

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ
«МУЗИЧНИЙ ПОРТАЛ»

(тема)

Виконав:

студентка 4 курсу, групи ІТІНФ-18-2
Кирилова В.А
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Кобилін О.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри Кобилін О.А.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Кириловій Вікторії Андріївні
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення мобільного застосунку «Музичний портал»

затверджена наказом університету від 16 травня 2022 року No 541Ст

2. Термін подання студентом роботи до екзаменаційної комісії 29 травня 2022 р.

3. Вихідні дані до роботи матеріали про розробку мобільних застосунків, інтернет-ресурси, результати досліджень розробки алгоритмів для штучного інтелекту, результати обробки текстової інформації за допомогою аналізу даних, теоретичні відомості про створення рекомендацій на основі існуючих даних

4. Перелік питань, що потрібно опрацювати в роботі

1. Огляд технологій для розробки інформаційної системи.2. Аналіз існуючих методів обробки даних.3. База даних для застосунку.4. Модель інформаційної системи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Схематичні зображення бази даних, візуалізація прототипу застосунку, схематичне зображення взаємодії користувачів застосунку, діаграма Чена.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Белова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.04.2022	
2	Аналіз завдання, підбір літератури	18.04.22-21.04.22	
3	Аналіз літератури з досліджуваної проблеми	22.04.22-25.04.22	
4	Аналіз технологій розробки вебсайтів	26.04.22-30.04.22	
5	Вибір підходу у створенні вебсайту	01.05.22-14.05.22	
6	Реалізація вебсайту	15.05.22-23.05.22	
7	Оформлення пояснювальної записки	24.05.22-26.05.22	
8	Перевірка на плагіат	27.05.22	
9	Рецензування	28.05.22	
10	Підготовка презентації та доповіді	29.05.22-30.05.22	
11	Занесення роботи в електронний архів	31.05.22	
12	Попередній захист кваліфікаційної роботи	01.06.22	

Дата видачі завдання 18 квітня 2022 р.

Студент _____

(підпис)

Керівник роботи _____ доц. Кобилін О.А.

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 46с., 14 рис., 10 табл., 30 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, БАЗИ ДАНИХ, РОЗРОБКА ПРОТОТИПУ, МОБІЛЬНИЙ ЗАСТОСУНОК.

Об'єктом роботи є музичні твори (пісні) та переваги у їх виборі серед слухачів.

Метою роботи є розробка інформаційної системи, яка за допомогою аналізу отриманих даних буде формувати рекомендації для прослуховування та реалізація ІС у вигляді прототипу для мобільного застосунку.

Використано принципи проектування та нормалізації баз даних. Проведено аналіз способів підбору рекомендацій на основі отриманої інформації. Досліджено принципи створення ІС для оптимального проектування застосунку.

У результаті роботи розроблено прототип програмної реалізації системи у вигляді мобільного застосунку.

INFORMATION SYSTEM, DATABASES, PROTOTYPE DEVELOPMENT, MOBILE APPLICATION.

The object of the work is musical works (songs) and the advantages in their choice among the listeners.

The aim of the work is to develop an information system that will use the analysis of the obtained data to form recommendations for listening and implementing IP in the form of a mobile application.

The principles of database design and normalization are used. The analysis of ways of selection of recommendations on the basis of the received information is carried out. The principles of creating IP for optimal application design are studied.

As a result, a prototype of the software implementation of the system in the form of a mobile application was developed.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ	8
1 Огляд технологій для розробки інформаційної системи.....	10
1.1 Принцип роботи додатку для потокової передачі музики.....	10
1.1.1 Ліцензійні вимоги для програми потокової передачі музики.....	11
1.2 Виявлення типу музичної програми	12
1.2.1 Музичні бібліотеки	12
1.2.2 Хмарне сховище.....	12
1.2.3 Радіостанції	13
1.3 Етап UI/UX дизайну.....	13
1.4 Поняття бази даних.....	14
1.5 Призначення та класифікація систем управління базами даних.....	15
1.6 Можливості мови SQL.....	16
1.7 Постановка задачі.....	17
2 Розробка моделі та бази даних для застосунку «музичний портал».....	19
2.1 Загальні вимоги.....	19
2.2 Розробка бізнес-правил	19
2.3 Розробка концептуальної моделі	20
2.4 Функціональна структура	22
2.5 Побудова логічної моделі даних.....	24
2.6 Побудова моделі даних за допомогою ERWIN.....	29
2.7 Фізичне проектування	30
3 Реалізація застосунку для інформаційної системи	34
3.1 Реєстрація та вхід.....	34
3.2 Профіль користувача	35
3.2.1 Інтерфейс профілю слухача.....	35
3.2.2 Інтерфейс профілю виконавця.....	37
3.3 Профіль технічної підтримки	38
3.4 Пошук	39
3.5 Музичний програвач.....	39
3.6 Алгоритм генерування рекомендацій.....	40

	6
3.7 Персоналізовані рекомендації.....	41
3.8 Технічна реалізація.....	41
Висновки.....	45
Перелік джерел посилання.....	46

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І
ТЕРМІНІВ

ІС – інформаційна система

ПЗ – програмне забезпечення

БД – бази даних

СУБД – системи управління БД

AWS – Amazon Web Services

CDN – Content Delivery Network

ВСТУП

Сьогодні одним з найважливіших аспектів нашого життя є інформація. Людина кожного дня отримує та обробляє величезні обсяги даних.

Інформація будь-якого виду передається за допомогою повідомлень, що формуються як послідовність знаків та символів або параметрів фізичних процесів, які відображаються на матеріальних носіях: папері, лініях зв'язку, магнітних та оптичних носіях тощо.

Музика є досить неочевидним джерелом інформації. На відміну від книжок або статей, музика являється окремим видом мистецтва, який у сьогоденні люди навчилися використовувати як ресурс для навчання. Зокрема, музика передає та відтворює настрій людини, що поширює її (музики) обсяг використання.

Будь-яка інформація має бути систематизованою для правильного її сприйняття. Для цього замало просто вести записи інформації. Необхідно мати змогу швидко її знаходити.

Для зберігання і полегшення пошуку потрібної інформації були вигадані різні способи. До них відносяться записні книжки, щоденники, перекидні календарі – так звані «бази даних».

База даних дозволяє швидко знаходити та обробляти інформацію. Для більшої зручності використання бази даних розробляють застосунки, які відповідають специфікації БД.

Інтеграція музики до мобільного телефону була складною. Технологія портативної музики почала розвиватися ще з 1980-х років, коли Sony керувала регіоном зі своїм програвачем. Але на той час розвиток мобільних телефонів була зосередженій на області створення та обробки зображень, що уповільнило процес.

Перший звіт про бізнес-план та необхідність успішної інтеграції музичних телефонів був написаний у 2004 році компанією Strategy Analytics: «Музичні телефони є ключовими для 3G», консалтинговою фірмою зі стільникового зв'язку з Массачусетсу. Цей звіт викликав у виробників телефонів, таких як Nokia та Motorola, потребу приєднатися до лідерів мобільного зв'язку та інтегруватися до музики, включаючи розробку стратегії музичного магазину Nokia та інтеграцію iTunes у телефон Motorola з її Rocker. Sony, Samsung та LG були надто зайняті підвищенням пікселізації та стабільності модулів камер. Таким чином Nokia покладалася на Ідею про те, що музику можна використовувати для залучення клієнтів. Rocker досяг успіху, хоча він все ще залишався прив'язаним до комп'ютера для завантаження музики. Можна сказати, що використання Rocker компанією АТТ як стратегія поглинання на ринку спонукало операторів купувати телефони з підтримкою музики, а виробників – розвивати їх.

Сьогодні більшість мобільних телефонів мають музичні можливості, які також були перенесені на смартфони.

Актуальність розробки мобільного застосунку «Музичний портал» полягає у наданні користувачам змоги знаходити та додавати у плейлисти музику з рекомендацій, які підбирає алгоритм на основі вже існуючих даних.

1 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Принцип роботи додатку для потокової передачі музики

Найпопулярніші потокові програми, такі як Spotify, Apple Music, Amazon Music та Youtube Music, зберігають свою музику на мільйонах серверів. Тому коли користувач хоче прослухати трек, мобільний застосунок «запитує» серверну частину для отримання даних.

Цей метод потокової передачі не вимагає завантаження файлів на пристрої. Натомість музика доставляється невеликими «пакетами» через Інтернет і буферизується на пристрої. Пакети залишаються невеликими, щоб уникнути затримки між подіями надсилання та отримання, тому музика відтворюється миттєво (Рис. 1.1).

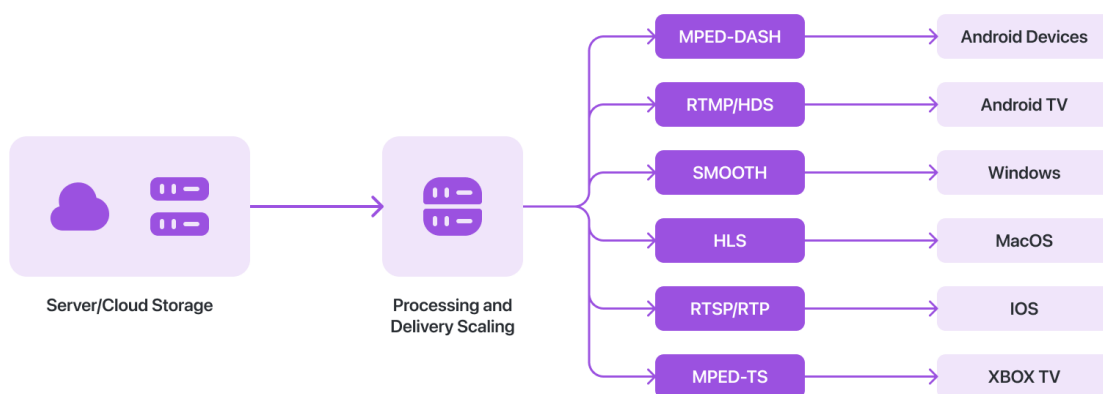


Рисунок 1.1 – Базова архітектура програми для потокової передачі музики

Архітектура сховища грає вирішальну роль потокового застосунку, оскільки вона обробляє мільйони запитів на секунду. Якщо у планах є

створення сховища даних із власними серверами, необхідно переконатися, що музичні файли записані надійно. Це дозволяє користувачам слухати музику плавно, без несподіваних спотворень чи збоїв.

Хмарні сервіси – гарний варіант для зберігання та хостингу програм для потокової передачі музики. Хмарні послуги кращі для потокової передачі в реальному часі і пропонують високу масштабованість при заощадженні грошей на покупці обладнання. Також є можливість вибрати готові хмарні рішення, такі як Amazon S3 від AWS. Наприклад, Soundcloud використовує Amazon Simple Storage Service (Amazon S3) у поєднанні з Amazon Glacier для свого рішення для зберігання.

Крім того, можна використовувати CDN для доставки контенту з низькою затримкою та безпекою. CDN можуть спричинити додаткові витрати, але вони допомагають масштабувати сервери і забезпечують бездоганну доставку.

1.1.1 Ліцензійні вимоги для програми потокової передачі музики

Якщо програма відтворює пісню без дозволу лейбла або виконавця, вона тягне за собою штраф у розмірі від 700 до 150 000 доларів за пісню щоразу, коли вона відтворюється.

По-перше, необхідно буде придбати права на публічне виконання. Це своєрідна угода між власником музичної програми та композитором або будь-якою іншою особою/агентством, що володіє авторськими правами на цей музичний твір.

По-друге, є можливість придбати ліцензію на звукозапис, яка є угодою, що охоплює використання прав на запис. Spotify, наприклад, має ліцензії, отримані від великих звукозаписуючих компаній, включаючи Sony Music

Entertainment Group, Universal Music Group та Warner Music Group, що надає йому законні права на публікацію контенту цих лейблів.

1.2 Виявлення типу музичної програми

Вибір типу музичної програми – це перший крок, щоб уточнити ідею програми. Програми для потокової передачі музики існують у різних форматах та пропонують різні варіанти прослуховування. Проте найпопулярніші музичні сервіси поділяються на три загальні категорії:

1.2.1 Музичні бібліотеки

У цю категорію потрапляють популярні програми потокової передачі музики, такі як Spotify, Pandora і Apple Music. Тут музика зберігається в серверних бібліотеках, і користувачі мають необмежений доступ до музики, але за умов власника програми [6].

1.2.2 Хмарне сховище

Ця категорія програм пропонує користувачам хмарне сховище для зберігання, організації та управління своєю музикою. Користувачі можуть транслювати музику з будь-якого місця та в будь-який час. AudioBox – популярний приклад хмарного музичного плеєра [6].

1.2.3 Радіостанції

Програми для музичних радіо дозволяють користувачам транслювати різні онлайн-радіостанції, впорядковані за певними темами, такими як жанри, плейлисти, орієнтовані на виконавців, десятиліття, настрої та багато іншого. Такі програми також фільтрують результати пошуку в залежності від ваших потреб, пропонують радіостанції, які можуть вам сподобатися, і є чудовим способом відкрити нову музику [6].

1.3 Етап UI/UX дизайну

UI/UX дизайн вашої програми є одним з найважливіших факторів, відповідальних за його успіх. Зручний, але привабливий, інтуїтивно зрозумілий та сучасний дизайн допоможе залучити та утримати ваших користувачів.

Дизайнерам потрібно мінімум 150 годин для розробки UI/UX для однієї платформи. Вона може включати такі етапи:

- wireframing, коли дизайнери створюють чорно-білий ескіз програми, щоб показати загальну концепцію дизайну програми;
- макети розробляються для демонстрації статичних, але барвистих екранів програми і включають значки, кнопки тощо;
- прототипування – це процес розробки інтерактивної моделі програми, щоб показати, як програма працює, де є можливість натиснути, провести пальцем, перетягнути і т.д., щоб побачити результати.

1.4 Поняття бази даних

База даних – сукупність даних, що організовані відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами. Зазвичай база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім самих даних, містить їх опис та може містити засоби для їх обробки.

Іншими словами, базою даних можна вважати будь-який впорядкований набір даних.

Основним призначенням бази даних є зберігання масивів даних. Але їх широко використовують і для збереження адміністративної інформації та спеціалізованих даних, наприклад, для інженерних даних чи для економічних моделей. Прикладами використання баз даних можуть бути:

- автоматизовані системи обліку;
- реєстри та каталоги;
- геоінформаційні системи;
- лінгвістичні бази даних, тобто машинні словники різного типу і призначення;
- бази даних транспортних систем;
- системи керування вмістом Інтернет-сайтів, які зберігають у базах даних інформацію про web-сторінки сайту.

База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні «файл-програма», вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, як у випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СУБД).

1.5 Призначення та класифікація систем управління базами даних

Системи управління базами даних (СУБД) – це складна програмна система накопичення інформації з можливістю подальшого управління даними, що представляють інтерес для користувача. Кожній прикладній програмі СУБД надає інтерфейс з базою даних та має засоби безпосереднього доступу до неї. Таким чином, СУБД відіграє центральну роль в функціонуванні автоматизованого банку даних.

Архітектурно СУБД складається з двох великих компонент. За допомогою мови опису даних (МОД) створюються описи елементів, груп та записів даних, а також взаємозв'язки між ними, які, як правило, задаються у вигляді таблиць. В залежності від конкретної реалізації СУБД мову опису даних підрозділяють на мову опису схеми бази даних (МОС) та мову опису підсхем бази даних (МОП). Слід особливо зазначити, що МОД дозволяє створити не саму базу даних, а лише її опис.

1.6 Можливості мови SQL

В сучасних реляційних СУБД таких як Oracle, Informix, Sybase, DB2, dBASE, Paradox тощо використовується мова структурованих запитів SQL, що є досить потужною та популярною мовою маніпулювання даними реляційних СУБД. Після внесення ряду доповнень в стандарт, мова SQL стала поєднувати в собі можливості мови визначення даних, мови маніпулювання даними та мови обробки запитів[24].

SQL-технологія реалізує наступні основні функції реляційних СУБД:

- визначення даних, тобто SQL дозволяє визначити структуру підтримуваних даних і організацію використовуваних реляційних відношень;
- доступ до даних – SQL забезпечує санкціонований доступ до даних клієнту;
- управління доступом – SQL забезпечує синхронізацію обробки бази даних різними прикладними програмами, захистом доступу від несанкціонованого доступу;
- розділення даних – SQL координує роботу клієнтів, що одночасно виконують операції з БД;
- забезпечення цілісності даних, тобто використання SQL може забезпечити досить складні обмеження, задоволення яким буде перевірятися у випадках модифікації БД;
- маніпулювання даними – SQL забезпечує клієнту можливість міняти дані в базі даних шляхом введення нової інформації, видалення старої та модифікації існуючої.

1.7 Постановка задачі

Об'єктом роботи є музичні твори (пісні) та переваги у їх виборі серед слухачів.

Метою роботи є розробка інформаційної системи, яка за допомогою аналізу отриманих даних буде формувати рекомендації для прослуховування та реалізація ІС у вигляді прототипу для мобільного застосунку.

Нижче наведені опис даної предметної області, вимоги користувача до даних і функціоналу інформаційної системи.

Програма повинна:

- виводити таблиці у зв'язку з рівнем доступу;
- мати пошук за назвою пісні/ім'ям автора;
- мати інформацію про пісню;
- надавати можливість створювати власні диски;
- надавати плейлисти з рекомендаціями пісень на основі вже створених.

Спочатку кожен користувач має авторизуватися. Після чого він, у зв'язку зі своїм рівнем доступу отримує певну інформацію щодо порталу.

Щодо ролі адміна, то він має права:

- додавати інформацію до таблиць(альбом, пісня, автор, користувач, жанр);
- редагувати їх;
- видаляти їх;
- дивитися інформацію, додану користувачами.
- користувач може:
 - додати свій власний диск;
 - редагувати або видалити його;
 - здійснювати пошук за автором або піснею;
 - дивитися інформацію про альбоми, авторів, пісні.

Для досягнення мети кваліфікаційної роботи необхідно вирішити наступні задачі:

- вивчити такі теоретичні питання: теорію проектування реляційних БД на основі побудови ER-моделі, основні принципи нормалізації БД, принципи побудови запитів до БД;

- ознайомитися з наступними інструментами проектування і програмування ІС: CASE-засобом візуального проектування даних ERwin, Intelij IDEA;

- застосувати отримані теоретичні та практичні знання для розробки ІС для ПО «Музичний портал»;

- вивчивши опис ПО і вимоги до функціональності ІС, розробити бізнес-правила;

- на основі бізнес-правил розробити ER-модель і відобразити її за допомогою ER-діаграми в синтаксисі Чена;

- розробити модель даних за допомогою CASE-засобу візуального проектування даних ERwin;

- сформувати структуру БД в СУБД;

- розробити програмний код для забезпечення необхідної функціональності ІС;

- відобразити процес і результати розробки ІС в пояснювальній записці до кваліфікаційної роботи.

Для розробки ІС необхідно використовувати такі інструменти:

- ERwin – CASE-засіб проектування БД;

- MS SQL Server – СУБД;

- IntelliJ IDEA, Java – середа і мова розробки програмного забезпечення.

2 РОЗРОБКА МОДЕЛІ ТА БАЗИ ДАНИХ ДЛЯ ЗАСТОСУНКУ «МУЗИЧНИЙ ПОРТАЛ»

2.1 Загальні вимоги

Інформаційна система повинна бути розроблена на мові програмування Java з використанням технології доступу до даних JDBC, тому що ця технологія дозволяє розробляти додатки з графічним інтерфейсом та включає засоби для роботи з базами даних різних форматів. З їх допомогою ми можемо виконувати будь-які SQL-запити та обробляти їх результати, отримувати доступ до окремих таблиць бази даних тощо [18].

2.2 Розробка бізнес-правил

Програма має мету відтворити реалізацію музичного порталу у вигляді мобільного додатку.

Таким чином було сформовано основні бізнес правила.

- кожна пісня може мати тільки одного автора;
- автор може виконувати багато пісень;
- альбом має багато пісень;
- пісня може належати тільки одному альбому;
- пісня належить лише одному жанру;
- до одного жанру належить багато пісень;
- кожен користувач може додавати свій диск та редагувати його.

Проаналізувавши ці бізнес правила можна перейти до побудови функціональної структури даної предметної області.

2.3 Розробка концептуальної моделі

Концептуальне проектування – побудова семантичної моделі предметної області, тобто інформаційної моделі найбільш високого рівня абстракції. Така модель створюється без орієнтації на якусь конкретну СУБД і модель даних. Терміни «семантична модель» і «концептуальна модель» є синонімами. Крім того, в цьому контексті рівноправно можуть використовуватися слова «модель бази даних» та «модель предметної області»[18].

Дослідивши предметну область, стало відомо, які сутності будуть брати участь у створенні інформаційної системи. Опис кожної сутності та особливості її використання зображено на таблиці 2.1.

Таблиця 2.1 – Відомості про типи сутностей

Назва сутності	Опис
Альбом	Містить основну інформацію про альбом
Користувач (PasUser)	Містить дані для авторизації користувача
Користувач (PasAdmin)	Містить дані для авторизації адміна
Жанр	Містить код та назву жанру
Пісня	Відомості про пісню
Виконавець (Автор)	Інформація щодо виконавця

Продовження таблиці 2.1

Назва сутності	Опис
Диск	Створюється користувачем для групування пісень

У даній таблиці виділені основні сутності, зв'язки яких можуть спровокувати виникнення інших сутностей (табл. 2.2).

Таблиця 2.2 – Відомості про типи зв'язків

Назва сутності батька	Назва сутності дочірня	Кардинальні числа	Тип зв'язку
Альбом	Пісня	1:М	Не ідентифікуюча
Виконавець	Пісня	1:М	Не ідентифікуюча
Жанр	Пісня	1:М	Не ідентифікуюча
Пісня	Диск	1:М	Ідентифікуюча

На основі цих таблиць було побудовано діаграму Чена (Рис. 2.1).

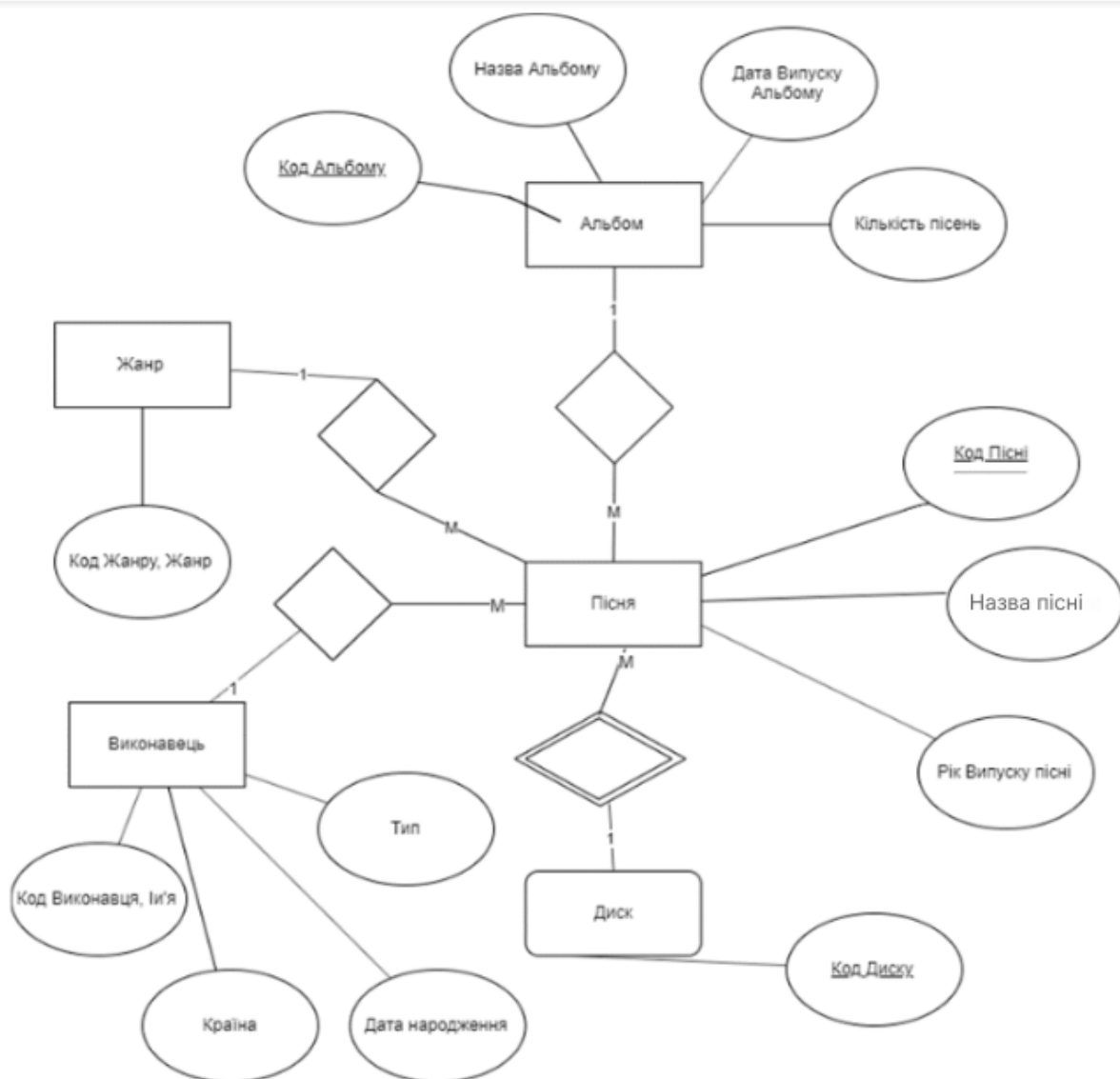


Рисунок 2.1 – Концептуальна модель даних

2.4 Функціональна структура

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед

прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання[7].

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою (Рис. 2.2).

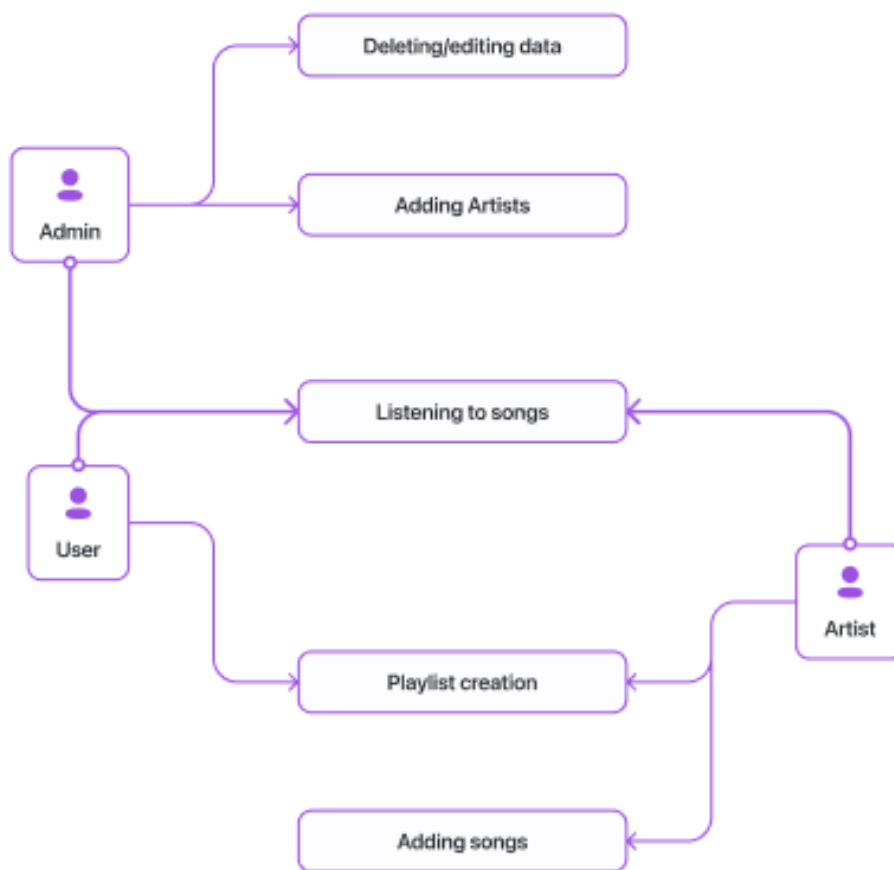


Рисунок 2.2 – Case-діаграма

Звичайний слухач – це роль, яку можна отримати після авторизації. Можливості цієї ролі невеликі, але задовольняють ідеї прослуховування музики.

Заходячи як виконавець, користувач отримує більше можливостей, але перша авторизація у цій ролі проходить лише через адміна. Тому просто так звичайні користувачі не мають змоги викладати музику, що частково обмежує їх можливості.

Адмін має повний доступ до бд, а разом з цим і величезну відповідальність. Він може редагувати, додавати та видаляти всі дані в таблицях.

2.5 Побудова логічної моделі даних

Для переходу з концептуальної моделі даних до логічної, потрібно позбутися багатозначних атрибутів, супер класів та зв'язків M:N. Багатозначні атрибут замінюється декількома іншими (Рис. 2.3).

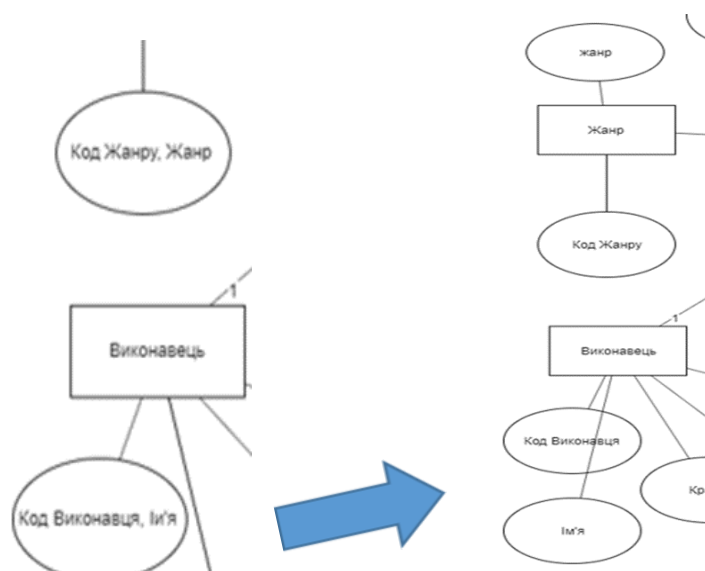


Рисунок 2.3 – Перехід до логічної моделі

Перевіривши БД засобами нормалізації, було з'ясовано, що вона знаходиться у третій нормальній формі. А саме:

– всі атрибути містять атомарні значення. Серед атрибутів не зустрічаються повторювані групи (кілька значень для кожного екземпляра);

– кожен неключовий атрибут повністю функціонально залежить від первинного ключа (не повинно бути залежності від частини ключа);

– ніякий неключових атрибут не залежить від іншого неключового атрибута (не має взаємозалежності між неключових атрибутами).

Детальний розбір атрибутів відображено на таблицях 2.3 – 2.11.

Таблиця 2.3 – Атрибути сутності “Диск”

Назва атрибута	Первинний ключ	Посилається на
Код Диску	Так	Не зовнішній ключ
Код Пісні	Так	Пісня (Код Пісні)

Таблиця 2.4 – Атрибути сутності “Пісня”

Назва атрибута	Первинний ключ	Посилається на
Код Пісні	Так	Не зовнішній ключ
Назва Пісні	Ні	Не зовнішній ключ
Дата випуску	Ні	Не зовнішній ключ

Продовження таблиці 2.4

Назва атрибута	Первинний ключ	Посилається на
Код Жанру	Ні	Жанр (Код Жанру)
Код Альбому	Ні	Альбом (Код Альбому)
Код Виконавця	Ні	Виконавець (Код Виконавця)

Таблиця 2.5 – Атрибути сутності “Жанр”

Назва атрибута	Первинний ключ	Посилається на
Код Жанру	Так	Не зовнішній ключ
Жанр	Ні	Не зовнішній ключ

Таблиця 2.6 – Атрибути сутності “Альбом”

Назва атрибута	Первинний ключ	Посилається на
Код Альбому	Так	Не зовнішній ключ
Назва альбому	Ні (потенційний)	Не зовнішній ключ
Дата Випуску	Ні	Не зовнішній ключ

Продовження таблиці 2.6

Назва атрибута	Первинний ключ	Посилається на
Кількість Пісень	Ні	Не зовнішній ключ

Таблиця 2.7 – Атрибути сутності “Користувач”

Назва атрибута	Первинний ключ	Посилається на
Login	Так	Не зовнішній ключ
Password	Ні	Не зовнішній ключ

Таблиця 2.8 – Атрибути сутності “Адмін”

Назва атрибута	Первинний ключ	Посилається на
Login	Так	Не зовнішній ключ
Password	Ні	Не зовнішній ключ

Усього в ER-діаграмі присутні 7 сутностей: “User”, “Admin”, “Album”, “Song”, “Genre”, “Disk”, “Performer”. Розкриття цих таблиць:

- User(Login, Password)
- Admin (Login, Password)
- Album(ID_Album, Name_Album, Date_RelAl, Number_of_Songs)
- Song(ID_Song, Name_Song, Date_Rel_S, ID_Album, ID_Performer, ID_Genre)

- Performer(ID_Performer, Name, Date, Country, Type)
- Genre(ID_Genre, Genre)
- Disk(ID_Disk, ID_Song)

Отримана логічна модель даних зображена на рисунку 2.4.

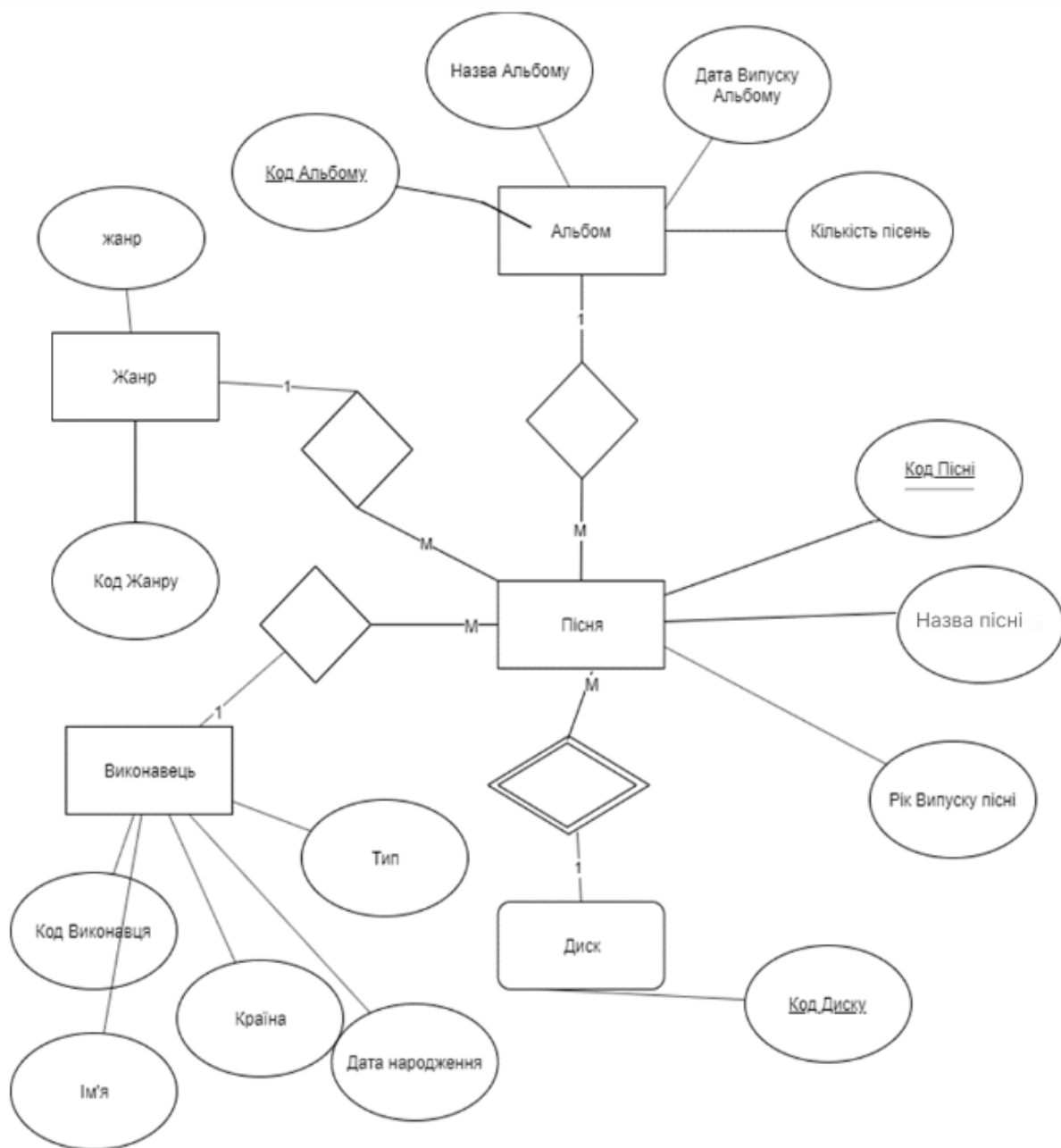


Рисунок 2.4 – Логічна модель даних

2.6 Побудова моделі даних за допомогою CASE-засобу візуального проектування ERWIN

На основі концептуальної та логічної моделі було побудовано модель даних за допомогою програми візуального проектування ERWIN (Рис. 2.5).

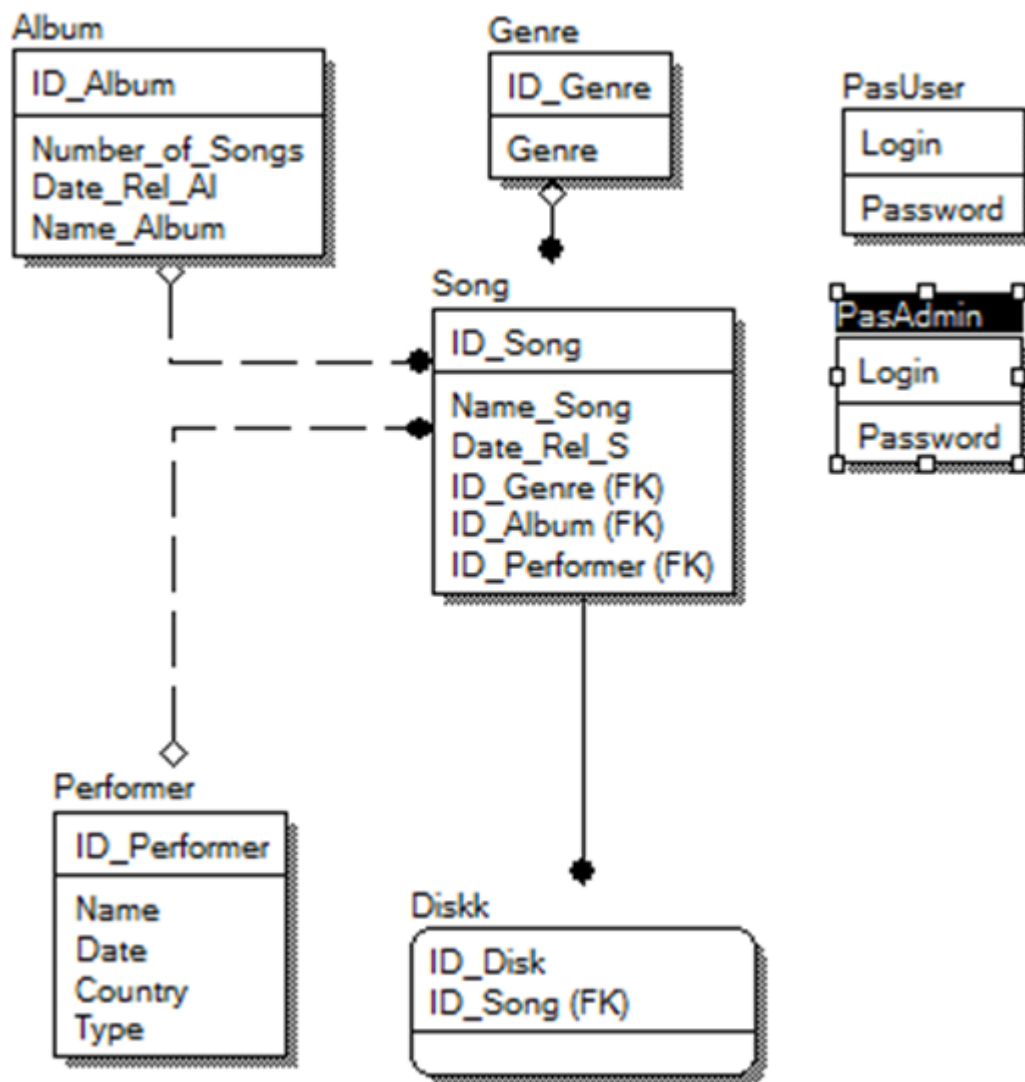


Рисунок 2.5 – ER-діаграма

2.7 Фізичне проектування

Фізичне проектування – створення схеми бази даних для конкретної СУБД. Специфіка конкретної СУБД може включати в себе обмеження на іменування об'єктів бази даних, обмеження на підтримувані типи даних та інші[24]. На рисунку 2.6 зображено схему бази даних для ІС музикального порталу.

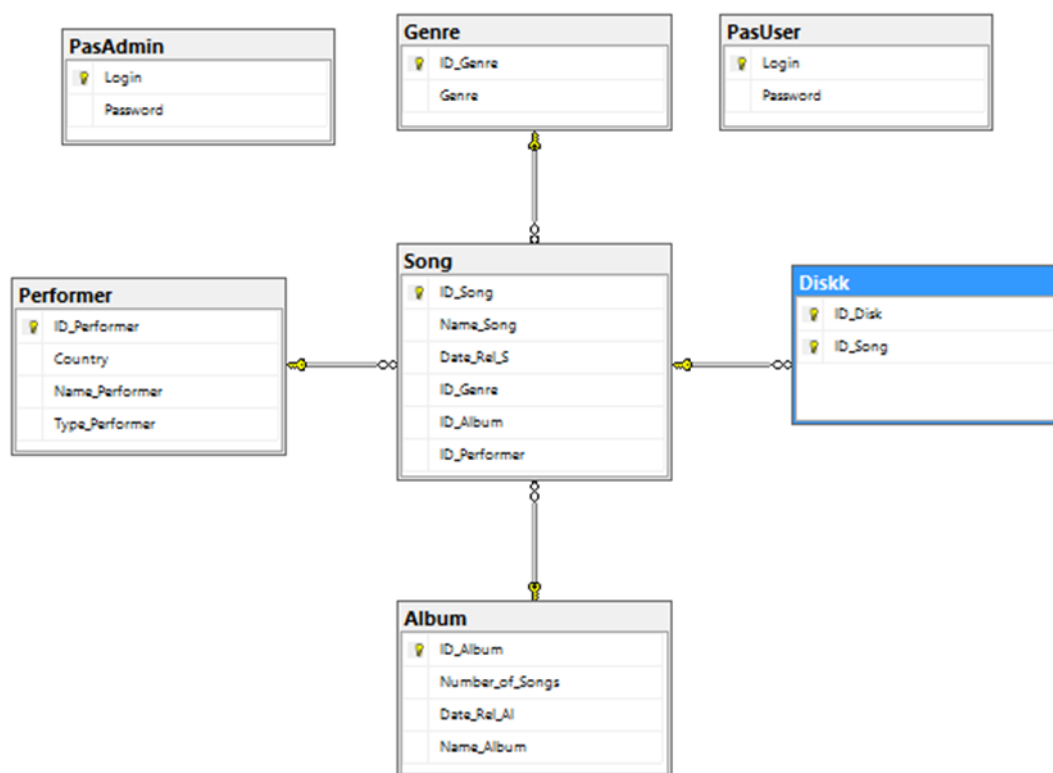


Рисунок 2.6 – БД в СУБД SQL server

Фізична структура бази даних для інформаційної системи «Музичний портал» наведена нижче за допомогою відповідних команд CREATE TABLE.

Лістинг 2.1 Створення структури бази даних:

- Структура таблиці «Користувач»:

```
CREATE TABLE PasUser
(
Login          VARCHAR(20) NOT NULL,
Password       VARCHAR(20) NULL
);
ALTER TABLE PasUser
ADD PRIMARY KEY (Login);
```

- Структура таблиці «Адмін»:

```
CREATE TABLE PasAdmin
(
Login          VARCHAR(20) NOT NULL,
Password       VARCHAR(20) NULL
);
ALTER TABLE PasAdmin
ADD PRIMARY KEY (Login);
```

- Структура таблиці «Альбом»:

```
CREATE TABLE Album
(
ID_Album       INTEGER NOT NULL,
Number_of_Songs  INTEGER NULL,
Date_Rel_Al    DATE NULL,
Name_Album     VARCHAR(20) NULL
);
ALTER TABLE Album
ADD PRIMARY KEY (ID_Album);
```

- Структура таблиці «Виконавець»:

```
CREATE TABLE Performer
(
ID_Performer    INTEGER NOT NULL,
Country        VARCHAR(20) NULL,
Name           char(18) NULL,
Date          DATE NULL,
Type          VARCHAR(20) NULL
);
ALTER TABLE Performer
ADD PRIMARY KEY (ID_Performer);
```

- Структура таблиці «Жанр»:

```
CREATE TABLE Genre
(
ID_Genre       INTEGER NOT NULL,
Genre         VARCHAR(20) NULL
);
ALTER TABLE Genre
ADD PRIMARY KEY (ID_Genre);
```

- Структура таблиці «Пісня»:

```
CREATE TABLE Song
(
ID_Song        INTEGER NOT NULL,
Name_Song     VARCHAR(20) NULL,
Date_Rel_S    DATE NULL,
ID_Genre      INTEGER NULL,
```

```

ID_Album      INTEGER NULL,
ID_Performer  INTEGER NULL
);
ALTER TABLE Song
ADD PRIMARY KEY (ID_Song);

```

- Структура таблиці «Диск»:

```

CREATE TABLE Diskk
(
ID_Disk      char(18) NOT NULL,
ID_Song      INTEGER NOT NULL
);
ALTER TABLE Diskk
ADD PRIMARY KEY (ID_Disk, ID_Song);

```

Лістинг 2.2 Додавання зовнішніх ключей:

- ALTER TABLE Diskk


```

ADD CONSTRAINT R_31 FOREIGN KEY (ID_Song) REFERENCES
Song (ID_Song);

```
- ALTER TABLE Song


```

ADD CONSTRAINT R_1 FOREIGN KEY (ID_Genre) REFERENCES
Genre (ID_Genre);

```
- ALTER TABLE Song


```

ADD CONSTRAINT R_23 FOREIGN KEY (ID_Album) REFERENCES
Album (ID_Album);

```
- ALTER TABLE Song


```

ADD CONSTRAINT R_29 FOREIGN KEY (ID_Performer) REFERENCES
Performer (ID_Performer);

```

3 РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДЛЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Реєстрація та вхід

Реєстрація користувача не є обов'язковою у програмах для потокової передачі музики, але настійно рекомендується. Її можна використовувати, щоб дізнатися більше про користувачів, таких як їх музичні уподобання в жанрі, виконавцях, мові тощо. Реєстрація та вхід до системи повинні бути максимально швидкими. В ідеалі всього кілька натискань.

У програмі використовуватимуться класичний підхід до авторизації: електронна пошта та пароль (Рис. 3.1).

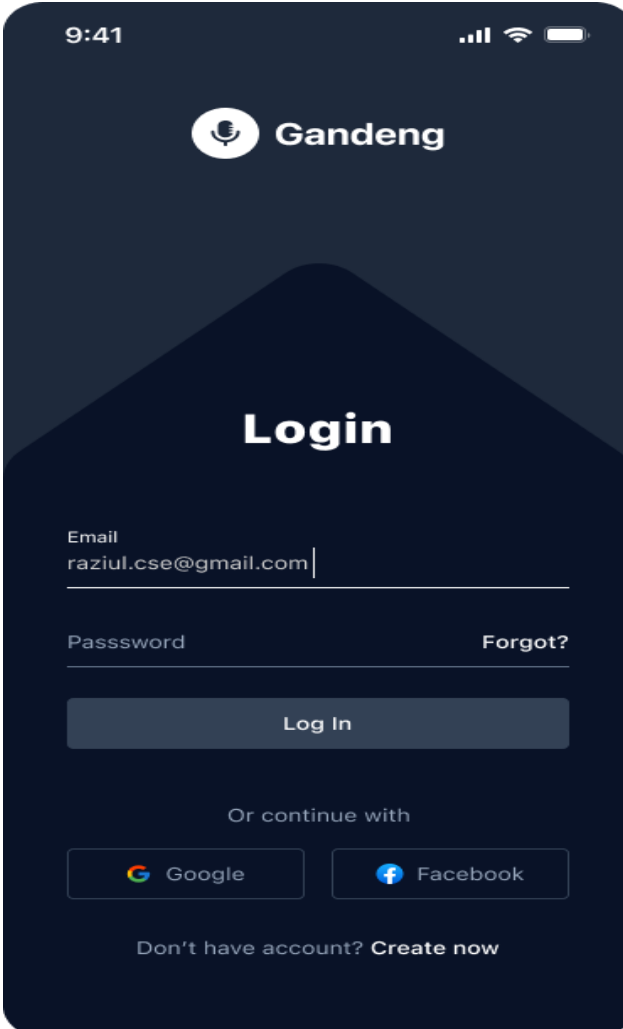


Рисунок 3.1 – Форма авторизації

Потенційний користувач має декілька рівнів доступу у музичному порталі: як звичайний слухач або як виконавець. Відповідно до рівню користувач може відтворювати різні дії. Таким чином слухач може створювати власні добірки музики, а виконавець має змогу додавати нові пісні від свого імені.

3.2 Профіль користувача

Після реєстрації користувачі повинні мати можливість налаштувати свій особистий профіль, що дозволить їм створювати та керувати своїми персоналізованими плейлистами. Це дозволить власнику програми аналізувати дані користувачів та дізнаватися про їх переваги. Особистий обліковий запис також може дозволити користувачам обмінюватися музикою та навіть завантажувати свої власні треки.

Для того, щоб зареєструватися як виконавець, користувач має створити профіль слухача, потім написати листа до вказаної пошти з деталями профілю для того, щоб адмін мав змогу перевірити його як потенційного виконавця.

3.2.1 Інтерфейс профілю слухача

Інтерфейс профілю слухача має базовий набір функцій. Одразу після реєстрації з'являється вікно вибору жанрів. На основі цього вибору і буде створена перша добірка пісень. Користувач на цьому рівні може додати запропоновані пісні до свого першого плейлисту (Рис. 3.2-3.4).

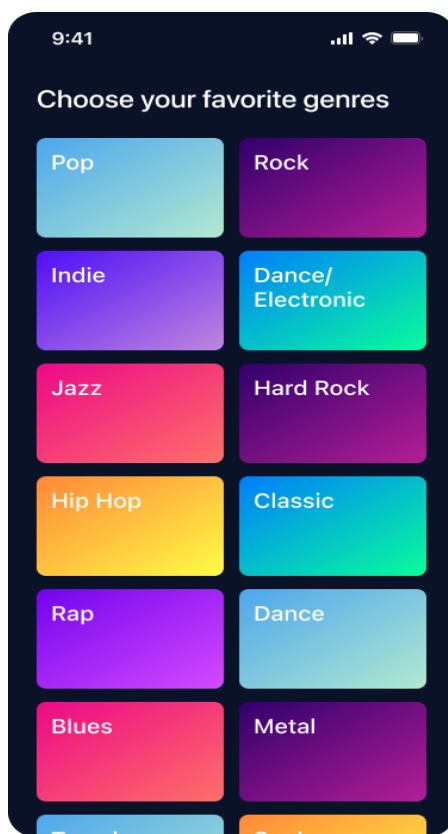


Рисунок 3.2 – Вибір жанру

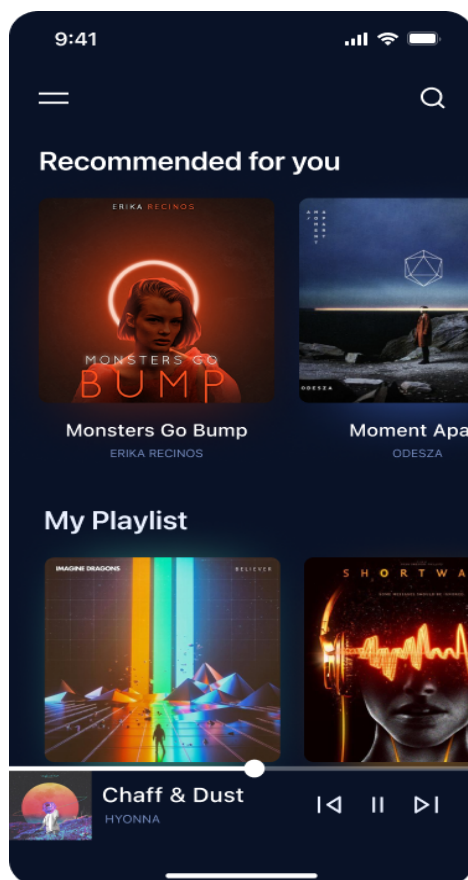


Рисунок 3.3 – Згенеровані рекомендації

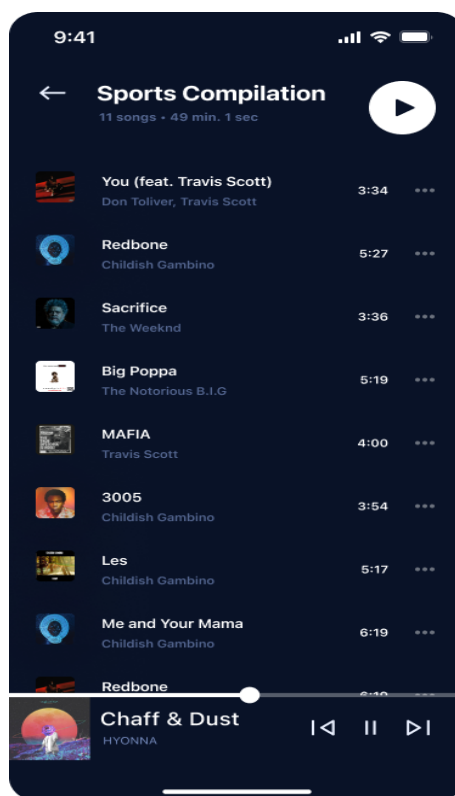


Рисунок 3.4 – Добірка з рекомендованими піснями

У налаштуваннях профілю користувач має змогу змінювати наступні дані: список жанрів, підписки на виконавців, та особисту інформацію, як то ім'я та пароль.

3.2.2 Інтерфейс профілю виконавця

Профіль виконавця має ті ж самі функції, що і в слухача, але до них додаються нові. Такий користувач може у своєму профілі додавати свої пісні та альбоми. Крім цього в нього є можливість бачити кількість прослуховувань його творів, та кількість підписників (Рис. 3.5).

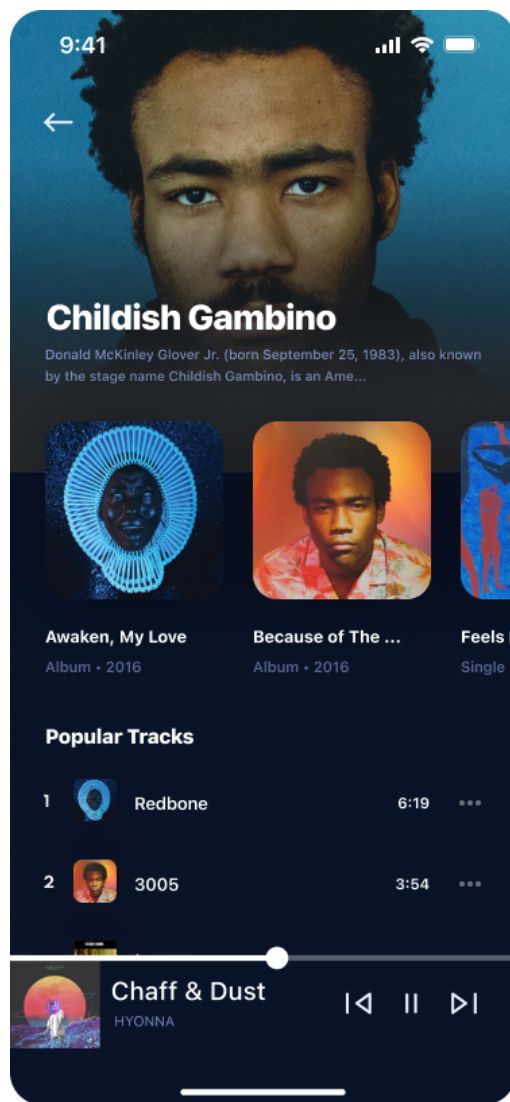


Рисунок 3.5 – Профіль виконавця

3.3 Профіль технічної підтримки

Профіль технічної підтримки (або ж профіль адміна) має суттєві відмінності від інтерфейсу звичайних користувачів. Цей профіль не націлений на створення добірок або ж прослуховування музики. Адмін має є дозвіл на видалення пісень та користувачів за певних причин (порушення політики порталу, видавання чужих пісень за власні тощо).

3.4 Пошук

Функція пошуку дозволяє користувачам швидко знаходити бажані пісні, а також виконавців, жанри, альбоми та багато іншого. Це допомагає користувачам легко рухатися платформою. Пошук відбувається по назвам пісень, або виконавцям (Рис 3.4).

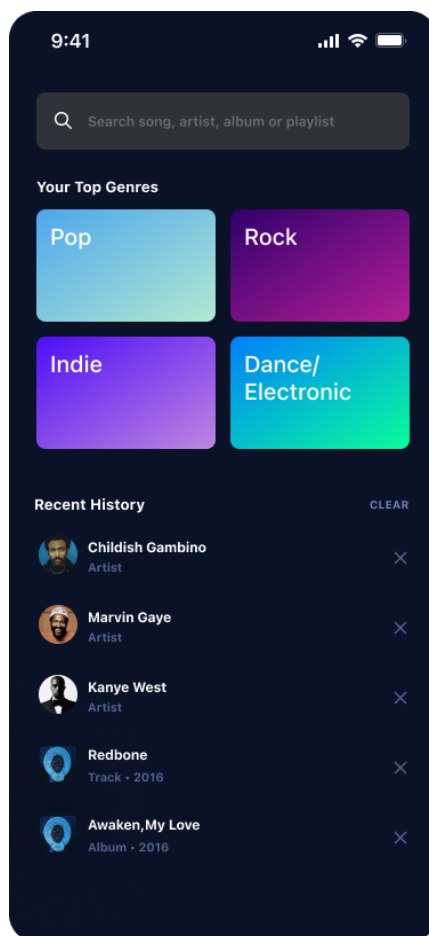


Рисунок 3.6 – Реалізація пошуку

3.5 Музичний програвач

Безумовно, музична програма не може працювати без музичного плеєра. Це основна функція, яка дозволяє користувачам транслювати та слухати музику. Екран програвача дозволяє користувачам виконувати такі дії, як

відтворення, пауза, наступний або попередній, керування гучністю та багато іншого. Він також може мати додаткові елементи керування для лайкання пісні, перемішування музики, додавання до списку відтворення користувача тощо. Крім того, на екрані програвача також відображається обкладинка альбому, інформація про виконавця або альбом, а також відображається список пісень в альбомі (Рис. 3.7).

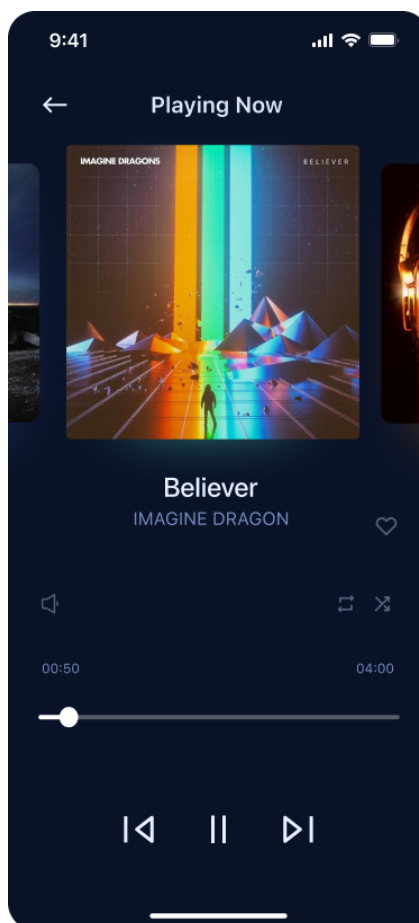


Рисунок 3.7 – Музичний програвач

3.6 Алгоритм генерування рекомендацій

Добірки з рекомендаціями генеруються декількома шляхами:

- на основі обраних жанрів;

- на основі “лайкнутих” пісень. У цьому випадку портал аналізує назву пісні, виконавця та жанрів. Тому така підборка може бути кращою для слухача.

3.7 Персоналізовані рекомендації

Вбудована функція штучного інтелекту буде аналізувати поведінку та уподобання користувачів у музиці, щоб давати персональні рекомендації. Він запропонує пісні, альбоми або навіть подкасти, які можуть сподобатися користувачеві, спираючись на спостереження, такі як звички користувача до прослуховування, географічне розташування тощо.

3.8 Технічна реалізація

По-перше, стек технологій залежатиме від платформ/пристроїв, на які орієнтується розробка – зазвичай це Android, iOS та веб-версія. Якщо бюджет обмежений, то можна створити програму для однієї платформи на основі цільової аудиторії, а потім налаштувати націлення на іншу платформу. Або ж використовувати такі технології, як Flutter або React Native для створення кросплатформенного додатка з єдиною кодовою базою, яка працює на декількох платформах.

Нижче наведений список різноманітних інструментів та технологій програмування, які можна використовувати для розробки музичної програми.

Таблиця 3.1 – Інструменти та технології програмування

Інструменти	Технології
Frontend	Java, Kotlin for Android; Swift, Objective-C for iOS
Web app development	Bootstrap, HTML5
Backend and server-side	PHP, Java, Python, Laravel, ROR
Frameworks	Hadoop, Apache Storm, Hub Framework
Server	Nginx
Database for music files storage compatible with both web and mobile platforms	PostgreSQL, Cassandra, MySQL

Продовження таблиці 3.1

Real-time data processing and management	Apache Kafka
Cloud-based storage and caching	WS, Amazon S3, Google Cloud Platform, Azure
CDN	Amazon CloudFront or Google Cloud CDN
Platform infrastructure	Dataproc
Building desktop-based UI	Chromium Embedded Framework
DevOps	Docker, Datadog, TestFlight
Utilities	Google Analytics, Twilio SendGrid

Продовження таблиці 3.1

Admin Panel	Silex, MySQL, Composer HTML5, CSS3, Jquery, Bower
-------------	---

Та інші, такі як Java Android SDK, Retrofit, Fresco, Crashlytics і Realm, для забезпечення безперебійної роботи додатків [2].

Також можуть знадобитися різні технології в залежності від інтеграції інших функцій програми, таких як:

Таблиця 3.2 – Варіанти додаткових інтеграцій

Інструменти	Технології
Data management	Datastax
Payment integration	PayPal, Braintree
Social media integration	Facebook, Google, and Spotify SDK
Push notifications	Twilio

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено інформаційну систему для ПЗ “Музичний портал” та реалізовано її у вигляді мобільного додатку.

У ході роботи були вирішені наступні завдання:

- розглянуто принципи нормалізації та побудови БД.
- проаналізовано теорію проектування реляційних БД на основі побудови ер-моделі
- досліджено особливості розробки бізнес правил на основі опису ПЗ та вимог до функціональності ІС
- розглянуто особливості розробки ER-моделі
- відображено модель за допомогою ER-діаграми в синтаксисі Чена;
- розроблено модель даних за допомогою CASE-засобу візуального проектування даних ERwin;
- сформовано структуру БД в СУБД
- розроблено моделі реалізації ІС у вигляді мобільного додатку
- проведено вибір інструментальних засобів для створення застосунку, що дозволило виявити необхідні технології для процесу розробки;
- наведено детальний опис етапів розробки застосунку для розпізнавання та класифікації образів кулінарних страв;
- виявлено перспективи подальшої роботи для подальшого покращення застосунку;
- зроблено висновки щодо виконаної роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Allen Veronica. (2021). *The New 2022 UI/UX For Beginners And Experts: UX/UI Design for Automatic Designers*. 92
2. Amasha, M. A., Areed, M. F., Khairy, D., Atawy, S. M., Alkhalaf, S., & Abougalala, R. A. (2021). Development of a Java-based Mobile application for mathematics learning. *Education and Information Technologies*, 26(1), 945-964.
3. Balagtas-Fernandez, F., & Hussmann, H. (2009, July). Evaluation of user-interfaces for mobile application development environments. In *International Conference on Human-Computer Interaction* (pp. 204-213). Springer, Berlin, Heidelberg.
4. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (pp. 930-935). IEEE.
5. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, 19(1), 23-28.
6. Bryan, N. J., Herrera, J., Oh, J., & Wang, G. (2010, June). MoMu: A Mobile Music Toolkit. In *NIME* (pp. 174-177).
7. Challa, J. S., Paul, A., Dada, Y., Nerella, V., Srivastava, P. R., & Singh, A. P. (2011). Integrated software quality evaluation: a fuzzy multi-criteria approach. *Journal of Information Processing Systems*, 7(3), 473-518.
8. Edlom, J. (2022). *The Engagement Imperative: Experiences of Communication Practitioners' Brand Work in the Music Industry*. *Media and Communication*, 10(1), 66-76.

9. Hagen, A. N. (2022). Datafication, literacy, and democratization in the music industry. *Popular Music and Society*, 45(2), 184-201.
10. Hagggar, P. (2000). *Practical Java: programming language guide*. Addison-Wesley Professional. 279
11. K. Ziegler. *Methods of designing software systems*. - M: Mir, 1985, p. 15-23.
12. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.
13. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, 5(57).
14. Lange, D. B., & Oshima, M. (1998). Mobile agents with Java: the Aglet API. *World Wide Web*, 1(3), 111-121.
15. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. *International Journal of Advanced Trends in Computer Science and Engineering*. 9(4). – 4701-4706.
16. Lyashenko, V., Mohammad, A., & Kobylin, O. (2015). Experiments with Fusion of Images with Use of Wavelet Transformation in Problems of the Text Information Analysis.
17. Maria Eriksson, Rasmus Fleischer, Anna Johansson, Pelle Snickars, Patrick Vonderau. (2019). MIT Press. *Spotify Teardown: Inside the Black Box of Streaming Music*. 286
18. Martin, D., Rooksby, J., Rouncefield, M., & Sommerville, I. (2007, May). 'Good'organisational reasons for'Bad'software testing: An ethnographic study of testing in a small software company. In 29th international conference on software engineering (ICSE'07) (pp. 602-611). IEEE.

19. Martins, D. M. L. (2019). Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities. *Information Systems*, 83, 89-100.
20. Mohamed Sarrab, Hafedh Al-Shihi, Naveen Safia(2021). Handbook of Mobile Application Development: A Guide to Selecting the Right Engineering and Quality Features. 115
21. Nathan Clark. (2018). *Ui/ux Design Basics and Fundamentals*. 203.
- Späth, P., & Friesen, J. (2020). *Learn Java for Android Development: Migrating Java SE Programming Skills to Mobile Development*. Apress.
22. R.C. Holt. *Structure of Computer Programs: A Survey // Proceedings of the IEEE*, 1975, 63(6). - p. 879-893.
23. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (pp. 665-670). IEEE.
24. Rawat, B., & Purnama, S. (2021). MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung. *International Journal of Cyber and IT Service Management*, 1(2), 173-179.
25. Sarrab, M., & Rehman, O. M. H. (2014). Empirical study of open source software selection for adoption, based on software quality characteristics. *Advances in Engineering Software*, 69, 1-11.
26. Shu, J., Shen, X., Liu, H., Yi, B., & Zhang, Z. (2018). A content-based recommendation algorithm for learning resources. *Multimedia Systems*, 24(2), 163-173.
27. Vlachos, P., Vrechopoulos, A. P., & Doukidis, G. (2003). Exploring consumer attitudes towards mobile music services. *International Journal on Media Management*, 5(2), 138-148.

28. Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura. (2012). Programming Android: Java Programming for the New Generation of Mobile Devices, 566.
29. Вискребенцева С.О., Кобилін О.А. (2019) Методи сегментації зображень. Матеріали XXIII міжнародного молодіжного форуму. Радіоелектроніка та молодь у XXI столітті, 19-20.
30. Работягов, А. В., Ляшенко, В. В., & Кобылин, О. А. (2016). Сегментация сложных изображений цитологических препаратов.