *Анотації (на двох мовах); Вступ; Аналітичний огляд літературних джерел (огляд літературних джерел за темою ДР, проаналізовано недоліки існуючих методів/підходів/ програмних продуктів з складання меню та графіків лікувальних раціонів, обґрунтовано обраний метод/підхід); Теоретична частина (теоретичне обґрунтування завдання / методу / алгоритму, формули, графіки); Практична реалізація задачі за темою ДР (реалізація програмного продукту, аналіз отриманих результатів, обґрунтовано доцільність використання програмного продукту / моделі / методу / алгоритму); Розрахунок економічного ефекту за темою ДР; Загальні висновки; Список використаних джерел.*
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій

тощо) 14 слайдів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата завдання	
		видав	прийняв
Дипломної роботи	Городецька О.К., доц. каф. БМК	30.05.2023	08.06.2023

7. Дата видачі завдання 30 травня 2023 р.

Календарний план

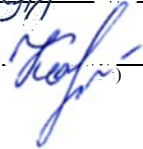
№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 15.02.2023р.	<i>виконано</i>
2	Переддипломна практика	За графіком	<i>виконано</i>
3	Виконання розділів ДР (Вступ, аналітичний огляд літературних джерел, теоретична частина)	До кінця практики	<i>виконано</i>
4	Виконання розділів ДР (практична частина, загальні висновки, список джерел)	Не пізніше 1 тижня до засідання каф-ри	<i>виконано</i>
5	Перевірка ДР науковим керівником	Не пізніше 1 тижня до засідання каф-ри	<i>виконано</i>
6	Подання в електронному вигляді ДР та анотації до неї на перевірку нормоконтролера та плагіат (UNICHECK).	---- « -----	<i>виконано</i>
7	Надання документів на засідання кафедри	За день до засідання	<i>виконано</i>
8	Предзахист ДР та допуск до захисту дисертації	Згідно плану каф.	<i>виконано</i>
9	Подання ДР рецензенту. Отримання рецензії.	До подання пакету документів до ЕК	<i>виконано</i>
10	Подання пакету документів по ДР та супровідних до неї документів до захисту в ЕК ¹	За 5 днів до дати захисту ДР за графіком	<i>виконано</i>
11	Захист ДР в ЕК		

Студент



Діана ПАВЛУСЬ

Керівник ДР



Галина КОРНІЧКО

(ім'я ПРІЗВИЩЕ)

Нормоконтролер

(підпис)

Галина КОРНІЧКО

(ім'я ПРІЗВИЩЕ)

¹ не пізніше ніж за 5 днів до затвердженої дати захисту ДР в ЕК

АНОТАЦІЯ

Дипломна робота за темою «Додаток для визначення психічного стану на основі музичних вподобань» виконана студентом кафедри біомедичної кібернетики ФБМІ Павлусь Діаною Святославівною зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 4 розділів (аналітичний огляд джерел, теоретична частина, практична частина, розрахунково-вартісний аналіз), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 15 джерел та додатків. Загальний обсяг роботи 66 сторінок.

Актуальність теми.. Розробка додатка для визначення психічного стану на основі музичних вподобань є актуальним завданням, яке відповідає потребам сучасного суспільства. Такий додаток може стати корисним інструментом для виявлення, моніторингу та покращення психічного стану користувачів.

Метою даної дипломної роботи є розробка та реалізація додатку, який буде заснований на принципах обробки даних про музичні вподобання користувача з метою визначення його психічного стану.

Використані методи: аналіз джерел, розробка архітектури додатку, вибір технологій та інструментів, реалізація функціоналу.

Результати: реалізовано додаток, що буде визначати психічний стан користувача за його музичними уподобаннями.

Ключові слова : музика, мобільний додаток, психічний стан, емоційний стан, оцінка стану, iOS, Swift.

Бібліографічний опис ДР

Павлусь Д. С. Додаток для визначення психічного стану на основі музичних вподобань : дипломна роб. бакалавра : 122 Комп'ютерні науки / Павлусь Діана Святославівна. – Київ, 2023. – 61с.

ABSTRACT

The diploma thesis titled "Application for Determining the Mental State Based on Musical Preferences" was carried out by Diana Sviatoslavivna Pavlus, a student of the Department of Biomedical Cybernetics at the Faculty of Biomedical Engineering and Computer Science, specializing in Computer Science, under the educational-professional program "Computer Technologies in Biology and Medicine." The thesis consists of an introduction, four chapters (analytical review of sources, theoretical part, practical part, cost analysis), conclusions for each of these chapters, general conclusions, a list of references comprising 15 sources, and appendices. The total length of the thesis is 66 pages.

Relevance of the topic: The development of an application for determining the mental state based on musical preferences is a relevant task that meets the needs of modern society. Such an application can become a useful tool for detecting, monitoring, and improving the mental state of users.

The aim of the thesis is the development and implementation of an application that will be based on the principles of processing data about the user's musical preferences in order to determine his mental state.

Used methods: analysis of sources, development of application architecture, selection of technologies and tools, implementation of functionality.

Results: an application was implemented that will determine the mental state of the user based on his musical preferences.

Keywords: music, mobile application, mental state, emotional state, state assessment, iOS, Swift.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД МАТЕРІАЛУ	11
РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА	17
2.4. iOS.....	21
2.5. Взаємодія інших платформ зі Spotify на основі фреймворків.....	22
2.6. Атрибути музичних творів у Spotify.....	23
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	27
3.1. Проектування та реалізація ТЗ	27
3.1.1. Технології створення застосунку.....	27
3.1.2. Архітектура застосунку	27
3.1.3. Технології та принципи, що були застосовані при створенні	
додатку.	30
3.2. Візуалізація роботи застосунку у вигляді UML діаграм.....	31
3.3. Реалізація програмного продукту.....	32
3.3.1. Створення інтерфейсу мобільного додатку	32
3.3.2. Створення функції для визначення ментального стану на базі	
отриманої вибірки пісень.	41
3.4. Графіки на статичні дані на основі БД «Music and Mental Health»	44
3.5. Розрахунок економічного ефекту	54
ЗАГАЛЬНІ ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

ПЕРЕЛІК УМОВНИХ СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

API - Application Programming Interface

Spotify API - Spotify Application Programming Interface

ARM - Advanced RISC Machine

ТЗ – Технічне завдання.

ПП – Програмний продукт.

REST - Representational State Transfer («передача репрезентативного стану») - підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів.

HTTP - HyperText Transfer Protocol - протокол передачі даних, що використовується в комп'ютерних мережах.

URI - Uniform Resource Identifier (уніфікований ідентифікатор ресурсів) - компактний рядок літер, який однозначно ідентифікує окремий абстрактний або фізичний ресурс.

SDK - Software Development Kit - набір із засобів розробки, утиліт і документації, який дає програмістам змогу створювати прикладні програми за визначеною технологією або для певної платформи.

JIT – «just-in-time» - відноситься до підходу, що використовується в компіляторі Swift для оптимізації виконання коду під час його виконання.

AOT – Ahead-of-Time – метод компіляції Swift, коли код компілюється в машинний код до його виконання.

MVC - Model-View-Controller – архітектурний шаблон, який використовується для розробки програмного забезпечення з графічним інтерфейсом користувача.

ВСТУП

«Музика — найсильніша форма магії».

Мерилін Менсон

Автор обраного епіграфу одним словом охопив суть музики. Магія. Музика має велику силу і може досягти нової глибини уяви, яку дехто вважає притулком. Музика, яку ми слухаємо, впливає на наш настрій, а наш настрій впливає на музику, яку ми обираємо.

Ця дипломна робота вносить важливий внесок у сферу психології та технологій, поєднуючи їх для створення інноваційного рішення. Сучасний ритм життя вимагає від нас постійного стресового та емоційного напруження, що негативно впливає на наше психічне здоров'я. Психологічний стан людини є одним з ключових аспектів її загального самопочуття і якості життя. У зв'язку з цим, все більше уваги приділяється розробці інноваційних методів визначення та моніторингу психічного стану, які б дозволяли ефективно виявляти та вирішувати проблеми психічного здоров'я. Останні роки характеризуються широким використанням технологій у різних сферах нашого життя. Особливо значимими є розробки в галузі мобільних додатків, які допомагають нам у вирішенні різноманітних завдань. Зважаючи на те, що музика є важливим елементом нашого щоденного життя і має сильний вплив на наші емоції та настрої, її можна використовувати як потужний інструмент для визначення психічного стану людини.

Залежно від того, що ми відчуваємо в певний момент, наш смак або жанр музики, який ми вибираємо для прослуховування, буде відрізнятися. Наприклад, коли ми відчуваємо пригнічення, ми слухаємо повільні пісні, щоб відчути цей блюз. Ми слухаємо ці поп-пісні, щоб підбадьоритися або коли нам потрібна мотивація. Ми звертаємося до пісень, наповнених спогадами, коли відчуваємо ностальгію й хочемо згадати, як добре провели час. Усі ці емоції можуть впливати на те, яку музику ми слухаємо.

У сучасному світі, де стрес, тривога і депресія стають все поширенішими

проблемами, психічне здоров'я стає надзвичайно важливою складовою нашого благополуччя. Музика завжди викликала сильні емоції у людей і використовувалася в якості засобу самовираження і релаксації. На сьогоднішній день все більше досліджень показують, що музика може мати вплив на наш психічний стан. Розробка додатка для визначення психічного стану на основі музичних вподобань є актуальним завданням, яке відповідає потребам сучасного суспільства. Такий додаток може стати корисним інструментом для виявлення, моніторингу та покращення психічного стану користувачів.

Метою даної дипломної роботи є розробка та реалізація додатку, який буде заснований на принципах обробки даних про музичні вподобання користувача з метою визначення його психічного стану. Додаток буде використовувати алгоритми машинного навчання та аналізу даних для автоматичного визначення емоційного стану користувача на основі його музичних вподобань. Це дозволить користувачам зрозуміти, як музика впливає на їхні емоції та допоможе їм контролювати свій психічний стан.

Завдяки даному додатку, користувачі змогли б отримувати персоналізовані рекомендації щодо музичних композицій, які можуть поліпшити їхній настрій та зменшити стресові стани. Також, враховуючи ритм життя сучасної людини, який часто пов'язаний з використанням мобільних пристроїв, мобільний додаток стає зручним і доступним засобом для надання підтримки та допомоги будь-коли та будь-де.

Результати цієї дипломної роботи можуть сприяти подальшому розвитку інноваційних рішень у сфері мобільних додатків для психічного здоров'я та допомогти людям краще розуміти себе, контролювати свої емоції та покращувати своє психічне благополуччя. Додаток, заснований на визначенні психічного стану на основі музичних вподобань, може стати цінним інструментом для самомоніторингу та саморегуляції емоцій.

Об'єктом дослідження виступає процес визначення психо-емоційного стану користувача.

Предметом дослідження виступає автоматизований процес визначення психо-емоційного стану користувача.

Вихідними даними є база «Music and mental health». Вивчення теми велося на основі статей та досліджень, взятих з різних світових джерел через інтернет.

Мовою програмування було обрано Swift, фреймворк UIKit, щоб створити сучасний застосунок для IOS.

Робота не має практичного застосування, проте максимально показує вміння *Павлусь Діани Святославівни* володіти спеціальними (фаховими) компетентностями, загальними компетентностями та отримані програмні результати навчання згідно ОПП відповідного року набору здобувачів.

Апробація результатів за темою дипломної роботи на практику не заплановано.

Публікації не заплановано.

Дипломна робота за темою «Додаток для визначення психічного стану на основі музичних уподобань» виконана студентом *Павлусь Діаною Святославівною* зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині», побудована за класичним типом та викладена на 61 сторінці машинописного тексту. Вона складається з: вступу; 3 розділів(аналітичний огляд матеріалу, теоретична частина, практична частина); висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який містить 15 джерел. У роботі представлено 23 рисунки.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД МАТЕРІАЛУ

Для того, щоб краще вникнути в тематику роботи, вівся огляд матеріалів та досліджень, пов'язаних з проблематикою, що розглядається, та зі схожим функціоналом. Було розглянуто музику як один з факторів, що є індикаторами психічного стану людини.

Вплив музики на емоційний стан людини є однією з ключових тем у дослідженнях, пов'язаних з психологією та музикотерапією. Деякі дослідження показують, що різні жанри музики можуть викликати різні емоційні реакції: весела та жвава музика може спричиняти радість та підвищення настрою, тоді як меланхолійна або сумна музика може викликати смуток або ностальгію. Це працює і навпаки: емоції, які переживає людина, впливають на вибір музичних творів для прослуховування [25 , 24].

Нижче наведена таблиця (табл. 1.1.), яка включає деякі популярні жанри музики та емоції, які вони зазвичай викликають у слухачів. Варто зазначити, що це загальні сприйняття та можуть бути варіації в індивідуальних реакціях на музику.

Таблиця 1.1

Жанри музики та емоції, що вони викликають

Жанр музики	Емоції, викликані ним
Поп	Радість, енергія, веселощі, настрої
Рок	Енергія, сила, азарт, драйв
Джаз	Релакс, настрої, емоційна глибина
Класична	Спокій, витонченість, естетика
Метал	Енергія, напруженість, експресія
Хіп-хоп	Самовираз, сила, ритм, активність
Фолк	Поезія, меланхолія, природність

Ця таблиця - це лише загальний огляд деяких існуючих жанрів музики та їх потенційного впливу на емоційний стан слухачів. Варто пам'ятати, що кожна людина може мати індивідуальну реакцію на певний жанр музики в залежності від свого смаку, життєвого досвіду та настрою [17, 18].

Взаємовплив між музикою та психо-емоційним станом людини доводить «Дослідження кореляції між музичними перевагами та психічним здоров'ям» студентів коледжу на базі трьох всесвітньо-відомих університетів (1 - Монгольський університет, Улан-Батор, Монголія; 2 - Південний університет науки і технологій, Шеньчжень, Китай; 3- Університет Dhurakij Pundit, Бангкок, Таїланд).

1.1. Дослідження 1

Дослідження кореляції між музичними перевагами та психічним здоров'ям на базі трьох університетів.

Теоретичною основою була взята теорія «Стимул-Організм-Реакція», де йдеться про те, що реакція індивідів викликається їхнім внутрішнім емоційним станом після стимуляції. Теорія стверджує, що більшість змін у навколишньому середовищі діють як стимул для індивіда, що призводить до передачі інформації до нервової системи індивіда, що призводить до реакції м'язів або психології. Дослідники прийняли цю теорію для того, щоб визначити кореляцію між музичними перевагами та психічним здоров'ям студентів коледжу, і розділили дослідження на дві задачі:

- а) дослідити зв'язок між музикою та індивідуальною психологією
- б) зв'язок між музичним середовищем та індивідуальними намірами щодо поведінки.

Музика є одним із стимулів, і слухачі віддають перевагу музиці, коли її стимуляція відповідає їхньому психічному стану. Крім того, музичні переваги між емоціями людини та психічним здоров'ям також пов'язані [19].

Дослідниками була виконана така робота: дослідницький факторний аналіз поп-музики, західної класичної, китайської традиційної та важкої музики (проводився після анкетованого опитування триста восьмидесяти студентів про їх музичні переваги); аналіз надійності; аналіз даних; описова статистика та кореляційний аналіз.

Дане дослідження показало, що музичні уподобання студентів коледжу щодо різних форм музики по-різному корелюють з їхнім психічним здоров'ям, та зробило три важливі внески:

Протягом останніх років було проведено обмежені емпіричні дослідження зв'язку між музичними перевагами та психічним здоров'ям шляхом тестування або проведення клінічних випробувань, однак дане дослідження включало анкетування для обговорення та статистичного аналізу.

Шкали музичних переваг, прийняті в попередніх емпіричних дослідженнях, відрізнялися через культурні відмінності між китайськими та західними країнами. Однак локалізована китайська шкала музичних переваг, прийнята в цьому дослідженні, допомогла краще зрозуміти музичні переваги саме китайських студентів.

Більшість попередніх досліджень проводилися в західному контексті, тоді як у цьому дослідженні вивчали кореляцію між музичними перевагами китайських студентів коледжу та їхнім психічним здоров'ям, взявши за вибірку китайських студентів коледжу та в контекст китайської культури.

Проте це вивчення даної теми має і недоліки. По-перше, воно базувалося на перехресних даних, і через це неможливо зробити висновок про причинно-наслідковий зв'язок. Але більш важливою є друге - вибірка дослідження. За основу були взяті дані про студентів коледжу, відібрані методом зручної вибірки; тому результати слід обережно узагальнювати на інші зразки. Цю проблему можна вирішити, зробивши реєстрацію користувача та запам'ятовуючи його уподобання. Тобто при реєстрації користувачеві потрібно буде вказувати базові дані про себе(країна/регіон проживання, вік,

дані про соціальний статус(студент/працівник/школяр/пенсіонер/безробітній), орієнтовна сфера музичних уподобань). Далі вже при безпосередньому користуванні додатком він буде ставити свою оцінку пісень, які прослухав. На основі результатів його уподобань додаток буде робити висновки та виводити йому на екран результат про те, в якому стані користувач на даний момент. Дані про уподобання одного користувача будуть зберігатись в пам'яті додатку протягом деякого часу, тому це дозволить зробити незалежну від країни, регіону, віку чи інших факторів вибірку і брати її як основи для нових досліджень у сфері впливу музики на психічний стан людини [1].

1.2. Дослідження 2

Дослідження про зв'язок психічного здоров'я та музичних переваг студентів на базі школи-інтернату MRSM TAIPING.

Ще одне дослідження було проведене студентом малайзійської школи Завані Мохд Таджудіном при написанні бакалаврської роботи. Його метою було визначення музичних уподобань та стану психічного здоров'я старшокласників і дослідження зв'язку між цими визначеними даними. У дослідженнях психології музичні уподобання є одним із аспектів, які вивчаються, щоб отримати повне розуміння соціальної психології. Попередні дослідження були більше зосереджені на зв'язку між музичними уподобаннями та особистістю, але останні дослідження розширили погляд на зв'язок стану психічного здоров'я з музичними уподобаннями. Підлітки, як правило, більш схильні до впливу музики та емоційних розладів через гормональний дисбаланс і фактори навколишнього середовища, що свідчить про можливе дослідження для глибшого вивчення музичних уподобань та їх зв'язку зі станом психічного здоров'я.

У цьому дослідженні взяли участь 386 старшокласників. Усі студенти з Мактаб Ренда Сейнс МАРА (MRSM), Тайпін, Перак. Роздатковий матеріал, що складається з демографічних даних і двох анкет, шкала депресії, стресу та

тривоги (DASS) і переглянутий короткий тест музичних уподобань (STOMP-R) були роздані кожному студенту.

У результаті дослідження було виявлено, що три найпоширеніші музичні жанри, яким віддають перевагу, це релігійний жанр ($5,21 \pm 1,88$), поп-жанр ($5,15 \pm 1,93$) і саундтрек/тематична пісня ($5,10 \pm 1,97$). Рівень депресії та стресу оцінено як нормальний у 138 зразках (35,3%) та 156 зразках (39,9%) відповідно, тоді як 37,9% вибірки (148 зразків) оцінено як сильну тривогу. Ранговий кореляційний тест Спірмена використовувався для аналізу зв'язку між музичними уподобаннями та станом психічного здоров'я. З'ясувалося, що рівень депресії достовірно пов'язаний із жанром панк ($p < 0,05$), тоді як класичний жанр також статистично релевантний рівню тривоги ($p < 0,05$). Цікаво, що є кілька жанрів, які були значущими з рівнем стресу, а саме блюз, класика, зарубіжна музика, джаз та старі ($p < 0,05$).

Висновком було вказано наступне. Музичні уподобання підлітків можуть допомогти передбачити їхній емоційний стан, щоб краще їх зрозуміти. Важливо було втручатися в ці ймовірні проблеми психічного здоров'я, щоб запобігти будь-яким провокуючим факторам, які можуть призвести до появи діагнозу психічного розладу в майбутньому.

Дане дослідження підтверджує той факт, що на основі музичних уподобань ми можемо та маємо право визначати, у якому емоційному стані знаходиться зараз людина. Проте воно обмежене по віку – вивчався взаємозв'язок між музикою та лише підлітками [2, 3].

Висновок до розділу 1

У даному розділі виконано аналіз двох досліджень, що доводять взаємовплив між музикою та психо-емоційним станом людини. Обидва показали те, що за музикою, прослуханою людиною, можна визначати, у якому стані вона знаходиться. Також опрацьовано статті про вплив музики на емоційний стан та про визначення психічного стану людини за допомогою

даних про її музичні уподобання.

Розглянуті дослідження дають нам право визначати стан користувача майбутнього додатку на основі музики, яку він уподобає. Недоліки дослідження (в першому – обмеження по регіону, у другому – по віку) ніяк не впливають на постановку головної задачі нашого проекту, яка лишається незмінною: додаток на основі музичних творів, які відкликаються користувачеві за його настроєм на даний момент часу, має визначати стан користувача і відразу ж його транслювати на екран. Проте на основі недоліків розглянутих досліджень було прийнято рішення про врахування деяких деталей проекту для покращення його архітектури та функціоналу, що буде розроблятися. Варто підкреслити, що аналіз вищенаведених вивчених матеріалів довів актуальність миттєвої оцінки стану користувача, тому що за допомогою функції визначення стану в реальному часі додаток буде ідеально підходитиме для покращеного самоаналізу та самоконтролю.

Отже, основними завданнями будуть:

1. Проаналізувати вже наявні дані про музичні вподобання користувачів.
2. Розробити алгоритми машинного навчання для визначення психічного стану на основі музичних вподобань.
3. Реалізувати додаток для визначення психічного стану на основі музичних вподобань: На основі розроблених алгоритмів буде створений мобільний додаток, який зможе аналізувати музичні вподобання користувачів і визначати їхній психічний стан. Додаток буде мати зручний інтерфейс, що дозволить користувачам легко використовувати його і отримувати інформацію про свій психічний стан у режимі реального часу.
4. Провести оцінку ефективності додатку: після розробки додатку буде проведено дослідження економічної ефективності.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА

Реалії сучасності музичного ландшафту наповнені безліччю якісно розроблених та популярних платформ для прослуховування музики, і Spotify - одна з них. Існують і інші, що користуються актуальністю. Наприклад, Apple Music, YouTube Music, Amazon Music та Deezer - ось деякі з основних конкурентів Spotify. Кожна з цих платформ має свої унікальні особливості та пропозиції для користувачів. Вони надають доступ до великої кількості музики, створюють персоналізовані рекомендації, пропонують ексклюзивний контент та взаємодіють з різними пристроями та платформами. Цей широкий вибір музичних платформ сприяє зростанню конкуренції та стимулює розвиток нових технологій та функцій, які полегшують користувачам знаходження, прослуховування та спільного ділінгу музики. Водночас, користувачі мають можливість вибирати серед різних платформ та знаходити ту, яка найкраще відповідає їхнім потребам та вподобанням [20].

Дані, на яких ґрунтуються дослідження даної роботи, отримуються прямо від Spotify платформи. У даному розділі буде розглянуто Spotify як платформу з багатьма можливостями, даними, дослідженнями, алгоритмами та потужним функціоналом. Також буде наведено, яким чином можна підключатися прямо до Spotify, створювати додатки прямо на цій платформі та використовувати розробки платформи для своїх інтеграцій.

2.1 Музична платформа Spotify.

Spotify – це цифровий музичний потоковий сервіс, що дозволяє легально прослуховувати музичні композиції, не скачуючи їх на пристрій. Він надає вам миттєвий доступ до величезної онлайн-бібліотеки музики, подкастів, аудіокниг та відео, дозволяючи слухати будь-що на ваш вибір і у

будь-який час, та має безкоштовну та преміум версії. Популярність Spotify доводить його статистика: на сервіс припадає понад 36% світового аудіострімінгу. У серпні 2021 року Spotify мав понад 365 мільйонів активних користувачів по всьому світу, понад 165 мільйонів з них – платні передплатники [4].

Основними можливостями Spotify, що приваблюють користувачів, є:

1. Стрімінг музики: Spotify дозволяє користувачам прослуховувати мільйони треків онлайн. Користувачі можуть шукати музику за жанром, виконавцем, альбомом та створювати свої власні плейлисти.

2. Персоналізовані рекомендації: Spotify використовує алгоритми машинного навчання, щоб рекомендувати користувачам нову музику, враховуючи їхні музичні вподобання, історію прослуховування та популярні треки серед схожих слухачів.

3. Соціальні можливості: Spotify дозволяє користувачам ділитися своїми улюбленими треками, плейлистами та альбомами з друзями. Користувачі можуть також слідкувати за активністю своїх друзів і відкривати нову музику через їх рекомендації.

4. Офлайн-режим: Spotify пропонує можливість завантажити музику для прослуховування в офлайн-режимі. Це дозволяє користувачам слухати музику без доступу до Інтернету, що особливо зручно під час подорожей або в обмежених мережевих умовах.

5. Підписка та безрекламне прослуховування: За підпискою на Spotify Premium, користувачі можуть насолоджуватись музикою без реклами, пропускати пісні, відтворювати музику у вищій якості звуку та мати безліч додаткових можливостей.

6. Інтеграція з різними пристроями та платформами: Spotify доступний на різних платформах, включаючи веб-браузер, комп'ютери, медіа системи автомобілів, смартфони та планшети. Ви також можете використовувати Spotify на різних пристроях одночасно і синхронізувати свою музичну бібліотеку та плейлисти.

7. Артисти та подкасти: Spotify не обмежується музичними треками, але також надає платформу для артистів та подкастерів. Артисти можуть завантажувати свою музику на Spotify, створювати свої власні артистичні профілі та отримувати дохід від стрімінгу своїх треків. Подкастери можуть також публікувати свої подкасти на платформі Spotify та досягати аудиторії своїх слухачів. Spotify надає інструменти та аналітику для артистів та подкастерів, щоб вони могли взаємодіяти зі своєю аудиторією та відстежувати популярність своїх робіт [5].

Загалом, Spotify є найбільшим та найпопулярнішим сервісом потокової трансляції музики у світі. Ця музична платформа поєднує в собі широкий вибір музики, персоналізовані рекомендації, соціальні можливості, офлайн-режим та підтримку для артистів та подкастерів. Вона стала однією з провідних платформ для стрімінгу музики та подкастів у світі.

2.2. API

Programming Interface(далі API) - це правила, за допомогою яких програмні додатки можуть спілкуватися та взаємодіяти один з одним. API визначає методи, формати даних та правила, які розробники можуть використовувати для доступу та взаємодії з функціональністю, що надається програмним сервісом, бібліотекою або платформою.

API дозволяє розробникам використовувати функції та дані іншого програмного додатку без необхідності знати деталі його внутрішньої реалізації. Використання API дозволяє прискорити розробку програмного забезпечення, забезпечити інтеграцію між різними додатками та підтримувати взаємну взаємодію між ними. API може бути використаний для отримання даних, відправки команд, виконання операцій та багатьох інших функцій [6].

2.3. Spotify API

Spotify Application Programming Interface (далі Spotify API) - це набір інструментів, протоколів та документації, які надають розробникам можливість взаємодіяти з музичним сервісом Spotify та отримувати доступ до різних функцій і даних. Він дозволяє створювати програми, веб-сайти або мобільні додатки, які можуть отримувати доступ до музичного контенту, створювати плейлисти, отримувати дані про виконавців, альбоми, треки та багато іншого.

Spotify API працює на основі HTTP-запитів і використовує формати даних, такі як JSON (JavaScript Object Notation). Розробник може зареєструвати свою програму на Spotify Developer Dashboard, отримати API ключі, а потім використовувати ці ключі для аутентифікації та авторизації запитів до Spotify API [15].

API, що розглядається, надає широкий спектр функцій, включаючи:

1. Пошук: Розробники можуть шукати артистів, альбоми, треки, плейлисти або відбирати музику за різними критеріями, такими як жанр, рік випуску тощо.
2. Відтворення музики: Розробники можуть керувати програвачем Spotify, відтворювати, призупиняти, перемотувати треки, змінювати гучність тощо.
3. Доступ до плейлистів та бібліотеки користувача: Розробники можуть отримувати доступ до плейлистів, створювати нові плейлисти, додавати треки або видаляти їх з плейлистів.
4. Отримання деталей виконавців та альбомів: Розробники можуть отримувати інформацію про виконавців, альбоми, треки, обкладинки альбомів та інші деталі.
5. Аналітика: API надає можливість отримати аналітичні дані, такі як потокові треки, відтворення, рейтинги пісень, тенденції слухання та інші метрики.

6. Аудіо: API може надавати доступ до аудіофайлів, які ви можете стрімити або завантажити, в залежності від функціональності API.

7. Пошук: API може мати функцію пошуку, яка дозволяє здійснювати пошук виконавців, альбомів, треків або інших даних за певними критеріями, такими як назва, жанр, рік випуску тощо.

8. Синхронізація: API може дозволяти синхронізувати дані користувача, такі як збережені треки, плейлисти або рекомендації між пристроями або платформами [16].

Також інші деталі: API може містити інші деталі про виконавців і альбоми, такі як біографічні дані, списки треків, тексти пісень, додаткові зображення, відео тощо. Важливо зазначити, що доступність певних деталей може варіюватися в залежності від API та правил використання відповідного сервісу музичного контенту [7].

2.4. iOS

iOS є мобільною операційною системою, розробленою компанією Apple. Вона використовується на пристроях iPhone, iPad та iPod Touch. iOS відома своєю швидкістю, стабільністю та високою безпекою.

Основні риси iOS:

1. Інтуїтивний і привабливий інтерфейс: iOS володіє простим та зручним для використання інтерфейсом зі стильними графічними елементами та мінімальними перешкодами для користувачів.

2. Широкі можливості додатків: App Store містить великий вибір додатків, які охоплюють різні галузі, включаючи соціальні мережі, ігри, продуктивність, здоров'я та фітнес, освіту та багато іншого. Розробники мають доступ до різноманітних інструментів та ресурсів для створення інноваційних додатків для iOS.

3. Висока безпека: Apple приділяє велику увагу безпеці своїх пристроїв та операційних систем. iOS використовує різні заходи безпеки, такі як шифрування даних, аутентифікація обличчя та відбитком пальця, додаткові контролю доступу та інші заходи, щоб забезпечити захист інформації користувачів.

4. Інтеграція з екосистемою Apple: iOS гармонійно взаємодіє з іншими продуктами та послугами Apple, такими як iCloud, Apple Music, Apple Pay, Siri та інші. Це дозволяє користувачам отримати синхронізацію даних, спільний доступ до медіа, безпечні платежі та інші зручні функції.

Apple має широку екосистему продуктів, включаючи iPhone, iPad, Apple Watch і Apple TV. Використання Swift, UIKit та iOS дозволить вам розробити додаток, який працюватиме на різних пристроях Apple, що забезпечить вам більше можливостей для розширення своєї аудиторії при умові, що сам код майже не буде відрізнятися для використання на iPhone, iPad та MacBook (на базі процесорів ARM). Це стає можливим через принцип програмування Auto Layout який ми будемо використовувати [8].

2.5. Взаємодія інших платформ зі Spotify на основі фреймворків

2.5.1 Фреймворк та його призначення

Фреймворк - це набір готових компонентів, бібліотек, інструментів та правил, які надають розробникам загальну структуру і платформу для розробки програмного забезпечення. Він надає основу для розробки додатків, пропонуючи вбудовані функції та рішення для спільних задач [9].

2.5.2 Фреймворки для взаємодії зі Spotify API на платформі iOS

Система iOS була обрана для розробки додатку на ній, розглянемо, яким чином можна це зробити.

Для взаємодії зі Spotify API на платформі iOS існує кілька популярних фреймворків. Найпоширенішими є:

1. Spotify iOS SDK: Це офіційний SDK, наданий Spotify. Він надає можливості для авторизації користувача, пошуку пісень, створення плейлистів

та відтворення музики. Ви можете знайти його на офіційному сайті Spotify для розробників.

2. Alamofire: Це популярний фреймворк для мережевого взаємодії на iOS. Ви можете використовувати його для здійснення HTTP-запитів до Spotify API та обробки отриманих даних. Ви можете дізнатися більше про нього на офіційному сайті.

3. SwiftyJSON: Цей фреймворк допомагає спростити роботу з JSON-даними, які ви отримуєте від Spotify API. Він надає зручні методи для отримання даних з JSON-об'єктів і робить їх легко доступними для подальшого використання. Ви можете знайти SwiftyJSON на GitHub.

4. AlamofireObjectMapper: Цей фреймворк є розширенням до Alamofire і дозволяє зручно мапити JSON-відповіді на моделі даних у форматі Swift. Він спрощує роботу з отриманими даними і дозволяє легко використовувати їх у вашому додатку. Ви можете знайти AlamofireObjectMapper на GitHub [10].

2.6. Атрибути музичних творів у Spotify

Кожен трек в Spotify має ряд атрибутів, які описують його характеристики. Деякі з основних атрибутів музичних творів у Spotify включають:

1. Назва треку.
2. Виконавець.
3. Альбом(назва).
4. Тривалість.
5. Жанр.
6. Рік виходу.
7. Популярність.
8. Темп.
9. Настрій.

10. Інструментальність.
11. Танцювальність.
12. Акустичність.
13. Енергія.
14. Тональність.
15. Гучність.
16. Приблизний тактовий розділ(метр).
17. Режим.
18. Мовність.
19. Валентність [11 , 12].

Ці атрибути допомагають користувачам Spotify шукати, відтворювати та оцінювати музику відповідно до їхніх вподобань.

2.7. Spotify Web API

Spotify Web API - це програмний інтерфейс застосунку (API), який надає доступ до функцій та даних Spotify для розробників. Він дозволяє розробникам створювати додатки, сервіси та інтеграції, які взаємодіють з музичними даними Spotify. За допомогою Spotify Web API розробники можуть отримувати доступ до різноманітних функцій, таких як:

1. Отримання інформації про користувача.
2. Пошук та отримання музичних даних.
3. Управління плейлистами.
4. Відтворення музики [13].

Основна мета Spotify Web API полягає в тому, щоб дозволити розробникам створювати сервіси, які можуть взаємодіяти з музичною бібліотекою Spotify, шукати пісні, отримувати дані про виконавців, альбоми, плейлисти та іншу інформацію, створювати та редагувати плейлисти користувачів, контролювати програвання музики та інше [21].

API забезпечується за допомогою REST-протоколу (Representational

State Transfer), який використовується для комунікації між клієнтом (розробником додатка) та сервером Spotify. Розробники можуть робити HTTP-запити до відповідних ендпоінтів (URL-адреси) з вказанням необхідних параметрів для отримання бажаних даних або виконання певних дій [14 , 22].

2.8. Програмний інтерфейс застосунку: обґрунтування

2.8.1 Мова програмування Swift

На основі поставлених цілей додатку було обрано писати платформу для iOS на Swift. Swift є сучасною мовою програмування, розробленою спеціально для роботи з платформами Apple, такими як iOS, macOS, watchOS і tvOS. Вибір даної мови для написання додатку базувався на кількох аргументах:

1. Легкість навчання та використання.
2. Безпека.
3. Швидкодія.
4. Широкі можливості.
5. Інтеграція з платформою Apple.

2.8.2 Фреймворк UIKit

UIKit - це фреймворк, розроблений компанією Apple для побудови інтерфейсів користувача в додатках для iOS та macOS. Він є частиною набору розробки програмного забезпечення (SDK, про який було зазначено у розділі 2.5.2) від Apple і використовується головним чином для створення додатків для iPhone, iPad та Mac [23].

2.8.3 Програмування за допомогою Auto Layout

Auto Layout є одним з ключових інструментів для розробки інтерфейсу користувача на платформі iOS з використанням Swift та UIKit. Він дозволяє створювати гнучкі і респонсивні інтерфейси, які адаптуються до різних розмірів екранів та пристроїв.

Основні поняття Auto Layout: Constraints (обмеження):

1. Constraints визначають відносини між різними елементами інтерфейсу.
2. Автоматичні обмеження (Automatic Constraints).
3. Підходи до розміщення елементів.
4. Різні види обмежень.

Висновок до розділу 2

Підсумовуючи, у теоретичному розділі були розкриті питання стосовно музичної платформи, з якої далі буде пов'язаний додаток, фреймворків для взаємодії з нею та програмного інтерфейсу, що надає доступ до даних для розробників. Також були розглянуті:

- атрибути музичних творів, які прив'язані до них автоматично на платформі та які будуть використані для обробки даних про користувача та виведення інформації про його стан;
- система iOS та її переваги для написання застосунку саме для неї;
- фреймворки для взаємодії Spotify з iOS;
- вибір деталей для програмного інтерфейсу застосунку та їх обґрунтування

і доведено причини та мету дослідження саме з обраними характеристиками.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1. Проектування та реалізація ТЗ

Ідея застосунку полягає у визначенні ментального стану користувача. Це виконується, надаючи користувачу 10 пісень, між якими він може обрати, подобаються вони чи ні із можливістю пропуску пісні, не враховуючи її для результату, якщо користувач вважає, що пісня просто не імпонує йому. Після створення вибірки зі 10 пісень, застосунок обробить цю інформацію, і далі користувач зможе дізнатись свій ментальний стан. В користувача буде можливість зберегти його або ж не робити цього. Паралельно з цим він в будь-який час зможе подивитись свою статистику в конкретний час, день та місяць, коли він проходив тест.

3.1.1. Технології створення застосунку

Мовою програмування було обрано Swift, фреймворк UIKit для того, щоб створити сучасний застосунок для IOS. UIKit є основним фреймворком для розробки користувацького інтерфейсу для iOS додатків.

3.1.2. Архітектура застосунку

Архітектурою нашого застосунку було обрано Model-View-Controller (далі MVC). Це шаблон, який використовують користувачі для розробки ПЗ з графічним інтерфейсом. З його допомогою можна розділити логіку програми, представлення даних та взаємодію користувача на трьох основних компонентах: моделі (Model), представлення (View) та контролери (Controller).

Основними перевагами використання MVC є:

- Розділення обов'язків: MVC дозволяє розділити логіку програми на окремі компоненти, що спрощує розробку, тестування та підтримку коду.
- Покращена перевикористовуваність: Компоненти MVC можуть бути

- перевикористані в інших частинах програми або навіть в інших проектах, оскільки вони слабо залежать один від одного.
- Легше супроводження: Завдяки чіткому розділенню логіки, внесення змін або вдосконалення в одному компоненті не впливає на решту системи.
 - Підтримка паралельного розвитку: Команда розробників може працювати над моделями, представленнями та контролерами незалежно одна від одної, що сприяє швидкому розвитку.
 - Використання архітектури MVC рекомендується для розробки додатків з графічним інтерфейсом користувача, оскільки вона забезпечує добру організацію коду та полегшує розширення та підтримку програми в майбутньому.

Таким чином, структура нашого проекту виглядає наступним чином (рис3.1.):

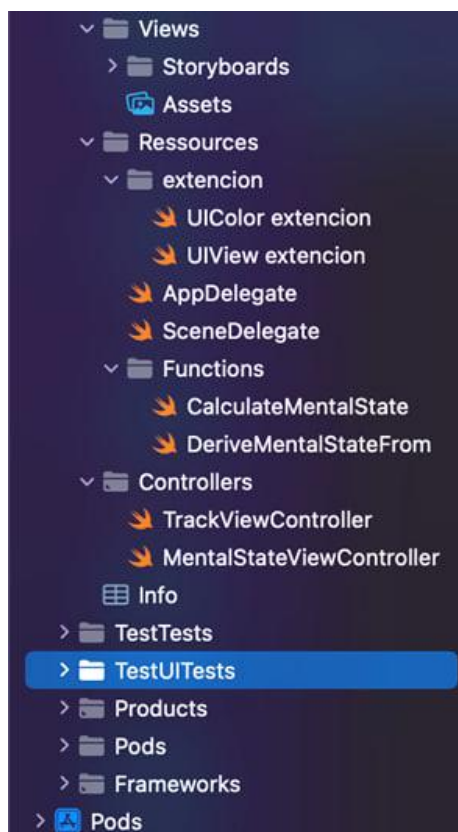


Рисунок. 3.1 Структура проекту застосунка

На рис. 3.1 представлено структуру проекту, що побудовано на обраною нами архітектурою.

У додатку використовується API Sptotify, через те, що збірка з піснями побудована таким чином, що на кожну пісню є її ID у Sptotify. Таким чином нам необхідно створити механізм, коли ми отримуватимемо картинку та вмикатимемо пісні через цей id..

Для цього нам треба виконати наступні кроки:

Крок 1: Реєстрація додатку в Spotify Developer Dashboard.

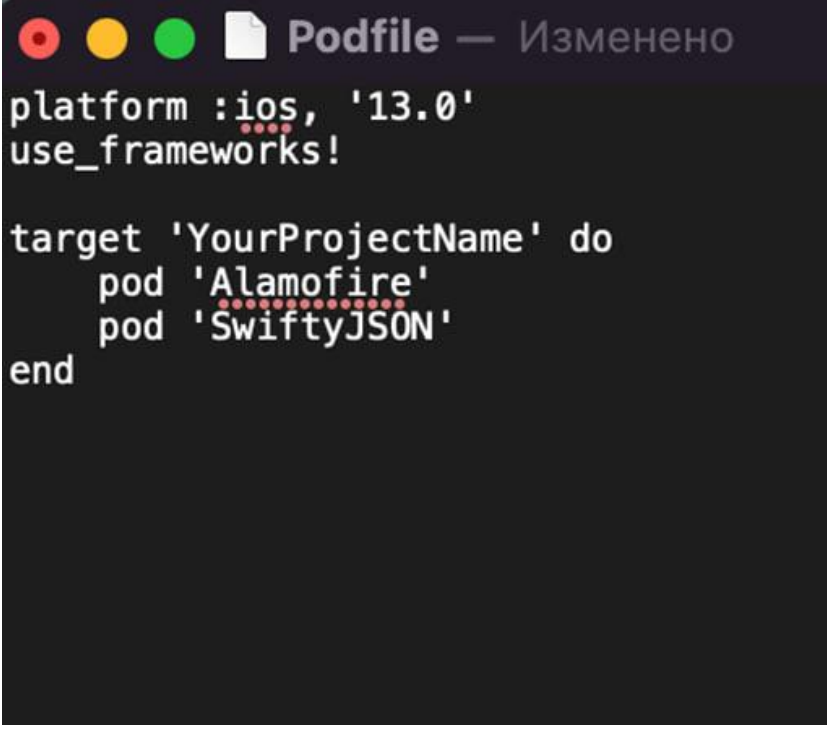
1. Створюємо обліковий запис за посиланням <https://developer.spotify.com/dashboard/>

2. Створюємо новий додаток, натиснувши на кнопку "Create an App". Заповнюємо необхідну інформацію про додаток, включаючи назву, опис та URL-адресу перенаправлення.

3. Після створення додатку отримаємо клієнтський ідентифікатор та секретний ключ. Зберігаємо ці дані, оскільки вони знадобляться нам пізніше.

Крок 2: Додавання Alamofire та SwiftyJSON до проекту Для цього нам треба написати наступну команду у терміналі `cd ./ (Назва папки проекту)` Після цього треба в директорії нашого проекту прописати наступну команду `pod init`

Далі в директорії нашого проекту з'явиться `podfile`. Він має виглядати наступним чином (рис.3.2.):



```

platform :ios, '13.0'
use_frameworks!

target 'YourProjectName' do
  pod 'Alamofire'
  pod 'SwiftyJSON'
end

```

Рисунок 3.2 - Podfile проекту

На рис. 3.3 представлено вигляд, що має Podfile нашого проекту.

Зберігаємо зміни у "Podfile" та запускаємо команду `pod install`. У терміналі в директорії нашого проєкт. Після успішного встановлення Alamofire та SwiftyJSON, відкриваємо файл проєкт з розширенням `.xcworkspace`.

Крок 3: Написання коду для підключення до API Spotify та вмикання треків.

3.1.3. Технології та принципи, що були застосовані при створенні додатку.

1. Auto Layout

Auto Layout є одним з ключових інструментів для розробки інтерфейсу користувача на платформі iOS з використанням Swift та UIKit. Він дозволяє створювати гнучкі і респонсивні інтерфейси, які адаптуються до різних розмірів екранів і пристроїв.

2. Замикання (closure)

Ми можемо передати замикання, яке виконує додаткові дії після того, як `MentalStateViewController` закриється. Наприклад, якщо ми зберігаємо дані після закриття `MentalStateViewController`, ми можемо викликати функцію збереження в `dismissCompletion`, в нашому випадку це треба для того, щоб передати до `TrackViewController` те, що користувач побачив результати проаналізував його за хоче закрити вікно щоб спробувати ще раз.

3.2. Візуалізація роботи застосунку у вигляді UML діаграм

Далі на рисунку буде представлено процес створення вибірки пісень та передачі її наступному класу для подальшого аналізу (рис.3.4.).

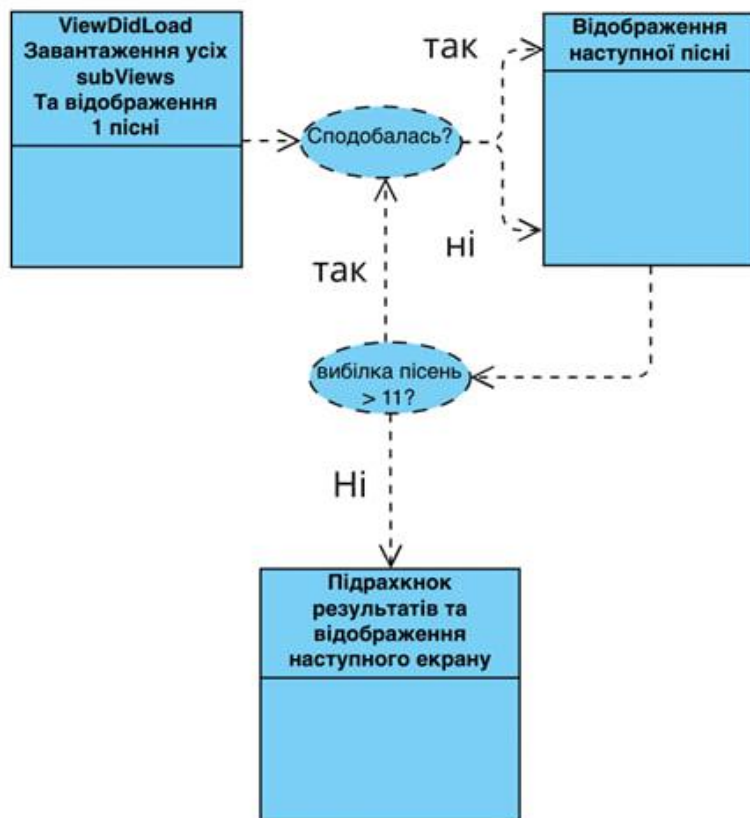


Рисунок 3.3 - Декомпозиція процесу створення вибірки пісень

На рис. 3.4 представлено процесу аналізу вибірки пісень, відображення

результату та вихід з екрана з варіантом зберегти чи ні результату користувачем.

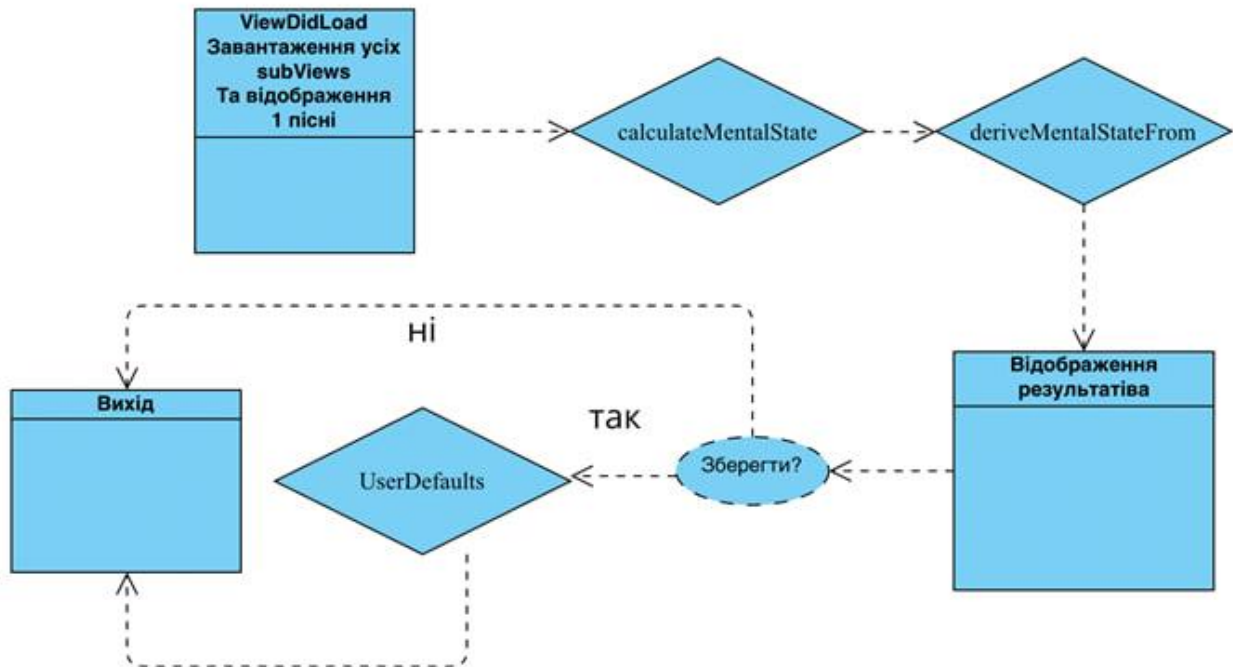


Рисунок 3.4. - Декомпозиція процесу аналізу вибірки пісень

3.3. Реалізація програмного продукту

3.3.1. Створення інтерфейсу мобільного додатку

Інтерфейс визначає спосіб взаємодії користувача з програмним забезпеченням, його зовнішній вигляд і способи навігації. Якщо інтерфейс додатку є зручним, привабливим і легким у використанні, це забезпечує приємний досвід для користувачів і підвищує ймовірність того, що вони будуть продовжувати використовувати додаток.

У світі, де прагнуть зробити все для всіх, універсальність стає тим, чого прагнуть досягти у всьому. Тому було обрано нейтральний інтерфейс застосунку як ідеальне рішення, яке слугуватиме на забезпечення рівних можливостей для всіх користувачів без зайвих перешкод або викликів,

прагнучи уникнути стереотипних асоціацій, універсалізувати дизайн і забезпечити його доступність і зрозумілість для всіх.

Вибір нейтрального чорно-біло-сірого дизайну для додатку, що визначає психічний стан на основі музичних смаків, має декілька аргументів:

- **Об'єктивність і нейтральність:** Чорно-біла-сіра палітра кольорів вважається нейтральною і не викликає сильних емоцій або асоціацій у багатьох людей. Це дозволяє уникнути кольорових перекручень, які можуть вплинути на об'єктивність оцінки психічного стану.
- **Простота і зрозумілість:** Використання нейтрального дизайну полегшує сприйняття інформації користувачем. Важливо, щоб додаток був легко зрозумілим і інтуїтивно зрозумілим для широкого кола користувачів, незалежно від їхнього досвіду з технологіями чи психологією. Чорно-біла-сіра гамма кольорів не відволікає увагу і дозволяє зосередитися на головній інформації.
- **Професійний вигляд:** Чорно-білий дизайн часто асоціюється з професійністю, серйозністю і стабільністю. Користувачі можуть бути більш схильні довіряти додатку, який має професійний вигляд, оскільки це створює враження надійності і добре продуманої розробки.
- **Універсальність:** Нейтральний чорно-білий дизайн може бути прийнятним для різних культур і груп користувачів. Кольори можуть мати різні емоційні асоціації у різних культурах, і використання нейтральних кольорів уникає можливих конфліктів або неправильних тлумачень.
- **Висвітлення інформації:** Використання нейтрального дизайну дозволяє ефективно виділяти важливі елементи інтерфейсу додатку. Наприклад, важлива інформація або елементи керування можуть бути виділені кольоровими акцентами або підсвічуванням, що залучає увагу користувача.

Загалом, нейтральний чорно-білий дизайн для додатку для визначення психічного стану на основі музичних смаків допомагає забезпечити об'єктивність, простоту, професійність, універсальність та висвітлення

важливої інформації. Враховуючи основну мету додатку - аналіз психічного стану, нейтральний дизайн може бути вдалим вибором для створення ефективного та зрозумілого інтерфейсу для користувачів.

Нейтральність зосереджується на важливому аспекті дизайну застосунку, який спрямований на створення універсального інтерфейсу, що підходить для широкого спектру користувачів. Нейтральний інтерфейс враховує різноманітність користувачів з різними потребами, вміннями та вподобаннями. Один з ключових принципів - простота. Мінімалістичний дизайн і мінімальна кількість складних деталей дозволяють зрозуміти додаток без зайвих зусиль, не розпиляти зорову увагу на різноманіття кольорів чи форм, а зосередитись на тому, заради чого користувач і скачуватиме застосунок.

Враховуючи наведені вище доведення та описану логіку, було обрано кольорову гаму, створено іконку та головні екрани додатку (рис. 3.5).

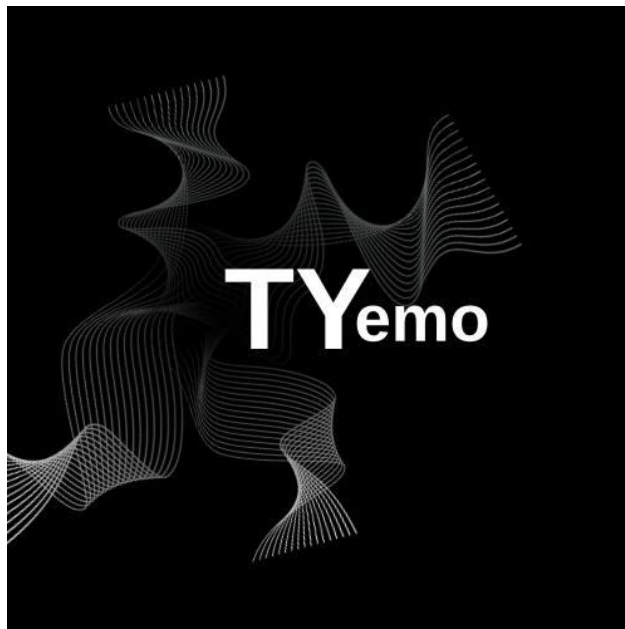


Рисунок 3.5 - Іконка додатку

Іконка містить логотип та основні кольори додатку та своїми символами відображає так званий музичний «вайб» (рис. 3.6).

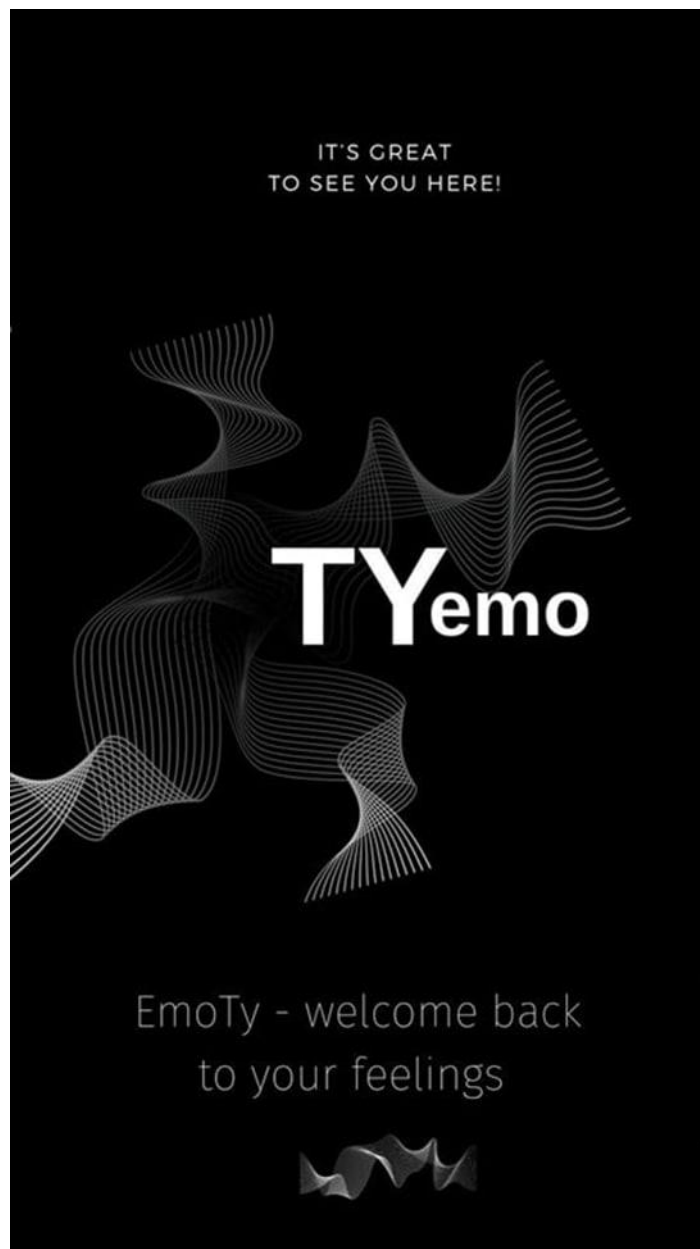


Рисунок 3.6 - Екран завантаження та привітання застосунку

На рис. 3.7 зображено початковий екран, з якого користувач може відкрити наступний екран для обрання (створення вибірки) пісень для подальшого їх аналізу

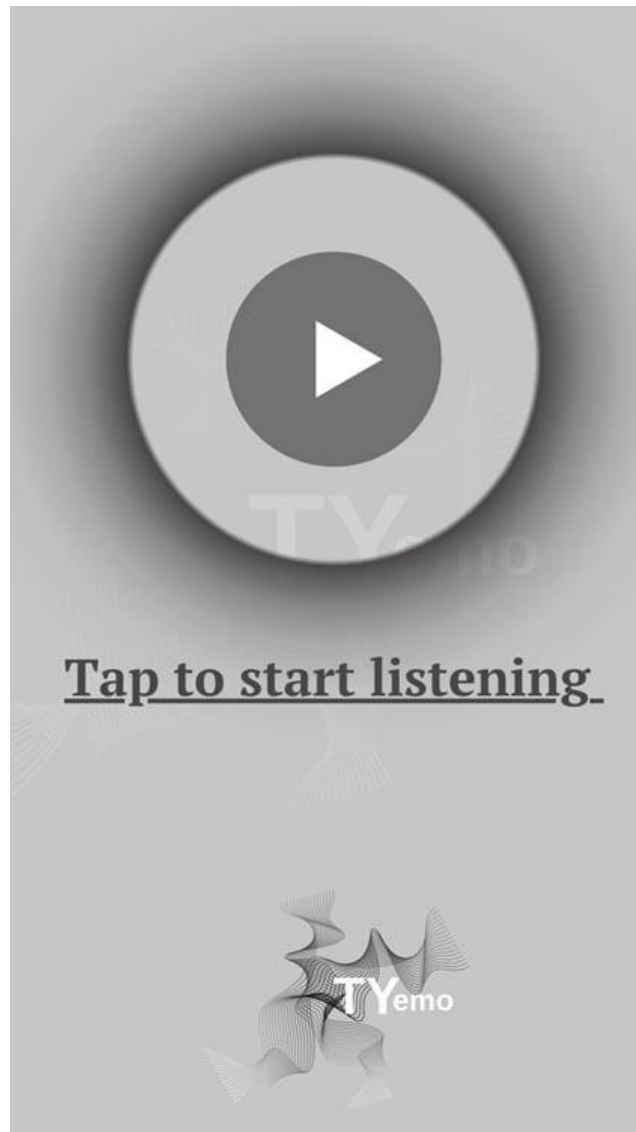


Рисунок 3.7 - Початковий екран

На рис. 3.8 зображено екран прослуховування пісні. Загальна ідея інтерфейсу полягає в відтворенні треків, відображенні обкладинки треку, отриманні реакції користувача щодо кожного треку та збереженні результатів



Рисунок 3.8 - Екран прослуховування пісні

Цей екран є головною частиною проекту, користувач прослуховуючи музику визначає чи імпонує йому та чи інша композиція, програмно, базуючись на виборах користувача ми підбираємо йому наступну композицію таким чином, щоб максимально не упереджено створити вибірку треків. Після

чого, передавши цю множину далі, для подальшого аналізу (рис. 3.12) і отримання результатів щодо ментального стану користувача.

Логіка застосунку передбачає можливість пропустити пісню, якщо користувачу просто вона не подобається і він не хотів би отримати відносно неї результатів про свій ментальний стан. Таким чином функція (рис. 3.12) прийнявши на вхід вибірку композицій зможе максимально чітко передати свій результат функції `deriveMentalStateFrom` (рис. 3.13), відобразити на екрані з результатом (рис 3.9) ментальний стан користувача.



Рисунок 3.9 - Екран з відображенням результату

На рис. 3.9 зображено екран з відображенням результату ментального стану користувача, інтерфейс передбачає можливість зберегти стан, для ведення статистики (рис 3.10) користувачем та відслідковування ним його власного стану у певний час на день.

MOTIVATION	10.05.2023	12:27
PLEASURE	12.05.2023	18:40
PLEASURE	13.05.2023	11:10
MOOD	15.05.2023	19:45
DESIRE	16.05.2023	9:30
INTENTION	16.05.2023	16:50

Рисунок 3.10 - Екран збережених станів

Збереження результатів користувача відбувається з використанням UserDefaults. UserDefaults є простим способом зберігання невеликих об'єктів даних, таких як рядки, числа, булеві значення, масиви або словники, в iOS за допомогою ключів-ідентифікаторів. Таким чином ми можемо зберегти дані користувача, не перевантажуючи додаток підключенням до серверу чи бази даних. При рішенні з API чи сервером, необхідно було додавати реєстрацію користувачам, що не є необхідним для використання застосунку. Також додавши реєстрацію, ми могли б викликати додаткові навантаження, щоб зробити інформацію користувача максимально захищеною, таким чином, щоб

це не викликало запитань у модератора Apple, так як захищеність є важливим пунктом зі списку правил Apple стосовно розробки та дистрибуції застосунків. Також, слід зазначити, що при подальшій монетизації, наприклад, додавання підписки для користування додатком, Apple вимагає створенню можливості попереднього огляду продукту до реєстрації і вводячи можливість реєстрацію буде досить складно відтворити це, так як нам необхідні дані користувача для того, щоб перевірити стан його підписки. Інакше кажучи, можливість попереднього огляду, до реєстрації, створювала б випадки недоброчесного використання користувачами застосунку. Використовуючи лише інструментарій Apple, такий як UserDefaults для збереження даних, та In-App Purchase" (IAP), для відтворення підписки, ми можемо створити більш приємний функціонал, що покращить user-experience, та не створить додаткових проблем при проходженні модерації.

На рисунку 3.10 зображено збережені стани користувача, що реалізовано за допомогою UserDefaults.

При створенні таблиць важливо пам'ятати, що у IOS для збереження оперативної пам'яті девайсу, cell видаляється в момент, коли користувач її не бачить. Що в свою чергу, може виглядати у певних випадках неприємно для користувача, чим погіршить загальний user experience. Щоб цього не відбулось існує технологія кількох потоків (рис 3.11), що підтримується процесорами Apple. У нашому випадку для використання кількох потоків ми використаємо DispatchQueue.

На рис. 3.11 зображено, що на serial (послідовній) черзі завершення замикань відбувається суворо в тому порядку, в якому вони надходили на виконання. До того ж, загальний час виконання на serial черзі вищий, ніж на concurrent. На serial (послідовній) черзі виконується в один час тільки одне завдання, а на concurrent (паралельній) черзі кількість завдань у будь-який поточний момент часу може змінюватися

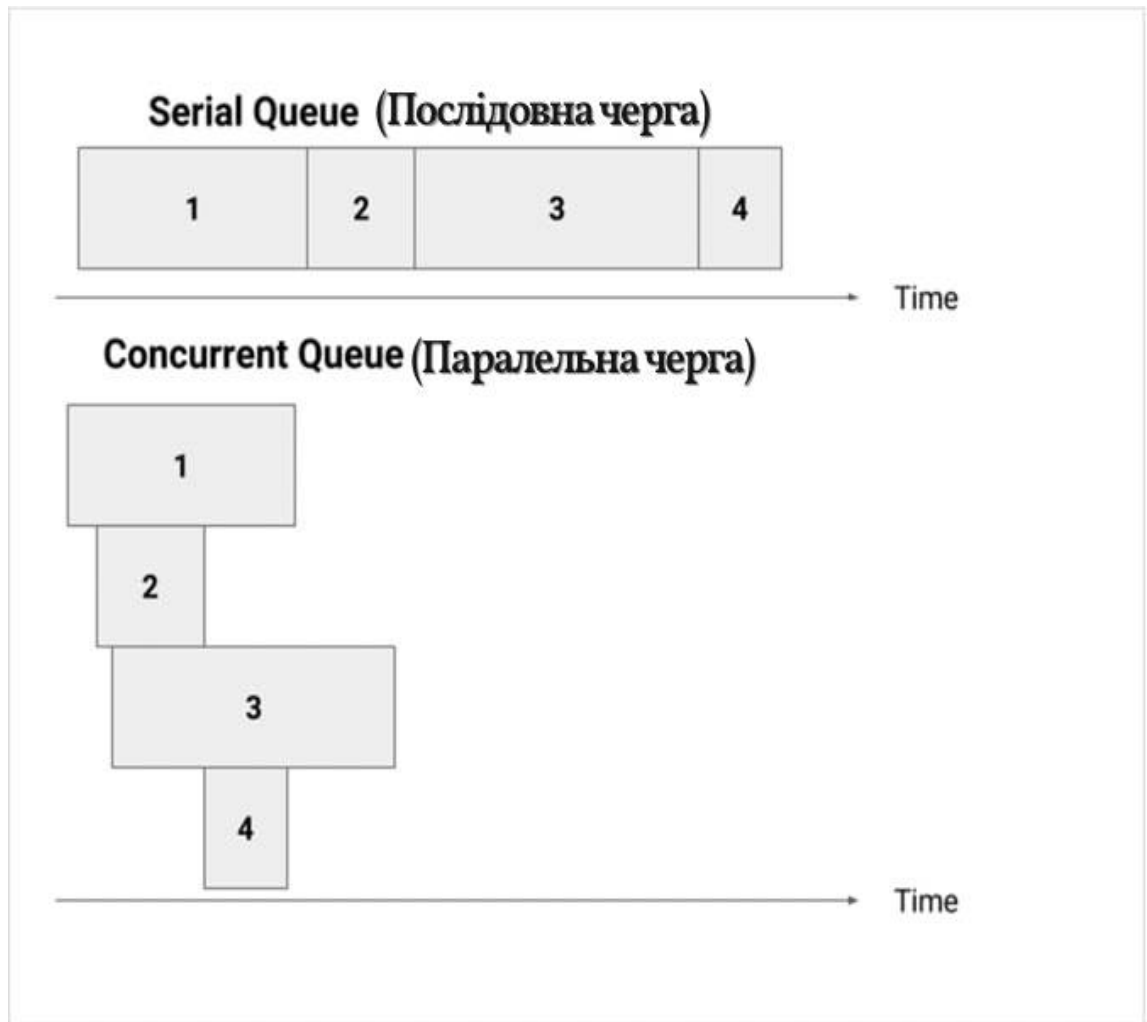


Рисунок 3.11 - Багатопотоковість

3.3.2. Створення функції для визначення ментального стану на базі отриманої вибірки пісень.

Для цього візьмемо таку вибірку оцінки ментального стану:

perception, bodily awareness, thought, belief, desire, motivation, intention, deliberation, decision, pleasure, emotion, mood, imagination and memory.

І розподілемо їх згідно з даними нашої бази даних пісень наступним чином:

1. High "Loudness" and "Danceability" -> Perception
2. High "Acousticness" and "Instrumental" -> Bodily Awareness
3. High "Valence" -> Pleasure

4. Low "Valence" -> Mood
5. High "Audio_class" -> Imagination
6. Low "Audio_class" -> Memory
7. High "Valence" and Low "Danceability" -> Deliberation
8. Low "Valence" and High "Danceability" -> Emotion
9. High "Loudness" and Low "Acousticness" -> Desire
10. Low "Loudness" and High "Acousticness" -> Belief
11. High "Danceability" and Low "Instrumental" -> Motivation
12. Low "Danceability" and High "Instrumental" -> Thought
13. High "Valence" and High "Audio_class" -> Intention
14. Low "Valence" and Low "Audio_class" -> Decision

"Психічний стан" визначається середнім значенням кожної характеристики (гучність, валентність, танцювальність, акустичність, інструментальність та аудіо_клас) для пісень, які подобаються користувачеві, мінус середнє значення кожної характеристики для пісень, які йому не подобаються. Результатом буде словник цих відмінностей.

Ми вже зчитали файл .csv у структуру Swift Dictionary або Array, до якої може отримати доступ функція.

Файл .csv має стовпці для "Назва_пісні", "Гучність", "Валентність", "Танцювальність", "Акустичність", "Інструментальність" та "Аудіо_клас". Стовпчик Audio_class містить числове значення.

Вподобання користувача передаються до функції у вигляді словника, де ключем є "Назва_пісні", а значенням - булеве, що вказує на те, чи подобається користувачеві пісня (істина), чи ні (брехня).

На рис. 3.12 зображено псевдокод функції calculateMentalState, що обчислює суму кожної характеристики для пісень, які користувачеві подобаються і не подобаються. Потім вона ділить кожну суму на загальну суму характеристик, що сподобалися, мінус загальна сума характеристик, що не сподобалися, щоб отримати диференційовану оцінку.

```

function calculateMentalState(songData, userPreferences):
    mentalState = empty dictionary
    likeCount = 0.0
    dislikeCount = 0.0

    for each song, preference in userPreferences:
        if song exists in songData:
            songAttributes = songData[song]
            for each attribute, value in songAttributes:
                if preference is true:
                    likeCount = likeCount + value
                    if attribute exists in mentalState:
                        mentalState[attribute] = mentalState[attribute] + value
                    else:
                        mentalState[attribute] = value
                else:
                    dislikeCount = dislikeCount + value
                    if attribute exists in mentalState:
                        mentalState[attribute] = mentalState[attribute] - value
                    else:
                        mentalState[attribute] = -value

    for each attribute, _ in mentalState:
        mentalState[attribute] = mentalState[attribute] / (likeCount - dislikeCount)

    return mentalState

```

Рисунок 3.12 - Псевдокод функції calculateMentalState

Далі ми знаходимо психічний стан, який відповідає найвищій диференціюючій характеристиці користувача. Функція calculateMentalState використовується як допоміжна функція для обчислення різниці в характеристиках пісень на основі вподобань користувача. Потім функція deriveMentalStateFrom використовує результат calculateMentalState для визначення найвищої диференціюючої характеристики і зіставляє її з психічним станом.

На рис. 3.13 зображено псевдокод функції deriveMentalStateFrom, Тобто на виході згенерувавши множину з пісень обраних на першому екрані, отримані значення ми можемо підставити у функцію calculateMentalState і

отриманий результат у deriveMentalStateFrom.

```
function deriveMentalStateFrom(songData, userPreferences):
    mentalState = calculateMentalState(songData, userPreferences)
    maxAttribute = getMaxAttribute(mentalState)

    switch maxAttribute:
        case "Loudness":
            return "Perception"
        case "Acousticness":
            return "Bodily Awareness"
        case "Valence":
            return "Pleasure"
        case "Audio_class":
            return "Imagination"
        default:
            return "Unknown"

function calculateMentalState(songData, userPreferences):
    // Perform calculations using songData and userPreferences
    // Return the mental state as a dictionary with attribute-value pairs

function getMaxAttribute(mentalState):
    maxAttribute = null
    maxValue = null

    for attribute in mentalState:
        value = mentalState[attribute]
        if maxValue == null or value > maxValue:
            maxAttribute = attribute
            maxValue = value

    return maxAttribute
```

Рисунок 3.13 - Псевдокод функції deriveMentalStateFrom

Ці функції було створено на основі графіків та статистичних даних отриманих у наступному розділі.

3.4. Графіки на статичні дані на основі БД «Music and Mental Health»

На рис. 3.14 графік графік кореляції між танцювальними параметрами та показниками психічного здоров'я. Для цього використовується plt.scatter для візуалізації окремих точок і plt.plot для відображення лінії, яку передбачає модель лінійної регресії.

Correlation between Danceability and Mental Health Metric

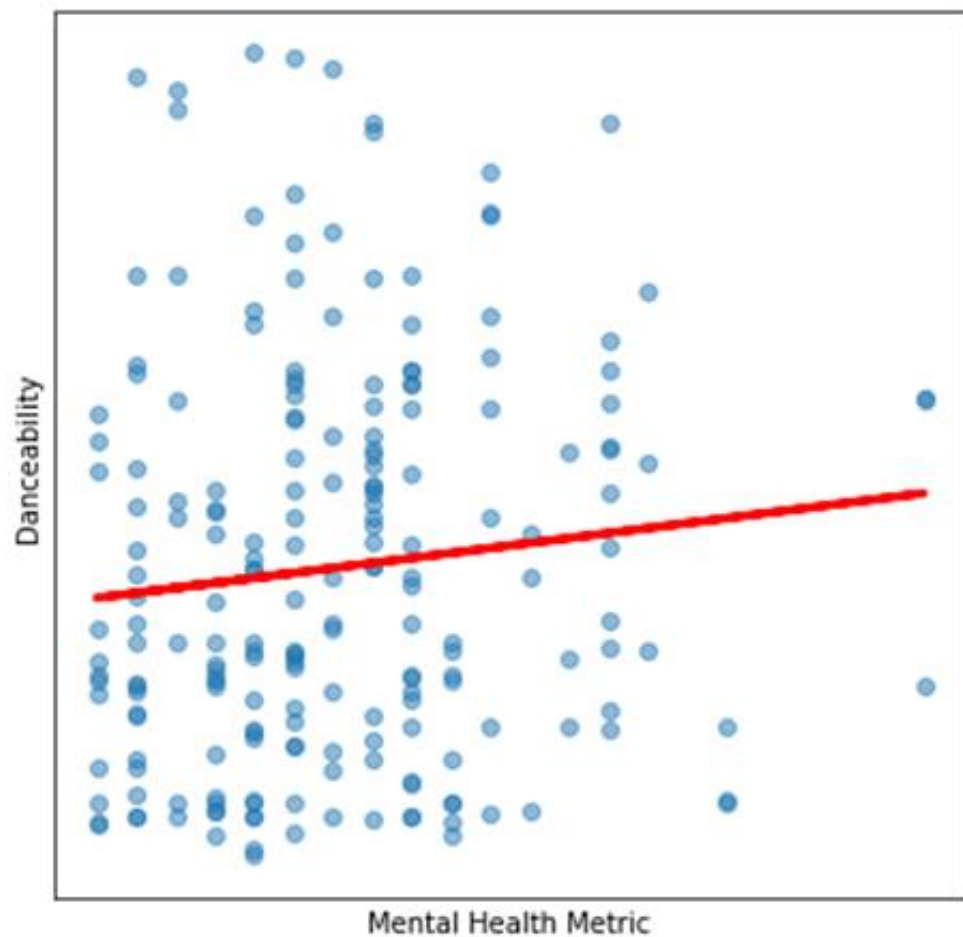


Рисунок 3.14 - Кореляція між танцювальними параметрами та показниками психічного здоров'я

На рис. 3.15 графік графік кореляції між оцінкою психічного здоров'я та валентністю. Після цього виконується візуалізація цієї залежності за допомогою розсіювального графіку і лінії регресії. На графіку встановлюються підписи осей та заголовок.

Correlation between mental health assessment and valence

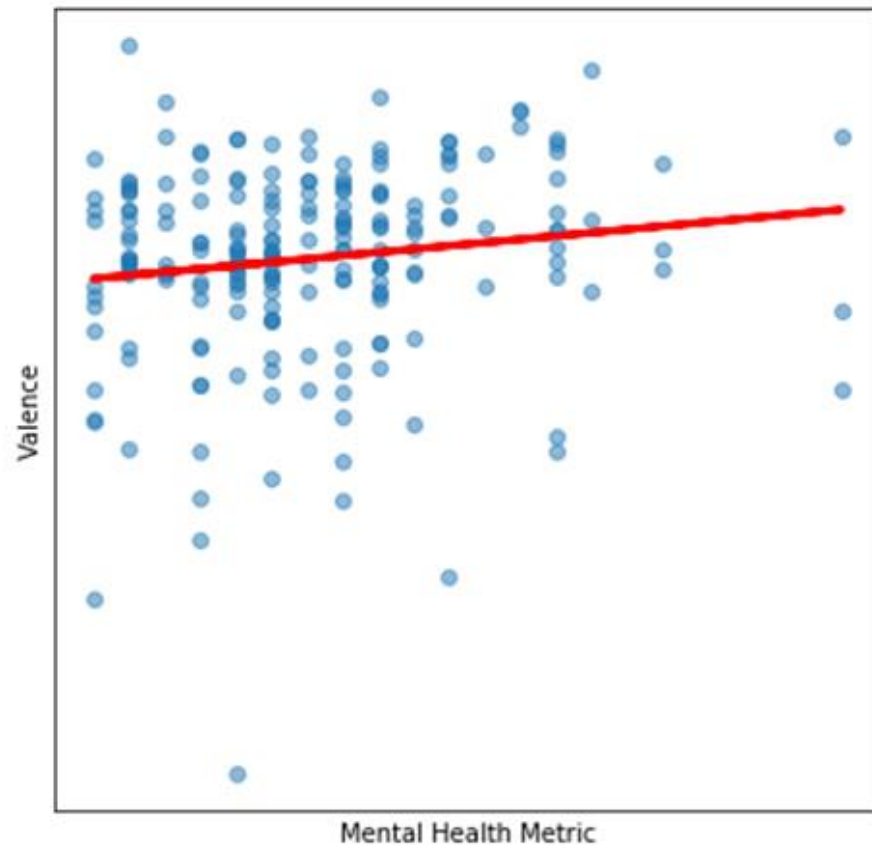


Рисунок 3.15 - Кореляція між оцінкою психічного здоров'я та валентністю

Тепер створимо кілька гістограм. Графік ліворуч ілюструє розподіл пісень за рівнем енергії, тоді як графік праворуч - це "теплова карта" (двомірна гістограма), яка ілюструє кількість пісень, знайдених при всіх значеннях валентності та танцювальності.

На рис. 3.16 графік типу "heatmap", які відображають зв'язок між двома змінними ("Loudness" і "Total_mental_health", та "Danceability" і "Total_mental_health").

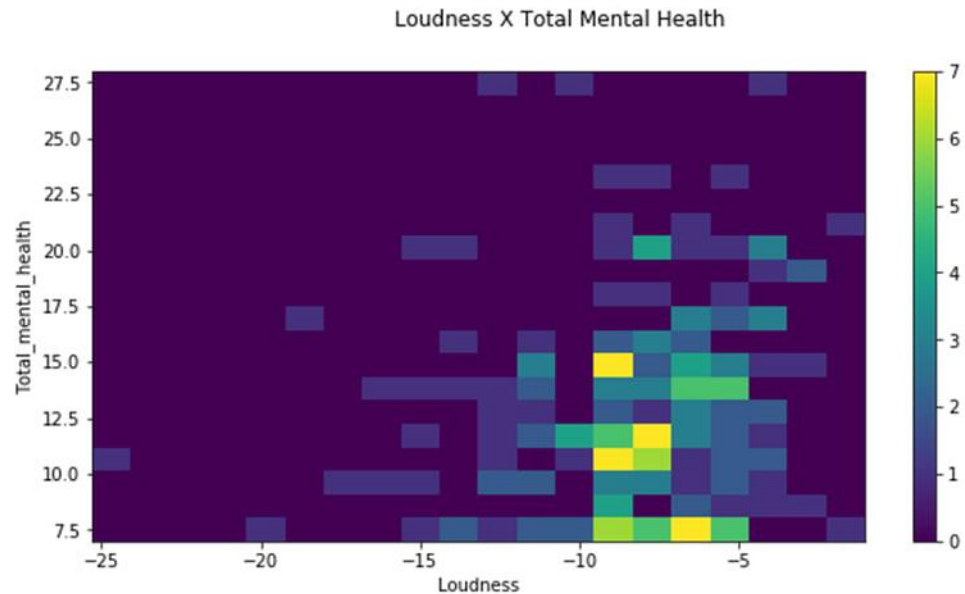


Рисунок 3.16 - Зв'язок між двома змінними "Loudness" і "Total_mental_health", та "Danceability" і "Total_mental_health".

Для першого графіку:

1. Визначаються змінні "x" і "y" як "Loudness" і "Total_mental_health".
2. Створюється фігура (Figure) з одним підграфіком за допомогою `plt.subplots()`.
3. Встановлюється заголовок фігури за допомогою `fig.suptitle()`.
4. Створюється "heatmap" за допомогою `ax2.hist2d()`, де передаються дані зі змінних "x" і "y" з датафрейму "data_frame". Параметр `bins` вказує на кількість бінів, у які розбиваються дані.
5. Встановлюються підписи осей для підграфіка за допомогою `ax2.set_xlabel()` і `ax2.set_ylabel()`.
6. Додається колірна шкала (`colorbar`) за допомогою `plt.colorbar()`, де передається параметр `h[3]`, що вказує на об'єкт "heatmap", і `ax=ax2`.

Другий графік має аналогічну структуру, але використовує змінні "Danceability" і "Total_mental_health".

Отримані фігури та графіки відображаються за допомогою `plt.show()`.

На рис. 3.17 відображено взаємозв'язку між двома змінними "Audio + Lyrics analysis" (x) і "Total_mental_health" (y).

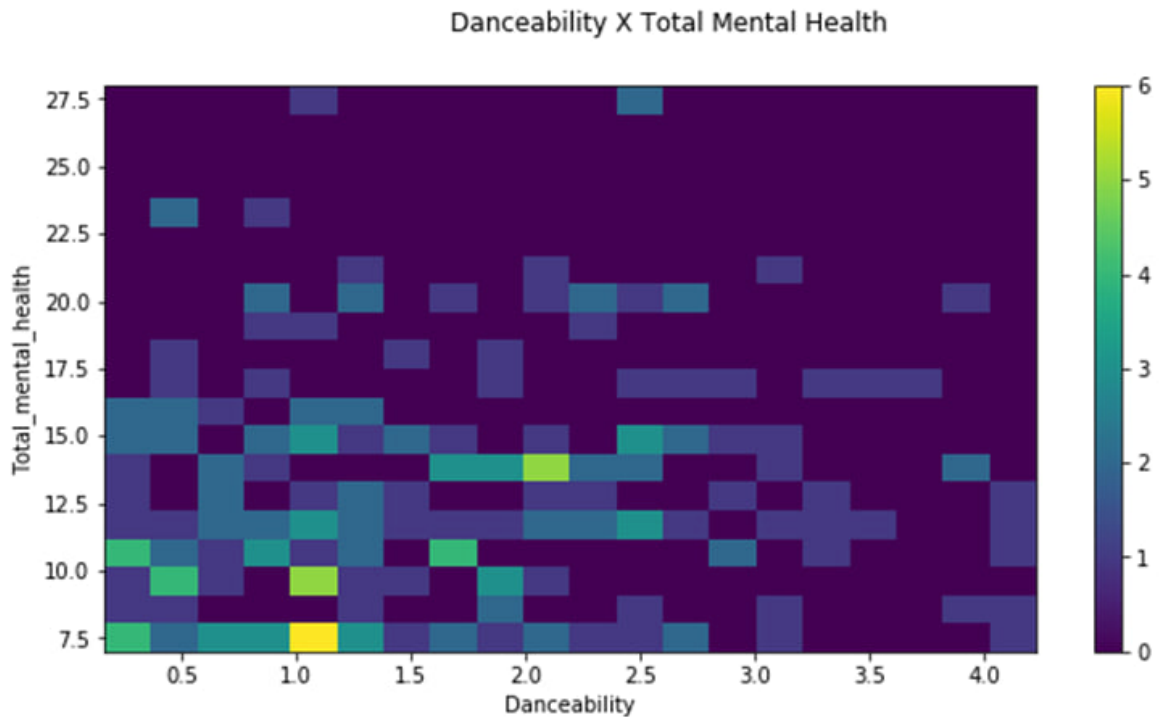


Рисунок 3.17 - Взаємозв'язок між двома змінними "Audio + Lyrics analysis" (x) і "Total_mental_health" (y)

Спочатку визначаються змінні "x" і "y", які вказують на назви стовпців "Audio + Lyrics analysis" і "Total_mental_health" відповідно.

Створюється фігура (Figure) з одним підграфіком за допомогою функції `plt.subplots()`. Параметри 1, 1 вказують на один рядок і один стовпець підграфіків.

Встановлюється заголовок для фігури за допомогою методу `suptitle()`, де передається рядок "Audio & Lyrics Sentiment X Total Mental Health".

Створюється "heatmap" за допомогою функції `hist2d()`, де передаються дані зі змінних "x" і "y" з датафрейму "data_frame". Параметр `bins` вказує на кількість бінів, у які розбиваються дані.

Встановлюються підписи осей для підграфіка за допомогою методів

set_xlabel() і set_ylabel()).

На рис. 3.18 відображено взаємозв'язок між двома змінними "Audio + Lyrics analysis" (x) і "Total_mental_health" (y).

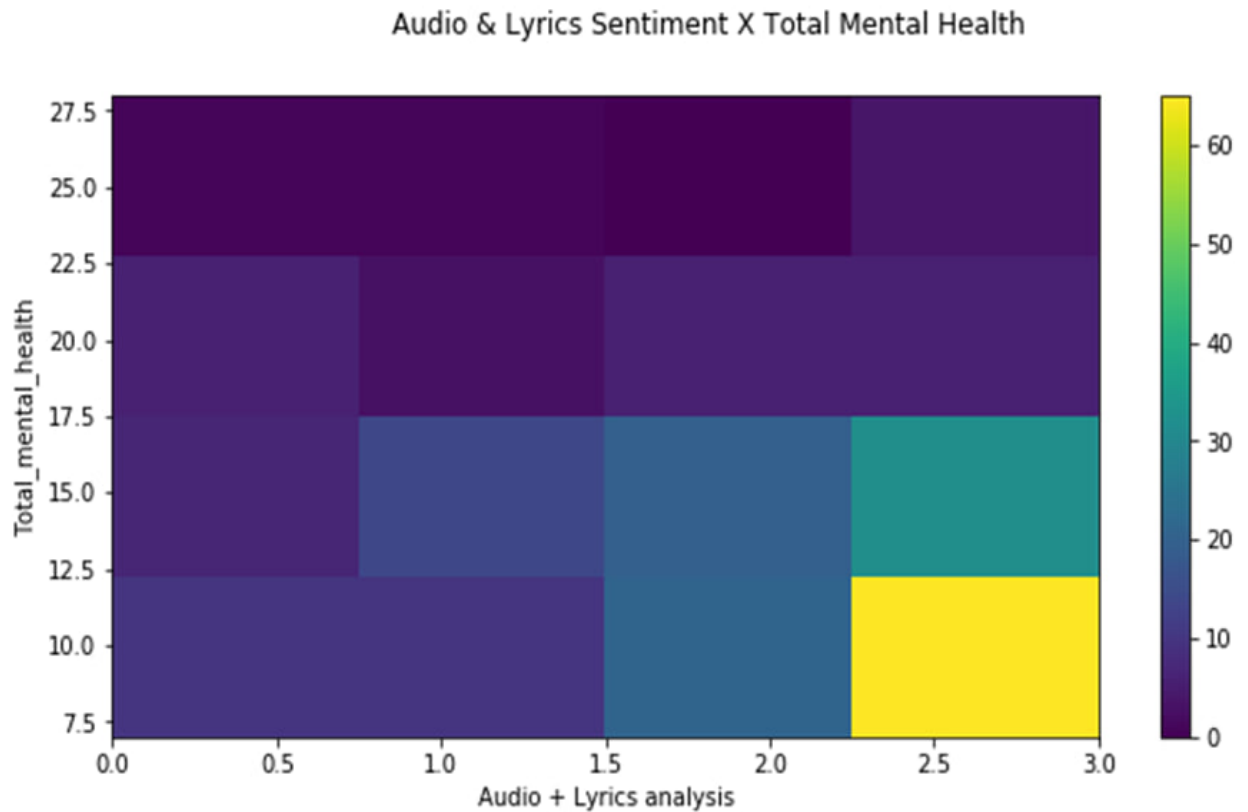


Рисунок 3.18 - Відображення взаємозв'язку між двома змінними "Audio + Lyrics analysis" (x) і "Total_mental_health" (y).

Спочатку визначаються змінні "x" і "y", які вказують на назви стовпців "Audio + Lyrics analysis" і "Total_mental_health" відповідно.

Створюється фігура (Figure) з одним підграфіком за допомогою функції plt.subplots(). Параметри 1, 1 вказують на один рядок і один стовпець підграфіків.

Встановлюється заголовок для фігури за допомогою методу suprtitle(), де передається рядок "Audio & Lyrics Sentiment X Total Mental Health".

Створюється "heatmap" за допомогою функції hist2d(), де передаються дані зі змінних "x" і "y" з датафрейму "data_frame". Параметр bins вказує на

кількість бінів, у які розбиваються дані.

Встановлюються підписи осей для підграфіка за допомогою методів `set_xlabel()` і `set_ylabel()`.

Додається колірна шкала (`colorbar`) за допомогою `plt.colorbar()`, де передається параметр `h[3]`, що вказує на об'єкт "heatmap", і `ax=ax2` для відповідного підграфіка.

Отриману фігуру і графік відображається за допомогою `plt.show()`.

На рис. 3.19 ми застосовуємо аналіз за методом головних компонент , щоб зменшити розмірність обраного списку атрибутів до 3, ефективно створюючи матрицю, придатну для побудови наступного 3-вимірного графіка. Чим менша відстань між будь-якими двома піснями, тим більша схожість у їхніх ознаках.

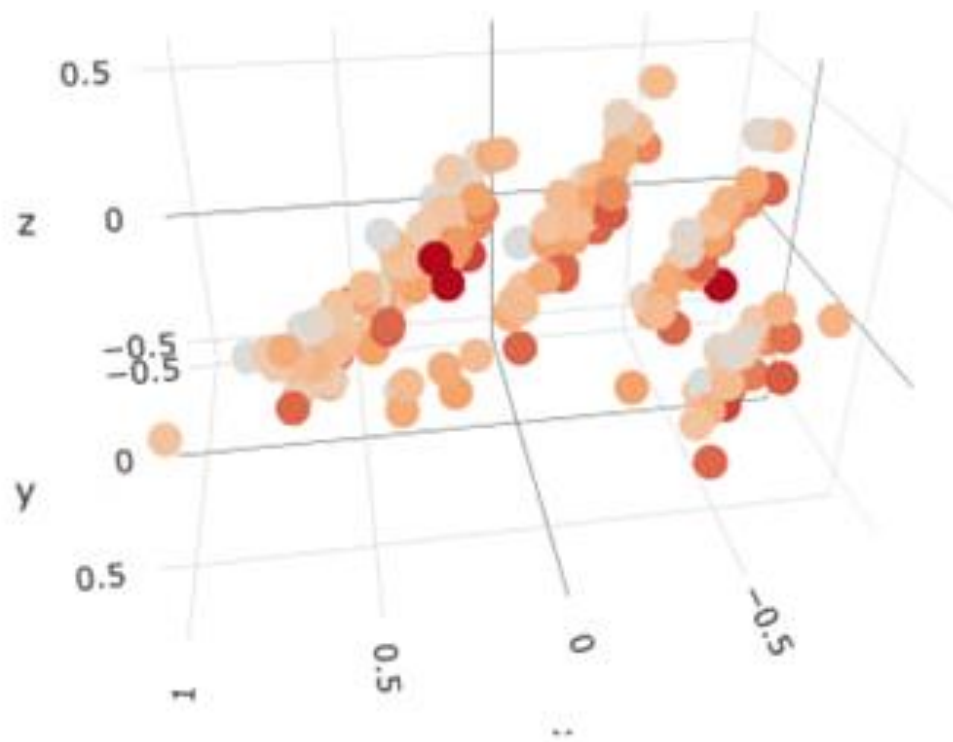


Рисунок 3.19 - Аналіз за методом головних компонент

В іншій спробі ми використовуємо менший список "обраних" і аналіз за методом головних компонент для створення двовимірного графіка. Цей конкретний графік було обрано через його, здавалося б, влучні назви осей. Код виконує аналіз головних компонентів (PCA) для вибраних ознак з датафрейму "data_frame".

Спочатку обираються певні ознаки, які зберігаються у змінній "chosen". Далі формується текстова інформація для кожної точки графіку, яка складається з поєднання значень стовпців "Artist" і "Song_name". Ознаки "chosen" та відповідний стовпець "Total_mental_health" витягуються з датафрейму і зберігаються у змінній "X" та "y" відповідно. Далі використовується масштабування методом Min-Max (MinMaxScaler), щоб нормалізувати значення ознак у діапазоні від 0 до 1. Проводиться аналіз головних компонентів (PCA) з використанням двох компонентів (n_components=2). Метод "fit_transform" застосовується до нормалізованих даних "X", щоб отримати нові перетворені значення "X".

Створюється об'єкт "fig" з даними для графіку. Для кожної точки графіку використовуються перетворені значення PCA "X[:, 0]" і "X[:, 1]". Текстова інформація для кожної точки використовується як "text". Графік представлений у вигляді розсіювання ("mode": "markers"). Розмір і колір точок визначаються за допомогою змінної "y".

Створюється макет графіку, де встановлюються підписи для вісей "xaxis" і "yaxis". У кінці графік відображається за допомогою функції "iplot" з бібліотеки Plotly. На рис. 3.20 модель кластеризації K-середніх на основі результатів аналізу головних компонентів (PCA). Спочатку відбувається фільтрація даних з датафрейму "data_frame_pca" для кожного кластера. Кожний кластер представлений окремим фрагментом даних, де значення стовпця "cluster" дорівнює відповідному кластеру.

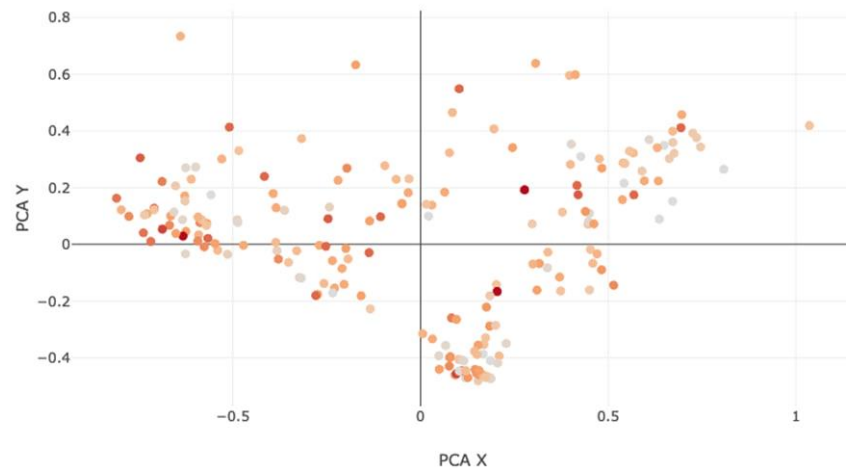


Рисунок 3.20 – Модель кластеризації К-середніх

Далі з даних кожного фільтрованого кластера витягуються значення координат для графіку. "pca1" використовується для вісі X, а "pca2" - для вісі Y. Крім того, формується текстова інформація для кожної точки графіку, яка складається з поєднання значень стовпців "Artist" і "Song_name". Для кожного кластера створюється окремий об'єкт "trace" з типом "scatter" (розсіювання), який представляє кластер на графіку. Кожен об'єкт "trace" містить координати, текст та налаштування для відображення точок кластера, такі як колір і розмір. У кінці об'єкти "trace" об'єднуються в список "data".

Створюються налаштування для графіку, такі як заголовок ("title") і підписи осей ("xaxis" і "yaxis"). Створюється об'єкт "fig" з використанням даних і налаштувань, а потім графік відображається за допомогою функції "iplot" з бібліотеки Plotly.

На рис. 3.21 показано «метод ліктя» - кластеризація методом k-середніх (KMeans) для різної кількості кластерів (від 2 до 10) , яка обчислює суму квадратів відстаней (SSE) між кожною точкою і центроїдою свого кластера. На кожній ітерації використовується модель KMeans з вказаною кількістю кластерів (`n_clusters=i`) і випадковим початковим станом (`random_state=99`).

Модель навчається на даних "kmeans_X_Y_sne".

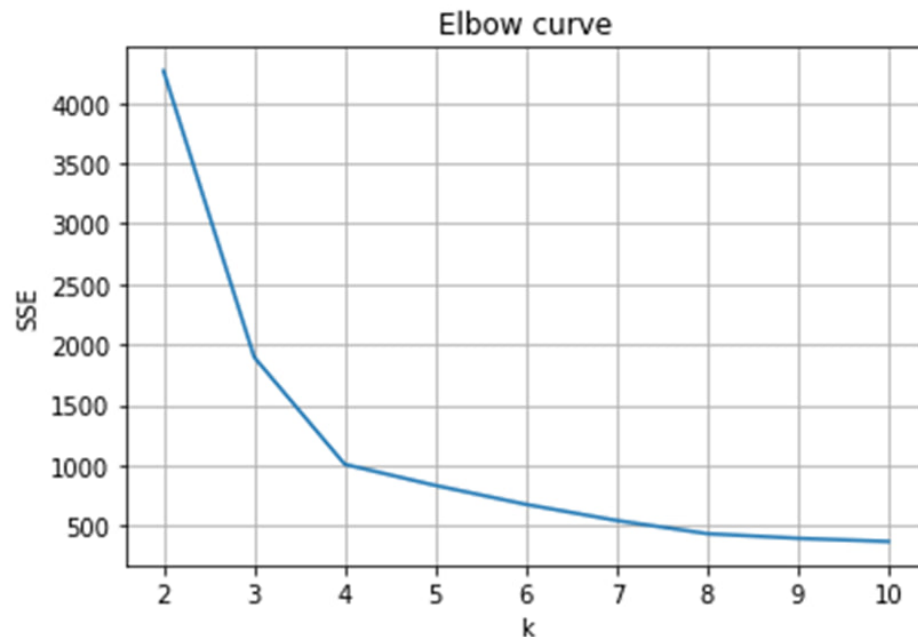


Рисунок 3.21 – Метод ліктя

Після навчання моделі обчислюється SSE за допомогою атрибута "inertia_" моделі KMeans, і це значення додається до списку SSE. У циклі генеруються значення k (2 до 10) для осі X графіку "Ліктьова крива" (Elbow Curve). Значення SSE використовується для осі Y графіку. Створюються налаштування графіку, такі як заголовок ("title"), підписи осей ("xlabel" і "ylabel") та сітка ("grid"). Графік SSE відображається за допомогою функції "plot" бібліотеки Matplotlib, а потім відображається вікно графіку за допомогою функції "show".

На рис. 3.22 зображено K-середньоквадратичну модель кластеризації (t-SNE), k=6 (різні групи даних залежно від значень "cluster" у DataFrame "data_frame_sne"). Для кожної групи даних створюються відповідні значення для використання у графіці. Кожна група даних представлена на графіку як розсіювальний графік з позначеними точками, де координати точок визначаються за допомогою "pca1" і "pca2". Також додається текстова

інформація для кожної точки графіку, яка відображається при наведенні курсора на точку.

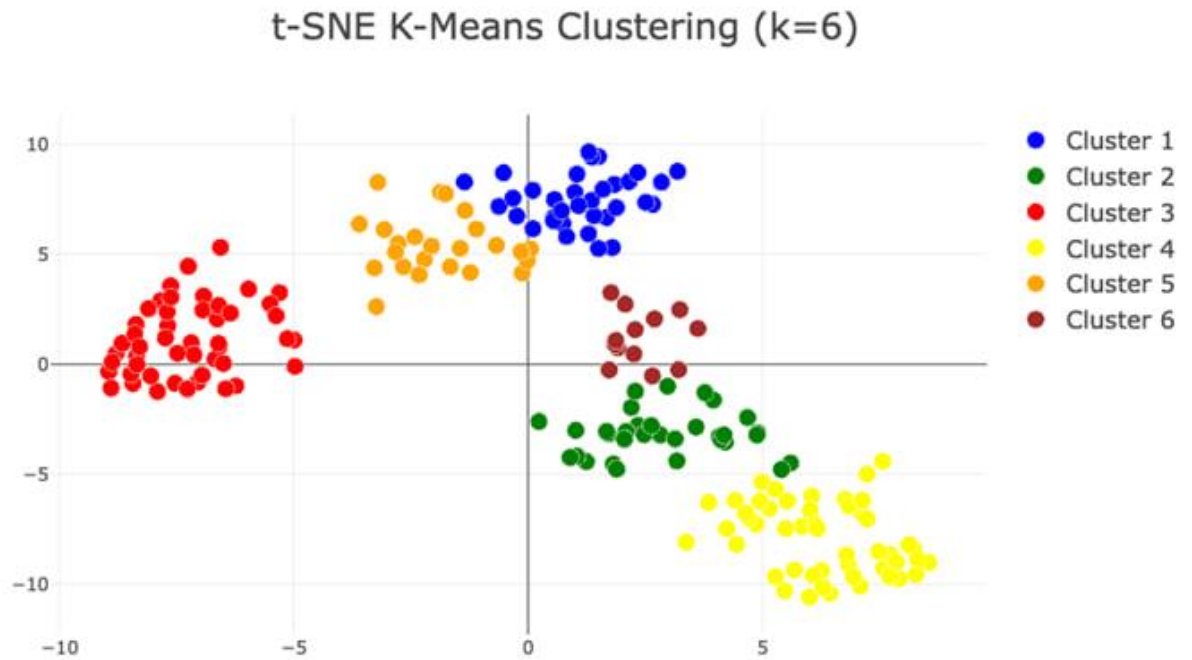


Рисунок 3.22 - К-середньоквадратична модель кластеризації (t-SNE),
k=6

Всі групи даних об'єднуються в список "data", а також встановлюються налаштування для графіку, такі як заголовок ("title"), осі координат ("xaxis" і "yaxis") та легенда ("legend").

Завершується код створенням об'єкту Figure за допомогою бібліотеки Plotly та візуалізацією графіку за допомогою команди "py.iplot".

3.5. Розрахунок економічного ефекту

Для проведення розрахунків було створено морфологічну карту системи (рис. 3.23), який показує можливі варіанти функцій.

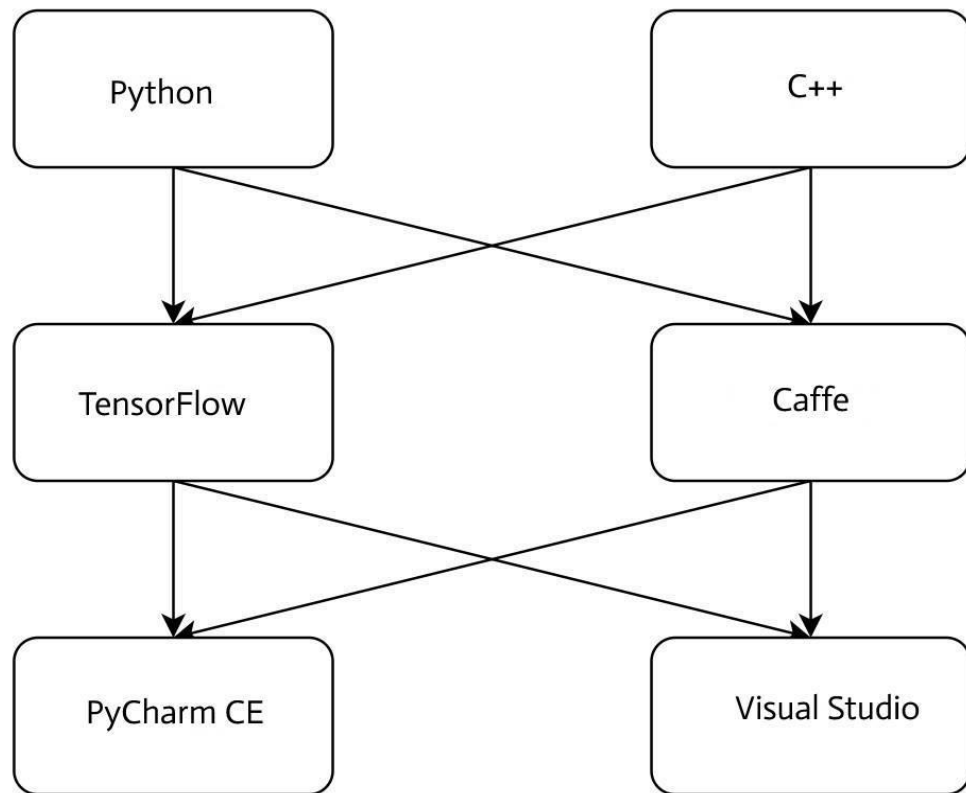


Рисунок 3.23 – Морфологічна карта системи

За результатами функціонально-вартісного аналізу ПП для визначення стану на основі музичних вподобань було розглянуто два можливі варіанти реалізації, і з них найбільш відповідним поставленій задачі є другий варіант з техніко-економічним рівнем якості $K_{\text{TEP}} = 1.65 * 10^{-5}$.

Даний варіант реалізації ПП має відповідні параметричні дані:

- мова програмування для back end – Python;
- мова програмування для front end – Swift;

Реалізація названого обраного варіанту обов'язково повинна виконувати такі завдання:

- Розробка back end частини програмного продукту;
- Розробка front end частини програмного продукту;

Обраний варіант реалізації поставленої для роботи мети та завдання

дасть оптимальну швидкість при функціоналі, який вимагається, та дозволять розробити максимально зручний інтерфейс при мінімально можливих витратах по пам'яті та швидкодії.

Висновок до розділу 3

У розділі «Практична реалізація» було виконано реалізацію практичного завдання дипломної роботи. Описувалося проектування застосунку, а саме технології створення та архітектурні деталі, та шлях реалізації, тобто принципи та технології, що були застосовані для досягнення сформульованого на початку даного розділу технічного завдання. Важливою частиною проектування була розробка доступу та підключення до сервісу Spotify API, наріжного каменю проекту. Була виконана візуалізація роботи застосунку, реалізація програмного інтерфейсу, тобто був придуманий дизайн додатку, основний лозунг, назва, ідея роботи (логіка програми). Важливим моментом у візуалізації роботи додатку було створення функції для визначення ментального стану користувача на основі отриманої вибірки пісень. Цю функцію було розроблено на основі графіків, статистичних даних та моделювання на базі даних «Music and Mental Health», взятої з платформи Kaggle. Тобто взяті там дані стали основою логіки роботи застосунку.

ЗАГАЛЬНІ ВИСНОВКИ

Під час проведення дослідження було виконано дипломної роботи. Було зроблено:

- огляд та аналіз джерел для дослідження;
- пошук та висвітлення необхідної для розкриття обраної теми теорії; сформульовано основне технічне завдання;
- спроектовано архітектуру та інтерфейс додатку, обрано методи та шляхи виконання реалізації;
- реалізовано застосунок, що оброблятиме музичні вподобання користувача та інтерпретуватиме їх як показник його психічного стану;
- розроблено функціонально-вартісний аналіз додатку.

Для того, щоб якісно сформулювати основні завдання для розробки, було розглянуто питання впливу музики на психічний стан людини та джерела, що це доказують. Також було проаналізовано два дослідження, що підтвердили музику як один з показників емоційного стану людини, та порівняно їх з функціоналом додатку, який розроблявся.

Було розроблено дизайн проекту – іконку додатку, його основні екрани, логотип, та обґрунтовано позицію саме такого вибору цих деталей.

Сформульовано основні опції застосунку. За допомогою UIKit було створено сучасний застосунок для IOS. UIKit взято основним фреймворком для розробки користувацького інтерфейсу для iOS додатків. Архітектурою нашого застосунку було обрано Model-View-Controller.

Додаток отримав такі користувацькі функції:

- прослуховування музичних творів (а саме десяти);
- уподобання пісні або вибір віконця «dislike»;
- можливість переключити трек на наступний за потреби;
- можливість переглядати стани, в яких перебував користувач, їх дату і точний час.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Оригінальне дослідження "Learning About Your Mental Health From Your Playlist? Investigating the Correlation Between Music Preference and Mental Health of College Students" [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.frontiersin.org/articles/10.3389/fpsyg.2022.824789/full#B33>
2. Стаття "Can Music Preference Indicate Mental Health Status in Young People?" [Електронний ресурс] - Режим доступу до ресурсу:
<https://journals.sagepub.com/doi/10.1080/10398560701879589>
3. Дослідження "Prevalence of Mental Health Status and Music Preferences of MRSM Taiping Students" [Електронний ресурс] - Режим доступу до ресурсу:
[file:///C:/Users/user/Downloads/THESIS_ZAWANI_MOHD_TAJUDIN%20\(1\).pdf](file:///C:/Users/user/Downloads/THESIS_ZAWANI_MOHD_TAJUDIN%20(1).pdf)
4. Стаття "What is Spotify and how does it work?" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.pocket-lint.com/what-is-spotify-and-how-does-it-work/>
5. Стаття "What is Spotify? Here's everything you need to know in 2023" [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.androidauthority.com/what-is-spotify-1129032/>
6. Стаття "What is an API? Application programming interfaces explained" [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.infoworld.com/article/3269878/what-is-an-api-application-programming-interfaces-explained.html>
7. Стаття "Using the Spotify API" [Електронний ресурс] - Режим доступу до ресурсу: <https://medium.com/nerd-for-tech/using-the-spotify-api-ad11410c0a6>
8. Стаття "Apple iOS" [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.techtarget.com/searchmobilecomputing/definition/iOS>
9. Стаття "What is a Framework? [Definition] Types of Frameworks"

- [Електронний ресурс] - Режим доступу до ресурсу: <https://hackr.io/blog/what-is-frameworks>
- 10.Стаття "iOS SDK | Spotify for Developers" [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.spotify.com/documentation/ios>
 - 11.Стаття "Spotify Song Attributes. An attempt to build a classifier that can predict whether or not I like a song" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.kaggle.com/datasets/geomack/spotifyclassification>
 - 12.Сторінка "AudioFeatures" [Електронний ресурс] - Режим доступу до ресурсу: <https://spotify-api.js.org/main/interface/AudioFeatures>
 - 13.Стаття "Web API Retrieve metadata from Spotify content, control playback or get recommendations" [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.spotify.com/documentation/web-api>
 - 14.Стаття "Understanding the Spotify Web API - Spotify Engineering" [Електронний ресурс] - Режим доступу до ресурсу: <https://engineering.atspotify.com/2015/03/understanding-spotify-web-api>
 15. Стаття "Spotify Developer. (2021). Get Track's Audio Features" [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.spotify.com/documentation/web-api/reference/get-audio-features>
 - 16.Стаття "Spotify Developer. (2021). Getting started with Web API" [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.spotify.com/documentation/web-api/tutorials/getting-started>
 - 17.Стаття "Jed Foundation. (2021). How Music Can Improve Your Mental Health" [Електронний ресурс] - Режим доступу до ресурсу: <https://jedfoundation.org/resource/how-music-can-improve-your-mental-health/>
 - 18.Стаття "Harvard Health Publishing. (2022). Can music improve our health and quality of life?" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.health.harvard.edu/blog/can-music-improve-our-health-and-quality-of-life-202207252786>
 - 19.Стаття "6 Benefits Of Music For Your Mental Health" [Електронний ресурс] -

- Режим доступа до ресурсу: <https://www.palladiumprivate.com/blog/6-benefits-of-music-for-your-mental-health/>
- 20.Стаття “Music, mental health, and immunity” [Электронний ресурс] - Режим доступа до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8566759/>
- 21.Стаття “Spotify Developer. (2021). API calls” [Электронний ресурс] - Режим доступа до ресурсу: <https://developer.spotify.com/documentation/web-api/concepts/api-calls>
- 22.Стаття “Python and REST APIs: Interacting With Web Services” [Электронний ресурс] - Режим доступа до ресурсу: <https://realpython.com/api-integration-in-python/>
- 23.Стаття “Top 15 Swift Frameworks” [Электронний ресурс] - Режим доступа до ресурсу: <https://www.dunebook.com/swift-frameworks/>
- 24.Стаття “Psychological effects of music on mental health” [Электронний ресурс] - Режим доступа до ресурсу: https://www.researchgate.net/publication/360396590_Psychological_effects_of_music_on_mental_health
25. Стаття “Developments in Optimization Algorithms for Smart Healthcare” [Электронний ресурс] - Режим доступа до ресурсу: <https://www.hindawi.com/journals/jhe/2022/7288788/>