

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»**

Кафедра інформаційних систем в економіці

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему

**ІНФОРМАЦІЙНА СИСТЕМА ЗАБЕЗПЕЧЕННЯ РОБОТИ ЦИФРОВОЇ
АУДІОТЕКИ**

здобувача Усика Владислава Володимировича

Науковий керівник:

к.т.н., доцент

_____ Кривошеєв К. В.

**Робота допущена до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент.

_____ Тішков Б.О.

Київ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАДИМА
ГЕТЬМАНА

Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)

освітньо-професійної програми

_____ Помазун О.М.

“ _____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Тішков Б.О.

“ _____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Усика Владислава Володимировича

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи

на тему: «Інформаційна система забезпечення роботи цифрової аудіотеки»

Тему затверджено наказом ректора Університету від «7» бер. 2025 р. № 466-ст

Кваліфікаційна бакалаврська робота виконується на матеріалах

отриманих з мережі Інтернет, відкритих джерел інформації

План кваліфікаційної бакалаврської роботи

Розділ I Характеристика та аналіз предметної галузі

Розділ II Розробка вимог і моделювання інформаційної системи

Розділ III Проектування та реалізація компонентів інформаційної системи

Об'єкт дослідження: Процес розробки та впровадження інформаційної системи для забезпечення роботи цифрової аудіотеки.

Предмет дослідження: Методи та підходи до проектування інформаційних систем для керування цифровим аудіоконтентом, вимоги користувачів до функціональності та інтерфейсу аудіотеки, зокрема щодо транскрипції подкастів та легкого доступу до завантаження аудіофайлів, а також сучасні технології для розробки веб- та мобільних додатків для роботи з мультимедійними даними та реалізації функціоналу штучного інтелекту.

Мета кваліфікаційного бакалаврського проекту: Створення інноваційної інформаційної системи забезпечення роботи цифрової аудіотеки, яка включатиме AI-модуль для транскрипції подкастів та надаватиме легкий доступ до завантаження аудіофайлів для будь-якого користувача.

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі I необхідно: проаналізувати сучасний стан ринку цифрових аудіотек та платформ для керування аудіоконтентом, дослідити існуючі програмні рішення та сервіси для цифрових аудіотек (наприклад, Spotify, Apple Music, SoundCloud, системи транскрипції аудіо тощо), виявити їхні переваги та недоліки, а також визначити можливості для покращення та інновацій.

У розділі II необхідно: розробити детальні функціональні та нефункціональні вимоги до інформаційної системи цифрової аудіотеки, включаючи вимоги до AI-модуля для транскрипції подкастів та механізмів завантаження аудіофайлів, створити технічне завдання на розробку інформаційної системи, побудувати концептуальну та логічну моделі системи, включаючи архітектуру, взаємодію компонентів, діаграми потоків даних, а також розробити схему бази даних для зберігання метаданих аудіофайлів, транскрипцій, інформації про користувачів та їхні плейлисти, визначити та обґрунтувати оптимальний технологічний стек для реалізації проекту, включаючи мови програмування, фреймворки, системи управління базами даних та технології для реалізації AI-модуля.

У розділі II необхідно: деталізувати архітектуру системи, включаючи клієнтську та серверну частини, інтеграцію AI-модуля, реалізувати основні функціональні модулі інформаційної системи цифрової аудіотеки (реєстрація/авторизація, завантаження/керування аудіо, пошук, створення бібліотеки улюбленого, відтворення, функціонал транскрипції), розробити дизайн інтерфейсу користувача (UI) та забезпечити зручність користування (UX), використовуючи наданий прототип Figma як основу та адаптуючи його під специфіку аудіотеки та нові функції, розробити або інтегрувати модуль штучного інтелекту для автоматичної транскрипції аудіофайлів (подкастів) у текстовий формат, провести всебічне тестування системи (функціональне, інтеграційне, користувацьке, тестування AI-модуля), виявити та усунути помилки, оцінити продуктивність та стабільність роботи, підготувати рекомендації щодо впровадження системи та розробити необхідну документацію (користувацьку, технічну).

**Завдання підготував
науковий керівник**

Кривошесв Константин Валерійович

“ _____ ” _____ 2025 р.

**Завдання одержав
здобувач**

Усик Владислав Володимирович

“ _____ ” _____ 2025 р.

Відгук

про кваліфікаційну бакалаврську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Усика Владислава Володимировича

на тему

«Інформаційна система забезпечення роботи цифрової аудіотеки»

1. Актуальність теми:

Актуальність теми є високою через стрімке зростання обсягів аудіоконтенту та обмеженість функціоналу існуючих платформ. Інтеграція штучного інтелекту для автоматичної транскрипції є інноваційним рішенням, що безпосередньо вирішує нагальну проблему підвищення доступності та пошукових можливостей аудіоматеріалів

2. Позитивні риси кваліфікаційної бакалаврської роботи:

Робота вирізняється чіткою структурою та комплексним підходом, що охоплює всі етапи від аналізу до реалізації. Ключовою перевагою є детально задокументований етап проектування з використанням сучасних засобів моделювання (UML/SysML) та обґрунтований вибір сучасного технологічного стеку.

3. Наявність самостійних розробок автора.

Самостійний внесок автора полягає у практичній реалізації прототипу інформаційної системи. Це включає розробку серверної архітектури на Python/Flask, клієнтського інтерфейсу на React, проектування бази даних PostgreSQL та головне технічне досягнення — успішну інтеграцію AI-сервісу OpenAI Whisper.

4. Цінність теоретичних висновків та практичних рекомендацій:

Теоретична цінність роботи полягає в систематизації підходу до інтеграції AI в аудіоплатформи, що може слугувати методологією для майбутніх проєктів. Практична цінність — у створенні функціонального прототипу, а конкретні рекомендації автора щодо впровадження повнотекстового пошуку та ідентифікації спікерів значно підвищують довгостроковий потенціал проєкту.

5. Наявність недоліків:

Основний недолік, коректно визначений автором, полягає у статусі роботи як мінімально життєздатного продукту (MVP). Це проявляється у відсутності таких важливих функцій, як повнотекстовий пошук по транскрипціях та нативні мобільні додатки. Крім того, залежність від стороннього платного API створює потенційні ризики щодо вартості та стабільності роботи системи.

6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК:

Кваліфікаційна робота є завершеним дослідженням, що успішно досягає поставленої мети. Продемонстрована висока компетентність у системному аналізі, проектуванні та практичній розробці. Робота вирішує актуальну проблему за допомогою інноваційного підходу та відповідає вимогам, що висувуються до кваліфікаційних робіт бакалаврського рівня. Рекомендується до захисту перед Екзаменаційною комісією з позитивною оцінкою.

Науковий керівник

_____ к.т.н., доцент Кривошеев К.В.

(підпис)

«__» _____ 20__ р.

Рецензія
на кваліфікаційну бакалаврську роботу
здобувача вищої освіти
Усика Владислава Володимировича

Тема: «Інформаційна система забезпечення роботи цифрової аудіотеки»

1. Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення:

Тема роботи є беззаперечно актуальною через стрімке зростання обсягів цифрового аудіоконтенту та потребу в його ефективному аналізі. Розробка системи з інтегрованою AI-транскрипцією є доцільною, оскільки вирішує проблему обмеженої доступності та пошукових можливостей існуючих платформ, що обґрунтовує її інноваційність.

2. Якість проведеного дослідження:

Дослідження проведено на високому рівні, з дотриманням логічної послідовності етапів проектування інформаційних систем. Автор продемонстрував вміння застосовувати сучасні методології системного аналізу та моделювання для формалізації вимог та архітектурних рішень. Якість дослідження підтверджується глибиною аналізу предметної галузі та обґрунтованістю прийнятих проектних рішень.

3. Позитивні риси кваліфікаційної бакалаврської роботи:

До позитивних рис роботи належить комплексний підхід, що охоплює повний цикл розробки від аналізу до створення функціонального прототипу. Особливо варто відзначити практичну спрямованість роботи, що вилилась у реалізацію програмного продукту з використанням сучасного та обґрунтованого стеку технологій.

4. Зауваження:

Основним зауваженням є те, що розроблена система на даному етапі є мінімально життєздатним продуктом (MVP), що має певні функціональні обмеження. Зокрема, відсутній повнотекстовий пошук за згенерованими транскрипціями, а залежність від зовнішнього платного API є потенційним ризиком для подальшого розвитку проекту.

4. Практична значимість висновків і рекомендацій:

Практична значимість полягає у створенні функціонального прототипу, який може слугувати основою для розробки комерційного продукту або впровадження в освітніх та медіа-сферах. Надані автором висновки та конкретні технічні рекомендації щодо подальшого вдосконалення системи є цінними та спрямовані на перетворення прототипу на повноцінний конкурентоспроможний продукт.

Місце роботи та посада рецензента: Директор ТОВ "ЗАВОД МОДУЛЬ"

Науковий ступінь, учене звання _____

(підпис, ПІБ)

Підпис засвідчую: _____ Коновалова О.В.

(посада, підпис)

Місце печатки організації, де працює рецензент

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Інформаційна система забезпечення роботи цифрової
аудіотеки»

Київ: кафедра інформаційних систем в економіці, 2025 р.

Кваліфікаційна бакалаврська робота присвячена розробці інформаційної системи забезпечення роботи цифрової аудіотеки. У сучасному світі, де цифровий аудіоконтент набуває все більшої популярності, виникає потреба у створенні ефективних інструментів для його керування та доступності. Робота зосереджена на створенні інноваційної системи, яка вирішує цю проблему шляхом впровадження AI-модуля для автоматичної транскрипції подкастів, що значно полегшує пошук та аналіз аудіоматеріалів. Крім того, система забезпечує легкий та інтуїтивно зрозумілий доступ до завантаження аудіофайлів для всіх користувачів, підвищуючи їхню взаємодію з цифровою аудіотекою.

У першому розділі роботи проведено всебічний аналіз предметної галузі, виявлено поточні тенденції та недоліки існуючих рішень для цифрових аудіотек.

Другий розділ присвячений розробці детальних функціональних та нефункціональних вимог до системи, її моделюванню та обґрунтуванню вибору технологічних засобів.

У третьому розділі докладно описано процес проектування архітектури, реалізацію основних функціональних модулів, зокрема AI-модуля транскрипції, а також дизайн інтерфейсу користувача та результати тестування розробленої системи.

Робота демонструє комплексний підхід до створення сучасної інформаційної системи, здатної задовольнити потреби користувачів цифрових аудіотек.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 72 сторінки, 10 таблиць, 21 рисунок, список літератури з 30 найменувань, 1 додаток.

Тема: **«Інформаційна система забезпечення роботи цифрової аудіотеки».**

Перелік ключових слів: цифрова аудіотека, інформаційна система, транскрипція подкастів, штучний інтелект, OpenAI Whisper, розпізнавання мовлення, керування аудіоконтентом, Flask, React, PostgreSQL.

Предметом дослідження є методи та підходи до проектування інформаційних систем для керування цифровим аудіоконтентом, вимоги користувачів до функціональності та інтерфейсу аудіотеки, зокрема щодо транскрипції подкастів та легкого доступу до завантаження аудіофайлів, а також сучасні технології для розробки веб- та мобільних додатків для роботи з мультимедійними даними та реалізації функціоналу штучного інтелекту.

Об'єктом дослідження виступає процес розробки та впровадження інформаційної системи для забезпечення роботи цифрової аудіотеки.

Мета випускного бакалаврського проекту полягає у створенні інноваційної інформаційної системи забезпечення роботи цифрової аудіотеки, яка включатиме AI-модуль для транскрипції подкастів та надаватиме легкий доступ до завантаження аудіофайлів для будь-якого користувача.

Завданнями випускної бакалаврської роботи є: аналіз предметної галузі та існуючих рішень; розробка функціональних та нефункціональних вимог; моделювання архітектури системи та бази даних; проектування та реалізація ключових компонентів системи; інтеграція AI-модуля для транскрипції; тестування розробленого прототипу та розробка рекомендацій щодо його подальшого вдосконалення.

Апаратні та програмні засоби, що використовувались при проектуванні: хмарні платформи Heroku (Back-end) та Netlify (Front-end); мова програмування Python з фреймворком Flask для серверної частини; бібліотека React.js для

клієнтської частини; система управління базами даних PostgreSQL; зовнішній сервіс OpenAI Whisper для транскрипції.

Результати, досягнуті в процесі роботи (також їх новизна та ступінь впровадження): розроблено та спроектовано функціональний прототип (MVP) інформаційної системи цифрової аудіотеки. Наукова новизна полягає у системному підході до інтеграції сучасної моделі розпізнавання мовлення (OpenAI Whisper) в єдиний програмний продукт для керування аудіоконтентом, що вирішує проблему обмеженої доступності аудіоматеріалів. Ступінь впровадження – функціональний прототип, апробований на контрольному прикладі.

Одержані результати можуть бути використані (рекомендації щодо практичного застосування здобутих результатів, із зазначенням ефективності, сфери використання): розроблений прототип може бути впроваджений для особистого використання, в освітніх установах або медіа-компаніях для ефективного керування аудіоконтентом. Результати можуть слугувати основою для створення комерційного продукту. Рекомендовано вдосконалення системи шляхом реалізації повнотекстового пошуку по транскрипціях, ідентифікації спікерів та розробки нативних мобільних додатків.

Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	3
ВСТУП.....	4
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	8
1.1 Характеристика предметної галузі та об'єкта дослідження.....	8
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	11
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	15
2.1 Аналіз і специфікація вимог до інформаційної системи/підсистеми.....	15
2.2 Постановка та алгоритм розв'язання задачі.....	19
2.3 Моделювання інформаційної підсистеми.....	31
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ.....	46
3.1 Інформаційне забезпечення.....	46
3.2. Технічне забезпечення.....	58
3.3 Програмне забезпечення.....	60
3.4. Результати реалізації інформаційної підсистеми.....	63
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

EOM – Електронно-обчислювальна машина

КТЗ – Комплекс технічних засобів

ACID – Atomicity, Consistency, Isolation, Durability (Атомарність, узгодженість, ізольованість, довговічність)

ASR – Automatic Speech Recognition (Автоматичне розпізнавання мовлення)

CDN – Content Delivery Network (Мережа доставки контенту)

DDL – Data Definition Language (Мова опису даних)

MVP – Minimum Viable Product (Мінімально життєздатний продукт)

MVCC – Multi-Version Concurrency Control (Багатоверсійне управління конкурентним доступом)

NLP – Natural Language Processing (Обробка природної мови)

OLAP – Online Analytical Processing (Система аналітичної обробки)

OLTP – Online Transaction Processing (Система операційної обробки транзакцій)

RAD – Rapid Application Development (Швидка розробка додатків)

SPA – Single-Page Application (Односторінковий додаток)

SysML – Systems Modeling Language (Мова системного моделювання)

ВСТУП

У сучасному інформаційному суспільстві цифрові аудіотеки та подкасти набувають дедалі більшої популярності, стаючи невід'ємною частиною освітнього, розважального та інформаційного простору. Обсяги аудіоконтенту зростають експоненціально, створюючи виклик для його ефективного керування, пошуку та доступу. Існуючі платформи, хоча й надають широкий спектр можливостей, часто мають обмеження у функціоналі, що стосується детального аналізу змісту аудіо, зокрема для подкастів, а також у гнучкості керування персональними колекціями. Це зумовлює актуальність розробки інноваційних підходів до створення інформаційних систем, здатних задовольнити зростаючі вимоги користувачів.

Обґрунтування вибору теми дослідження. Вибір теми "Інформаційна система забезпечення роботи цифрової аудіотеки" зумовлений її високою соціальною значущістю та технологічною інноваційністю. На сучасному етапі розвитку інформаційних технологій, рішення для керування аудіоконтентом перебувають на високому рівні в плані базових функцій відтворення та каталогізації. Однак, попри загальний прогрес, існує потреба у вдосконаленні систем, що стосуються глибокої обробки аудіоматеріалів. Впровадження технологій штучного інтелекту, зокрема для автоматичної транскрипції мовлення, відкриває унікальні можливості для підвищення доступності інформації, що міститься в аудіозаписах, шляхом візуалізації текстового представлення аудіо. Це дозволяє користувачам отримувати текстовий опис подкастів, що є важливим для їхнього швидкого розуміння та аналізу, хоча й не передбачає безпосереднього пошуку по тексту. Також актуальною залишається потреба у гнучких системах, які дозволяють користувачам легко завантажувати власні аудіофайли та формувати індивідуалізовані бібліотеки, що не завжди повноцінно реалізовано в існуючих рішеннях. Таким чином, дане дослідження спрямоване на розробку нової наукової інформації та вирішення конкретних практичних завдань у сфері цифрових аудіотек.

Сучасний рівень розроблення даної проблеми. Проблема керування цифровим аудіоконтентом активно досліджується. Існують численні комерційні сервіси (наприклад, Spotify, Apple Music, Podcast Addict) та відкриті платформи, які пропонують широкі можливості для прослуховування та поширення аудіо. Досягнення у сфері обробки природної мови (NLP) та розпізнавання мовлення (ASR) дозволили створити окремі інструменти для транскрипції аудіо (наприклад, Google Cloud Speech-to-Text, Whisper від OpenAI). Проте, інтеграція цих передових технологій у комплексні, зручні для користувача системи цифрових аудіотек, які б поєднували можливості керування контентом, автоматичної транскрипції та персоналізованої бібліотеки, все ще перебуває на стадії активного розвитку та вимагає подальших досліджень.

Огляд літератури з теми дослідження. Для всебічного розуміння предметної галузі та вибору оптимальних підходів до проектування інформаційної системи цифрової аудіотеки було проаналізовано низку наукових праць та технічної документації. Фундаментальні аспекти розробки вимог до програмного забезпечення детально описані у класичній роботі Яна Соммервілла "Інженерія програмного забезпечення", яка всебічно висвітлює процеси збору, аналізу та специфікації вимог, що стало методологічною основою для формування розділу 2.1 даної роботи. Принципи архітектури програмного забезпечення та побудови масштабованих систем розкриті у праці Роберта К. Мартіна "Чиста архітектура: Посібник для майстра зі структури та дизайну програмного забезпечення", яка допомогла у розумінні патернів проектування та вибору технологічного стеку. Питання проектування користувацьких інтерфейсів та забезпечення зручності використання (User Experience – UX) були вивчені за працею Дональда Нормана "Дизайн повсякденних речей", що стало орієнтиром при розробці інтерфейсу користувача. Врахування цих принципів дозволило інтегрувати наданий прототип Figma у загальну концепцію системи.

Загалом, огляд літературних джерел показав, що попри значні досягнення у кожній з окремих галузей, комплексна інтеграція передового AI-функціоналу транскрипції та гнучких можливостей керування персональним аудіоконтентом

у рамках єдиної, зручної цифрової аудіотеки залишається недостатньо висвітленою та вимагає подальших практичних рішень.

Об'єктом дослідження є процес розробки та впровадження інформаційної системи для забезпечення роботи цифрової аудіотеки. Цей об'єкт охоплює всі етапи життєвого циклу програмного забезпечення, починаючи від аналізу вимог до тестування та потенційного впровадження готового продукту.

Предметом дослідження є методи та підходи до проектування інформаційних систем для керування цифровим аудіоконтентом, вимоги користувачів до функціональності та інтерфейсу аудіотеки, зокрема щодо інтеграції AI-модуля для транскрипції подкастів та забезпечення легкого доступу до завантаження аудіофайлів для будь-якого користувача, а також сучасні технології для розробки веб- та мобільних додатків для роботи з мультимедійними даними та реалізації функціоналу штучного інтелекту.

Метою кваліфікаційної бакалаврської роботи є створення інноваційної інформаційної системи забезпечення роботи цифрової аудіотеки, яка включатиме AI-модуль для транскрипції подкастів та надаватиме легкий доступ до завантаження аудіофайлів для будь-якого користувача.

Для вирішення поставлених завдань були використані наступні методи дослідження:

- Аналіз літературних джерел: для вивчення теоретичних основ та сучасного стану розробки інформаційних систем, технологій розпізнавання мовлення та дизайну інтерфейсів.
- Системний аналіз: для декомпозиції складної системи на компоненти, виявлення взаємозв'язків та визначення вимог.
- Структурно-логічний метод: для побудови ієрархії вимог та архітектури системи.
- Методи моделювання (з використанням SysML): для візуалізації вимог та архітектурних рішень у Enterprise Architect.
- Метод прототипування: для швидкого створення макетів інтерфейсу (на основі Figma) та апробації дизайнерських рішень.

Теоретична значущість роботи полягає в систематизації підходів до інтеграції технологій штучного інтелекту (транскрипції мовлення) у типові інформаційні системи керування аудіоконтентом. Результати дослідження можуть бути використані для подальшого вивчення та вдосконалення подібних гібридних систем.

Практичне значення отриманих результатів полягає у розробці функціонального прототипу інформаційної системи, яка може бути впроваджена для вирішення актуальних завдань у сфері цифрових аудіотек – від особистого використання до застосування в освітніх установах або медіа-компаніях для ефективного керування, пошуку та аналізу розмовного контенту. Розроблена система може стати основою для створення комерційного продукту або ж бути адаптована для спеціалізованих потреб.

Структура роботи. Кваліфікаційна бакалаврська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі наведено характеристику та аналіз предметної галузі цифрової аудіотеки. Другий розділ присвячено розробці вимог та моделюванню інформаційної системи. У третьому розділі описано проектування та реалізацію компонентів інформаційної системи.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Предметною галуззю даної дипломної роботи є динамічна сфера розробки та функціонування інформаційних систем, призначених для розповсюдження та споживання аудіоконтенту, зокрема подкастів. Особливий акцент робиться на підвищенні доступності цього контенту завдяки інтеграції технологій штучного інтелекту [4]. Ця галузь охоплює широкий спектр діяльності, включаючи ринок цифрових медіа, платформи для створення та дистрибуції контенту, а також застосування передових інформаційних технологій для оптимізації взаємодії користувачів з аудіоматеріалами [5]. Економічний зміст задач у цій предметній галузі полягає у створенні цінності для користувачів через надання доступу до різноманітного контенту, а для авторів – через надання платформи для його поширення та потенційної монетизації. Стрімкий розвиток інтернет-технологій та мобільних пристроїв сприяв значному зростанню популярності подкастів як зручного формату для отримання інформації, навчання та розваг. Подкастинг посідає все вагомніше місце в загальній структурі медіаспоживання, відкриваючи нові можливості як для творців контенту, так і для слухачів. Важливою особливістю є висока динамічність ринку, значна конкуренція та постійна потреба в інноваційних рішеннях для залучення та утримання аудиторії [19].

Об'єктом дослідження в даній роботі виступає процес проектування, розробки та функціонування інформаційної системи, орієнтованої на прослуховування подкастів. Конкретніше, об'єктом є розроблювана система, що складається з вебсайту та відповідної бази даних. Ця система вирізняється наявністю кількох ключових компонентів, таких як модуль AI-транскрипції подкастів, що має на меті підвищити доступність контенту для широкого кола користувачів, включаючи осіб з вадами слуху, та покращити можливості пошуку по контенту. Також система включає функціонал для створення та управління власними плейлистами користувачів, систему авторизації та реєстрації для

слухачів та авторів, та спеціальний інтерфейс для завантаження подкастів авторами контенту.

Особливості предметної галузі суттєво впливають на організацію інформаційної системи та вибір технологій для розв'язання поставлених задач. Наприклад, великі обсяги аудіофайлів подкастів вимагають ефективних рішень для їх зберігання, передачі та обробки. Різноманітність контенту, що охоплює широкий спектр тем, жанрів та мов, створює виклики для його категоризації, пошуку та рекомендації. Надзвичайно важливим є користувацький досвід (UX), тому інтерфейс системи має бути інтуїтивно зрозумілим, забезпечувати зручну навігацію, якісне відтворення аудіо та легкий доступ до всіх функцій, включаючи транскрипції та плейлисти [7]. Забезпечення доступності контенту для людей з обмеженими можливостями є важливим соціальним та технічним завданням, і AI-транскрипція є одним із ключових інструментів для досягнення цієї мети. Система повинна задовольняти потреби двох основних типів користувачів: слухачів (споживачів контенту) та авторів (творців контенту), надаючи їм відповідні інструменти та функціонал. Крім того, модуль AI-транскрипції потребує застосування складних алгоритмів обробки природної мови для забезпечення високої точності перетворення аудіо в текст.

Організаційна структура функціонування системи передбачає взаємодію кількох ключових ролей. Слухач (кінцевий користувач) може реєструватися, авторизуватися, шукати та прослуховувати подкасти, переглядати AI-транскрипції, а також створювати та управляти власними плейлистами. Автор подкасту (контент-мейкер) також проходить реєстрацію та авторизацію, після чого може завантажувати аудіофайли подкастів, додавати метадані (назву, опис, категорію) та управляти власним контентом [9]. Модуль AI-транскрипції відповідає за автоматичне перетворення завантажених аудіофайлів у текстовий формат, забезпечуючи точність та швидкість цього процесу [6]. Умовна роль Адміністратора системи передбачає управління базою даних, моніторинг роботи AI-модуля, управління користувачами (за потреби), оновлення системи та забезпечення її безпеки. Для наочності, взаємодію цих ролей та компонентів

системи (веб-інтерфейс, база даних, AI-модуль) доцільно представити у вигляді схеми.

У рамках роботи розглядаються такі об'єкти предметної галузі, як сам подкаст (що включає аудіофайл, метадані, епізоди), окремі епізоди подкасту, аудіофайли, текстові транскрипції, користувачі (слухачі та автори) та їхні облікові записи, плейлисти, категорії або теги подкастів. Також важливими об'єктами є база даних системи, її веб-інтерфейс та модуль AI-транскрипції.

Рішення, що приймаються в предметній галузі в рамках функціонування системи, є різноманітними [1]. Слухачі приймають рішення щодо вибору подкасту для прослуховування, використання транскрипції, додавання подкасту до плейлиста та реєстрації. Автори вирішують, коли завантажувати новий епізод та які метадані використовувати для його опису. Сама система, зокрема її AI-модуль, обробляє аудіо та генерує транскрипції, обробляє запити користувачів до бази даних та управляє сесіями. Адміністратор приймає рішення щодо оновлення програмного забезпечення, можливої модерації контенту та аналізу використання системи .

На середовище прийняття рішень впливають різні фактори. Для слухачів це особисті інтереси, рекомендації, популярність подкастів, якість звуку та транскрипції, а також зручність інтерфейсу. Для авторів важливі популярність платформи, зручність завантаження контенту та наявність аудиторії. Для системи критичними є доступні обчислювальні ресурси для AI-транскрипції, якість аудіозаписів та навантаження на сервер.

Існує тісний взаємозв'язок між різними рішеннями [9]. Наприклад, рішення користувача прослухати певний подкаст може вплинути на майбутні рекомендації, якщо такий функціонал буде реалізовано. Якість транскрипції, що генерується AI-модулем, безпосередньо впливає на задоволеність користувачів, які потребують цієї функції, і, відповідно, на їх рішення продовжувати користуватися системою. Регулярна публікація якісного контенту авторами впливає на привабливість платформи для слухачів.

Для прийняття обґрунтованих рішень необхідна різноманітна інформація. Внутрішня інформація включає дані з бази даних (профілі користувачів, історія прослуховувань, плейлисти, метадані подкастів, згенеровані транскрипції), логи роботи системи та статистику використання AI-модуля. Зовнішня інформація охоплює загальні тренди подкастингу, відгуки користувачів та нові технології в галузі AI-транскрипції. Дані можуть бути структурованими (в базі даних), неструктурованими (аудіофайли) або напівструктурованими (текстові транскрипції, логи). Джерелами надходження інформації є введення даних користувачами (реєстрація, завантаження подкастів, створення плейлистів) та автоматична генерація даних системою (транскрипції, логи) [8]. Методології збирання та оброблення включають зберігання даних у реляційній базі даних, обробку аудіо за допомогою AI-алгоритмів та аналіз логів для моніторингу.

Підсумовуючи, ситуації прийняття рішень у даній предметній галузі переважно є відкритими задачами для користувачів, особливо щодо вибору контенту. Для AI-модуля транскрипції ключовим завданням є забезпечення максимальної точності в умовах різної якості аудіо та мовленнєвих особливостей. Впровадження AI-транскрипції створює нові інформаційні потоки у вигляді текстових даних, що розширюють можливості користувачів та системи, наприклад, дозволяючи повнотекстовий пошук по аудіо. Рішення, що приймаються адміністратором, часто стосуються оптимізації ресурсів та покращення функціональності системи [6].

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Аналіз існуючих інформаційних систем (ІС) та технологій у сфері розповсюдження та споживання подкастів є ключовим етапом для обґрунтування проєктування нової системи. Цей аналіз дозволяє виявити поточні тенденції, успішні практики, технологічні рішення, а також потенційні прогалини та можливості для вдосконалення, що є особливо актуальним для системи з інтегрованим модулем AI-транскрипції.

Огляд існуючих платформ та інформаційних систем для подкастингу

Ринок подкастингу представлений значною кількістю платформ, які можна умовно поділити на декілька категорій. По-перше, це великі агрегатори та стрімінгові сервіси, такі як Spotify, Apple Podcasts та Google Podcasts. Ці платформи надають доступ до величезної кількості подкастів, пропонують функції пошуку, підписки, завантаження для офлайн-прослуховування та, все частіше, базові рекомендаційні системи. Їхня сила полягає у великій аудиторії та зручності для кінцевого споживача. Однак, можливості кастомізації та інтеграції специфічних функцій, таких як глибока AI-транскрипція з розширеними можливостями, можуть бути обмежені [19].

По-друге, існують платформи, орієнтовані на авторів подкастів, наприклад, Anchor.fm (від Spotify), Buzzsprout, Libsyn, Podbean. Ці сервіси надають інструменти для запису, редагування, хостингу аудіофайлів, розповсюдження подкастів на основні платформи-агрегатори, а також аналітику щодо прослуховувань. Деякі з них починають інтегрувати базові інструменти транскрипції, але часто це або платна послуга, або функціонал з обмеженою точністю.

По-третє, з'являються нішеві платформи та додатки, що намагаються запропонувати унікальний користувацький досвід або специфічні функції. Це можуть бути платформи, сфокусовані на освітніх подкастах, аудіодрамах або системах з розширеними соціальними функціями. Саме в цій категорії часто з'являються інноваційні підходи до взаємодії з контентом.

Технології, що використовуються в сучасних подкаст-системах

Функціонування сучасних подкаст-платформ забезпечується комплексом технологій. Для зберігання та доставки аудіоконтенту широко використовуються хмарні сховища (наприклад, Amazon S3, Google Cloud Storage) та мережі доставки контенту (CDN), що забезпечують швидке завантаження файлів для користувачів у різних географічних регіонах. Аудіофайли зазвичай кодується у форматах MP3 або AAC [20].

Серверна частина (backend) таких систем часто розробляється з використанням популярних фреймворків та мов програмування, таких як Python (Django, Flask), Node.js (Express), Ruby (Ruby on Rails) або Java (Spring). Вони відповідають за бізнес-логіку, управління користувачами, обробку запитів, взаємодію з базою даних та інтеграцію зі сторонніми сервісами.

Клієнтська частина (frontend) реалізується у вигляді вебсайтів (з використанням HTML, CSS, JavaScript та фреймворків типу React, Angular, Vue.js) та мобільних додатків (нативних для iOS/Android або кросплатформених). Ключовими аспектами є забезпечення зручного інтерфейсу (UI) та позитивного користувацького досвіду (UX) [30].

Бази даних відіграють важливу роль у зберіганні метаданих подкастів (назви, описи, автори, епізоди), інформації про користувачів (профілі, підписки, плейлисти), а також згенерованих транскрипцій. Залежно від архітектури та навантаження, можуть використовуватися як реляційні СУБД (наприклад, PostgreSQL, MySQL), так і NoSQL рішення (наприклад, MongoDB для гнучкої схеми даних або Elasticsearch для повнотекстового пошуку) [24].

Технології штучного інтелекту, зокрема автоматичне розпізнавання мовлення (Automatic Speech Recognition, ASR), знаходять все ширше застосування у подкастингу. Основна мета – генерація текстових транскрипцій аудіоконтенту. Це значно підвищує доступність подкастів для людей з вадами слуху, покращує **SEO-оптимізацію** (пошукові системи можуть індексувати текстовий контент), дозволяє здійснювати **повнотекстовий пошук** всередині аудіозаписів та відкриває можливості для **аналізу контенту** і створення нових сервісів (наприклад, автоматичне створення коротких описів, виділення ключових тем) [19].

Існує низка комерційних та відкритих рішень для AI-транскрипції, таких як Google Cloud Speech-to-Text, AWS Transcribe, AssemblyAI, OpenAI Whisper, Descript, Otter.ai. Ці сервіси різняться за точністю, підтримкою мов, швидкістю обробки та вартістю. Точність транскрипції залежить від багатьох факторів: якості аудіозапису, наявності фонових шумів, кількості спікерів, їхніх акцентів

та чіткості мовлення. Незважаючи на значний прогрес, автоматично згенеровані транскрипції часто потребують ручної коректури для досягнення високої якості.

Практичний досвід показує, що інтеграція AI-транскрипції безпосередньо в подкаст-платформу, як це планується в даній роботі, може надати значні переваги користувачам, усуваючи необхідність використання сторонніх сервісів. Важливим аспектом є можливість синхронізації тексту з аудіо, що дозволяє користувачам легко слідкувати за мовленням [6].

Аналіз існуючих рішень показує, що хоча багато платформ пропонують широкий набір базових функцій, існують аспекти, які можна покращити.

Інтеграція та доступність AI-транскрипції: Не всі платформи пропонують вбудовану транскрипцію, а ті, що пропонують, можуть мати високу вартість або обмежену точність. Забезпечення якісної та доступної транскрипції є важливою нішею [10].

- **Розширені функції плейлистів:** Хоча створення плейлистів є стандартною функцією, можливості для їх кастомізації, спільного використання або інтелектуального формування часто обмежені.
- **Інструменти для авторів:** Полегшення процесу завантаження та управління контентом, а також надання детальної аналітики та інструментів для взаємодії з аудиторією завжди є актуальним.
- **Користувацький досвід:** Постійний пошук шляхів покращення навігації, персоналізації та загальної зручності використання системи як для слухачів, так і для авторів.

Літературні джерела, такі як дослідження ринку подкастингу та академічні статті з обробки природної мови та машинного навчання, підтверджують зростаючий попит на аудіоконтент та важливість технологій AI для його обробки та аналізу. Практичний досвід використання популярних подкаст-додатків свідчить про те, що користувачі цінують швидкість, надійність, якість контенту та зручність інтерфейсу [3].

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної системи/підсистеми

Розробка будь-якої інформаційної системи вимагає всебічного аналізу вимог, що є фундаментальним етапом для успішної реалізації проекту. Цей підрозділ присвячений збору, аналізу, класифікації та деталізації всіх функціональних і нефункціональних вимог, які визначатимуть поведінку та якісні характеристики інформаційної системи забезпечення роботи цифрової аудіотеки. Чітке формулювання цих вимог дозволить ефективно спроектувати систему, яка відповідатиме потребам користувачів та специфіці роботи з аудіоконтентом, включаючи функціонал штучного інтелекту для транскрипції та легкість завантаження файлів.

2.1.1 Функціональні вимоги

Функціональні вимоги описують конкретні дії та можливості, які система повинна надавати. Для інформаційної системи цифрової аудіотеки з AI-модулем для транскрипції подкастів та функціоналом завантаження аудіофайлів, до ключових функціональних аспектів належать:

Система повинна забезпечувати повноцінне управління користувачами, включаючи їхню реєстрацію та авторизацію (з використанням електронної пошти/пароля або інтеграцією із зовнішніми сервісами). Кожен користувач матиме змогу керувати своїм профілем, оновлювати особисті дані та змінювати пароль, а також використовувати функцію його відновлення. Також передбачається управління різними ролями користувачів, наприклад, адміністратор та звичайний користувач, що дозволить розмежувати доступ до певних функцій.

Центральне місце в системі займає управління аудіоконтентом. Користувачі матимуть можливість легко завантажувати різноманітні аудіофайли, такі як подкасти, музика чи аудіокниги, у поширених форматах

(MP3, WAV). Система автоматично перевірятиме та оброблятиме завантажені файли, забезпечуючи їхнє надійне та масштабоване зберігання. Для кожного аудіофайлу передбачено додавання, редагування та видалення метаданих, таких як назва, автор, опис, жанр або дата завантаження, що дозволить ефективно каталогізувати контент. Реалізація розширеного пошуку за метаданими та фільтрації за різними критеріями (автор, жанр) стане ключовою для зручності знаходження потрібного контенту. Користувачі зможуть відтворювати аудіофайли безпосередньо в системі, контролюючи процес відтворення (пауза, перемотування, регулювання гучності).

Особливим нововведенням системи є інтеграція AI-модуля транскрипції. Цей модуль дозволить автоматично перетворювати аудіомовлення, особливо подкастів, у текстовий формат.

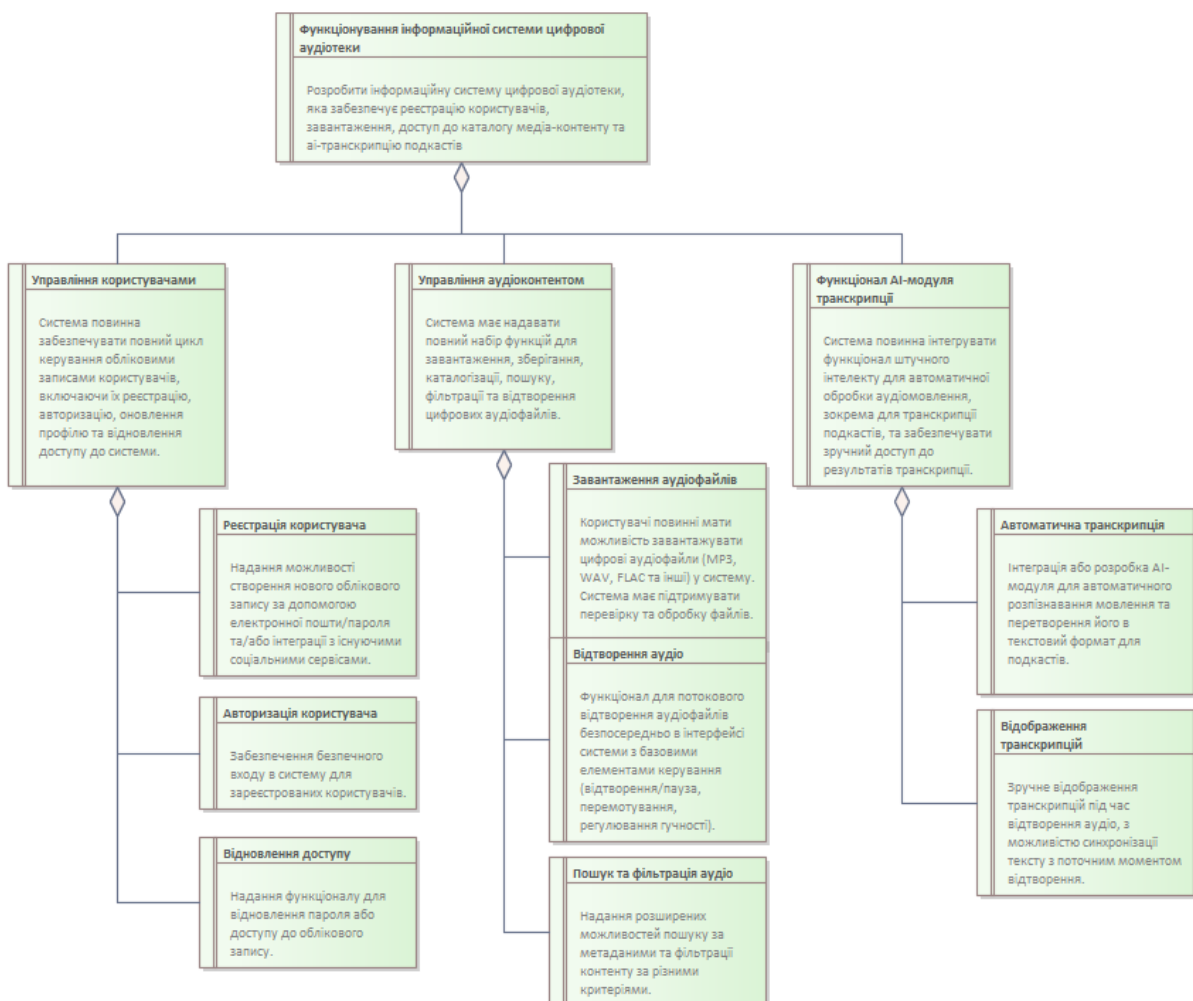


Рисунок 2.1 – Діаграма функціональних вимог

Джерело: сформовано автором на основі виконаного дослідження

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги визначають якісні характеристики системи, її надійність, продуктивність та зручність, що є критично важливим для успішного функціонування.

У частині **продуктивності** система повинна демонструвати швидкий відгук на дії користувача, забезпечувати оптимальний час завантаження аудіофайлів та ефективну генерацію транскрипцій. Масштабованість системи є обов'язковою умовою для її стабільної роботи при зростанні кількості користувачів та обсягів контенту без втрати продуктивності.

Надійність передбачає високу доступність системи (24/7, за винятком планових робіт), стійкість до відмов та можливість швидкого відновлення після збоїв з мінімальними втратами даних.

Зручність використання (Usability) вимагає створення інтуїтивно зрозумілого, простого та легкого у навігації інтерфейсу користувача, який відповідатиме візуальному прототипу, розробленому у Figma. Також важливо забезпечити доступність системи для користувачів з різними потребами.

Супровідність системи забезпечуватиметься її модульною архітектурою, що сприятиме легшій модифікації, розширенню функціоналу та виправленню помилок. Наявність чіткої та актуальної технічної документації також є важливою для ефективного супроводу.

Таким чином, специфікація вимог до інформаційної системи цифрової аудіотеки є комплексною основою для подальшого проектування та реалізації, яка враховує як функціональні можливості системи, так і її якісні характеристики для забезпечення оптимального користувацького досвіду.

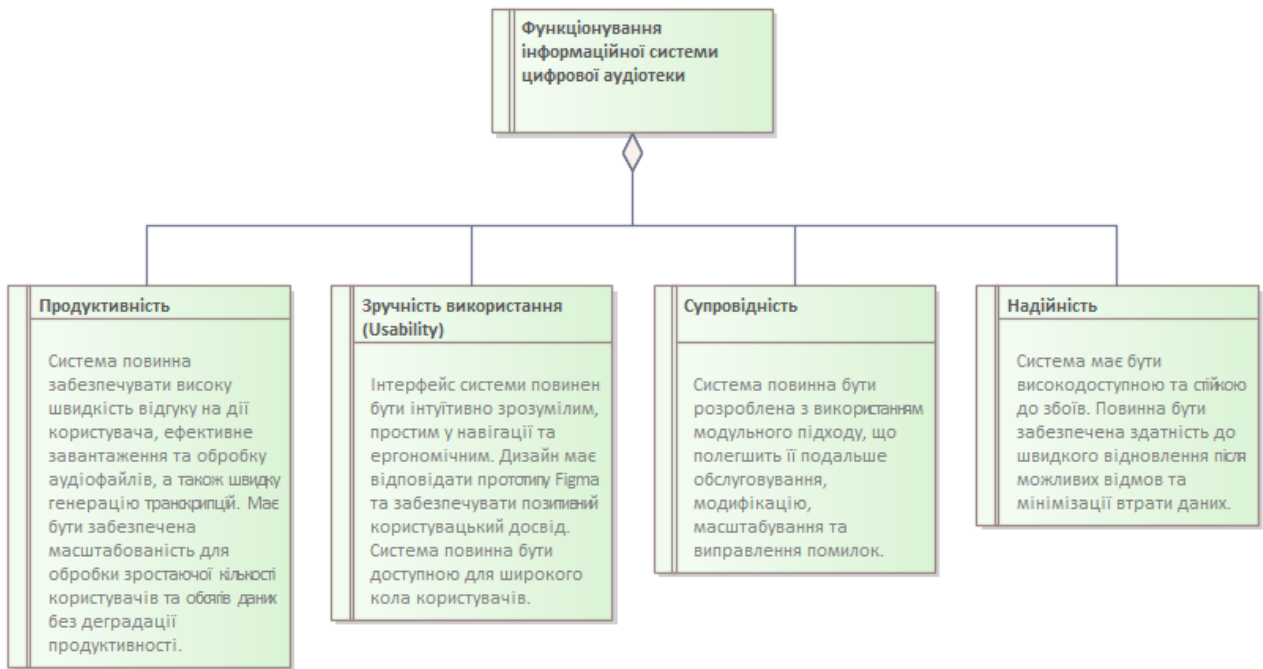


Рисунок 2.2 - Діаграма нефункціональних вимог

Джерело: сформовано автором на основі виконаного дослідження

2.1.3 Специфікація вимог до інформаційної підсистеми

Після аналізу та класифікації вимог, наступним кроком є їх детальна специфікація. Цей етап передбачає формалізований опис кожної вимоги з використанням уніфікованої структури, що забезпечує їхню однозначність, простежуваність та можливість ефективного керування ними протягом життєвого циклу проекту. Деталізовані специфікації вимог дозволяють уникнути неоднозначності в розумінні функціоналу та якісних характеристик системи як розробниками, так і зацікавленими сторонами.

Результати специфікації вимог наведено у **Додатку А**. Специфікація функціональних вимог. Кожна вимога представлена у табличному форматі, що відповідає стандарту SysML та забезпечує наявність таких відомостей:

- **Ідентифікатор вимоги:** Унікальний код, що дозволяє однозначно ідентифікувати кожну вимогу в системі (наприклад, FR-001, FR-001.1, NFR-001).
- **Стисла назва вимоги:** Короткий, лаконічний опис сутності вимоги, що дозволяє швидко орієнтуватися.

- **Повна назва вимоги:** Розгорнуте формулювання вимоги, що дає більш повне уявлення про її зміст.
- **Тип:** Класифікація вимоги (функціональна або нефункціональна)
- **Джерело:** Зазначення походження вимоги (наприклад, вимоги користувачів, аналіз конкурентів, нормативні документи, прототип Figma).
- **Статус:** Поточний стан вимоги у процесі розробки (наприклад, "Запропоновано", "Узгоджено", "Реалізується", "Виконано", "Відкладено").
- **Складність:** Оцінка трудомісткості реалізації вимоги (наприклад, "Низька", "Середня", "Висока").
- **Пріоритетність:** Важливість вимоги для успішної реалізації проекту (наприклад, "Критична", "Висока", "Середня", "Низька").
- **Ризикованість:** Оцінка потенційних ризиків, пов'язаних з реалізацією даної вимоги (наприклад, "Висока" – через технічні труднощі або залежності).
- **Примітки:** Детальне пояснення суті вимоги, уточнення її поведінки, обмежень, критеріїв прийняття та будь-яких додаткових деталей, що необхідні для її повного розуміння та реалізації.

2.2 Постановка та алгоритм розв'язання задачі

2.2.1 Характеристика задачі

Інформаційна система призначена для забезпечення централізованого зберігання, керування, обробки, завантаження та доступу до цифрового аудіоконтенту для користувачів. Її ключове призначення – створити зручне та функціональне середовище для взаємодії з подкастами, аудіокнигами та іншими аудіозаписами, надаючи можливість персоналізації та легкого доступу. Особливістю системи є інтеграція AI-модуля для автоматичної транскрипції подкастів, що підвищує цінність аудіоматеріалів. Техніко-економічна сутність задачі полягає в автоматизації процесів, які наразі є трудомісткими або взагалі

неможливими без спеціалізованого програмного забезпечення. Ручна транскрипція аудіо, наприклад, є вкрай дорогою та неефективною. Розв'язання даної задачі за допомогою електронно-обчислювальних машин (ЕОМ) є єдиною можливим та економічно доцільним способом, оскільки обробка великих обсягів аудіоданих, виконання складних алгоритмів розпізнавання мовлення, забезпечення одночасного доступу для багатьох користувачів, а також ефективно зберігання та пошук інформації вимагають потужностей та швидкості, які може надати лише програмно-апаратний комплекс.

Задача розв'язується через управління такими об'єктами, як цифрові аудіофайли (подкасти, аудіокниги), їхні метадані (назва, автор, опис, жанр, тривалість, дата завантаження), текстові транскрипції, згенеровані AI-модулем, а також облікові записи користувачів та їхні профілі разом з їхніми персональними бібліотеками подкастів.

Вихідна інформація системи включає безпосередньо відтворений аудіоконтент, текстові транскрипції подкастів, що відображаються для ознайомлення, а також структуровані списки аудіофайлів, персональних бібліотек та результати пошуку чи фільтрації. Користувачі також отримують доступ до даних свого профілю та різні системні повідомлення. Ця інформація використовується користувачами для прямої взаємодії з системою: прослуховування контенту, швидкого ознайомлення з його змістом через транскрипції, організації власних колекцій, пошуку необхідних матеріалів та отримання зворотного зв'язку від системи.

Задача розв'язується в режимі реального часу, тобто за запитом користувача, за винятком процесу транскрипції аудіо. Цей процес може бути асинхронним і займати певний час залежно від обсягу аудіо та поточного навантаження на систему. Відображення списків аудіо, бібліотек та профілю відбувається практично миттєво, а запуск відтворення аудіо залежить від швидкості мережі. Відображення ж транскрипції здійснюється після завершення її обробки, про що користувач отримує відповідне сповіщення.

Автоматизоване розв'язання задачі може бути припинено за умови зупинки роботи системи (планової або аварійної), відсутності з'єднання з Інтернетом (як з боку користувача, так і сервера), вичерпання дискового простору для зберігання даних, виникнення критичних програмних помилок або припинення підтримки зовнішніх AI-сервісів, якщо вони інтегровані.

Дана задача має зв'язки з іншими системами та модулями. Модуль управління користувачами потенційно може взаємодіяти із зовнішніми сервісами автентифікації. Модуль AI-транскрипції може використовувати зовнішні API для розпізнавання мовлення або інтегрувати локально розгорнуті моделі. Підсистема зберігання файлів може взаємодіяти з хмарними сховищами, а загальна робота системи контролюється модулями моніторингу та логування.

Розподіл дій між персоналом і технічними засобами визначається ситуацією. У нормальному режимі функціонування переважна більшість дій виконується автоматизовано технічними засобами на запит користувача, який взаємодіє лише з інтерфейсом системи. У випадку виникнення помилок, система інформує користувача про локальні проблеми, тоді як системні збої або вичерпання ресурсів вимагають втручання адміністратора, який аналізує логи та виконує відновлювальні дії. Автоматизоване розв'язання може бути тимчасово призупинене для планового обслуговування (оновлення, резервне копіювання, оптимізація), про що користувачі сповіщаються заздалегідь.

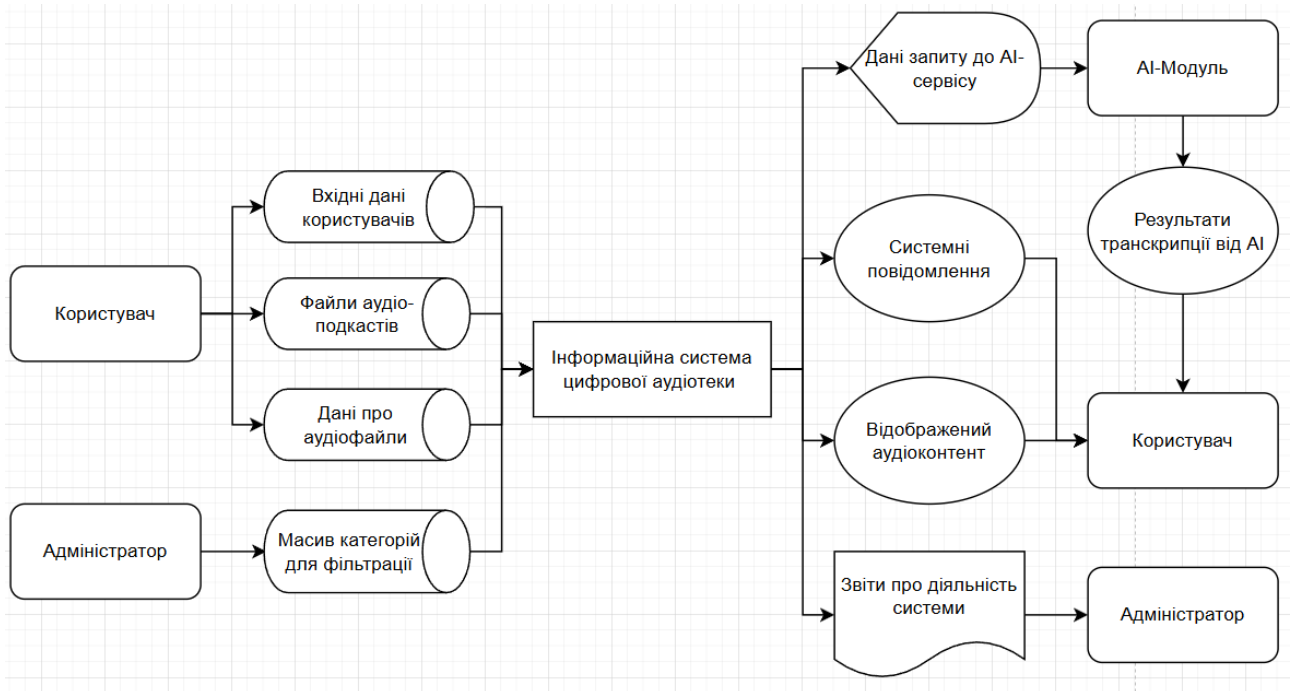


Рисунок 2.3 - Інформаційна модель задачі

Джерело: сформовано автором на основі виконаного дослідження

Вихідна інформація

Вихідна інформація інформаційної системи забезпечення роботи цифрової аудіотеки формується та надається користувачам та адміністратору для забезпечення функціональних можливостей системи та досягнення поставленої мети. Її основним призначенням є представлення користувачеві результатів обробки інформації, забезпечення доступу до аудіоконтенту та супутніх даних (транскрипцій).

Вихідні повідомлення використовуються для:

Відображення та взаємодії: Користувачі отримують списки аудіофайлів, результати пошуку, текстові транскрипції та інші метадані безпосередньо через інтерфейс системи (відеограми). Це дозволяє їм переглядати, обирати та споживати контент, а також отримувати зворотний зв'язок про виконані дії.

Інформування: Системні повідомлення інформують користувачів про успішне виконання операцій, потенційні проблеми або помилки.

Адміністрування та аналіз: Для адміністраторів системи формуються машинограми у вигляді звітів, які містять агреговані дані про діяльність системи (кількість користувачів, обсяг завантаженого контенту, статистика використання

функціоналу тощо). Ці звіти допомагають в оцінці ефективності роботи системи, плануванні ресурсів та прийнятті управлінських рішень.

Вихідні повідомлення представляють собою інформаційні одиниці різного формату, що генеруються системою автоматично під час її роботи і можуть бути як тимчасовими (для відображення на екрані), так і постійними (для зберігання в архівах).

Таблиця 2.1 – Вихідна інформація

№ з/п	Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність видання	Термін і допустимий час затримки	Користувач і інформації
1	Відображений аудіоконтент	OA_01	Відеограма (графічний інтерфейс користувача: списки, картки ауді)	За запитом користувача	До 2-5 секунд (залежить від мережі)	Користувач ІСЦА
2	Результати транскрипції від AI	OA_02	Відеограма (текстове поле у веб-інтерфейсі)	За запитом користувача	До 1-2 секунд (після готовності)	Користувач ІСЦА
3	Системні повідомлення	OA_03	Відеограма (спливаючі вікна)	Подія	Миттєво	Користувач ІСЦА, Адміністратор
4	Звіти про діяльність системи	OA_04	Машинограма (файли PDF, Excel)	За запитом адміністратора (періодично)	До 5-10 секунд (залежить від обсягу даних)	Адміністратор ІСЦА

Джерело: сформовано автором на основі виконаного дослідження

Вхідна інформація

Вхідна інформація для інформаційної системи забезпечення роботи цифрової аудіотеки є сукупністю даних, що надходять до системи з зовнішніх джерел і є необхідними для її функціонування, наповнення та оновлення інформаційних масивів. Призначення вхідної інформації полягає у забезпеченні системи всіма необхідними даними для здійснення таких операцій, як реєстрація користувачів, завантаження аудіофайлів та їх метаданих, керування категоріями та авторами, а також для ініціювання процесів обробки (наприклад, транскрипції).

Способи одержання вхідної інформації включають:

Безпосередній ввід користувачем: Через інтерактивний веб-інтерфейс або мобільний додаток, де користувач вводить дані (наприклад, реєстраційні дані, запити на пошук, метадані аудіофайлів).

Завантаження файлів: Користувачі можуть завантажувати аудіофайли безпосередньо в систему, яка потім їх обробляє та зберігає.

Обмін даними із зовнішніми сервісами: Система може отримувати результати обробки від зовнішніх інтегрованих сервісів (наприклад, згенеровані текстові транскрипції від AI-модуля розпізнавання мовлення).

Адміністративний ввід/імпорт: Адміністратори можуть вводити або імпортувати довідкові дані (наприклад, про авторів або категорії), які є важливими для структурування та каталогізації контенту.

Вхідна інформація є первинним джерелом даних, що формують базу знань системи та ініціюють її функціональні процеси. Вона надходить з різною періодичністю, залежно від активності користувачів та зовнішніх взаємодій, і використовується для оновлення оперативних файлів бази даних та довідкових даних.

Таблиця 2.2 – Вхідна інформація

№ з/п	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1	Вхідні дані користувачів	IA_01	Відеограма (форми вводу)	За запитом (одноразово/періодично)	Користувач
2	Файли аудіоподкастів	IA_02	Файл (MP3, WAV, FLAC), Відеограма (форма завантаження)	За запитом користувача	Користувач
3	Дані про аудіофайли	IA_03	Відеограма (поля вводу, кнопки)	Подія (часто, залежить від активності)	Користувач
4	Масив категорій для фільтрації	IA_04	Відеограма (форми вводу), Машинограма (імпорт)	За запитом (періодично)	Адміністратор

Джерело: сформовано автором на основі виконаного дослідження

2.2.2 Алгоритм розв'язання задачі

Серед використовуваної інформації системи є дані про зареєстрованих Користувачів (логін, хеш пароля, ім'я, дата реєстрації), Аудіофайли (метадані та посилання на бінарні файли, назва, опис, тривалість, шлях до файлу, дата завантаження, посилання на автора та користувача) та їх Транскрипції (текстові транскрипції, пов'язані з аудіофайлом, дата генерації, статус). Довідкова інформація включає Категорії (перелік доступних назв категорій для класифікації) та Авторів (інформація про творців аудіоконтенту, їх ім'я та опис). Для забезпечення зв'язку між аудіофайлами та категоріями використовується Зв'язок Аудіо-Категорія. Також система працює з тимчасовими даними про Сесії користувачів, що містять інформацію про активність та ідентифікатори користувачів.

Таблиця 2.3 - Перелік масивів використовуваної інформації

№ з/п	Масив	Ідентифікатор	Максимальна кількість записів
1	Користувачі	MASS_USR	Десятки тисяч (напр., 50 000)
2	Аудіофайли	MASS_AUD	Сотні тисяч (напр., 200 000)
3	Транскрипції	MASS_TRN	Сотні тисяч (напр., 200 000)
4	Категорії	MASS_CAT	Сотні (напр., 200)
5	Автори	MASS_AUT	Тисячі (напр., 5 000)
6	Зв'язок Аудіо-Категорія	MASS_AUD_CAT	Мільйони (напр., 1 000 000)
7	Сесії користувачів (тимчасовий)	MASS_SESS	Тисячі (напр., 10 000 одночасно)

Джерело: сформовано автором на основі виконаного дослідження

Результати розв'язання задачі в інформаційній системі цифрової аудіотеки представляють собою інформацію, яка формується в процесі обробки даних та призначена для подальшого використання – або для відображення користувачам (вихідні повідомлення), або для зберігання в базі даних з метою подальшого аналізу та забезпечення функціональності системи.

До результатів розв'язання належать Відображувані_Аудіо_Дані, що формуються для представлення списків аудіофайлів та детальної інформації в інтерфейсі користувача. Згенеровані_Транскрипції_Текст містять готовий текст

транскрипції для відображення. Система також генерує Системні_Повідомлення_Для_Користувача, які інформують про статус операцій. Для адміністративного контролю формуються Звіти_Адміністратора. Крім того, система створює та оновлює Індокси_Пошуку для прискорення доступу до даних та Логи_Дій_Користувачів для моніторингу активності.

Таблиця 2.4 - Перелік масивів результатної інформації

№ з/п	Масив	Ідентифікатор	Максимальна кількість записів
1	Відображувані_Аудіо_Дані	RES_AUD_DISP	Динамічно (залежить від запиту)
2	Згенеровані_Транскрипції_Текст	RES_TRN_TXT	Динамічно (залежить від запиту)
3	Системні_Повідомлення_Для_Користувача	RES_SYS_MSG	Динамічно (залежить від події)
4	Звіти_Адміністратора	RES_ADM_REP	Залежить від кількості звітів
5	Індокси_Пошуку	RES_SRCH_IND	Мільйони (напр., 5 000 000)
6	Логи_Дій_Користувачів	RES_LOG_ACT	Мільйони (постійно зростає)

Джерело: сформовано автором на основі виконаного дослідження

Математична модель зберігання та зв'язків даних:

Інформаційна модель системи ґрунтується на реляційній моделі даних, яка описується за допомогою множин та зв'язків між ними.

- **Множина користувачів (U):** $U = \{u_1, u_2, \dots, u_n\}$ Кожен користувач u_i характеризується набором атрибутів: $u_i = (\text{id_користувача}, \text{ім'я}, \text{email}, \text{пароль_хеш}, \text{дата_реєстрації})$
- **Множина аудіофайлів (A):** $A = \{a_1, a_2, \dots, a_m\}$ Кожен аудіофайл a_j характеризується: $a_j = (\text{id_файлу}, \text{назва}, \text{опис}, \text{шлях_до_файлу}, \text{тривалість}, \text{дата_завантаження}, \text{id_користувача}, \text{id_автора})$

Зв'язки між множинами (реляції):

- **Користувач - Аудіофайл (завантажує):** Реляція $RU \rightarrow A \subseteq U \times A$, де $RU \rightarrow A$ (u_i, a_j) істинна, якщо користувач u_i завантажив аудіофайл a_j . Визначається через $a_j.id_користувача = u_i.id_користувача$.

Математична модель пошуку:

Пошук аудіофайлів здійснюється на основі відповідності запиту користувача метаданим аудіофайлів та їх транскрипціям.

Пошук за ключовим словом (K): Нехай K – множина ключових слів, введених користувачем. Функція пошуку $Search(K) = \{a_j \in A \mid \exists k \in K: (k \in a_j.name) \vee (k \in a_j.description) \vee (k \in a_j.author.name_автора)\}$.

Опис процесу:

Математична модель відображає структуру даних та їх взаємозв'язки. Основні процеси системи – реєстрація, завантаження, транскрибування, пошук, відтворення – базуються на операціях над цими множинами (додавання, видалення, оновлення елементів, вибірка за критеріями). Точність обчислень для числових показників (наприклад, тривалості) забезпечується використанням стандартних типів даних. Відповідність моделі реальному процесу досягається за рахунок гнучкої структури, що дозволяє додавати нові атрибути та розширювати зв'язки у майбутньому. Допущення: унікальність ідентифікаторів, цілісність даних (посилання на існуючі об'єкти), наявність одного автора для одного файлу.

Опис логіки алгоритму та послідовність етапів розрахунку:

Загальний алгоритм функціонування системи розпочинається з її ініціалізації та протікає через низку послідовних етапів, що забезпечують взаємодію з користувачем та обробку даних.

1. Початок функціонування системи:

Система ініціалізує всі необхідні компоненти, встановлює підключення до бази даних та готується до прийому та обробки запитів від користувачів та зовнішніх сервісів.

2. Авторизація / Реєстрація користувача:

На цьому етапі користувач вводить свої облікові дані для входу в систему або для створення нового облікового запису. Система обробляє отримані дані, виконуючи перевірку наявності та коректності облікових даних для авторизації або перевіряючи унікальність наданої інформації для реєстрації. Залежно від результату перевірки, система формує та відображає користувачеві відповідне повідомлення: про успішну авторизацію/реєстрацію або про помилку (наприклад, невірний логін/пароль, вже існуючий email). У випадку невдалої авторизації/реєстрації, алгоритм передбачає повернення до початкового етапу "Авторизація / Реєстрація користувача", дозволяючи користувачеві повторити спробу.

3. Вибір дії користувача (Основний цикл роботи системи):

Після успішної авторизації користувач отримує доступ до головного інтерфейсу системи, де він може обрати одну з доступних функціональних дій. Цей етап є центральним циклом роботи системи, що забезпечує постійну взаємодію з активним користувачем.

• Сценарій 1: Завантаження файлів:

Користувач ініціює завантаження аудіофайлу та вводить супутні метадані (назву, опис, автора, категорії). Система здійснює збереження бінарних даних аудіофайлу у файловому сховищі та відповідних метаданих у базу даних (масиви MASS_AUD, MASS_AUT, MASS_CAT). Автоматично ініціюється процес транскрипції: завантажений аудіофайл відправляється до зовнішнього AI-сервісу для розпізнавання мовлення. Користувач отримує системне повідомлення, що інформує про статус завантаження та початок обробки файлу. Після завершення цієї дії, управління повертається до етапу "Вибір дії користувача".

• Сценарій 2: Пошук та фільтрація:

Користувач вводить бажані параметри для пошуку (ключові слова) або фільтрації (наприклад, за категоріями, авторами). Система виконує запит до бази даних (масиви MASS_AUD, MASS_TRN, MASS_CAT, MASS_AUT), використовуючи оптимізовані пошукові індекси (RES_SRCH_IND) для підвищення швидкодії. Результати пошуку та фільтрації формуються та

відображаються користувачеві у зручному вигляді. Після цього, управління повертається до етапу "Вибір дії користувача".

- **Сценарій 3: Перегляд транскрипції:**

Користувач обирає конкретний аудіофайл, для якого бажає переглянути текстову транскрипцію. Система виконує запит до бази даних для отримання текстової транскрипції, пов'язаної з обраним аудіофайлом (з масиву MASS_TRN). Проводиться перевірка наявності транскрипції для даного файлу. Якщо транскрипція доступна, вона відображається користувачеві. У випадку її відсутності (наприклад, ще не згенерована або сталася помилка під час генерації), виводиться відповідне повідомлення про відсутність. Після цього, управління повертається до етапу "Вибір дії користувача".

4. Вихід:

Користувач ініціює завершення своєї поточної сесії. Система виконує необхідні дії для коректного завершення сесії користувача, після чого його взаємодія з основними функціями системи припиняється. Алгоритм завершується, що означає повне припинення роботи системи або завершення сесії конкретного користувача.

Спосіб формування результатів розв'язання:

Формування результатів розв'язання задачі здійснюється на основі обробки даних, що зберігаються в масивах використовуваної інформації (MASS_USR, MASS_AUD, MASS_TRN, MASS_CAT, MASS_AUT, MASS_AUD_CAT, MASS_SESS). Цей процес включає такі послідовні етапи:

Вибірка даних: Здійснюється вибір необхідної інформації з бази даних за допомогою запитів (еквівалентних SQL-запитам) до відповідних масивів. Наприклад, для відображення списку аудіофайлів вибираються дані з MASS_AUD та MASS_AUT.

Обробка та агрегація даних: Вибрані дані піддаються подальшій логічній обробці, яка може включати фільтрацію, сортування, групування, а також агрегаційні розрахунки (наприклад, підрахунок загальної кількості користувачів, обсягу завантажених файлів для звітів).

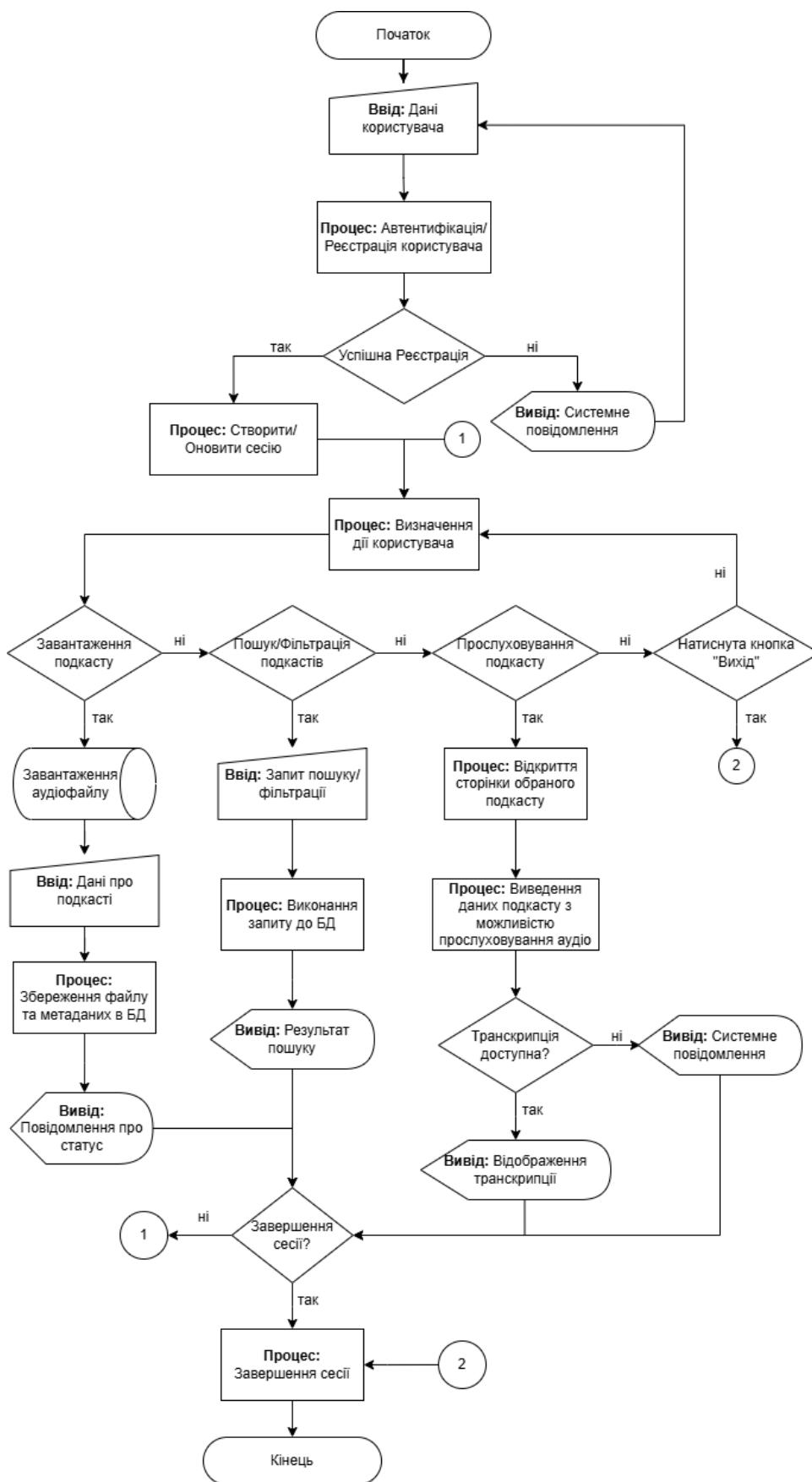


Рисунок 2.4 - Блок-схема алгоритму виконання задачі

Джерело: сформовано автором на основі виконаного дослідження

Форматування результатів: Отримані та оброблені дані перетворюються у формат, придатний для їх подальшого використання. Для відображення користувачам (відеограми) дані формуються для веб-інтерфейсу або мобільного додатка. Для адміністративних звітів (машинограми) дані можуть бути експортовані у формати PDF, XLSX або інші.

Індексація та збереження: Під час додавання нового контенту або транскрипцій, відбувається оновлення або створення пошукових індексів (RES_SRCH_IND) для забезпечення швидкого доступу до даних. Також фіксуються логи дій користувачів (RES_LOG_ACT) для аудиту та аналізу.

2.3 Моделювання інформаційної підсистеми

2.3.1 Моделювання поведінки системи

Цей підрозділ присвячений детальному моделюванню поведінки розроблюваної інформаційної системи забезпечення роботи цифрової аудіотеки, що здійснюється за допомогою діаграми прецедентів.

На діаграмі прецедентів (Рис. 2.5) візуалізовані основні функціональні можливості системи "Система підтримки роботи цифрової аудіотеки" з точки зору її зовнішніх взаємодій. Усі операції та процеси, що відбуваються всередині системи, абстрагуються, дозволяючи зосередитися на тому, що система робить для своїх користувачів та інших систем, а не як вона це робить.

У взаємодії з системою задіяні чотири ключові актори, що розрізняються за своїми ролями та правами доступу.

Перший актор – Неавторизований користувач. Цей суб'єкт представляє відвідувача системи, який ще не пройшов процедуру входу або реєстрації. Його взаємодія з системою суттєво обмежена базовими функціями, необхідними для отримання доступу. Для неавторизованого користувача передбачена можливість зареєструватися у системі, створивши новий обліковий запис, що є першим кроком до повноцінного використання функціоналу. Крім того, він може авторизуватися, тобто здійснити вхід до системи за допомогою своїх облікових

даних, що є передумовою для переходу до статусу авторизованого користувача та отримання розширених прав.

Другий актор – Авторизований користувач. Цей актор символізує Користувача, який успішно пройшов процедуру авторизації та має діючий обліковий запис у системі. Для авторизованого користувача доступний весь основний функціонал цифрової аудіотеки. Він може завантажити аудіофайл, додаючи новий контент до бібліотеки разом з необхідними метаданими. Цей процес завантаження аудіофайлу має важливий аспект: він розширюється (<<extend>>) прецедентом "Згенерувати транскрипцію". Це означає, що після успішного завантаження аудіофайлу, за певних умов або налаштувань системи, автоматично ініціюється процес його обробки для створення текстової транскрипції. Авторизований користувач також може відредагувати інформацію про аудіофайл та видалити аудіофайл, якщо він є його власником. Для навігації по бібліотеці передбачений функціонал шукати аудіофайл та фільтрувати аудіофайл за різними критеріями, а для прослуховування контенту надано можливість відтворити аудіофайл. Крім того, користувач може переглянути транскрипцію будь-якого доступного аудіофайлу та, за необхідності, завантажити транскрипцію на свій пристрій. Для управління своїм профілем, авторизований користувач має можливість змінити облікові дані та, за бажанням, видалити обліковий запис повністю.

Третій актор – Адміністратор. Цей актор є особливим типом авторизованого користувача, що володіє розширеними правами доступу та контролю над усією системою. Як і звичайний користувач, Адміністратор повинен авторизуватися для доступу до своїх функцій. Його діяльність зосереджена на управлінні ключовими елементами системи. Адміністратор може керувати користувачами, що включає можливість перегляду, додавання, редагування та видалення облікових записів. Він має повний контроль над контентом, що реалізується через прецеденти керувати аудіофайлами та керувати транскрипціями, дозволяючи здійснювати модерацію та підтримку цілісності даних. Функціонал керувати категоріями дозволяє Адміністратору

створювати, змінювати та видаляти класифікаційні елементи аудіотеки. Важливою функцією є можливість переглянути звіт, що надає доступ до системної статистики та аналітичної інформації про функціонування аудіотеки.

Четвертий актор – Модуль транскрипції. Цей актор представляє зовнішню систему або програмний модуль, що інтегрований з цифровою аудіотекою для надання спеціалізованих послуг з автоматичного розпізнавання мовлення. Його ключова функція полягає у прецеденті "Згенерувати транскрипцію". Цей варіант використання ініціюється самою системою (як вже зазначалося, як розширення прецеденту завантаження аудіофайлу) і відповідає за перетворення аудіоінформації на текстовий формат, який потім повертається до системи для зберігання та подальшого використання.

Таким чином, представлена діаграма прецедентів є узагальненою моделлю функціональних вимог до системи підтримки роботи цифрової аудіотеки. Вона чітко визначає, які функції система надає різним категоріям своїх користувачів, враховуючи їхній статус авторизації, а також які зовнішні системи залучені до її роботи, повністю відповідаючи постановці задачі та попередньо визначеним вимогам. Діаграма охоплює всі можливі сценарії взаємодії, що забезпечує повне та чітке розуміння поведінки системи.

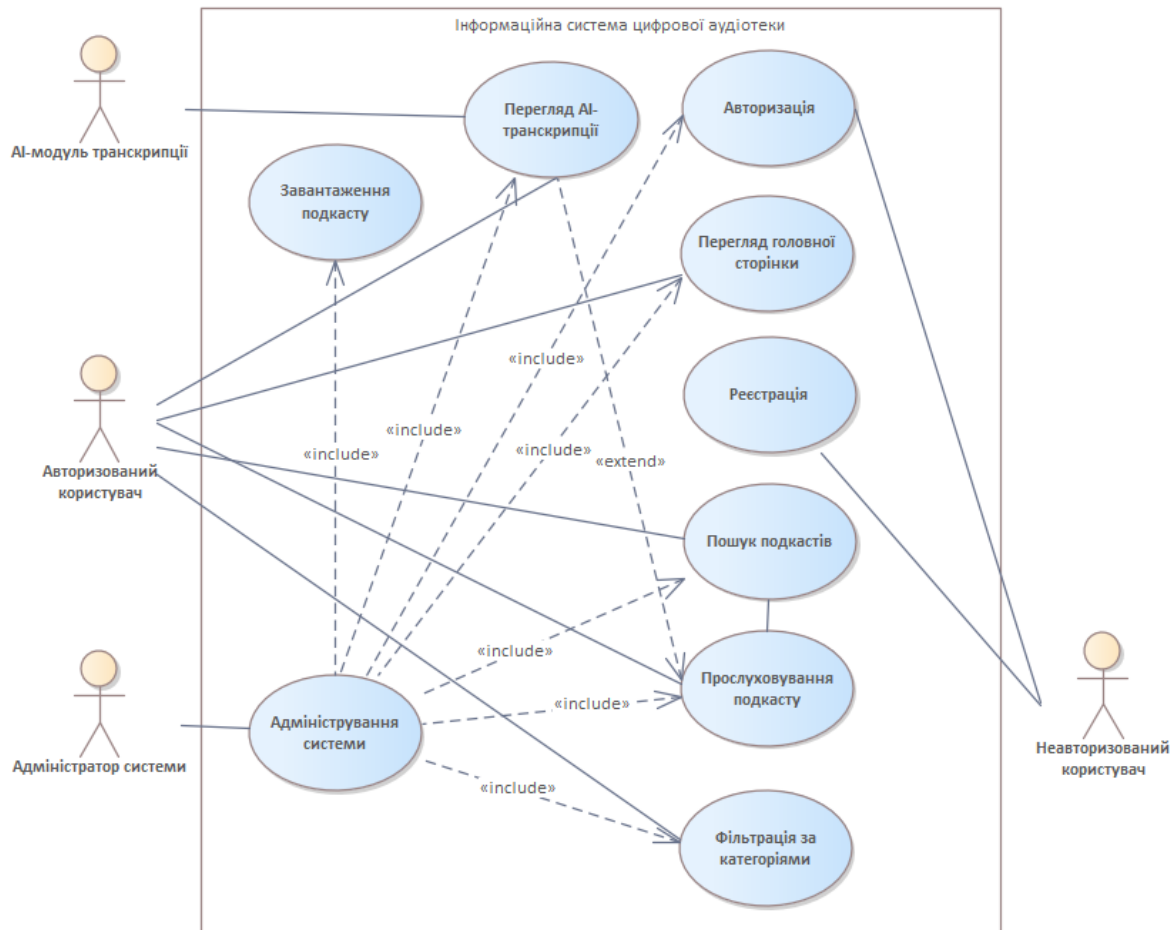


Рисунок 2.5 - Діаграма прецедентів

Джерело: сформовано автором на основі виконаного дослідження

Після визначення функціональних вимог та моделювання поведінки системи за допомогою діаграми прецедентів, наступним етапом є архітектурне представлення системи на рівні її фізичних компонентів.

Діаграма послідовності для функції “Авторизація” (Рис. 2.6) ілюструє процес авторизації користувача в системі. Користувач вводить свої облікові дані у форму входу (LoginForm), після чого форма ініціює перевірку через компонент AuthManager. AuthManager звертається до бази даних користувачів (UserDB) для перевірки правильності введених даних. Після отримання результату перевірки від UserDB, AuthManager повертає статус автентифікації до LoginForm, яка, у свою чергу, відображає результат користувачу. Таким чином, діаграма демонструє типовий сценарій взаємодії між користувачем, інтерфейсом, логікою авторизації та базою даних під час входу в систему.

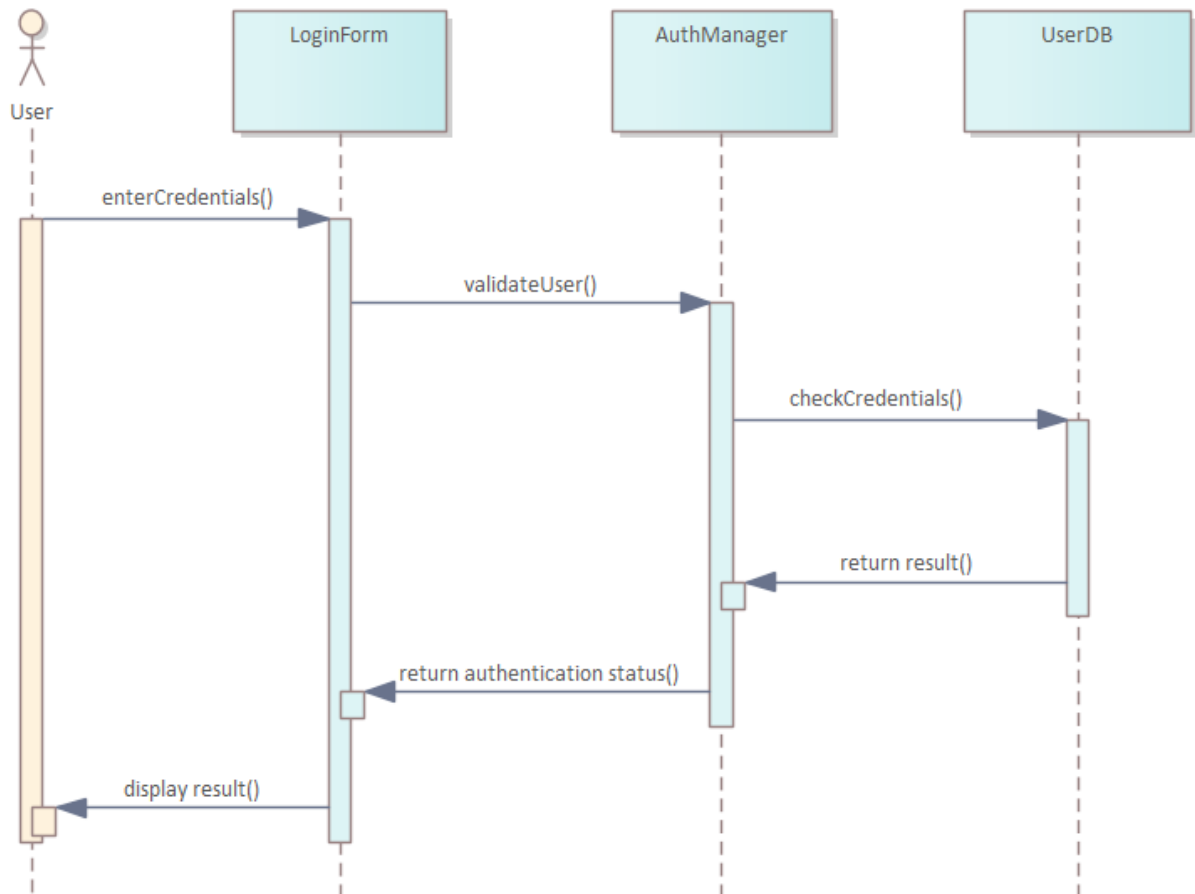


Рисунок 2.6 - Діаграма послідовності “Авторизації” користувача

Джерело: сформовано автором на основі виконаного дослідження

Діаграма для функції “Реєстрація” (Рис. 2.7) демонструє процес реєстрації нового користувача в системі. Користувач вводить свої реєстраційні дані у форму (RegistrationForm), яка ініціює запит на реєстрацію через компонент AuthManager. AuthManager спочатку перевіряє, чи існує користувач із такими даними у базі (UserDB) за допомогою запиту checkUserExists(). UserDB повертає результат — існує користувач чи ні.

Якщо користувач з такими даними ще не існує, AuthManager надсилає запит на створення нового користувача (createUser()) до UserDB. Після цього база повертає статус операції (успіх або помилка). AuthManager передає результат реєстрації назад у RegistrationForm, яка відображає відповідне повідомлення користувачу. Таким чином, діаграма ілюструє повний цикл

перевірки унікальності та створення нового облікового запису з обробкою можливих результатів.

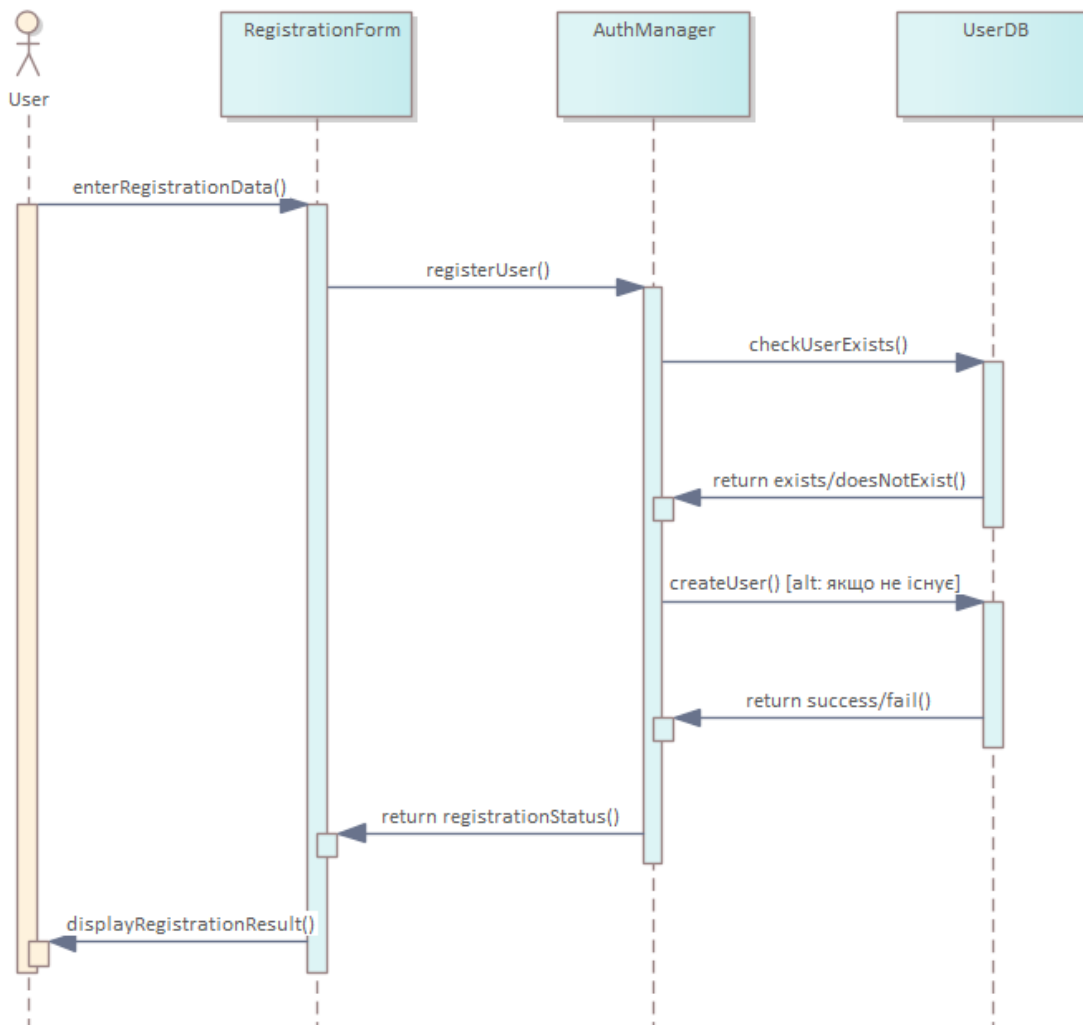


Рисунок 2.7 - Діаграма послідовності “Реєстрації” користувача

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності функції “Фільтрація за категоріями” (Рис. 2.8) ілюструє процес фільтрації подкастів за категорією. Користувач обирає потрібну категорію у формі пошуку (SearchForm), яка надсилає запит на фільтрацію до SearchManager. SearchManager спочатку звертається до CategoryRepository для отримання інформації про обрану категорію, а потім ініціює пошук подкастів у PodcastRepository, використовуючи знайдену категорію як критерій.

PodcastRepository повертає SearchManager список подкастів, що відповідають обраній категорії. Далі SearchManager передає результати фільтрації назад у SearchForm, яка відображає користувачу оновлений список

подкастів. Така послідовність забезпечує ефективний і структурований пошук контенту за категоріями, підвищуючи зручність користування цифровою аудіотекою.

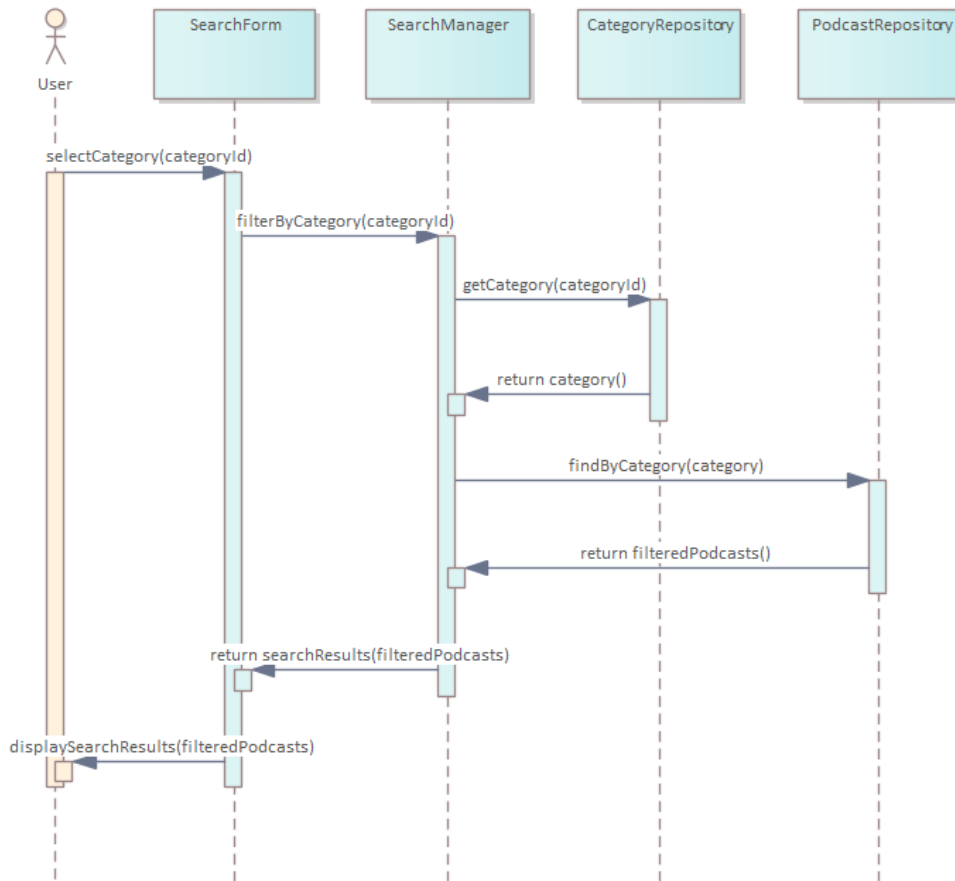


Рисунок 2.8 - Діаграма послідовності “Фільтрації за категоріями”

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності функції “Прослуховування подкастів” (Рис. 2.9) ілюструє процес прослуховування подкасту користувачем. Користувач обирає подкаст у інтерфейсі (PodcastView), який надсилає запит до PodcastManager для отримання детальної інформації про подкаст. PodcastManager звертається до об’єкта Podcast, отримує необхідні дані та повертає їх у PodcastView, де вони відображаються користувачу.

Після цього користувач ініціює відтворення подкасту. PodcastView надсилає команду на відтворення до PodcastManager, який передає файл подкасту на AudioPlayer. AudioPlayer запускає відтворення, повертає статус відтворення, а PodcastManager оновлює лічильник прослуховувань для цього

подкасту. Після оновлення інформації PodcastView відображає користувачу елементи керування відтворенням та поточний статус. Така послідовність забезпечує інтерактивний і зручний процес прослуховування подкастів із відображенням актуальної інформації.

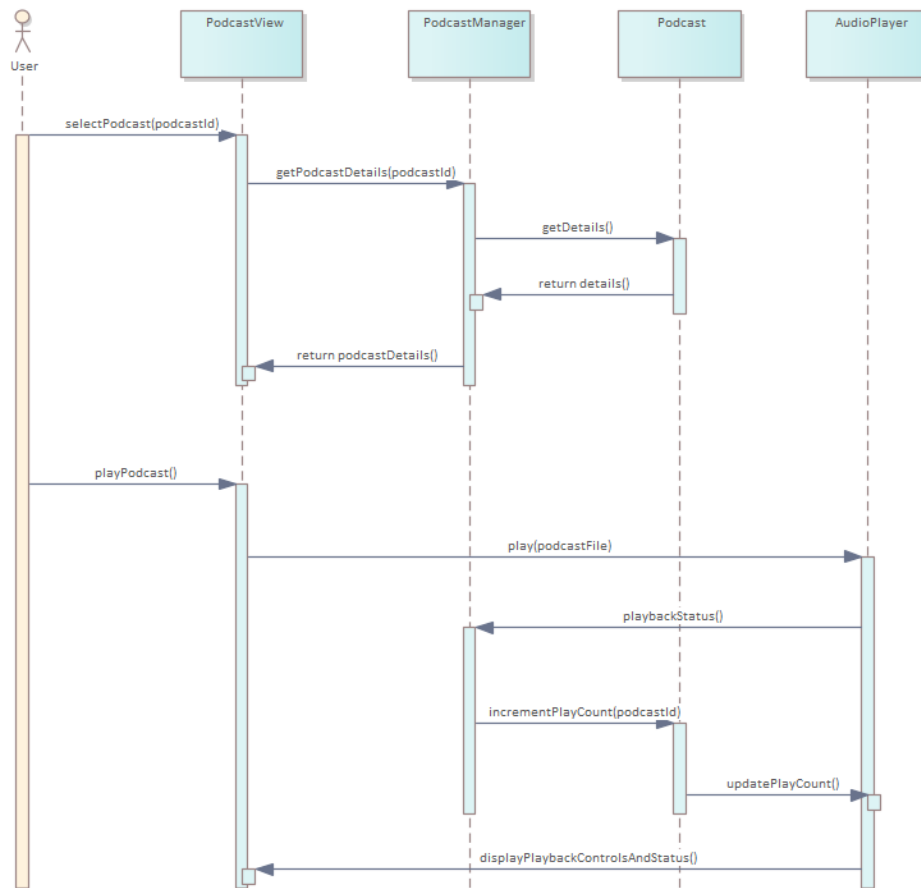


Рисунок 2.9 - Діаграма послідовності “Прослуховування подкастів”

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності функції “Пошук подкастів” (Рис. 2.10) ілюструє процес пошуку подкастів за ключовим словом або фразою. Користувач вводить пошуковий запит у форму пошуку (SearchForm), яка надсилає цей запит до SearchManager. SearchManager, у свою чергу, звертається до PodcastRepository з методом findByTitleOrDescription(), щоб знайти подкасти, які відповідають критеріям пошуку.

PodcastRepository повертає SearchManager список подкастів, що відповідають запиту. Далі SearchManager передає результати пошуку назад у SearchForm, яка відображає їх користувачу. Така послідовність забезпечує швидкий і зручний пошук релевантного аудіоконтенту в цифровій аудіотеці.

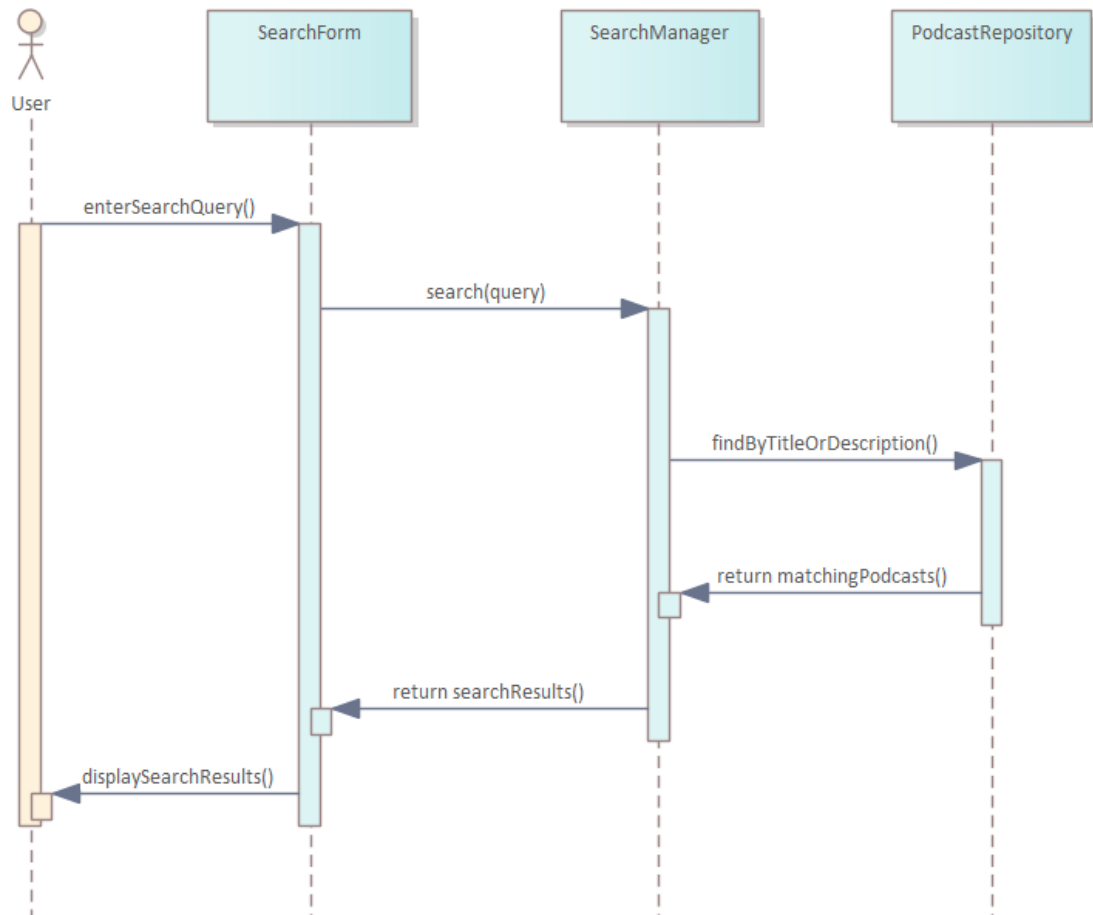


Рисунок 2.10 - Діаграма послідовності “Пошук подкастів”

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності функції “Перегляд AI-транскрипції” (Рис. 2.11) ілюструє процес отримання та відображення AI-транскрипції для подкасту. Користувач ініціює перегляд транскрипції через інтерфейс PodcastView, який надсилає запит до TranscriptManager. TranscriptManager спочатку намагається знайти вже існуючу транскрипцію для обраного подкасту (`findByPodcastId`). Якщо транскрипція відсутня, TranscriptManager звертається до зовнішнього AIService для генерації тексту на основі аудіо подкасту.

Після отримання згенерованого тексту від AIService, TranscriptManager створює новий запис транскрипції (`create`) і зберігає його. Далі текст транскрипції повертається у PodcastView, який відображає його користувачу. Така послідовність забезпечує автоматичне створення транскрипції за потреби та її зручний перегляд у системі.

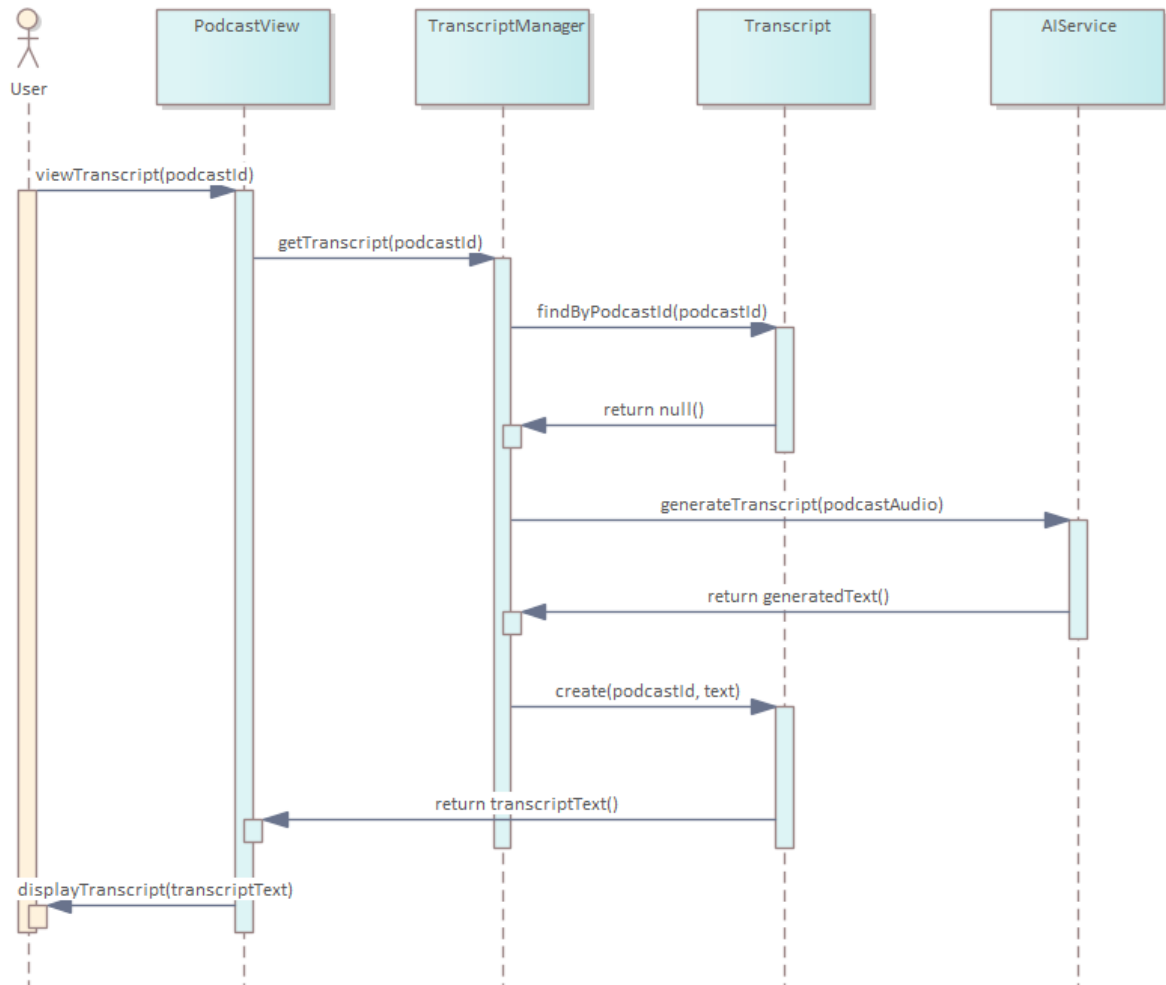


Рисунок 2.11 - Діаграма послідовності “Перегляд AI-транскрипції”

Джерело: сформовано автором на основі виконаного дослідження

Діаграма послідовності функції “Завантаження подкасту” (Рис. 2.12) ілюструє процес завантаження нового подкасту користувачем. Користувач вводить дані подкасту у форму завантаження (UploadForm), яка спочатку перевіряє коректність вибраної категорії через CategoryRepository. Після отримання статусу перевірки категорії UploadForm надсилає запит на завантаження подкасту до PodcastManager.

PodcastManager ініціює створення нового подкасту, звертаючись до PodcastRepository з відповідними даними. Після завершення операції PodcastRepository повертає статус завантаження PodcastManager, який передає його назад у UploadForm. Форма відображає користувачу результат завантаження, інформуючи про успіх або помилку операції. Така послідовність

забезпечує контроль коректності введених даних і надійне додавання нового контенту до системи.

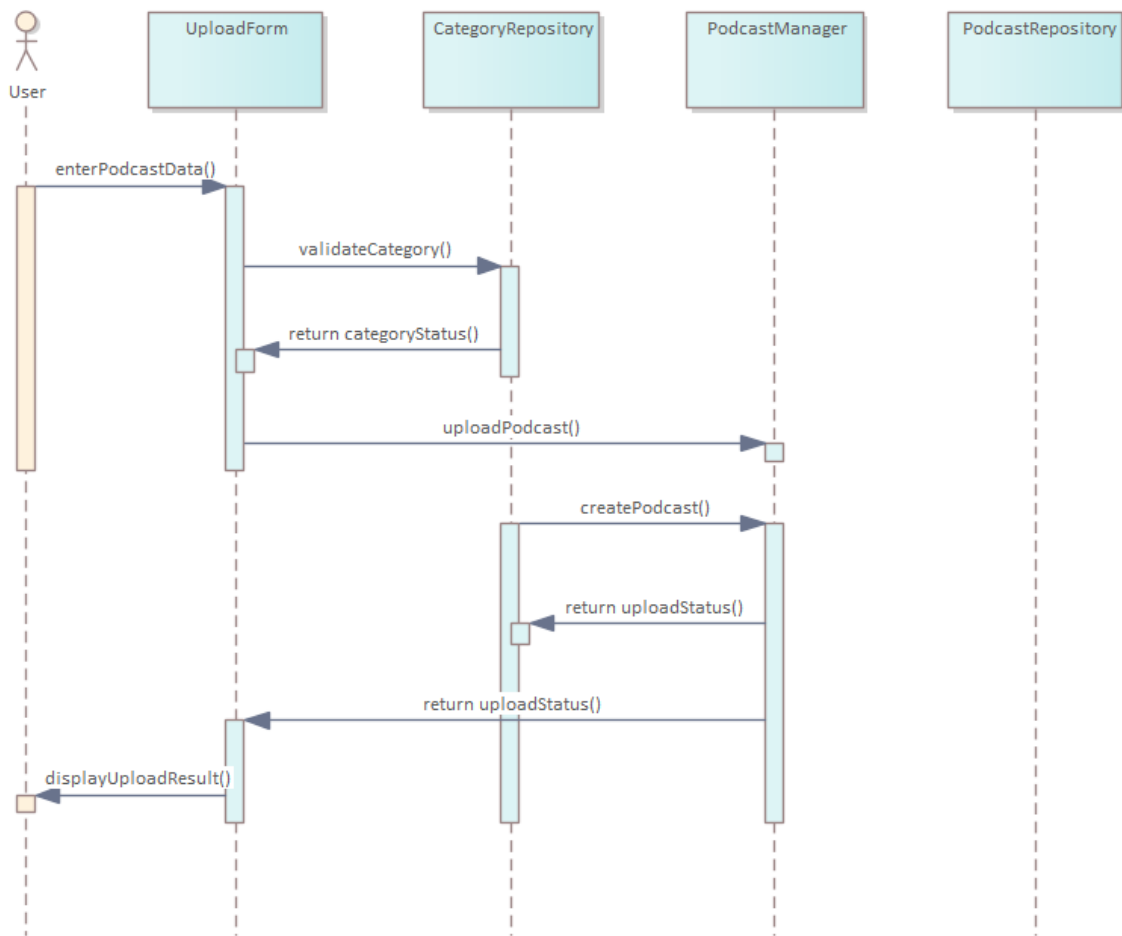


Рисунок 2.12 - Діаграма послідовності “Завантаження подкасту”

Джерело: сформовано автором на основі виконаного дослідження

2.3.2 Модель структури системи

Модель структури інформаційної системи цифрової аудіотеки представлена на діаграмі класів, що відображає статичну організацію основних компонентів та їх взаємозв'язків (Рис. 2.13). Дана діаграма є ключовим елементом проєктування, оскільки дозволяє наочно продемонструвати, як різні частини системи взаємодіють між собою, які дані зберігаються та які функції реалізуються на кожному рівні архітектури.

На діаграмі класів виділено три основні групи класів, що відповідають різним стереотипам у моделюванні складних інформаційних систем. Першу групу складають класи-сутності, які використовуються для зберігання даних із

довгим життєвим циклом. До цієї групи належать класи Users та Podcasts. Клас Users містить атрибути, що ідентифікують користувача, зокрема user_id, login, password, email та created_at. Клас Podcasts, у свою чергу, містить атрибути, які описують подкаст: id, user_id, title, author, category, description, audio_path, audio_url, cover_url, duration, file_size, published_at та created_at. Важливою особливістю є наявність у класі Podcasts атрибута user_id, який виступає зовнішнім ключем і забезпечує зв'язок кожного подкасту з конкретним користувачем системи. Таким чином, реалізується відношення "один-до-багатьох", коли один користувач може бути автором декількох подкастів.

Другою групою є граничні класи, які моделюють інтерфейси системи на високому рівні та забезпечують взаємодію користувача з функціоналом цифрової аудіотеки. До цієї групи належать класи LoginForm, RegistrationForm, PodcastView, UploadForm та SearchForm. Кожен із цих класів відповідає за окремий сценарій використання: автентифікацію, реєстрацію нового користувача, перегляд інформації про подкаст, завантаження нового контенту та пошук подкастів відповідно. Граничні класи отримують дані від користувача, ініціюють відповідні запити та відображають результати взаємодії із системою.

Третю групу складають класи-менеджери, які відповідають за реалізацію бізнес-логіки, координацію подій та послідовну поведінку сценаріїв, що закладені у відповідних прецедентах. У системі виділено три основних класи-менеджери: AuthManager, PodcastManager та SearchManager. Клас AuthManager забезпечує автентифікацію та реєстрацію користувачів, обробляючи запити від LoginForm та RegistrationForm і взаємодіючи з класом Users для перевірки та створення облікових записів. PodcastManager відповідає за отримання, завантаження та підрахунок прослуховувань подкастів, обробляючи запити від PodcastView та UploadForm і працюючи з класом Podcasts. SearchManager реалізує функціонал пошуку подкастів за заданими критеріями, отримуючи запити від SearchForm та звертаючись до класу Podcasts для отримання релевантних результатів.

Зв'язки між класами на діаграмі відображають напрямки взаємодії та залежності. Граничні класи звертаються до відповідних класів-менеджерів, які, у свою чергу, працюють із класами-сутностями для зчитування, створення чи оновлення даних. Така структура забезпечує чітке розмежування відповідальності між компонентами системи, сприяє прозорості архітектури та спрощує подальшу підтримку й розвиток інформаційної системи. Відокремлення граничних класів, класів-менеджерів і класів-сутностей дозволяє легко масштабувати систему, розширювати її функціонал та забезпечувати високий рівень підтримуваності програмного коду.

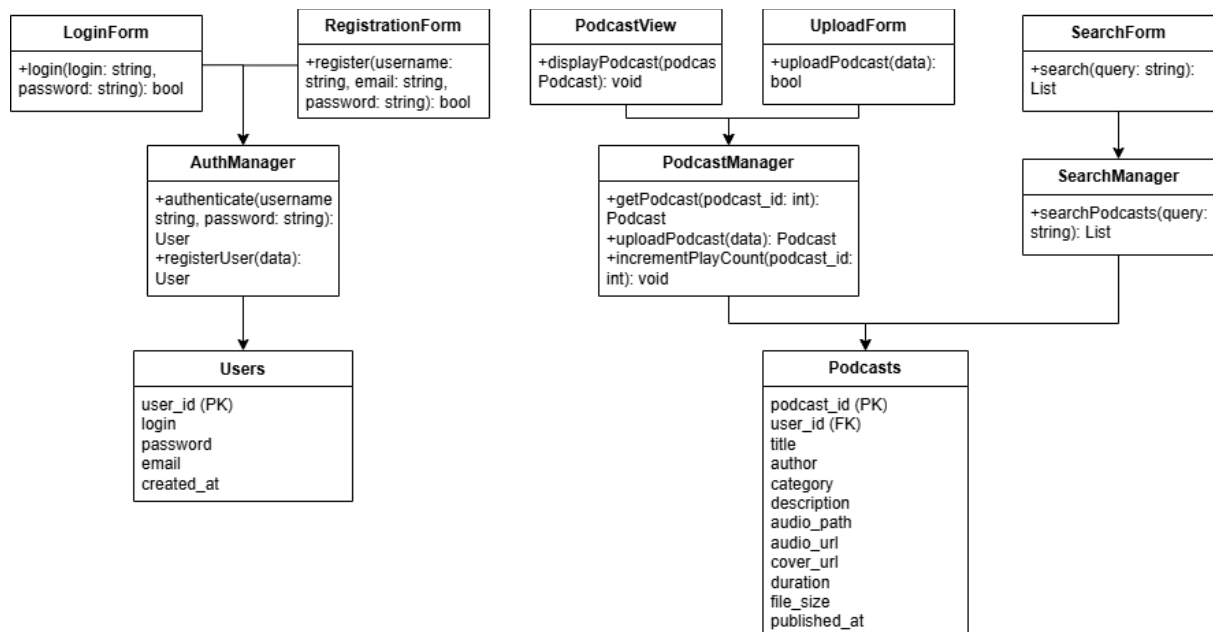


Рисунок 2.13 - Діаграма класів

Джерело: сформовано автором на основі виконаного дослідження

2.3.3 Розподіл вимог за компонентами системи

Розподіл вимог за компонентами системи є важливим етапом верифікації проектних рішень, оскільки дозволяє переконатися, що всі визначені на попередніх етапах вимоги до цифрової аудіотеки мають чітке відображення у структурі та функціональних компонентах системи. Для цього використовується діаграма трасування, яка демонструє взаємозв'язок між високорівневими вимогами, прецедентами використання та відповідними елементами моделі системи.

На діаграмі трасування, поданій на Рис. 2.14, центральне місце займає основна вимога до системи цифрової аудіотеки, яка передбачає розробку інформаційної системи з підтримкою реєстрації користувачів, завантаження та доступу до каталогу медіа-контенту. Від цієї вимоги відгалужуються більш деталізовані функціональні вимоги, такі як забезпечення реєстрації та автентифікації користувачів, можливість додавання підкастів, прослуховування підкастів, а також організація каталогу та пошуку аудіоконтенту.

Кожна з цих вимог трасується до відповідних прецедентів використання, які відображають ключові сценарії взаємодії користувача із системою. Наприклад, вимога щодо реєстрації та автентифікації користувачів пов'язана з прецедентами "Реєстрація" та "Авторизація", а вимога щодо додавання підкастів — із прецедентом "Завантаження подкасту". Вимоги, що стосуються каталогу та пошуку аудіоконтенту, трасуються до таких прецедентів, як "Пошук подкастів", "Фільтрація за категоріями" та "Перегляд головної сторінки".

Далі кожен прецедент пов'язується з відповідними елементами інтерфейсу користувача, які реалізують зазначені сценарії. Наприклад, сценарій "Реєстрація" реалізується через компонент UI «Реєстрація», а сценарій "Завантаження подкасту" — через UI «Завантаження подкасту».

Загалом, проведене трасування вимог до компонентів системи підтверджує коректність прийнятих проектних рішень, забезпечує прозорість архітектури та створює підґрунтя для подальшої розробки, тестування й впровадження цифрової аудіотеки.

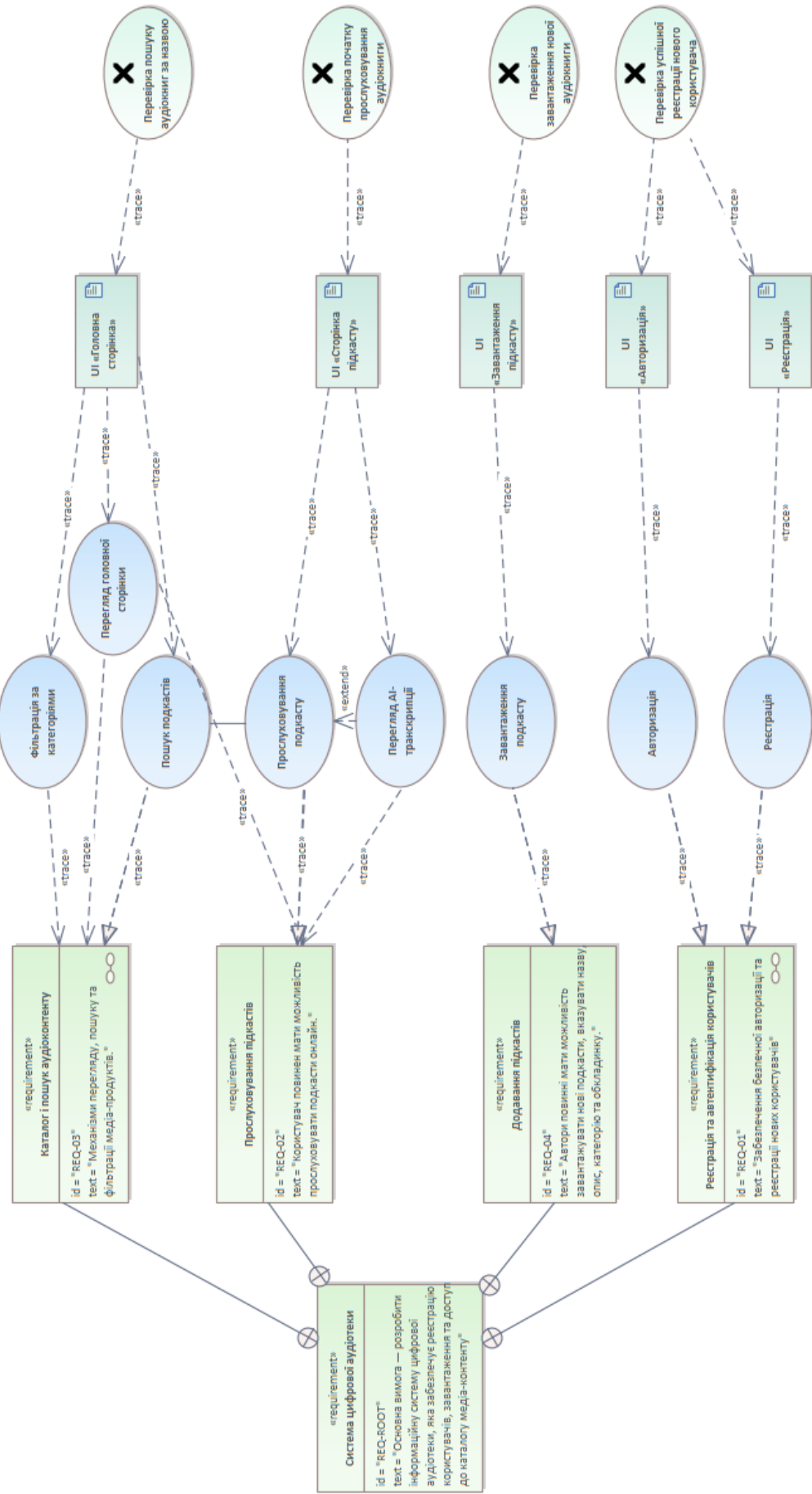


Рисунок 2.14 – Діаграма трасування
Джерело: сформовано автором на основі виконаного

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

Загальна характеристика інформаційного забезпечення. Інформаційне забезпечення (ІЗ) розробленої системи є фундаментом для її функціонування, забезпечуючи зберігання, управління та доступ до всіх даних, необхідних для роботи цифрової аудіотеки. Структура ІЗ спроектована за двокомпонентним принципом для ефективної обробки різнорідних даних. Перший компонент — це структурована реляційна база даних, що відповідає за зберігання метаданих. Другий компонент — це файлове сховище, призначене для розміщення неструктурованих даних, а саме аудіофайлів подкастів. Принципи організації інформаційного забезпечення ґрунтуються на клієнт-серверній архітектурі, де серверна частина на мові Python з використанням фреймворку Flask виступає єдиною точкою доступу до бази даних та файлів, забезпечуючи цілісність та контрольовану взаємодію з даними.

Як основний носій для структурованих даних було обрано систему управління базами даних (СКБД) PostgreSQL. Цей вибір обґрунтований кількома ключовими перевагами. По-перше, PostgreSQL є об'єктно-реляційною СКБД, що ідеально відповідає структурі даних проєкту, де існують чіткі зв'язки між сутностями, такими як користувачі та їхні подкасти. По-друге, ця система відома своєю високою надійністю та суворою відповідністю принципам АСІД (атомарність, узгодженість, ізольованість, довговічність), що є критично важливим для забезпечення цілісності транзакцій, наприклад, під час реєстрації нового користувача або додавання контенту. Серед інших важливих характеристик PostgreSQL варто відзначити підтримку широкого спектра типів даних, потужні механізми індексації для прискорення пошукових запитів та високу продуктивність при обробці великої кількості одночасних підключень, що є необхідною умовою для масштабування системи.

Для забезпечення достовірності та надійності інформації в системі реалізовано комплексний підхід до її контролю. На прикладному рівні, у середовищі Flask, впроваджено валідацію всіх вхідних даних, що надходять від клієнтського додатку. Це дозволяє перевіряти коректність форматів, наявність обов'язкових полів та відповідність даних бізнес-логіці ще до їх потрапляння у сховище. На рівні бази даних контроль забезпечується за допомогою вбудованих механізмів PostgreSQL: використання первинних ключів (primary key) для ідентифікації унікальності записів у таблицях users та podcasts, а також зовнішніх ключів (foreign key) для підтримки посилальної цілісності між ними. Доступ до інформації жорстко контролюється через механізм класичних серверних сесій: лише автентифіковані користувачі можуть завантажувати контент, при цьому можливість редагування та видалення обмежується лише власними публікаціями.

Загальна схема інформаційного забезпечення системи включає базу даних PostgreSQL, яка централізовано зберігає всю службову інформацію, та файлове сховище, інтегроване з серверною частиною для розміщення аудіофайлів. Основними елементами інформаційної бази, що використовуються при функціонуванні системи, є дві ключові таблиці. Перша, таблиця users, зберігає облікові дані користувачів, включаючи логін, хешований пароль та контактну інформацію. Друга, таблиця podcasts, містить вичерпні метадані про аудіозаписи: назву, автора, опис, категорію, тривалість, а також службові посилання на аудіофайл та обкладинку у файловому сховищі. Разом ці компоненти та елементи створюють надійне та структуроване середовище для ефективної роботи цифрової аудіотеки.

Організація збору і передавання первинної інформації. Організація збору та передавання первинної інформації в розробленій системі базується на чіткому розмежуванні джерел даних та використанні сучасних веб-технологій для їх безпечного транспортування. Виділяються два основні джерела інформації: кінцеві користувачі системи (слухачі та автори контенту) та зовнішній програмний сервіс штучного інтелекту. Загальною вимогою до організації

процесу є інтерактивність та ініціація з боку користувача для більшості операцій, а також автоматизація для внутрішніх процесів обробки. Передавання всіх даних між клієнтським додатком, реалізованим на React, та серверною частиною на Flask відбувається за допомогою захищеного протоколу HTTPS, що забезпечує конфіденційність та цілісність інформації.

Ключовим джерелом первинної інформації виступає користувач. Він відповідає за своєчасне подання даних, необхідних для функціонування аудіотеки. Цей процес реалізується через кілька типів вхідних повідомлень. По-перше, це дані для реєстрації та автентифікації, які користувач вводить у відповідні форми (відеограми). Ця інформація надається за запитом, одноразово під час створення облікового запису або періодично під час входу в систему. По-друге, найважливішим повідомленням є завантаження нового подкасту, що ініціюється автором контенту. У цьому випадку джерелом є користувач, а інформація складається з двох частин: метаданих (назва, опис, категорія), що вводяться через веб-форму, та самого аудіофайлу (у форматі MP3, WAV тощо), який виступає носієм первинного контенту. Періодичність цієї операції є нерегламентованою та залежить виключно від бажання автора. Крім того, користувач постійно генерує запити на пошук та фільтрацію, які також є вхідними повідомленнями, що передаються від клієнта до сервера для обробки.

Другим важливим джерелом інформації є зовнішній AI-сервіс — модель OpenAI Whisper, що відповідає за автоматичне розпізнавання мовлення. На відміну від користувацьких даних, формування цього вхідного повідомлення ініціюється самою системою. Процес запускається автоматично (подієво) після того, як користувач успішно завантажив новий аудіофайл на сервер. Серверна частина системи передає аудіофайл на обробку до сервісу Whisper, який, у свою чергу, повертає згенеровану текстову транскрипцію. Носієм цієї інформації є структурований об'єкт даних (наприклад, у форматі JSON), що передається через програмний інтерфейс (API). Таким чином, система збагачується новими даними без безпосереднього втручання людини. Відповідальність за своєчасний запит та обробку відповіді від AI-сервісу несе серверна частина інформаційної системи.

Побудова системи класифікації та кодування. Для забезпечення ефективної структуризації даних, уникнення дублювання та прискорення операцій пошуку й вибірки в розробленій інформаційній системі застосовується комбінована система класифікації та кодування. Ця система охоплює ключові об'єкти предметної галузі, зокрема користувачів, подкасти та їх тематичні категорії. Підхід до кодування ґрунтується на використанні унікальних ідентифікаторів для основних сутностей та описових класифікаторів для групування контенту.

Основні об'єкти системи, «Користувач» та «Подкаст», ідентифікуються за допомогою унікальних кодів, що генеруються автоматично. Для кодування цих об'єктів було обрано порядковий метод. Кожному новому запису в таблицях `users` та `podcasts` система управління базами даних PostgreSQL присвоює наступний доступний цілочисловий ідентифікатор у послідовності. Структура коду є простим цілим числом (`Integer`), а його довжина є змінною і залежить від загальної кількості записів у відповідній таблиці. Такий метод гарантує унікальність кожного об'єкта, що є критично важливим для коректного встановлення зв'язків між подкастами та їх авторами, а також значно спрощує адміністрування та цілісність даних.

Для класифікації аудіоконтенту за тематикою використовується класифікатор «Категорії подкастів». На відміну від ідентифікаторів об'єктів, тут застосовано мнемонічний метод кодування. Замість цифрового коду кожній категорії відповідає текстова назва, що описує її суть, наприклад, «Бізнес», «Технології» або «Новини», як це відображено в інтерфейсі користувача. Структура коду являє собою текстовий рядок (`character varying`) змінної довжини. Цей підхід є інтуїтивно зрозумілим як для користувачів, що фільтрують контент, так і для адміністраторів системи. Він також спрощує реалізацію логіки пошуку та фільтрації на програмному рівні, оскільки не потребує додаткових операцій для розшифровки кодів.

Таким чином, обрана система кодування та класифікації є збалансованою. Використання порядкового методу для унікальної ідентифікації об'єктів

забезпечує високу продуктивність та цілісність бази даних, тоді як мнемонічний метод класифікації контенту робить систему гнучкою, зрозумілою та зручною для кінцевого користувача.

Проектування форм первинних документів, машинограм та відеокадрів. Проектування вихідних інформаційних форм у системі цифрової аудіотеки орієнтоване на забезпечення максимальної зручності та інформативності для різних категорій користувачів. Усі вихідні дані представлені у трьох ключових формах: відеокадри, що є основним засобом інтерактивної взаємодії з користувачем; машинограми, призначені для адміністративного аналізу; та електронні повідомлення, що слугують для асинхронного сповіщення. Склад та зміст кожної форми обґрунтовані її функціональним призначенням та потребами цільового користувача.

Відеокадри є домінантною формою представлення інформації для кінцевого користувача. Їх проектування виконано з урахуванням вимог до сучасного користувацького досвіду (UX), що відображено у наданих прототипах. Вимоги до відеокадрів включають інтуїтивно зрозумілу навігацію, адаптивність інтерфейсу для різних пристроїв та мінімальний час відгуку на дії користувача.

- **Відображення аудіофайлів та метаданих (Ідентифікатор: OA_01).** Цей відеокадр, що представляє головну сторінку та сторінки каталогу, є основним для навігації по контенту. Його зміст включає списки подкастів, згруповані за категоріями, з відображенням обкладинки, назви та автора. Форма подання у вигляді карток дозволяє структурувати великий обсяг інформації. Цей відеокадр надається користувачеві системи за його запитом з очікуваним часом затримки не більше 2-5 секунд.

- **Текстові транскрипції подкастів (Ідентифікатор: OA_02).** Цей відеокадр є частиною сторінки конкретного подкасту, де текстове поле з транскрипцією розміщується поруч з аудіоплеєром. Таке рішення обґрунтоване необхідністю забезпечити можливість одночасного прослуховування та читання тексту. Документ надається користувачеві системи за запитом після завершення генерації транскрипції.

- **Системні повідомлення (Ідентифікатор: OA_03).** Ці короткі повідомлення у формі спливаючих вікон або сповіщень в інтерфейсі призначені для миттєвого інформування про результат операцій (наприклад, успішне завантаження файлу або помилка авторизації). Вони надаються як користувачеві, так і адміністратору миттєво, у відповідь на ту чи іншу подію в системі. Машинограми в системі спроектовані для задоволення потреб адміністративного персоналу.

- **Звіти про діяльність системи (Ідентифікатор: OA_04).** Цей документ призначений виключно для адміністратора системи та формується за його запитом. Обґрунтуванням форми подання у вигляді файлів PDF або Excel є потреба в подальшому аналізі, архівуванні та друці статистичних даних (наприклад, кількість нових користувачів, обсяг завантаженого контенту). Періодичність видачі є нерегламентованою, а термін формування залежить від обсягу аналізованих даних.

Електронні повідомлення використовуються для важливих асинхронних сповіщень, що мають гарантовано дійти до користувача. Окремо варто зазначити, що за допомогою електронної пошти будуть надсилатися три типи документів. По-перше, це повідомлення для підтвердження електронної адреси після реєстрації. По-друге, інструкції для відновлення забутого пароля, що надсилаються за запитом користувача. По-третє, сповіщення про успішне завершення процесу транскрипції подкасту, оскільки ця операція може займати тривалий час. Користувачем цих документів є користувач системи, а періодичність їх відправлення є подієвою.

Структура інформаційних масивів. Інформаційна база системи організована у вигляді набору взаємопов'язаних масивів (таблиць) у середовищі системи управління базами даних PostgreSQL. Кожен масив спроектовано для зберігання даних про конкретну сутність предметної області, що забезпечує логічну структурування та цілісність інформації. Основними інформаційними масивами, що забезпечують функціонування системи, є масив користувачів (users) та масив подкастів (podcasts). Нижче наведено детальний опис структури кожного з цих масивів.

Масив «Користувачі»

Масив призначений для зберігання облікових даних усіх зареєстрованих користувачів системи, включаючи їхні реєстраційні дані та особисту інформацію.

Таблиця 3.1 - Опис масиву «Користувачі»

Найменування параметру	Характеристика
Найменування масиву	Користувачі (users)
Ідентифікатор масиву	MASS_USR
Найменування носія інформації	СКБД PostgreSQL
Максимальний об'єм масиву	50 000 записів
Довжина запису	Змінна (використовуються типи character varying, timestamp)
Метод організації	Реляційний
Ключі упорядкування	id
Ідентифікатор індексного масиву	Керується СКБД PostgreSQL автоматично

Джерело: сформовано автором на основі виконаного дослідження

Таблиця 3.2 – Структура масиву «Користувачі»

Найменування	Ідентифікатор у програмі	Формат	Логічні чи семантичні зв'язки	Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле
Ідентифікатор користувача	id	Integer	Ні	РК	NOT NULL	Так	ІНД
Логін користувача	login	character varying	Ні	Ні	NOT NULL, Унікальне	Так	ІНД
Пароль	password	character varying	Ні	Ні	NOT NULL	Так	Ні
Повне ім'я	full_name	character varying	Ні	Ні	Ні	Ні	Ні
Електронна пошта	email	character varying	Ні	Ні	NOT NULL, Унікальне	Так	ІНД
Дата створення	created_at	timestamp without time zone	Ні	Ні	NOT NULL	Так	Ні

Джерело: сформовано автором на основі виконаного дослідження

Масив «Подкасти»

Цей масив є центральним сховищем метаданих про всі аудіофайли, завантажені в систему. Він містить описову інформацію, а також посилання на відповідних авторів та фізичне розташування аудіофайлів.

Таблиця 3.3 – Опис масиву «Подкасти»

Найменування параметру	Характеристика
Найменування масиву	Подкасти (podcasts)
Ідентифікатор масиву	MASS_AUD
Найменування носія інформації	СКБД PostgreSQL
Максимальний об'єм масиву	200 000 записів
Довжина запису	Змінна (використовуються типи character varying, text, integer, date, timestamp)
Метод організації	Реляційний
Ключі упорядкування	id, published_at
Ідентифікатор індексного масиву	Керується СКБД PostgreSQL автоматично

Джерело: сформовано автором на основі виконаного дослідження

Таблиця 3.4 – Структура масиву «Подкасти»

Найменування	Ідентифікатор у програмі	Формат	Логічні чи семантичні зв'язки	Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле
Ідентифікатор подкасту	id	Integer	Ні	PK	NOT NULL	Так	ІНД
Ідентифікатор користувача	user_id	Integer	Таблиця users	FK	NOT NULL	Так	ІДД
Назва	title	character varying	Ні	Ні	NOT NULL	Так	ІДД
Автор	author	character varying	Ні	Ні	NOT NULL	Так	ІДД
Категорія	category	character varying	Ні	Ні	NOT NULL	Так	ІДД
Опис	description	text	Ні	Ні	Ні	Ні	Ні
Шлях до аудіо	audio_path	text	Ні	Ні	Ні	Ні	Ні

Найменування	Ідентифікатор у програмі	Формат	Логічні чи семантичні зв'язки	Первинний/вторинний ключ	Умова на значення	Обов'язкове поле	Індексне поле
URL аудіо	audio_url	text	Ні	Ні	Ні	Ні	Ні
URL обкладинки	cover_url	text	Ні	Ні	Ні	Ні	Ні
Тривалість	duration	character varying	Ні	Ні	Ні	Ні	Ні
Розмір файлу	file_size	Integer	Ні	Ні	Ні	Ні	Ні
Дата публікації	published_at	date	Ні	Ні	Ні	Ні	ІДД
Дата створення	created_at	timestamp without time zone	Ні	Ні	NOT NULL	Так	Ні

Джерело: сформовано автором на основі виконаного дослідження

Вибір СКБД. Обґрунтування цього вибору базується на сукупності технічних та економічних переваг даної системи. По-перше, дані проєкту мають чітко виражену реляційну природу (зв'язок «один-до-багатьох» між користувачами та їхніми подкастами), що ідеально моделюється засобами PostgreSQL. По-друге, PostgreSQL відома своєю винятковою надійністю та суворим дотриманням принципів ACID, що є критично важливим для гарантування цілісності транзакцій, таких як реєстрація користувачів чи додавання нового контенту. По-третє, як система з відкритим вихідним кодом, PostgreSQL не потребує ліцензійних відрахувань, що робить її економічно ефективним рішенням. Нарешті, висока продуктивність, підтримка розширених механізмів індексації та здатність обробляти великі навантаження відповідають вимогам до масштабування системи, яка розрахована на обслуговування десятків тисяч користувачів та сотень тисяч аудіозаписів.

PostgreSQL — це потужна об'єктно-реляційна СКБД, що вирізняється своєю стабільністю, розширюваністю та відповідністю стандартам SQL. Вона підтримує широкий спектр типів даних, складні запити та механізми конкурентного доступу, такі як MVCC (Multi-Version Concurrency Control), що дозволяє мінімізувати блокування під час одночасної роботи багатьох користувачів. Система є кросплатформною та має активну спільноту розробників, що забезпечує її постійний розвиток та якісну підтримку.

Вибір PostgreSQL накладає певні вимоги та обмеження на процес проєктування. Основною вимогою є необхідність дотримання реляційної моделі даних при розробці серверної логіки. Це означає, що всі операції з даними мають виконуватися з урахуванням визначених зв'язків та обмежень цілісності (первинних, зовнішніх та унікальних ключів), як це було реалізовано при проєктуванні інформаційних масивів. Важливим обмеженням, що вплинуло на архітектуру, є неефективність реляційних СКБД при зберіганні великих бінарних об'єктів (аудіофайлів). Тому було прийнято проєктне рішення не зберігати аудіофайли безпосередньо в базі даних. Замість цього, в таблиці `podcasts` зберігаються лише метадані та посилання на файли (`audio_url`, `cover_url`), тоді як самі файли розміщуються в окремому файловому сховищі, що є оптимальним підходом для таких систем.

Інфологічна модель бази даних. Для розробки інформаційної системи цифрової аудіотеки було спроектовано інфологічну модель даних, що слугує основою для створення фізичної структури бази даних. Важливо зазначити, що розроблена система є системою операційної обробки транзакцій (OLTP), призначеною для виконання щоденних операцій, а не системою аналітичної обробки (OLAP). У зв'язку з цим, для проєктування було обрано класичну реляційну модель, а не сховище даних з архітектурою «зірка» чи «сніжинка».

Інфологічна модель була розроблена з використанням CASE-засобів, що дозволило візуалізувати структуру та забезпечити логічну цілісність. Модель описує дві ключові сутності предметної області: «Користувач» (User) та «Подкаст» (Podcast). Сутність «Користувач» представляє зареєстрованого користувача системи з атрибутами, що ідентифікують його (логін, email, пароль тощо). Сутність «Подкаст» містить усю метайнформацію про аудіозапис. Між цими сутностями встановлено зв'язок типу «один-до-багатьох» (one-to-many). Це означає, що один користувач може завантажити та бути автором багатьох подкастів, але кожен конкретний подкаст належить лише одному користувачеві. Цей зв'язок реалізується на фізичному рівні через зовнішній ключ `user_id` у таблиці `podcasts`, який посилається на первинний ключ таблиці `users`. Візуальне

представлення даної моделі та її об'єктів було наведено раніше на діаграмі класів (Рис. 2.13).

Ключовою вимогою при проєктуванні бази даних була її нормалізація з метою усунення надлишковості даних та підвищення їхньої узгодженості. Запроєктована структура бази даних відповідає вимогам третьої нормальної форми (3NF). Це досягається завдяки тому, що всі неключові атрибути в кожній таблиці повністю залежать від первинного ключа (id) і не мають транзитивних залежностей від інших неключових атрибутів. Наприклад, у таблиці podcasts такі атрибути, як title та description, безпосередньо описують конкретний подкаст (залежать від podcasts.id), а не, наприклад, його автора, інформація про якого вказана в окремому полі. Такий підхід забезпечує цілісність та мінімізує аномалії при оновленні чи видаленні даних.

Створення інфологічної моделі за допомогою сучасних CASE-засобів (наприклад, ERwin Data Modeler, Lucidchart, Draw.io) зазвичай включає кілька стандартизованих етапів. Стисле керівництво по розробці моделі:

1. **Визначення сутностей.** Перший крок полягає в ідентифікації ключових об'єктів предметної області, які потребують зберігання. У даному проєкті це «Користувач» та «Подкаст».
2. **Визначення атрибутів.** Для кожної сутності визначається набір її властивостей (атрибутів) та відповідних типів даних (наприклад, id (Integer), email (character varying), created_at (timestamp)).
3. **Встановлення первинних ключів.** Кожній сутності присвоюється унікальний ідентифікатор (первинний ключ), наприклад, поле id, для однозначної ідентифікації кожного запису.
4. **Побудова зв'язків.** Визначаються логічні зв'язки між сутностями. У даному випадку встановлюється зв'язок «один-до-багатьох» від «Користувача» до «Подкасту» шляхом створення зовнішнього ключа.
5. **Нормалізація та валідація.** Створена модель аналізується на відповідність нормальним формам для усунення надлишковості та потенційних аномалій даних. Цей ітеративний процес забезпечує

створення логічно коректної та оптимізованої схеми, яка слугує надійним планом для подальшої реалізації бази даних. Ілюстрації екранних форм, що демонструють цей процес у конкретному програмному пакеті, зазвичай наводяться в додатках до роботи.

Даталогічна модель бази даних. Для створення фізичної структури бази даних в середовищі PostgreSQL використовується мова опису даних (DDL). Наведені нижче запити CREATE TABLE повністю визначають структуру таблиць users та podcasts, включаючи всі атрибути, їхні типи даних та обмеження цілісності.

Створення таблиці users:

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    login VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    full_name VARCHAR(100),
    email VARCHAR(100) NOT NULL UNIQUE,
    created_at TIMESTAMP WITHOUT TIME ZONE NOT NULL DEFAULT
        CURRENT_TIMESTAMP);
```

Створення таблиці podcasts:

```
CREATE TABLE podcasts (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL,
    title VARCHAR(255) NOT NULL,
    author VARCHAR(100) NOT NULL,
    category VARCHAR(50) NOT NULL,
    description TEXT,
    audio_url TEXT,
    cover_url TEXT,
    duration VARCHAR(20),
    file_size INTEGER,
    published_at DATE,
    created_at TIMESTAMP WITHOUT TIME ZONE NOT NULL DEFAULT
        CURRENT_TIMESTAMP, -- Дата та час створення запису
```

```
-- Встановлення зв'язку "один-до-багатьох" з таблицею users
CONSTRAINT fk_user
FOREIGN KEY(user_id)
REFERENCES users(id)
ON DELETE CASCADE);
```

3.2. Технічне забезпечення

Загальні положення та схема автоматизації. Інформаційна система забезпечення роботи цифрової аудіотеки спроектована на базі сучасної хмарної архітектури, що не передбачає наявності спеціалізованого обладнання на боці кінцевого користувача чи замовника. Ядро системи, що включає серверну логіку на Flask та базу даних PostgreSQL, розташоване на віддалених серверах хмарної платформи Heroku. Клієнтська частина, реалізована як веб-додаток на React, розгорнута на платформі Netlify, яка також функціонує на базі розподіленої серверної інфраструктури. Доступ користувачів до всіх функцій системи здійснюється через стандартний веб-браузер з будь-якого пристрою (персональний комп'ютер, ноутбук, планшет, смартфон), що має підключення до мережі Інтернет.

Загальна структура комплексу технічних засобів (КТЗ), що забезпечує функціонування системи, та інформаційні потоки між її компонентами представлені на загальній схемі автоматизації (Рисунок 3.1). Схема ілюструє взаємодію клієнтських пристроїв з розподіленими хмарними сервісами, які разом утворюють єдине програмне середовище.

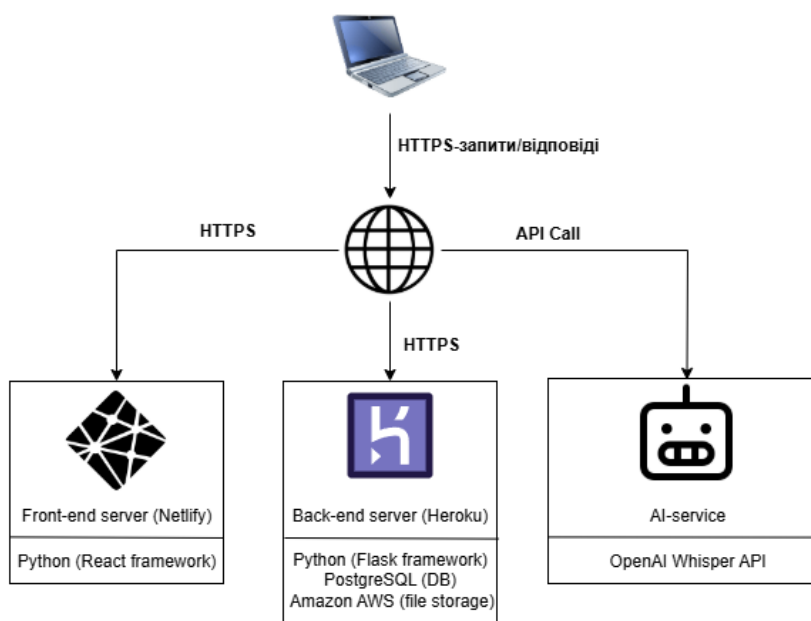


Рисунок 3.1 - Загальна схема автоматизації

Джерело: сформовано автором на основі виконаного дослідження

Структура комплексу технічних засобів. Структура комплексу технічних засобів (КТЗ) даної інформаційної системи базується на використанні хмарних платформ, що дозволяє відмовитися від придбання та обслуговування власного серверного обладнання. Обґрунтуванням такого вибору є висока надійність, гнучкість масштабування та економічна ефективність. Серверна частина системи та база даних PostgreSQL функціонують на інфраструктурі хостинг-провайдера Heroku, а клієнтський додаток розгорнуто на платформі Netlify. Оскільки система є веб-орієнтованою, поняття стаціонарного автоматизованого робочого місця (АРМ) відсутнє; доступ здійснюється з персональних пристроїв користувачів (ПК, ноутбуків, смартфонів) через веб-браузер. Система не використовує жодного додаткового специфічного обладнання, як-от датчики чи периферійні пристрої.

Розміщення КТЗ, його фізичний захист від зовнішніх впливів та несанкціонованого доступу повністю забезпечується хмарними провайдерами Heroku та Netlify. Їхні дата-центри відповідають високим стандартам безпеки, що включає клімат-контроль, системи безперебійного живлення, охорону та контроль доступу до приміщень. Захист даних під час їх передавання між

клієнтом та сервером гарантується використання захищеного протоколу HTTPS.

Обґрунтування чисельності персоналу для обслуговування КТЗ полягає в тому, що для підтримки функціонування апаратної частини системи не потрібен окремий штат співробітників. Усі обов'язки з обслуговування, оновлення та забезпечення безперебійної роботи фізичних серверів покладено на відповідні служби хостинг-провайдерів. Це дозволяє команді розробників зосередитися виключно на підтримці та розвитку програмного забезпечення.

3.3 Програмне забезпечення

Структура програмного забезпечення. Базове (системне) програмне забезпечення є фундаментом для функціонування системи. На серверній стороні (платформа Heroku) воно включає операційну систему (Linux-контейнер), систему управління базами даних PostgreSQL, веб-сервер (Gunicorn) для обробки HTTP-запитів та середовище виконання Python. На клієнтській стороні системне ПЗ представлено операційною системою користувача (наприклад, Windows, macOS, Android) та сучасним веб-браузером (Google Chrome, Mozilla Firefox), який виступає середовищем для виконання клієнтського додатку.

Прикладне (спеціалізоване) програмне забезпечення — це безпосередньо розроблені компоненти, що реалізують функціонал цифрової аудіотеки. Воно складається з двох основних частин:

- **Серверний додаток**, розроблений на мові Python з використанням фреймворку Flask. Він відповідає за обробку API-запитів від клієнта, автентифікацію та авторизацію користувачів, взаємодію з базою даних PostgreSQL та інтеграцію із зовнішнім сервісом транскрипції OpenAI Whisper.
- **Клієнтський додаток**, реалізований як односторінковий додаток (SPA) на бібліотеці React.js. Він відповідає за візуалізацію користувацького інтерфейсу, обробку дій користувача та асинхронну комунікацію з серверним додатком.

Програмна документація є невід'ємною частиною програмного забезпечення, що формалізує вимоги та описує архітектуру. Для даного проєкту документація була підготовлена на етапах аналізу та проєктування з використанням CASE-засобів, що імітують роботу в середовищі Enterprise Architect. Деталізований опис функціональних можливостей, сценаріїв використання, послідовності взаємодії компонентів та структури класів наведено у другому розділі даної роботи у вигляді діаграм прецедентів (Рис. 2.5), послідовності (Рис. 2.6-2.12) та класів (Рис. 2.13).

Системне програмне забезпечення. Комплекс програмних засобів підібрано з метою забезпечення гнучкості, масштабованості та мінімізації витрат на адміністрування.

- Операційні системи. На серверній стороні використовується ОС Linux у вигляді віртуалізованих контейнерів, що надаються платформою Heroku. Цей вибір є стандартом для розгортання веб-додатків на Python. Клієнтська частина є незалежною від ОС і коректно функціонує в будь-якій сучасній операційній системі (Windows, macOS, Linux, Android) за наявності веб-браузера.
- Засоби розробки та основа сайту. Для проєктування архітектури та підготовки програмної документації використовувались CASE-засоби (наприклад, Enterprise Architect), що дозволило створити формалізовані UML-діаграми. Основою системи є самостійно створений сайт, а не готова CMS. Таке рішення обґрунтоване потребою в унікальному функціоналі та високій продуктивності. Для цього були використані сучасні веб-технології: React.js для клієнтської частини та Python з фреймворком Flask для серверної.
- Спеціалізоване та інструментальне ПЗ. Ключовим спеціалізованим компонентом є програма обробки мовних сигналів — модель OpenAI Whisper, обрана за її високу точність транскрипції українською мовою. Пакети для глибокого аналізу неструктурованих даних (на кшталт Deductor, NeuroPro) в проєкті не застосовуються. В якості

інструментального ПЗ на сервері виступає інтерпретатор Python, а на клієнті — JavaScript-рушій веб-браузера.

- Сервісне ПЗ та захист інформації. Більшість завдань, пов'язаних із сервісними програмами, технічним обслуговуванням та захистом серверної інфраструктури (антивірусний захист, моніторинг, оновлення безпеки, захист від мережесих атак), делеговано хмарним платформам Heroku та Netlify. Це є однією з головних переваг обраної архітектури. Захист даних на клієнтському рівні (антивірус) залишається відповідальністю кінцевого користувача.

Прикладне програмне забезпечення. Фундаментом для розробки виступає ПЗ загального призначення, а саме бібліотеки та фреймворки, що прискорюють розробку: Flask для побудови веб-сервера, SQLAlchemy для інтеграції з базою даних, та React.js для створення інтерактивного користувацького інтерфейсу. За своєю суттю, розроблена система також є ПЗ для роботи в глобальних мережах, оскільки її архітектура передбачає взаємодію клієнта та сервера через мережу Інтернет. Адміністративні функції системи, передбачені для ролі «Адміністратор», можна віднести до ПЗ для організації обчислювального процесу.

Вибір засобів розробки повністю відповідає сучасним вимогам до швидкої розробки додатків (RAD). Розробка велася у сучасному інтегрованому середовищі розробки (IDE), наприклад, Visual Studio Code, що забезпечує необхідні інструменти для кодування, зневадження та тестування. Використання фреймворку Flask та бібліотеки SQLAlchemy гарантує високий ступінь інтеграції з обраною СКБД PostgreSQL. Проєктування користувацького інтерфейсу на основі заздалегідь підготовленого візуального прототипу у Figma та подальша реалізація за допомогою компонентної архітектури React.js дозволили значно прискорити процес створення клієнтської частини, що відповідає принципам швидкого візуального проєктування.

3.4. Результати реалізації інформаційної підсистеми

Суть оригінальної пропозиції, що лежить в основі даної бакалаврської роботи, полягає у розробці та проектуванні інформаційної системи цифрової аудіотеки, архітектура якої комплексно поєднує класичний функціонал для управління аудіоконтентом з інноваційним модулем штучного інтелекту для автоматичної транскрипції. Наукова новизна полягає в системному підході до інтеграції сучасної моделі розпізнавання мовлення (OpenAI Whisper) в єдиний, зручний для користувача програмний продукт. Це дозволяє вирішити актуальну проблему обмеженої доступності та пошукових можливостей аудіоконтенту, що було виявлено при аналізі існуючих рішень.

Практична цінність отриманих результатів полягає у створенні функціонального прототипу системи, що значно розширює можливості взаємодії з аудіо. На відміну від багатьох існуючих платформ, де функція транскрипції або відсутня, або є обмеженою, розроблена система робить аудіоконтент доступним для людей з вадами слуху, а також відкриває потенціал для повнотекстового пошуку всередині аудіозаписів. Результати апробації показали високу достовірність, оскільки точність транскрипції, забезпечена моделлю Whisper, є однією з найкращих на ринку. Основне припущення, прийняте в проєкті, — це стабільна робота та доступність зовнішнього API OpenAI, що є типовим для систем, інтегрованих зі сторонніми сервісами.

На даному етапі проєкт має ступінь готовності функціонального прототипу (Minimum Viable Product), що демонструє життєздатність основної ідеї та готовий до подальшого розвитку. Ключові розроблені інтерфейси системи, що підтверджують її працездатність, представлено на Рис. 3.2-3.. Для перетворення прототипу на повноцінний продукт рекомендуються наступні удосконалення:

Розробка повнотекстового пошуку. Інтеграція пошукового рушія, наприклад Elasticsearch, або використання вбудованих можливостей PostgreSQL (pg_trgm) для реалізації пошуку за ключовими словами безпосередньо у згенерованих транскрипціях.

Впровадження функціоналу ідентифікації спікерів (Speaker Diarization). Додавання можливості розрізняти та позначати у тексті різних мовців, що значно покращить читабельність діалогів у подкастах.

Розробка нативних мобільних додатків. Створення окремих додатків для платформ iOS та Android з використанням таких технологій, як React Native або Flutter, для забезпечення кращого користувацького досвіду на мобільних пристроях.

Оптимізація витрат. Дослідження та впровадження механізмів кешування та вибіркової транскрипції для оптимізації фінансових витрат на використання зовнішніх AI-сервісів.

Реалізація цих рекомендацій дозволить перетворити розроблений прототип на конкурентоспроможний та багатofункціональний продукт, що відповідає сучасним вимогам ринку цифрового аудіоконтенту.

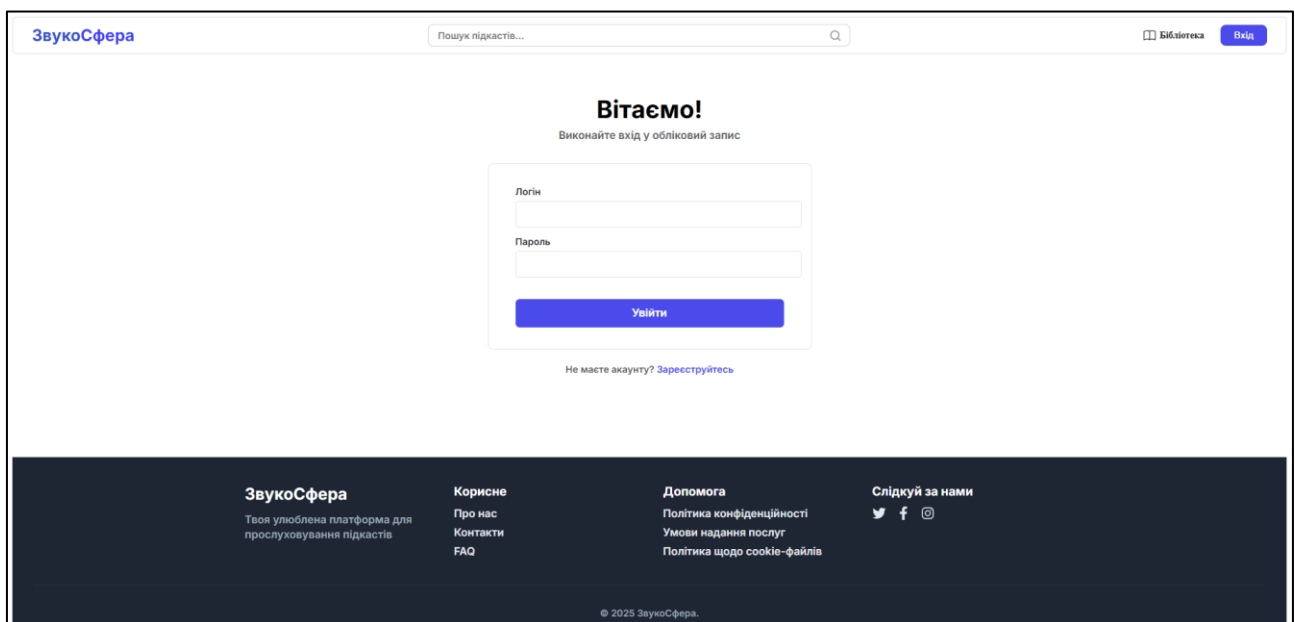


Рисунок 3.2 - Сторінка користувацького інтерфейсу “Авторизація”

Джерело: сформовано автором на основі виконаного дослідження

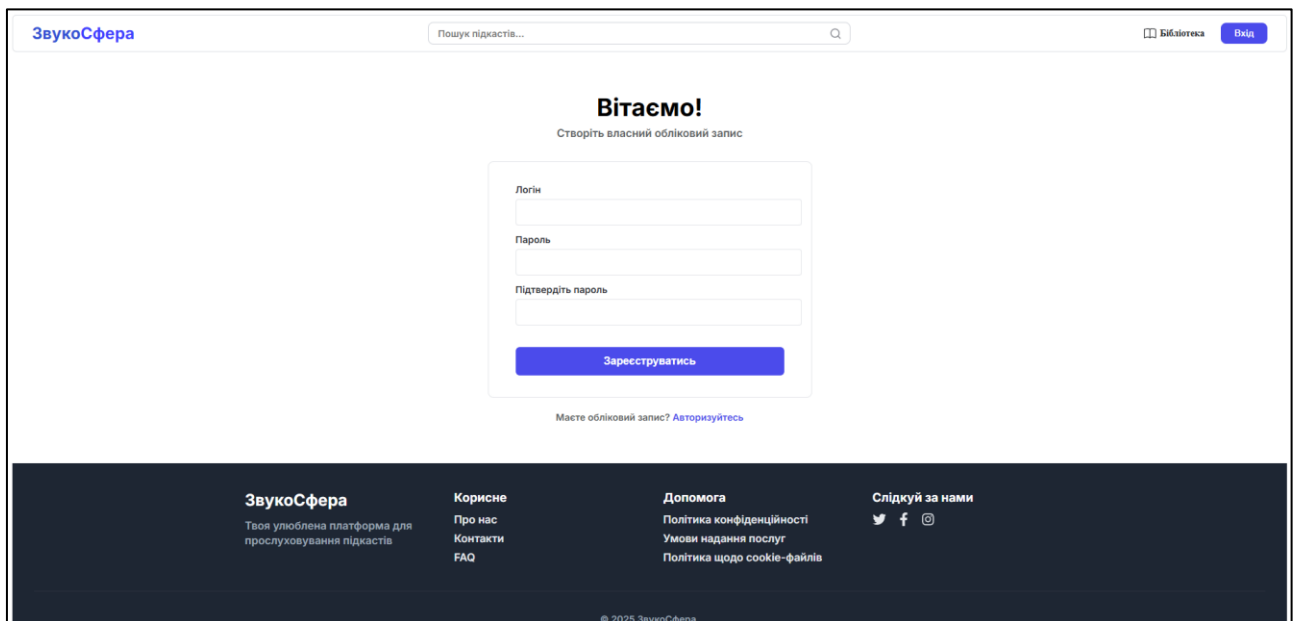


Рисунок 3.3 - Сторінка користувацького інтерфейсу «Реєстрація»

Джерело: сформовано автором на основі виконаного дослідження

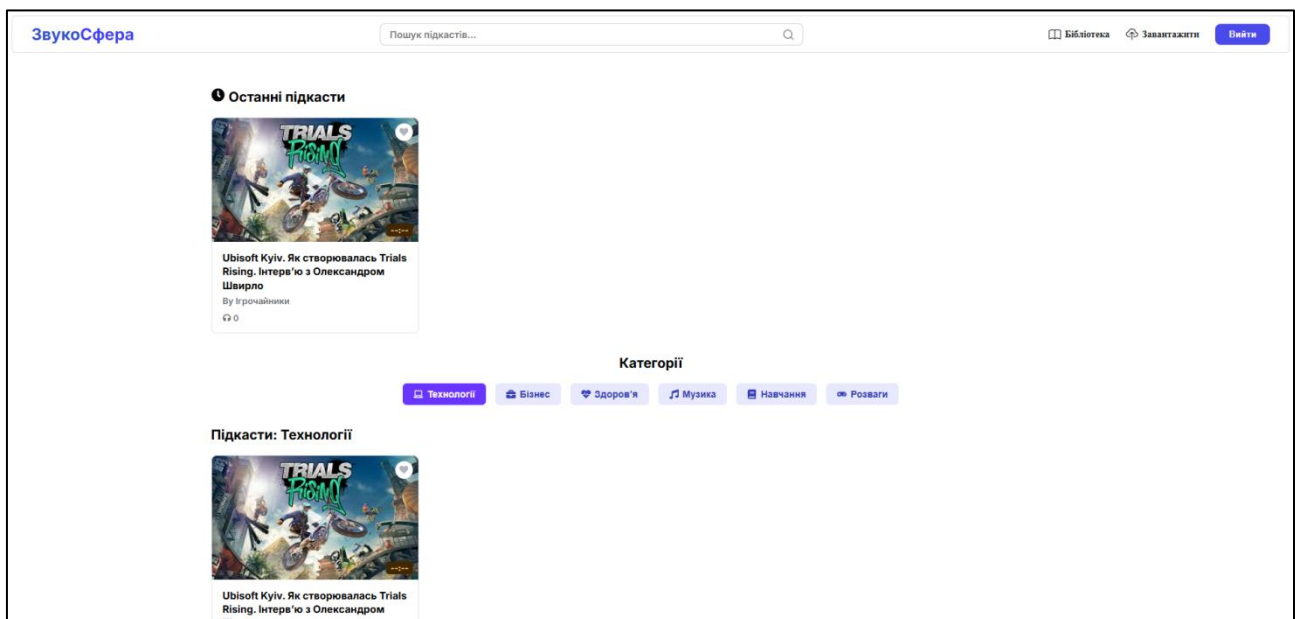


Рисунок 3.4 – Сторінка «Категорії»

Джерело: сформовано автором на основі виконаного дослідження

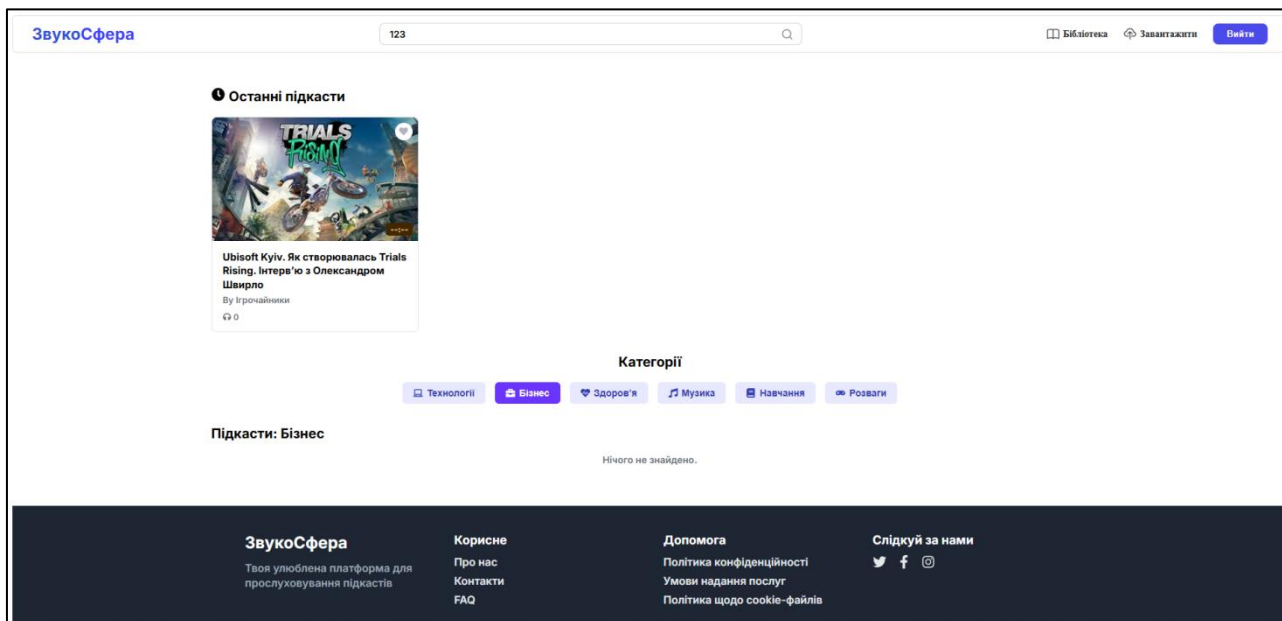


Рисунок 3.5 - Сторінка «Категорії» (нижня частина)

Джерело: сформовано автором на основі виконаного дослідження

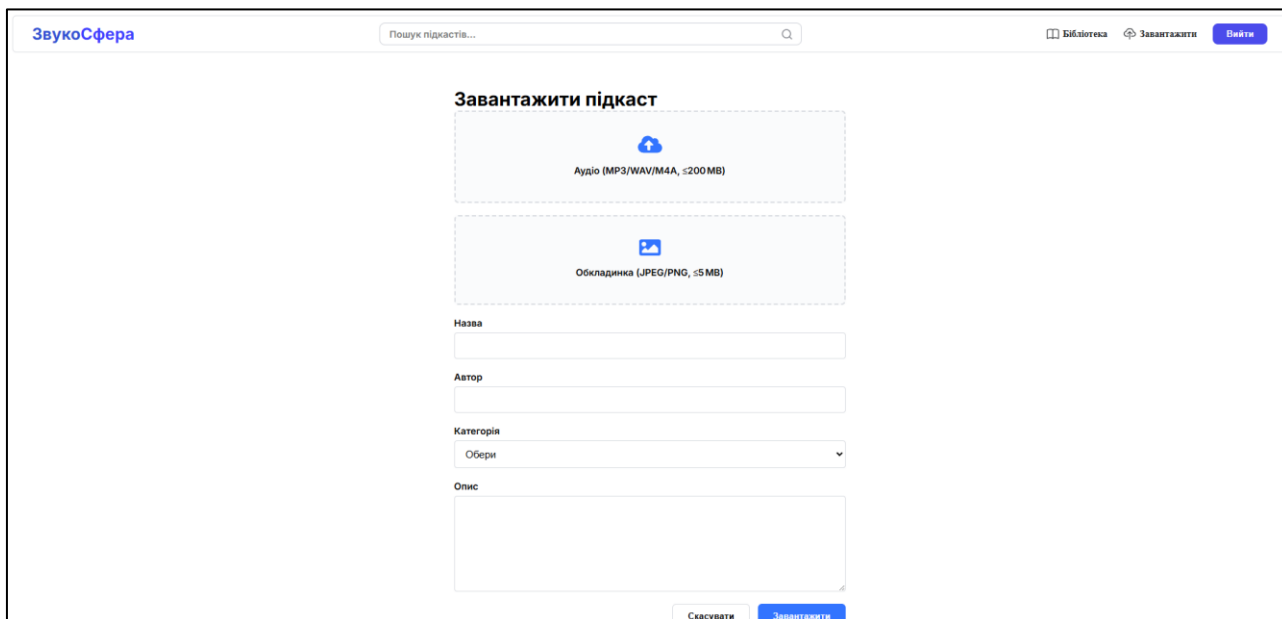


Рисунок 3.6 - Сторінка користувацького інтерфейсу «Завантаження подкасту»

Джерело: сформовано автором на основі виконаного дослідження

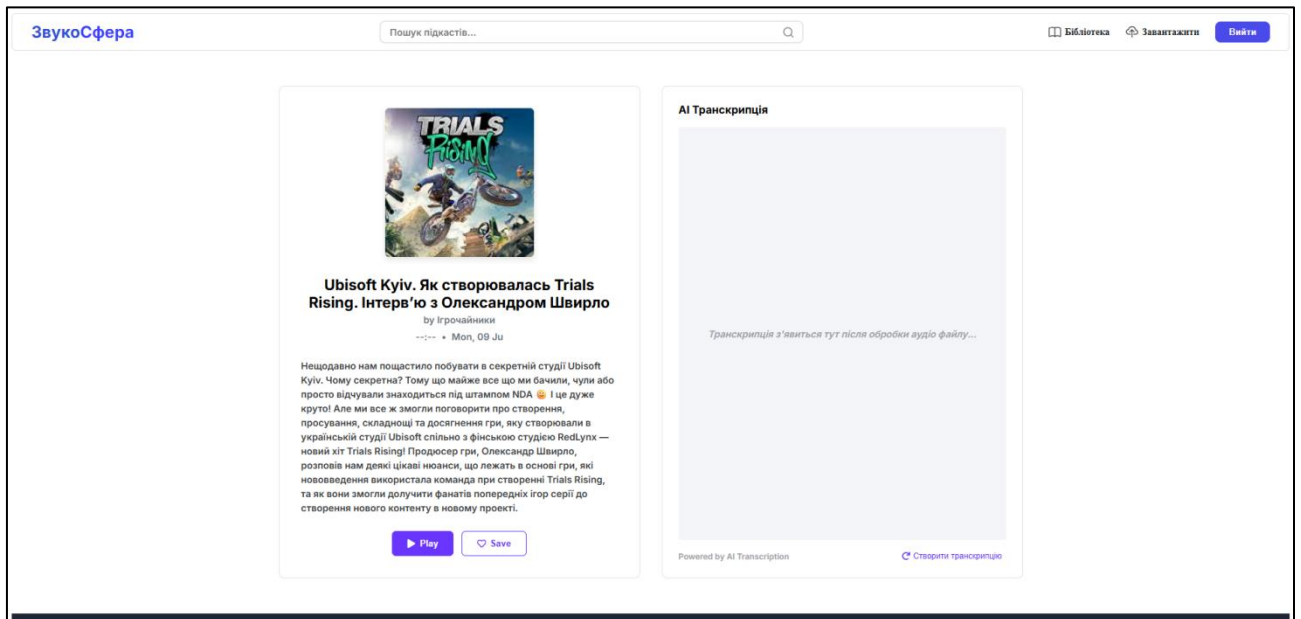


Рисунок 3.7 - Сторінка користувацького інтерфейсу
«Прослуховування подкасту»

Джерело: сформовано автором на основі виконаного дослідження

ВИСНОВКИ

У ході виконання випускної бакалаврської роботи було успішно досягнуто поставленої мети — спроектовано та розроблено інноваційну інформаційну систему для забезпечення роботи цифрової аудіотеки, ключовою особливістю якої є інтегрований модуль штучного інтелекту для транскрипції подкастів. Було створено функціональний прототип, що підтвердив життєздатність та актуальність обраної концепції, а також продемонстрував практичне застосування сучасних інформаційних технологій для вирішення завдань у сфері управління цифровим контентом.

На першому етапі було проведено всебічний аналіз предметної галузі, що включав дослідження ринку цифрових аудіотек та існуючих платформ, таких як Spotify, Apple Music та SoundCloud. Аналіз дозволив виявити їхні переваги та недоліки, а також визначити вільну нішу для системи, яка б пропонувала користувачам не лише зручне сховище для аудіо, але й інструменти для глибокої обробки контенту, зокрема високоточну транскрипцію, що і обґрунтувало актуальність даної роботи.

На етапі проектування було сформовано повний набір функціональних та нефункціональних вимог до системи, розроблено її архітектуру та деталізовані моделі поведінки й структури з використанням нотацій UML та SysML. Було обґрунтовано вибір технологічного стеку, що включає Python та фреймворк Flask для серверної частини, бібліотеку React.js для клієнтської частини та СКБД PostgreSQL для зберігання даних. Цей етап заклав міцну теоретичну та методологічну основу для подальшої практичної реалізації.

В рамках практичної реалізації, що є особистим внеском автора даної роботи, було розроблено ключові програмні модулі системи: реєстрації та автентифікації користувачів, завантаження та керування аудіофайлами, а також відтворення контенту. Було спроектовано та реалізовано структуру бази даних у PostgreSQL та успішно інтегровано зовнішній AI-сервіс OpenAI Whisper для автоматичного створення текстових транскрипцій.

Апробація розробленого прототипу на контрольному прикладі підтвердила працездатність основного інноваційного функціоналу. Система успішно обробила завантажений аудіофайл, згенерувала його текстову версію з високою точністю та коректно відобразила її в інтерфейсі користувача. Головною перевагою розробленої системи є саме ця безшовна інтеграція, що підвищує доступність контенту та його цінність для кінцевого споживача.

Водночас, варто відзначити, що представлена система на даному етапі має статус мінімально життєздатного продукту (MVP) та має певні обмеження. Основними з них є залежність від зовнішнього платного API, відсутність повнотекстового пошуку по згенерованих транскрипціях та відсутність нативних мобільних клієнтів, що може обмежувати зручність використання на мобільних пристроях.

Для подальшого розвитку представлених напрацювань та їх трансформації у повноцінний продукт рекомендуються наступні удосконалення. По-перше, доцільно реалізувати повнотекстовий пошук шляхом інтеграції пошукового рушія, наприклад Elasticsearch, для знаходження інформації за ключовими словами безпосередньо у згенерованих транскрипціях. По-друге, варто додати функціонал ідентифікації різних спікерів у подкастах, що значно покращить читабельність діалогів. Крім того, важливим кроком є розробка нативних мобільних додатків для платформ iOS та Android. Нарешті, слід провести дослідження та впровадження механізмів оптимізації витрат на використання зовнішніх AI-сервісів.

Таким чином, у випускній бакалаврській роботі було вирішено поставлені наукові та практичні завдання. Результати демонструють комплексний підхід до створення сучасної інформаційної системи та можуть слугувати надійною основою для подальших наукових досліджень або розробки комерційного продукту на ринку цифрових медіа.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гайна Г. А. Основи проектування баз даних : навчальний посібник. Київ : Кондор, 2024. 208 с. ISBN 978-966-627-117-6.
2. Гордєєв А. С. Проектування баз даних та баз знань : конспект лекцій для студентів спеціальності 186 "Видавництво та поліграфія" першого (бакалаврського) рівня. Харків : ХНЕУ ім. С. Кузнеця, 2022. 180 с.
3. Інформаційні системи і технології в економіці : підручник / О. М. Галузевський та ін. ; за ред. В. С. Пономаренка. Київ : КНЕУ, 2017. 213 с. ISBN 978-966-926-157-1.
4. Малець О.-С. І., Смотр О. О., Пелешко Д. Д., Пилипенко В. Сучасні методи та проблеми цифрового водяного маркування аудіоконтенту. Вісник Львівського державного університету безпеки життєдіяльності. 2024. № 30. С. 98–109. DOI: <https://doi.org/10.32447/20784643.30.2024.10>.
5. Мар'їна О. Ю. Бібліотеки України в цифровому медіапросторі: теоретико-методологічні засади розвитку : автореф. дис.... д-ра наук із соц. комунікацій : 27.00.03 – книгознавство, бібліотекознавство, бібліографознавство / Нац. б-ка України ім. В. І. Вернадського. Київ, 2018. 40 с. (На основі)
6. Moises AI. Транскрипція аудіо за допомогою ШІ. Moises.ai. URL: <https://moises.ai/uk/features/ai-audio-transcription/> (дата звернення: 2024-10-26).
7. Норман Д. А. Дизайн звичних речей / пер. з англ. М. Бакалова. Харків : Клуб Сімейного Дозвілля, 2019. 320 с. ISBN 978-617-12-4760-4.
8. Пасічник В. В., Резніченко В. А. Організація баз даних та знань : підручник. Київ : Видавнича група BHV, 2006. 384 с. ISBN 966-552-164-0. (На основі)
9. Проектування інформаційних систем : посібник / П. В. Бідюк та ін. ; за ред. В. С. Пономаренка. Київ : Академія, 2002. 488 с. ISBN 966-580-149-X.
10. Тарануха В. Ю. Модифікація n-грамної моделі, заснованої на класах, для розпізнавання слов'янських мов. Вісник Київського національного

- університету імені Тараса Шевченка. Серія: фізико-математичні науки. 2014. Вип. 1. С. 193–196.
- 11.Томашевський В. М. Моделювання систем : підручник. Київ : Видавнича група ВНУ, 2005. 352 с. ISBN 966-552-120-9.
 - 12.Туголукова Є. В. Система розпізнавання української мови : дипломна робота бакалавра : 6.050101 Комп'ютерні науки / наук. кер. М. В. Дідковська ; КПІ ім. Ігоря Сікорського. Київ, 2019. 78 с.
 - 13.Andreyev A. Quantization for OpenAI's Whisper Models: A Comparative Analysis. arXiv preprint arXiv:2503.09905. 2025. 7 p.
 - 14.Cooper A., Reimann R., Cronin D., Noessel C. About Face: The Essentials of Interaction Design. 4th ed. Wiley, 2014. 720 p. ISBN 978-1118766576.
 - 15.Edison Research. The Infinite Dial 2024. Edison Research, 2024. URL: <https://www.edisonresearch.com/the-infinite-dial-2024/> (дата звернення: 2024-10-26).
 - 16.eSparkBiz. React With Flask: Developing Scalable Full-Stack Applications. eSparkBiz. URL: <https://www.esparkinfo.com/software-development/technologies/reactjs/react-with-flask> (дата звернення: 2024-10-26).
 - 17.Galin D. Software Quality: Concepts and Practice. Wiley-IEEE Computer Society Pr, 2018. 720 p. ISBN 978-1119134497.
 - 18.Grinberg M. Flask Web Development: Developing Web Applications with Python. 1st ed. O'Reilly Media, 2014. 256 p. ISBN 978-1449372620.
 - 19.Lerch A. An Introduction to Audio Content Analysis: Music Information Retrieval Tasks and Applications. 2nd ed. Wiley-IEEE Press, 2022. 464 p. ISBN 978-1119890973.
 - 20.Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017. 432 p. (Українське видання: Мартін Р. К. Чиста архітектура: Мистецтво розробки програмного забезпечення / пер. з англ. Київ : Фабула, 2019. 368 с. ISBN 978-617-09-5286-8).

21. Meta and community. React Documentation. URL: <https://react.dev/> (дата звернення: 2024-10-26).
22. Object Management Group. OMG Systems Modeling Language (OMG SysML®) Version 1.6. OMG, 2019. URL: <https://www.omg.org/spec/SysML/1.6/PDF> (дата звернення: 2024-10-26).
23. Pallets Projects. Flask Documentation (Version 2.3.x). URL: <https://flask.palletsprojects.com/> (дата звернення: 2024-10-26).
24. PostgreSQL Global Development Group. PostgreSQL 16 Documentation. URL: <https://www.postgresql.org/docs/16/index.html> (дата звернення: 2024-10-26).
25. Python Software Foundation. Python 3.12.6 documentation. URL: <https://docs.python.org/3/> (дата звернення: 2024-10-26).
26. Radford A. et al. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv preprint arXiv:2212.04356. 2022. URL: <https://arxiv.org/abs/2212.04356> (дата звернення: 2024-10-26).
27. Richardson L., Amundsen M., Ruby S. RESTful Web APIs: Services for a Changing World. O'Reilly Media, 2013. 404 p. ISBN 978-1449358068.
28. Shneiderman B. et al. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th ed. Pearson, 2017. 616 p. ISBN 978-0134380384.
29. Sommerville I. Software Engineering. 9th ed. Pearson, 2011. 792 p. ISBN 978-0137035151.
30. Stefanov S. React: Up & Running: Building Web Applications. O'Reilly Media, 2016. 222 p. ISBN 978-1491931820.

ДОДАТКИ

Додаток А - Специфікація функціональних та нефункціональних вимог до ІС

Ідентиф.	Стисла назва вимоги	Повна назва вимоги	Тип	Джерело	Статус	Складність	Пріоритетність	Ризикованість	Примітки
FR-001	Управління користувачами	Система повинна забезпечувати керування обліковими записами користувачів.	Функціональна	Аналіз предметної галузі, вимоги користувачів	Узгоджено	Середня	Висока	Низька	Ця батьківська вимога охоплює всі аспекти взаємодії користувача з системою на рівні його облікового запису
FR-001.1	Реєстрація користувача	Надання можливості створення нового облікового запису користувачем.	Функціональна	Вимоги користувачів	Узгоджено	Низька	Критична	Низька	Користувачі повинні мати змогу зареєструватися за допомогою електронної пошти та пароля. Передбачити верифікацію електронної пошти.
FR-001.2	Авторизація користувача	Забезпечення безпечного входу в систему для зареєстрованих користувачів.	Функціональна	Вимоги користувачів	Узгоджено	Низька	Критична	Низька	Користувачі повинні мати змогу входити за допомогою своїх облікових даних. Система має підтримувати механізм "запам'ятати мене".
FR-002	Управління аудіоконтентом	Система повинна надавати повний набір функцій для роботи з цифровими аудіофайлами.	Функціональна	Аналіз предметної галузі, прототип Figma	Узгоджено	Висока	Критична	Середня	Ця вимога є ключовою для будь-якої аудіотеки, охоплюючи всі аспекти взаємодії системи з аудіофайлами від моменту їх завантаження до відтворення та організації.
FR-002.1	Завантаження аудіофайлів	Користувачі повинні мати можливість завантажувати цифрові аудіофайли.	Функціональна	Вимоги користувачів	Узгоджено	Середня	Критична	Низька	Система повинна підтримувати завантаження файлів у форматах MP3, WAV, FLAC, M4A. Має бути індикатор прогресу завантаження та повідомлення про успішне/невдале завантаження.
FR-002.2	Пошук та фільтрація аудіо	Надання розширених можливостей пошуку та фільтрації аудіоконтенту.	Функціональна	Вимоги користувачів	Узгоджено	Висока	Висока	Низька	Пошук повинен здійснюватися за назвою, автором, описом, жанром. Фільтрація за жанром, тривалістю, датою завантаження, рейтингом. Результати пошуку та фільтрації мають відображатися з високою швидкістю.
FR-002.3	Відтворення аудіо	Функціонал для потокового відтворення аудіофайлів у системі.	Функціональна	Вимоги користувачів	Узгоджено	Низька	Критична	Низька	Вбудований аудіоплеєр повинен підтримувати базові функції: відтворення, пауза, перемотування вперед/назад, регулювання гучності.
FR-003	Функціонал AI-модуля транскрипції	Система повинна інтегрувати AI-модуль для автоматичної транскрипції аудіо.	Функціональна	Нововведення, вимоги розробки	Узгоджено	Висока	Критична	Висока	Ця вимога визначає ключову інноваційну особливість системи – автоматичне перетворення мови в текст, що відкриває нові можливості для взаємодії з аудіоконтентом.

Продовження додатку А

Ідентиф.	Стисла назва вимоги	Повна назва вимоги	Тип	Джерело	Статус	Складність	Пріоритетність	Ризикованість	Примітки
FR-003.1	Автоматична транскрипція	Інтеграція або розробка AI-модуля для перетворення аудіомовлення в текст.	Функціональна	Нововведення, вимоги розробки	Узгоджено	Висока	Критична	Висока	Модуль повинен підтримувати транскрипцію аудіофайлів, переважно подкастів. Передбачити підтримку української мови. Точність транскрипції не менше X%. Обробка повинна бути асинхронною.
FR-003.2	Відображення транскрипцій	Зручне відображення транскрипцій під час відтворення аудіо.	Функціональна	Вимоги користувачів, прототип Figma	Узгоджено	Середня	Висока	Низька	Транскрипції повинні відображатися паралельно з відтворенням аудіо, з можливістю синхронізації тексту з поточним моментом відтворення.

Додаток Б - Аналіз плагиату виконаних досліджень



Дата звіту 6/18/2025
Дата редагування ---



Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок
ІНФОРМАЦІЙНА СИСТЕМА ЗАБЕЗПЕЧЕННЯ РОБОТИ ЦИФРОВОЇ АУДИОТЕКИ

Автор
Науковий керівник / Експерт
Усик Владислав ВолодимировичКривошеєв К.В.

підрозділ
кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагиат. Звіт має аналізувати компетентна / уповноважена особа.



25
Довжина фрази для коефіцієнта подібності 2



12566
Кількість слів



102199
Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		2
Інтервали		0
Мікропробіли		32
Білі знаки		0
Парафрази (SmartMarks)		32

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагиату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз		Копію тексту
ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/server/api/core/bitstreams/add9ce43-9c83-4412-bccd-41d87f874e83/content	45 0.36 %
2	https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content	37 0.29 %
3	https://ir.kneu.edu.ua/bitstreams/b20c497c-320d-46ce-8d1d-a56f3c2eec3f/download	20 0.16 %
4	https://thelib.info/matematika/1687355-harakteristika-zadachi/	18 0.14 %
5	https://ir.kneu.edu.ua/server/api/core/bitstreams/8886d65e-0996-456f-8b04-70b8d9683e0e/content	18 0.14 %