

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА  
Факультет інформаційних технологій  
Кафедра прикладних інформаційних систем

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА  
БАКАЛАВРА  
НА ТЕМУ**

Програмний застосунок добирання музичного контенту  
з використанням хмарних технологій

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

Освітня програма «Прикладне програмування»

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи ПП-41

Грицунь Я. Б.

(прізвище та ініціали)

Керівник Сайко В.Г.

(прізвище та ініціали)

д.т.н., професор

(науковий ступінь, звання)

Унікальність тексту 95%

Випускна кваліфікаційна робота бакалавра допущена до захисту рішенням  
кафедри *прикладних інформаційних систем*

Протокол №     від     р.  
зав. кафедри проф. Плєскач В.Л.

Київ - 2023

## КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

№з/п	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	14.10.2022	виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	24.10.2022	виконано
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	31.10.2022	виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	01.11.2022	виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	08.11.2022	виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	21.12.2022	виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	31.01.2023	виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	30.03.2023	виконано
9.	Подання роботи у першому варіанті	28.04.2023	виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	03.05.2023	виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	22.05.2023	виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	26.05.2023	виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	12.06.2023	виконано
14.	Захист кваліфікаційної роботи бакалавра	26.06.2023	

Здобувач вищої освіти



  
(підпис)

Керівник


  
(підпис)

## ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

Складові частини кваліфікаційної роботи бакалавра	Обсяг, арк.
Титульний аркуш	1
Календарний план кваліфікаційної роботи бакалавра	1
Відомість кваліфікаційної роботи бакалавра	1
Анотація	1
Анотація (іноземною мовою-англійською)	1
Зміст	3
Перелік скорочень, умовних позначень, термінів	1
Вступ	2
1	19
2	26
3	8
Висновки	1
Перелік використаних джерел	5
Додатки	2

				ДП ХХХХ 00.000.00		
	ПІБ	Підп.	Дата			
Розробн.	Грицунь Я.Б.			Відомість дипломної роботи	Лист	Листів
Керівн.	Сайко В.Г.					
Н/контр.	Макаренко С.А.		26.05.2023			
Зав.каф.	Плескач В.Л.					

## АНОТАЦІЯ

Дипломна робота: 72 с., 5 рис., 2 табл., 55 джерел.

Ця дипломна робота присвячена проектуванню та розробленню застосунку добирання музичного контенту з використанням хмарних технологій.

**Метою кваліфікаційної роботи** є зручне добирання музичного контенту за допомогою розробленого мобільного застосунку добирання музичного контенту, а саме, його клієнтської та серверної частин, написаних однією мовою програмування.

### **Завдання дослідження:**

- дослідити сучасні методології розроблення застосунків;
- проаналізувати архітектурні рішення, технології та інструменти для реалізації застосунку;
- спроектувати, впровадити та поєднати клієнтську та серверну частини мобільного застосунку для добирання музичного контенту.

**Об'єктом дослідження** є процеси передачі та відтворення аудіофайлів.

**Предметом дослідження** є програмно-технічні, організаційні засади, принципи, підходи щодо побудови мобільного застосунку добирання музичного контенту.

**Методи дослідження:** теорія управління для дослідження теоретичних аспектів побудови застосунків, емпіричний аналіз і синтез систем, що застосовувався при вивченні прикладів сучасних методів побудови застосунків добирання музичного контенту.

**Ключові слова:** аудіофайл, мобільний застосунок, клієнт-серверна архітектура, хмарні обчислення, стрімінг, програмна система.

## ABSTRACT

Thesis: 72 pages, 5 figures, 2 tables, 55 sources.

This thesis is devoted to the development of an application for selection of music content using cloud technologies.

**The purpose** of the thesis is to enable convenient selection of music content using the developed mobile application for music content recommendation. This includes both the client and server components, which are written in the same programming language. To achieve this goal, one should resolve the following **tasks**:

- explore modern application development methodologies;
- analyze architectural solutions, technologies, and tools for implementing the application;
- design, implement, and integrate the client and server components of the mobile application for music content recommendation;

**Object of study** is the processes of audio file transmission and playback.

**Subject of study** includes the software-technical and organizational foundations, principles, and approaches for building a mobile application for music content selection.

**The research methods** employed include management theory for investigating theoretical aspects of application development and empirical analysis and synthesis of systems, applied in studying examples of modern methods for building music content selection applications.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	9
ВСТУП	10
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА СТАН МУЗИЧНОЇ ІНДУСТРІЇ В УМОВАХ ЦИФРОВІЗАЦІЇ	12
1.1 Стан розвитку цифрової музичної індустрії в Україні та світі, основні її поняття та зміст	12
1.1.1 Визначення музичної індустрії, передумови та стан розвитку в умовах сьогодення. ....	13
1.1.2 Стан цифрової музичної індустрії та її вплив на світову економіку	14
1.2 Нормативно-правове регулювання цифрової музичної індустрії	15
1.2.1 Проблеми та стан правового регулювання цифрової музичної індустрії в світі .....	17
1.2.2 Міжнародна нормативно-правова база цифрової музичної сфери..	18
1.2.3 Правовий аспект захисту персональних даних користувачів систем цифрової музичної індустрії .....	20
1.2.4 Правова база України в сфері цифрової музичної індустрії .....	21
1.3 Організаційно-економічні моделі цифрової музичної індустрії	22
1.3.1 Модель прямого продажу (Direct Sales Model).....	22
1.3.2 Модель стрімінгових платформ (Streaming Platform Model).....	22
1.3.4 Модель співпраці та партнерства (Collaboration and Partnership Model): .....	23
1.3.5 Традиційні бізнес-моделі музичної індустрії .....	23
1.3.6 Цифрові методи взаємодії зі споживачами .....	24
1.4 Тренди і тенденції цифрової музичної індустрії	26
1.4.1 Сучасні платформи цифрової музичної індустрії .....	27
1.4.2 Сучасні підходи платформ цифрової музичної індустрії .....	28
Висновок до першого розділу	30
РОЗДІЛ 2 АНАЛІЗ ПРОГРАМНИХ РІШЕНЬ ПЛАТФОРМ ДОБИРАННЯ МУЗИЧНОГО КОНТЕНТУ	31
2.1 Архітектурні рішення систем добирання музичного контенту	31
2.1.1 Автоматичний збір даних .....	31
2.1.2 Рекомендаційні системи.....	31

2.1.3 Системи фільтрації та сортування .....	32
2.1.4 Масштабованість та надійність .....	32
2.2 Хмарні технології як основа серверної частини клієнт-серверної архітектури систем добирання музичного контенту .....	32
2.2.1 Публічні хмари .....	33
2.2.2 Переваги публічних хмар .....	34
2.2.3 Недоліки публічних хмар .....	36
2.2.4 Приватні хмари .....	36
2.2.5 Переваги приватних хмар .....	37
2.2.6 Недоліки приватних хмар .....	38
2.2.7 Гібридні хмари .....	38
2.2.8 Переваги гібридних хмар .....	39
2.2.9 Недоліки гібридних хмар .....	40
2.2.10 Вибір та порівняння можливостей публічних, приватних та гібридних хмар .....	40
2.3 Віртуалізація та контейнеризація як основа хмарних технологій .....	41
2.3.1 Визначення віртуалізації та її основні типи .....	43
2.3.2 Переваги та недоліки віртуалізації .....	44
2.3.3 Хмарна віртуалізація .....	46
2.3.4 Визначення контейнеризації, структура й основні принципи .....	47
2.3.5 Переваги та недоліки контейнеризації .....	49
2.3.6 Локальні і хмарні засоби реалізації контейнеризації .....	50
2.4 Використання мови програмування Swift для створення серверної та клієнтської частин мобільного застосунку добирання музичного контенту .....	52
2.4.1 Мобільний застосунок для ОС iOS як клієнтська частина застосунку .....	53
2.4.2 Можливості аудіовідтворення в ОС iOS .....	54
2.4.3 Varog як фреймворк для побудови серверної частини застосунку добирання музичного контенту .....	55
Висновок до другого розділу .....	56
Розділ 3. ПРОЄКТУВАННЯ, РОЗРОБЛЕННЯ ТА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДОБИРАННЯ МУЗИЧНОГО КОНТЕНТУ .....	57
3.1 Інженерія вимог до застосунку добирання музичного контенту .....	57

3.1.1 Постановка задачі .....	57
3.1.2 Аналіз функціоналу систем добирання музичного контенту .....	57
3.1.3 Функціональні можливості застосунку, що проєктується .....	59
3.1.4 Нефункціональні можливості застосунку, що проєктується .....	59
3.2 Архітектурні рішення застосунку добирання музичного контенту	60
3.2.1 Архітектура клієнтської частини застосунку .....	60
3.2.2 Архітектура клієнтської частини застосунку .....	61
3.3 Проєктування та реалізація застосунку добирання музичного контенту	62
3.3.1 Структура бази даних .....	62
3.3.2 Структура користувацького інтерфейсу.....	63
3.3.3 Захист персональних даних користувачів.....	64
3.3.4 Тестування програмної системи.....	64
Висновок до третього розділу	64
ВИСНОВОК	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66
ДОДАТКИ	71
ДОДАТОК А	71
ДОДАТОК Б	72

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- СДМК – система добирання музичного контенту
- WIPO – всесвітня конвенція з авторського права
- ВМ – віртуальна машина
- API – Application Programming Interface
- HTTP – HyperText Transfer Protocol
- БД – база даних
- ІС – інформаційна система
- iOS – мобільна операційна система від компанії Apple
- Swift – мова програмування для розробки мобільних iOS застосунків
- СУБД – система управління базами даних
- SPM – Swift Package Manager
- DTO – Data Transfer Object
- MVC – Model-View-Controller (Модель-Представлення-Контролер)
- Стрімінг – потокова передача даних
- Back-end – серверна частина застосунку
- Varog – бібліотека на Swift для написання back-end застосунків
- Xcode – інтегрована середовище розробки, створена Apple
- UIKit – бібліотека з базовими компонентами інтерфейсу
- JSON – текстовий формат даних, який представляє собою об'єкт або об'єкти
- REST – спосіб організації мережевих протоколів

## ВСТУП

Ера інформаційних технологій сьогодення трансформує способи створення та поглинання інформації людством. Зокрема, інтернет надав можливість користувачам мобільних пристроїв прослуховувати аудіотреки без необхідності попередньо купувати компакт-диск або ж копіювати їхній вміст безпосередньо до пам'яті девайсів. Дійсно, у минулому люди слухали музику вдома, у своїх автомобілях або на комп'ютерах, проте нині всі ці підходи переважно замінюються смартфоном. Це змінює спосіб взаємодії людей з музикою і відкриває нові можливості для музичної індустрії: як безпосередньо виконавців, так і технічної сторони реалізації прослуховування аудіотреків.

**Актуальність теми** зумовлена тим, що, згідно з дослідженням [1], 56% від загального часу прослуховування музики припадають на смартфони та радіо. Відповідний показник є доволі значним, тому виконання досліджень і розробка нових підходів до впровадження сучасних технічних рішень в цьому напрямку є актуальним та доречним.

**Метою кваліфікаційної роботи** є зручне добирання музичного контенту за допомогою розробленого мобільного застосунку добирання музичного контенту, а саме, його клієнтської та серверної частин, написаних однією мовою програмування.

### **Завдання дослідження:**

- дослідити сучасні методології розроблення застосунків;
- проаналізувати архітектурні рішення, технології та інструменти для реалізації застосунку;
- спроектувати, впровадити та поєднати клієнтську та серверну частини мобільного застосунку для добирання музичного контенту.

**Об'єктом дослідження** є процеси передачі та відтворення аудіофайлів.

**Предметом дослідження** є програмно-технічні, організаційні засади, принципи, підходи щодо побудови мобільного застосунку добирання музичного контенту.

**Методи дослідження:** теорія управління для дослідження теоретичних аспектів побудови застосунків, емпіричний аналіз і синтез систем, що застосовувався при вивченні прикладів сучасних методів побудови застосунків добирання музичного контенту.

**Практичне значення одержаних результатів** полягає у тому, що розроблений мобільний застосунок може стати зручним рішенням для добирання музичного контенту, а також слугувати практичним прикладом впровадження практики перевикористання коду[2] між клієнтською та серверною частинами застосунку, що написані однією мовою програмування (Swift).

**Апробація результатів дослідження** була виконана шляхом оприлюднення результатів дослідження (тез) у таких виданнях:

- Грицунь Я.Б. Захист інформації в прикладних інформаційних системах. Прикладні системи та технології в інформаційному суспільстві: зб. тез доповідей і наук. повідомл. учасників VI Міжнародної науково-практичної конференції. Київ, 2022. С. 118–123;
- Saiko V., Hrytsun Ya. Data Protection in the iOS Operating System. Інформаційні технології та взаємодії (сателітна): матеріали конференції. Київ, 2022. С. 134–135;
- Грицунь Я.Б. Оптимізація використання криптографічного фреймворку Keuchain: розробка зручного прикладного програмного інтерфейсу для зберігання конфіденційних даних в застосунках для операційної системи iOS. Проблеми кібербезпеки інформаційно-телекомунікаційних систем: Збірник матеріалів доповідей та тез. Київ, 2023. (в друці).

**Структура кваліфікаційної роботи:**

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, розподілених на підрозділи та висновку.

## Розділ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА СТАН МУЗИЧНОЇ ІНДУСТРІЇ В УМОВАХ ЦИФРОВІЗАЦІЇ

### 1.1 Стан розвитку цифрової музичної індустрії в Україні та світі, основні її поняття та зміст

У сучасному світі, цифрова музична індустрія знаходиться на перехресті технологій, культури та комерції, що призводить до значних змін у способах створення, розповсюдження та споживання музики[3]. В Україні та світі, цифрова музична індустрія відіграє важливу роль у формуванні сучасної музичної культури, а також має значний економічний вплив.

Одним з основних понять, пов'язаних з цифровою музичною індустрією, є цифрова музика. Цифрова музика відображає звукові сигнали у вигляді цифрових даних, що дозволяє їх зберігати, обробляти та передавати за допомогою комп'ютерних технологій[5]. Цей підхід забезпечує більшу гнучкість у створенні та маніпулюванні музикою, а також полегшує її поширення через Інтернет.

Інше важливе поняття в цифровій музичній індустрії – це цифрова дистрибуція. Цифрова дистрибуція передбачає електронне розповсюдження музики через Інтернет шляхом завантаження аудіофайлів або використання потокового відтворення[6]. Цей спосіб дозволяє музикантам і музичним виконавцям прямо взаємодіяти зі своєю аудиторією, обходячи традиційні медіакомпанії та фізичні носії.

Цифрова музична індустрія також включає в себе поняття стрімінгу музики. Стрімінг музики є формою цифрової дистрибуції, при якій користувачі мають можливість безпосередньо прослуховувати музику онлайн без необхідності завантажувати її. Цей спосіб споживання музики набуває все більшої популярності, забезпечуючи зручність та широкий доступ до музичного контенту[7].

Зміст цифрової музичної індустрії включає різноманітні аспекти, такі як створення музики за допомогою комп'ютерних програм та синтезаторів,

звукзапис та редагування, розробка музичних платформ та сервісів, юридичні аспекти авторських прав та ліцензування музичного контенту, аналіз даних про споживання музики, розвиток нових технологій у сфері звукзапису та музичних інструментів, музичний маркетинг та просування[4].

Україна, як і багато інших країн, спостерігає розширення цифрової музичної індустрії, причому зростають як кількість музикантів, що використовують цифрові технології для створення та поширення своєї музики, так і популярність цифрових музичних платформ серед слухачів. Розвиток цифрової музичної індустрії в Україні сприяє розширенню культурного впливу країни, збільшує можливості для творчих починань музикантів та сприяє економічному зростанню у галузі музичного бізнесу.

1.1.1 Визначення музичної індустрії, передумови та стан розвитку в умовах сьогодення.

Музична індустрія може бути визначена як сукупність суб'єктів, процесів та інституцій, що займаються виробництвом, дистрибуцією, просуванням та споживанням музичних творів. Вона включає музикантів, композиторів, виконавців, звукорежисерів, музичні лейбли, продюсерські компанії, концертні агентства, музичні магазини, цифрові платформи, медіа-компанії та широке коло споживачів музики.

Передумовами розвитку музичної індустрії в сучасних умовах є впровадження технологій, особливо цифрових, які значно змінили способи створення, запису, дистрибуції та споживання музики. Інтернет та цифрові технології дозволили музикантам самостійно записувати та продюсувати свою музику, а також безпосередньо взаємодіяти зі своєю аудиторією. Це знизило витрати на створення та поширення музичного контенту, зробило його більш доступним для масового споживання і стимулює зростання незалежних музичних проєктів.

У сучасних умовах стан розвитку музичної індустрії є динамічним і комплексним. З одного боку, цифрові технології привернули більше уваги до музики і розширили можливості для її споживання. З'явилися нові музичні платформи та сервіси, які дозволяють користувачам легко знаходити, стрімити та завантажувати музику за запитом. Музичні алгоритми рекомендацій активно використовуються для персоналізованого підбору музичного контенту[8].

З іншого боку, цифрова музична індустрія також постала перед викликами, пов'язаними з піратством, нелегальним поширенням музики та проблемами авторських прав[9]. Однак з'явилися також нові моделі ліцензування та монетизації музичного контенту, які дозволяють музикантам та правовласникам отримувати прибуток від своєї творчості.

Стан розвитку музичної індустрії в сучасних умовах відображається у постійному зростанні кількості музичних творів, їх різноманітності та доступності. Цифрові технології стимулюють інновації у сфері музичного творчості, сприяють появі нових жанрів та звукових експериментів. Розвиток цифрової музичної індустрії також має значний економічний вплив, забезпечуючи зайнятість та прибуток для багатьох суб'єктів галузі[10].

Україна, як і багато інших країн, спостерігає зростання цифрової музичної індустрії. Музиканти та виконавці активно використовують цифрові технології для створення, запису та просування своєї музики. Цифрові музичні платформи стають популярними серед українських слухачів, що сприяє поширенню та просуванню музичного контенту. За останні роки спостерігається зростання кількості українських музичних проектів, що отримують визнання як на внутрішньому, так і на міжнародному рівні.

#### 1.1.2 Стан цифрової музичної індустрії та її вплив на світову економіку

Цифрова музична індустрія в сучасних умовах має значний вплив на світову економіку, що виявляється у зростанні обсягів виробництва,

розповсюдження та споживання музичного контенту, а також у створенні нових робочих місць та економічних можливостей[11].

Один з ключових аспектів стану цифрової музичної індустрії – це зростання цифрового споживання музики. Впровадження стрімінгових платформ та цифрових музичних сервісів дозволяє користувачам миттєво мати доступ до широкого спектру музичного контенту. Це призводить до зростання популярності стрімінгових послуг, збільшення кількості передплатників та збільшення обсягів прибутку для цифрових музичних платформ[12]. За даними звіту International Federation of the Phonographic Industry (IFPI)[18], у 2020 році дохід від цифрового музичного контенту склав 62% від загального доходу глобальної музичної індустрії.

Цифрова музична індустрія також сприяє економічному зростанню через створення нових робочих місць та підтримку музичних талантів. Завдяки доступності цифрових технологій, музиканти та виконавці мають змогу самостійно створювати, записувати та просувати свою музику без потреби великих витрат. Це сприяє розвитку незалежної музичної сцени та забезпечує можливості для творчого виразу. Крім того, розширення цифрової музичної індустрії вимагає фахівців у галузі звукозапису, звукорежисури, цифрового маркетингу, аналітики даних та інших суміжних областей, що сприяє створенню нових робочих місць[13].

Значний вплив цифрової музичної індустрії на світову економіку також проявляється у формі ліцензування музичного контенту та колекціонування авторських винагород. За допомогою цифрових платформ та технологій, музичні правовласники мають змогу контролювати поширення свого контенту та отримувати відповідну винагороду за його використання[14]. Моделі ліцензування дозволяють музичним правовласникам отримувати роялті за використання їх творів на цифрових платформах та в медіа-проектах, що сприяє їхньому фінансовому стимулюванню.

## 1.2 Нормативно-правове регулювання цифрової музичної індустрії

Цифрова музична індустрія підлягає нормативно-правовому регулюванню як на національному, так і на міжнародному рівні. Ці нормативні рамки створюються для забезпечення захисту авторських прав, ліцензування музичного контенту, регулювання відносин між різними суб'єктами цифрової музичної індустрії та забезпечення справедливої конкуренції.

Національне нормативно-правове регулювання цифрової музичної індустрії відрізняється в різних країнах. Воно включає закони та правила, які стосуються авторських прав, ліцензування музичного контенту, договорів між музикантами та музичними лейблами, сплати роялті, захисту споживачів тощо. Такі закони і правила мають на меті забезпечити справедливий розподіл доходів від музичного контенту між музикантами, продюсерами, правовласниками та іншими суб'єктами галузі.

Одним з найважливіших міжнародних документів, що регулюють цифрову музичну індустрію, є Всесвітня конвенція з авторського права (WIPO Copyright Treaty)[19], прийнята Всесвітньою організацією інтелектуальної власності (WIPO). Цей документ встановлює мінімальні стандарти захисту авторських прав у цифровій сфері, включаючи захист від несанкціонованого відтворення, поширення та змінення музичного контенту. Країни-учасниці зобов'язуються впроваджувати в свої національні законодавства вимоги, встановлені в цьому документі.

Також важливими міжнародними організаціями, що регулюють цифрову музичну індустрію, є Міжнародна федерація фонографічної промисловості (IFPI) та Міжнародна конфедерація авторів і композиторів (CISAC). Ці організації сприяють встановленню стандартів ліцензування, збору роялті, захисту авторських прав та співпраці між різними суб'єктами цифрової музичної індустрії на міжнародному рівні.

Нормативно-правове регулювання цифрової музичної індустрії постійно еволюціонує, адаптуючись до змін у технологічному, економічному та соціокультурному середовищі. Розвиток цифрових технологій вимагає розробки нових законодавчих актів, спрямованих на забезпечення ефективного функціонування цифрової музичної індустрії та захисту правовласників та споживачів музики.

### 1.2.1 Проблеми та стан правового регулювання цифрової музичної індустрії в світі

Правове регулювання цифрової музичної індустрії в світі стикається з рядом проблем, які виникають у зв'язку зі швидким розвитком технологій та зміною способів створення, поширення та споживання музичного контенту. Деякі з основних проблем включають:

- авторські права та ліцензування: цифрова музична індустрія стикається з проблемами пов'язаними з визначенням та захистом авторських прав у цифровому середовищі. Незаконне копіювання, піратство та нелегальне поширення музичного контенту є поширеними явищами, що призводять до втрати прибутку для музикантів та правовласників. Крім того, складність ліцензування музичного контенту в цифровій сфері вимагає удосконалення процесу отримання та надання ліцензій для забезпечення справедливої компенсації за використання творів;
- справедлива оплата музикантів: цифрові музичні платформи стикаються з критикою щодо справедливості оплати музикантів та виконавців. Винагорода, яку вони отримують за стрімінг та продажі музики, часто вважається недостатньою, особливо для незалежних та менш відомих артистів. Виникає потреба у встановленні більш справедливих моделей розподілу доходів між музикантами та цифровими платформами;
- географічні обмеження: цифрові музичні платформи часто обмежують доступ до музичного контенту за географічними межами. Це створює

проблеми для музикантів, які бажають розповсюджувати свою музику на міжнародному рівні, а також для слухачів, які не мають можливості отримати доступ до певного музичного контенту через географічні обмеження;

- захист даних: цифрова музична індустрія працює з великим обсягом особистих даних користувачів, що створює проблеми з приватністю та захистом цих даних. Необхідно розвивати законодавство, що регулює збір, зберігання та використання особистих даних у музичній індустрії для забезпечення конфіденційності та захисту прав споживачів.

Стан правового регулювання цифрової музичної індустрії в світі може варіюватися залежно від країни та регіону. Деякі країни приймають нові закони та політики для адаптації до змін у цифровому середовищі, забезпечення захисту авторських прав та стимулювання розвитку музичної індустрії. Однак існують і проблеми з ефективним застосуванням та виконанням національних та міжнародних правових норм у цифровій сфері.

### 1.2.2 Міжнародна нормативно-правова база цифрової музичної сфери

Міжнародна нормативно-правова база цифрової музичної сфери складається з ряду міжнародних документів та угод, які встановлюють стандарти та принципи правового регулювання цифрової музичної індустрії. Основні документи, що регулюють цю сферу, включають:

- **всесвітня конвенція з авторського права (WIPO Copyright Treaty)[19]:** прийнята Всесвітньою організацією інтелектуальної власності (WIPO), ця угода встановлює мінімальні стандарти захисту авторських прав у цифровому середовищі. Вона включає положення про захист від несанкціонованого відтворення, поширення та змінення музичного контенту;
- **угода про торгові аспекти прав інтелектуальної власності (TRIPS)[20]:** угода, прийнята Всесвітньою організацією торгівлі (WTO), містить положення про захист авторських прав та інших форм інтелектуальної

власності. Вона встановлює мінімальні стандарти захисту авторських прав та зобов'язує країни-учасниці регулювати використання творів в цифровому середовищі;

- римська конвенція про захист аудіовізуальних спектаклів (Rome Convention)[21]: Ця конвенція, прийнята Європейською організацією інтелектуальної власності (EUIPO), стосується правового захисту аудіовізуальних творів, включаючи музичні відеокліпи та концертні виступи;
- директива про авторське право в цифровому одинадцятому столітті (Directive on Copyright in the Digital Single Market)[22]: Ця директива, прийнята Європейським Союзом, має на меті оновлення та гармонізацію законодавства про авторське право у цифровому середовищі. Вона включає положення про права авторів, ліцензування музичного контенту, захист авторських прав у цифровій сфері та відповідальність інтернет-провайдерів;
- міжнародна угода про торгівлю послугами (GATS)[23]: Угода, прийнята Всесвітньою організацією торгівлі (WTO), містить положення про торгівлю послугами, включаючи музичні послуги. Вона сприяє вільному перетину кордонів та забезпечує прозорість та недискримінацію у використанні цифрових музичних платформ та сервісів.

Ці міжнародні документи та угоди створюють основу для розвитку національного та міжнародного законодавства щодо цифрової музичної сфери. Вони встановлюють загальні стандарти та принципи, що сприяють захисту авторських прав, ліцензуванню музичного контенту та регулюванню відносин у цифровій музичній індустрії. Однак, кожна країна має свої внутрішні законодавчі акти, які встановлюють деталізовані правила та процедури регулювання цифрової музичної сфери відповідно до національних потреб та реалій.

### 1.2.3 Правовий аспект захисту персональних даних користувачів систем цифрової музичної індустрії

Захист персональних даних користувачів в системах цифрової музичної індустрії є важливою та актуальною проблемою в сучасному інформаційному суспільстві. Розвиток цифрових технологій та широке використання онлайн-платформ та сервісів спричиняють збільшення обсягів збирання, обробки та зберігання особистих даних користувачів. З метою забезпечення приватності, конфіденційності та контролю за використанням цих даних, правові норми та механізми були введені з метою регулювання цього процесу.

Одним з основних документів, що впливає на правовий аспект захисту персональних даних у цифровій музичній індустрії, є Загальний регламент про захист персональних даних (GDPR)[24]. Цей регламент, прийнятий Європейським Союзом, має на меті забезпечити високий рівень захисту персональних даних користувачів та забезпечити їх право на приватність у контексті цифрової сфери. Він встановлює обов'язки для суб'єктів, що збирають та обробляють персональні дані, включаючи операторів цифрових музичних платформ та сервісів.

Згідно з GDPR[24], персональні дані визначаються як будь-яка інформація, що стосується ідентифікованої або ідентифікованої фізичної особи. Це може включати імена, контактну інформацію, адреси, IP-адреси, дані про місцезнаходження, а також специфічні дані, пов'язані з музичними вподобаннями та звичками користувачів.

Оператори цифрових музичних платформ та сервісів мають бути свідомі своїх обов'язків щодо захисту цих персональних даних. Вони повинні забезпечувати адекватні технічні та організаційні заходи для захисту персональних даних від несанкціонованого доступу, витоку, втрати або знищення. Крім того, оператори повинні надавати достатню інформацію користувачам щодо збору, обробки та використання їх персональних даних,

а також забезпечувати право користувачів на доступ, виправлення та видалення своїх даних.

Додатково, існують інші правові норми, такі як Конвенція про захист осіб у зв'язку з автоматизованою обробкою персональних даних (Convention 108) Ради Європи, а також національні законодавства, які визначають правила та процедури щодо захисту персональних даних у контексті цифрової музичної індустрії.

#### 1.2.4 Правова база України в сфері цифрової музичної індустрії

Україна має відповідну правову базу, яка регулює сферу цифрової музичної індустрії. Ця база включає конституційні норми, закони, постанови та інші правові акти, що стосуються захисту авторських прав, ліцензування музичного контенту, розповсюдження музики через цифрові канали та інші аспекти цифрової музичної індустрії.

Одним з основних законодавчих актів у цій сфері є Закон України "Про авторське право та суміжні права"[25]. Цей закон визначає основні принципи та положення щодо авторського права, включаючи захист авторських прав на музичні твори, їх ліцензування та використання.

Також важливими правовими актами є Закон України "Про інформацію"[26], який встановлює правила щодо збору, обробки та захисту інформації, включаючи персональні дані, та Закон України "Про захист персональних даних"[27], який регулює збір, зберігання та використання персональних даних, включаючи дані користувачів цифрових музичних платформ та сервісів.

Україна також ратифікувала міжнародні угоди та конвенції, що стосуються авторського права та захисту інтелектуальної власності, такі як Бернська конвенція про захист літературних та художніх творів, Римська конвенція про захист аудіовізуальних спектаклів тощо. Ці міжнародні угоди доповнюють внутрішні правові норми та надають додатковий захист правовласникам музичних творів.

Окрім того, в Україні діє ряд постанов та наказів відповідних органів влади, таких як Міністерство культури та інші, які встановлюють процедури ліцензування, збору роялті, контролю та нагляду в цифровій музичній сфері.

### 1.3 Організаційно-економічні моделі цифрової музичної індустрії

Цифрова музична індустрія представляє собою комплексний економічний сектор, в якому створюється, поширюється та споживається музичний контент за допомогою цифрових технологій[15]. У цьому підрозділі розглянуто основні організаційно-економічні моделі, що використовуються в цифровій музичній індустрії.

#### 1.3.1 Модель прямого продажу (Direct Sales Model)

Модель прямого продажу передбачає прямий контакт між виконавцем та кінцевим споживачем музичного контенту без посередників. Виконавці можуть продавати свою музику на власних веб-сайтах або цифрових платформах, отримуючи прибуток від продажу або завантаження своїх пісень[17]. Ця модель надає більшу автономію виконавцям і дозволяє їм зберігати більшу частку прибутку, але вимагає власної просування та маркетингових зусиль.

#### 1.3.2 Модель стрімінгових платформ (Streaming Platform Model)

Модель стрімінгових платформ передбачає доступ до музики через цифрові платформи, які надають користувачам можливість стрімінгу музичного контенту в режимі реального часу. Користувачі мають можливість слухати музику за встановленою передплатою або безкоштовно з рекламними вставками. Ця модель забезпечує широкий доступ до музики та зручність споживання, а прибуток генерується з реклами, передплат та роялті.

#### 1.3.3 Модель ліцензування та синхронізації (Licensing and Synchronization Model)

Модель ліцензування та синхронізації передбачає використання музичного контенту у комерційних цілях, таких як рекламні ролики, фільми,

телевізійні шоу та інші медіа-продукти. У цій моделі музиканти та правовласники музичних творів отримують дохід через ліцензійні угоди з компаніями, які використовують їхню музику. Синхронізація музики зображуваним змістом додає емоційну та настроєву складову до медіа-продукту.

#### 1.3.4 Модель співпраці та партнерства (Collaboration and Partnership Model):

Модель співпраці та партнерства передбачає об'єднання виконавців, лейблів, продюсерів та інших стейкхолдерів для спільного створення, продажу та просування музичного контенту. Ця модель дозволяє пулу учасників спільно працювати над проектами, об'єднувати ресурси та знання для досягнення більшого успіху.

Кожна з цих організаційно-економічних моделей має свої переваги та обмеження, і вибір моделі залежить від конкретних потреб та цілей музикантів, правовласників та інших учасників цифрової музичної індустрії. Постійний розвиток технологій та зміна споживацьких звичок можуть призвести до зміни чи появи нових організаційно-економічних моделей у цій сфері.

#### 1.3.5 Традиційні бізнес-моделі музичної індустрії

Традиційні бізнес-моделі музичної індустрії є сукупністю стратегій та підходів, що використовуються у сфері створення, запису, продажу та використання музичного контенту з метою отримання прибутку[16]. Ці моделі є історично усталеними та базуються на розподілі ролей та функцій між виконавцями, лейблами, дистриб'юторами та іншими учасниками музичної індустрії.

Однією з найпоширеніших традиційних бізнес-моделей є продаж фізичних носіїв. Ця модель передбачає виготовлення та продаж музичних альбомів на фізичних носіях, таких як компакт-диски або вінілові платівки. Виконавці та лейбли залучаються до процесу виробництва, просування та

дистрибуції цих носіїв, отримуючи прибуток від продажу. Ця модель дозволяла контролювати виробництво та розповсюдження музики, але вимагала значних інвестицій у виробництво, логістику та маркетинг.

Іншою традиційною бізнес-моделлю є продаж цифрового контенту. Зараз велика частина музичного споживання відбувається через онлайн-магазини та стрімінгові платформи. Ця модель дозволяє користувачам придбати окремі пісні або альбоми в електронному форматі. Виконавці та лейбли отримують прибуток від продажу цифрового контенту, а витрати на виробництво та дистрибуцію мінімізуються. Проте, ця модель також стикається з проблемою незаконного завантаження та розповсюдження музики, що може впливати на прибутковість.

Окрім того, традиційна бізнес-модель музичної індустрії включає модель концертних виступів. Живі виступи стали важливим джерелом доходу для багатьох виконавців, особливо в умовах супернакопичених мас-медіа та безкоштовного доступу до музики. Концерти, фестивалі та інші події створюють можливість для виконавців заробляти гроші через продаж квитків, спонсорські угоди та мерчандайзинг. Крім того, концертні виступи відіграють важливу роль у побудові фан-бази та підтримці зв'язку з прихильниками музики.

### 1.3.6 Цифрові методи взаємодії зі споживачами

Цифрова музична індустрія прискорила розвиток нових методів взаємодії зі споживачами, завдяки використанню цифрових технологій та онлайн-платформ. У цьому розділі розглянуто ключові цифрові методи, які використовуються в цифровій музичній індустрії для залучення, залучення та задоволення потреб споживачів.

- Стрімінгові платформи: Стрімінгові платформи, такі як Spotify, Apple Music, Deezer та інші, стали популярними каналами для споживання музичного контенту. Ці платформи надають користувачам доступ до широкого вибору музики за підпискою або безкоштовно з рекламою.

Вони пропонують індивідуальні рекомендації на основі алгоритмів аналізу поведінки користувачів, що сприяє персоналізованому досвіду та залученню споживачів;

- соціальні медіа та мережі: соціальні медіа платформи, такі як YouTube, Instagram, Facebook та Twitter, стали важливими каналами для спілкування, просування та споживання музичного контенту. Виконавці та лейбли використовують соціальні медіа для побудови фан-бази, презентації нових пісень, відеокліпів та взаємодії зі своїми прихильниками. Це створює можливість для прямої взаємодії з аудиторією та збільшення її участі в музичному процесі;
- персоналізовані рекомендації та аналітика: цифрові технології дозволяють збирати та аналізувати великі обсяги даних про поведінку користувачів. Застосування алгоритмів машинного навчання та штучного інтелекту дозволяє створювати персоналізовані рекомендації музичного контенту для кожного користувача. Це сприяє покращенню користувацького досвіду та збільшенню задоволення від споживання музики;
- віртуальні та розширені реальності: віртуальна та розширена реальність надають можливості для нових форм споживання музичного контенту. Виконавці можуть створювати віртуальні концерти, інтерактивні відеокліпи та віртуальні світи, в яких користувачі можуть взаємодіяти з музикою та артистами у новому форматі. Це сприяє поглибленню зв'язку між артистами та шанувальниками, а також розширює межі творчості;
- краудсорсинг та співпраця: цифрова музична індустрія також сприяє співпраці між виконавцями та шанувальниками через краудсорсингові платформи та сервіси. Це дозволяє шанувальникам брати участь у створенні музики, реміксах, музичних відео та інших творчих проектах, що сприяє залученню та взаємодії зі споживачами.

Цифрові методи взаємодії зі споживачами цифрової музичної індустрії створюють нові можливості для взаємодії, залучення та задоволення потреб споживачів. Вони розширюють межі музичного досвіду, надають персоналізовані пропозиції та сприяють активній взаємодії між виконавцями та їх шанувальниками.

#### 1.4 Тренди і тенденції цифрової музичної індустрії

Цифрова музична індустрія є динамічною галуззю, яка постійно еволюціонує та пристосовується до змін у технологіях, споживацьких звичках та ринкових умовах. У цьому підрозділі розглянуто сучасні тренди і тенденції, що впливають на розвиток цифрової музичної індустрії.

- зростання стрімінгових платформ: стрімінгові платформи продовжують займати провідні позиції у цифровій музичній індустрії. За останні роки, кількість підписників на стрімінгових платформах значно збільшилась, що стало ключовим джерелом доходу для виконавців. Розширення глобального доступу до швидкого Інтернету та збільшення кількості пристроїв, які підтримують музичний стрімінг, сприяють подальшому зростанню цього сегменту;
- вплив штучного інтелекту та аналітики даних: штучний інтелект та аналітика даних знаходять все більше застосувань у цифровій музичній індустрії. Вони використовуються для рекомендацій музичного контенту, прогнозування трендів, аналізу споживацьких пристрастей та звичок. Це допомагає виконавцям, лейблам та стрімінговим платформам краще розуміти свою аудиторію та надавати персоналізовані послуги;
- розширення впливу соціальних медіа: соціальні медіа мають суттєвий вплив на музичну індустрію, особливо через спільноти шанувальників, відео-контент та взаємодію з артистами. Музиканти активно використовують соціальні медіа для просування своєї музики, побудови фан-бази та залучення нових шанувальників. Також

спостерігається зростання впливу віртуальних артистів та віртуальних концертів через соціальні медіа платформи;

- експерименти з форматами та взаємодією: музичні виконавці та творці стежать за новими форматами взаємодії з аудиторією. Це може включати віртуальні та розширені реальності, інтерактивні відеокліпи, мережеві виступи та інші форми експериментування з музичним виразом та комунікацією з фанами. Ці нові формати створюють унікальні можливості для залучення та взаємодії зі споживачами;
- розвиток незалежної та DIY-сцени: завдяки доступності технологій та інтернету, незалежні виконавці та творці мають більше можливостей для самостійного створення, запису та поширення своєї музики. DIY (Do-It-Yourself) підхід стає все популярнішим, що дозволяє виконавцям більшу автономію та контроль над своїм творчим процесом та кар'єрою.

Ці тренди і тенденції відображають головні напрямки розвитку цифрової музичної індустрії. Інновації в технологіях, зміна споживацьких звичок та взаємодія зі споживачами відкривають нові можливості для виконавців, лейблів, стрімінгових платформ та інших учасників галузі. І враховуючи швидке темпи розвитку, варто очікувати подальшого зміщення та появи нових трендів у цифровій музичній індустрії.

#### 1.4.1 Сучасні платформи цифрової музичної індустрії

Сучасні платформи цифрової музичної індустрії відіграють важливу роль у поширенні та споживанні музичного контенту. вони надають можливості для виконавців презентувати свою музику широкій аудиторії, а також дозволяють користувачам з легкістю знаходити та насолоджуватися музикою за допомогою сучасних технологій. ось кілька ключових платформ цифрової музичної індустрії:

- стрімінгові сервіси: сервіси стрімінгового аудіо, такі як Spotify, Apple Music, Amazon Music та Google Play Music, забезпечують

користувачам безпосередній доступ до великого каталогу музики. вони пропонують мільйони треків для безкоштовного або платного прослуховування, дозволяючи користувачам створювати власні плейлисти, отримувати рекомендації та насолоджуватися музикою без необхідності завантаження файлів;

- відеохостингові платформи: платформи відеохостингу, такі як YouTube, Vimeo та Dailymotion, є важливими джерелами музичного контенту. вони надають можливість виконавцям завантажувати відеокліпи, концертні записи та музичні виступи, а також дозволяють користувачам переглядати та взаємодіяти з цим контентом;
- соціальні медіа платформи: соціальні медіа, такі як Instagram, Facebook, Twitter та TikTok, стають важливими каналами для спілкування, взаємодії та просування музики. виконавці активно використовують ці платформи для побудови фан-бази, презентації нових треків та взаємодії зі своїми шанувальниками;
- музичні форуми та спільноти: музичні форуми та спільноти, такі як Reddit, Last.fm та Discogs, створюють платформу для обміну думками, дискусій та спільного споживання музичного контенту. вони забезпечують споживачам можливість обговорювати улюблену музику, ділитися рекомендаціями та знаходити нові твори.

Ці платформи є ключовими акторами у цифровій музичній індустрії, які створюють зручний та доступний спосіб споживання музики для користувачів і надають можливості для просування та просування виконавців. завдяки їм, музика стає більш доступною та широко поширеною.

#### 1.4.2 Сучасні підходи платформ цифрової музичної індустрії

Сучасні підходи платформ цифрової музичної індустрії відображають постійний розвиток та адаптацію до змін в технологічному середовищі та споживацьких потребах. З найбільш визначних підходів, які застосовуються сьогодні, можна виокремити такі:

- персоналізовані рекомендації: платформи цифрової музичної індустрії активно використовують алгоритми та штучний інтелект для надання персоналізованих рекомендацій користувачам. Вони аналізують музичні вподобання, прослуховування та інші дані, щоб пропонувати відповідний контент та допомагати відкривати нову музику, яка може сподобатися;
- взаємодія зі споживачами: сучасні платформи акцентують на взаємодії зі споживачами, створюючи місце для спілкування, коментарів та обміну думками між користувачами. Наприклад, вони надають можливість залишати відгуки, створювати спільноти шанувальників та ділитися враженнями про музичний контент;
- участь виконавців: багато платформ надають можливості для активної участі виконавців у процесі розповсюдження їх музики. Це може включати самостійне завантаження треків, побудову фан-бази, організацію концертів та зв'язок зі своїми шанувальниками через соціальні медіа;
- мережевий ефект: платформи цифрової музичної індустрії використовують мережевий ефект, де ростуть як користувачі, так і виконавці. Більше користувачів привертають більше виконавців, а більше виконавців привертають більше користувачів, що створює позитивний цикл розвитку;
- співпраця та партнерство: платформи співпрацюють зі співтовариством музичних лейблів, виконавців та інших гравців галузі для спільного розвитку та просування музичного контенту. Вони укладають ліцензійні угоди, організовують спільні проекти та співпрацюють у побудові сприятливих умов для всіх сторін.

Ці підходи показують, що платформи цифрової музичної індустрії активно вдосконалюють свої послуги та створюють інноваційні способи споживання, взаємодії та просування музичного контенту. Вони

відображають зусилля платформ для задоволення споживацьких потреб та стимулювання розвитку цифрової музичної індустрії.

#### Висновок до першого розділу

В цілому, цифрова музична індустрія має значний вплив на світову економіку, забезпечуючи зростання прибутку, створення нових робочих місць та розвиток музичного бізнесу. Цифрові технології допомагають розширити межі музичного творчості та забезпечують широкий доступ до музики для мільйонів людей по всьому світу.

Правове регулювання цифрової музичної індустрії постійно постає перед викликами, і його стан вимагає постійного оновлення та удосконалення для вирішення проблем, забезпечення справедливості та створення стійкого та розвинутого середовища для музичної індустрії.

Правовий аспект захисту персональних даних користувачів систем цифрової музичної індустрії є важливим елементом для забезпечення конфіденційності, приватності та контролю користувачів над їхніми особистими даними. Застосування регуляторних вимог, таких як GDPR[24], та впровадження відповідних заходів безпеки є необхідними для створення довіри та забезпечення ефективного функціонування цифрової музичної індустрії.

Традиційні бізнес-моделі музичної індустрії забезпечували джерело доходу для виконавців та інших учасників галузі. Однак, з появою цифрових технологій та зміни споживацьких звичок, музична індустрія поступово переходить до нових моделей, які використовують сучасні технології, такі як стрімінгові платформи та співпрацю з брендами.

Саме через перехід користувачів до прослуховування та добирання музичного контенту у смартфоні, дана робота надалі фокусуватиметься на цифрових платформах добирання музичного контенту; зокрема, детально розглядатимуться доступні мобільні застосунки за цією тематикою.

## РОЗДІЛ 2 АНАЛІЗ ПРОГРАМНИХ РІШЕНЬ ПЛАТФОРМ ДОБИРАННЯ МУЗИЧНОГО КОНТЕНТУ

### 2.1 Архітектурні рішення систем добирання музичного контенту

Системи добирання музичного контенту (СДМК) є невід'ємною частиною сучасного цифрового музичного середовища. Вони виконують важливу функцію у забезпеченні користувачів відповідним музичним вмістом, що відповідає їхнім уподобанням та настрою. Однак, належне функціонування СДМК вимагає глибокого розуміння їх архітектурних рішень та технологічних принципів. У цьому підрозділі розглядаються ключові архітектурні рішення, які лежать в основі систем добирання музичного контенту.

#### 2.1.1 Автоматичний збір даних

Одним із головних компонентів архітектури СДМК є автоматичний збір даних. Цей процес включає в себе збір та аналіз різноманітної інформації, що пов'язана з музичним контентом та користувачами. Для забезпечення ефективного збору даних в СДМК використовуються різноманітні методи, такі як аналіз метаданих, обробка аудіосигналів та аналіз змісту. Наприклад, використання алгоритмів машинного навчання дозволяє автоматично класифікувати музичний контент за жанрами, настроєм або іншими характеристиками.

#### 2.1.2 Рекомендаційні системи

Одним з найважливіших аспектів архітектури СДМК є рекомендаційні системи. Ці системи використовують зібрані дані про користувачів та музичний контент для генерації персоналізованих рекомендацій. Вони враховують уподобання користувачів, їхню історію прослуховування, спільноти, до яких вони належать, а також контекстуальну інформацію, таку як час доби або настрої. Рекомендаційні системи використовуються для підтримки особистого досвіду користувачів, забезпечуючи їм музичний контент, який вони ймовірно знайдуть привабливим.

### 2.1.3 Системи фільтрації та сортування

Ще одним важливим архітектурним рішенням у СДМК є системи фільтрації та сортування. Ці системи використовуються для відбору та представлення музичного контенту користувачам. Вони здатні враховувати різноманітні критерії, такі як популярність, релевантність або новизна. Системи фільтрації та сортування допомагають забезпечити користувачам зручний спосіб навігації та відкриття нового музичного контенту, враховуючи їхні індивідуальні вподобання.

### 2.1.4 Масштабованість та надійність

Якість роботи СДМК в значній мірі залежить від їхньої масштабованості та надійності. З огляду на великий обсяг музичного контенту та велику кількість користувачів, системи добирання музичного контенту повинні бути здатні працювати швидко та ефективно. Архітектурні рішення, такі як розподілені системи, кешування та горизонтальне масштабування, допомагають забезпечити масштабованість СДМК. Крім того, надійність системи досягається за допомогою механізмів резервного копіювання, реплікації даних та моніторингу.

Архітектурні рішення в системах добирання музичного контенту відіграють важливу роль у забезпеченні задоволення потреб користувачів у музичному вмісті. Автоматичний збір даних, рекомендаційні системи, системи фільтрації та сортування, а також масштабованість та надійність є ключовими компонентами архітектури СДМК. Розуміння та правильне впровадження цих архітектурних рішень допомагають створити ефективні та зручні системи, які забезпечують відповідний музичний контент для користувачів.

## 2.2 Хмарні технології як основа серверної частини клієнт-серверної архітектури систем добирання музичного контенту

Раніше, для розгортання програм інформаційні системи вимагали встановлення фізичного сервера на території компанії. Однак, такий підхід

виявився обмеженим з точки зору вартості та масштабованості для сучасних проєктів, зокрема веб-сайтів електронної комерції. Інвестиції в фізичне обладнання можуть стати значною перешкодою для розвитку проєкту, особливо в умовах непередбачуваного попиту в онлайн-бізнесі. Таким чином, потрібно шукати раціональніші рішення, які забезпечать гнучкість, масштабованість та вартість, а також знизять ризик витрат ресурсів. Один із таких прикладів – хмарна технологія, яка відповідає на всі ці вимоги та надає ефективні архітектурні рішення для систем добирання музичного контенту.

Хмарна технологія змінює парадигму розгортання інформаційних систем, забезпечуючи гнучкість, масштабованість та легкість розгортання на безпечних серверах, розташованих по всьому світу. У системах добирання музичного контенту, хмарні архітектурні рішення можуть бути використані для забезпечення доступу до музичного контенту для користувачів з різних куточків світу, забезпечуючи надійність та швидкість доступу.

Нижче розглянемо три найпоширеніші типи хмар.

## Рисунок 2.1 – Основні типи хмар

### 2.2.1 Публічні хмари

Публічні хмари відносяться до моделі хмарних обчислень, у якій ІТ-послуги надаються через Інтернет[28]. Будучи найпопулярнішою моделлю хмарних обчислювальних послуг, публічна хмара пропонує широкий вибір

рішень і обчислювальних ресурсів для задоволення зростаючих потреб організацій будь-якого розміру та вертикалі.

Визначальними особливостями публічного хмарного рішення є:

- висока еластичність і масштабованість;
- недорогий рівень ціноутворення на основі передплати.

Послуги в загальнодоступній хмарі можуть бути безкоштовними або на основі передплати, коли з користувача стягується плата залежно від обчислювальних ресурсів, які спожиті.

Обчислювальні функції можуть варіюватися від звичайних служб – електронної пошти, програм і сховища – до платформи ОС корпоративного рівня або інфраструктурних середовищ, які використовуються для розробки та тестування програмного забезпечення[29] .

Хмарний постачальник відповідає за розробку, керування та підтримку обчислювальних ресурсів, які спільно використовуються між декількома користувачами з усієї мережі.

Публічні хмари є найбільш поширеною та відомою моделлю хмарних обчислень. Усі ресурси, необхідні для роботи інфраструктури – сервери, сховища, мережеві компоненти та допоміжне програмне забезпечення – належать і управляються стороннім постачальником, і доступ до них мають користувачі через Інтернет через веб-браузер[30]. Прикладами публічних хмар є Amazon Web Services, Microsoft Azure і Google Cloud Platform.

### 2.2.2 Переваги публічних хмар

Серед основних переваг публічних хмар доцільно виокремити такі:

- зниження витрат: замість того, щоб робити великі початкові інвестиції для придбання та встановлення апаратного та програмного забезпечення, користувач просто платить за те, що використовує, іншими словами – перехід від моделі капітальних витрат (CAPEX) на модель операційних витрат (OPEX). Загальний необхідний бюджет

зазвичай набагато нижчий, ніж для традиційної локальної інфраструктури;

- відсутність необхідності обслуговувати сервер: постачальник послуг займається всім цим, тобто користувачам не потрібно залучати велику власну команду ІТ-спеціалістів для запуску, керування та оптимізації середовища;
- масштабованість за вимогою: постачальники володіють величезними ресурсами, які користувачі можуть збільшувати за потреби, щоб задовольнити періоди пікового попиту, а потім знову зменшувати масштаб;
- висока надійність: хмарні постачальники підтримують великі територіально розподілені мережі серверів разом із складними стратегіями резервування та відновлення після збоїв, щоб захистити клієнтів від збоїв і простоїв;
- знижена складність: у традиційних локальних системах до 80% часу та ресурсів ІТ-спеціалістів витрачається на просто «увімкнене світло». Публічна хмара усуває більшість цих витрат і складності. Це дає користувачам змогу підтримувати набагато менший штат ІТ-фахівців, які зосереджені на створенні та наданні високоцінних продуктів і послуг;
- гнучкі варіанти ціноутворення: хмарні постачальники пропонують різні рівні обслуговування. Вищі гарантовані рівні швидкості, доступності та забезпечення ресурсами мають перевагу; користувачі можуть вибрати їх для критично важливих робочих навантажень і заплатити нижчу ставку за менш важливі та маловикористовувані програми. Клієнти також можуть шукати найкраще поєднання ціни та продуктивності на сьогоднішньому висококонкурентному публічному хмарному ринку[31].

### 2.2.3 Недоліки публічних хмар

Серед основних недоліків публічних хмар доцільно навести такі:

- погано підходять для конфіденційних даних та робочих процесів: деякі галузі, наприклад фінанси та охорона здоров'я, вимагають дуже високого рівня безпеки та підлягають суворому контролю конфіденційності. Ці фактори роблять публічні хмари невідповідними до вимог або повністю виключають можливість їх використання;
- високі витрати: загальнодоступна хмара може спричинити високу загальну вартість власника у випадку інтенсивного розгортання на рівні підприємства в організаціях середнього та великого розміру;
- обмежена видимість і контроль за інфраструктурою: постачальник хмарних технологій керує, підтримує та захищає інфраструктуру відповідно до стандартів і стратегій, які вони встановлюють і застосовують. Цього може бути недостатньо для відповідності нормативним вимогам деяких галузей[32].

### 2.2.4 Приватні хмари

Приватна хмара – це обчислювальна інфраструктура, призначена для використання однією організацією. Її можна розмістити у власному центрі обробки даних або у стороннього постачальника послуг. Визначальною характеристикою є те, що ІТ-ресурси запускаються та обслуговуються в приватній мережі лише для вашого використання.

Приватна хмара стосується будь-якого хмарного рішення, призначеного для використання однією організацією. У приватній хмарі користувачі не діляться ресурсами хмарних обчислень з жодною іншою організацією.

Ресурси центру обробки даних можуть розташовуватися локально або керуватися стороннім постачальником за межами підприємства. Обчислювальні ресурси ізольовані та доставляються через захищену приватну мережу, а не через публічні з'єднання.

Приватну хмару можна налаштувати відповідно до чітких бізнес-потреб і потреб організації. Завдяки кращому огляду та контролю інфраструктури організації можуть керувати ІТ-робочими навантаженнями, чутливими до безпеки, без шкоди для продуктивності, які раніше досягалися лише за допомогою виділених локальних центрів обробки даних.

На відміну від публічних хмар, приватна хмара нікому з третіх осіб не надається. Це полегшує керування налаштуваннями та дотриманням нормативних вимог, тому приватні хмари часто використовуються фінансовими установами, урядовими установами та суворо регульованими галузями, які потребують високого контролю над своїми робочими навантаженнями.

#### 2.2.5 Переваги приватних хмар

Серед переваг приватних хмар доречно навести такі:

- сувора безпека: середовище, призначене для одного клієнта, забезпечує максимальний рівень захисту від несанкціонованого доступу до даних і зламів;
- відповідність нормативним вимогам: інфраструктуру приватної хмари можна налаштувати відповідно до строгих, обов'язкових правил конфіденційності та управління для конфіденційних робочих навантажень і даних;
- масштабованість: з технічної точки зору, приватні хмари можна масштабувати, хоча надання виділених ресурсів, які значну частину часу залишатимуться бездіяльними, може бути дорогим і неефективним;
- висока продуктивність: користувачі можуть легко обладнати приватну хмару відповідно до конкретних стандартів доступності та пропускну здатності;
- гнучкість реагування на мінливі потреби: оскільки середовище призначене лише для використання однією компанією, користувачі

можуть трансформувати його відповідно до розвитку технологій, ринків і можливостей.

#### 2.2.6 Недоліки приватних хмар

Проте, приватні хмари мають також деякі недоліки, серед яких:

- обмежений мобільний доступ: суворі заходи безпеки можуть ускладнити доступ до приватної хмари для користувачів мобільних пристроїв;
- високі накладні витрати: значні початкові інвестиції, тривалі цикли розгортання та великий власний ІТ-відділ — усе це необхідно для інфраструктури приватної хмари, якою володіє та керує сама організація;
- обмеження масштабованості: розширення може потребувати ресурсів і часу через необхідність придбання та налаштування як апаратного, так і програмного забезпечення.

#### 2.2.7 Гібридні хмари

Гібридні хмари поєднують публічні та приватні хмарні ресурси, щоб отримати переваги обох. Високі рівні інтеграції та керування є обов'язковими для безперебійного переміщення даних і програм між двома типами в міру зміни потреб. Конфіденційні операції, такі як фінансова звітність, наприклад, можна виконувати в приватній хмарі. Великі обсяги менш конфіденційних робочих навантажень, як-от електронна пошта, або навіть тимчасові робочі навантаження, розробка та тестування, можуть виконуватися в публічній хмарі.

Крім того, публічні хмарні ресурси можна використовувати для задоволення короткострокових стрибків попиту на програми приватної хмари (технологія, відома як «вибух хмари»). Це дозволяє уникнути високих витрат, пов'язаних із закупівлею та наданням додаткової потужності, яка буде використовуватися лише інколи.

Іншими словами, гібридна хмара надає краще з обох світів: отримання гнучкості у виборі оптимальної хмари для кожної програми та отримання можливості вільно переміщувати робочі навантаження між хмарами відповідно до мінливих обставин. Стратегія гібридної хмари дає змогу організації досягати своїх технічних і бізнес-цілей економніше та ефективніше, ніж це можна було б зробити лише за допомогою публічної чи приватної хмари.

### 2.2.8 Переваги гібридних хмар

Впровадження гібридної хмари до процесів підприємства надає ряд переваг:

- відповідність вимогам і безпека: конфіденційними програмами можна керувати приватно, а менш конфіденційні робочі навантаження можна розгортати в загальнодоступній хмарі;
- висока гнучкість: користувачі можуть швидко й ефективно реагувати на непередбачувані стрибки попиту за допомогою публічної хмарної інфраструктури та зменшувати масштаб, коли сплеск спадає, без впливу на інші робочі навантаження, що виконуються у приватній хмарі;
- оптимізація ресурсів: гібридна хмара дає змогу гнучко розгортати застосунки для максимального використання як локальних ресурсів, так і економії публічної хмарної інфраструктури;
- легкість і гнучкість: такі хмари дозволяють змінювати поєднання публічних і приватних хмар відповідно до мінливих потреб і можливостей. Гібридна хмара також дозволяє поступово модернізувати застарілі програми, перебудовуючи та переносючи їх до економічно ефективної загальнодоступної хмарної інфраструктури з часом;

- висока надійність: програми та дані можна розподілити між кількома центрами обробки даних для високоефективного резервування, відновлення після відмови та аварійного відновлення.

### 2.2.9 Недоліки гібридних хмар

Модель гібридної хмари все-ж таки має такі недоліки:

- висока вартість: експлуатація, обслуговування та оптимізація локального сегмента гібридної хмари є дорогою задачею, особливо для невеликих організацій;
- складність інфраструктури: щоб гібридна хмара забезпечувала максимальну вигоду, її публічні та приватні компоненти мають бути міцно пов'язані та організовані. Управління, інтеграція та безпека стають дедалі складнішими, оскільки кількість хмар зростає, особливо якщо вони надходять від різних постачальників.

### 2.2.10 Вибір та порівняння можливостей публічних, приватних та гібридних хмар

У той час як невеликі підприємства прагнуть до економічних переваг публічної хмари, великі організації часто використовують більше одного типу. Кожен вид хмари має свої переваги та недоліки. Все залежить від ресурсів і випадків використання користувачів.

Публічна хмара найбільше підходить для таких ситуацій:

- передбачувані обчислювальні потреби, наприклад послуги зв'язку для переважно фіксованої кількості користувачів;
- програми та служби, критичні для ІТ та бізнес-операцій;
- резервні ресурси, що виділяються для задоволення пікового використання;
- середовище розробки та тестування програмного забезпечення.

Приватна хмара часто використовується у таких випадках:

- суворо регульовані галузі та державні установи;

- технологічні компанії, яким потрібен надійний контроль і безпека їхнього ІТ-робочого навантаження та базової інфраструктури;
  - організації з ресурсами та вимогами для інвестування в передові технології, що забезпечують найвищу продуктивність і доступність.
- Модель гібридної хмари найбільше підходить для:
- підприємств, які обслуговують численні ринки з різними вимогами до ІТ-безпеки, нормативними вимогами та продуктивністю;
  - оптимізації загальних інвестицій у хмару з одночасним використанням переваг моделей публічної та приватної хмари;
  - забезпечення підвищеної безпеки для таких рішень, як робочі навантаження SaaS, які потрібно надавати через безпечні приватні мережі;
  - гнучка хмарна стратегія, яка вимагає часте чергування постачальників і хмар для отримання найкращого поєднання ціни та продуктивності для різних робочих навантажень.

Узагальнимо порівняння можливостей різних видів хмар таблицею.

Таблиця 2.1 – Порівняльна таблиця трьох типів хмар

Характеристика/тип хмари	Публічна	Приватна	Гібридна
Перетворення CapEx в OpEx	відмінно	відсутнє	погано
Продуктивність і доступність	погано	відмінно	відмінно
Гнучкість конфігурацій	відмінно	відмінно	відмінно
Масштабованість	відмінно	добре	добре
Прогнозованість витрат	відмінно	погано	погано
Підтримка застарілих додатків	відсутнє	відмінно	відмінно
Безпека даних	добре	відмінно	відмінно
Продуктивність при роботі з потужними аналітичними онлайн-системами	погано	відмінно	відмінно
Продуктивність при роботі з «важкою» графікою	погано	відмінно	відмінно

Контроль над ІТ-інфраструктурою з боку власника даних	відсутнє	відмінно	добре
---	----------	----------	-------

### 2.3 Віртуалізація та контейнеризація як основа хмарних технологій

Віртуалізація та контейнеризація – це дві сучасні технології, які дозволяють ефективно використовувати ресурси комп'ютерних систем. Віртуалізація – це процес створення віртуальних машин, які імітують повноцінну операційну систему на одному фізичному сервері. Контейнеризація – це процес ізоляції окремих програм або сервісів в легковагових контейнерах, які спільно використовують ядро операційної системи. Обидві технології мають свою історію, процес розвитку, актуальність та питання, які вони вирішують.

Історія віртуалізації починається з 1960-х років, коли компанія ІВМ розробила першу систему віртуалізації для своїх мейнфреймів. Це дало можливість запускати декілька операційних систем на одному фізичному пристрої, збільшуючи його продуктивність та надійність. Згодом виникли інші види віртуалізації, такі як апаратна, програмна, сховищ даних, мережева тощо. В 1990-х роках виникла концепція хмарних обчислень, яка базувалася на використанні великої кількості взаємопов'язаних віртуальних машин.

Історія контейнеризації починається з 1979 року, коли у систему UNIX було додано функцію `chroot`, яка дозволяла запускати програми в ізольованому середовищі. Згодом з'явилися інші технології для контейнеризації, такі як FreeBSD jails, Solaris zones, Linux-VServer тощо. У 2008 році було створено проект LXC (Linux Containers), який став основою для багатьох сучасних реалізацій контейнеризації. У 2013 році було запущено Docker – платформу для управління контейнерами, яка швидко набрала популярності серед розробників та системних адміністраторів.

Актуальність віртуалізації та контейнеризації полягає у тому, що вони дозволяють оптимізувати використання ресурсів комп'ютерних систем, зменшити витрати на обладнання та електроенергію, підвищити гнучкість та

швидкість розгортання програмних сервісів. Крім того, віртуалізація та контейнеризація дозволяють ефективніше використовувати обчислювальні ресурси для тестування програм, розгортання веб-серверів, побудови складних мережових топологій та інших завдань.

Однак, при використанні цих технологій необхідно враховувати деякі питання, такі як збільшення складності управління інфраструктурою, збільшення навантаження на мережу та зменшення продуктивності за рахунок додаткового прошарку віртуалізації. Крім того, необхідно враховувати питання безпеки, так як віртуальні середовища можуть бути менш стійкими до кібератак[36].

### 2.3.1 Визначення віртуалізації та її основні типи

Віртуалізація – це процес створення віртуальних версій фізичних ресурсів, таких як сервери, мережі, сховища даних, операційні системи та програми. Віртуалізація дозволяє оптимізувати використання ресурсів, покращити продуктивність, гнучкість та безпеку ІТ-середовища.

Існує кілька основних типів віртуалізації, які можна розрізнити: за об'єктом віртуалізації та методом її реалізації. Найпоширенішими типами віртуалізації є:

- віртуалізація сервера: це тип віртуалізації, при якому один фізичний сервер поділяється на кілька ВМ, кожна з яких може виконувати свою операційну систему та програми. ВМ ізолюються одна від одної та використовують ресурси фізичного сервера за допомогою спеціального програмного забезпечення - гіпервізора. Віртуалізація сервера допомагає зменшити кількість фізичних серверів, збільшити ефективність використання ресурсів, полегшити управління та масштабування ІТ-середовища;
- віртуалізація мереж: це тип віртуалізації, при якому створюються віртуальні мережові елементи, такі як комутатори, маршрутизатори, фаєрволи, балансувальники навантаження тощо. Віртуальні мережові

елементи можуть бути розгорнуті на фізичних або віртуальних серверах та управлятися за допомогою централізованого програмного забезпечення. Віртуалізація мереж дозволяє забезпечити більшу гнучкість, безпеку та автоматизацію мережевих сервісів;

- віртуалізація сховищ даних: це тип віртуалізації, при якому створюються віртуальні сховища даних з різних фізичних носіїв даних, таких як жорсткі диски, SSD-диски, флеш-пам'ять тощо;
- віртуалізація програмного забезпечення: це процес створення віртуальної платформи для запуску додатків на різних операційних системах або апаратних архітектурах. Для цього використовується спеціальне програмне забезпечення - емулятор або інтерпретатор, який моделює поведінку і функціонування потрібної платформи[36].

### 2.3.2 Переваги та недоліки віртуалізації

Серед переваг використання віртуалізації можна виділити такі:

- економія ресурсів: віртуалізація дозволяє ефективно використовувати ресурси фізичного сервера, такі як процесор, пам'ять, диск і мережа. Завдяки віртуалізації можна запускати декілька операційних систем і додатків на одному сервері, зменшуючи кількість необхідного обладнання і електроенергії;
- гнучкість і масштабованість: віртуалізація дозволяє легко створювати, копіювати, переміщувати і видаляти віртуальні машини без впливу на функціонування інших віртуальних машин або фізичного сервера. Також віртуалізація дозволяє динамічно змінювати розподіл ресурсів між віртуальними машинами в залежності від потреб;
- безпека і ізоляція: віртуалізація забезпечує високий рівень безпеки і ізоляції віртуальних машин. Кожна віртуальна машина має свою операційну систему і додатки, які не можуть бути заражені або пошкоджені іншими віртуальними машинами або фізичним сервером. Також в разі необхідності можна шифрувати дані на віртуальних

машинах або застосовувати різні політики безпеки для кожної віртуальної машини;

- сумісність і портативність: віртуалізація дозволяє запускати різноманітні операційні системи і додатки на одному фізичному сервері, незалежно від їх архітектури і версії. Також віртуалізація дозволяє легко переносити віртуальні машини між різними фізичними серверами або хмарними сервісами, зберігаючи всю конфігурацію і стан віртуальних машин;

Однак, слід враховувати і недоліки такого підходу:

- витрати на ліцензування: для запуску віртуальних машин потрібно мати ліцензії на операційну систему і програмне забезпечення, яке використовується на них. Це може збільшити вартість віртуалізації, особливо якщо потрібно багато віртуальних машин з різними конфігураціями;
- швидкодія: віртуалізація вимагає додаткових ресурсів від фізичного сервера, таких як процесор, пам'ять, диск і мережа. Це може призвести до зниження продуктивності фізичного сервера і вплинути на швидкодію віртуальних машин. Також можливі конфлікти і залежності між віртуальними машинами, якщо вони використовують спільні ресурси;
- безпека: віртуалізація створює новий рівень складності для захисту інформації і систем. Все програмне забезпечення на віртуальних машинах повинно бути оновлене і захищене від загроз. Також потрібно контролювати доступ до віртуальних машин і їх конфігурації. Якщо фізичний сервер або програмне забезпечення для віртуалізації будуть скомпрометовані, то це може призвести до порушення безпеки всіх віртуальних машин на ньому;
- сумісність: не всі операційні системи і програмне забезпечення можуть працювати на віртуальних машинах без проблем. Деякі можуть мати

певні обмеження або вимоги до апаратного забезпечення або конфігурації. Також можуть бути проблеми з підтримкою драйверів або периферійних пристроїв. Це може ускладнити процес встановлення і налаштування віртуальних машин і потребувати додаткового часу і ресурсів[36].

### 2.3.3 Хмарна віртуалізація

Хмарна віртуалізація – це технологія, яка дозволяє створювати та використовувати віртуальні ресурси в хмарному середовищі. Віртуалізація дозволяє розділяти фізичні ресурси (наприклад, сервери, мережі, сховища даних) на логічні одиниці (наприклад, віртуальні машини, віртуальні мережі, віртуальні сховища), які можна динамічно призначати та змінювати за потребами користувачів та застосунків.

Хмарна віртуалізація має багато особливостей, які роблять її привабливою для розгортання хмарних сервісів. Розглянемо деякі з них:

- масштабованість: хмарна віртуалізація дозволяє швидко масштабувати ресурси вгору або вниз залежно від поточного навантаження та запиту;
- ефективність: хмарна віртуалізація дозволяє оптимально використовувати фізичні ресурси, зменшуючи витрати на електроенергію, охолодження та обслуговування;
- гнучкість: хмарна віртуалізація дозволяє легко переносити та мігрувати віртуальні ресурси між різними фізичними локаціями та хмарними провайдерами;
- ізоляція: хмарна віртуалізація дозволяє ізолювати віртуальні ресурси один від одного, забезпечуючи безпеку та конфіденційність даних та застосунків;
- стандартизація: хмарна віртуалізація дозволяє створювати стандартні та унормовані образи віртуальних ресурсів, яким можна легко керувати та оновлювати.

Хмарна віртуалізація є однією з ключових технологій, що лежить в основі хмарних обчислень. Вона дозволяє користувачам звертатися до потрібних ресурсів, таких як обчислювальні, мережеві та сховища даних, віртуальним шляхом, без необхідності мати власні фізичні сервери та обладнання[36].

Використання хмарної віртуалізації дозволяє зменшити витрати на придбання та підтримку обладнання, а також забезпечує більшу гнучкість та ефективність використання ресурсів. Крім того, хмарна віртуалізація забезпечує високу ступінь безпеки та конфіденційності даних, оскільки віртуальні ресурси можуть бути ізольовані один від одного та захищені від несанкціонованого доступу.

Одним з найбільш популярних прикладів хмарної віртуалізації є віртуальні приватні сервери (VPS), які надають користувачам доступ до віртуальних серверів у хмарному середовищі. Також хмарна віртуалізація дозволяє розгортати та керувати хмарними сервісами, такими як електронна пошта, зберігання даних та веб-хостинг, що робить її незамінною для бізнесу та інших організацій[36].

#### 2.3.4 Визначення контейнеризації, структура й основні принципи

Контейнеризація – це метод розподіленого обчислення, який дозволяє запускати ізольовані процеси на одній фізичній машині або в хмарному середовищі. Контейнеризація забезпечує високу ефективність, гнучкість та надійність застосунків, оскільки вони не залежать від операційної системи або апаратного забезпечення[33].

Історія контейнеризації в ІТ сягає 1979 року, коли було запроваджено концепцію chroot в UNIX V7. Chroot дозволяв встановити кореневий каталог для окремого процесу та його нащадків, тим самим ізолюючи їх від решти файлової системи. Це було першим кроком до створення віртуальних середовищ для програм.

У 2000-х роках з'явилися нові технології контейнеризації, такі як FreeBSD Jails, Solaris Zones, Linux VServer, OpenVZ, LXC та інші. Вони використовували різні механізми ядра ОС для ізоляції процесів, мереж, користувачів, файлових систем тощо. Однак ці технології були нестандартизовані, несумісні та складні у використанні.

У 2013 році компанія Docker випустила свою платформу для контейнеризації, яка стала дуже популярною та визначила новий стандарт у галузі. Docker використовував LXC як основу для своїх контейнерів, але надавав простий та зручний інтерфейс для їх створення, управління, розподілу та оркестрації. Docker також запровадив концепцію образів контейнерів - статичних снапшотів програмного забезпечення, які можна легко переносити між хостами.

З того часу контейнеризація в ІТ продовжує розвиватися та отримувати все більше прихильників. З'явилися нові проекти та інструменти для контейнеризації, такі як rkt, podman, buildah, cri-o, containerd, Kubernetes, Docker Swarm, Mesos, Nomad тощо.

Основні принципи контейнеризації такі:

- кожен контейнер має свою власну файловою систему, мережевий стек, процеси та ресурси, які ізольовані від інших контейнерів;
- контейнери можуть легко переноситися між різними середовищами без зміни коду або конфігурації застосунку;
- контейнери можуть швидко створюватися, запускатися, зупинятися та видалятися за допомогою стандартних інструментів та команд;
- контейнери можуть масштабуватися горизонтально або вертикально в залежності від потреб застосунку та навантаження на систему;
- контейнери можуть співпрацювати між собою за допомогою мережевих протоколів та спеціальних механізмів оркестрації[34].

Контейнери мають свою структуру, яка складається з таких складових елементів:

- образ контейнера – це файл, який містить усе необхідне для запуску контейнера, таке як код, бібліотеки, конфігурація тощо. Образи контейнерів можна створювати, зберігати і розповсюджувати за допомогою спеціальних репозиторіїв, наприклад Docker Hub;
- екземпляр контейнера – це запущений образ контейнера, який має свою адресу, порти, об'єм пам'яті й інші ресурси. Екземпляри контейнерів можна створювати, запускати, зупиняти і видаляти за допомогою команд або інтерфейсу користувача;
- мережа контейнерів – це спосіб забезпечення зв'язку між контейнерами та зовнішнім світом. Мережа контейнерів може мати різні типи та може бути налаштована за допомогою параметрів або конфігураційних файлів;
- сховище (volume) контейнера – це спосіб збереження і передачі даних між контейнерами або між контейнером і хостом. Сховище контейнера може бути створене як частина образу контейнера або окремо в процесі роботи[35].

### 2.3.5 Переваги та недоліки контейнеризації

Використання підходу контейнеризації має багато переваг, таких як:

- ефективність ресурсів: контейнери використовують менше пам'яті, процесорного часу та дискового простору, ніж віртуальні машини, оскільки вони не мають повної операційної системи. Це дозволяє запускати більше контейнерів на одному фізичному сервері або хмарному сервісі, зменшуючи витрати на інфраструктуру та енергоспоживання;
- швидкість розгортання: контейнери можуть бути створені та запущені за лічені секунди, що полегшує швидке оновлення та масштабування програмного забезпечення. Контейнери також спрощують процес розробки та тестування, оскільки вони гарантують однакове середовище виконання на різних платформах та пристроях;

- безпека та ізоляція: контейнери ізолюють програмне забезпечення від зовнішнього впливу, що зменшує ризик вразливостей та атак. Контейнери також дозволяють застосовувати специфічні політики безпеки та доступу для кожного контейнера окремо, що покращує контроль та аудит;
- сумісність та портативність: контейнери можуть бути легко перенесені між різними операційними системами, хмарними провайдерами та архітектурами, не втрачаючи функціональності та продуктивності. Контейнери також сприяють стандартизації та інтеграції програмного забезпечення, оскільки вони дотримуються загальних форматів та протоколів[36].

Однак контейнеризація також має деякі недоліки, які слід враховувати при її застосуванні.

Один з недоліків контейнеризації – це складність управління та моніторингу великої кількості контейнерів, які можуть бути розгорнуті на різних середовищах. Контейнери потребують спеціальних інструментів та платформ для оркестрації, конфігурації, оновлення, резервного копіювання та виявлення проблем. Це може збільшити складність ІТ-операцій та вимагати додаткових навичок та ресурсів від ІТ-персоналу.

Інший недолік контейнеризації – це потенційна втрата продуктивності та якості програмного забезпечення через надмірну абстракцію та стандартизацію. Контейнери ізолюють програмне забезпечення від особливостей операційної системи та апаратного забезпечення, що може призвести до погіршення продуктивності та сумісності. Контейнери також можуть приховувати деталі реалізації та залежностей програмного забезпечення, що може ускладнити налагодження та тестування. Крім того, контейнери можуть сприяти фрагментації та дублюванню коду через використання багатьох джерел та версій програмного забезпечення[36].

### 2.3.6 Локальні і хмарні засоби реалізації контейнеризації

Основні інструменти для контейнеризації, зокрема Docker та Kubernetes, мають багато переваг для розробників та адміністраторів.

Docker – це платформа для створення, розгортання та управління контейнерами. Контейнер Docker - це ізольоване середовище, яке містить все необхідне для запуску програми: код, бібліотеки, налаштування тощо. Контейнери Docker можуть бути легко перенесені між різними хостами, оскільки вони не залежать від операційної системи або апаратного забезпечення. Крім того, контейнери Docker дозволяють забезпечити високу продуктивність, ефективне використання ресурсів та швидке масштабування.

Kubernetes – це система для оркестрації контейнерів. Оркестрація контейнерів - це процес автоматизації розгортання, координації та управління контейнерами на кластерах серверів. Kubernetes дозволяє керувати контейнерами Docker за допомогою декларативних конфігурацій, які визначають бажаний стан застосунків.

Docker та Kubernetes є основними інструментами для контейнеризації, які допомагають розробникам та адміністраторам створювати надійні, гнучкі та ефективні застосунки.

Окрім Docker та Kubernetes, існують й інші інструменти для контейнеризації, такі як LXC/LXD, rkt, OpenVZ, і т.д. Кожен з цих інструментів має свої переваги та недоліки, тому вибір конкретного інструменту залежить від потреб проекту та вмінь користувача.

Для керування контейнерами у хмарних середовищах використовуються спеціальні платформи, так звані оркестратори контейнеризованих додатків. Найпопулярніші з них на сьогоднішній день - Kubernetes, Docker Swarm, Amazon ECS, Google Kubernetes Engine (GKE) та Microsoft Azure Kubernetes Service (AKS). Оркестратори контейнеризованих додатків дозволяють керувати розгортанням, масштабуванням, моніторингом та управлінням контейнерами в автоматизованому режимі,

забезпечуючи високу доступність та надійність додатків у хмарних середовищах. Використання оркестраторів також дозволяє ефективно використовувати ресурси хмарної інфраструктури та забезпечити балансування навантаження між контейнерами[36].

#### 2.4 Використання мови програмування Swift для створення серверної та клієнтської частин мобільного застосунку добирання музичного контенту

У сучасному цифровому середовищі, розробка мобільних застосунків для добирання музичного контенту вимагає вибору відповідних технологій та мов програмування. Одним зі засобів розробки, який набуває популярності, є мова програмування Swift, розроблена компанією Apple. У даному розділі розглядається використання мови програмування Swift для створення серверної та клієнтської частин мобільного застосунку добирання музичного контенту та його переваги у порівнянні з іншими мовами програмування.

Swift є мовою програмування загального призначення, яка поєднує в собі високу продуктивність та безпеку з простотою та елегантністю синтаксису. Вона надає розробникам широкий набір інструментів для створення ефективних та надійних мобільних додатків. Для створення серверної частини мобільного застосунку добирання музичного контенту, Swift може бути використана разом зі SwiftNIO або Vapor - фреймворками для розробки високопродуктивних серверних застосунків. Ці фреймворки надають потужні інструменти для обробки запитів, маршрутизації, збереження даних та забезпечення безпеки.

Однією з переваг використання Swift для створення клієнтської частини мобільного застосунку є його безпосередня інтеграція з операційною системою iOS. Swift дозволяє розробникам використовувати всі сучасні функції та можливості пристроїв Apple, зокрема камеру, сенсори, GPS і багато іншого. Це дозволяє створювати інтерактивні та

багатофункціональні мобільні застосунки для добирання музичного контенту[38].

Крім того, Swift пропонує розробникам високу швидкість виконання та ефективне управління пам'яттю. Вбудовані можливості мови, такі як автоматичне керування пам'яттю та оптимізація компілятора, дозволяють досягти високої продуктивності і швидкості роботи мобільного застосунку добирання музичного контенту[39].

#### 2.4.1 Мобільний застосунок для ОС iOS як клієнтська частина застосунку

У сучасному цифровому середовищі мобільні пристрої з операційною системою iOS від компанії Apple є популярними платформами для розробки та використання застосунків добирання музичного контенту. Мобільний застосунок для ОС iOS, який виступає в якості клієнтської частини, є ключовим елементом для забезпечення користувачам зручного та персоналізованого доступу до музичних ресурсів.

Клієнтська частина мобільного застосунку добирання музичного контенту, розроблена для ОС iOS, здатна забезпечити інтерактивність та зручний інтерфейс користувача. Вона може включати в себе такі компоненти, як екран входу, домашню сторінку зі списком рекомендованого контенту, пошукову функцію, відтворювач медіа, віджети, сповіщення тощо. Клієнтська частина дозволяє користувачам переглядати, відтворювати та взаємодіяти з музичним контентом, додавати пісні до списку відтворення, оцінювати та залишати коментарі.

Одним з ключових аспектів клієнтської частини є її сумісність та оптимізація для пристроїв з операційною системою iOS. Мова програмування Swift, рекомендована для розробки застосунків для ОС iOS, надає розробникам доступ до всіх функціональних можливостей та API платформи[40]. Завдяки цьому, розробники можуть використовувати специфічні функції та інтегрувати можливості, такі як Touch ID, Face ID,

геолокація та інші, для забезпечення більш глибокої та персоналізованої взаємодії з музичним контентом[41].

Клієнтська частина мобільного застосунку для ОС iOS також може використовувати інші технології та бібліотеки, такі як AVFoundation, Core Data, Alamofire та інші, для реалізації функціональності, такої як відтворення аудіо, кешування даних, забезпечення мережевої взаємодії[42] тощо.

#### 2.4.2 Можливості аудіовідтворення в ОС iOS

Аудіовідтворення є важливою складовою функціональності мобільних пристроїв, які працюють під операційною системою iOS. Розробка програмного забезпечення, яке забезпечує відтворення аудіофайлів з високою якістю звуку та різноманітними функціями, вимагає знань про можливості, які надає операційна система iOS. У цьому підрозділі розглянемо різноманітні можливості аудіовідтворення в операційній системі iOS та їх використання для розробки застосунків для добирання музичного контенту.

Фреймворк AVFoundation в операційній системі iOS надає розробникам широкі можливості для роботи з аудіофайлами. Використовуючи цей фреймворк, розробники можуть завантажувати, відтворювати, записувати, змінювати гучність та керувати аудіофайлами. За допомогою класів, які надає AVFoundation, можна створювати музичні плеєри, налаштовувати ефекти звуку, реалізовувати потокове аудіо та реалізовувати інші функції, які сприяють забезпеченню зручного та якісного аудіовідтворення в мобільних застосунках[43].

Фреймворк MediaPlayer – це ще один інструмент для роботи з аудіофайлами в операційній системі iOS. Цей фреймворк надає можливості для відтворення музики, керування відтворенням, створення плейлистів та використання віджетів для керування музичним відтворенням зі сторонніх додатків. За допомогою MediaPlayer, розробники можуть створювати

музичні додатки зі зручним інтерфейсом користувача та різноманітними функціями для навігації та управління музичним контентом[44].

Системні функції та API. Операційна система iOS також надає розробникам доступ до системних функцій та API, які допомагають в реалізації аудіовідтворення. Наприклад, можливості пристрою, такі як використання аудіо-вихідних пристроїв, регулювання гучності, мультитаскінг і фонове відтворення, доступні для розробників через відповідні API[45]. Ці функції дозволяють розробникам створювати застосунки, які надають розширені можливості аудіовідтворення та покращують взаємодію з користувачем.

### 2.4.3 Varog як фреймворк для побудови серверної частини застосунку добирання музичного контенту

В розробці застосунків для добирання музичного контенту, серверна частина відіграє важливу роль у забезпеченні надійності, масштабованості та продуктивності системи. Один із фреймворків, який набуває популярності для розробки серверної частини, є Varog. У цьому підрозділі розглянемо Varog як фреймворк для побудови серверної частини застосунку добирання музичного контенту та його переваги у контексті цього типу застосунків.

Varog є фреймворком для розробки серверних додатків, написаних на мові програмування Swift. Він базується на протоколах SwiftNIO, які забезпечують високошвидкісну обробку мережевих запитів. Використання Varog дозволяє розробникам ефективно створювати та розгортати серверні застосунки для обробки запитів, що стосуються добирання музичного контенту.

Однією з основних переваг Varog є його простота використання та зрозумілий синтаксис мови Swift. За допомогою Varog, розробники можуть швидко використовувати потужність мови Swift для реалізації функціональності серверної частини, такої як обробка запитів, маршрутизація, робота з базами даних та безпека.

Крім того, Vapor надає розробникам гнучкі можливості для масштабування системи. Завдяки асинхронній природі SwiftNIO, Vapor може обробляти тисячі одночасних з'єднань із високою продуктивністю. Це робить його ідеальним вибором для серверної частини застосунків добирання музичного контенту, де швидка обробка багатьох запитів є важливою вимогою.

#### Висновок до другого розділу

Вибір між публічними, приватними та гібридними хмарними рішеннями залежить від низки факторів, сценаріїв використання та обмежень. На реальних проєктах, розробники часто прагнуть використовувати всі три типи хмарних рішень для охоплення всього спектру переваг та недоліків, які пропонуються.

Мобільний застосунок для ОС iOS, що виступає в ролі клієнтської частини застосунку добирання музичного контенту, відіграє важливу роль у забезпеченні зручного та персоналізованого досвіду користувачів. Завдяки використанню мови програмування Swift та інтеграції зі специфічними можливостями операційної системи iOS, розробники можуть створювати потужні та ефективні клієнтські застосунки, що надають широкий спектр функціональності для доступу та взаємодії з музичним контентом.

Операційна система iOS надає широкі можливості для реалізації аудіовідтворення в мобільних застосунках. Використання фреймворків, таких як AVFoundation і MediaPlayer, дозволяє розробникам створювати потужні та функціональні музичні застосунки для добирання музичного контенту. Крім того, системні функції та API допомагають забезпечити розширені можливості та покращену взаємодію з користувачем. Використання цих можливостей дозволяє створювати застосунки, які забезпечують зручне та якісне аудіовідтворення на пристроях, що працюють під управлінням операційної системи iOS.

Varog є потужним фреймворком для побудови серверної частини застосунків добирання музичного контенту. Він надає простий та зрозумілий синтаксис мови Swift разом з високошвидкісною обробкою мережесих запитів за допомогою SwiftNIO. Завдяки своїй гнучкості та можливостям масштабування, Varog стає відмінним вибором для розробки серверної частини, яка забезпечує обробку запитів та взаємодію з музичним контентом у швидкому та ефективному способі.

## РОЗДІЛ 3. ПРОЄКТУВАННЯ, РОЗРОБЛЕННЯ ТА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДОБИРАННЯ МУЗИЧНОГО КОНТЕНТУ

### 3.1 Інженерія вимог до застосунку добирання музичного контенту

Інженерія вимог є ключовим етапом у процесі розробки застосунку добирання музичного контенту. Цей етап передбачає визначення та документування функціональних та нефункціональних вимог, які визначають основні характеристики та властивості застосунку. У цьому підрозділі розглянемо процес інженерії вимог до застосунку добирання музичного контенту та ключові аспекти, що потрібно враховувати для забезпечення високого рівня користувацького досвіду, достатнього переліку програмних опцій та можливостей; також, в цьому розділі виконано аналіз функціоналу систем добирання музичного контенту, що вже представлені на ринку та мають базу клієнтів.

#### 3.1.1 Постановка задачі

Ціллю застосунку добирання музичного контенту є створення зручного та персоналізованого середовища, яке дозволить користувачам знаходити, прослуховувати, добирати, зберігати, сортувати та насолоджуватися різноманітним музичним контентом. Основними характеристиками цієї мети є забезпечення широкого вибору музичних треків, зручності пошуку, індивідуалізованого досвіду користувача та забезпечення безпеки даних користувача.

#### 3.1.2 Аналіз функціоналу систем добирання музичного контенту

Нижче наведено аналіз програмних рішень платформ добирання музичного контенту. Платформи, що порівнюються, включають Spotify, Apple Music, Amazon Music, YouTube Music, Tidal, Deezer, SoundCloud, Pandora, iHeartRadio та JioSaavn. Ці платформи порівнюються за такими характеристиками: наявність алгоритмів рекомендацій, якість користувацького інтерфейсу, доступність та величина каталогу, якість звуку, що пропонується, та наявністю чи відсутністю мобільного застосунку.

Таблиця 3.1 – Порівняльна характеристика СДМК

Платформа	Алгоритми рекомендацій	Інтерфейс та зручність	Доступність та каталог	Якість звуку	Мобільний застосунок
Spotify	Так	Зручний інтерфейс	Великий каталог	Висока	Так
Apple Music	Так	Привабливий інтерфейс	Широкий каталог	Висока	Так
Amazon Music	Так	Простий інтерфейс	Широкий каталог	Висока	Так
YouTube Music	Так	Інтуїтивний інтерфейс	Широкий каталог	Висока	Так
Tidal	Так	Стильний та сучасний інтерфейс	Великий каталог	Висока	Так
Deezer	Так	Приємний і простий інтерфейс	Широкий каталог	Висока	Так
SoundCloud	Ні	Креативний та експериментальний інтерфейс	Великий каталог	Середня	Так
Pandora	Так	Зручний та інтуїтивний інтерфейс	Різноманітний каталог	Висока	Так
iHeartRadio	Ні	Простий та зрозумілий інтерфейс	Широкий каталог	Середня	Так
JioSaavn	Так	Естетичний та функціональний інтерфейс	Широкий каталог	Висока	Так

### 3.1.3 Функціональні можливості застосунку, що проєктується

На основі аналізу потреб користувачів, інженери визначають функціональні вимоги до застосунку добирання музичного контенту. Ці вимоги описують функції та можливості, які повинен надавати застосунок.

Опишемо такі вимоги:

- реєстрація, авторизація та аутентифікація як обов'язкова вимога до користувача для взаємодії з застосунком;
- пошук музичного контенту за критеріями, такими як назва артиста та назва пісні;
- відтворення музичних треків та радіостанцій у режимі онлайн;
- створення переліка улюблених треків та його зміна;
- режим підбору нових артистів та композицій;
- реалізація права на забуття, видалення облікового запису користувача та інших даних, що асоціюються з ним.

### 3.1.4 Нефункціональні можливості застосунку, що проєктується

Крім функціональних вимог, інженери також визначають нефункціональні вимоги до застосунку добирання музичного контенту. Ці вимоги визначають якісні аспекти та обмеження системи. Опишемо такі нефункціональні вимоги:

- швидкодія: система повинна забезпечувати швидку відповідь на запити користувачів та мінімальні часи завантаження музичних треків;
- безпека: застосунок повинен забезпечувати захист персональних даних користувачів та не зберігати важливі дані у незашифрованому форматі;
- надійність: система повинна бути стійкою до помилок та забезпечувати неперервну доступність музичного контенту.
- сумісність: застосунок повинен бути сумісним з різними версіями ОС iOS.

## 3.2 Архітектурні рішення застосунку добирання музичного контенту

### 3.2.1 Архітектура клієнтської частини застосунку

Архітектура клієнтської частини iOS застосунку для добирання музичного контенту реалізовуватиметься за допомогою підходу Model-View-Controller (MVC). MVC – це архітектурний шаблон, що дозволяє розділити компоненти застосунку на три основні частини з різними відповідальностями: модель, вид та контролер[46].

Модель (Model): модель відповідає за представлення даних та бізнес-логіки у застосунку[47]. У випадку застосунку добирання музичного контенту модель може містити дані про музичні треки, плейлисти, вподобання користувача та іншу інформацію, що пов'язана з музичним контентом. Модель також відповідає за операції збереження, оновлення та отримання даних, а також за виконання бізнес-логіки, такої як обробка пошуку, сортування та фільтрації музичного контенту.

Представлення (або вид) (View): вид відповідає за представлення даних моделі та інтерфейс користувача[48]. У контексті застосунку добирання музичного контенту вид може включати графічний інтерфейс користувача, списки музичних треків, плейлисти, кнопки відтворення та інші елементи, які дозволяють користувачу взаємодіяти з музичним контентом. Вид отримує дані від моделі та відображає їх у відповідний спосіб для користувача. Він також може реагувати на взаємодію користувача та передавати події контролеру для подальшої обробки.

Контролер (Controller): контролер відповідає за керування взаємодією між моделлю та представленням[49]. Він отримує події та ініціює відповідні дії у моделі, такі як пошук музичного контенту, оновлення плейлистів або вподобань користувача. Контролер також приймає дані від моделі та передає їх виду для відображення користувачу. Він відповідає за обробку взаємодії користувача, такої як натискання кнопок, переміщення по списку треків тощо.

### 3.2.2 Архітектура клієнтської частини застосунку

Архітектура серверної частини iOS застосунку для добирання музичного контенту з використанням фреймворку Vapor реалізовуватиметься за допомогою підходу, що базується на концепції моделей, контролерів та маршрутизації. Vapor, як було розглянуто раніше, – це фреймворк для серверного розробки, який побудований на основі мови програмування Swift та використовується для створення потужних та масштабованих серверних застосунків.

Модель відповідає за представлення даних та бізнес-логіки серверної частини застосунку. У випадку застосунку добирання музичного контенту модель може містити сутності, такі як користувачі, музичні треки, плейлисти тощо. Вона відповідає за зберігання та операції над цими даними, такі як створення, зчитування, оновлення та видалення. Модель також включає логіку бізнес-правил, перевірку даних та зв'язки між різними сутностями.

Контролер відповідає за обробку запитів та взаємодію з моделлю у відповідь на ці запити. Використовуючи фреймворк Vapor, контролери реалізовані як класи, які включають методи для обробки різних маршрутів. Контролери отримують HTTP-запити від клієнтів, виконують відповідні дії, які включають зчитування або запис даних, та повертають HTTP-відповідь з результатом обробки запиту.

Маршрутизація визначає зв'язок між URL-шляхами та відповідними контролерами для обробки запитів. За допомогою фреймворку Vapor, використовуються різні маршрути для різних дій, таких як створення ресурсу, отримання даних, оновлення або видалення. Маршрути вказуються за допомогою HTTP-методу (GET, POST, PUT, DELETE) та URL-шляху, який співпадає з конкретним контролером та методом обробки.

### 3.2.3 Використання спільного коду в клієнтській та серверній частинах застосунку

Вибір мови програмування Swift як базису для створення і клієнтської, і серверної частини застосунку надає деякі важливі переваги, які у традиційних підходах розробки веб-сайтів та застосунків не є доступними.

Питання використання спільного коду між клієнтською та серверною частинами застосунку, таким чином, може бути реалізоване за допомогою SPM. Доречність створення такого спільного модуля між обома частинами системи полягає в тому, що ті сутності, що відправляються з сервера на клієнт і навпаки, таким чином можна стандартизувати, значно знижуючи потенційні проблеми при кодуванні/декодуванні JSON файлу.

Тому, створено Swift Package з назвою `melodyYaShared`, і в нього покладено код, що описує сутності, що надсилаються, або DTO. Також, цей пакет додано до проєкту клієнтської та серверної частин застосунку, щоб отримати змогу користуватися визначеними в цьому пакеті сутностями.

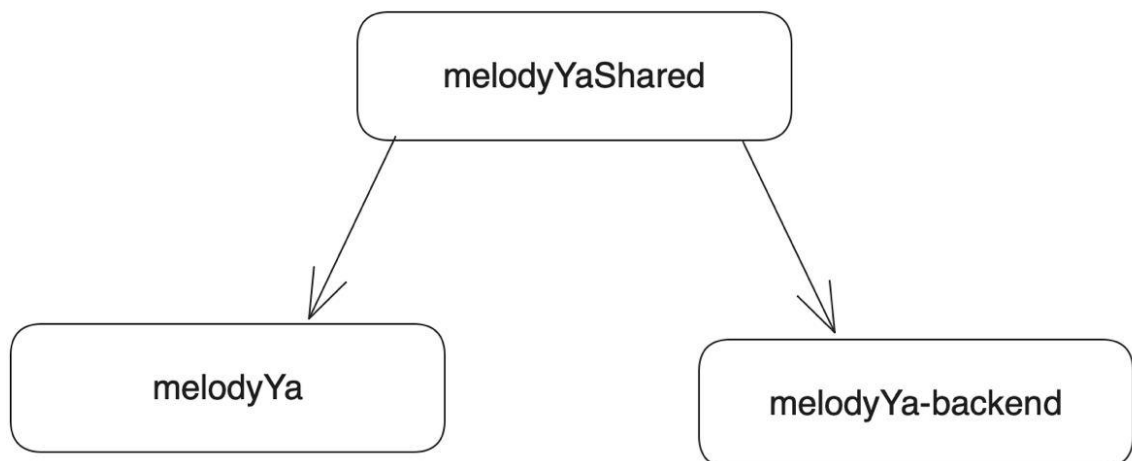


Рисунок 3.1 – Схема взаємодії системи зі спільним кодом

### 3.3 Проєктування та реалізація застосунку добирання музичного контенту

#### 3.3.1 Структура бази даних

На серверній стороні застосунку наявні такі сутності: `User`, `Station`, `Token`, `AllStation`. Приватну частину гібридної бази даних можна зобразити такою схемою:

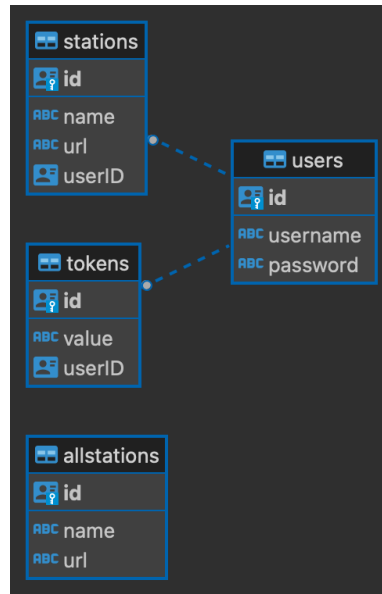


Рисунок 3.2 – Приватна частина гібридної хмари

Опис відношень є таким:

- користувач (User): має поле "id", "username", та "password". Користувач може мати кілька станцій (Station) та токенів (Token);
- станція (Station): має поля "id", "name", та "url". Кожна станція належить одному користувачу;
- всі станції (AllStation): має поля "id", "name", та "url";
- токен (Token): має поля "id" та "value". Кожен токен належить одному користувачу.

### 3.3.2 Структура користувацького інтерфейсу

Головним компонентом інтерфейсу є `YaTabBarController`, що є підкласом `UITabBarController`. `UITabBarController` є одним із основних контейнерних контролерів, який надає можливість відображати та керувати множиною підконтролерів за допомогою вкладок[50]. Він забезпечує інтерфейс, в якому користувач може переключатись між різними вкладками, представляючи різні функціональні частини додатку.

Підконтролерами даного контролера є: `MusicController`, `RadioController`, `FavoriteViewController` і `AccountController`. Також даний підклас реалізує можливість підлаштовувати параметри `safeArea` як спосіб

запобігання перекриття вмісту контролерів плеєром відтворення; вона визначає область екрану, яка доступна для розміщення важливого контенту та інтерфейсних елементів без перекриття системних компонентів, таких як панель стану, панель навігації, панель вибору вкладок тощо[51]. Це забезпечує візуальну чистоту та коректну роботу інтерфейсу незалежно від розмірів екрану.

Кожен з підконтролерів мають свої компоненти представлення, такі як `CollectionView`, `Cell`, `Buttons` тощо. Деякі з підконтролерів, як наприклад `MusicController`, додатково міститься у контролері `UINavigationController`[52], який надає змогу робити вкладення контролерів у вигляді ієрархії.

### 3.3.3 Захист персональних даних користувачів

Захист персональних даних користувачів виконується як на стороні сервера, завдяки використанню хешування паролів з «сіллю», так і на стороні клієнтського застосунку [37]. Після авторизації, мобільний застосунок зберігає в `Keychain` [53] токен авторизації. Зберігання токена відбувається за допомогою одного з методів-обгортки для роботи з `Keychain` [54].

### 3.3.4 Тестування програмної системи

Тестування системи проведено методами мануального тестування [55]. Проведено вдалі спроби створення нового облікового запису, авторизації зі створеними параметрами, фільтрація виконавців та треків за допомогою пошукового рядка, додавання треків та радіостанцій до улюблених, видалення їх звідти, видалення облікового запису користувача, та вихід з облікового запису. За результатами тестування неочікуваної поведінки застосунку не виявлено.

### Висновок до третього розділу

Таким чином, в результаті виконання третього розділу кваліфікаційної роботи були здійснені такі кроки: планування, проектування, імплементація

та тестування застосунку і системи загалом. Основним результатом цих дій є створений застосунок добирання музичного контенту з використанням хмарних технологій.

## ВИСНОВОК

У ході виконання бакалаврської кваліфікаційної роботи було спроектовано та розроблено застосунок добирання музичного контенту, а також виконано такі завдання:

- досліджено сучасні методології розроблення застосунків;
- проаналізовано архітектурні рішення, технології та інструменти для реалізації застосунку;
- спроектовано, впроваджено та поєднано клієнтську та серверну частини мобільного застосунку для добирання музичного контенту.

Були проаналізовані вже існуючі реалізації таких застосунків, а також були виділені їх переваги та недоліки.

Для проектування та розробки застосунку були використані технології на мові програмування Swift, до яких входять:

- UIKit, що є засобом створення користувацького інтерфейсу;
- Varog, який було використано як спосіб створення серверної частини застосунку;
- Swift Package Manager, що використовується в даному випадку як спосіб додавати спільні кодові бази до декількох проєктів, а саме, клієнтської та серверної частин застосунку.

В результаті був розроблений застосунок для музичного контенту, який забезпечує швидкий та зручний доступ до музичного контенту.

Головними особливостями застосунку є підвищений рівень доступності та надійності системи завдяки використанню гібридної хмари в якості серверної частини застосунку, а також практичний приклад поширення спільного коду між клієнтською та серверною частинами застосунку для оптимізації загальної кодової бази та поліпшення стабільності роботи системи загалом.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Share of time spent listening to music on selected devices worldwide as of May 2019 *Statista* : веб-сайт. URL: <https://www.statista.com/statistics/1102356/music-device-usage-worldwide/> (дата звернення: 17.02.2023)
2. Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. New Jersey, 2008. 464 с.
3. Patrik Wikström. *The Music Industry: Music in the Cloud*. Cambridge, 2020. 224 с.
4. Steve Knopper. *Appetite for Self-Destruction: The Spectacular Crash of the Record Industry in the Digital Age*. New York, 2009. 288 с.
5. David Byrne. *How Music Works*. New York, 2012. 384 с.
6. Larry Lessig. *Remix: Making Art and Commerce Thrive in the Hybrid Economy*. New York, 2008. 352 с.
7. Ari Herstand. *How to Make It in the New Music Business: Practical Tips on Building a Loyal Following and Making a Living as a Musician*. New York, 2017. 528 с.
8. Maria Eriksson, Rasmus Fleischer. *Spotify Teardown: Inside the Black Box of Streaming Music*. Cambridge, 2019. 288 с.
9. K John Seabrook. *The Song Machine: Inside the Hit Factory*. New York, 2016. 352 с.
10. Anil Prasad. *Innerviews: Music Without Borders*. Austin, 2010. 400 с.
11. Peter Tschmuck. *Creativity and Innovation in the Music Industry*. Berlin, 2012. 342 с.
12. Chris Anderson. *Free: The Future of a Radical Price*. New York, 2009. 288 с.
13. Greg Kot. *Ripped: How the Wired Generation Revolutionized Music*. New York, 2009. 288 с.
14. Simon Napier-Bell. *Ta-Ra-Ra-Boom-De-Ay: The Dodgy Business of Popular Music*. Ebury Press, 2014. 400 с.

15. Paul Allen. *Artist Management for the Music Business*. New York, 2011. 432 с.
16. David Baskerville, Tim Baskerville. *Music Business Handbook and Career Guide*. Thousand Oaks, 2015. 640 с.
17. Timothy J. Cooley. *Surfing about Music*. Chicago, 2014. 336 с.
18. IFPI Global Music Report 2020 *International Federation of the Phonographic Industry* : веб-сайт. URL: [https://www.ifpi.org/wp-content/uploads/2020/03/GMR2021\\_STATE\\_OF\\_THE\\_INDUSTRY.pdf](https://www.ifpi.org/wp-content/uploads/2020/03/GMR2021_STATE_OF_THE_INDUSTRY.pdf) (дата звернення: 17.02.2023)
19. WIPO Copyright Treaty (WCT) *World Intellectual Property Organization* : веб-сайт. URL: <https://www.wipo.int/treaties/en/ip/wct/> дата звернення: 17.02.2023)
20. Agreement on Trade-Related Aspects of Intellectual Property Rights *The World Trade Organization* : веб-сайт. URL: [https://www.wto.org/english/docs\\_e/legal\\_e/27-trips\\_01\\_e.htm](https://www.wto.org/english/docs_e/legal_e/27-trips_01_e.htm) (дата звернення: 17.02.2023)
21. Rome Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organizations *World Intellectual Property Organization* : веб-сайт. URL: <https://www.wipo.int/treaties/en/ip/rome/> дата звернення: 17.02.2023)
22. Digital Single Market and amending Directives *EUR-Lex* : веб-сайт. URL: <https://eur-lex.europa.eu/eli/dir/2019/790/oj> (дата звернення: 17.02.2023)
23. The General Agreement on Trade in Services *The World Trade Organization* : веб-сайт. URL: [https://www.wto.org/english/tratop\\_e/serv\\_e/gatsqa\\_e.htm](https://www.wto.org/english/tratop_e/serv_e/gatsqa_e.htm) (дата звернення: 17.02.2023)
24. General Data Protection Regulation *GDPR* : веб-сайт. URL: <https://gdpr-info.eu/> (дата звернення: 17.02.2023)
25. Про авторське право і суміжні права *Верховна Рада України* : веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/3792-12#Text> (дата звернення: 17.02.2023)

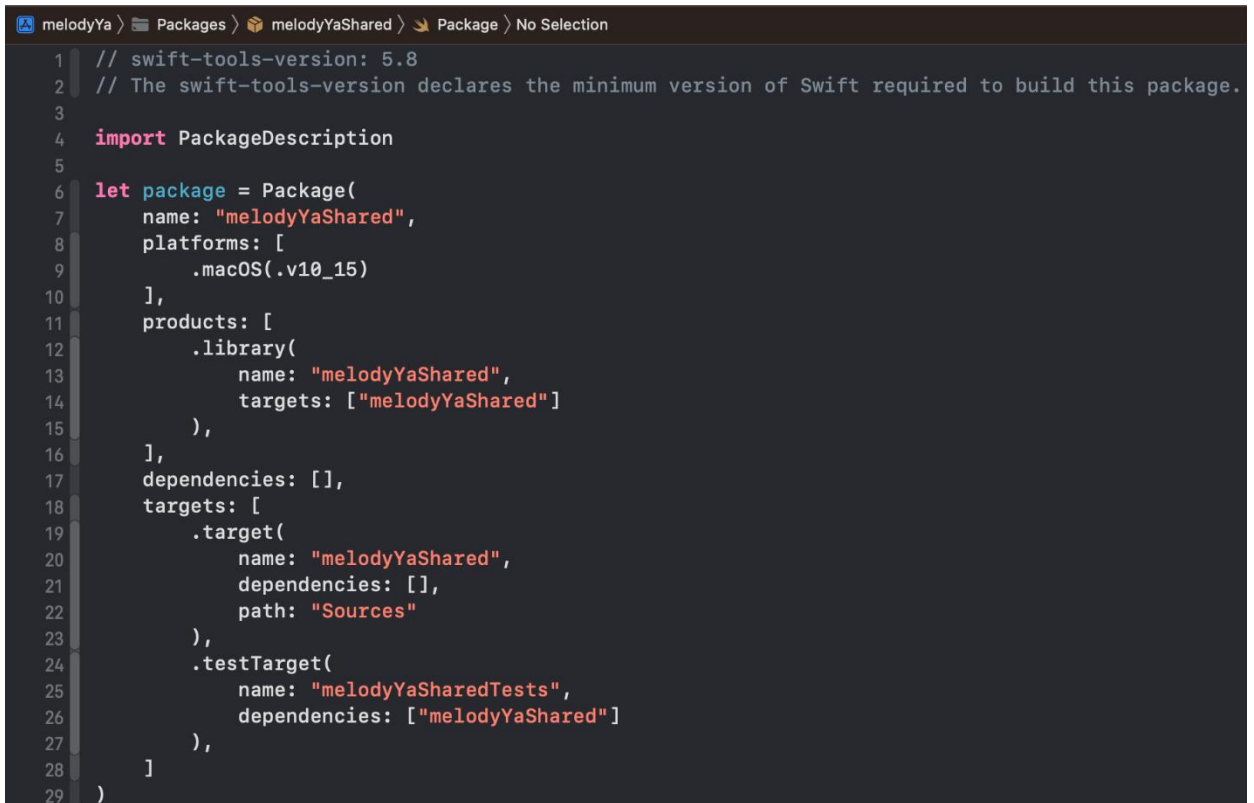
26. Про інформацію *Верховна Рада України* : веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/2657-12#Text> (дата звернення: 17.02.2023)
27. Про захист персональних даних *Верховна Рада України* : веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text> (дата звернення: 17.02.2023)
28. Public vs Private vs Hybrid Cloud: Cloud Models Compared *KNOWLEDGEHUT SOLUTIONS* : веб-сайт. URL: <https://www.knowledgehut.com/blog/cloud-computing/public-private-hybrid-cloud> (дата звернення: 17.02.2023)
29. Public vs Private vs Hybrid: Cloud Differences Explained *BMC* : веб-сайт. URL: <https://www.bmc.com/blogs/public-private-hybrid-cloud/> (дата звернення: 17.02.2023)
30. Public Cloud vs Private Cloud vs Hybrid Cloud *MongoDB* : веб-сайт. URL: <https://www.mongodb.com/cloud-database/private-public-hybrid-cloud> (дата звернення: 17.02.2023)
31. Публічна, приватна і гібридна хмари — порівняння підходів *Denovo* : веб-сайт. URL: <https://denovo.ua/blog/publiczna-privatna-i-gibridna-hmari-porivnyannya-pidhodiv-20> (дата звернення: 17.02.2023)
32. Public vs. private vs. hybrid cloud: Key differences defined *Techtarget* : веб-сайт. URL: <https://www.techtarget.com/searchcloudcomputing/feature/Public-cloud-vs-private-cloud-Key-benefits-and-differences> (дата звернення: 17.02.2023)
33. What is a Container? *Docker* : веб-сайт. URL: <https://www.docker.com/resources/what-container/> (дата звернення: 17.02.2023)
34. Overview | Kubernetes *Kubernetes* : веб-сайт. URL: <https://kubernetes.io/docs/concepts/overview/> (дата звернення: 17.02.2023)
35. What is container-native virtualization? *Red Hat* : веб-сайт. URL: <https://www.redhat.com/en/topics/containers/what-is-container-native-virtualization> (дата звернення: 17.02.2023)

36. What is a Virtual Machine? | VMware Glossary *VMWare* : веб-сайт. URL: <https://www.vmware.com/topics/glossary/content/virtual-machine.html> (дата звернення: 17.02.2023)
37. Грицунь Я.Б. Захист інформації в прикладних інформаційних системах. *Прикладні системи та технології в інформаційному суспільстві*: зб. тез доповідей і наук. повідомл. учасників VI Міжнародної науково-практичної конференції. Київ, 2022. С. 118–123.
38. Jon Manning, Paris Buttfield-Addison, Tim Nugent. *Swift Development with Cocoa*. Raleigh, 2014. 460 с.
39. Ray Wenderlich. *Swift Apprentice*. Boston, 2018. 550 с.
40. Matthew Mathias and John Gallagher. *Swift Programming: The Big Nerd Ranch Guide*. Atlanta, 2021. 480 с.
41. Chris Eidhof, Florian Kugler, and Wouter Swierstra. *Functional Swift: Updated for Swift 5.0*. Swiftbysundell, 2020. 350 с.
42. Erica Sadun. *The Swift Developer's Cookbook*. Addison-Wesley Professional, 2018. 400 с.
43. Matt Neuburg. *iOS 14 Programming Fundamentals with Swift*. O'Reilly Media, 2020. 800 с.
44. Chris Eidhof, Airspeed Velocity, Florian Kugler. *Advanced Swift: Updated for Swift 5.3*. Swiftbysundell, 2021. 350 с.
45. Keith Harrison. *Modern Auto Layout*. Useyourloaf, 2021. 503 с.
46. Renan Dias, Aaqib Hussain, Josh Steele. *Real-World iOS by Tutorials*. Raywenderlich, 2022. 349 с.
47. Jon Hoffman. *Swift Protocol-Oriented Programming: Increase productivity and build faster applications with Swift 5*. Packt Publishing, 2019. 224 с.
48. Andrew S. Tanenbaum and Maarten Van Steen. *Distributed Systems: Principles and Paradigms*. Pearson, 2016. 704 с.
49. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. McGraw-Hill Education, 2019. 1376 с.

50. Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. Database Systems: The Complete Book. Pearson, 2013. 1272 с.
51. Neil Smyth. iOS 14 Programming Fundamentals with Swift: Swift, Xcode, and Cocoa Basics. CreateSpace Independent Publishing Platform, 2020. 700 с.
52. BJ Miller. Sams Teach Yourself Swift™ in 24 Hours. Indianapolis, 2015. 421 с.
53. Saiko V., Hrytsun Ya. Data Protection in the iOS Operating System. *Інформаційні технології та взаємодії (сателітна)*: матеріали конференції. Київ, 2022. С. 134–135;
54. Грицунь Я.Б. Оптимізація використання криптографічного фреймворку Keychain: розробка зручного прикладного програмного інтерфейсу для зберігання конфіденційних даних в застосунках для операційної системи iOS. *Проблеми кібербезпеки інформаційно-телекомунікаційних систем: Збірник матеріалів доповідей та тез*. Київ, 2023. (в друці).
55. Software Testing | Manual Testing *GeeksforGeeks*: веб-сайт. URL: <https://www.geeksforgeeks.org/software-testing-manual-testing/> (дата звернення: 17.02.2023)

## ДОДАТКИ

## ДОДАТОК А

A screenshot of a code editor showing a Swift Package Manifest file. The file is named 'Package.swift' and is located in the directory 'melodyYaShared'. The code defines a package named 'melodyYaShared' for macOS (version 10.15). It includes a library target named 'melodyYaShared' and a test target named 'melodyYaSharedTests'. The package has no dependencies.

```
1 // swift-tools-version: 5.8
2 // The swift-tools-version declares the minimum version of Swift required to build this package.
3
4 import PackageDescription
5
6 let package = Package(
7     name: "melodyYaShared",
8     platforms: [
9         .macOS(.v10_15)
10    ],
11    products: [
12        .library(
13            name: "melodyYaShared",
14            targets: ["melodyYaShared"]
15        ),
16    ],
17    dependencies: [],
18    targets: [
19        .target(
20            name: "melodyYaShared",
21            dependencies: [],
22            path: "Sources"
23        ),
24        .testTarget(
25            name: "melodyYaSharedTests",
26            dependencies: ["melodyYaShared"]
27        ),
28    ]
29 )
```

Рисунок 1А – Структура пакета спільного для клієнтської та серверної частин застосунку коду

```
melodyYa-backend > Package > No Selection
1 // swift-tools-version:5.2
2
3 import PackageDescription
4
5 let package = Package(
6     name: "melodyYa",
7     platforms: [
8         .macOS(.v10_15)
9     ],
10    dependencies: [
11        .package(url: "https://github.com/vapor/vapor.git", from: "4.50.0"),
12        .package(url: "https://github.com/vapor/fluent.git", from: "4.3.0"),
13        .package(url: "https://github.com/vapor/fluent-postgres-driver.git", from: "2.2.0"),
14        .package(url: "https://github.com/vapor/leaf.git", from: "4.1.0"),
15        .package(url: "https://github.com/gr-yarik/melodyYaShared", from: "1.0.0"),
16    ],
17    targets: [
18        .target(
19            name: "App",
20            dependencies: [
21                .product(name: "Fluent", package: "fluent"),
22                .product(name: "Vapor", package: "vapor"),
23                .product(name: "Leaf", package: "leaf"),
24                .product(name: "FluentPostgresDriver", package: "fluent-postgres-driver"),
25                .product(name: "melodyYaShared", package: "melodyYaShared")
26            ],
27            swiftSettings: [
28                .unsafeFlags(["-cross-module-optimization"], .when(configuration: .release))
29            ]
30        ),
31        .target(
32            name: "Run",
33            dependencies: [
34                .target(name: "App"),
35            ]
36        ),
37        .testTarget(
38            name: "AppTests",
39            dependencies: [
40                .target(name: "App"),
41                .product(name: "XCTVapor", package: "vapor"),
42            ]
43        )
44    ]
45 )
```

Рисунок 1Б – Структура пакета, що описує серверну частину застосунку  
(приватну хмару)