МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

БАНДУРА Ігор Олександрович

Метод трансформації відеоматеріалів на мову користувача для гіперконвергентної платформи / A method for transforming video materials into user language for a hyper-converged platform

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи
КНм-21
І. О. Бандура
_____

Науковий керівник:
к.т.н., доцент І. В. Турченко
_____

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
Завідувач кафедри
_____ М.П. Комар

ТЕРНОПІЛЬ - 2022

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «магістр»
спеціальність: 122 «Комп'ютерні науки»
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«____» _____20__року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Бандурі Ігору Олександровичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Метод трансформації відеоматеріалів на мову користувача для гіперконвергентної платформи / A method for transforming video materials into user language for a hyper-converged platform
керівник роботи к.т.н., доцент І.В. Турченко
затверджені наказом по університету від 31 грудня 2021 року № 606.

2. Строк подання студентом кваліфікаційної роботи 16 листопада 2022 року.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити
– дослідити існуючі підходи до перетворення відеоматеріалів на мову користувача;
– провести аналіз літературних джерел з досліджуваної тематики;
– проаналізувати функціонування гіперконвергентної платформи;
– розробити метод трансформації відеоматеріалів на мову користувача;
– реалізувати метод трансформації та дослідити функціонування системи.

5. Перелік графічного матеріалу у роботі
– схема алгоритму трансформації відеоматеріалів на мову користувача
– схема алгоритму озвучення тексту

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 11 жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області і постановка задачі дослідження	12.2021 р. – 03.2022 р.	
2	Метод трансформації відеоматеріалів на мову користувача	03.2022 р. – 05.2022 р.	
3	Реалізація методу трансформації відеоматеріалів на мову користувача	05.2022 р. – 11.2022р.	
4	Повне завершення та представлення кваліфікаційної роботи на кафедру	16.11.2022 р.	

Студент _____ І.О. Бандура

підпис

Керівник роботи _____ к.т.н., доцент І.В. Турченко

підпис

РЕЗЮМЕ

Кваліфікаційна робота на тему «Метод трансформації відеоматеріалів на мову користувача для гіперконвергентної платформи» на здобуття ступеня вищої освіти «Магістр» зі спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки» написана обсягом 93 сторінок і містить 19 ілюстрацій, 3 додатки та 56 використаних джерел.

Метою кваліфікаційної роботи є розроблення методу трансформації відеоматеріалів на мову користувача та його реалізація.

Методи досліджень: теорія штучних нейронних мереж, системний аналіз, методи розпізнавання мови.

Результати дослідження: запропоновано метод трансформації відеоматеріалів на мову користувача, який на відміну від існуючих, дозволяє змінювати проміжні результати етапів трансформації згідно вимог користувача, що дозволить покращити якість перекладу відеоматеріалу.

Результати роботи можуть бути використані для розширення функціоналу гіперконвергентної платформи або інших систем, що потребують сервісу трансформації відеоматеріалів на мову, яка зручна користувачеві.

Ключові слова: ТРАНСФОРМАЦІЯ, ВІДЕОМАТЕРІАЛИ, РОЗПІЗНАВАННЯ МОВЛЕННЯ, ОЗВУЧЕННЯ ТЕКСТУ, ПЕРЕКЛАД, НЕЙРОННІ МЕРЕЖІ, АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

ABSTRACT

Qualification work on the topic «A method for transforming video materials into user language for a hyper-converged platform» for the degree of «Master» in the specialty 122 «Computer Science» of the educational and professional program «Computer Science» is written in 93 pages and contains 19 illustrations, 3 annexes and 56 used sources.

The purpose of the qualification work is to develop a method of transformation of video materials into user's language and its implementation.

Research methods: artificial neural networks, system analysis, language recognition methods.

Research results: The method of transformation of video materials into a user's language is proposed. This method unlike existing ones, allows to change the intermediate results of the transformation stages according to the user's requirements. This will improve the quality of video translation.

The results of the work can be used to expand functionality of the hyperconverged platform or other systems that require the service of transformation of video materials into a language that is convenient for user.

Keywords: TRANSFORMATION, VIDEO MATERIALS, SPEECH RECOGNITION, TEXT SYNTHESIZING, TRANSLATION, NEURAL NETWORKS, SOFTWARE ARCHITECTURE.

ЗМІСТ

ВСТУП

Актуальність теми. Сьогодні дуже популярними є ведення власних блогів, проведення онлайн трансляцій чи просто поширення у соціальних мережах відео. Звичайно, викладаючи відео тільки одною мовою, їх власники втрачають багатьох глядачів, які б могли допомогти їм у реалізації деяких проектів, монетизації власних продуктів, тощо. Сервіси для трансформації відеоматеріалів на різні мови стануть їм у нагоді. Спостерігається велика увага до платформ, що надають можливість обмінюватись повідомленнями, відеоматеріалами, проводити онлайн-трансляції та веб-конференції, планувати та проводити події з можливістю переводу їх в онлайн. Тому тема кваліфікаційної роботи є актуальною.

Мета і завдання дослідження. Метою є розроблення методу трансформації відеоматеріалів на мову користувача для гіперконвергентної платформи та його реалізація.

Для реалізації мети необхідно вирішити наступні завдання:

− дослідити існуючі підходи до перетворення відеоматеріалів на мову користувача;

− провести аналіз літературних джерел з досліджуваної тематики;

− проаналізувати функціонування гіперконвергентної платформи «Iconnect»;

− розробити та реалізувати метод трансформації відеоматеріалів на мову користувача;

− дослідити функціонування системи.

Методи досліджень: : теорія штучних нейронних мереж, системний аналіз, методи розпізнавання мови.

Об’єкт дослідження: процес обробки інформації системи трансформації системи відеоматеріалів на мову користувача.

Предмет дослідження: метод трансформації відеоматеріалів на мову користувача.

Наукова новизна одержаних результатів: запропоновано метод трансформації відеоматеріалів на мову користувача, який на відміну від існуючих, дозволяє змінювати проміжні результати етапів трансформації згідно вимог користувача, що дозволить покращити якість перекладу відеоматеріалу.

Практичне значення отриманих результатів: результати роботи можуть бути використані для розширення функціоналу гіперконвергентної платформи або інших систем, що потребують сервісу трансформації відеоматеріалів на мову, яка зручна користувачеві.

Публікації та апробація кваліфікаційної роботи. Результати дослідження будуть опубліковані та апробовані в матеріалах міжнародної науково-практичної інтернет-конференції "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення", 08-09 грудня 2022 р., м. Тернопіль (Додаток Г).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Підходи до перетворення відеоматеріалів на мову користувача

На даний момент спостерігається час розквіту інформаційних технологій, проникнення яких відбувається в усі галузі і сфери життєдіяльності людини. Важливе місце серед інформаційних технологій посідають також онлайн-конференції, прямі відео-трансляції, що дозволяють людям обмінюватись думками, ідеями, знаннями, тощо. Думки інших людей завжди були важливою частиною інформації для більшості із нас, оскільки саме завдяки цьому ми навчаємось та розвиваємось.

Під час карантину звичайно зріс попит на онлайн-конференцій, перегляду відео онлайн, віддаленого навчання за допомогою таких систем, як «Zoom» [1], «Google Meet» [2], тощо. Більшість компаній успішно перевели всіх своїх працівників на віддалену роботу. І навіть після закінчення карантину багато компаній вирішили і на далі працювати віддалено, і стали більш лояльними до найму працівників з інших країн, проте для цього потрібно, щоб працівник, звичайно знав мову тієї країни, де розташована компанії, щоб мати змогу успішно комунікувати з колегами та іншими працівниками цієї компанії.

У світі за час карантину, а в Україні в умовах війни, онлайн-навчання набуло великої популярності, оскільки це найзручніший спосіб для людей вивчати щось нове. Проте, як вже згадувалось раніше, не у всіх сферах можна знайти хороші відеоматеріали для ознайомлення та вивчення на зручній для користувача мові. Особливо у сфері інформаційних технологій важко знайти якісні навчальні матеріали українською мовою, зокрема про сучасні новітні технології, методи, алгоритми або принципи роботи деяких систем.

У сфері розваг також є цей мовний бар’єр, оскільки переклад текстових, відео- чи аудіо-матеріалів займає доволі багато часу та вартує немалих коштів, саме тому компанії не завжди можуть собі дозволити переклад свого продукту на велику кількість мов, а якщо й готові зробити це, то повна локалізація продукту все одно займає багато часу і може відкласти реліз продукту навіть на декілька місяців і це

в свою чергу приведе до втрати значної кількості потенційних клієнтів.

Те ж саме можна сказати про людей, які займаються різними дослідженнями та ведуть власні блоги, проводять власні онлайн трансляції або ж записують відео, де діляться різною корисною інформацією, яку було б цікаво почути людям з різних куточків Землі. Звичайно, викладаючи відео тільки на одній мові вони втрачають багатьох глядачів, які б могли допомогти їм у реалізації деяких проектів, монетизації власних продуктів, тощо. Якщо ж вже великі компанії часто відмовляються від локалізацій своїх продуктів багатьма мовами, то що можна сказати тоді про людей, які займаються своєю справою, як захопленням і не готові витрачати багато коштів на переклад власних відео-матеріалів різними мовами, тим більше на велику кількість мов.

Зрозуміло, що машинне розпізнавання мови, переклад та озвучення не замінить на 100 відсотків людину, оскільки якість буде набагато нижчою, проте завдання є не повністю автоматизувати переклад відеоматеріалів, а зробити їх набагато легшим, простішим, зменшити час, який потрібен для перекладу, а також оскільки ці питання будуть вирішені, то і вартість локалізації може зменшитись. Використання методу трансформації відео-матеріалів на мову користувача дозволить компаніям інтегрувати у свої продукти набагато більше мов, чим саме підвищать кількість зацікавлених осіб у даному продукті. Те ж саме стосується не тільки розваг, але й перекладу відеоматеріалів для онлайн-навчання і взагалі будь-яких відео.

Людям, що ведуть відеоблоги або записують різного формату відео буде набагато простіше локалізувати їх для більшої кількості людей, оскільки ціна локалізації з даним сервісом може набагато зменшитись, або ж вони можуть спробувати локалізувати власні продукти самотужки, використовуючи метод, який пропонується у кваліфікаційній роботі.

Трансформація відеоматеріалів на іншу мову відбувається у три кроки, тобто для того, щоб перекласти відео потрібно розпізнати мову з відео, а саме перетворити її у текст, перекласти текст, який розпізнали на попередньому кроці і в останньому кроці нам потрібно використати машинне озвучення тексту [3] для

того, щоб створити аудіофайл з перекладеним текстом та прикріпити його до відео.

Автоматичне розпізнавання мови використовує технології для перетворення мовних сигналів на послідовність слів або інших лінгвістичних одиниць за допомогою алгоритму, реалізованого у вигляді комп'ютерної програми. Нині системи розпізнавання мови здатні розуміти мовне введення для словників, які вміщають тисячі слів в оперативному середовищі. Мовний сигнал передає два важливих типи інформації, а саме зміст мови і гендер людини, мовлення якої було розпізнано. Розпізнавачі мови націлені на вилучення лексичної інформації з мовного сигналу незалежно від того на яких частотах та як розмовляє диктор. Розпізнавання мови диктора також пов'язане і з вилученням гендеру людини.

Ідентифікація стать актора дає змогу в подальшому використати цю інформацію, наприклад, для машинного озвучення розпізнаного тексту в іншій мові. Також розпізнавання мови дає можливість використовувати виголошену мову для перевірки особи мовця та контролю доступу до певних послуг. Розпізнавання мови також дає можливість людям з обмеженими можливостями більшу свободу, наприклад, працювати у таких галузях, як виробництво, медицина та телефонна мережа. На рисунку 1.1 [54] показано систему розпізнавання мови без ідентифікації диктора.

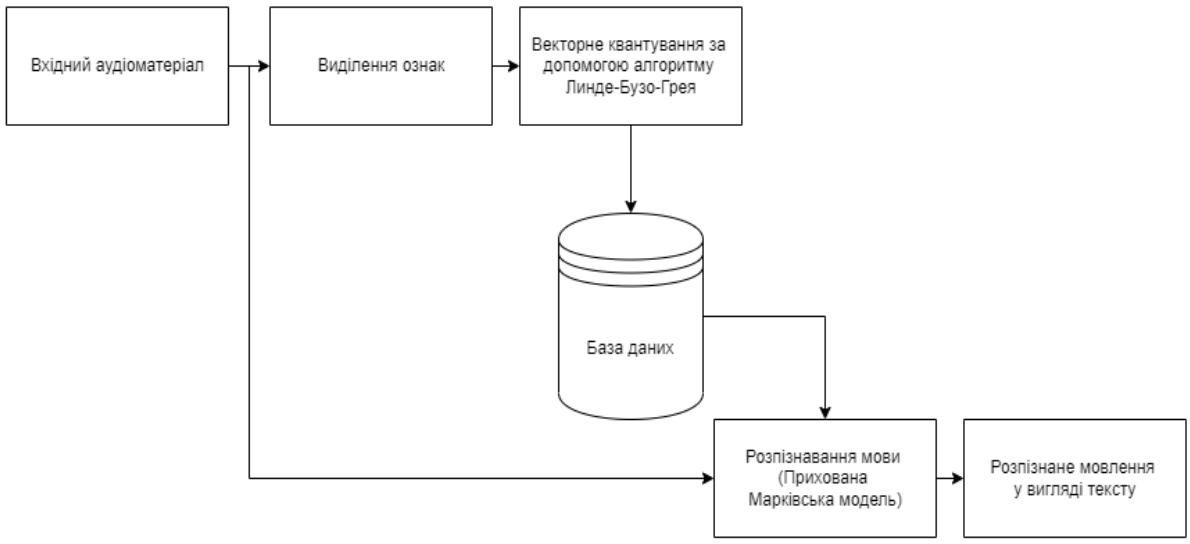

Рисунок 1.1 – Схема розпізнавання мови без індентифікації диктора

На рисунку 1.2 [54] представлена схема розпізнавання мови разом із індентифікацією диктора. За такого підходу база даних буде розділена на більш дрібні частини по відношенню до різних дикторів. Отже, швидкість розпізнавання мови покращується для відповідного диктора.

Рисунок 1.2 – Схема розпізнавання мови із індентифікацією диктора

Для того, щоб провести розпізнавання мови потрібно також багато часу, щоб розробити спеціальні моделі для нейронної мережі, що буде розпізнавати потрібну мову. Для того щоб розробити ці моделі потрібно спочатку записати всі можливі звуки, які можуть бути у данній мові, та окремі слова цією мовою. Після запису аудіофайлів потрібно вказати у тексті фонетичний розбір звуків та слів, а тоді вже просто поєднати всі ці дані у текстових файлах, які називаються словниками, і запустити процес тренування нейронної мережі. Чим більше різних слів буде у словниках, то тим краще буде працювати нейронна мережі і набагато краще буде розпізнавати мовлення з аудіоматеріалів.

Після успішного проведення трансформації аудіоматеріалу в текст потрібно провести переклад цього тексту у мову, яка зручна користувачеві. Для цього також потрібно використати нейронні мережі та набір словників для відповідних мов, у яких зберігаються відповідні слова та їх переклад.

Переклад за допомогою нейронних мереж дає можливість не просто

перекласти слова по значенню, але й зробити лінгвистичне дослідження речення та підібрати відповідний переклад або синонім, який буде краще відображати сенс речення. Це значно покращує якість перекладеного тексту та полегшує роботу при перекладі. Діаграму системи перекладу тексту представлена на рисунку 1.3.

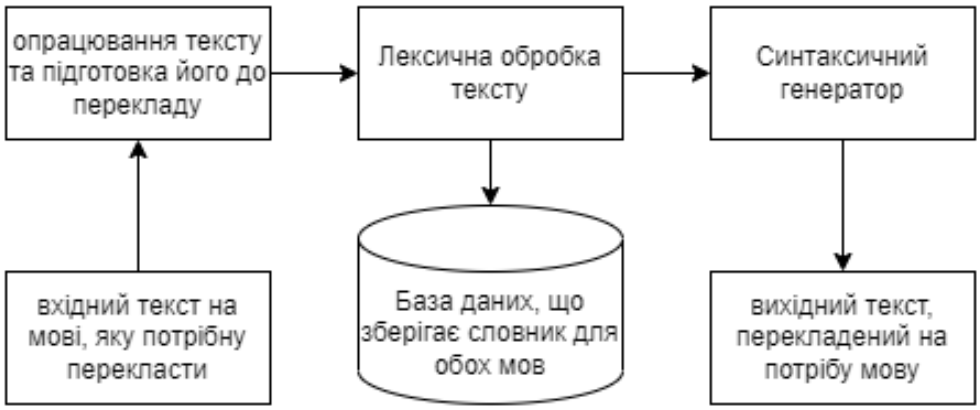

Рисунок 1.3 – Схема перекладу тексту у мову, що зручна користувачеві

Спочатку відбувається опрацювання тексту, оскіль потрібно підготувати текст до перекладу, а саме видалити непотрібні пробіли у тексті, перевести символи у потрібний регістр, порахувати кількість слів у фразі, перевірити чи дана фраза складається з одного слово, чи це ціле речення, тощо.

Після опрацювання тексту потрібно провести лексичну обробку. На даному етапі кожне слово перекладається окремо від інших слів. Для того, щоб запустити даний крок, потрібно надати базу даних, у якій будуть присутні потрібні словники для мов, адже саме звідси на цьому кроці будуть братись потрібні слова та їх переклад у потрібній мові.

Коли ж лексична обробка тексту закінчилась, тоді запускається синтаксичний генератор, робота якого полягає у тому, щоб створити вихідну фразу, перекладену на потрібну мову. На даному етапі слова, які були перекладені на попередньому кроці, опрацьовуються і складаються у фразу так, щоб вони відповідали змісту фрази, а сама фраза була граматично правильною відповідно до граматичних правил цільової мови.

Наступним етапом після перекладу мови у методі трансформації

аудіоматеріалів на мову, яка зручна користувачеві, являється етап машинного озвучення перекладеного тексту, який ми получили на попередньому кроці системи. Діаграма, що приблизно відображає систему машинного озвучення тексту, зображена на рисунку 1.4.

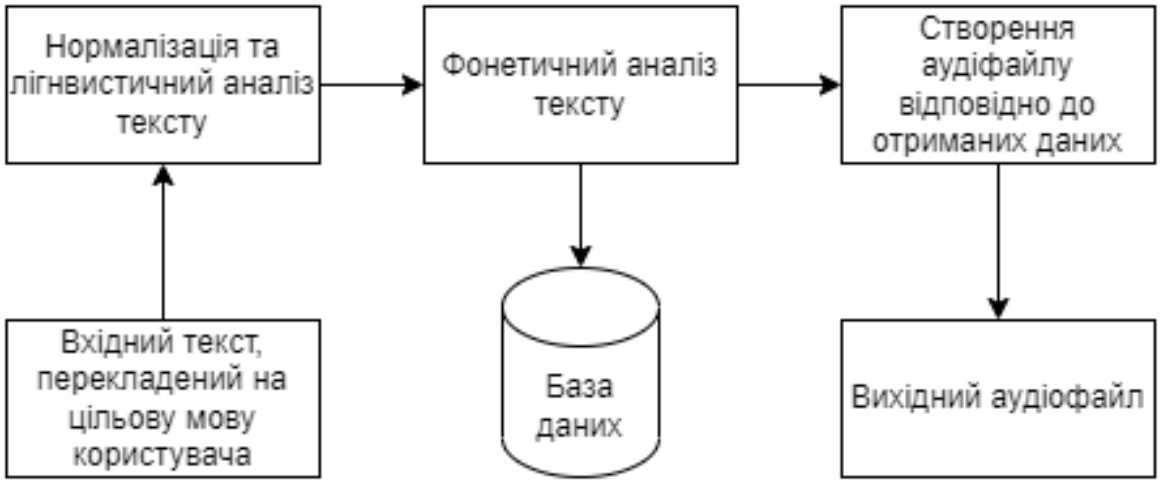

Рисунок 1.4 – Схема озвучення перекладеного тексту

У систему озвучення тексту передається текст перекладений на попередньому етапі системи трансформації аудіоматеріалів. Як тільки система озвучення тексту отримала вхідний текст, то запускається перший крок цієї системи, а саме опрацювання тексту.

На кроці опрацювання тексту система опрацьовує його, тобто, видаляє зайві пробіли, оскільки один пробіл для системи означає закінчення слова і початок наступного, тому потрібно опрацювати текст, щоб між словами завжди був один пробіл. Також на даному етапі обчислюється кількість слів у тексті, перевіряється регістр кожного символа, тощо. Також на даному етапі текст ділиться на окремі слова, щоб в подальшому було можливо провести фонетичний аналіз кожного слова окремо.

Після того, як текст опрацювався, пройшов лінгвистичний аналіз і готовий до озвучення, запускається фонетичний аналіз тексту. Для даного етапу системи нам потрібна база даних, тобто модель звуків потрібної нам мови та їх озвучення. На

даному етапі для кожного символу або набору символів у слові шукається відповідний фонетичний розбір у базі даних.

Модель звуків мови складається з файлів, у яких містяться фонеми [4], інформація про звуки, який звук відноситься до відповідного фонетичного розбору, тощо. Деякі готові моделі для озвучення тексту можна знайти у відкритому доступі, проте більшість з них неякісні, а для власного користування потрібно самому розширювати їх, оскільки слів у кожній мові велика кількість і прописати фонеми, озвучити кожне слово буде дещо проблематичним. Звичайно, система опрацює слово, якого немає у базі даних, проте це буде звучати нереалістично, саме тому потрібно якомога більше розширювати базу даних фонем, слів та звуків, оскільки саме від цього залежить якість вихідного аудіоматеріалу.

Після фонетичного аналізу [5] кожного слова у фразі ми отримуємо словник у якому кожне слово має відповідний фонетичний розбір та відповідні звуки, а також інформацію про те, чи має бути пауза між словами і наскільки довгою вона має бути. Також у виході фонетичного аналізу має бути присутня інформація про емоційне забарвлення фрази, тобто якщо у кінці фрази стоїть знак питання, то фраза має промовлятись з відповідною інтонацією, те ж саме і знаком оклику.

Дані, які отримуються після виконання фонетичного аналізу, використовуються для створення вихідного аудіофайлу. На даному етапі складаються звуки відповідно до отриманих даних один за одним, між цими звуками вставляється пауза, якщо вона потрібна, а звуки вибираються відповідно до потрібної інтонації у фразі. Після закінчення даного етапу на виході получається вихідний аудіофайл, який в подальшому цілком можна використовувати для закінчення процесу трансформації відеоматеріалу на мову, яка зручна користувачеві.

Загалом, дані три етапи, а саме трансформація аудіоматеріалів тексту, переклад тексту на мову, яка зручна для користувача, та машинне озвучення тексту, являються основою методу трансформації відеоматеріалів на мову, яка зручна користувачеві. Загальну діаграму системи перекладу аудіоматеріалів на мову, яка щручна користувачеві можна побачити на рисунку 1.5.

Рисунок 1.5 – Схема перекладу аудіоматеріалів на мову, яка зручна користувачеві

Як можна побачити на рисунку 1.5 кожен етап надзвичайно важливий та впливає на якість вихідного результату системи трансформації аудіоматеріалів на мову, яка зручна користувачеві. Тобто кожен наступний етап у системі безпосередньо залежить від результату попереднього етапу, що впливає на якість виконання роботи кожного етапу. Саме тому після кожного етапу бажано надавати користувачу можливість змінити результат і коли він буде готовий, то запустити наступний етап у системі трансформації аудіоматеріалів на мову, яка зручні користувачеві. Оскільки користувач матиме змогу перевірити та змінити розпізнаний текст з вхідного аудіофайлу, то це значно збільшить якість розпізнавання тому, що якою б не була база даних для розпізнавання завжди потрібно, щоб була людина, яка перевірить результат і у випадку дефектів поправила їх. Те ж саме і з етапом перекладу тексту, тобто, потрібно, щоб людина перевірила результат і впевнилась, що результат відповідає потрібній якості і може проходити на наступний етап системи. Тільки в такому разі можна досягти максимальної якості у трансформаці аудіоматеріалів на мову користувача.

У висновку можна сказати, що для того, щоб створити систему трансформації відеоматеріалів на мову кристувача потрібно використати системи розпізнавання мовлення, перекладу тексту та його озвучення. А також те, що найкращий підхід

для створення будь-якої системи перекладу – це завжди надавати можливість користувачам доступ до редагування вихідного результату. Тільки так можна покращити якість розпізнавання мовлення, перекладу тексту та машинного озвучення перекладеного тексту. Оскільки користувач завжди зможе переглянути результати, перевірити його на наявність помилок та поправити їх у разі потреби.

1.2 Аналіз літературних джерел з досліджуваної тематики

Трансформація відеоматеріалів на мову користувача проходить кілька етапів, основними є: розпізнавання мовлення, переклад тексту та машинне озвучення тексту.

Оскільки кожен етап залежить від попереднього, то якість вихідних даних кожного етапу повинна бути висока. Тільки у цьому випадку вихідний результат трансформації буде дійсно якісним і задовільнятиме вимогам користувача.

Доцільно провести аналіз відомих рішень, щодо питань розпізнавання мовлення, перекладу тексту та машинного озвучення тексту.

Автори [6] зазначають, що надзвичайно важливо для машинного розпізнавання мовлення також розпізнавати інтонацію, з якою вимовляється певна фраза, адже за допомогою цього буде можливим правильно розставити розділові знаки у вихідному тексті. Ці розділові знаки повинні зберегтись під час перекладу тексту і вони будуть потрібні для подальшого озвучення перекладеної фрази на мову користувача.

Автори [6] також зробили висновок, що для того, щоб правильно визначити емоційне забарвлення фрази та розставити розділові знаки у фразі потрібно використовувати при смисловому аналізі [7] мовних сигналів параметр “основний тон” тому, що це найбільш сильна гармоніка у спектрі мовного сигналу, що визначається частотою коливань голосових зв’язувань людини при вимові голосних і дзвінких приголосних звуків. Виявити чи речення питальне можна перевіривши частоти фрази, наприклад, якщо частота на кінці фрази зростає, то це означає, що дана фраза являється питальною, якщо частота лишається низькою і

стабільною, то це означає, що дана фраза була сказано із стверджувальною інтонацією, а якщо ж часто надто високі, то це означає, що фраза оклична і потрібні вкінці речення поставити знак оклику.

Також за допомогою частоти, з якою була сказана фраза, можна визначити стать промовця, оскільки основний тон людської мови лежить у діапазоні приблизно 60-150 Гц для чоловіків та 150-300 Гц для жінок. Звичайно, цей підхід можна використовувати на практиці і він буде працювати у більшості випадках, проте даний підхід не є до кінця правильним і точним, бо не всі жінки говорять у зазначених вище частотах, як і чоловіки. Саме тому у [8] Корі Беккер провела дослідження, у ході якого отримала модель нейронної мережі для розпізнавання статі. Цей підхід набагато краще спрацює для визначення статі під час розпізнавання мовлення, оскільки він базується на багатьох різних ознаках, що робить реалізацію завдання визначення статі промовця цим методом набагато якіснішим. Цей підхід буде видавати набагато частіше правильні дані, бо базується не тільки на одній частоті, на якій говорить диктор, але й також на багатьох різних ознаках.

Оскільки етап розпізнавання мовлення та трансформація його у текст являється найважливішим етапом, бо це перший етап, що запускається у системі перекладу аудіоматеріалів на мову користувача. Саме цей етап виводить текст, з яким система в подальшому буде працювати, тому саме від даного етапу залежить наскільки якісно спрацюють наступні етапи перекладу аудіоматерілу.

Автор [9] зробив висновок, що для того, щоб зробити розпізнавання мовлення якомога якіснішим потрібно використовувати відповідні натреновані моделі для потрібної мови та предметної області. Безкоштовні акустичні моделі для процесу розпізнавання мови можна знайти у вільному доступі в мережі інтернет, проте ці моделі не є досконалими і їх все одно потрібно тренувати, щоб вони відповідали потребам проекту. Отже, потрібно використовувати найкращі, наскільки це можливо моделі для розпізнавання мовлення, оскільки саме від цього етапу найбільше залежатиме кінцевий результат системи перекладу аудіоматеріалів на мову користувача.

Переклад тексту є також дуже важливим етапом у процесі трансформації аудіоматеріалів на мову користувача, оскільки саме вихідний текст даного етапу буде озвученим за допомогою системи машинного озвучення. Проте, як зазначив [10] на жаль, на даному етапі завжди буде багато проблем і він завжди буде недостатньо якісним, оскільки машинний переклад ще досі не є якісним і дуже часто у вихідному тексті можуть бути присутні граматичні помилки, лінгвістичні помилки, тощо.

Для етапу перекладу тексту, так само, як і для етапу розпізнавання мовлення важлива модель нейронної мережі для системи перекладу. Саме в цій моделі зберігаються словник та граматичні правила цільової мови, за допомогою яких і складається вихідний текст. Модель можна натренувати, розширити та доробити, проте цього все одно буде недостатньо, оскільки для того, щоб перекласти художній текст або ж якісь певні художні прийоми і описи завжди буде потрібна людина, яка зможе розпізнати подібні емоційні забарвлення і подібні речі.

Оскільки найкращою перевіркою якості перекладеного тексту є тільки сама людина, то найкращим рішенням для покращення якості системи перекладу тексту буде не тільки зміна моделі нейронної мережі на кращу, але й надання кінцевому користувачеві доступу для перегляду та зміни перекладеного тексту. У такому випадку кінцевий користувач зможе перевірити перекладений текст після закінчення етапу перекладу тексту на мову зручну користувачеві та виправити помилки, якщо ж вони присутні.

Важливим етапом процесу трансформації відео/аудіо-матеріалів на мову користувача є машинне озвучення перекладеного тексту. Нижче описаний підхід до автоматизованого машинного озвучення тексту, що пропонує автор [11] Спочатку перекладений текст повинен обробитись у модулі лінгвістичного аналізу. Саме у цьому модулі проводиться автоматичне транскрибування, іншими словами, процес перетворення тексту у його фонетичне представлення чи транскрипцію. Для цього етапу потрібно задати модель, у якій міститься інформація про фонетичний розбір та шаблони розстановки наголосів. Дана модель може міститись у базі

даних, до якої система перекладу аудіо матеріалів на мову користувача повинна мати доступ.

Під час лінгвістичного аналізу може частково відбуватись просодичне моделювання. Також просодичне моделювання може відбуватись і при генеруванні мовлення з перекладеного тексту, проте в обох випадках цей етап у системі перекладу аудіо матеріалів на мову, яка зручна користувачеві, може пропускатись, саме тому він може не вноситись як окремий елемент системи.

Просодичне моделювання використовується для надання емоційного забарвлення синтезу тексту, адже саме на цьому етапі аналізуються пунктуаційні знаки і відповідно до них розставляють паузі в синтезі, підвищується або зменшується гучність синтезу мовлення, вибирається тривалість звучання звуку, тощо. Тривалість пауз між словами визначається тільки за допомогою таких розділових знаків, як коми, крапки, знак оклику, тощо. Емоційна забарвленість на даному етапі також може досягатись завдяки додаванням слів-вигуків і їм подібним, що знаходяться у базі даних, це, наприклад, такі слова, як ой-ой, ай-ай, сумно, весело, погано, добре, красиво, жахливо, тощо. Тобто даний метод не є обов’язковим, проте він значно підвищить якість синтезу тексу та зможе наблизити його до людського.

Останнім кроком автоматизованого машинного озвучення перекладеного тексту являється генерація мови. На цьому кроці генерується аудіо-матеріал з готовим озвученням тексту, який був наданий системі для синтезу. Для того, щоб етап генерації мовлення був запущений, потрібно передати йому транскрипцію фрази, що повинна бути озвучена. Транскрипцію фрази повертає попередній крок, а саме – етап лінгвістичного аналізу. Звичайно, для того, щоб згенерувати озвучений аудіо файл тексту, враховуються параметри звучання, тобто, з якою інтонацією повинна звучати озвучена фраза, який це тип речення та коли і наскільки довгими повинні бути паузи між словами. Також на даному етапі для генерації аудіо файлу з тексту використовуються аудіо файли записаних звуків цільової мови, які знаходяться у базі даних, а саме у звуковій базі даних. Дані аудіо

файли конкатенуються у потрібному порядку, щоб у результаті створити вихідний аудіо файл із озвученою фразою в цільовій мові користувача;

Саме такий підхід є найпоширенішим серед систем машинного озвучення тексту. Проте для даного підходу потрібні якісні моделі мов, щоб, відповідно, покращити якість озвучення перекладеного тексту, адже чим кращою буде модель мови, то тим краще система синтезу озвучить текст.

Також перед тим, як передавати перекладений текст, що потребує синтезу, в модуль лінгвістичного аналізу на обробку, потрібно спочатку опрацювати цей текст, перевірити на наявність дефектів у вигляді неправильних символів, що не можуть бути прочитані, видалити зайві пробіли з вхідного тексту, тощо. Тільки в такому разі можна хоча б трішки підвищити якість роботи методу озвучення перекладеного на мову користувача тексту.

У [12] були виділено декілька основних типів систем озвучення тексту, а саме:

– параметричний синтез;
– компіляційний синтез;
– повний синтез мовлення.

Для параметричного синтезу тексту потрібно надати системі готові фрази, які будуть озвучені. Саме тому для цього типу синтезу притаманна дуже хороша якість, проте це також є і величезним мінусом даного методу, оскільки передавши у дану систему фразу, якої немає в підключеній базі даних, то даний метод просто не зможе провести синтез для цієї фрази. Оскільки цей спосіб діє тільки для заздалегідь наданих даних і не зможе опрацювати дані, які немає в базі даних, то він не підходить для методу трансформації аудіо матеріалів на мову, яка зручна користувачеві, бо не зможе успішно адаптуватись до будь-яких даних.

Компіляційний синтез також базується на заздалегідь записаних та збережених даних. Проте його відмінність у тому, що кожне слово записується окремо і під час синтезу кожне слово, що містить передана фраза, перевіряється на наявність у базі даних і якщо воно існує у базі даних, то запис, тобто, озвучення, додається до результату і вкінці отримується аудіо файл синтезу переданої фрази. Цей спосіб, як і попередній, не зможе опрацювати дані, яких немає в базі даних і не

зможе адаптуватись до невідомих даних, тому він також не підходить для методу трансформації аудіо матеріалів на мову, яка зручна користувачеві.

Повний синтез мовлення працює безпосередньо із звуками та надає більше свободи для експериментів. Для цього способу також потрібно мати заздалегідь створену базу даних із моделями, які містять у собі готові записи звуків та голосів, готові словники із словами та їх фонетичними розборами для кожної мови. Оскільки цей спосіб працює із звуками, а не словами, фразами та реченнями, то він зможе прочитати будь-яке передане йому слово чи фразу і навіть повернути хоча б щось, якщо слово було озвучено неправильно під час синтезу. Звичайно, що цей спосіб найкраще підходить для методу трансформації аудіо матеріалів на мову, яка зручна користувачеві, адже цей спосіб гнучкий і зможе адаптуватись до будь-яких даних та озвучити будь-яке слово, якщо всі можливі звуки цільової мови присутні у базі даних.

Таким чином для трансформації відеоматеріалів на мову користувача доцільним є використання способу повного синтезу мовлення, бо результат після розпізнавання мовлення та перекладу мови може бути різним і потрібно адаптуватись під ту фразу, яку передав етап перекладу тексту.

Досліджуючи аналоги методу трансформації відеоматеріалів на мову користувача, було виявлено два сервіси, що дозволяють зробити подібну трансформацію.

Один із цих сервісів був розроблений компанією «Yandex», яка на даний момент заборонена в Україні, через що неможливо якісно дослідити їхні підходи до трансформації відеоматеріалів на зручну мову для користувача. Проте з загальних джерел інформації відомо, що метод «Yandex» [8] не є до кінця реалізованим, оскільки дозволяє перекладати відеоматеріал лише з англійської мови та на російську, що не вирішує повністю всіх проблем. Після розпізнавання тексту не має можливості відредагувати текст, так само як і після перекладу, що робить неможливим використання цього підходу компаніями та простими користувачами, оскільки якість перекладу буде дуже низькою. Також згаданий

сервіс не надає можливості вибрати яким голосом буде озвучене відео та немає можливості редагування кроку озвучення відео.

Інший подібний сервіс – «IconnectFx» [13]. Це гіперконвергентна платформа залучення громад, яка також дає можливість перекладу відео на інші мови. Приклад сторінки перегляду відео з можливістю зміни мови відео можна переглянути на рисунку 1.6.

Рисунок 1.6 – Сторінка перегляду відео з можливістю зміни мови перегляду відео платформи «IconnectFx»

Один із головних плюсів даної платформи – можливість перекладу відео з однієї мови на іншу та можливість редагування тексту на кожному кроці, тобто, редагування тексту після розпізнавання тексту та редагування тексту після його перекладу, що збільшує якість перекладеного тексту. Проте у цієї платформи є декілька мінусів.

Ця платформа не дає можливості поредагувати крок озвучення відео, тобто на цій платформі не має можливості вписати коли має починатись та коли закінчуватись певна фраза, а також не має можливості поміняти для кожної фрази голос, яким вона буде озвучена, лише один голос на все відео. Також немає можливості добавляти паузи у відео, через що після перекладу відео перекладене

аудіо пришвидшується або сповільнюється, щоб відповідати довжині відео, що призводить до непередбачуваних результатів. Зазвичай це просто те, що аудіо не відповідає тому, що відбувається на відео.

Також велика проблема методу трансформації відео на інші мови даної платформи в тому, що після перекладу відео всі фонові звуки зникають і чути лише машинний переклад. Це викликано тим, що дана платформа не розділяє фонові звуки та голос перед тим, як поєднати перекладений аудіофайл з відеофайлом.

Отже, було проведено аналіз існуючих літературних джерел з досліджуваної тематики та проведено дослідження існуючих подібних методів. Було досліджено, що результат кожного з етапів методу повинен бути максимально якісним, оскільки це впливає на на якість роботи методу. Також досліджено, що результати роботи нейронних мереж завжди мають похибку і не будуть якісними, тому завжди потрібно давати користувачу змогу змінити їх результати. Проведено аналіз існуючих аналогів методу трансформації відеоматеріалу на мову користувача, знайдено їй позитивні та негативні сторони.

1.3 Аналіз функціонування гіперконвергентної платформи «IConnect»

Гіперконвергентна платформа залучення громад «IConnect» складається з багатьох модулів. Для того, щоб повністю представити функціонал даної платформи потрібно описати хоча б основні його бізнес-процеси. Серед них можна виділити такі:

- авторизація та реєстрація користувачів;
- редагування відеоматеріалів;
- управління відеоматеріалами;
- проведення онлайн-конференцій;
- проведення прямих трансляцій;
- можливість переглядати відеоматеріали інших користувачів;
- система комунікації користувачів;
- система сповіщень користувачів про зміни в системі;

– управління подіями.

Платформа «IConnect» - це веб-додаток розроблений на технології .Net Web Forms [14], який призначений для проведення різного типу подій, онлайн-конференцій, прямих трансляцій, онлайн уроків, лекцій тощо. Оскільки ця система була розроблена з метою залучення громад і позволяє не тільки проводити онлайн трансляцій, онлайн-конференцій, але й роботу з відеоматеріалами, то для цієї системи також потрібно додати модуль перекладу з однієї мову на іншу. Разом з цим модулем система позволить користувачам не тільки переглядати відеоматеріали на мові оригіналу, але й перекладати їх на зрозумілу для себе мову, редагувати ці відео, щоб зробити озвучку відео набагато якіснішою та кращою. Також модуль перекладу відео можна частково використати для онлайн-трансляцій, які присутні у даній системі тому, що у методі трансформації відео на мову користувача також присутні функції розпізнавання мовлення та перекладу тексту. Відповідно, ці функції можна використати для перекладу в реально часі під час онлайн-трансляцій. Також можливо зробити повний переклад відео під час трансляцій або онлайн-конференцій, проте системі потрібен буде певний час на розпізнавання фрази диктора, перекладу цієї фрази та її озвучення. Тому при реалізації повного перекладу під час онлайн-трансляцій або онлайн-конференцій буде певна затримка озвучки після кожної фрази, що може зробити спілкування людей некомфортним, бо користувачам потрібно буде робити короткі паузи після кожного озвученого речення, щоб система мала змогу обробити це речення і програти озвучений текст користувачу, який в даний момент слухає.

Модуль авторизації та реєстрації користувачів позволяє користувачам входити у систему, а системі цей модуль вказує, що цей користувач присутній у базі даних, відповідно система може надавати або не надавати певних доступів або привілегій користувачеві відповідно до даних в базі даних. Оскільки система складається з багатьох менших систем, то цей модуль позволяє користувачам реєструватись та авторизовуватись в одній системі і переносити всі їх дані з однієї системи в іншу. Тобто якщо зареєструватись у системі, що позволяє працювати з відеоматеріалами, то, відповідно, всі дані користувача буде перенесено на всі інші

системи, що дасть можливість не реєструватись в подальшому для них або ж авторизовувати користувача ще один раз. Тож в подальшому потрібно продумувати реалізацію метода трансформації відеоматреалів на мову користувача так, щоб використовувати цей модуль в реалізації методу.

Редагування відеоматеріалів надає користувачам можливість редагувати відео, що вони загрузили, а саме змінювати довжину відео, обрізати його, додавати надписи на відео, поєднувати декілька відео в одне, додавати субтитри до нього, тощо. Тож тут також потрібно реалізовувати метод трансформації відеоматеріалів так, щоб він використовував дану функцію гіперконвергентної платформи. А саме, щоб була змога в користувача в редакторі відео розпізнати текст з відео, яке редагується користувачем на даний момент, в тій же системі проредагувати розпізнаний текст, перекласти розпізнаний та відредагований текст, також дати змогу відредагувати перекладений текст та озвучити його. В результаті до редактора відео у гіперконвергентній платформі повинне добавитись аудіо з озвученим перекладеним текстом і користувач повинен мати змогу редагувати цей аудіо-матеріал.

Модуль для управління відеоматеріалами в гіперконвергентній платформі залучення громад «Iconnect» передбачає, що користувачі матимуть змогу загружати відеоматеріали, описувати їх, вводити інформацію про дані відео, вказувати кому дані відео будуть показані в системі, переглядати відео інших користувачів, видаляти відео, тощо. Під час реалізації методу трансформації відеоматеріалів на мову користувача потрібно також взяти до уваги, що перекладений відеоматеріал на іншу мову також повинен добавитись в базу даних та відображатись в даному модулі.

Модулі для проведення онлайн-трансляцій [15] та онлайн-конференцій [16] позволяють користувачам комунікувати між собою за допомогою аудіо та відео зв’язку. Онлайн-трансляція надає користувача в режимі реального часу записувати відео і в той же час ділитись ним з іншими користувачами, а користувачі мають змогу спілкуватись з диктором за допомогою модуля комунікації між користувачами, який надає користувачам обмінюватись текстовими

повідомленнями. Онлайн-конференція в той час дає можливість користувачам спілкуватись між собою в режимі реального часу за допомогою передачі відео та аудіо даних між ними. Ці модулі також важливі для реалізації методу трансформації відеоматеріалів на мову користувача, адже потрібно розробити систему так, щоб кожен етап можна було запускати окремо, тоді дану реалізацію можна використовувати для цього модуля, а саме етапи розпізнавання мовлення та перекладу тексту. В такому разі буде можливість добавити змогу користувачам переводити їх фрази за допомогою субтитрів.

На рисунку 1.7 представлена схема бізнес-процесів гіперконвергентної платформи «Iconnect».

Рисунок 1.7 – Схема бізнес-процесів гіперконвергентної платформи «Iconnect»

На рисунку 1.7 представлені всі бізнес-процеси гіперконвергентної платформи, які важливі під час реалізації трансформації відеоматеріалів на мову користувача для даної платформи.

Було проведено аналіз гіперконвергентної платформи, виявлено та представлено всі важливі для методу трансформації відеоматеріалів на мову користувача бізнес-процеси, досліджено технології та вихідний код гіперконвергентної платформи.

1.4 Постановка задачі дослідження

Після проведеного аналізу в попередніх підрозділах було визначено, що трансформація відеоматеріалів на мову користувача є актуальною задачою.

Метою кваліфікаційної роботи є розробка методу трансформації відеоматеріалів на мову користувача.

Для реалізації мети необхідно вирішити наступні завдання:

- дослідити підходи до перетворення відеоматеріалів на мову користувача;
- проаналізувати існуючі літературні джерела з досліджуваної тематики;
- проаналізувати функціонування гіперконвергентної платформи;
- розробити метод трансформації відеоматеріалів на мову користувача;
- реалізувати метод трансформації та дослідити функціонування системи.

Розробка методу повинна забезпечувати підтримку максимально можливу кількість мов для кожного етапу трансформації згідно вимог кінцевого користувача, давати можливість користувачам редагувати текст на кожному етапі перекладу відео. При цьому слід врахувати проблему з редагуванням тексту користувачем перед початком кроку озвучення відео, щоб забезпечити на цьому етапі дружній користувацький інтерфейс. Для забезпечення передачі мовних ознак у системі необхідно використовувати розмітку тексту у форматі SSML. Виходячи з цього інтерфейс користувач, що буде розроблено, надаватиме можливість позначати потрібну йому опцію в тексті та виставляти потрібні налаштування для цієї опції у вигляді рисунків, а клієнтське забезпечення відповідно повинне збирати інформацію про дані рисунки, їх позицію в тексті та замінювати їх на відповідний

тег або атрибут в мові розмітки синтезу мовлення. За допомогою пропонованої розмітки будуть передаватись дані про те, коли фраза має починатись, коли вона повинна закінчуватись, яким голосом дана фраза має бути озвучена, а також гендерні ознаки голосового повідомлення, та коли і наскільки довго повинні тривати паузи, щоб зробити переклад відеоматеріалу якісним та комфортним для сприйняття користувача. Забезпечення управління тривалістю паузи у системі, що буде розроблена, та тривалістю фраз уможливить змогу користувачеві поставити фрази у шкалі часу відеоматеріалу так, щоб вони співпадали із рухами губ диктора або ж просто перекривали оригінальну озвучку відео. Також потрібно добавити крок розділення голосів з фоновими звуками, що дозволить після перекладу відео залишити фонові звуки з оригінального відео у перекладеному та залишити опцію «Приглушити оригінальне озвучення», щоб користувач мав можливість залишити оригінальне озвучення у перекладеному відео.

Таким чином для системи, що реалізовуватиме трансформації відеоматеріалів на мову користувача, необхідно застосувати повний синтез мовлення, щоб результати після розпізнавання мовлення та перекладу мови були якісними для відповідності вище вказаним вимогам та враховували можливість адаптуватись під ту фразу, яку передав етап перекладу тексту.

Висновки до розділу 1

1. Проаналізовано підходи до перетворення відеоматеріалів на мову користувача, основними етапами перетворення є: розпізнавання мовлення, переклад тексту та машинне озвучення тексту.
2. Проведено аналіз існуючих літературних джерел з досліджуваної тематики.
3. Проаналізовано функціонування гіперконвергентної платформи, розглянуто можливість її покращення.
4. Сформовано мету та завдання дослідження.

2 МЕТОД ТРАНСФОРМАЦІЇ ВІДЕОМАТЕРІАЛІВ НА МОВУ КОРИСТУВАЧА

2.1 Суть методу трансформації відеоматеріалів на мову користувача

Метод трансформації відеоматеріалів на мову користувача складається з кількох кроків:

– видобути аудіо файл із завантаженого відеоматеріалу;

– розпізнати мовлення за допомогою нейронних мереж з аудіофайлу, тобто провести трансформацію видобутого аудіофайлу в текст;

– перекласти текст на іншу мову, яку вибрав користувач;

– провести трансформацію перекладеного тексту в аудіофайл, тобто озвучити цей текст;

– відділити голоси людей від інших фонових звуків за допомогою штучного інтелекту в оригінальному аудіофайлі;

– за допомогою програмних інструментів, що дозволяють працювати з відео та аудіофайлами, зменшити гучність голосів людей в оригінальному аудіофайлі;

– поєднати аудіофайл озвученого перекладеного тексту, а саме: аудіофайл із зменшеною гучністю голосів оригіналу, оригінальний аудіофайл із фоновими звуками та оригінальний відеофайл без звуків.

У результаті трансформації буде отримано відеофайл перекладений на іншу мову.

Оскільки на останньому етапі, перед злиттям файлів, голоси людей від фонових звуків за допомогою штучного інтелекту будуть відділені, то всі фонові звуки, наприклад, стукіт, музика, клацання ручкою і тому подібні звуки, не будуть втрачені; файл не буде позбавлений емоційного забарвлення.

Також є можливість залишити оригінальні голоси із зменшеною гучністю, якщо цього, звичайно, потребує користувач.

Розпізнавання мовлення – це процес, що потребує дуже багато потужності сервера, на якому він буде відбуватись, а саме він використовує дуже багато

потужності процесора та оперативної пам’яті сервера. Оскільки цей процес являється надто ресурсозатратним, то найкраще запускати цей процес на окремому сервері, щоб він не перешкоджав роботі іншим потрібним програмним забезпеченням, що потрібні для реалізації метода трансформації відеоматеріалів на мову користувача.

Автоматизоване розпізнавання мови – це вирішення задачі розпізнавання людського мовлення та перетворення його в текст. Ця галузь привернула до себе велику увагу за останні декілька десятиліть. Ранні методи трансформації людського мовлення у текст були зосереджені на ручному виділенні ознак і звичайних методах, таких як змішані моделі Гауса, також відомі, як GMM [17], алгоритм динамічного викривлення часу, для якого існує також відоме скорочення DTW [18], а також приховані моделі Маркова, якого зазвичай називають скорочено HMM [19]. А зовсім недавно популярності набрали нейронні мережі, такі як рекурентні нейронні мережі, скорочено RNN [20], згорткові нейронні мережі, відомі також як CNN [21], і в останні роки нейронні мережі типу трансформери [22], що були застосовані в методах автоматизованого розпізнавання мовлення, досягли великого успіху у них та показали хорошу якість в роботі.

Отож, основна ціль систем автоматизованого розпізнавання мовлення - це трансформація вхідного аудіо сигналу x = (x1, x2, ..., xt), із зазначеною довжиною T, в послідовність слів або символів y = (y1, y2, ..., yn), де n – це кількість слів, що система розпізнала з вхідного аудіо сигналу. Всі символи та слова, що повертає система розпізнавання мови, являються підмножиною множини символів та слів, що знаходяться у словнику.

Всі методі розпізнавання мовлення мають один і той же принцип роботи, який складається з наступних кроків:

– опрацювання та нармалізація вхідного сигналу, цей крок також називають попередньою обробкою;

– виділення ознак;

– класифікація;

– моделювання мовлення.

Крок попередньої обробки вхідного сигналу націлений на те, щоб покращити вхідний аудіо сигнал за допомогою зниження шумів в аудіо сигналі та фільтрування сигналу.

Взагальному, ознаки, які використовуються для автоматизованого розпізнавання мовлення, визначаються за допомогою певної кількості значень або коефіцієнтів, що генеруються шляхом застосування різних методів на вхідних даних. Цей крок має бути бути дуже надійним, оскільки це стосується різних факторів якості, таких як шум або ефект відлуння.

Велика кількість методів атоматизованого розпізнавання мовлення використовують наступні методи вилучення ознак:

– мел-частотні кепстральні коефіцієнти або скорочено MFCC [23];
– дискретне вейвлет перетворення, скорочено DWT[24].

Мел-шкала [25] є емпіричною шкалою, яка грунтується на людському відчутті частоти звуку. На основі мелчастотних кепстральних коефіцієнтів розраховуються ознаки для нейронних мереж при розпізнавання певних голосових команд.

Дискретне вейвлет перетворення – це будь-яке вейвлет перетворення, що розкладає даний сигнал на кілька наборів, де кожен набір є часовим рядом коефіцієнтів, що описують еволюцію сигналу в часі у відповідному діапазоні частот.

Вейвлет-ряд [26] – це математичний вираз, що описує математичну функцію, яка позволяє аналізувати різні частотні компоненти даних.

Модель класифікації спрямована на пошук усного тексту, який міститься у вхідному сигналі. Він бере виділені ознаки, що були результатом роботи кроку попередньої обробки вхідного сигналу, та генерує вихідний текст.

Мовна модель є важливим модулем, оскільки вона фіксує граматичні правила або семантичну інформацію мови. Мовні моделі важливі для розпізнавання вихідного токена з моделі класифікації, а також для внесення виправлень у вихідний текст. Тобто саме даний крок в автоматизованому розпізнаванні мовлення відповідає за якість тексту, який розпізнається тому, що саме цей крок відповідає

за збереження змісту, граматичні виправлення тексту, що розпізнався на попередньому кроці, тощо.

Для того, щоб автоматизоване розпізнавання мовлення працювало потрібно, щоб у наявності були готові натреновані мовні моделі. Набір даних для тренування мовних модель може записуватись у студії або ж можна взяти якісь готові записи з відкритих джерел, наприклад, аудіокниги, різні діалоги, записи розмови людей, тощо.

Мел-частотні коефіцієнти Кепстрала [27] являються найпоширенішим методом виділення ознак мовлення. Людське вухо є нелінійною системою, якщо говорити про те, як воно сприймає звуковий сигнал. Для того, щоб впоратись зі зміною частоти, була розроблена Мел-шкала для створення лінійної моделі слухової системи людини. Лише частоти в діапазоні від 0 до 1 кГц включно можуть бути перетворені в Мел-шкалу, а решта частот вважаються лошарифмічними. Частота Мел-шкали обчислюється за формулою 2.1.

$$fmel = \frac{1000}{\log(2)}(1 + \frac{fHz}{1000}) \quad (2.1)$$

Для того, щоб обчислити Мел-шкалу, потрібно передати у формулу параметр fHz, що представляє частоту оригінального сигналу.

Метод мел-частотних коєфіцієнтів Кепстарала для виділення ознак з сигналу, що повернув крок попередньої обробки вхідного сигналу, зазвичай включає в собі такі кроки:

– процес для посилення гармонік, згладжування крайніх частот вхідного сигналу. Зазвичай цей процес також називають Windowing [28];

– застосування дискретного перетфорення Фур’є;

– логарифмування величини;

– переведення результатів попереднього кроку в Мел-шкалу;

– застосування зворотного дискретно-косинусного перетворення, скорочено DCT [29].

В епоху глибокого навчання нейронні мережі продемонстрували значне вдосконалення завдання розпізнавання мовлення. Були також застосовані різні методи для автоматизованого розпізнавання мовлення, такі як згорткові нейронні мережі, скорочено CNN, рекурентні нейронні мережі, які також скорочено називають RNN, а нещодавно нейронні мережі типу трансформери досягли великої продуктивності та якості.

Оскільки на даний момент нейронні мережі типу трансформери досягли значних покращень, хорошої якості у результаті використання їх в різних методах розпізнавання мовлення, перекладу фраз на інші мови. Моделі нейронних мереж типу трансформери, що призначені для розпізнавання мовлення, трасформери зазвичай базуються на архітектурі кодера-декодера, подібній до моделей seq2seq [30]. Якщо розглянути їх більш детальніше, то вони базуються на механізмі самоуважності замість рецидивів, що був застосований у рекурентних нейронних мереж. Механізм самоуважності – це одна із багатьох технік навчання нейронних мереж. Механізм самоуваги може звертати увагу на різні позиції послідовності та витягувати значущі представлення. Механізм самоуваги використовує три параметра: запити, значення та ключі. Позначимо запити, як Q, значення, як V, ключі, як K, а коефіцієнт масштабування, як dk. Тож результати механізму самоуваги розраховуються за формулою 2.2.

$$Attention(Q, K, V) = softmax\left(QK^T/\sqrt{(dk)}\right)V \quad (2.2)$$

Для розпізнавання мовлення потрібно використувати нейронні мережі типу трансформери, які призначені для розпізнавання мовлення, зазвичай їх також називають Speech-Transformer. Приклад такої нейронної мережі та її вихідний код можна знайти на сайті Github [31], цей проєкт має назву Speech-Transformer, він розроблений на мові Python, як бібліотека, що легко підключається в інші проєкти, тому його можна просто використовувати у власних проєктах.

Нейронні мережі типу трансформери для розпізнавання мовлення проводять трансформацію послідовності голосових ознак на відповідну послідовність

символів, що разом становлять слова та фрази. Послідовність ознак, яка є довшою за вихідну послідовність символів, будується з двовимірних спектограм із розмірами часу та частоти. Якщо бути більш конкретним, то згорткові нейронні мережі використовуються для використання локальності спектограм і пом'яшення невідповідності довжини за допомогою кроку в часі. Приклад роботи такого трансформера показано на рисунку 2.1

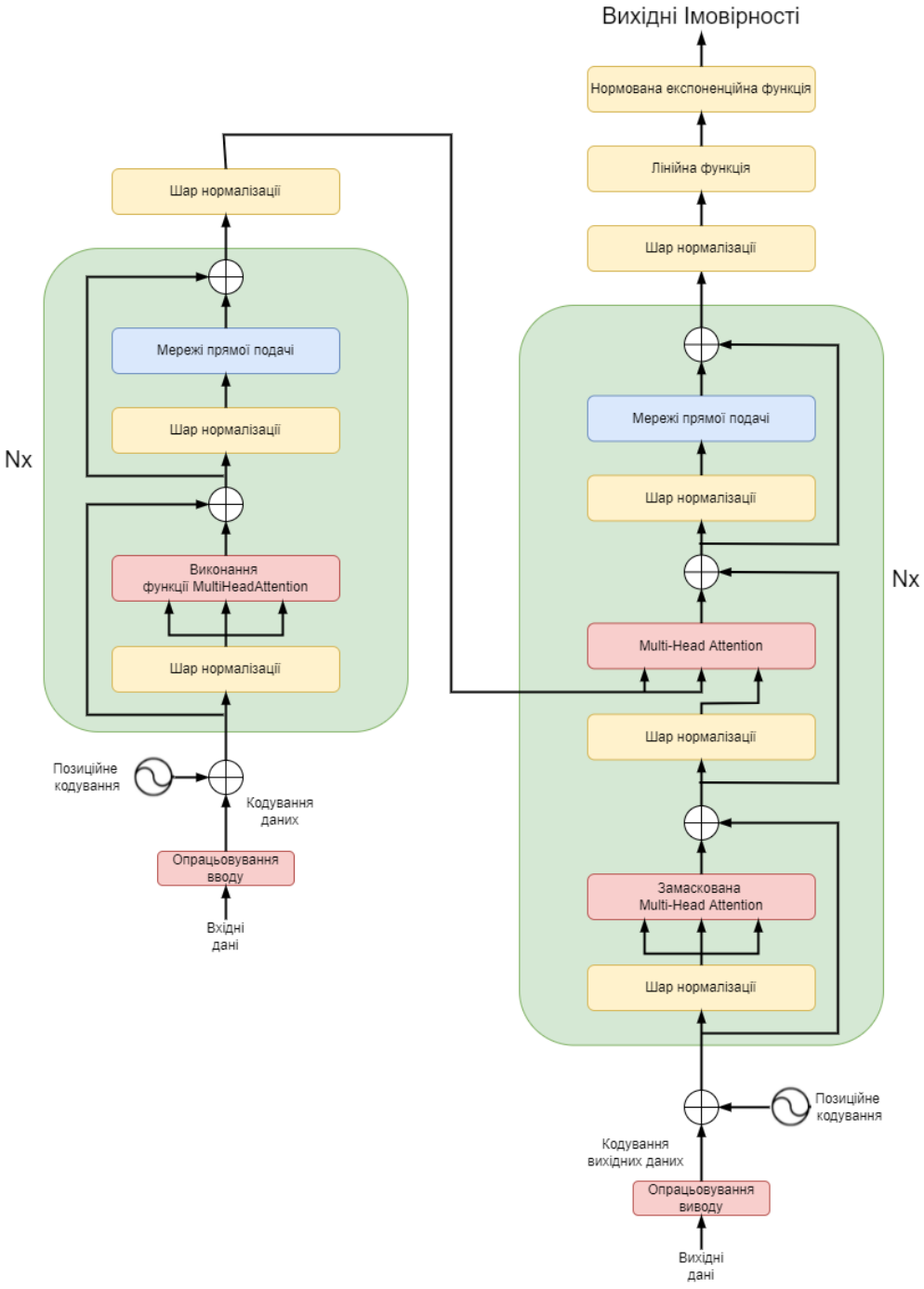

Рисунок 2.1 – Схема роботи нейронної мережі типу трансформера.

Запити, ключі та значення витягуються із згорткових нейронних мереж і передаються до двох модулів самоуважності. Трансформе розпізнавання голосових команд оцінюється за наборами даних WSJ [32] і досягає конкурентоспроможних результатів розпізнавання з частотою помилок у словах 10.9% [33], у той час як йому потрібно приблизно на 80% менше часу на навчання, ніж звичайним рекурентним нейронним мережам, або скорочено RNN, та згортковим нейронним мережам, які також називають CNN. Саме тому було обрано використовувати саме цей тип нейронних мереж для розпізнавання мовлення з відеоматеріалів, оскільки цей тип нейронних мереж дуже швидко навчається та видає набагато кращі та якісні результати.

Отже, Було досліджено суть методу траснформації відеоматеріалів на мову користувача, розроблено етапи для цього методу, досліджено методи трансформації аудіоматеріалів у текст, типи нейронних мереж.

2.2 Алгоритм трансформації відеоматеріалів на мову користувача

Алгоритмічне забезпечення – сукупність алгоритмів рішення задач функціонального наповнення системи. Алгоритмічне забезпечення включає засоби опису узагальнених процедур функціонування і моделювання об’єктів системи, а також засоби детального опису рішення конкретних задач.

Схема алгоритму трансформації відеоматеріалів на мову користувача [55] зображена на рисунку 2.2.

На рисунку 2.2 поетапно показано як відбувається переклад відеоматеріалів на мову користувача. Спочатку зчитується інформація з бази даних для потрібного відео, а саме: інформація про сервер, на якому розташоване відео, інформація про те, як можна дістатись до цього відео, яка мова оригіналу даного відео, у яку мову потрібно перекласти це відео та різна інформація, яка потрібна для роботи системи, наприклад, ідентифікатор мережевого рівня потрібних серверів, на яких розгорнуті потрібні для системи сервіси перекладу тексту, розпізнавання мовлення, трансформації перекладеного тексту в аудіо файл, роботи з відео.

Рисунок 2.2 – Схема алгоритму трансформації відеоматеріалів на мову користувача

Тоді розпочинається процес розділення аудіо та відео за допомогою сервісу, який являється консольним програмним забезпеченням для роботи з відеоматеріалами та розташований на окремому сервері задля покращення якості та швидкості обробки відео. Результат даного процесу записується на файловий сервер, а інформація про файл та його відношення до відео записується у базу даних. Після цього запускається етап розпізнавання тексту в результаті якого на файловий сервер записується текстовий файл розпізнаного тексту з аудіо.

Як тільки у наявності є розпізнаний текст, то користувач може приступити до перекладу розпізнаного тексту. Результат даного етапу також записується на файловий сервер і в базу даних, щоб зберігати історію всіх змін користувача, надавати користувачеві постійний доступ до всіх його файлів та для того, щоб система могла автоматично витягнути останні дані для деяких етапів. Всі попередні етапи, а саме етапи розділення відео та аудіо файлів з оригінального відео, яке загрузив користувач на файловий сервер, розпізнавання мовлення, перекладу тексту та машинне озвучення перекладеного тексту, запускається на окремому сервері, оскільки вони потребують багато потужності процесора сервера та використовують багато оперативної пам’яті, тому бажано запускати їх на окремій машині, щоб інше програмне забезпечення не перешкоджало даним процесам.

Машинне озвучення тексту запускається одразу після того, як був отриманий перекладений текст розпізнаного тексту з відео. Після закінчення даного етапу результат у вигляді аудіоматеріалу також зберігається на файловому сервері, а всі відповідні дані про файл, файловий сервер на якому він розміщений та всі потрібні зв’язки з цим файлом та відео поміщаються в базу даних.

Як можна побачити на рисунку після озвучення перекладеного тексту фонові звуки та оригінальні голоси добавляються тільки тоді, коли користувач вибрав ці опції.

Після виконання вище описаних етапів відбувається поєднання всіх отриманих аудіоматеріалів в один відповідно до того, як це налаштував користувач. Тобто якщо користувач хоче відділити фонові звуки з відео, то всі отримані

аудіоматеріали поєднуються в один, проте без фонових звуків, якщо ж користувач бажає, щоб фонові звуки були присутні у результаті, то вкінці всі аудіоматеріали поєднуються в один разом із фоновими звуками. Також якщо користувач бажає, щоб оригінальні голоси людей були присутні в результаті роботи реалізованого методу трансформації відео на мову користувача, то до аудіоматеріалів, що будуть об'єднуватись в один, також додається аудіоматеріал із оригінальними голосами людей, проте перед цим його гучність знижується трішки, щоб це аудіо не мішало користувачам при перегляді перекладеного відео, якщо ж користувач не бажає голосів людей з оригінального відео, то об'єднуються всі аудіоматеріали, що були отримані в результаті роботи методу, окрім аудіоматеріалу з голосами людей з оригінального відео.

Озвучення тексту працює за алгоритмом, представленим на рисунку 2.3.

Спочатку зчитується файл із текстом, тоді запускається попередня обробка та нормалізація тексту, що у відповідь повертає фонеми. Ця функція перетворює текст у лінгвістичні особливості цільової мови у формі вектора, введеного у акустичну модель. Після цього запускаються акустичні моделі – алгоритми оптимізовані для перетворення попередньо обробленого та нормалізованого тексту в Mel-спектограми [34], як вихідні дані. Спектограма гарантує, що врахували всі відповідні аудіофункції. Після акустичної моделі запускається нейронний вокодер, який приймає Mel-спектограми, що перетворюються на хвилю за допомогою нейронного вокодера. У результаті отримується аудіоматеріал із озвученим перекладеним текстом, який зберігається на файловому сервері. Файловий сервер вибирається за допомогою інформації про всі доступні сервери, на яких зберігаються дані. Ця інформація зчитується з бази даних, яка постійно оновлюється після кожного нового завантаженого файлу. Таким чином система постійна має актуальну інформацію про те наскільки завантажений певний файловий сервер і скільки місця на ньому залишилось. За допомогою цієї інформації і вибирається сервер, на який можна загрузити файл. Також бажано загружати файл саме на той сервер, який розташований найближче до користувача, інформація про розташування сервера також присутня у базі даних. Після того, як

файл був збережений на сервер, то потрібно записати всю інформацію про файл у базу даних, а саме його розмір, де він розташований на сервері та посилання, по якому можна завантажити даний файл. Також потрібно оновити інформацію про сервер на який був записаний даний файл та додати нові записи, що показують до якого саме відео цей файл відноситься.

Рисунок 2.3 – Схема алгоритму озвучення тексту

Отже, було розроблено алгоритм трансформації відеоматеріалів на мову користувача, алгоритм озвучення тексту та детально описано їх кроки.

2.3 Загальна структура системи трансформації відеоматеріалів на мову користувача

Метод трансформації відеоматеріалів на мову користувача буде реалізовано у вигляді системи. Дана система розробляється для гіперконвергентної платформи, а це означає, що потрібно дотримуватись кросплатформеності, гнучкості та максимальної підтримки системи на різних браузерах, оскільки це полегшить розробку та введення система в платформу.

Гіперконвергентна платформа – це велика система, яка складається з багатьох інших підсистем таких, як система роботи з відеоматеріалами, система роботи з прямими трансляціями, система роботи з подіями, система управління контрактами, система управління працівниками, тощо. Оскільки дана гіперконвергентна платформа та всі її модулі повинні запускатись на операційні системі Windows на даний момент, а в майбутньому повинна бути перенесена на операційну систему Linux, то потрібно використовувати технології, бібліотеки та мови програмування, що підтримують ці обидві операційні системи. Саме тому для реалізації методу трансформації відеоматеріалів на мову користувача, який являється модулем гіперконвергентної платформи, потрібно в основному використовувати технології Dotnet, бо саме ці технології позволять підтримувати кросплатформеність.

Серверна частина проекту буде реалізована на технології «DotNet» [35]. Ця технологія було вибрано тому, що вона забезпечує стабільність і розроблена якраз таки для подальшої розробки великих проектів з різноманітною бізнес логікою. Ця технологія кросплатформена, тобто вона може працювати на багато ОС, а саме на «Windows» [36], «Linux» [37] та «Mac» [38], що значно полегшує вибір сервера для неї. Ця технологія позволяє розробляти програмне забезпечення різного виду, а

саме програми для персональних комп’ютерів, додатки для мобільних пристроїв, веб сайти, API [39], ігри, тощо. Для даного проекту буде розроблене програмне забезпечення типу API для того, щоб було зручно передавати дані клієнтові. Також оскільки інші модулі гіперконвергентної платформи було розроблено саме на цій технології, то її вибір позволить легко підключити даний модуль у платформу.

Загалом, розробка програмного забезпечення із методом комунікації API позволяє підключити розроблену систему будь-куди, бо цей метод комунікації базується на веб запитах. Таким чином метод трансформації відеоматеріалів на мову користувача реалізований на методі комунікацій API буде працювати для персональних комп’ютерів, ігрових консолей, телефонів, для будь-якої операційної системи, а також для будь-якого пристрою, що має доступ до інтернету та підтримує веб запити. Також реалізація за допомогою API дозволить підключати дану систему не тільки до гіперконвергентної платформи, але й до багатьох інших програмних забезпечень, що можуть виконувати веб-запити, тож цю систему можна буде використовувати і поза межами платформи, наприклад, продаючи до неї іншим розробникам програмного забезпечення або компаніям.

Загальну архітектуру системи трансформації відеоматеріалів на мову користувача можна побачити на рисунку 2.4.

На рисунку 2.4 представлено сервер, на якому розміщені серверна та клієнтська частини проекту. Це основа всього рішення, оскільки саме через них відбувається взаємодія користувача з проектом. Саме цей проект виконує веб-запити до серверної частини, що обробляє вхідні запити, зчитує та записує дані до бази даних, обробляє дані, працює з відеоматеріалами, проводить розпізнавання мовлення, переклад тексту, машинне озвучення тексту та видає результат у вигляді JSON об’єкту [40].

Клієнтська частина реалізації методу трансформації відеоматеріалів була розроблена на технології «React» [41] оскільки вона дуже швидка, а також дуже зручна під час використання. Також ця технологія дає нам можливість посилати користувача на інші сторінки без загрузки сторінки, а разом з цим ми можем показувати процес трансформації відеоматеріалів у текст в реальному часі. Тобто,

коли користувач загрузив власне відео на сайт та запустив процес розпізнавання мови з відео, то він зможе спостерігати за цим процесом завдяки тому, що текст, який розпізнається буде по реченню добавлятись на сторінку в режимі реального часу. Теж саме можна застосувати і до інших етапів, наприклад, для перекладу тексту можна в режимі реального часу показувати як поступово змінюється текст і перекладається на цільову мову.

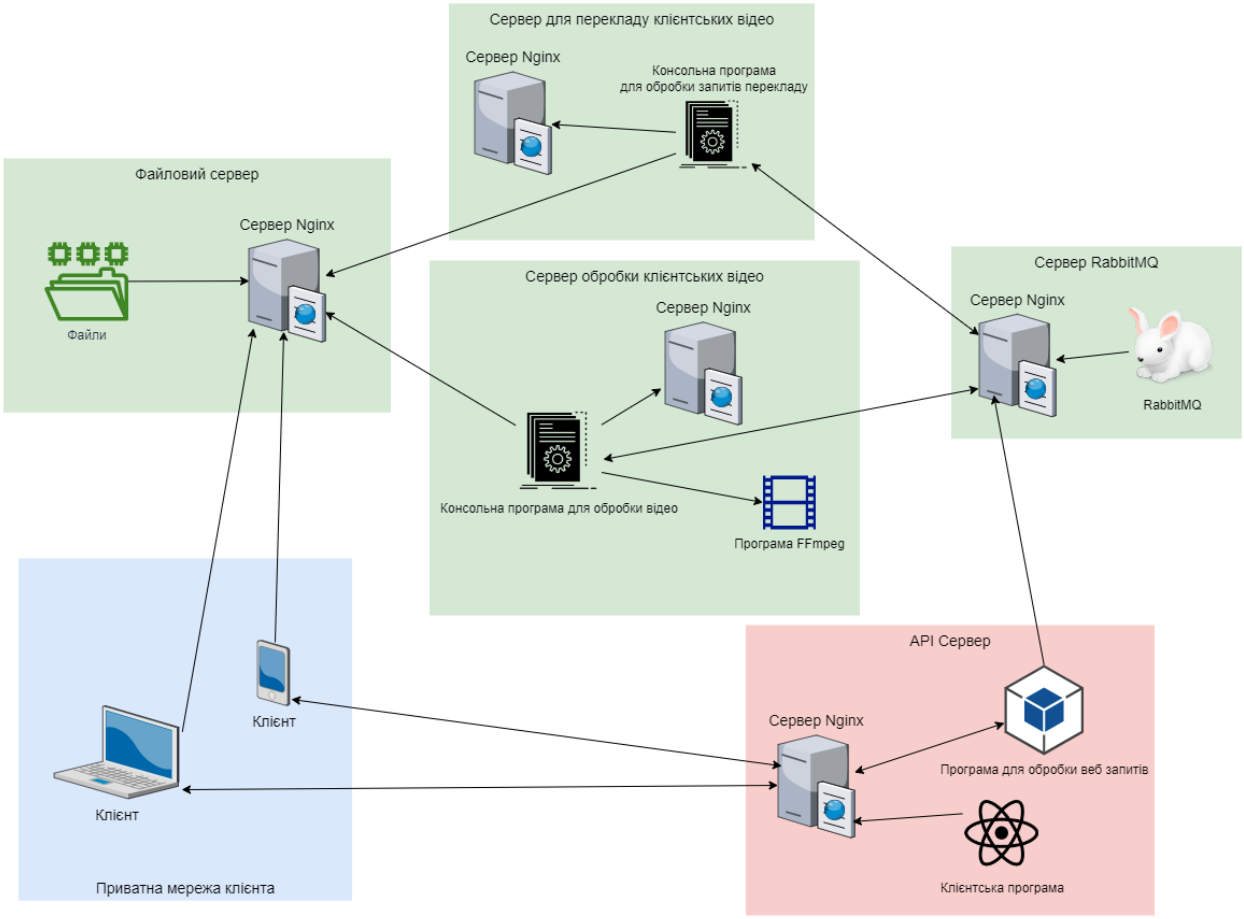

Рисунок 2.4 – Архітектура системи трансформації відеоматеріалів на мову користувача

Оскільки перший крок у нашому методі – це процес відділення звуку від відеоматеріала, то нам потрібен окремий сервер, на якому ми будем це проводити, оскільки цей процес займає дуже багато потужності комп’ютера, а саме використання процесора та оперативки. Також наявність хорошої відеокарти на сервері може значно пришвидшити цей процес, проте такі сервери на даний момент

дуже дорогі, тому для розробки та тестування реалізації методу трансформації відеоматеріалів на мову користувача було обрано виділені сервера, які не мають в наявності відеокарти, проте мають хороший та швидкий процесор. Сервера також розташовані в Україні у місті Харків, оскільки це найближча можлива точка і в такому разі буде кращий зв'язок з ними, а це потрібно для постійного зв'язку, який встановлюється для того, щоб показувати процес трансформації відео в аудіо, процес трансформації аудіо в текст та переклад тексту. Для того, щоб легко працювати з відео та не вигадувати існуючих рішень було обрано програмне забезпечення «ffmpeg» [42]. Це програмне забезпечення кросплатформене, не займає багато місця на диску та легке у використанні, тому для роботи з відео було вибране саме воно. Проте для роботи з цим програмним забезпеченням немає офіційних збірок для технології «dotnet» [43], тому був розроблене окреме програмне забезпечення на мові Python [44], яке підключається в інші проекти, і використовується для того, щоб з правильним синтаксисом та у правильній послідовності запускати команди «ffmpeg».

На жаль, через те, що робота з відео завжди забирає багато потужності процесора або ж відеокарти необхідно виділити окремий сервер для програмного забезпечення «ffmpeg» і поставити разом з ним консольну програму, яка його запускає і вказує яке відео відкрити та що саме з ним потрібно зробити. В іншому випадку, якби ми залишили все на одному сервері, то, скоріше за все, сервер не витримував би нагрузок і працював з перебоями.

Для роботи з програмним забезпеченням «ffmpeg» було розроблено програмне забезпечення на мові Python тому, що на технології «Dotnet» важко працювати з цим програмним забезпеченням і не завжди можливо витягнути текст помилки, якщо щось пішло не так. Оскільки інформація про помилки дуже важлива, коли системою користуються справжні користувачі, то було вирішено все ж таки використовувати Python разом із готовим рішенням, бібліотекою, для даної мови програмування – «ffmpeg-python». Дана бібліотека позволяє легко працювати з командами програмного забезпечення «ffmpeg», правильно зчитувати помилки, які повертає дане програмне забезпечення, та складати запити до даного

програмного забезпечення. Дане програмне забезпечення на мові python являється консольним, тому для того, щоб запускати його та передавати йому інформацію про те, яке відео потрібно обробити та як саме його потрібно опрацювати, потрібно використати додаткове програмне забезпечення, яке буде реалізовувати ідею черги повідомлень. Тобто дана програма для роботи з програмним забезпеченням «ffmpeg» повинна бути постійно запущеною на сервері та постійно слухати деякий порт веб-застосунку, черги повідомлень, щоб миттєво відреагувати на запит користувача про необхідність початку опрацювання відео. Дана програма повинна постійно очікувати повідомлення від користувача і як тільки це повідомлення прийшло, то вона повинна негайно на нього зреагувати та запустити відповідний процес у програмному забезпеченні «ffmpeg», щоб обробити відео, і видіти відповідний результат клієнту.

Для того щоб реалізувати просту та легку систему для роботи з перекладом відео було розроблено декілька консольних програм. Для систем розпізнавання тексту з аудіо матеріалу, перекладу тексту та синтезу мовлення для перекладеного тексту було розроблено три програмних забезпечення консольного типу, які очікують запиту від користувача і при його надходженні негайно реагують на нього, запускають відповідні процеси та видають результат користувачеві у вигляді JSON об'єкту. Кожна з цих програм використовує сервіси Azure[45] для того, щоб швидко і якісно розпізнати текст, перекласти його на цільову мову та озвучити перекладений текст. Щоб спростити використання даних програм та зробити архітектуру гнучкішою, тобто зробити її такою, щоб будь-які зміни функціоналу або ж додавання нового функціоналу було набагато простішим та займало набагато менше часу, потрібно використати патерн фасад [46] для даних програм.

Патерн фасад являється структурним патерном проектування, який надає простий інтерфейс до складної системи класів, програм, бібліотек або ж фреймворку. Так, як цей патерн є структурним, то він показує хороший та простий спосіб побудови зв'язків між деякими об'єктами, а саме між програмами, які вирішують задачі трансформації аудіо матеріалів в текст, переклад розпізнаного тексту на цільову мову, тобто мову користувача, та машинного озвучення

перекладеного тексту. Приклад патерну фасад у реалізації методі трансформації відеоматеріалів на мову користувача продемонстрований на рисунку 2.5.

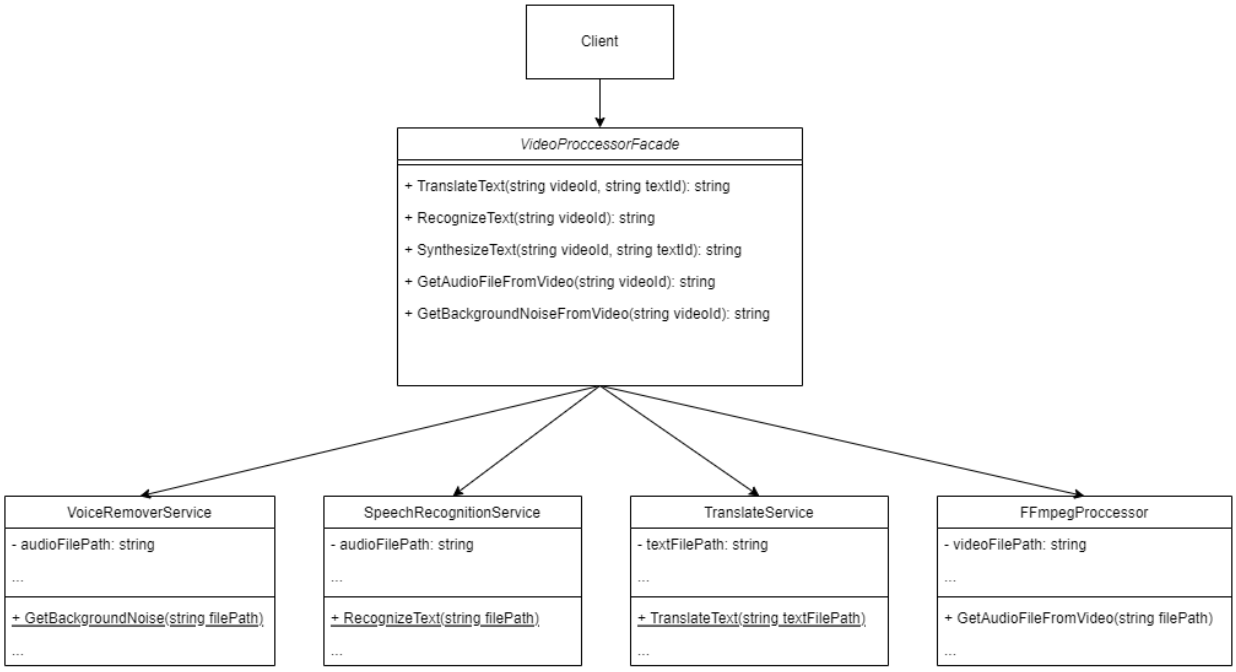

Рисунок 2.5 – Схема класів, що демонструє реалізацію патерна фасад для трансформації відео на мову користувача

Використовуючи патерн фасад у реалізації методу трансформації відеоматеріалів на мову користувача надається можливість легкої зміни системи та гнучкості системи до змін. Все це досягається тим, що ядро системи, тобто вся основна її логіка, викликається в одному місці, а код для кожного окремого сервісу створюється в окремому класі. Також для кожного сервісу бажано робити окремий інтерфейс і в такому випадку можна буде легко замінити цей клас на інший, якщо змінить кінцева програма або сервіс, який займається обробкою відео, перекладом тексту, розпізнаванням мовлення та машинним озвученням тексту.

Щоб надати можливість користувачам зберігати відеоматеріали, їх зміни розпізнаного та перекладеного текстів, озвучення на різних мовах, редагувати це все та видаляти, то потрібно налаштувати файловий сервер та базу даних для нього, щоб зберігати інформацію про те на якому сервері зберігається файл, всю інформацію про нього та до якого відео він відноситься. Для налаштування

файлового сервера потрібно налаштувати сервер Nginx [47], щоб мати змогу керувати налаштуваннями файлового серверу, добавляти різні фільтри, захистити сервер від різних можливих атак, тощо.

Користувача також потрібно постійно нотифікувати про стан системи, що відбувається з його даними і надати можливість спостерігати за роботою трансформації відеоматеріалів у текст та перекладу опрацьованого тексту на цільову мову, оскільки тільки так система буде дружньою для користувача. Для цього потрібно налаштувати ще один сервер, на якому буде реалізоване програмне забезпечення розроблене на технології Dotnet SignalR [48]. Ця технологія позволяє встановлювати постійне підключення з клієнтами та обмінюватись повідомленнями із підключеними клієнтами в режимі реального часу. Зазвичай цю технологію використовують для розробки сповіщень в системі, які працюють в режимі реально часу, кооперативних онлайн ігор, чатів, тощо. За допомогою цієї технології для реалізації методу трансформації відеоматеріалів на мову користувача можна в режимі реального часу відправляти користувачам текст який розпізнається або перекладається на сервері, а також відправляти сповіщення користувачам про зміни в системі.

Для того щоб налагодити комунікацію між серверною частиною та іншими програмами на інших серверах було виділено сервер та встановлено на нього програмне забезпечення «RabbitMQ». «RabbitMQ» [49] – це програмне забезпечення, що займається надсиланням та прийманням повідомлень від однієї програми до іншої, якщо вони розташовані, наприклад, на різних серверах. Також оскільки це програмне забезпечення не просто надсилає та приймає повідомлення, але й тримає їх у черзі, відфільтровує по даним яке повідомлення куди має піти і тому подібне, то воно являється одним із готових рішень для черг повідомлень. Це програмне забезпечення позволяє нам запускати обробку відео з інших машин, тобто, наприклад, якщо користувач натиснув кнопку для початку розпізнавання тексту, то запит від клієнта надсилається повідомлення до «RabbitMQ» про те, що обробка закінчилась і можна приступати до наступних дій. Черга повідомлень має налагоджувати комунікацію між програмним забезпеченням типу API, що

опрацьовує запити від кінцевого користувача, та іншими серверними програмами такими, як консольна програма для роботи з відео, консольна програма для роботи з сервісами Azure, а також іншими програмними забезпеченнями, що опрацьовують різну інформацію та не мають доступу до кінцевого користувача або ж прямого доступу до API. На рисунку 2.6 можна наочно побачити функціонал відправки повідомлення про початок розпізнавання мови та перекладу тексту. На даному рисунку можна побачити, як опрацьовуються запити від клієнта і як система реагує на них. Спочатку клієнт через клієнтську програму, що реалізована на технології React, відправляє запит на розділення відео та аудіо на окремі файли до серверу, на якому розгорнуте програмне забезпечення, що обробляє запити клієнта, опрацьовує їх та відправляє користувачеві відповідні дані у відповідь. Цей сервер у свою чергу поміщає повідомлення про те, що потрібно розділити відео та аудіо на окремі файли, у чергу повідомлень RabbitMQ.

У свою чергу сервер, на якому розгорнуте консольне програмне забезпечення для обробки відеоматеріалів, повинен постійно прослуховувати дану чергу та очікувати нових запитів для опрацьовування, звичайно, якщо він не зайнятий в даний момент обробкою якогось запиту. Таким чином можна горизонтально масштабувати систему без жодних змін в програмному коді програмного забезпечення, адже кожен сервер після того, як отримав повідомлення з черги повідомлень одразу його видаляє звідти і таким чином, якщо два сервера одночасно будуть очікувати повідомлення, то отримує його тільки один, а інший буде очікувати на нові запити далі. Це дає можливість обслуговувати велику кількість користувачів, адже якщо не буде вистачати потужності, то можна буде просто добавити ще один налаштований сервер в систему і він буде одразу готовий до використання. Це все можна легко налаштувати та автоматизувати за допомогою сервісів Azure та технології Kubernetes [50]. У такому разі навіть не потрібно буде самотужки розгортати тому, що система буде автоматично сама масштабуватись та розгортати нові сервери у разі потреби.

Після того, як сервер закінчив опрацьовувати відео, то він надсилає повідомлення серверу для сповіщення користувачів в реальному часі, на якому

розгорнуте програмне забезпечення розроблене на технології Dotnet SignalR, що реалізовує даний функціонал, а цей сервер у свою чергу сповіщає користувача, що відео опрацювалось.

Подібний функціонал відбувається і для запиту на розпізнавання тексту з відео та перекладу тексту. Відмінність тільки в тому, що під час обробки відео або тексту сервер для сповіщення користувача в реальному часі, тобто, сервер SignalR, постійно перевіряє текстовий файл на зміни і як тільки у файлі щось змінюється, то сервер одразу ж передає зміни користувачу. Таким чином користувач може спостерігати за розпізнаванням або перекладом тексу в режимі реального часу, що робить систему дружньою для користувача.

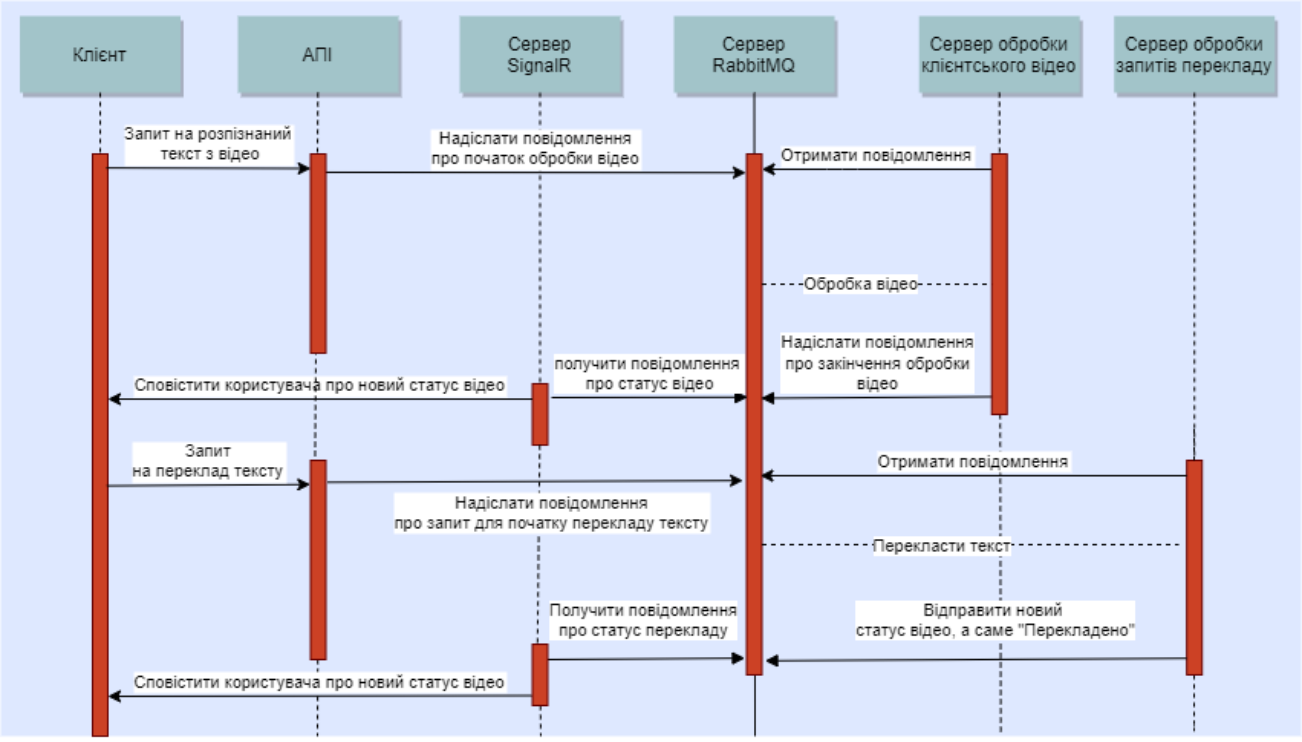

Рисунок 2.6 – Схема послідовності відправки повідомлення для розпізнавання мови і перекладу тексту

Отже, була розроблена загальна структура системи трансформації відеоматеріалів на мову користувача, розроблена схема послідовності відправки повідомлення для розпізнавання мови та перекладу тексту, розроблена архітектура серверів реалізації проекту для методу трансформації відеоматеріалів на мову

користувача, розроблена схема класів, що демонструє реалізацію патерна фасад для трансформації відеоматеріалів на мову користувача

Висновки до розділу 2

1. Запропоновано метод трансформації відеоматеріалів на мову користувача.

2. Розроблено алгоритми трансформації відеоматеріалів на мову користувача та озвучення тексту.

3. Розоблена архітектура системи, досліджено технології, які будуть використані для реалізації методу трансформації відеоматеріалів на мову користувача.

3 РЕАЛІЗАЦІЯ МЕТОДУ ТРАНСФОРМАЦІЇ ВІДЕОМАТЕРІАЛІВ НА МОВУ КОРИСТУВАЧА

3.1 Розробка бази даних системи трансформації

Логічна модель даних детально описує дані та взаємозв’язки на високому рівні. Сюди не входить те, як дані представлені в базі даних [51], але описується на абстрактному рівні. В основному він включає сутності та зв’язки між ними.

Логічна модель даних включає первинні ключі кожної сутності, а також зовнішні ключі. При створенні логічної моделі даних перші сутності та їх взаємозв'язки ідентифікуються за допомогою ключів. Потім ідентифікуються атрибути кожної сутності. Після цього багато-багато відносин вирішуються і відбувається нормалізація. Логічна модель даних не залежить від системи управління базами даних, оскільки не описує фізичну структуру реальної бази даних. При проектуванні логічної моделі даних неформальні довгі імена можуть використовуватися для сутностей та атрибутів.

Фізична модель даних описує, як дані насправді зберігаються в базі даних. Він включає специфікацію всіх таблиць та стовпців всередині них. Специфікація таблиці включає такі деталі, як назва таблиці, номер стовпців s, а специфікація стовпців включає ім'я та тип даних. Фізична модель даних також містить первинні ключі кожної таблиці, а також показує взаємозв'язок між таблицями за допомогою зовнішніх ключів. Більше того, фізична модель даних містить обмеження, що застосовуються до даних та компонентів, таких як тригери та збережені процедури.

Фізична модель даних залежить від того, яку систему управління базами даних ми будем використовувати. Тобто фізична модель даних для MySQL [52] буде відрізнятись від моделі даних для PostgreSQL.

Логічна модель бази даних для деяких основних таблиць [53], які використовуються для збереження завантажених користувачами відео, відправки повідомлень в чаті та проведення онлайн відеотрансляцій представлена на рисунку 3.1.

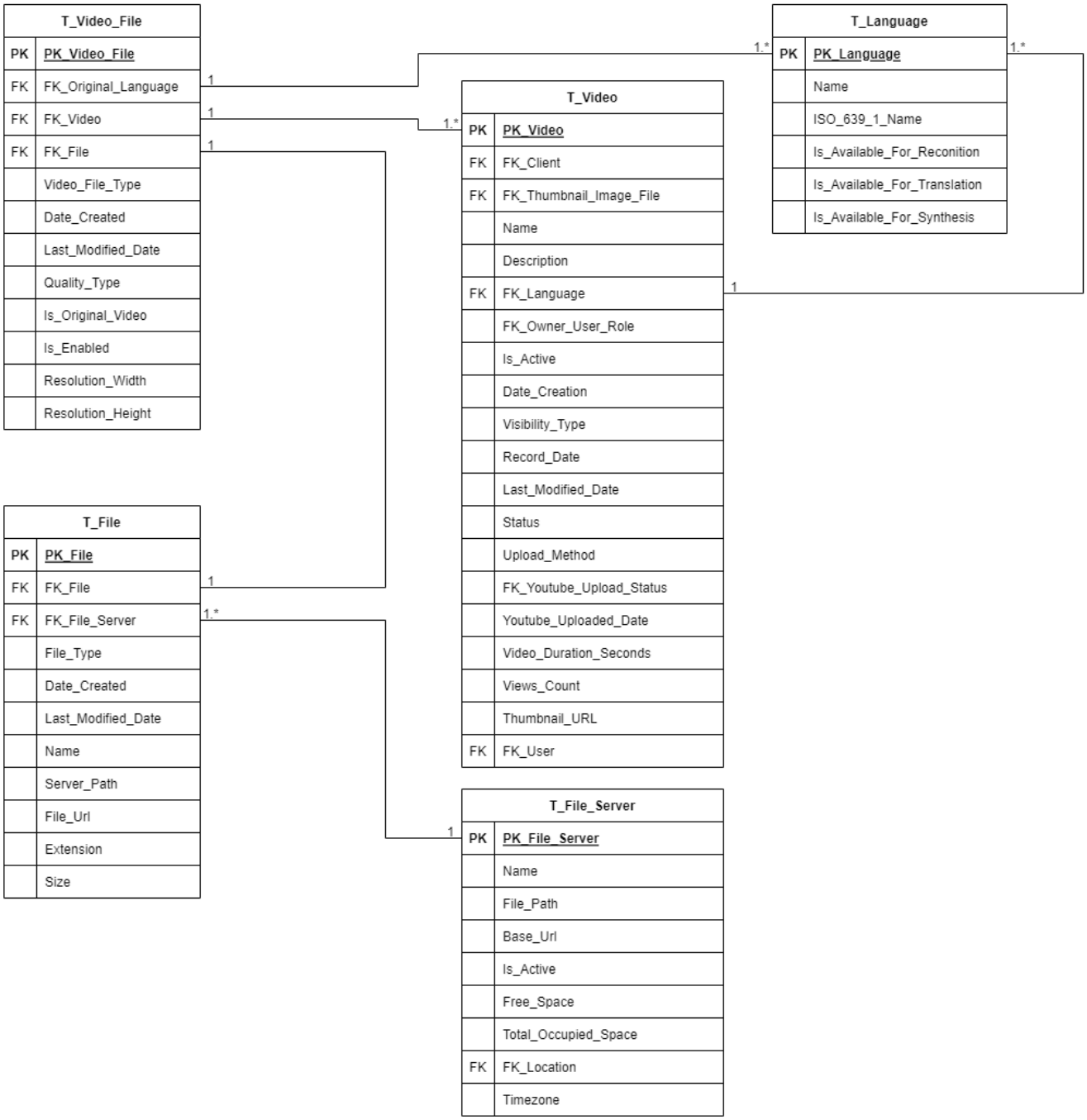

Рисунок 3.1 – Логічна модель бази даних для основних таблиць системи, які призначені для роботи з відео

Повна база даних складається з приблизно ста таблиць, які тільки відносяться до основних функцій платформи. Також є ще інші таблиці, які використовуються в інших проектах, проте вони загальні для всіх, наприклад, таблиці для авторизації користувача. Тобто, щоб не реєструвати одного користувача на всіх проектах окремо після реєстрації на одному з них він зможе авторизуватись з тими ж даними і на інших. Даний функціонал також називають єдиним входом. Існують навіть

сервіси які дозволяють реалізовувати цей функціонал швидше та простіше без написання даної логіки у власному проекті. Проте у даній системі цей функціонал реалізований власними силами, що позволяє змінювати його логіку, якщо це потрібно.

Фізична модель бази даних для деяких основних таблиць, що призначені для збереження даних про відео, повідомлення в чатах та онлайн відеотрансляції, представлена на рисунку 3.2.

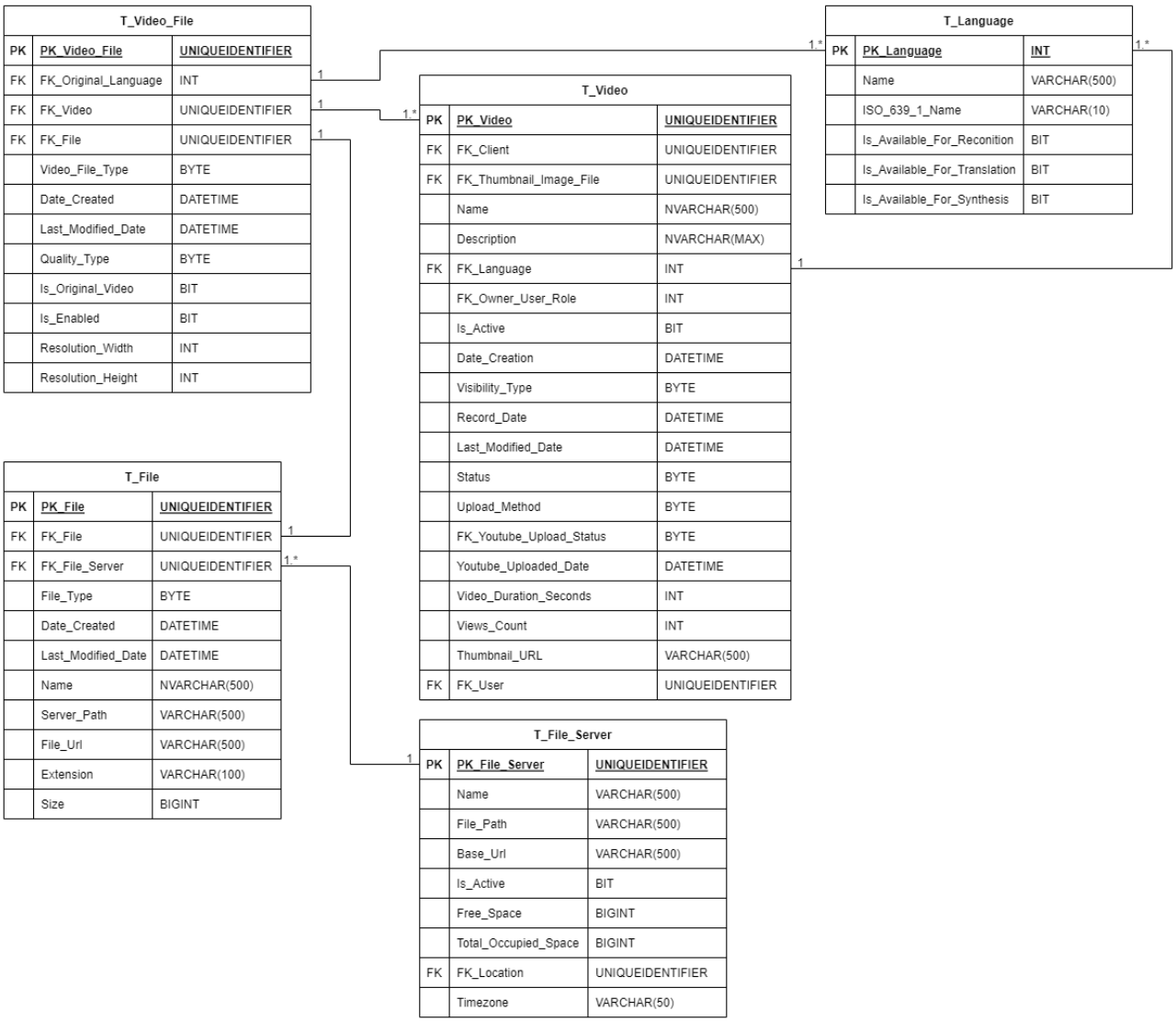

Рисунок 3.2 – Фізична модель бази даних для основних таблиць системи

Дана фізична модель бази даних відображає основні таблиці, які потрібні для того, щоб зберігати відеоматеріали та інші файли різного типу на файловому сервері та зберігати всю потрібну інформацію про них у базі даних.

Для того, щоб зберегти інформацію про відео, а саме про назву відео, його опис, дати створення, завантаження та будь-якої зміни цього відео, готові посилання, по яких можна завантажити один із кадрів цього відео, тощо, було створено таблицю «T_Video». Дана таблиця описує відео і використовується для того, щоб надати всю основну інформацію про відео користувачеві. Тут також присутня колонка, що відповідає за статус відео.

Статус відео змінюється в залежності від того на якому етапі зараз воно. Тобто коли користувач буде завантажувати відео, то спочатку у базі даних в дану таблицю добавиться запис про це відео і статус відео буде як «Завантажується», а після того, як добавився запис в базу даних, то почнеться завантаження. Коли ж відео повністю завантажиться на файловий сервер, то статус відео буде змінено на «Опрацьовуються різні якості відео». Оскільки дана платформа має підтримувати максимальну кількість різних пристроїв, операційних систем та програмних забезпечень, то після завантаження відео потрібно опрацювати це відео і в результаті отримати те ж відео, проте в різних якостях.

Таким чином відео зможе відображатись у користувачів, в яких можуть бути проблеми із швидкістю передачі даних. Також це потрібно для того, щоб створити різні копії відео з різними кодеками, бо тільки так можна реалізувати підтримку різних пристроїв, систем та програмних забезпечень для системи. Після того, як якості опрацювались, то статус відео зміниться на «Готове до перегляду». Даний статус означає, що відео може бути переглянуте користувачами, а також для користувачів має відкритись доступ до перекладу даного відео на інші мови.

Колонка оригінальної мови відео у табличці, яка відображає основну інформацію про відео, потрібна для того, щоб від користувача отримати інформацію про те, яка мова присутня в оригінальному відео. Це потрібно для того, щоб ініціалізувати перший етап реалізованого методу трансформації відео на мову користувача, а саме етап трансформації відеоматеріалів у текст. А також

інформацію про оригінальну мову відео потрібна для того, щоб показувати цю інформацію користувачу та надавати йому можливість переключати мови перегляду відео.

Колонки з префіксом «Youtube» у табличці з основною інформацією відео потрібні для того, щоб надати користувачу можливість загрузки даних відео на популярний сервіс «Youtube». Це позволить зробити систему дружньою для користувача і полегшить роботу з відео та його перекладами для користувача.

В майбутньому для підтримки даної системи потрібно слідкувати за подібними сервісами та добавляти їх в базу даних, щоб підтримувати максимальну можливу кількість популярних сервісів, які позволяють користувачами загружати та переглядати відео.

Для збереження інформації про файли, які відносять до відео, створена табличка в базі даних під назвою «T_Video_File». У даній таблиці є колонка, що зберігає ключ до файлу, а інформацію про самі файли та файлові сервери, на яких вони зберігаються, потрібно створювати в окремих таблицях. Така модель бази даних була розроблена тому, що дана таблиця відображає інформацію не про самий файл, а про те чим цей файл являється для певного відео, до якого він відноситься. Тобто у даній табличці є інформація про те до якого відео цей файл відноситься, якість відео, якщо даний файл являється відеоматеріалом, чи являється даний файл оригінальним завантаженим файлом користувача та тип даного файлу. Тип файлу відображає те, що саме це є, тобто, наприклад, коли користувач загружає відео, то тип файлу в даній табличці буде «Відео», а коли користувач запустить перший етап реалізованого методу трансформації відеоматеріалів на мову користувача, то тип файлу, який створиться в результаті даного етапу, буде «Розпізнаний текст». Коли ж користувач запустить переклад тексту, то тип файлу, який створиться в результаті виконання буде «Перекладений текст».

Оскільки для даної системи потрібно дуже багато вільного простору на сервері, щоб зберігати відео та результати роботи реалізованого методу трансформації відеоматеріалів на мову користувача, то було вирішено розробити базу даних та систему так, щоб була можливість підтримки більше одного

файлового серверу. Саме тому у базі даних було створено табличку з назвою «T_File_Server», яка вміщає в собі інформацію про всі доступні сервери. У цій табличці є колонки, які позволяють дізнаватись системі чи можна туди завантажувати файл певного розміру. Ці колонки описують вільний простір на файловому сервері та простір, який вже зайнятий на сервері. Ці дані дають можливість вичислити чи вміщається певний файли на сервер. Також у цій табличці є колонка, яка показує чи сервер активний, тобто, чи він запущений на даний момент. Ця колонка позволяє системі дізнаватись чи сервер активний, тобто, чи можна його на даний момент використовувати для загрузки файлів та для читання файлів з нього.

Таким чином якщо файловий сервер буде відключений по деяким причинам, то система може адекватно відреагувати на це і відображати коректну помилку користувачам, які відкриють відео, що зберігається на даному сервері.

Для того, щоб мати можливість загружати відео на сервер, який найближче до користувача, щоб зменшити затримку під час передачі даних, потрібно було добавити також колонку про поточне розташування сервера. За допомогою інформації від клієнта про його поточне розташування можна дізнатись відстань до найближчого сервера і вибрати саме його для загрузки відео.

Табличка з назвою «T_File» у базі даних відображає усі потрібні дані про файл та його розташування на файловому сервері. По цій табличці можна дізнатись про те, якого типу файл, наприклад, відео, аудіо, фото, текст, тощо. Також у цій табличці присутні колонки, які містять інформацію про назву файлу, його розширення та фізичне розташування на сервері, по цим колонкам можна скласти повний шлях до файлу, щоб в подальшому мати змогу управляти ним, наприклад видалити, зчитати або змінити його. Для того, щоб не робити зайвих запитів до бази даних було вирішено добавити сюди також колонку, яка відображає повне посилання, по якій можна зчитати даний файл.

Остання табличка, що була показана у фізичній моделі бази даних для основних таблиць системи на рисунку 3.7, має назву «T_Language» та створена для того, щоб містити в собі інформацію про мови, які підтримує система, що

реалізовує метод трансформації відеоматеріалів на мову користувача. У цій таблиці є дві колонки для імені мови, тобто для повного імені та для скороченого. А також три колонки, які можуть приймати значення тільки 1 та 0, для того, щоб показати яка мова підтримується кожним етапом у реалізованому методі трансформації відеоматеріалів на мову користувача.

Таким чином на кожному етапі буде можливо показати користувачеві можливі мови для того, щоб провести трансформацію відеоматеріалу в текст, перекласти розпізнаний текст та провести трансформацію розпізнаного тексту в аудіофайл.

Повну логічну модель бази даних для основних таблиць систем, які призначені для роботи з відео, обробки відео у різні формати, перекладу відео, тощо, було представлено у додатку Б.

У цій базі даних також присутня ще табличка з назвою «T_Video_Process_Status», яка відображає інформацію про статус перекладу та опрацювання відеоматеріалу. Тобто як тільки користувач завантажує відео і воно розбивається на декілька різних якостей, то для кожної якості в базу даних добавляється запис про неї та її поточний статус, як тільки опрацювання відео закінчується, то статус для відповідного запису змінюється на відповідний, а саме на «Готове до використання». Теж саме відбувається під час запуску та закінчення кожного етапу перекладу відеоматеріалів на мову користувача. Ця табличка потрібна, щоб навіть якщо користувач вийде з програми або зачинить браузер, то поточний статус на якому він зупинився під час перекладу відео зберігся, а також для того, щоб система мала змогу дізнатись ці дані та не давати користувачу почати, наприклад, озвучення перекладеного тексту, якщо даного тексту немає на файловому сервері та етап перекладу тексту ще не закінчений.

Отже, було спроектовано базу даних для системи трансформації відео на мову користувача у гіперконвергентній платформі залучення громад та реалізовано її у середовищі MSSQL.

3.2 Програмна реалізація

Основна мова програмування для реалізації системи – C#. Ця мова належить до технології Dotnet. На вибір її вплинуло те, що гіперконвергентна платформа «Iconnect» в загальному реалізована на даній технології, а саме на двох її мовах: C# та Visual Basic. Оскільки технологія Dotnet позволяє об’єднувати в собі код методів, написаних на різних мовах цієї технології , то була обрана мова програмування C#.

Дана технологія позволяє розбивати програмний код на модулі та підключати їх у проєкти. Це надає можливість написати програмний код, що буде реалізовувати певний модуль та легко підключити його у іншу систему. Ця можливість позволяє легко проєктувати архітектуру системи, адже кожен окремий модуль може містити у собі проєкт із контрактами, які також називають моделями, що описують інформацію про те, шо саме повертає певна функція модуля. Такі контракти складаються із набору змінних, що представляють інформацію про певний об’єкт, що може бути повернений функцією модуля. За допомогою таких контрактів окремі модулі системи можуть коректно передавати дані між собою. Приклад програмного коду такого контракту, що вміщає в собі інформації про відеоматеріал та його відношення до інших об’єктів наведено нижче.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace VideoTranslate.WebApiClient.DTO
{
   public record VideoInfo
   {
      public Guid Id { get; set; }
      public string Name { get; set; }
      public string Description { get; set; }
```

```
        public Guid? ThumbnailFileId { get; set; }
        public string? ThumbnailUrl { get; set; }
        public DateTime CreatedOnUtc { get; set; }
    }
}
```

Бібліотеки проєктів з такими контрактами зберігаються на окремому виділеному сервері для того, щоб для кожного окремого модуля була можливість завантажити таку бібліотеку та використати її у програмному коді для коректного передавання даних. Даний підхід до розробки модулів в системі дає можливість модулям спілкуватись із будь-яким іншим модулем, що дає можливість використати архітектуру мікросервісів.

Мікросервіси – це архітектура, за якою система будується, як сукупність невеликих, самодостатніх, незалежних, не тісно зв’язаних сервісів, що передають дані між собою за допомогою легких у використанні та мало затратних, відносно потужності процесора та оперативної пам’яті сервера, механізмів. Для передачі даних між сервісами зазвичай використовують такі механізми, як черга повідомлення або HTTP. Кожен сервіс створюється для вирішення однієї задачі або декілька невеликих задач.

Такий підхід до розробки програмного забезпечення дає можливість швидко вносити змінити у проекти. Проте він також і додає додаткову складність проекту, адже ускладняється система моніторингу та доводиться більше часу приділяти не розробці системи, а розгортанню системи.

Оскільки реалізація трансформації відеоматеріалів на мову користувача споживає велику кількість оперативної пам’яті та вимагає максимальної потужності процесора, то дана система була реалізована за архітектурою мікросервісів. Адже за допомогою такої архітектури масштабування системи проводиться набагато простіше, що позволяє підтримувати велику кількість користувачів, а також сервіси, що вимагають максимальної потужності, розгортаються на власних виділених серверах та не нагружають всю систему. Під час розробки даної системи було виділено такі основні мікросервіси:

− «FFmpegProcessorService», що відповідає за опрацювання відеоматеріалів;

− «VideoTranslatorService», який відповідає за основні кроки для проведення перекладу відео;

− «AuthenticationService», основна задача якого полягає в авторизації та реєстрації користувачів;

− «VideoService», що відповідає за опрацьовування запитів від користувачів.

Для всіх сервісів у системі був написаний програмний код у вигляді бібліотеки, що використовується для відправки повідомлень у чергу повідомлень RabbitMQ та для очікування нових повідомлень з цієї черги. Програмний код наведений нижче представляє головні змінні для підключення до черги повідомлень.

```
public class ServiceRabbitMQ : IHostedService
{
private readonly ILogger<ServiceRabbitMQ> logger;
private readonly RabbitMQConfiguration rabbitMQConfiguration;
private ConnectionFactory connectionFactory;
private IConnection connection;
private IModel channel;
```

У наведеному коді створюється клас «ServiceRabbitMQ», який унаслідується від інтерфейсу «IHostedService». Це наслідування потрібне, щоб відправляти та очікувати повідомлення із черги асинхронно та не закривати основний потік виконання програмних функцій. Після опису класа оголошуються змінні класу. Змінна «logger» вміщає в собі об’єкт із функціоналом для збереження повідомлень про помилку в системі у файловій системі сервера. Оскільки використовується інтерфейс «ILogger», то даний код не залежить від конкретної реалізації об’єкту логування, що дає можливість використати цей клас у будь-якій системі розробленій на технології Dotnet. Після цього оголошується змінна

«rabbitMQConfiguration», яка потрібна для збереження даних про підключення до черги повідомлень. Наступні змінні потрібні для того, щоб створювати та зберігати підключення до черги повідомлень.

Код, що наведений нижче, представляє спосіб підключення до черги повідомлень.

```
public ServiceRabbitMQ(
ILogger<ServiceRabbitMQ> logger,
RabbitMQConfiguration rabbitMQConfiguration)
{
this.logger = logger;
this.rabbitMQConfiguration = rabbitMQConfiguration;
this.connectionFactory = new ConnectionFactory()
{
    HostName = this.rabbitMQConfiguration.HostName,
    UserName = this.rabbitMQConfiguration.User,
    Password = this.rabbitMQConfiguration.Password
};

this.connection = this.connectionFactory.CreateConnection();
this.channel = this.connection.CreateModel();
}
```

У конструктор класа, що призначений для роботи з чергою повідомлень, передається об’єкт логування та об’єкт конфігурацій для підключення до черги повідомлень. Як тільки об’єкт даного класу буде створений, то запуститься даний конструктор і у свою чергу створить підключення до черги повідомлень за допомогою вхідних даних

Наведений нижче програмна функція запускає прослуховування черги повідомлення.

```
private void Run()
{
```

```
this.channel.QueueDeclare(queue: this.QueueName,
                          durable: false,
                          exclusive: false,
                          autoDelete: false,
                          arguments: null);
 var consumer = new EventingBasicConsumer(this.channel);
 consumer.Received += OnMessageRecieved;
 this.channel.BasicConsume(queue: this.QueueName,
                          autoAck: false,
                          consumer: consumer);
}
```

Дана функція не тільки запускає асинхронне прослуховування черги повідомлень, але й запускає відповідну функцію, що повинна виконатись після отримання повідомлення, а на її вхід подає отримане повідомлення.

Таким чином кожен окремий сервіс системи має змогу спілкуватись з іншими, а за допомогою контрактів сервіси мають повну інформацію про те, які саме дані мають бути отримані або відправленні.

Також було розроблено програмне забезпечення для системи, щоб запускати саме ті типи розпізнавання мовлення, перекладу тексту та озвучення тексту, які встановленні в базі даних. Адже якщо для системи буде потрібне змінити метод трансформації мовлення у текст, перекладу тексту або озвучення перекладеного тексту, то для цього буде достатньо додати лише нові кінцеві точки у базу даних та вибрати їх.

Приклад головних функцій у програмному коді даного програмного забезпечення наведено нижче.

```
IExecutor executor = null;

if (technology == TechnologyTypes.Vosk)
{
executor = new Executors.VoskExecutor();
}
else if (technology == TechnologyTypes.Coqui)
```

```
{
executor = new Executors.CoquiExecutor();
}
else if(technology == TechnologyTypes.DeepSpeech)
{
    executor = new Executors.DeepSpeechExecutor();
}

if (executor != null)
{
executor.Execute(webSocketUrl,inputFileFullPath, outputFileFullPath);
Console.WriteLine("Wav file was Successfully Recognized.");
}
else
{
Console.WriteLine("");
    Console.WriteLine("Please, Select correct technology.");
}
```

Було створено інтерфейс «IExecutor» для того, щоб в подальшому добавляти програмний код для інших систем перетворення мовлення в текст за допомогою створення нового класу для нього та реалізовуючи в ньому даний інтерфейс. Відповідно програмний код може використовувати будь-який метод перетворення не зважаючи на його реалізацію. Як представлено у коді спочатку зчитується інформація з бази даних про доступну на даний момент систему перетворення мовлення в текст. А тоді не зважаючи на те, який клас передався у змінну, завжди запускаються метод запуску даної системи. Це робить систему гнучкою, адже таким чином вся потрібна логіка для вибору методу завжди міститься в одному місці.

Кожен клас, який реалізовує інтерфейс «IExecutor» може описувати функціонал розпізнавання мовлення по іншому. Приклад програмного коду, що описує клас, який реалізовує даний інтерфейс, наведено нижче.

```
 private async Task DecodeFile()
```

```
        {
            Console.WriteLine("");
            Console.WriteLine($"{DateTime.UtcNow.Date.Year}-{DateTime.UtcNow.Date.Month}-
{DateTime.UtcNow.Day}
{DateTime.UtcNow.Hour}:{DateTime.UtcNow.Minute}:{DateTime.UtcNow.Second}  ||  INFO  ||
Connectiong to the Server...");

            ClientWebSocket ws = new ClientWebSocket();
            ws.Options.RemoteCertificateValidationCallback += (sender, cert, chain, sslPolicyErrors) =>
true;

                await ws.ConnectAsync(new Uri(this.webSocketUrl), CancellationToken.None);

            FileStream fsSource = new FileStream(
                this.inputWav,
                FileMode.Open,
                FileAccess.Read);

            int sizeBuffer = 1024*512;
            byte[] data = new byte[sizeBuffer];
            while (true)
            {
                int count = fsSource.Read(data, 0, sizeBuffer);
                if (count == 0)
                {
                    break;
                }

                await this.SendFile(ws, data, count);
            }

            await this.ProcessResult(ws);
            await          ws.CloseAsync(WebSocketCloseStatus.NormalClosure,          "OK",
CancellationToken.None);
        }
```

У даному коді спочатку створюється підключення до веб-сокетів сервера, на якому відбувається розпізнавання мовлення. Оскільки на кінцевому сервері не встановлений сертифікат безпеки, то потрібно відключити перевірку на нього у налаштуваннях підключення. Після цього запускається підключення до сервера та очікується на результат від кінцевого сервера. Якщо підключення до сервера пройшло успішно, то відкривається потік, у який записується зчитаний з файлового сервера файл, а саме аудіоматеріал. Цей аудіоматеріал відправляється за допомогою веб-сокетів на кінцевий сервер. Як тільки файл було загружено, то відбувається процес розпізнавання мовлення і під час його виконання результат поступово відправляється назад у відповіді. Як тільки процес розпізнавання закінчився, то виконання роботи припиняється і програма повертається назад до очікування повідомлень із черги повідомлень RabbitMQ. Так як, для реалізації даного сервісу використовуються інтерфейси, інверсія залежностей та інші корисні підходи до розробки програмного забезпечення, то додавання нових методів розпізнавання мовлення, перекладу тексту, озвучення перекладеного тексту до системи не займе багато часу.

Фрагменти коду серверної частини програмного забезпечення для обробки запитів перекладу представлено у додатку А.

Отже, було розроблено програмну реалізацію, а саме систему трансформації відеоматеріалів на мову користувача, представлено фрагменти коду системи.

3.3 Дослідження функціонування системи

Під час, а також і після завершення, реалізації методу трансформації відеоматеріалів на мову користувача потрібно провести дослідження функціонування реалізованої системи. Спочатку потрібно провести дослідження функціонування даної системи окремо від гіперконвергентної платформи «Iconnect», а тоді провести дослідження функціонування системи разом із

гіперконвергентною платформою, щоб зрозуміти чи дана реалізована система сумісна із платформою.

Оскільки для реалізації методу трансформації відеоматеріалів на мову користувача була обрана технологія Dotnet, то для проведення дослідження функціонування системи можна використати існуючі бібліотеки, які сумісні з даною технологією. Наприклад, можна використати бібліотеку xUnit, що позволяє автоматизувати деякі процеси дослідження функціонування системи та перевірити чи працюють деякі функції у коді.

Проте дана бібліотека може провести дослідження функціонування не тільки для окремих функцій в одному проєкті, але й провести дослідження функціонування декількох модулів одночасно, а саме те, як вони співпрацюють разом. Тому окрім модульного дослідження функціонування потрібно також провести інтеграційне, що позволить більше дослідити функціонування системи, що реалізовує метод трансформації відеоматеріалів на мову користувача, та допоможе реалізувати дану систему якісно. Для цього можна також використати бібліотеку xUnit.

Потрібно також автоматизувати дослідження функціонування системи, оскільки його потрібно проводити після кожної зміни в реалізації методу трансформації відеоматеріала на мову користувача, адже змін буде багато і робити це вручну кожного разу буде важко. Для цього можна використати сервіси Azure. Тобто спочатку потрібно розробити невелике програмне забезпечення, що буде розгортати в тестовому середовищі систему, що реалізовує метод трансформації відеоматеріалів на мову користувача, розгортати базу даних та проводити автоматизоване модульне та інтеграційне дослідження функціонування системи, що було реалізоване за допомогою бібліотек xUnit.

Для реалізації даного автоматизованого дослідження функціонування системи потрібно зареєструватись на платформі Azure, щоб получити доступ до сервісів даної платформи. Після цього потрібно зберегти весь написаний код на платформі Github, оскільки саме звідси сервер, що призначений для автоматизованого дослідження функціонування системи, буде отримувати

актуальний код системи, що реалізовує метод трансформації відеоматеріалів на мову користувача. Тоді потрібно написати невеличкий скрипт, у якому буде описана послідовність команд, які потрібно запустити, щоб створити з вихідного коду програму, запустити її, розгорнути базу даних, провести дослідження функціонування системи, а після цього зберегти результати дослідження у базу даних або у файловій системі сервера, на якому розташована дане програмне забезпечення для дослідження функціонування. Після цього кожен раз, як у коді будуть проводитись зміни, то сервіси Azure будуть отримувати інформацію про зміни та запускати автоматизоване дослідження функціонування системи. Важливо також відмітити, що для того, щоб налаштувати цю систему потрібно також власний сервер, де будуть розгортатись дані системи, оскільки сервіси Azure пропонують тільки автоматизацію для розгортання систем, запуску команд для збірки або ж скриптів написаних на інших мовах в тестовому середовищі.

На рисунку 3.3 зображено приклад роботи сервісів Azure для проєкту, що призначений для приймання запитів від користувача, їх обробку та видачі результату користувачу у вигляді JSON.

Як показано на прикладі спочатку скрипт повинен викачати останню версію програмного коду з платформи Github. Сповіщення про те, що код змінився і потрібно запустити автоматизоване дослідження функціонування системи, відправляє платформа Github і дані сповіщення можна налаштувати, що дає змогу розробити власну подібну систему або ж підключити будь-яку іншу подібну систему до платформи Github. Після цього завантажуються всі необхідні бібліотеки для того, щоб розпочати збірку.

Після закінчення завантаження бібліотек використаних у коді, як вже зазначалось, розпочинається збірка програми. У результаті збірки програми можуть появитись бібліотеки, які потрібно завантажити на сервер артефактів, щоб мати змогу ці бібліотеки використовувати у інших проектах.

Після цих кроків починається налаштування середовища та запуск програми, а перед запуском нової версії програми попередня версія, звичайно, зберігається на

файловому сервері, що дає змогу в будь який момент повернути попередню версію, якщо нова версія, наприклад, не працює.

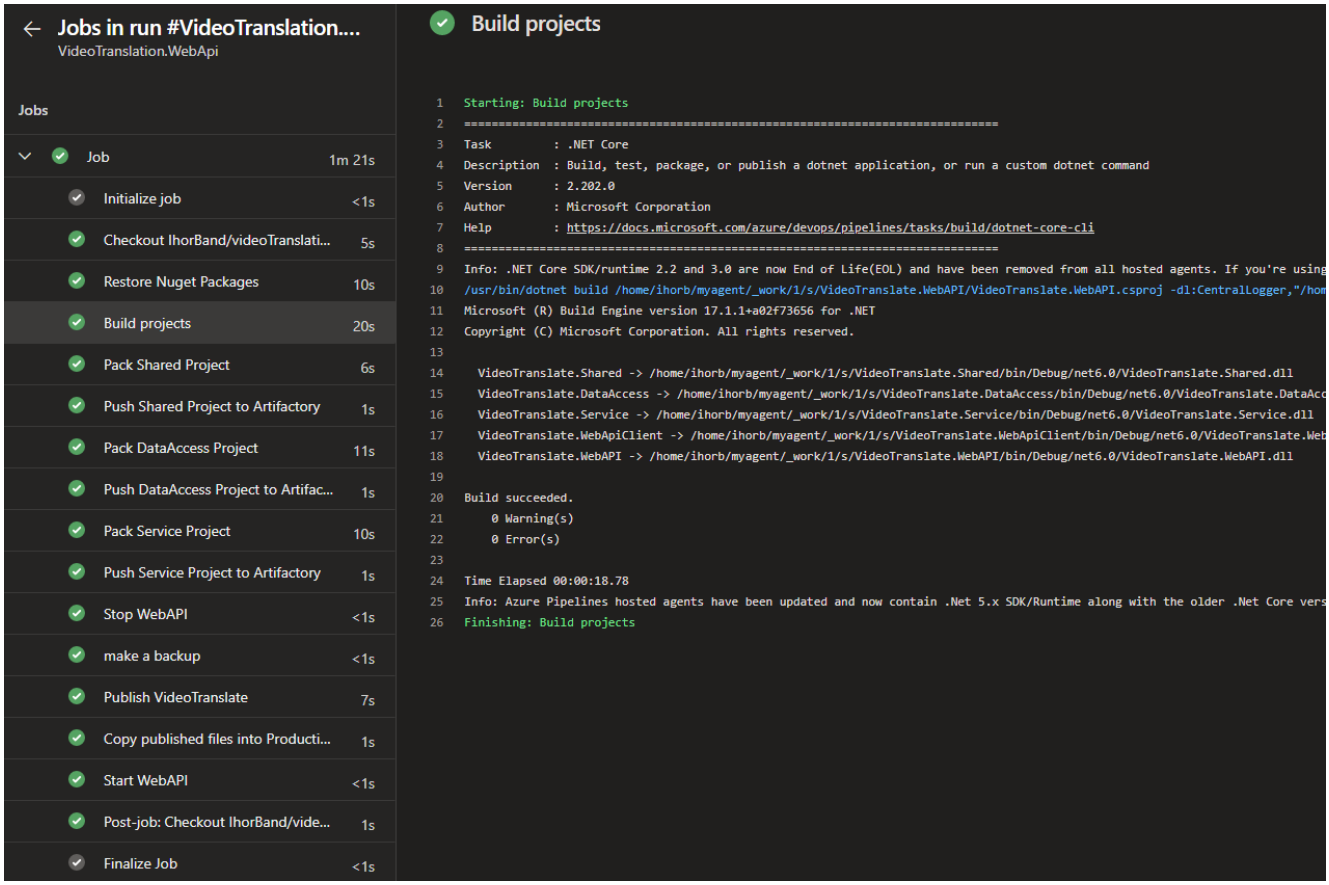

Рисунок 3.3 – Приклад автоматизації формування складу продукту проєкту для подальшого дослідження функціонування системи

На рисунку 3.4 представлено повний цикл автоматизації дослідження функціонування системи із використанням сервісів Azure та платформи Github.

Спочатку користувач вносить зміни в код та завантажує його у систему контролю версій. Система контролю версій у свою чергу сповіщає про зміни сервіси Azure. Як тільки Azure отримує таке сповіщення, то завантажує оновлений код на підключений до них сервер, шукає у коді спеціальні файли, у яких буде описано скрипт для розгортання системи на цьому сервері. Після того, як система була розгорнута, то запускаються скрипти для проведення автоматизованого дослідження функціонування системи. Результати його записуються на файловий сервер для ведення журнала дій на даному сервері. А також одразу після отримання

результатів вони відправляються розробнику, щоб той, у свою чергу, мав змогу проаналізувати їх.

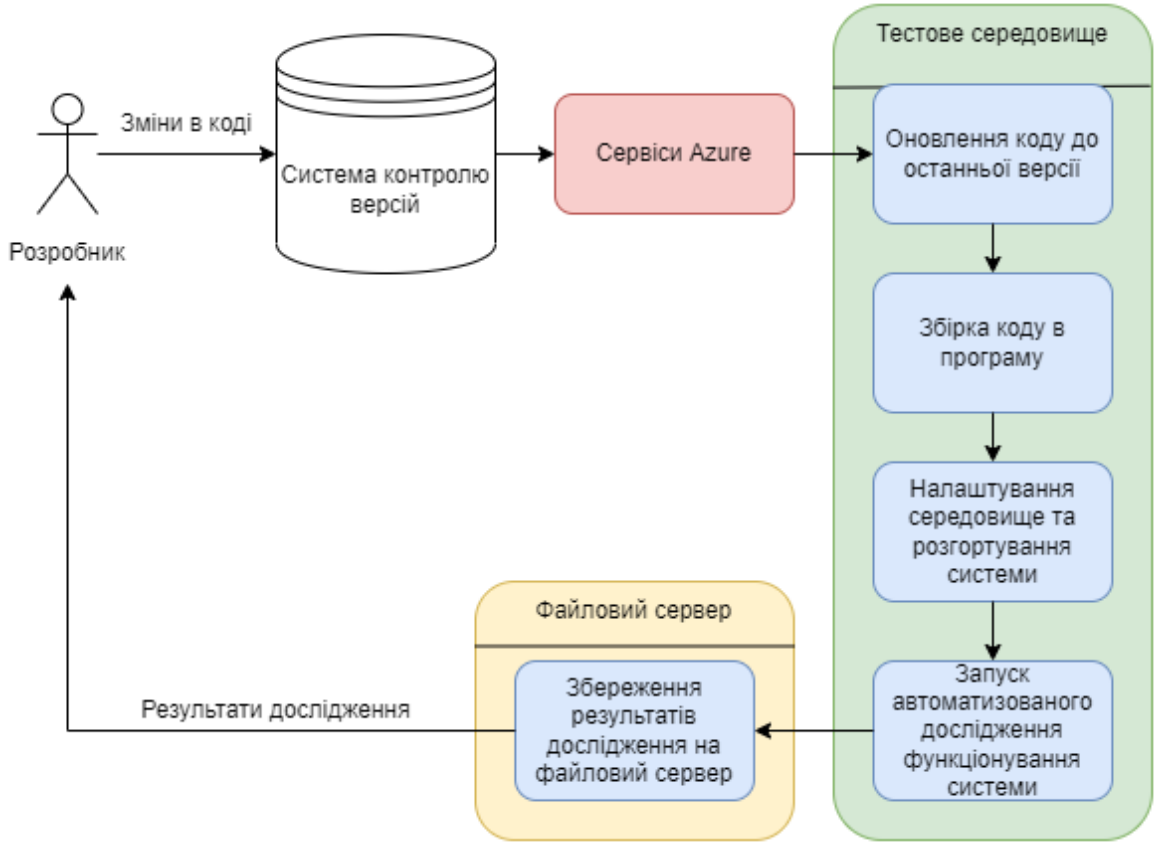

Рисунок 3.4 – Схема перевірки коректного функціонування системи

Отже, було проведено дослідження функціонування системи, розроблено автоматизацію дослідження функціонування та представлено схему представлення етапів автоматизації дослідження функціонування системи.

Висновки до розділу 3

1. Спроектовано базу даних для системи трансформації відео на мову користувача у гіперконвергентній платформі «Iconnect» та реалізовано її у середовищі MSSQL.

2. Розроблено програмну реалізацію, представлено фрагменти коду системи.

3. Проведено дослідження функціонування системи, використано набір бібліотек для автоматизації перевірки коректного функціонування системи.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи:

1. Досліджено існуючі підходи до перетворення відеоматеріалів на мову користувача.

2. Проведено аналіз літературних джерел з досліджуваної тематики.

3. Проаналізовано функціонування гіперконвергентної платформи «Iconnect».

4. Розроблено метод трансформації відеоматеріалів на мову користувача та реалізовано у вигляді системи, яка розширить функціональні можливості гіперконвергентної платформи «Iconnect».

5. Досліджено функціонування системи трансформації відеоматеріалів на мову користувача.

6. Результати плануються до використання (додаток В).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zoom [Електронний ресурс] – Режим доступу: https://www.zoom.us
2. Google Meet [Електронний ресурс] - Режим доступу: https://meet.google.com
3. Машинне озвучення тексту [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Speech_synthesis
4. Фонема [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Phoneme
5. Фонетичний аналіз [Електронний ресурс] - Режим доступу: https://studfile.net/preview/5186695/page:2/
6. Розпізнавання мовлення в системах штучного інтелекту [Електронний ресурс] / Бердников О.М., Богуш К.Ю., Богуш Ю.П. // 2012 - Режим доступу: http://its.iszzi.kpi.ua/article/view/52717/48777
7. Смисловий аналіз тексту [Електронний ресурс] - Режим доступу: https://studfile.net/preview/2227325/page:9/
8. Gender Recognition by voice. [Електронний ресурс] / Kory Becker // 2019 - Режим доступу: https://www.kaggle.com/datasets/primaryobjects/voicegender
9. Розпізнавання спонтанного мовлення на основі акустичних композитних моделей слів у реальному часі. [Електронний ресурс] / Робейко В.В., Сажок М.М. // 2012 - Режим доступу: http://dspace.nbuv.gov.ua/handle/123456789/57739
10. Машинний переклад у сучасному суспільстві [Електронний ресурс] / Тетяна Смірнова - Режим доступу: https://dspace.nau.edu.ua/bitstream/NAU/47336/1/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D0%BA%D0%BB%D0%B0%D0%B4_%D1%83_%D1%81%D1%83%D1%87%D0%B0%D1%81%D0%BD%D0%BE%D0%BC%D1%83_%D1%81%D1%83%D1%81%D0%BF%D1%96%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D1%96.PDF
11. Один з підходів до розробки системи автоматичного озвучення текстів українською мовою [Електронний ресурс] / Ю.В. Крак, В.В. Горбань // 2021 -

Режим доступу: http://www.iai.dn.ua/public/JournalAI_2004_1/Razdel2/06_Krak_Gorban'.pdf]

12. Розробка алгоритмів і програм синтезу якісного мовлення за правилами / О.М. Карпов, К. Г. Чебров // 2012 [Електронний ресурс] – Режим доступу: https://actualproblems.dp.ua/index.php/APAIT/article/download/31/31

13. IconnectFx [Електронний ресурс] - Режим доступу: https://www.iconnectfx.com

14. Dotnet Web Forms [Електронний ресурс] - Режим доступу: https://learn.microsoft.com/ru-ru/aspnet/web-forms/what-is-web-forms

15. Онлайн-трансляція - Режим доступу: https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%86%D0%B8%D1%8F

16. Онлайн-конференція [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D1%81%D0%B5%D0%BC%D1%96%D0%BD%D0%B0%D1%80

17. Gaussian mixture model [Електронний ресурс] - Режим доступу: https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

18. Dynamic time warping [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Dynamic_time_warping

19. Hidden Markov model [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Hidden_Markov_model

20. Recurent neural networks [Електронний ресурс] - Режим доступу: https://www.ibm.com/cloud/learn/recurrent-neural-networks

21. Convolutional neural network [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Convolutional_neural_network

22. Нейронні мережі типу трансформери [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

23. Mel-frequency cepstrum [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

24. Discrete wavelet transform [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Discrete_wavelet_transform

25. Mel scale [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Mel_scale

26. Wavelet [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Wavelet

27. Mel frequency cepstral coefficients [Електронний ресурс] - Режим доступу: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

28. Windowing/Digital signal processing [Електронний ресурс] - Режим доступу: https://en.wikibooks.org/wiki/Digital_Signal_Processing/Windowing

29. Discrete cosine transform [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Discrete_cosine_transform

30. Моделі seq2seq [Електронний ресурс] - Режим доступу: https://habr.com/ru/company/otus/blog/430780/

31. Speech-transformer source code [Електронний ресурс] – Режим доступу: https://github.com/sooftware/speech-transformer

32. WSJ language data set [Електронний ресурс] – Режим доступу: https://catalog.ldc.upenn.edu/LDC93s6a

33. WSJ [Електронний ресурс] - Режим доступу: https://www.wsj.com/

34. Understanding the mel spectogram [Електронний ресурс] - Режим доступу: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

35. Dotnet Framework [Електронний ресурс] - Режим доступу: https://dotnet.microsoft.com/en-us/

36. Windows OS [Електронний ресурс] - Режим доступу: https://ru.wikipedia.org/wiki/Windows

37. Linux OS [Електронний ресурс] - Режим доступу: https://ru.wikipedia.org/wiki/Linux

38. Mac OS [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/Mac_OS

39. API [Електронний ресурс] - Режим доступу: https://ru.wikipedia.org/wiki/API

40. JSON [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/JSON

41. React Framework [Електронний ресурс] - Режим доступу: https://uk.reactjs.org/

42. FFmpeg [Електронний ресурс] - Режим доступу: https://ffmpeg.org/ –

43. Що таке Dotnet [Електронний ресурс] - Режим доступу: https://training.epam.ua/News/Items/301?lang=ua

44. Python [Електронний ресурс] - Режим доступу: https://www.python.org/ –

45. Azure [Електронний ресурс] - Режим доступу: https://azure.microsoft.com/ru-ru/

46. Патерни проєктування / Ерік Фрімен, Елізабет Робсон – Фабула Про, 2020. – 291с.

47. Nginx [Електронний ресурс] - Режим доступу: https://nginx.org/ru/

48. Dotnet SignalR [Електронний ресурс] - Режим доступу: https://learn.microsoft.com/ru-ru/aspnet/signalr/overview/getting-started/introduction-to-signalr

49. RabbitMQ [Електронний ресурс] - Режим доступу: https://www.rabbitmq.com/

50. Kubernetes [Електронний ресурс] - Режим доступу: https://kubernetes.io/

51. Коннолли Т. Базы данных: проектирование, реализация и сопровождение / Т. Коннолли, К. Бегг, А. Страчан – М.:Вильямс, 2000. – 1120 с.

52. Бьюли А. Изучаем SQL / Алан Бьюли. - Символ-Плюс, 2007. - 312 с

53. Ицик Бен-Ган - Microsoft SQL Server 2008. Основы T-SQL. БХВ-Петербург. 2009. 430с.

54. . Бандура І.О. Підходи до розпізнавання мови при трансформації відеоматеріалів на мову користувача / І.О. Бандура, І.В. Турченко // Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: міжнар. наук.-техн. конф, 08-09 грудня 2022 р.: збірник тез доповідей: випуск 73, 2022

[Електронний ресурс] – Режим доступу: http://www.konferenciaonline.org.ua/ua/article/id-822/

55. Бандура І.О. Алгоритм трансформації відеоматеріалів на мову користувача / І.О. Бандура // Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: міжнар. наук.-техн. конф, 08-09 грудня 2022 р.: збірник тез доповідей: випуск 73, 2022 [Електронний ресурс] – Режим доступу: http://www.konferenciaonline.org.ua/ua/article/id-823/

56. Загальні рекомендації з підготовки, оформлення, захисту та оцінювання випускних кваліфікаційних робіт здобувачів вищої освіти першого «бакалаврського» і другого «магістерського» рівнів / За ред. доц. М.І. Шинкарика. Тернопіль: ТНЕУ, 2018. 67 с

57. Комар М.П., Саченко А.О., Васильків Н.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за другим (магістерським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2021. 32 с