Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка інтелектуального застосунку психологічної допомоги з тестуванням
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІ-21-4

Артем Голубєв
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект

(повна назва освітньої програми)
Керівник ас. Микола Черненко
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Голубєву Артему Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтелектуального застосунку психологічної допомоги з тестуванням

затверджена наказом університету від 19 травня 20 25 р. № 378Ст
2. Термін подання студентом роботи до екзаменаційної комісії 20 червня 20 25 р.
3. Вихідні дані до роботи документація Java, документація Python, документація React, наукові дослідження, соціальні опитування, підручники зі штучного інтелекту, документація Kaggle, набори даних

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі
2) Проектування застосунку та реалізація серверної частини
3) Розробка моделі штучного інтелекту
4) Розробка клієнтської частини

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Аналіз предметної галузі	21.05.2025	виконано
3	Постановка задачі	22.05.2025	виконано
4	Проектування застосунку	23.05.2025	виконано
5	Реалізація серверної частини	25.05.2025	виконано
6	Розробка моделі штучного інтелекту	27.05.2025	виконано
7	Розробка клієнтської частини	30.05.2025	виконано
8	Написання пояснювальної записки	09.06.2025	виконано
9	Перевірка на академічний плагіат	10.06.2025	виконано
10	Проходження нормоконтролю	11.06.2025	виконано
11	Підготовка презентації	12.06.2025	виконано
12	Попередній захист	13.06.2025	виконано
13	Рецензування	14.06.2025	виконано
14	Захист перед ЕК	20.06.2025	

Дата видачі завдання 19 травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Микола Черненко
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 85 с., 46 рис., 1 табл., 1 дод., 35 джерел.

АНАЛІЗ ЕМОЦІЙ, НЕЙРОННІ МЕРЕЖІ, ПСИХОЛОГІЧНА ДОПОМОГА, CNN, JAVA, MONGODB, POSTGRESQL, SPEECH EMOTION RECOGNITION, TRANSFORMERS.

Об’єкт дослідження – аналіз психологічного стану людини шляхом розпізнавання емоційного забарвлення в голосі.

Предмет дослідження – використання нейронних мереж для розпізнавання емоційного забарвлення в записі голосу людини, що дозволяє аналізувати її психологічний стан.

Мета роботи – розробка інтелектуального застосунку для отримання психологічної допомоги шляхом консультацій та тестування психологічного стану на основі комплексного тесту з використанням нейронної мережі.

Методи дослідження – теоретичний (збір та структуризація наявних досліджень в галузі), експериментальний (програмна реалізація застосунку та нейронної мережі, її навчання).

У результаті роботи було досліджено актуальність теми роботи, особливо в умовах війни в Україні, проведено аналіз обраної предметної галузі, аналіз та порівняння наявних рішень, що впроваджені у цій сфері, існуючі набори даних з записом голосу, натренована та протестована нейронна мережа з різними значеннями гіперпараметрів для досягнення найоптимальнішого можливого результату. Був розроблений веб-застосунок який складається з бекенд частини на Java та фронтенд частини на React.JS. Нейронна мережа була впроваджена до розробленого веб-застосунку як одна з частин компоненти інтелектуального тестування.

ABSTRACT

Bachelor’s thesis contains: 85 pp., 46 fig., 1 tabl., 1 ann., 35 references.

CNN, EMOTION ANALYSIS, EMOTION RECOGNITION, JAVA, MONGODB, NEURAL NETWORKS, POSTGRESQL, PSYCHOLOGICAL HELP, SPEECH, TRANSFORMERS.

An object of the research is an analysis of a person’s psychological state using emotions recognition in the voice.

A subject of the research is a usage of neural networks for emotions recognition in a person's voice recording, enabling analysis of their psychological condition.

A purpose of the work is development of an intelligent application for providing psychological assistance via consultations and psychological state testing. The application performs comprehensive tests using a neural network.

Research methods include theoretical and experimental. Theoretical methods consist of collecting and structuring existing studies in the area. Experimental methods consist of designing and implementing the application, the neural network training, and usage.

As a result of the work, the relevance of the topic was explored, especially in the context of the war in Ukraine. A defined domain area analysis has been conducted, along with a comparison of existing implemented solutions and available datasets containing voice recordings. A neural network has been trained and tested with various hyperparameter values to achieve the most optimal possible result. A web application is designed and developed. It consists of a backend component (built using Java) and a frontend component (developed with React.JS). The neural network has been integrated into the web application as a dedicated software component for the intelligent testing.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект – поняття для позначення систем створених людиною, що здатні виконувати завдання, які зазвичай потребують людського інтелекту;

CNN – Convolutional Neural Network – клас нейронних мереж, що використовується здебільшого для аналізу візуальних зображень;

LLM – Large Language Model – модель загального призначення яка натренована на нерозміченому тексті, щоб справлятися з великим спектром задач, замість того щоб бути сфокусованою на одній конкретній задачі чи класу задач;

LSTM – Long Short-Term Memory – різновид архітектури рекурентних нейронних мереж, яка має можливість запам’ятовувати значення як на короткі, так і на довгі проміжки часу;

RNN – Recurrent Neural Networks – клас штучних нейронних мереж, що має внутрішній стан, який дозволяє послідовно оброблювати довільну послідовність входів;

SER – Speech Emotion Recognition – модель нейронної мережі, яка призначена розпізнавання емоції з аудіозапису;

Transformer – архітектура глибокого навчання, що використовує механізм «уваги», що допомагає зберегти контекст та обробляти вхідну послідовність паралельно.

ВСТУП

В 21 сторіччі, в епоху технологій та наукових досягнень, такі розлади проблеми з ментальним здоров'ям почали траплятися все частіше [1], [2].

Постійний стрес, понаднормовий робочій день, катастрофи та надзвичайні ситуації, людському мозку дуже важко сприймати та оброблювати такий об'єм інформації та стресових подій майже без перерви. До цього додається постійне життя в умовах війни та загроз, які вона несе. В таких обставинах людям потрібно отримувати якісну психологічну підтримку якомога скоріше, задля запобігання погіршенню ментального стану, а як наслідок погіршенню концентрації, уважності, працездатності тощо. В деяких випадках люди відмовляються від обстежень, прийомів та консультацій у лікарів відповідного профілю через стереотипи, страх, стрес, інші причини [3].

Визначена проблема має сучасні рішення: в наш час технології штучного інтелекту розвиваються швидше ніж будь коли. Вони проникають в усі сфери нашого життя та їх вплив на нас стає все більшим з кожним днем. Не виключенням стала і психологічна допомога та споріднені сфери. Інтелектуальні агенти починають ставати важливим компонентом в медицині, зокрема в поведінковій психології, біопсихології, клінічній психології тощо [4]. Велику популярність набирають чат-боти на базі штучного інтелекту, з якими людина зможе поспілкуватись. Проте в великих мовних моделях (Large Language Models – LLM) ще досі можуть знаходитись недоліки, галюцинації тощо, а в такій чутливій сфері як психологічна підтримка це не допускається. Це не єдиний з перспективних напрямів досліджень використання моделей штучного інтелекту у сфері ментального здоров'я. В останні роки починає набирати популярність окремий напрям досліджень – Speech Emotion Recognition (SER), що досліджує розпізнавання емоцій в голосі. Цей підхід до аналізу настрою має великий потенціал для застосування в сфері психологічної підтримки, а саме

в психологічному тестуванні. Необхідність в розробці нових програмних рішень для надання такої інноваційної допомоги особливо посилюється в умовах війни, через те, що багато верств населення та категорій громадян за тих чи інших обставин не можуть отримати якісну та своєчасну психологічну допомогу. Пацієнти з сіл та маленьких міст не завжди мають у своєму населеному пункті лікарню яка надає послуги такого напряму, а навіть якщо вони і є, їх якість не завжди може бути такою, на яку розраховує пацієнт. Маломобільним верствам населення також може бути незручно добиратися до лікарні, тому що не усі міста наразі інклюзивні.

Окрема увага може бути приділена особам, що тимчасово виїхали за кордон у різні країни світу, проте багато з них не вивчили мову країни в якій вони перебувають, достатньою мірою для того щоб звернутися до місцевого спеціаліста. Багато людей мають психічні порушення, пов’язані з війною, «провину вцілілого», труднощі з адаптацією в новому соціумі. Тому важливо щоб українці в усьому світі мали можливість звернутися за допомогою до українського спеціаліста, бо він буде більше в контексті подій, які їх турбують. В даній роботі основна фокус-група – громадяни України (через обставини, що викладено раніше), в цей же час, архітектура застосунку враховує можливості масштабування рішення для використання іншими мовними групами та врахування альтернативних аспектів. Значну роль в такій допомозі грає діагностування проблеми. Для цього використовуються різні методи – психологічні вправи, тестування тощо. Проте тестування в його звичному вигляді (тести з варіантами відповідей) не завжди можна назвати об’єктивним. В стресовій ситуації людина схильна применшувати або перебільшувати свої проблеми. Саме тут штучний інтелект може прийти на допомогу, з ним людина може комунікувати відвертіше, не боячись сторонніх оцінок, невпевненості або осуду. Окрім того вхідне тестування робиться безкоштовним, таким, що не потребує консультації з лікарем. За допомогою запису голосу людини нейронна

мережа має змогу розпізнати емоції в голосі, що зменшує вірогідність того, що людина пройде тестування необ’єктивно навіть не усвідомлюючи цього.

Тому ціллю цієї роботи є розробка інтелектуального застосунку для психологічної допомоги, який буде поєднувати в собі функції комбінованого тестування психологічного та емоційного стану за допомогою поєднання традиційних тестів та штучного інтелекту, та надання онлайн консультацій дипломованими лікарями за допомогою телеметрії, призначення рекомендацій щодо лікування, тощо. Це включає в себе проектування архітектури застосунку та його розробку, візуального інтерфейсу, тренування моделі та впровадження її до веб-застосунку.

Окрему увагу слід приділити конфіденційності інформації – жодна персоналізована інформація не може бути збережена на сервері. У випадку збереження будь якої інформації вона має бути деперсоналізована та користувач має бути проінформований та дати свою згоду на це.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Задача розпізнавання емоцій людини зацікавила вчених ще в 70-х роках минулого століття. Одним з найперших вчених який зацікавився цією темою був Пол Екман, який в 1972 році опублікував дослідження [5], в якому вперше намагався класифікувати емоції людини за виразом обличчя. В 2016 році в аналізі [6] вчені стверджують, що найбільших успіхів можна досягти якщо аналізувати мовлення людини, а не її емоції, проте це дослідження дало початок більш широким дослідженням.

Дослідження щодо розпізнавання емоцій продовжилися та в 90-з роках минулого сторіччя вченими з MIT був розроблений [7] робот, який за допомогою камер та мікрофонів міг визначати такі емоції як схвалення, заборона, уважність, комфорт та нейтральність.

Наразі найновішим способом визначення емоцій людини є методи штучного інтелекту, від традиційних машин опорних векторів (Support Vector Machines – SVM), до більш провідних методів глибоких нейронних мереж [8].

Виходячи з наведених досліджень можна стверджувати, що напрямок розпізнавання емоцій за голосом є досить популярним для досліджень та може використовуватись в сфері психологічної допомоги.

На тему отримання психологічної допомоги громадянами України особливо під час війни проводилися та проводяться різноманітні дослідження, розглянувши результати яких стало зрозуміло, що проблему доступності та нормалізації отримання психологічної допомоги треба вирішувати. У дослідженні центру «Коло сім’ї» [9] зокрема зазначається, що у світі приблизно 80% дітей та молоді з розладами у сфері психологічного здоров’я не отримують належної допомоги через вплив різних факторів, частина результатів дослідження представлена на рисунку 1.1.

Рисунок 1.1 – Отримання психологічної допомоги молоддю

Проте молодь не є єдиною категорією людей, що стикається з проблемою недоотримання психологічної допомоги. У результаті дослідження [10], проведеного компанією 4Service у межах Всеукраїнської програми ментального здоров’я виявляється, що тільки 17% людей на початок 2025 року звертаються до психолога за потреби, не зважаючи на те, що 71% респондентів відчуває необхідність у психологічній допомозі. Ця разюча розбіжність свідчить про наявність серйозних бар'єрів – соціальних, фінансових чи особистісних, – які залишають переважну більшість людей без фахової підтримки. З того ж дослідження видно, що тільки 8% людей визначають свій стан за результатами тестів, коли 93% керуються лише власними почуттями, що показано на рисунку 1.2.

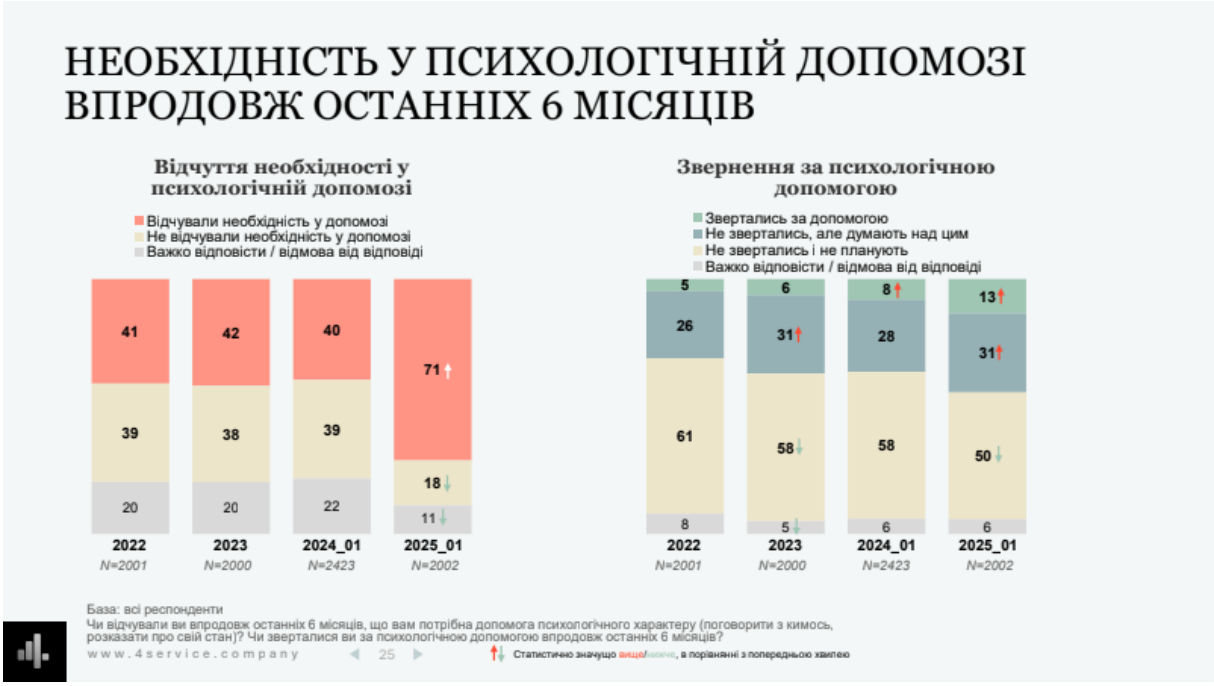


Рисунок 1.2 – Потреба українців в отриманні психологічної допомоги

З проведеного спеціалістами American University Kyiv і дослідницької лабораторії Rating Lab дослідження [11] йдеться, що найкраще покращити психоемоційний стан людям допомагає розмова з друзями або прогулянками. Проте звернення до лікаря, судячи з усього, потрапляє в категорію «Інше» що становить усього 10% відсотків від результату, що видно з рисунку 1.3.

Рисунок 1.3 – Способи боротьби зі стресом

Виходячи з наведених досліджень та беручи до уваги, що на даний момент велика кількість українців всередині країни та закордоном за тих чи інших обставин не хочуть або не можуть отримати своєчасну та якісну психологічну допомогу варто запропонувати інноваційне онлайн рішення, яке зможе розширити доступ до психологічної допомоги для осіб з різних вікових та соціальних груп, які вагаються, обмежені фізично чи фінансово.

В цій роботі виконується розробка інтелектуального застосунку психологічної допомоги з тестуванням за допомогою моделі штучного інтелекту, а саме нейронних мереж. За останні роки розвиток штучного інтелекту прискорився суттєво [12], тому це тільки питання часу, коли він займе своє місце в питанні психологічної діагностики пацієнтів та допомоги лікарям у лікуванні. Вже існують потужні LLM на основі аналізу великих масивів даних, які можуть надавати допомогу у вигляді звичайного спілкування з людиною. На даний час ці системи дуже дороговартісні, використовують велику кількість електроенергії [13] та не мають вузького призначення саме для психологічної допомоги. Окрім того використання таких систем користувачем без контролю з боку лікаря або спеціалізованої системи може призводити або до невірних результатів, або до неправильної інтерпретації людиною та вчиненню невірних кроків.

Наявні застосунки на ринку, як буде досліджено нижче, не виконують свої функції повною мірою, особливо в контексті війни в Україні, для українських користувачів. Ті застосунки, які на даний час присутні на українському ринку не мають в собі інноваційних технологій з використанням штучного інтелекту, що, зокрема, може знизити навантаження на лікарів, проводячи початкове тестування, надаючи пацієнту рекомендацію щодо необхідності звернутися до лікаря відповідного профілю, або зробити якісь кроки для покращення ментального здоров’я та самопочуття.

Інтелектуальний застосунок для надання психологічної допомоги для більш ефективного та зрозумілого використання користувачем має

виконувати певні функції. Застосунок такого роду не має бути занадто орієнтований на людей, які мають досвід в медицині. В наданні психологічної допомоги широким верствам населення потрібно брати до уваги, що не всі з них раніше зверталися до лікарів такого профілю, можуть не мати досвіду та розуміння, що саме від них очікується. З метою досягнення такого ефекту потрібно уникати складних медичних термінів, які можуть бути неприйнятні для користувача без досвіду. В процесі впровадження інтелектуального тестування потрібно дотримуватися повсякденних формулювань, висвітлювати проблеми та ситуації які будуть зрозумілі пересічній людині.

Окрему увагу в задачах такого напрямку слід приділяти конфіденційності та точності тестування. По-перше, тестування має носити рекомендаційний характер та не надавати рекомендацій що можуть нашкодити. Також слід зауважити, що зберігання та використання аудіозаписів користувачів в інших цілях без їхньої прямої згоди не допускається. Моделі розпізнавання емоції (Speech Emotion Recognition) розвиваються досить швидко, проте їх точність досі не наблизилась до точності моделей, які вирішують інші задачі, на наборі даних зі стандартизованими фразами, що промовляють професійні актори зі студійною якістю вони набирають приблизно 65 – 80% [14]. Тому досі доцільно комбінувати підходи, тобто до інноваційного тестування за допомогою аудіозапису включати ще й традиційне опитування на основі тесту, для того, щоб в подальшому комбінувати відповіді з обох джерел для отримання більш достовірного та точного результату.

1.2 Аналіз існуючих застосунків

BetterHelp – американська платформа для допомоги з психічним здоров’ям, яка почала свою роботу в 2013 році, має відгуки вище середнього та користується попитом. Після аналізу цього застосунку було виявлено, що

платформа використовує тест на основі запитань, які включають в себе питання про країну перебування, стать, релігію, можливі проблеми, працевлаштування тощо. Проте ця платформа більш орієнтована на англомовну аудиторію та не дуже підходить для українців в контексті війни, окрім того, платформа не дає можливості ознайомитись зі своїм функціоналом, списком лікарів, послугами які вони надають, без оплати першого тижня, що видно з рисунку 1.4.

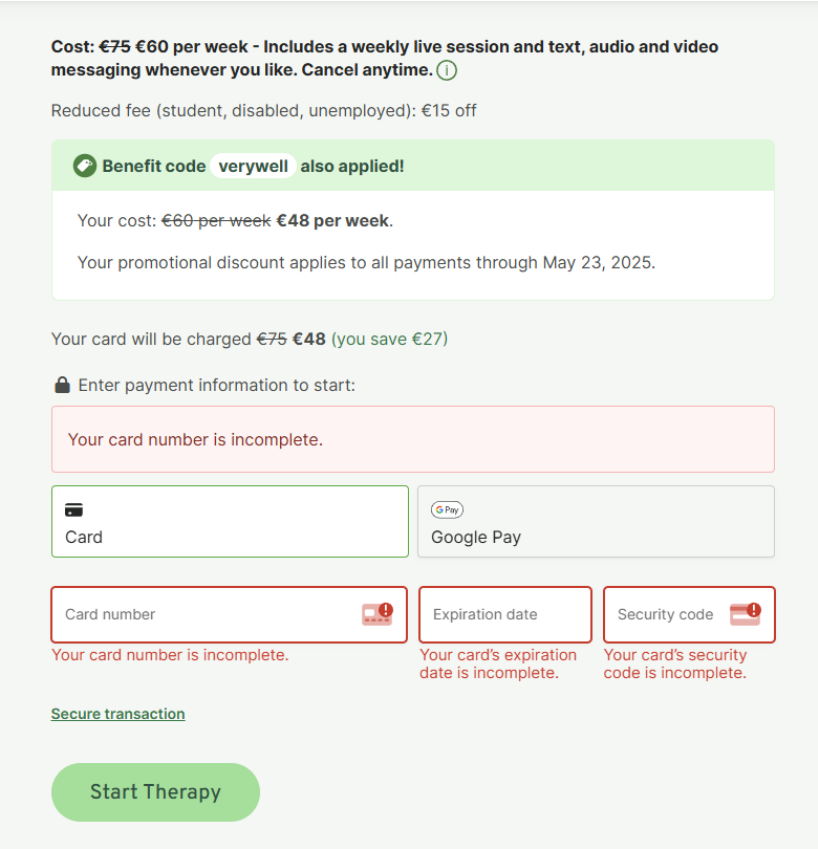


Рисунок 1.4 – Сторінка оплати BetterHelp

Цей сервіс надає можливість пройти психологічний тест, який складається з запитань, проте без оплати неможливо як дізнатись його результати, так і ознайомитись з функціоналом застосунку на власні очі.

Платформа «Розкажи мені» це українська онлайн платформа для отримання безкоштовних психологічних консультацій в короткий термін.

На цій платформі зв’язок з лікарем здійснюється після заповнення форми через будь який месенджер або по телефону. Консультації здійснюються одноразово, ніде не записуються, платформа не надає можливостей для тестування, що продемонстровано на рисунку 1.5.

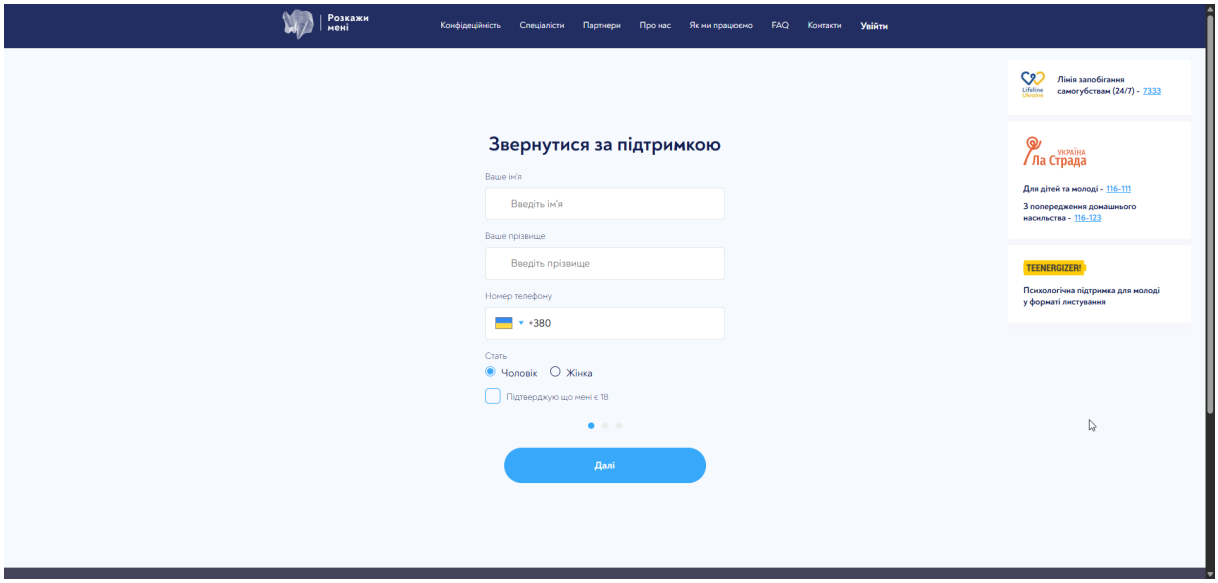


Рисунок 1.5 – Інтерфейс для запису платформи «Розкажи мені»

Платформа психологічної допомоги «Rozmova» є одним з провідних українських застосунків, що значно спрощує доступ до психотерапевтичної допомоги. Вона надає користувачам широкий спектр можливостей для вибору психотерапевта, дозволяючи фільтрувати спеціалістів за такими важливими критеріями, як напрям допомоги, мова спілкування, стать та інші. Це значно полегшує пошук фахівця, який максимально відповідає індивідуальним потребам та вподобанням користувача.

Що відрізняє «Rozmova» від багатьох інших платформ, так це можливість детально ознайомитися з профілями лікарів, що дає користувачам додаткову впевненість у своєму виборі. Крім того, платформа пропонує пройти декілька психологічних тестів, наприклад, тест для визначення оптимального способу терапії, а також інші, як показано на рисунку 1.6. Однак, варто зазначити, що ці тести включають в себе лише

традиційні підходи. Це може бути їхнім обмеженням, оскільки результати можуть бути необ’єктивними через нестабільний психологічний стан користувача або інші порушення, що можуть перешкоджати адекватному проходженню традиційного тестування.

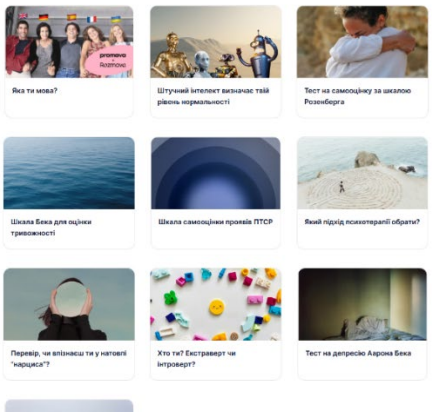


Рисунок 1.6 – Психологічні тести на платформі «Rozmova»

Останнім застосунком, на який варто звернути увагу, є стартап «Hedepy», що заснований для пошуку психологічної допомоги. Він надає можливість вибору психотерапевта як вручну, так і обирає конкретного спеціаліста за результатами тесту, що спрощує вибір та дає можливість приступити до терапії зі спеціалістом, що підходить найбільше (рисунок 1.7).

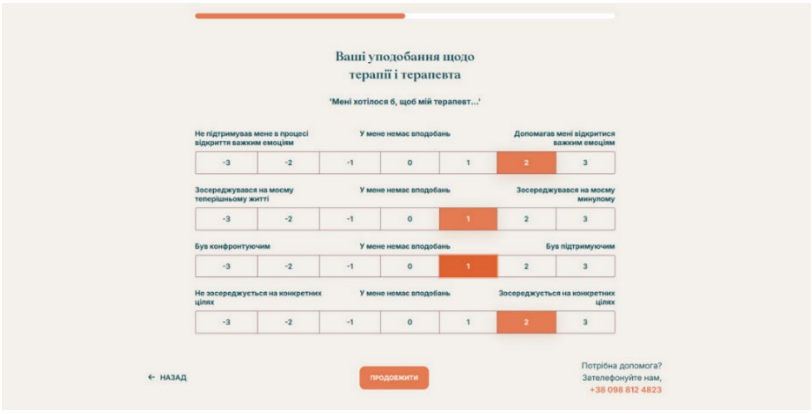


Рисунок 1.7 – Частина психологічного тесту платформи «Hedepy»

Після аналізу наявних на ринку рішень очевидно, що на даний момент немає повноцінних рішень для отримання психологічної допомоги з використанням інноваційного тестування, зокрема з використанням штучного інтелекту. Застосунки з наданням онлайн допомоги існують, проте вони надають лише традиційний формат тестування у формі питань, що може бути незручним, неприйнятим або необ'єктивним для деяких категорій користувачів, а ті застосунки, які використовують інноваційні аспекти тестування, на даний час здебільшого розгорнуті на англомовний ринок, що є неприйнятним для більшості користувачів з України, через мовний бар'єр, нерозуміння іноземними спеціалістами контексту війни в Україні, соціальних потрясінь, повсякденної специфіки тощо.

1.3 Аналіз необхідних технологічних рішень

Для того, щоб натренувати модель штучного інтелекту потрібно перетворити аудіозаписи в числові або інші ознаки. Для цього існує низка різних методів, які будуть описані далі. Ознаки з аудіосигналу розподіляються на три типи – ручні ознаки (handcrafted features), векторні ознаки та глибокі ознаки. Огляд та порівняння кожної з категорій наведені нижче, з метою обрати найоптимальніший варіант.

Першою з категорій є ручні ознаки – ті, які розроблялись вченими та фахівцями з обробки даних. Серед них найпоширенішими є спектральні ознаки, кількість переходів через нуль та середньоквадратична енергія (Root mean square energy, RMSE).

Спектральні ознаки включають в себе значну кількість ознак, деякими з яких є кепстральні коефіцієнти, що працюють подібно до слуху людини, як він сприймає частотні діапазони, або спектральний центроїд, що являє собою «центр мас» спектру та вказує на середню частоту сигналу мови, загалом він дає уявлення щодо висоти та тембру мовлення, його приклад представлений на рисунку 1.8.

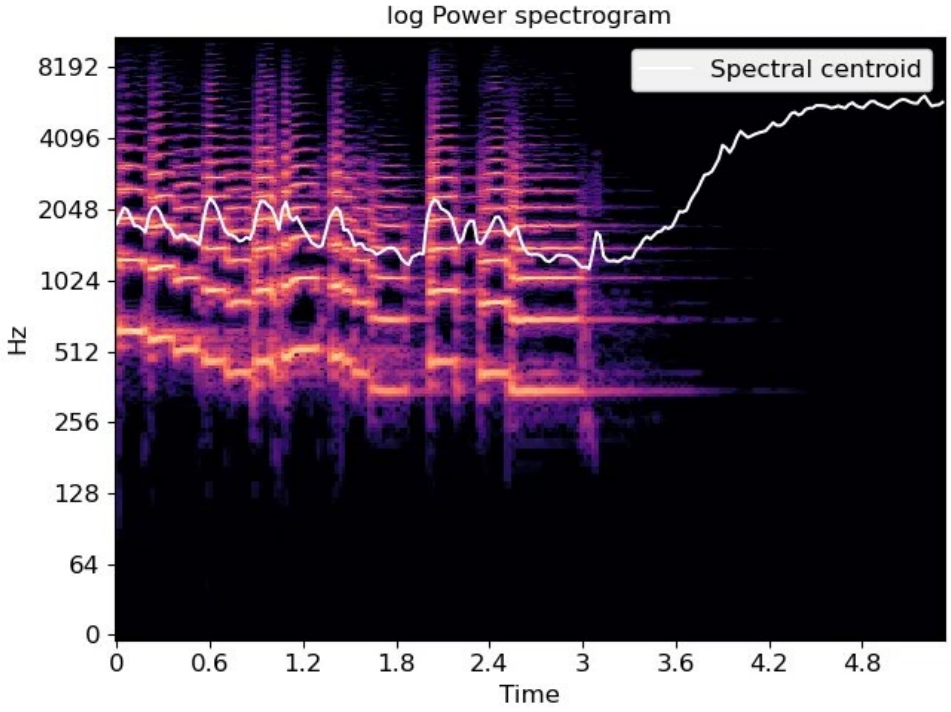


Рисунок 1.8 – Приклад спектрального центроїду [15]

Окрім середньої частоти також важливо виміряти асиметрію у розподілі частот, для того, щоб дізнатись, чи переважають у запису вищі або нижчі частоти. Ця міра вказує на асиметрію спектра навколо спектрального центроїду, щоб побачити по який бік від центроїда знаходиться основна частина спектру. Асиметрія приймає як позитивні так і негативні значення, але якщо вона нульова то асиметрії немає та енергія в спектрі розподілена рівномірно. Це лише одні з найрозповсюдженіших спектральних ознак, які використовуються для аналізу звуку. Кількість переходу через нуль дозволяє виміряти плавність сигналу шляхом підрахунку кількості переходів сигналу через вісь X. Наприклад, наголошені звуки більш плавні ніж ненаголошені, а сигнал частотою 100Гц може перетинати нуль 100 разів за секунду, у порівнянні з глухим фрикативним, що може перетинати нуль 3000 разів на секунду [16]. Наочний приклад такого виміру продемонстрований на рисунку 1.9.

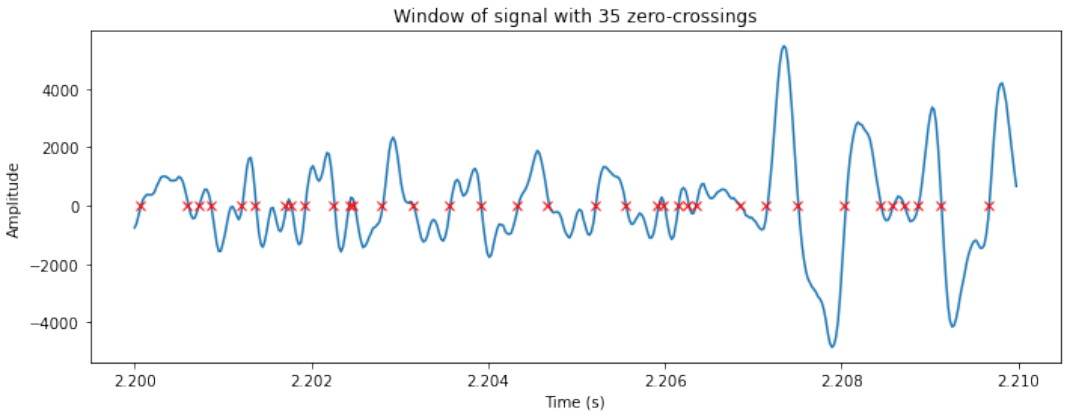


Рисунок 1.9 – Розрахування кількості переходу через нуль [16]

Останньою ознакою з першої категорії є середньоквадратична енергія сигналу, що певною мірою позначає його гучність, та корисна також корисна для аналізу, зокрема емоційного. Відповідність значення цієї ознаки до справжньої гучності аудіо наведено на рисунку 1.10.

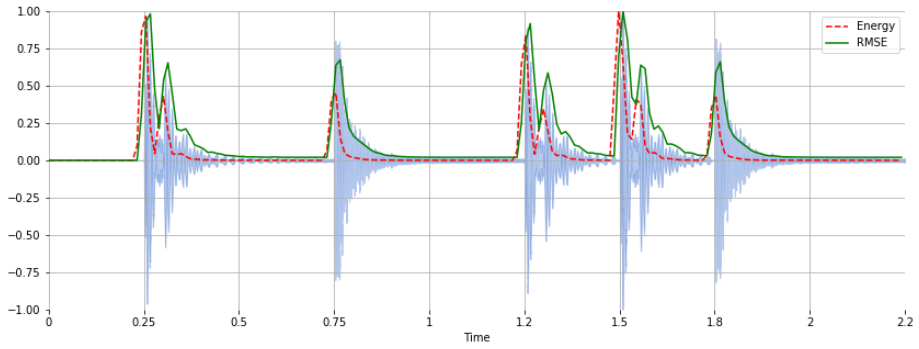


Рисунок 1.10 – Середньоквадратична енергія сигналу [17]

Ці ознаки прості, зрозумілі для людини інтуїтивно, проте через те, що вони найпростіші, існують кращі способи отримати дані з аудіозапису, чим і є категорія класичних векторних ознак, описана нижче.

Векторні ознаки це спосіб описати будь які дані у формі числового вектору, для їх отримання існує низка підходів, наприклад Bag-of-Audio-Words, або OpenSMILE.

Bag-of-Audio-Words це метод, що використовує штучний інтелект для відокремлення ознак з аудіофайлу таким чином – спочатку до аудіозапису застосовується перетворення Фур'є, що дозволяє розбити один запис на декілька частин, кожна з яких репрезентує окрему частоту, таким чином стає можливим отримання даних про зміну частоти з плином часу, та на основі цього побудувати спектрограму, таку як наведено на рисунку 3.1, чим яскравіше окрема частота в кожен момент часу, тим більше вона представлена на кожному «кадрі» аудіо. Після отримання амплітуд кожної частоти за допомогою, наприклад, дискретних косинусних перетворень, зменшується розмірність отриманого вектору ознак. Після цього модель кластеризації розподіляє ці вектори за N кластерами, які обираються експериментально, після цього для вектору ознак кожного кадру вираховується найближчий центр кластеру [18]. Діаграма використання цього методу наведена на рисунку 1.11.

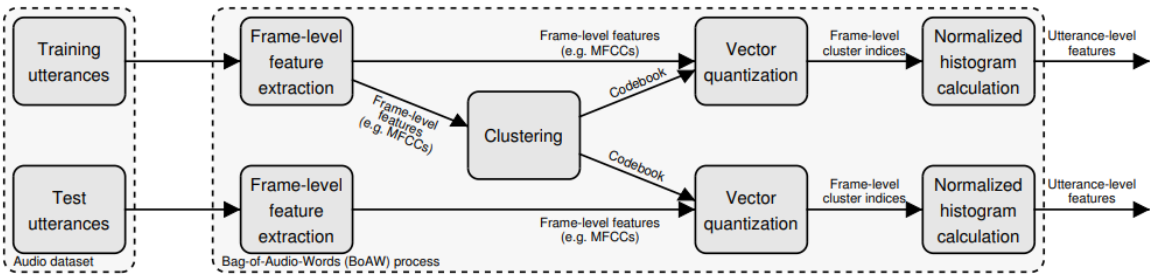


Рисунок 1.11 – Схема використання Bag-of-Audio-Words [18]

В результаті роботи цього методу для кожного аудіозапису отримується N мірна гістограма, яка вказує, скільки кадрів із запису потрапили в певний кластер, що показує певну характеристику мовлення.

Приклад гістограми наведений на рисунку 1.12, вона перетворюється на вектор фіксованої довжини, який являє собою ознаки аудіозапису для подальшої класифікації.

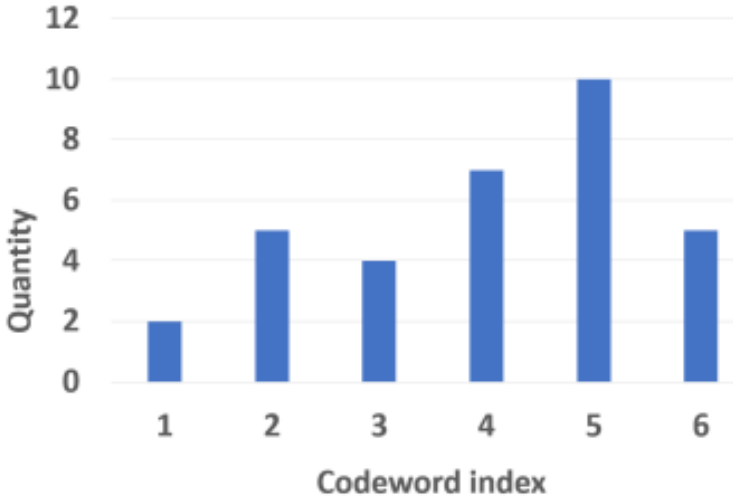


Рисунок 1.12 – Гістограма для окремого аудіозапису [18]

OpenSMILE це екстрактор ознак, розроблений вченими для виокремлення ознак з аудіозаписів різного формату, обробкою даних перед вилученням ознак з використанням різних технологій, таких як вилучення ручних ознак, які були описані вище, енергії кадру, інтенсивності, якості голосу, його тремтіння, мелчастотних кепстральних коефіцієнтів, які також використовуються у вищезазначеному Bag-of-Audio-Words. Загалом цей екстрактор містить в собі різноманітні функції для виокремлення традиційних ознак, має вбудовані традиційні моделі штучного інтелекту, такі як SVM тощо [19].

Останньою категорією ознак є глибокі ознаки, які видобуваються з аудіо шляхом використання глибоких нейронних мереж, зокрема це CNN на спектрограмах, LSTM на MFCC/спектрограмах або заздалегідь натреновані моделі, такі як Wav2Vec2.0.

Згорткова нейронна мережа (CNN) – вид нейронних мереж, який широко використовується для аналізу зображень. Ця нейронна мережа до прогнозування фінального класу за допомогою фільтрів, функцій активації та шарів об’єднання перетворює числові значення пікселів зображення на вектор ознак для зображення, який можна використовувати далі. У якості зображень, які подаються на вхід до моделі, використовуються

спектрограми, процес побудови яких описано раніше. Зображення, які перетворюються у матрицю числових значень, передаються до згорткового шару, який застосовує фільтри до отриманої матриці задля пошуку ознак, що показано на рисунку 1.13.

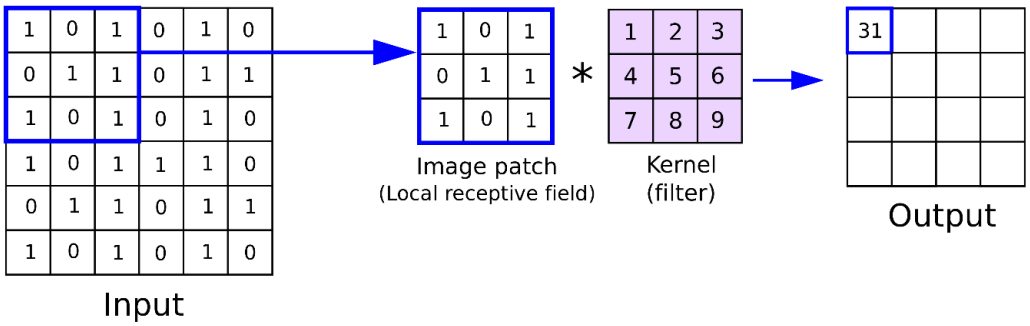


Рисунок 1.13 – Застосування згорткового шару [20]

Після цього за допомогою функцій активації (найчастіше ReLU), змінюють результуючу матрицю задля пошуку нелінійних зв’язків. Потім за допомогою шару об’єднання зменшують розмірність, шляхом обирання максимального (MaxPooling) або середнього значення (AvgPooling) в кожному секторі n×m матриці, задля збереження тільки найвагоміших ознак. Процес згортки та зменшення розмірності може проводитись декілька разів, після чого, результуюча матриця розгортається у вектор, що надає вектор ознак; класифікаційна голова CNN при цьому не застосовується. Таким чином, отриманий вектор інкапсулює у собі абстрактні характеристики вхідних даних, роблячи їх компактним та інформативним представленням. Даний вектор може бути використаний для подальшого аналізу іншими алгоритмами машинного навчання, наприклад, для завдань кластеризації чи класифікації. Процес повної роботи вилучення ознак за допомогою CNN наведено на рисунку 1.14.

Рисунок 1.14 – Схема feature extraction за допомогою CNN

Наступним варіантом вилучення ознак є використання моделі LSTM (Long short-term memory) який є вдосконаленням рекурентних нейронних мереж (RNN), та призначені для більш точного аналізу рядів даних. Так само як і згорткові нейронні мережі, описані раніше, LSTM можна використовувати не тільки для прогнозування, а зупинитись тільки на етапі feature extraction. Ця нейронна мережа, на відміну від згорткових нейронних мереж, не побудована саме для роботи з зображеннями, проте її архітектура, що дозволяє запам’ятовувати минулі дані та враховувати їх є важливим аспектом в контексті вилучення ознак в аудіозаписах. LSTM складається з однакових «клітин», кожна з яких приймає наступне значення вхідного потоку та значення з минулої частини. Кожна клітина LSTM складається з декількох частин, які йдуть одна за одною:

– forget gate layer – шар забування
– input gate layer – вхідний шар;
– output gate layer – вихідний шар;
– short term memory – короткострокова пам’ять;
– long term memory – довгострокова пам’ять.

Функціонування клітини довготривалої короткочасної пам’яті (LSTM) починається з шару забування (forget gate), який відповідає за визначення того, яка частина інформації з попереднього стану довготривалої пам’яті є застарілою і має бути видалена. Це рішення приймається на основі аналізу нових вхідних даних та стану короткострокової пам’яті з попереднього кроку. Ці вектори об’єднуються, множаться на відповідну матрицю ваг, після чого результат пропускається через сигмоїдну активаційну функцію, яка генерує значення від 0 до 1 для кожного елемента стану.

Наступним кроком є оновлення довготривалої пам’яті, що відбувається шляхом додавання нової релевантної інформації. Цей процес складається з двох паралельних операцій: по-перше, вхідний шар (input gate) за допомогою сигмоїдної функції визначає, які саме значення потребують оновлення. По-друге, гіперболічний тангенс (tanh) створює вектор нових значень-кандидатів, які потенційно можуть бути додані до стану клітини. Результати цих двох операцій поелементно перемножуються, що дозволяє додати до довготривалої пам'яті лише значущу частину нової інформації.

Останньою частиною роботи клітини є оновлення короткострокової пам’яті, яка також слугує вихідним сигналом для поточного часового кроку. Для цього виконуються дві операції: спочатку вихідний шар (output gate) за допомогою сигмоїдної активаційної функції фільтрує оновлене значення довготривалої пам’яті, визначаючи, яка її частина буде використана на виході. Потім це значення поелементно множиться на результат проходження стану довготривалої пам'яті через тангенціальну активаційну функцію, що формує фінальний вектор короткострокової пам’яті [21].

Детальна схема роботи однієї клітини LSTM продемонстрована на рисунку 1.15.

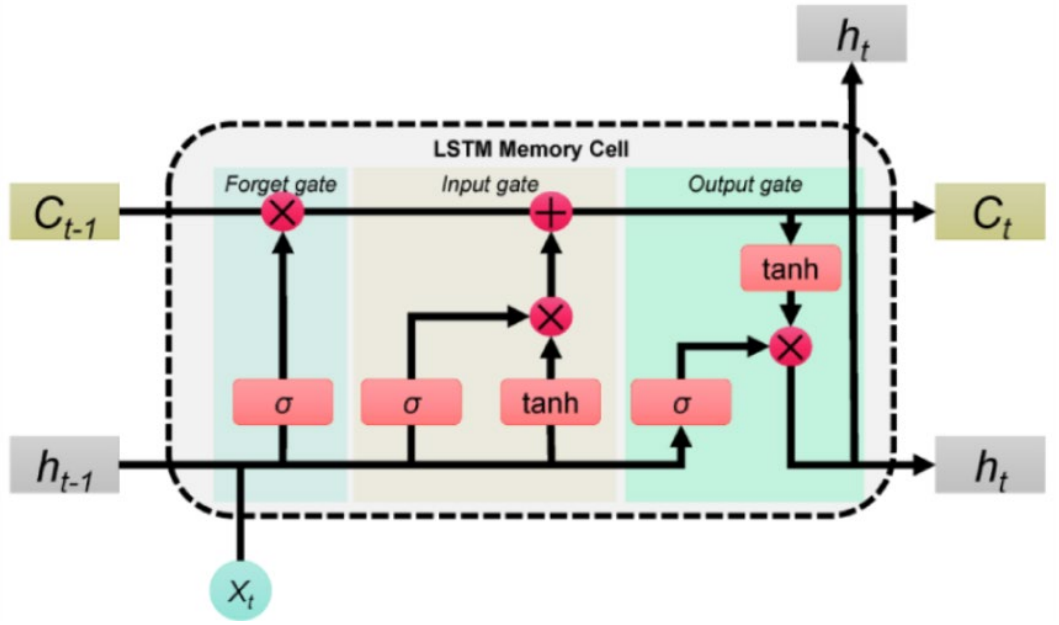


Рисунок 1.15 – Схема роботи клітини LSTM [21]

Останнім варіантом вилучення ознак є використання натренованих моделей, наприклад Wav2Vec2. Ця модель використовує трансформери, що вирішує проблеми, які залишились в LSTM, наприклад вибухання градієнту, що робить перші елементи послідовності дуже впливовими. Трансформери пропонують вирішення цієї проблеми шляхом аналізу усіх частин послідовності, що надана на вхід, одночасно. В такому випадку кожен елемент буде «знати» про інші, що дозволить зробити модель більш стійкою. Модель Wav2Vec2 складається з декількох шарів:

– latent speech representation;
– quantified representation;
– masked transformer;
– context representation.

Першим шаром є latent speech representation, в його основі лежить згорткова нейронна мережа (CNN), архітектура якої була частково описана

раніше. Цей модуль складається зі стеку із семи послідовних згорткових шарів, які обробляють вихідні аудіосигнали. Метою цього етапу є вилучення первинних векторів ознак. Ключова відмінність від звичайних згорткових мереж полягає у використанні одномірних згорток (Conv1D) замість двомірних (Conv2D), оскільки обробка відбувається вздовж часової осі аудіосигналу, а не у двовимірному просторі. Відповідно, на виході кожного шару формується вектор ознак, а не матриця.

Після того як вектори ознак вилучено, вони спрямовуються у два паралельні потоки обробки. Перший потік веде до шару векторного квантування (quantified representation). Цей процес функціонально нагадує метод кластеризації, що використовується в моделях Bag-of-Audio-Words, і слугує для дискретизації неперервних ознак. Кожен вектор ознак зіставляється з найближчим йому вектором із заздалегідь визначеної «кодової книги» перетворюючись на дискретну мітку. Ця кодова книга не є статичною та її вектори також оптимізуються та оновлюються в процесі навчання моделі.

Другий потік являє собою передачу ознак до замаскованого трансформера, що є ядром моделі. Перед подачею в трансформер застосовується стратегія маскування: певна частина векторів ознак навмисно замінюється на нульовий вектор або на випадково згенерований вектор з тієї ж кодової книги. Цей прийом є формою аугментації даних і ключовим елементом самокерованого навчання, оскільки змушує модель відновлювати оригінальний вміст на основі контексту, що буде детальніше пояснено нижче.

Далі підготовлені дані потрапляють у трансформер, процес роботи якого буде описано в наступному підрозділі. Як зазначалося раніше, архітектура трансформерів була розроблена для подолання недоліків рекурентних мереж, як-от LSTM, завдяки паралельній обробці всіх даних одночасно. Оскільки в задачах розпізнавання мовлення послідовність елементів є критично важливою, для збереження інформації про порядок

аудіоколивань до кожного вхідного вектора додається позиційне кодування. Це реалізується шляхом додавання значень, отриманих із синусоїдних функцій різних періодів, що дозволяє моделі розрізняти позиції фреймів у послідовності. Цей механізм детально продемонстровано на рисунку 1.16.

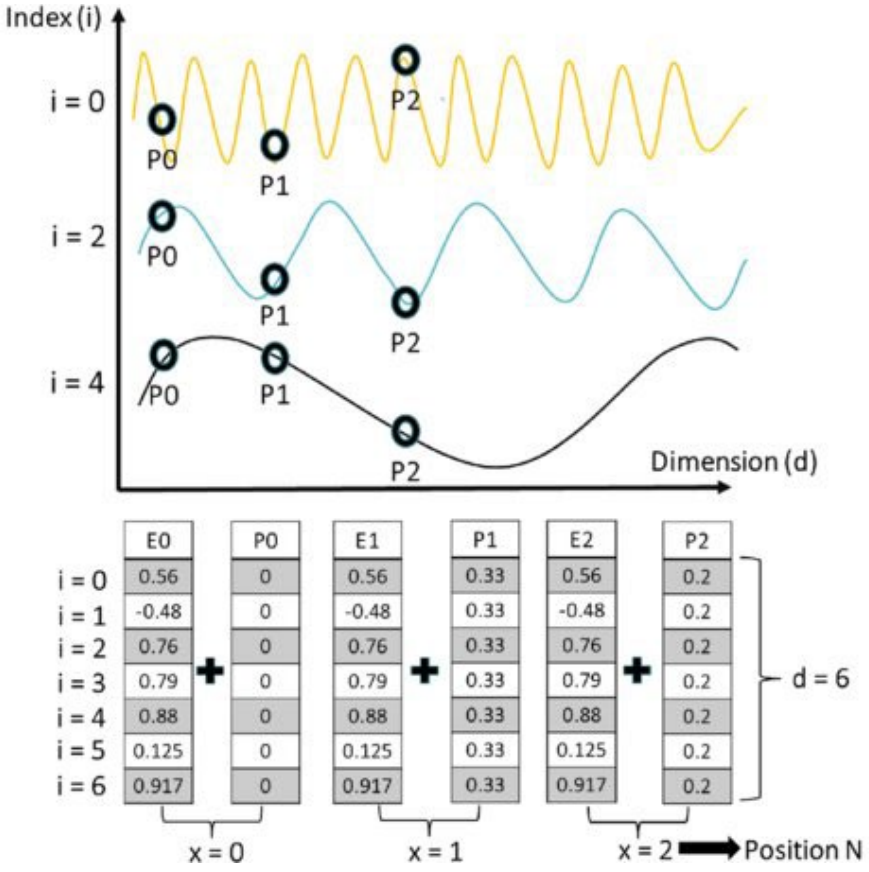


Рисунок 1.16 – Урахування позиції кожного фрейму [22]

Для того, щоб кожен окремий фрейм, що представлений у вигляді вектору ознак, мав інформацію про усі інші, наступним кроком має бути обраховування декількох значень для фрейму, а саме query, key та value. Усі вони вираховуються за допомогою множення вхідних значень на ваги, при чому для кожного фрейму ці ваги будуть однакові. Потім для кожного фрейму вираховується self-attention значення, яке включає в себе значення key з кожного іншого фрейму що дозволяє зрозуміти взаємозв’язки між різними фреймами. Наприкінці цього процесу вихідне значення додається

до вхідного, результат нормалізується, кожен фрейм окремо проходить через нейронну мережу прямого поширення, задля глибшого вивчення кожного фрейму окремо. Результат нормалізується та відправляється до наступного ідентичного шару трансформера, яких може бути декілька, після проходження всіх шарів в результаті формуються контекстні вектори, розмірністю такі самі як віхдні, проте кожен з них має інформацію про усіх інших. Навчанням моделі є її здатність до визначення приналежності вектору який маскується до значення, яке надалось цьому вектору при квантуванні, тобто при відсутності даних знаходити з контексту вірне рішення.

Таким чином, в цьому огляді прослідковується еволюція методів feature extraction з аудіозаписів, причому майже кожен з методів що був описаний раніше, прямо чи посередньо використовується в наступному методі, отримуючи вдосконалення та збільшення точності ознак, що допоможе в задачах класифікації.

Окрім того, в тренуванні моделі буде використана агументація. Агументація – це маскування деякої частини з набору даних, що імітує відсутність певних частот або фреймів аудіозапису, що допомагає не допустити перенавчання, так і навчитись робити висновки з неповних даних. Приклад агументації наведено на рисунку 1.17.

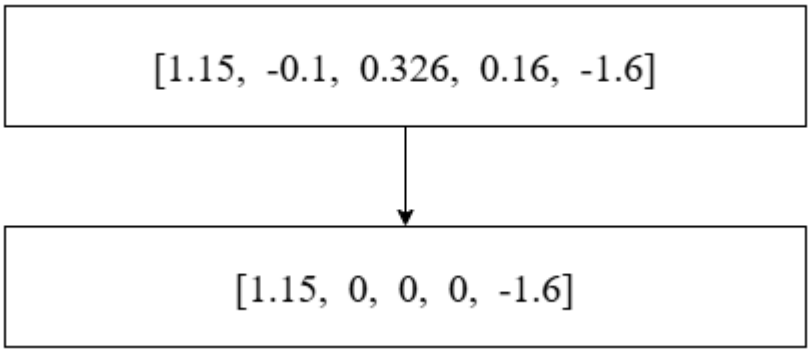


Рисунок 1.17 – Агументація даних

В цій процедурі задається довжина та кількість замаскованих частин, позиція початку маскування обирається випадковим чином. Окрім того, використовується асамблювання, яке об’єднує результати декількох моделей, навчених на даних, що поділені по різному, та обирає результат як середнє з цих моделей.

Останньою процедурою, що відноситься до тренування моделей, є підбір гіперпараметрів. Для того, щоб вирішити цю проблему більш ефективно, в цій роботі буде використовуватись бібліотека Optuna, що дозволяє підібрати гіперпараметри наступним чином – вручну задаються значення різних гіперпараметрів, таких як learning rate, dropout тощо, також задається кількість ітерацій. Бібліотека буде комбінувати значення наданих параметрів та середніх між ними та тренувати модель невелику кількість епох, порівняно зі звичайним тренуванням. Це дозволить зберегти час на невдалих комбінаціях та не тренувати модель до кінця. На кожній ітерації запам’ятовується комбінація параметрів, що була використана, та в результаті повертається найкращій варіант.

1.4 Постановка задачі

Отже, виходячи з актуальності проблеми, огляду існуючих аналогів, обов’язкових елементів, вимог для задач цієї сфери та предметної галузі, перейдемо до визначення задачі дослідження.

У зв’язку з погіршенням психологічного стану українців через війну та інші фактори постає необхідність створення інтелектуального застосунку психологічної допомоги, який має задовольняти наступним критеріям:

– поєднання результатів психологічного тестування з аналізом емоцій за допомогою голосу;

– автоматичне визначення емоційного стану користувача;

– формування рекомендацій відповідно до потреб пацієнта;

– надання можливості запису на прийом та проведення консультацій через відеозв'язок;

– реалізація інструментів для заповнення діагнозу та рекомендацій за підсумками консультації;

– реєстрація користувачів, можливість змінювати свої дані, зв’язуватися з службою підтримки для вирішення технічних або медичних питань;

– керування акаунтами користувачів з можливістю видалення авторизованим користувачем за потреби.

В ході розробки необхідно вирішити наступні задачі:

– розробити метод комбінованої оцінки емоційного стану пацієнта на основі аналізу голосових характеристик та результатів психологічного тестування;

– створити алгоритм об'єднання результатів тесту та голосового аналізу за допомогою методів нормалізації та вагового об'єднання й побудови фінальної емоційної оцінки;

– реалізувати систему організації консультацій через відеоконференції з безпечним підключенням пацієнта та лікаря;

– забезпечити можливість фіксації результатів консультацій, збереження рекомендацій, діагнозів та записів прийомів для подальшого перегляду;

– створити зручний користувацький інтерфейс для роботи пацієнтів та психологів з урахуванням вимог конфіденційності та безпеки даних.

1.5 Висновки за розділом

В цьому розділі було проведено детальний опис та аналіз предметної галузі, що дозволило прослідкувати розвиток ідей розпізнавання емоцій за голосом, еволюцію методів та технологічних рішень для цієї задачі, які були

зумовлені загальним прогресом в розвитку штучного інтелекту та нейронних мереж зокрема. Також був проведений аналіз визначеної проблематики в контексті українського суспільства, шляхом вивчення та систематизації результатів опитувань різних соціологічних агенцій, що дає розуміння щодо наявної актуальності цієї теми та необхідність розробки нових рішень. Для початку планування та розробки рішення, що буде задовільняти нагальним потребам ринку, був проведений аналіз наявних рішень, як українських, так і закордонних. В результаті цього аналізу можна зробити висновок, що закордонні аналоги не відповідають потребам українського ринку в умовах війни, а наявні українські застосунки можуть не підходити деяким користувачам через обмежену можливість проходити об’єктивне тестування самостійно.

Після підтвердження актуальності та аналізу аналогічних застосунків було проведено дослідження методів та засобів, що знадобляться для тренування моделі нейронної мережі, а саме простежено еволюцію ознак, що в різний час виокремлювались з аудіозаписів, були описані методи покращення роботи моделі шляхом проведення агументації над векторами ознак.

Після дослідження усіх необхідних інструментів було проведено постановку задачі, що визначила основні функції, що має виконувати застосунок.

Усі вищезазначені процедури надають можливість перейти до проектування застосунку та реалізації його серверної частини, що описується в наступному розділі.

2 ПРОЕКТУВАННЯ ЗАСТОСУНКУ ТА РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ

2.1 Архітектура застосунку

Виходячи з постановки задачі, необхідно спроектувати застосунок, який буде задовільняти поставленим функціональним та нефункціональним вимогам та надавати усім користувачам можливість отримувати необхідні послуги. Для цього для кожної задачі доцільно розробити окремий модуль який буде містити відповідну бізнес логіку та виконувати відповідну функцію.

Слід зазначити, в рамках кваліфікаційної роботи функціональний прототип застосунку буде розроблений із застосуванням монолітної архітектури, задля спрощення реалізації. Хоча, у реальних проектах такого масштабу доцільніше було б розробити застосунок з використанням мікросервісної архітектури, що додає гнучкості горизонтального масштабування конкретної бізнес-функції. Так, пропонується розробити 13 функціональних модулів, які дозволять виконувати цілі та задачі, які були поставлені в минулому розділі. В майбутньому ці модулі можуть бути природньо використані для переходу на масштабовану архітектуру. На рисунку 2.1 представлено контекстну діаграму з моделі C4 [23] яка репрезентує зовнішню взаємодію системи в з користувачами та іншими системами.

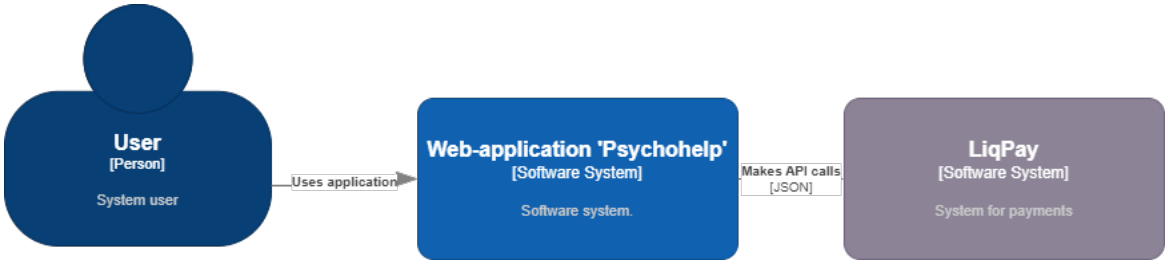


Рисунок 2.1 – Контекстна діаграма

Очевидно, що застосунок комунікує з зовнішньою системою для прийому платежів, у випадку цієї роботи – LiqPay. Для взаємодії з користувачем пропонується реалізувати Web-базований графічний інтерфейс. Для визначення більш детального рівня опису системи доцільно перейти до наступного класу C4 діаграм.

Цим кроком є контейнерна діаграма (рисунок 2.2). На даному рівні система розбивається на контейнери з описом зв'язків між ними. Кожен контейнер реалізує одну чи декілька бізнес-функцій, що запрограмовані в ньому (це можуть бути як бізнес-логіка чи функціональна можливість, наприклад збереження даних).

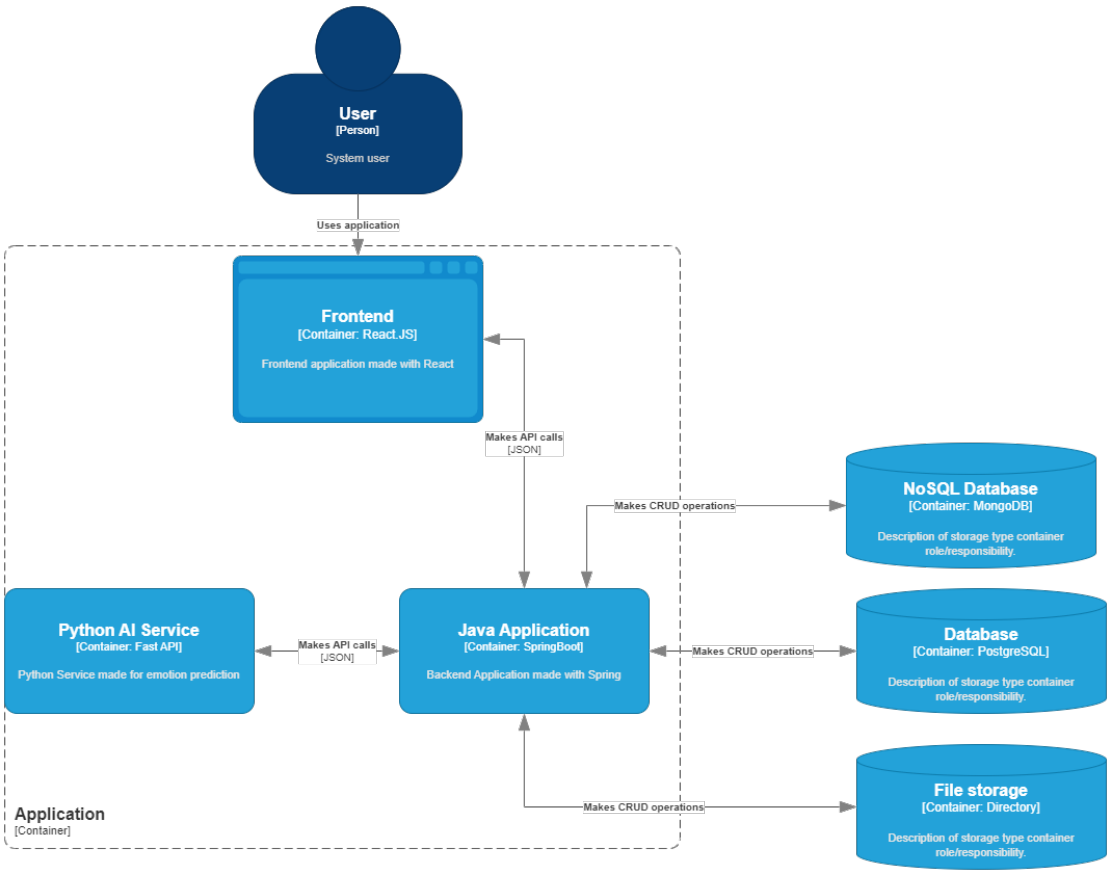


Рисунок 2.2 – Контейнерна діаграма

На даному класі діаграм визначено, що застосунок має в собі три окремі контейнери для реалізації бізнес-логіки, а саме:

– графічна частина, написана на React.JS, про яку мова піде в четвертому розділі;

– основної частини застосунку, яка написана на мові програмування Java, ця мова програмування була обрана через те, що вона оптимально підходить для високонавантажених застосунків великих організацій та завдяки JVM та компіляції в Java bytecode що дозволяє запускати один і той самий код на будь якій платформі, що оптимізує процес розробки [24];

– компонента штучного інтелекту для прогнозування емоцій за записом голосу, про яку більшою мірою мова піде в наступному, третьому розділі, для тренування моделі та розробки якої використовується Python.

На даній діаграмі також визначено, що доступ до компоненти штучного інтелекту здійснюється через Java додаток, що захищає від несанкціонованого доступу та робить API уніфікованим, що спрощує відправку запитів з графічної частини.

Перед тим, як перейти до опису специфіки зберігання даних, необхідно розкрити поняття «Purpose-built databases». Воно описує можливість використовувати на одному проекті різні системи управління базами даних (СУБД), які створені спеціально для обробки певного типу даних або для виконання конкретних задач, вони оптимізовані під конкретні сценарії використання. В англомовних ресурсах часто підміняють поняття СУБД та БД. Тому, в теоретичній частині, будемо використовувати обидва терміни в контексті визначення систем управління базами даних.

Як приклад можна навести звичну реляційну базу даних, базу даних «ключ-значення», документну, графову, тощо. Кожна з яких краще вирішує певні класи задач та типи даних, такі як доступ за ключем, що пришвидшує швидкодію, або робота з пов’язаними даними, прикладом чого

можуть слугувати соціальні мережі або маршрути, рекомендаційні системи тощо. Приклади таких баз даних, що надає AWS наведено на рисунку 2.3.

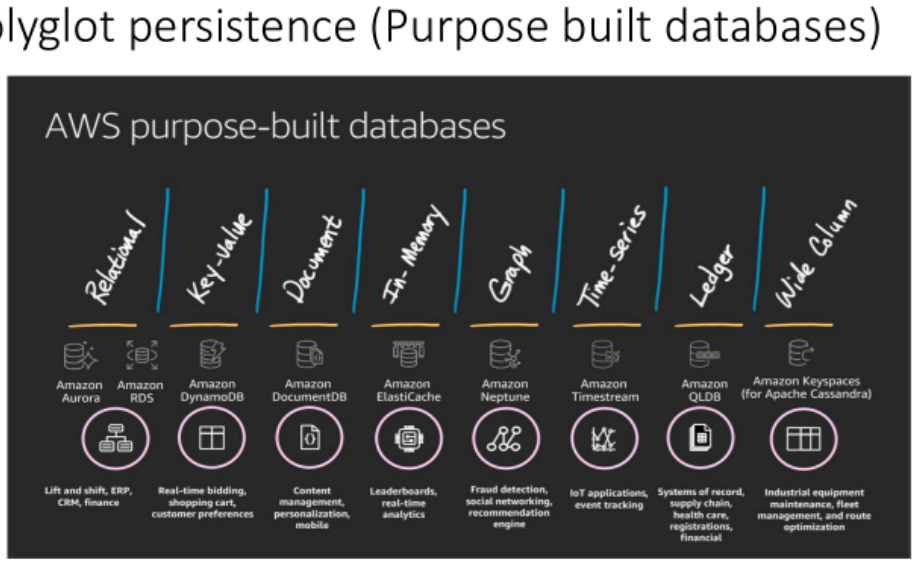


Рисунок 2.3 – Purpose built databases

На рисунку 2.2 можна бачити, що у якості засобів постійного зберігання використовуються:

– система управління реляційними базами даних (СУРБД) PostgreSQL, є основним засобом збереження та маніпулювання даними, її всеосяжна схема буде розглянута пізніше, вона відповідає за збереження інформації про користувачів, їх прийоми, анонімні історії якими вони бажають поділитися з іншими, скарги на інших користувачів тощо. Ця СУБД була обрана як найпоширеніша реляційна СУБД, що підтримує ACID принципи [25], що гарантує консистентність даних, що особливо важливо в таких чутливих даних як пов’язані з медициною;

– нереляційна документна СУБД MongoDB, яка слугує для збереження розкладу лікарів та призначення слотів прийому лікаря до прийому який створюється. Ця високопродуктивна СУБД була обрана тому що вона наразі є одним з найкращих рішень на ринку NoSQL баз даних [26],

що працюють з слабоструктурованими даними та теж задовольняють потребам безпеки даних, ACID принципам та розширенням потужностей за потреби, можуть мати безкоштовні дистрибутиви. Схеми баз даних, що реалізуються зазначеними СУБД, будуть детально описані далі;

– файлове сховище – наразі файлове сховище реалізовано в якості директорії на диску, та дозволяє зберігати файли, такі як зображення профілю користувачів та можливого збереження деперсоніфікованого запису голосів користувачів, задля подальшого аналізу та можливого донавчання моделі з використанням записів української мови.

Після опису визначених контейнерів перейдемо до розгляду їх компонентів, який буде виконувати поставлені задачі та функції. Для цього моделлю C4 передбачено використовувати діаграму компонентів. Діаграму компонентів можна побачити на рисунку 2.4.

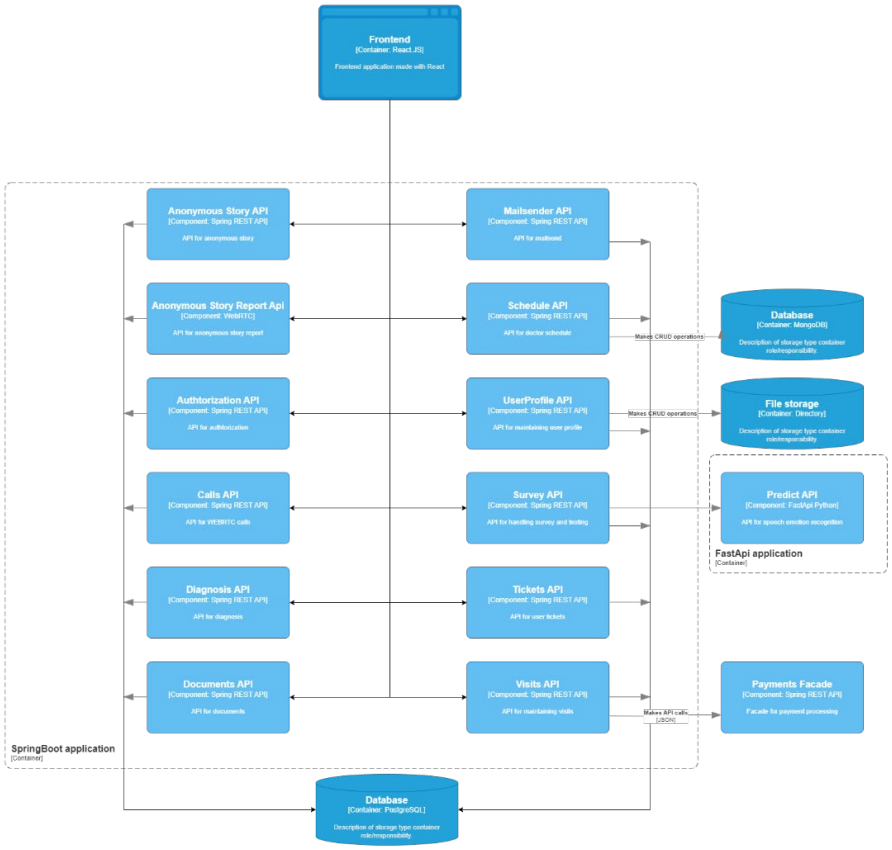


Рисунок 2.4 – Діаграма компонентів

На рисунку 2.4 зображена діаграма компонентів, яка декомпозує контейнери на компоненти та виділяє зв’язки між компонентами, які планується інтегрувати або розробляти у застосунку. Наявні модулі відповідають визначеним функціям застосунку та надають користувачам можливість користуватися системою та отримувати необхідні послуги.

Детальний перелік функцій кожного модулю та їх призначення:

– модуль авторизації має функції реєстрації користувачів із роллю доктор або пацієнт, видалення користувачів, зміни їх пошти та паролю, також використовується як допоміжний модуль для вибору користувачів за їх роллю. Окрім цього цей модуль оперує допоміжними класами для генерації та валідації JWT токенів, хешування паролю для безпеки, конфігурації авторизації та інших фільтрів. В базі даних PostgreSQL оперує таблицею під назвою users;

– модуль анонімних історій має функції перегляду історій які опублікували інші пацієнти, зокрема з використанням пагінації для більш зручного та ефективного доступу, також цей модуль дозволяє публікувати анонімні історії, для адміністраторів він надає змогу видаляти та змінювати зміст історій на основі скарг від користувачів або самостійної перевірки. Окрім цього цей модуль оперує допоміжним модулем тегів, який дозволяє користувачам створювати унікальні теги для своїх історій, або додавати вже існуючі, для можливості пошуку історії за тегами. Допоміжний модуль оперує таблицею під назвою tags, а модуль анонімних історій оперує таблицею під назвою anon_stories;

– модуль скарг на анонімні історії дозволяє пацієнтам скаржитись на наявні історії, а адміністраторам переглядати скарги та надавати відповіді користувачам за допомогою модулю mailsender. У базі даних оперує таблицею під назвою anonstories_reports;

– модуль дзвінків за допомогою WebRTC та WebSocket дозволяє створювати безпечне P2P з’єднання між лікарем та пацієнтом,

використовуючи власний механізм для забезпечення безпеки, щоб до визначеної сесії міг приєднатись тільки певний лікар та пацієнт;

– модуль діагнозів є допоміжним модулем для лікаря, в якому містяться усі діагнози з категорії розладів психіки та поведінки за міжнародною класифікацією МХК-10, які лікар може знаходити за кодом діагнозу, назвою або частиною назви, модуль оперує таблицею diagnosis;

– модуль документів надає можливість лікарю після створення профілю завантажити необхідні документи, такі як паспорт, сертифікати та дипломи, інші документи які можуть знадобитися лікарю для підтвердження його статусів. Цей модуль зберігає документи у файловому сховищі на сервері, створюючи для них окрему директорію для кожного лікаря та форматуючи назви файлів за номером їх отримання;

– модуль відправки повідомлень виконує функцію відправлення користувачам інформаційні повідомлення щодо стану їх звернень, а саме їх створення, вирішення або відміни адміністратором. Цей модуль дозволяє адміністраторам в автоматичному режимі надсилати на пошту користувачам рішення щодо їх скарг або запитів, прибираючи необхідність технічному адміністратору робити це самостійно;

– модуль виконує функції керування розкладом лікарів, а саме дає можливість змінювати шаблони кожного дня тижня для лікаря, керувати окремими днями, а також виконує функцію керування слотами прийомів які створюються або видаляються за допомогою модулю прийомів. Цей модуль використовує нереляційну систему управління базами даних MongoDB для зберігання та керування розкладом кожного лікаря у форматі JSON через те, що такий тип даних забезпечує контрольовану гнучкість для структур, що можуть активно змінювати свою схему в залежності від функціональних потреб (контроль схеми відбувається за рахунок функціоналу СУБД та принципів моделювання – використання версій структур);

– модуль тестування реалізує комбінований алгоритм тестування психологічного стану та має можливості формувати результати як за

допомогою запису голосу, передаючи його до модулю, написаному на Python, так і додаючи за потреби результати тестування за допомогою традиційного тесту, комбінуючи результати з різним можливим значенням «ваги» кожного компоненту, що пояснюється далі;

– модуль звернень реалізує можливість лікарям та пацієнтам написати в службу підтримки з метою вирішення будь якого питання, для реагування та відповіді адміністратора, та отримання відповіді на пошту за допомогою модулю mailsender. В базі даних оперує таблицею tickets;

– модуль користувачів надає можливість усім користувачам змінювати дані профілю, додавати, оновлювати та видаляти зображення профілю, видаляти свій профіль, надає можливість для докторів дивитися профілі пацієнтів, для пацієнтів дивитися профілі докторів, для адміністраторів керувати профілями користувачів. Також використовується в модулі запису на прийом як допоміжний модуль. Використовує модуль авторизації для зміни пошти профілю. В базі даних оперує таблицею user_profiles;

– модуль прийомів забезпечує створення, редагування та відміну прийомів до лікаря, яке може здійснюватися лікарем чи пацієнтом, у комунікації з модулем розкладу керує слотами прийому лікаря, в комбінації з модулем діагнозів можливо додавати діагнози для прийомів. Також модуль містить функцію оплату прийомів, шляхом інтеграції платіжного сервісу (в даному випадку LiqPay), генеруючи посилання на оплату, за якими перейде пацієнт для оплати. В базі даних модуль оперує таблицею visits та допоміжними таблицями про що буде зазначено нижче.

2.2 Дизайн баз даних

Після опису архітектури застосунку доцільно визначити СУБД які будуть використовуватись та розробити їх схеми даних. Як вже зазначалось

раніше, для реалізації цього проекту знадобиться реляційна СУБД PostgreSQL та нереляційна MongoDB.

Виходячи з того, що для реалізації проекту обрано використання декількох СУБД, етап концептуального проектування було винесено в загальний розділ. При цьому етапи логічного і фізичного – в окремі, через те, що різні класи СУБД мають свою вимоги до проектування на зазначених етапах.

2.2.1 Концептуальна модель даних

Для коректного та послідовного проєктування бази даних доцільно йти від абстрактних понять до більш повних. Одним з інструментів такого проєктування може виступати перехід за структурою від концептуальної до логічної та потім до фізичної моделі даних [27].

Для цього спочатку було спроєктовано концептуальну модель даних, яка представлена на рисунку 2.5.

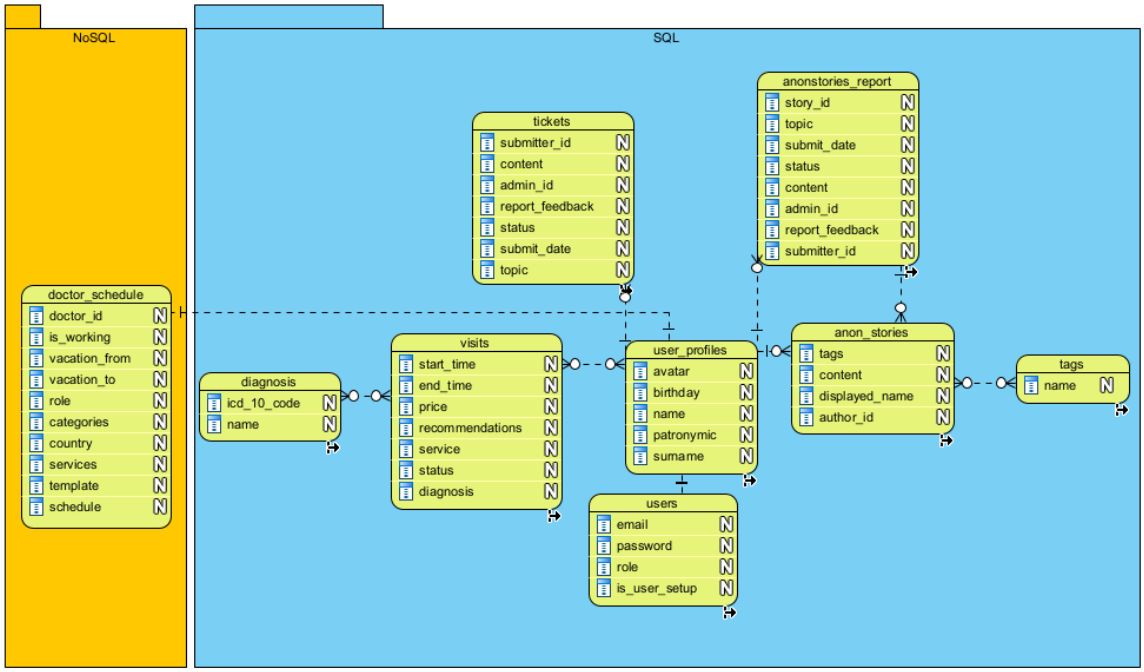


Рисунок 2.5 – Концептуальна модель даних

На цьому рисунку концептуальна модель даних дозволяє визначити сутності на високому рівні абстракції, наближеного до бізнес-визначень принципових концептів, не переходячи одразу до повної атрибутної специфікації, що одразу дає можливість побачити зв'язки між сутностями. Подальший розгляд розпочнемо з традиційної реляційної СУБД.

2.2.2 Дизайн реляційної бази даних

Подальшим кроком в цьому підході є створення логічної моделі даних, яка представлена на рисунку 2.6. Проєктування відбувається з використанням Visual Paradigm (Community version).

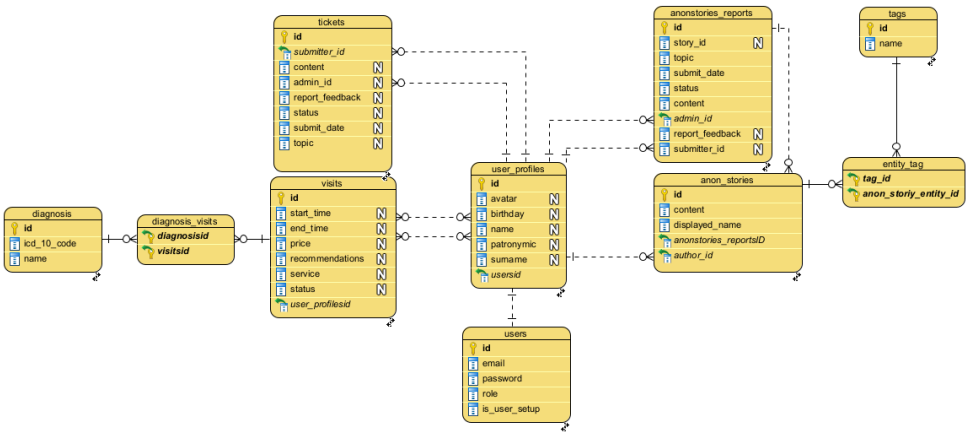


Рисунок 2.6 – Логічна модель даних

На цьому етапі проєктування було визначено атрибути кожної сутності. Логічна модель адаптована під цільовий тип бази даних, таким чином, описує бізнесові концепти і зв'язки між ними моделюються з урахуванням реляційної моделі даних і вимог до її проектування. Наступним та фінальним кроком є перехід до фізичної моделі даних яка залежить від обраної СУБД (PostgreSQL) та є фінальним варіантом. Діаграма цієї моделі даних представлена на рисунку 2.7.

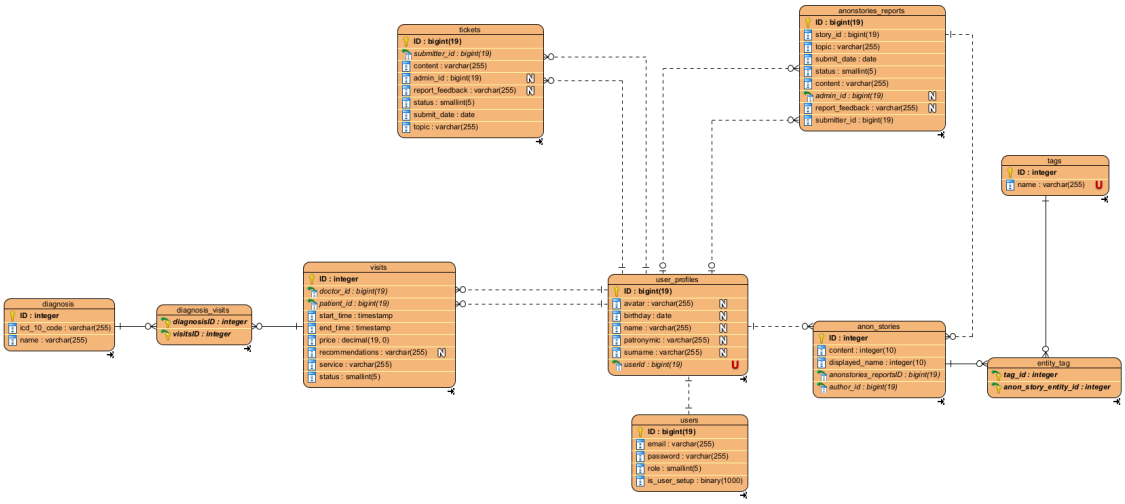


Рисунок 2.7 – Фізична модель даних

На рисунку 2.7 зображено третій та фінальний етап проєктування бази даних – фізична діаграма. З неї можна бачити конкретні типи даних для кожного атрибуту кожної таблиці. На основі цієї діаграми можна с повною впевненістю створювати зазначені сутності в конкретній СУБД. Окремо зазначимо, що для проектування схеми БД використовується Top-Down підхід, коректність результату перевіряється шляхом використання практик нормалізації.

2.2.3 Дизайн нереляційної документної бази даних MongoDB

Для зберігання даних було обрано нереляційну документно-орієнтовану СУБД. Ключовою особливістю такого підходу є те, що він передбачає збереження даних у вигляді колекцій об'єктів. Ці об'єкти, як правило, зберігаються у форматі JSON, що дозволяє представляти складні ієрархічні структури природним чином. На відміну від реляційних моделей, схема даних не повинна бути заздалегідь жорстко визначеною, що надає значну гнучкість при розробці та подальшій модифікації системи. Для того, щоб зберігати дані в зручній формі було сплановано структуру, яку наведено на рисунку 2.8.

Рисунок 2.8 – Схема нереляційної бази даних

На рисунку зображено детальну структуру даних для розкладу, який автоматично створюється для кожного лікаря в момент його реєстрації в системі. Дана діаграма була спроєктована з використанням спеціалізованого засобу гетерогенного моделювання Hackolade (polyglot modelling). Важливим архітектурним рішенням стало додавання до розкладу поля зі значенням версії схеми. Цей підхід забезпечує гнучкість системи, оскільки в подальшому, при зміні структури, з'явиться можливість коректно працювати з різними версіями об'єктів, уникаючи помилок сумісності. Розклад, який встановлюється лікарем, складається з основного шаблону, що визначає типові робочі години, та конкретних днів розкладу, які створюються на їх основі. Схематично ця ієрархічна структура з детальними поясненнями представлена на рисунках 2.9 та 2.10.

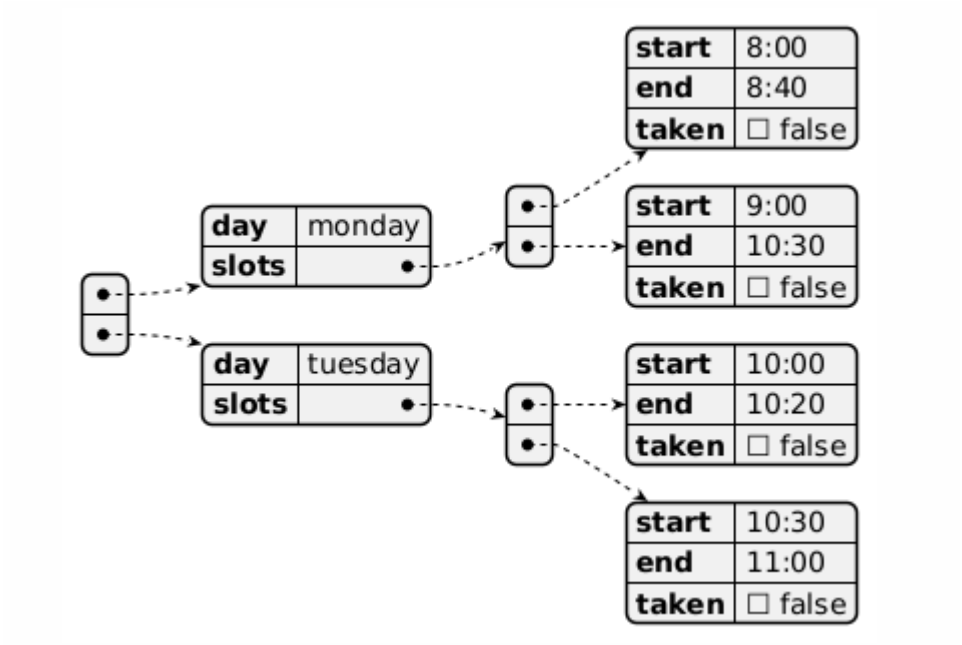


Рисунок 2.9 – Шаблон розкладу лікаря

В кожного лікаря існують шаблони на кожен день тижня, які слугують базовими налаштуваннями кожного дня. На основі цих шаблонів можливо створювати розклад лікаря на конкретні дні, зі слотами в яких буде зазначено ідентифікатор візиту, як продемонстровано на рисунку 2.10

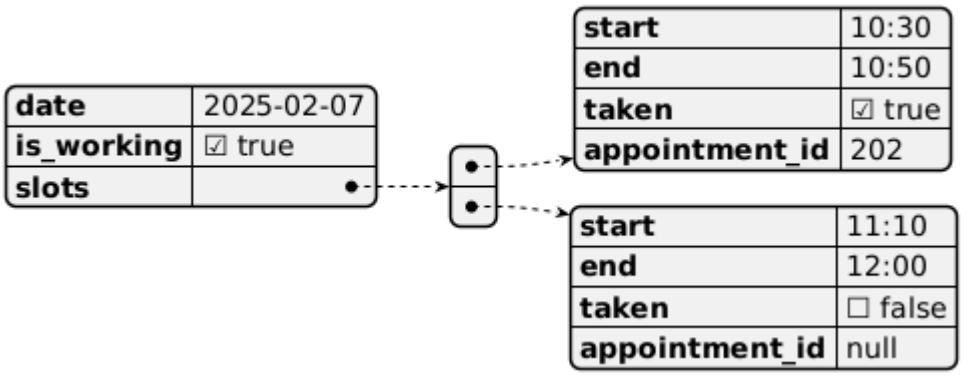


Рисунок 2.10 – Розклад лікаря на визначений день

Завдяки плануванню сутності розкладу саме таким чином, стає можливим керування окремими днями, загальним розкладом, відпусткою.

Через те що кожен з зайнятих слотів розкладу містить в собі ідентифікатор прийому, стає можливим відміна прийому та звільнення слоту.

2.3 Реалізація застосунку

В цьому підрозділі будуть описані лише найважливіші частини застосунку, які виконують функції які описані в постановці задачі.

2.3.1 Модуль тестування

Однією з найважливіших задач які вирішуються в даній роботі, є тестування психологічного стану пацієнта з використанням аналізу голосу. Для цього потрібно реалізувати логіку комплексного тесту який буде поєднувати результати традиційного тестування та тестування за допомогою нейронної мережі. Для тренування нейронної мережі було використано набір даних IEMOCAP, через причини які будуть пояснені в наступному розділі, цей набір даних має в собі шість емоцій для аналізу а саме: злість, роздратованість, щастя, захоплення, нейтральність та сум.

Через це для створення тесту з варіантами відповідей було обрано питання для визначення таких емоцій. Питання тесту з варіантами відповідей та відповіді на них наведено нижче:

– ви часто дратуєтесь, якщо хтось заважає вашим планам;
– вас легко вивести з себе у пробці чи черзі;
– коли щось йде за планом, вам важко зберігати спокій;
– чи буває, що ви відчуваєте внутрішнє роздратування через дрібниці;
– ви часто радієте дрібницям, як теплій погоді чи смачній їжі;
– чи легко ви переймаєте гарний настрій від інших людей;
– вас легко надихнути новою ідеєю чи проектом;
– чи часто ви відчуваєте передчуття перед цікавими подіями;

– у більшості ситуацій ви почуваєтеся спокійно та врівноважено;
– ваш настрій рідко змінюється раптово протягом дня;
– чи буває, що без причини вам стає сумно;
– ви часто згадуєте неприємні моменти і переживаєте їх заново.

В цьому списку питань кожна емоція представлена двома питаннями, та на кожне з них є п’ять варіантів відповіді, наведених нижче:

– ніколи;
– рідко;
– іноді;
– часто;
– дуже часто.

Кожне з цих значень на сервері конвертується в числове значення: 0, 0.25, 0.5, 0.75, 1 відповідно. Тоді для кожної емоції з шести представлених формується оцінка, за формулою:

$$e_i = \frac{r_{i1} + r_{i2}}{2}, \tag{2.1}$$

де e_i – ймовірність емоції з індексом i;

r_{i1}– відповідь на перше питання емоції з індексом i;

r_{i1}– відповідь на друге питання емоції з індексом i.

Після отримання оцінок емоцій з традиційного тесту потрібно нормалізувати їх, шляхом застосування до кожної з них формули 2.2.

$$f_i = \frac{e_i}{\sum_{j=0}^{5} e_j}, \tag{2.2}$$

де f_i – фінальна ймовірність емоції з індексом i;

$\sum_{j=0}^{5} e_j$– сума ймовірностей всіх емоцій.

Завдяки цим операціям було отримано фінальні оцінки для кожної з емоцій, які для отримання кінцевого результату потрібно скомбінувати з результатами тестування голосу. Як буде описано в наступному розділі, нейронна мережа повертає ймовірності для кожної з емоцій, які в сумі складають одиницю. Результати традиційного тесту вже нормалізовані, тому останньою операцією є комбінування результату за функцією 2.3.

$$f_i = n_i * \alpha + t_i * (1 - \alpha), \tag{2.3}$$

де f_i – фінальна ймовірність емоції з індексом i;

n_i – ймовірність емоції з індексом i з результату нейронної мережі;

t_i – ймовірність емоції з індексом i з результату традиційного тесту;

α – вага результату нейронної мережі (обирається від 0 до 1).

В результаті проведених обчислень отримано значення ймовірності для кожної з емоцій, які сервер повертає як відповідь. Про те як ця відповідь обробляється на стороні клієнта детальніше описано в четвертому розділі. На рисунку 2.10 представлена діаграма послідовностей для проходження психологічного тестування.

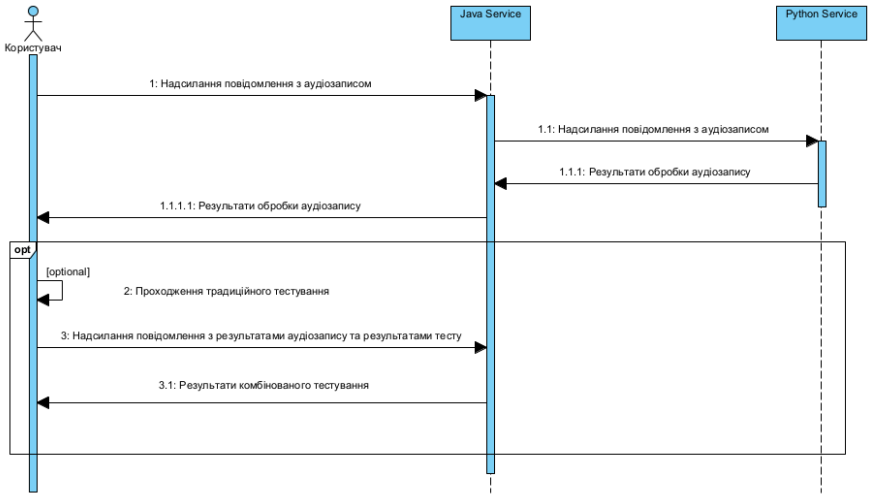


Рисунок 2.10 – Діаграма послідовностей для проходження тестування

На цій діаграмі продемонстровано проходження психологічного тестування пацієнтом, з якої видно, що частина тестування з відповідями на питання не є обов’язковою, та проводиться за бажанням пацієнта, заради того щоб покращити точність, тому що наразі не можна покладатись цілком на частину зі штучним інтелектом.

2.3.2 Модуль дзвінків

Наступною важливою функцією застосунку є проведення відеодзвінків між пацієнтом та лікарем. Для реалізації цього модулю було використано інтернет-протокол WebRTC для з’єднання та зокрема його веб інтерфейс RTCPeerConnection для зв’язку.

WebRTC це протокол який дозволяє створити можливість зв’язку в режимі реального часу, що підтримує передачу голосу, відео та загальних даних, що доступно у всіх поширених браузерах [28]. RTCPeerConnection це частина специфікації WebRTC яка дозволяє двом комп’ютерам створювати з’єднання за допомогою протоколу peer-to-peer, також відомим як одноранговий зв’язок, з’єднання напряму двох користувачів мережі [29].

Для того щоб встановити безпечне з’єднання потрібно зробити певні дії, які наведені далі. Початкове з’єднання створюється за допомогою WebSocket, протоколу, що забезпечує двостороннє з’єднання між сервером та клієнтом, на відміну від REST запитів. Для того, щоб з’єднання було безпечним, потрібно перевірити чи є користувач, що під’єднується до дзвінка, пацієнтом чи лікарем в цьому прийомі. Для цього створюється перехоплювач авторизації, який за допомогою вищеописаного модулю прийомів перевіряє чи дозволено користувачу під’єднатися на основі його JWT (Json Web Token) із запиту. Якщо з’єднання дозволене користувач або стає ініціатором з’єднання, якщо воно ще не створене, або відповідачем, якщо з’єднання вже почалося. Той користувач, що стає ініціатором створює RTCPeerConnection та створює offer. Offer це повідомлення яке створюється

ініціатором та надає іншій стороні пропозицію встановити однорангове з’єднання, це повідомлення містить в собі інформацію про WebRTC сесію та дані про ініціатора. Відповідач отримує цей offer, зберігає дані про ініціатора та створює повідомлення типу answer, яке відправляється ініціатору та містить дані про відповідача. Відповідач їх зберігає та переходить до наступного кроку. Окрім обміну медіаінформацією за допомогою SDP як було пояснено вище, абонентам потрібно обмінятись інформацією про мережеве з’єднання. Для цього використовується ICE (Interactive connection establishment) – встановлення інтерактивного з’єднання. WebRTC використовує цей фреймворк для з’єднання двох вузлів (peers) незалежно від топології мережі [30]. В кожної сторони є список ICE-кандидатів, тобто можливих адрес для підключення. Вони обмінюються цими кандидатами один з одним та ICE обирає найкращу пару адрес для підключення, яка використовується в подальшому. На цьому процес з’єднання завершується та розпочинається трансляція аудіо- та відеопотоків, які були передані на минулих етапах, через ICE з’єднання яке було встановлене.

Для візуалізації створено діаграму послідовностей з процесом створення веб сокету, що наведена на рисунку 2.11.

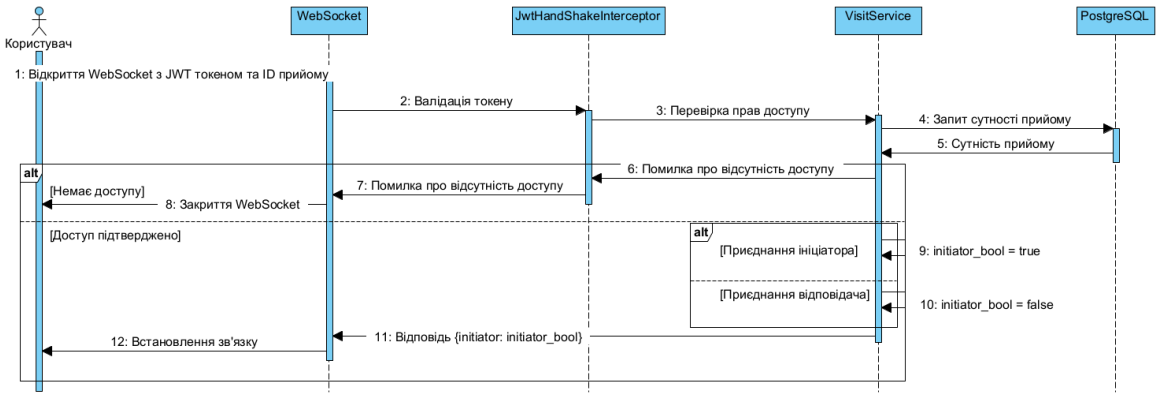


Рисунок 2.11 – Створення веб сокету

2.3.3 Безпека та конфіденційність застосунку

Окрім функціональних вимог, що були описані раніше, існують важливі нефункціональні вимоги, які мають виконуватись в застосунках медичного спрямування. Одними з найважливіших таких вимог виступають вимоги безпеки та конфіденційності. Для їх впровадження в застосунок було впроваджено авторизацію та аутентифікацію з використанням JWT. JWT (JSON Web Token) – компактний та безпечний спосіб представлення інформації, що передається між двома сторонами. JWT представляє собою JSON об’єкт який містить в собі інформацію при користувача, час життя цього токену, та інформацію, яка передається до серверу у кожному запиті [31]. Цей токен видається сервером при авторизації та містить дані про користувача, які знадобляться в його запитах до сервера. Після отримання токену клієнт має додавати його в заголовки для кожного запиту до сервера. Токен, що був використаний в цій роботі, наведено на рисунку 2.12.

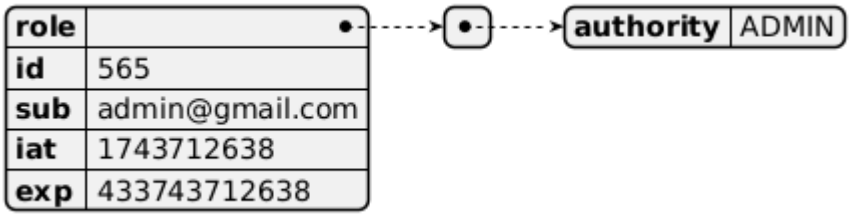


Рисунок 2.12 – Структура JWT

Також є необхідним створити механізм перевірки JWT на стороні серверу, для чого необхідно використовувати механізми безпеки, які надає Spring Security. В кожного сервлету за замовчуванням є фільтри які має пройти кожен запит. Типова діаграма цього процесу показана на рисунку 2.13.

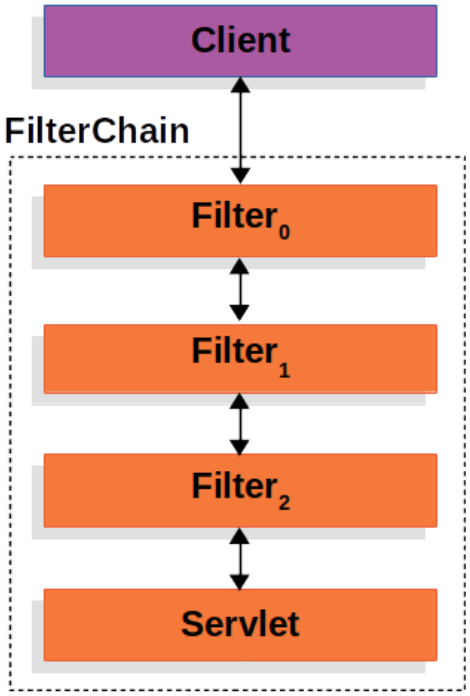


Рисунок 2.13 – Ланка фільтрів запиту [32]

Заради того, щоб аутентифікувати запит було створено створити власний фільтр, який буде вбудовуватись в типову ланку фільтрів та перевіряти необхідні доступи. Цей процес показаний на рисунку 2.14.

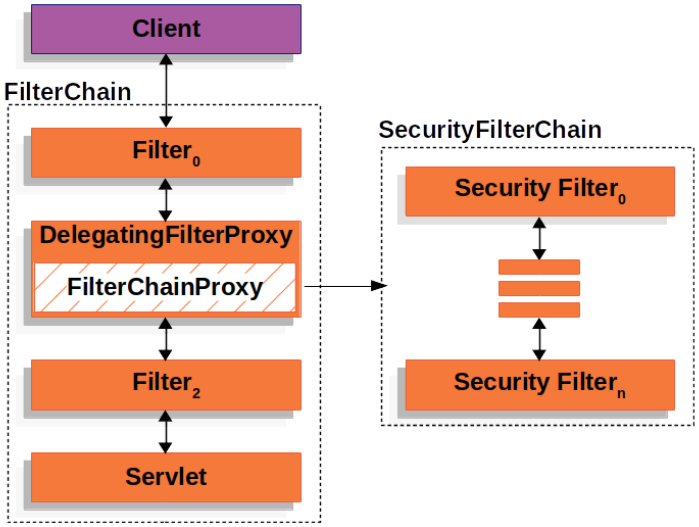


Рисунок 2.14 – Вбудовування власного фільтру [32]

Як показано на діаграмі, в ланку фільтрів вбудовується фільтр, який насправді може являти собою декілька фільтрів, які можуть виконуватись послідовно, наприклад для різних типів авторизації. В рамках цього проєкту використовується один фільтр, задачею якого є валідація JWT та відправка запиту далі по ланцюгу фільтрів. Окрім того існує можливість захищати різні шляхи по різному, що показано на рисунку 2.15.

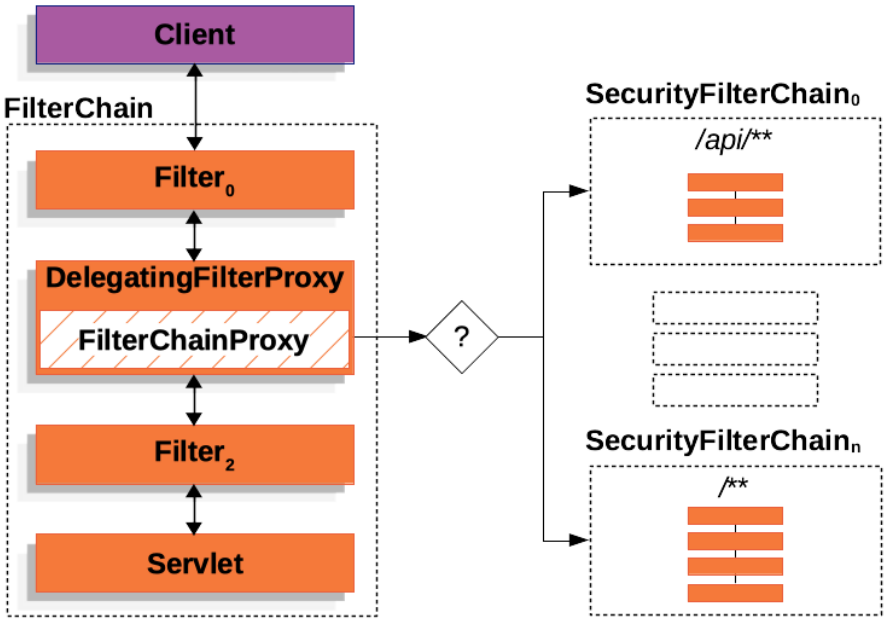


Рисунок 2.15 – Множинний фільтр [32]

В застосунку, що було розроблено деякі шляхи, наприклад реєстрація або доступ до викликів не потребують захисту, тому що вони або загальнодоступні, або використовують власний перехоплювач авторизації. Найважливіша логіка безпеки яка використовується в застосунку наведена в лістингу 2.1.

Лістинг 2.1 – Конфігурація аутентифікації

```
http.
csrf(AbstractHttpConfigurer::disable)
.authorizeHttpRequests(auth->auth
```

Продовження лістингу 2.1

```
.requestMatchers("/auth/login","/auth/register",
"/avatars/**", "/ws/**").permitAll()
.anyRequest().authenticated())
```

Усі запити окрім вищезазначених потребують валідації токену та, таким чином, підтвердження доступу. Окремою задачею, для вирішення є авторизація для кожного запиту. Для вирішення цієї задачі існують декілька підходів, наприклад написання фільтру як для авторизації або анотація над методами. Анотація це метадані до коду, які не впливають на сам код, що вони анотують, проте можуть робити інші операції до або під час коду, такі як логування чи генерувати додатковий код який буде виконувати другорядні функції[33]. В цьому проєкті використовується анотація @PreAuthorize яка анотує метод та дозволяє перевірити чи задовольняє умові певний вираз який задається в цій анотації. Ця анотація інтегрується з контекстом авторизації, спеціальним об’єктом де зберігається інформація про користувача після його успішної аутентифікації. Завдяки цьому застосунок може отримати усі дані про користувача, які сервер заклав в токен та на їх основі прийняти рішення про те, чи можна допустити виконання анотованого методу. Однак існують випадки, коли вбудованих функцій недостатньо, для цього було створено конфігурацію для перевірки id із сутності, представлену в лістингу 2.2.

Лістинг 2.2 – Конфігурація перевірки авторизації

```
Authentication auth =
SecurityContextHolder.getContext().getAuthentication();
CustomUserDetails userDetails = (CustomUserDetails)
auth.getPrincipal();
return userDetails.getId().equals(id);
```

2.4 Висновки за розділом

В цьому розділі було спроектовано архітектуру застосунку шляхом поступової деталізації за рахунок декомпозиції. Для чого був створений набір діаграм в нотації C4, від контекстної діаграми, до діаграми контейнерів та діаграми компонентів. Кожен наступний рівень діаграм описує систему, що проектується, більш детально. Також в цьому розділі пояснено основні архітектурні рішення, такі як використання polyglot persistence. В процесі проектування було визначено, з яких модулів складається застосунок та їх основний функціонал, які способи зберігання інформації використовуються та наведено обґрунтування для застосування кожного з них. Окрім того була описана концептуальна модель бази даних, що дозволило визначити сутності на високому рівні абстракції. Після чого було покроково описано дизайн реляційної бази даних, починаючи від логічної моделі даних, яка адаптована під цільовий тип бази даних, переходячи до фізичної моделі, що спроектована під конкретну СУБД. Для цього використовувалися Visual Paradigm та Hackolade. Цей аналіз надає повне розуміння того, як влаштована реляційна база даних та спрощує її розробку. Для нереляційної бази даних був проведений опис, що визначає структуру кожної сутності, яка в ній зберігається. Було представлено кожну частину сутності в JSON форматі для наочного розуміння структури. Для візуалізації структур JSON використовувався PlantUML.

У цьому розділі описано ключові модулі застосунку: психологічного тестування, включаючи його структуру та математичні операції, що виконуються, модуль дзвінків між лікарем та пацієнтом, а також модуль безпеки, що охоплює аутентифікацію та структуру JWT токену.

Наступним кроком є розробка моделі штучного інтелекту, що вимагає аналізу та підготовки наявних наборів даних, як це описано в наступному розділі.

3 РОЗРОБКА МОДЕЛІ ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Вибір та опис набору даних

Для досягнення мети, що була визначена у постановці задачі, необхідно натренувати модель штучного інтелекту, що буде аналізувати голос користувача. Для тренування цієї моделі потрібно обрати набір даних, який буде аналізуватися та використовуватись для тренування моделі. Одними з найпопулярніших датасетів є RAVDESS, зібраний Стівеном Лівінгстоном та Франком Руссо [34], який містить в собі записи голосу 24 професійних акторів, 12 чоловіків та 12 жінок, які проговорюють дві стандартизовані фрази у спокійній, веселій, радісній, злій, боязкій, здивованій та огидній інтонаціях, та також містить пісні в деяких з цих емоцій, окрім того інформація доступна в якості аудіозаписів, відеозаписів зі звуком та відеозаписів без звуку, що також дозволяє аналізувати міміку.

Датасет оприлюднений у 2018 році, що робить його одним з останніх відкритих досліджень, так як подальших робіт знайти не вдалося. Незважаючи на те, що цей датасет добре підходить для тренування моделей, очищений та підготовлений, через те що він містить в собі всього дві стандартизовані фрази, сказані професійними акторами, не зважаючи на те, що моделі з його використанням частіше за все дають більшу точність ніж ті, що натреновані на інших наборах даних, цей датасет не дуже підходить до комерційного та особливо медичного застосування, тому що не можна бути точно впевненим в достовірності такого тестування, бо однієї фрази довжиною в 3 секунди буде недостатньо для визначення емоційного стану, тому для тренування цієї моделі було розглянуто ще набір даних IEMOCAP, який був опублікований в 2008 році Університетом Південної Каліфорнії та лабораторією аналізу та інтерпретації сигналів [35].

Цей датасет містить в собі приблизно 12 годин аудіо- та відеозапису, в яких 5 акторів чоловічої статі та 5 акторок жіночої статі протягом 6 сесій

запису проговорюють діалоги з різним емоційним забарвленням, такими як злість, щастя, сум, спокій, розчарування та захоплення. Також цей набір даних містить в собі інформацію про рух тіла одного з учасників кожного діалогу, що в деяких випадках дозволить зробити аналіз точнішим. Відмінною рисою цього набору даних є те, що на відміну від коротких стандартизованих фраз, актори беруть участь у діалогах, які надають більше повне емоційне забарвлення, та є більш інформативними загалом.

Окрім того в цьому наборі даних присутні також імпровізовані діалоги, незаплановані, без чіткої спрямованості, які й будуть найбільш схожі на те, як звичайний користувач буде говорити без підготовки. Цей набір даних також відрізняється тим, що емоція якою в кінцевому результаті буде промаркований діалог або речення, була обрана голосуванням кількох незалежних експертів, що забезпечує більш природний розподіл та вибір емоцій, оснований на виборі реальної людини, а не просто заскриптованої однакової фрази з різним емоційним забарвленням. З огляду на вищевикладену інформацію, для цього проєкту більш доцільно використовувати набір даних IEMOCAP, через те, що більш наближені до реальних умови запису дадуть менш упереджений результат, навіть ціною зниження точності моделі загалом.

Для того щоб натренувати модель потрібно виконати певні кроки, такі як первинна обробка набору даних, вилучення ознак та тренування самої моделі. Процес, що виконується в цій роботі показаний на рисунку 3.1.

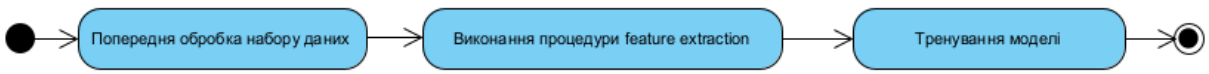


Рисунок 3.1 – Діаграма діяльності для тренування моделі

Створена activity diagram відображає процес, що буде описуватися в наступних підрозділах, першим з яких є попередня обробка набору даних.

3.2 Попередня обробка набору даних

В цій роботі обробка даних включає в себе два етапи, опис набору даних та вилучення мітки емоції для кожного аудіозапису для зберігання її в стандартизованому форматі. Перший етап описаний в підрозділі нижче.

3.2.1 Опис обраного набору даних

Набір даних IEMOCAP, який був обраний для тренування нейронної мережі, розповсюджується в архіві, що містить в собі певну структуру даних. Ця структура на першому рівні складається з папок, кожна з яких репрезентує окрему сесію діалогів. Кожна сесія однакова, тому необхідно та достатньо розібрати лише одну з них. У кожній сесії дані розбиті на діалоги та окремі речення, папка діалогів має в собі транскрипції діалогів, їх записи, інформацію про рух людини, що говорить, мітки емоцій для кожного діалогу в різних форматах. Мітки емоцій для кожного діалогу містяться в окремих файлах з вказанням кожного речення та прогнозовану емоцію для цього речення. Приклад одного речення наведено в лістингу 3.1.

Лістинг 3.1 – Опис емоційного забарвлення речення

```
[14.8872 - 18.0175] Ses01F_impro01_F002 neu  [2.5000,
2.5000, 2.5000]
C-E2:     Neutral;  ()
C-E3:     Surprise; ()
C-E4:     Neutral;  ()
C-F1:     Neutral; Anger;     ()
A-E3:     val 3; act 2; dom  2;    ()
A-E4:     val 2; act 3; dom  3;    (superior, indifferent,
menacing)
A-F1:     val 3; act 3; dom  3;    ()
```

В ньому представлено текстовий опис одного речення з діалогу, який включає такі компоненти:

– [14.8872 – 18.0175] – часові межі речення в діалозі;

– Ses01F_impro01_F002 – назва файлу речення в папці з реченнями;

– [2.5000, 2.5000, 2.5000] – значення валентності, збудженості та домінування в емоції, що виражене в шкалі від 1 до 5, та означає відповідно приємні чи неприємні емоції, рівень емоційного збудження чи спокою та почуття контролю чи безпорадності, в цьому випадку все вказує на нейтральність;

– C-значення – категоріальне значення емоції, які різні анотатори надали певному реченню, значення в дужках означає довільну примітку від кожного анотатора, якщо вона є;

– A-значення – оцінки за шкалою VAD, середні значення для кожної компоненти з якої описані вище, примітки анотатора теж присутні та необов’язкові.

Аналіз кожного з таких речень дозволяє зрозуміти структуру набору даних для проведення подальших операцій.

Папка з окремими реченнями з кожної сесії містить в собі записи кожного речення та інформацію про рух людини.

Окрім того, слід зазначити, що дані в датасеті не збалансовані, що продемонстровано в лістингу 3.2, проте через невелику кількість записів було прийнято рішення не виключати навчальний матеріал для моделі та тренувати на усіх наявних даних.

Лістинг 3.2 – Розподіл емоцій в датасеті

```
Емоції в DataFrame:
emotion_label
fru    1849
neu    1708
ang    1103
```

Продовження лістингу 3.2

```
sad    1084
exc    1041
hap     595
Name: count, dtype: int64
```

Як бачимо з лістингу, емоція щастя представлена найменше, а роздратованість та нейтральна – найбільше. Наступним кроком є виокремлення міток з формату в якому вони знаходяться в датасеті, в формат який підходить для подальшого тренування.

3.2.2 Виокремлення міток емоцій з даних

Для тренування моделі будуть використовуватись не діалоги, а речення з них які будуть розбиватися на частини фіксованої довжини. Після аналізу обох частин кожної сесії було сформовано csv файл зі шляхом до кожного аудіофайлу з реченням та емоцією до нього яку визначили експерти. Для цього доцільно написати скрипт, логіку та частини якого наведено нижче.

На додаток до вищезгаданого, для коректного виокремлення текстових значень емоцій, скануються всі файли з розширенням *.txt для кожної сесії в діалогах. Це дозволяє отримати повний доступ до текстових даних, які містять мітки емоцій. Приклад частини такого файлу наведено в лістингу 3.1, що демонструє формат даних, з якими працює система.

За допомогою RegEx – регулярного виразу, який допомагає знайти рядки в тексті за заданим шаблоном, відокремлюються назва файлу речення в папці з реченнями та мітка емоцій. Ці мітки та назви пояснені вище. Для тренування цієї моделі залишаються тільки 6 основних емоцій, а ті, для яких анотатори не визначились остаточно, не приймаються.

Частину скрипту для вилучення емоцій з файлів наведено в лістингу 3.2.

Лістинг 3.2 – Пошук по кожному файлу анотацій зі збереженням

```
def parse_iemocap_annotation_file(filepath):
    utterances_info = []
    try:
        with open(filepath, 'r', encoding='utf-8') as f:
            for line in f:
                match =
annotation_line_pattern.search(line)
                if match:
                    turn_name = match.group(1)
                    emotion_label = match.group(2)
                    utterances_info.append((turn_name,
emotion_label))
    except Exception as e:
        print(f"Помилка при читанні файлу {filepath}:
{e}")
        return []
    return utterances_info
```

В результаті роботи всього скрипту, в зазначеній директорії формується CSV файл один запис якого складається з:

– повного шляху до файлу;
– відносного шляху до файлу;
– емоції речення з файлу.

На цьому етап попередньої обробки даних завершено. Наступним кроком з діаграми діяльності є процедура feature extraction, що виконується в наступному розділі.

3.2 Вилучення ознак з аудіозаписів

Для процедури feature extraction застосовується три підходи. Перший це застосування моделі wav2vec2-large-960h. Робота моделей класу wav2vec2 детально описана в підрозділі 1.3. Для вилучення ознак з

аудіозаписів модель переводиться в inference mode, тобто поведінка, яка використовується під час навчання, вимикається, що дозволяє зупинитися на контекстуальному векторі ознак. Для проведення feature extraction аудіозапис за допомогою бібліотеки librosa завантажується до скрипту, з частотою 16кГц, що є стандартом для wav2vec2, та доповнюється нулями, щоб усі аудіозаписи були однакової довжини. На результаті цих перетворень застосовується wav2vec2-large-960h, та результат у вигляді контекстуального вектору зберігається в файл з розширенням .*npy у вигляді масиву. На виході отримується вектор довжиною 512 ознак.

Наступним підходом є застосування згорткової нейронної мережі, застосування якої також описана в підрозділі 1.3, в цьому алгоритмі початкові дії ідентичні – файли завантажуються в python, заповнюються нулями або обрізаються, після чого перетворюються у логарифмічні mel-спектрограми, що подібні до того, як чує людина, за процедурою що описана в підрозділі 1.3 та передаються до CNN feature extractor, архітектура якого наведена на рисунку 3.2.

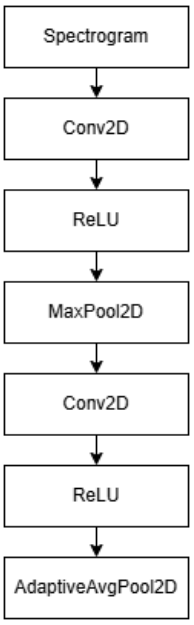


Рисунок 3.2 – Послідовність дій CNN feature extractor

Після цього отримані вектори передаються до лінійного шару для збільшення розмірності з 32 до 512, задля того, щоб довжина кожного вектору співпадала з довжиною з Wav2Vec2.

Останнім підходом вилучення ознак, що буде використовуватись, є екстрактор на основі LSTM. Початкова процедура завантаження даних ідентична з CNN, проте аудіофайли перетворюються не на mel-спектрограми, а на mel-частотні кепстральні коефіцієнти, кожен вектор довжиною 40 значень, що подаються на вхід до LSTM. В цьому процесі використовується двонаправлена LSTM, що складається з двох моделей, перша приймає вхідний вектор у його незмінному вигляді, а друга приймає його починаючи з кінця, що дозволяє аналізувати дані з двох боків для більшої інформативності, після чого виходи двох LSTM поєднуються та перетворюються лінійним шаром з розмірності 256 (128 на кожній LSTM) до розмірності шириною 512.

На цьому етапі отримано вектори ознак трьома методами, в наступному підрозділі описується останній крок з діаграми діяльності – тренування моделі.

3.3 Тренування моделей

Для кожного вилученого набору ознак пропонується натренувати модель однакової архітектури для порівняння ефективності. До набору ознак з кожного feature extractor застосовується бібліотека Optuna способом, який було пояснено в підрозділі 1.3. Після отримання оптимальних гіперпараметрів проводиться агументація, після якого частини в векторах ознак маскуються. Після підготовки тренується ансамбль з 5 моделей, кожна з яких складається з об’єднання CNN та Transformer для більш глибокого аналізу вхідних ознак. Схема роботи моделі класифікації представлена на рисунку 3.3.

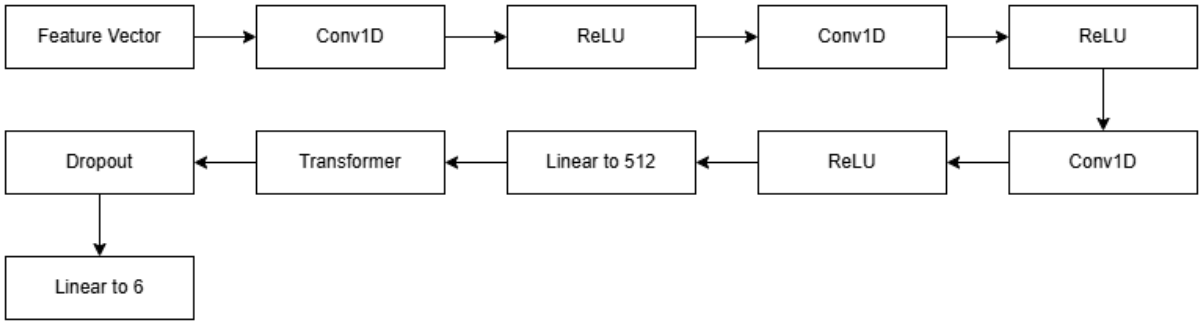


Рисунок 3.3 – Схема роботи класифікаційної моделі

В результаті було отримано 5 моделей які були об’єднані в ансамбль та матриці помилок для кожного з ансамблів. Матриця помилок моделі на Wav2Vec2 продемонстрована на рисунку 3.4.

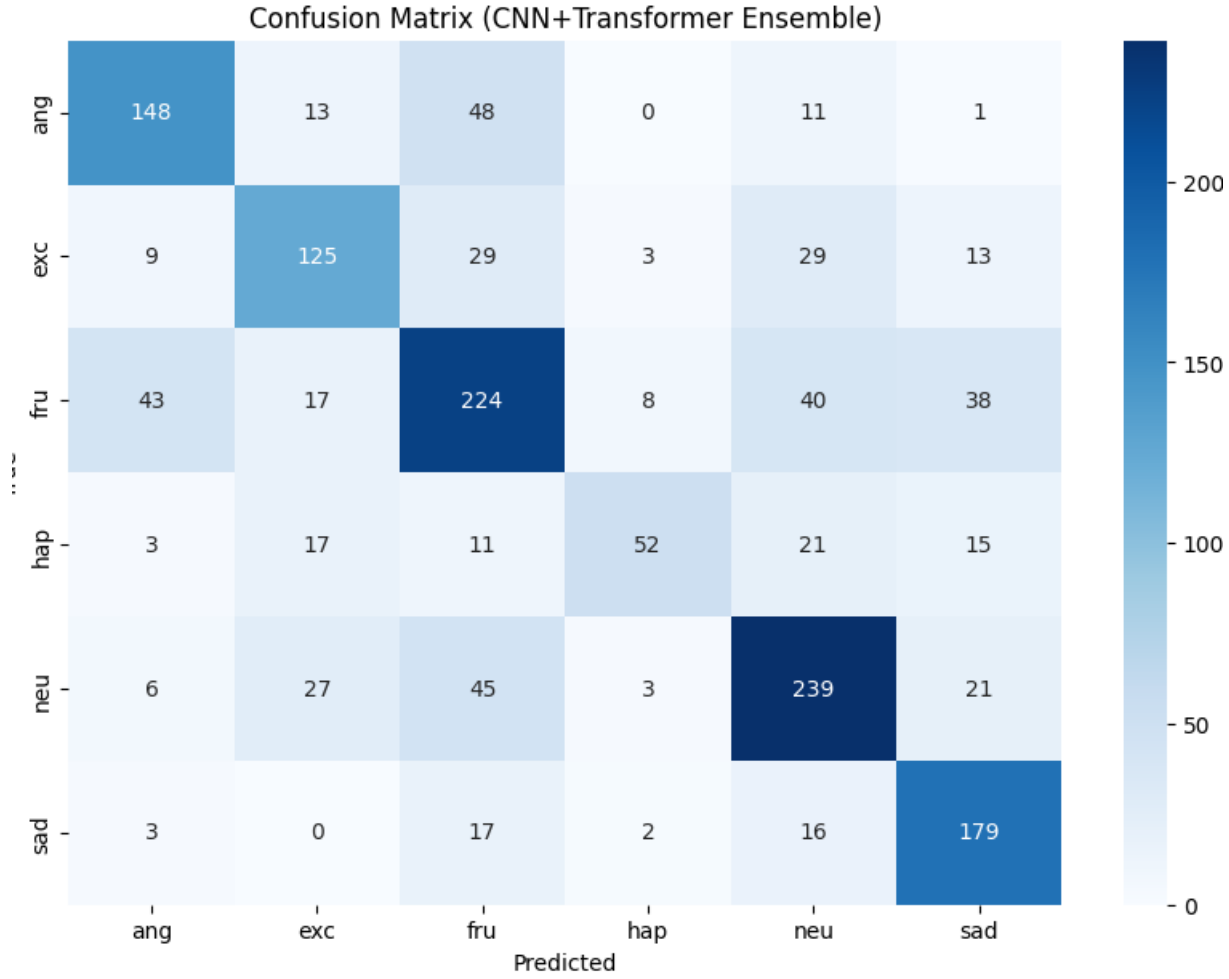


Рисунок 3.4 – Результати Wav2Vec2 ознак

З цієї матриці видно, що найбільше помилок допущено в класифікації angry та frustrated, що для застосунку, що проектується, можна вважати допустимим, бо система все одно скерує до спеціаліста. Звіт класифікації продемонстровано на рисунку 3.5.

Classification Report:

	precision	recall	f1-score	support
ang	0.70	0.67	0.68	221
exc	0.63	0.60	0.61	208
fru	0.60	0.61	0.60	370
hap	0.76	0.44	0.56	119
neu	0.67	0.70	0.69	341
sad	0.67	0.82	0.74	217
accuracy			0.66	1476
macro avg	0.67	0.64	0.65	1476
weighted avg	0.66	0.66	0.65	1476

Рисунок 3.5 – Звіт класифікації Wav2Vec ознак

Наступною моделлю є модель на ознаках CNN, результат якої представляє рисунок 3.6.

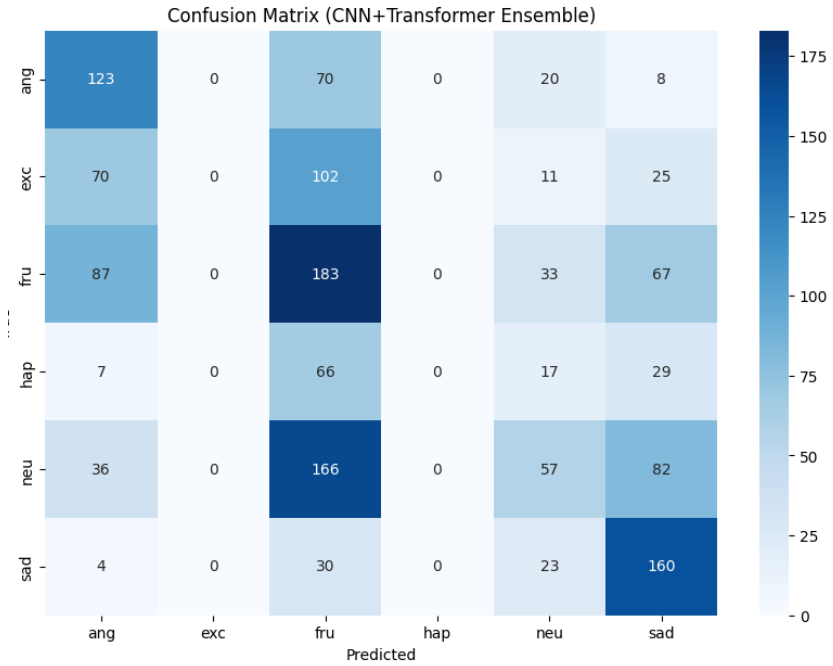


Рисунок 3.6 – Результат CNN ознак

З цієї діаграми наочно можемо бачити недолік архітектури, що не враховує весь контекст, та міру впливу цього фактору коли тренована модель складається з неперервних даних, особливо аудіо. Останньою натренованою моделлю є та, що використовує ознаки LSTM, матрицю помилок цієї моделі наведено на рисунку 3.7.

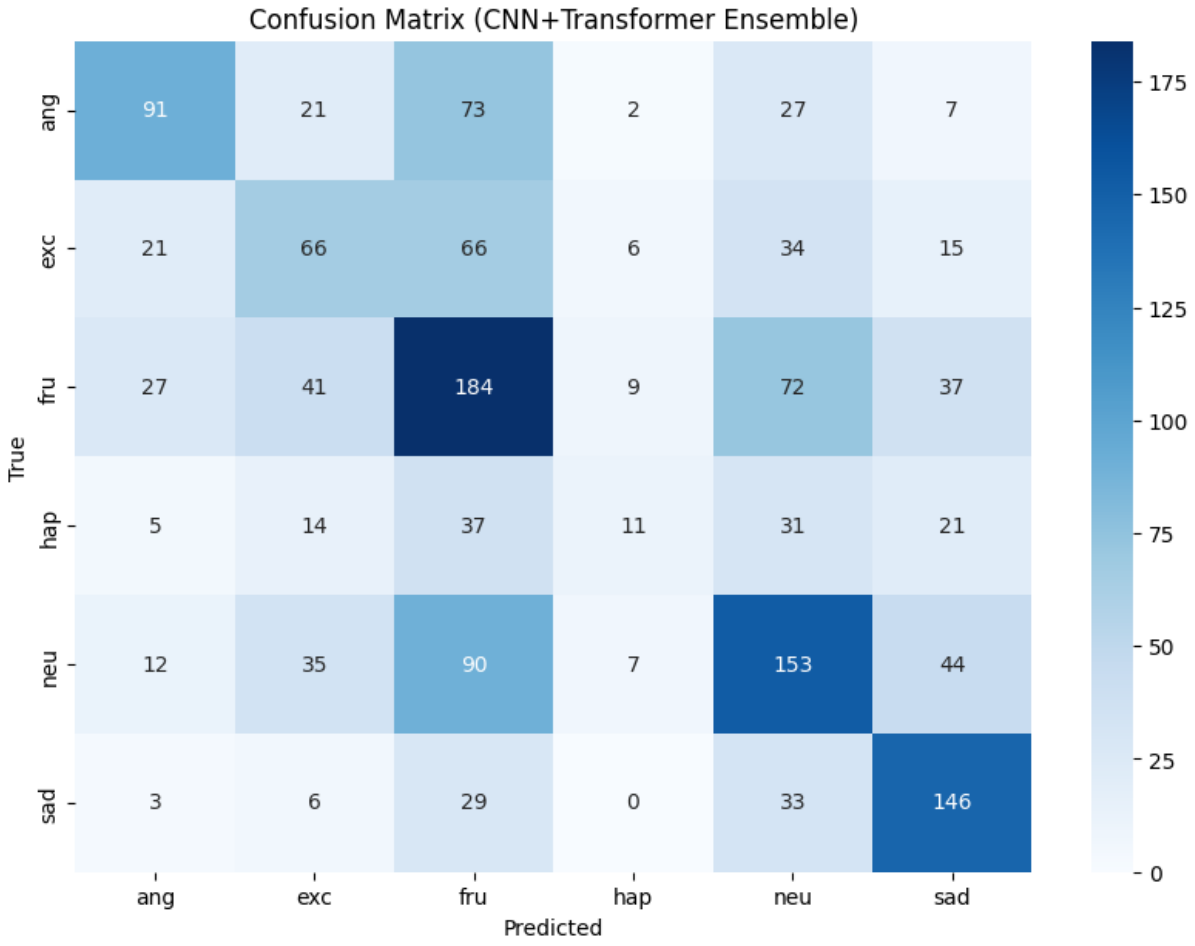


Рисунок 3.7 – Результати моделі на ознаках LSTM

Результати цієї моделі покращились порівняно з CNN, що обумовлено ефектом «пам'яті» LSTM, що важливо в контексті послідовностей, у вигляді яких представлені аудіодані, проте результати нижче за рівень Wav2Vec2, яка використовує трансформери, що дозволяє «бачити» усі ознаки одночасно, що виявилось важливою складовою для feature extraction.

Міри точності всіх моделей представлені в таблиці 3.1.

Таблиця 3.1 – Точності натренованих моделей

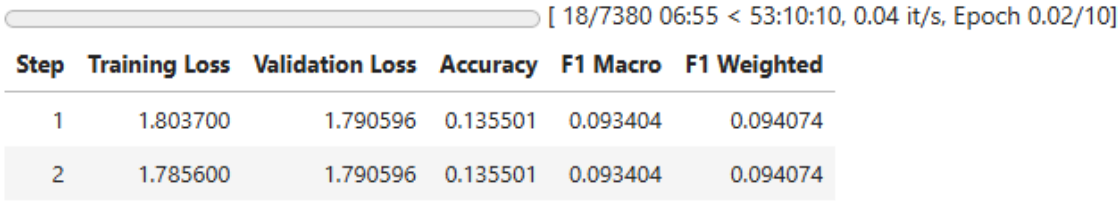
Точності натренованих моделей	Рівень точності
	%
CNN	35
LSTM	44
Wav2Vec2	68

Отриманий рівень точності в цьому проєкті співставний з моделями, що презентовані в інших дослідженнях, наприклад як показано на рисунку 3.8, точність Wav2Vec2 PreTrained приблизно однакова.

Pretrained model	Features	Dataset	
		IEMOCAP	RAVDESS
None	eGeMAPS	52.4 ± 0.1	57.0 ± 2.4
	Spectrogram	49.8 ± 1.0	44.5 ± 0.8
Wav2vec2-PT	Local enc.	60.3 ± 0.7	65.4 ± 1.7
	Cont. enc.	58.5 ± 0.6	69.0 ± 0.2
	All layers	**67.2 ± 0.7**	**84.3 ± 1.7**
Wav2vec2-FT	Local enc.	57.3 ± 1.0	58.8 ± 2.7
	Cont. enc.	44.6 ± 1.0	37.5 ± 3.0
	All layers	63.8 ± 0.3	68.7 ± 0.9

Рисунок 3.8 – Результати тренування моделей з дослідження [8]

Також в вищезгаданому дослідженні була використана модель дотренована (FineTuned) на невеликому наборі даних спеціально під задачу дослідження. Для feature extraction було використано wav2vec2-large-960h, а дотренування навіть її меншої версії wav2vec2-base на датсеті IEMOCAP займає приблизно 50 годин, що продемонстровано на рисунку 3.9 та є неможливим з використанням Kaggle через встановлені обмеження за часом прискорювача, що використовувався.



Step	Training Loss	Validation Loss	Accuracy	F1 Macro	F1 Weighted
1	1.803700	1.790596	0.135501	0.093404	0.094074
2	1.785600	1.790596	0.135501	0.093404	0.094074

Рисунок 3.9 – Процес fine-tuning для wav2vec2-base

Загалом результат отриманий після тренування цієї моделі є досить високим з урахуванням технічних обмежень, покращення результату теоретично можливе з використанням Fine Tuned Wav2Vec2 моделі або іншої моделі, що вийшли нещодавно. Також в дослідженні [8] використовувався набір даних RAVDESS, та точність моделі на його основі склала 85%, але його не було використано для тренування через причини зазначені в підрозділі 3.1.1.

3.4 Висновки за розділом

В цьому розділі було проведено тренування моделі штучного інтелекту, що складається з декількох кроків. Вибір набору даних було проведено після аналізу доступних наборів запису голосу, зокрема детально були розглянуті RAVDESS та IEMOCAP, та було обґрунтовано рішення на користь другого. Після вибору набору даних було проведено аналіз структури, в якій він розповсюджується, збалансованості класів, що він містить, та обґрунтування, чому частина даних не була вилучена. Було проведено вилучення міток емоцій з даних та створено файл формату .csv, що включає в себе шлях до аудіофайлу та емоції, яку він містить. Після цього було проведено процедуру feature extraction трьома методами, а саме за допомогою CNN, LSTM та Wav2Vec2. Результати з кожного методу було використано для навчення моделі та порівняно результати трьох моделей що були натреновані. Також їх було порівняно з іншими роботами з цієї

галузі, що свідчить про високу якість натренованої моделі. Це підтверджується тим, що точність, яку було досягнуто, не нижча за інші дослідження. Також був проведений аналіз можливості покращення існуючої моделі, шляхом застосування більш досконалого способу feature extraction, та пояснені причини, через які це не може виконано в цій роботі. Після тренування моделі та досягнення оптимального результату останнім кроком для завершення розробки застосунку є розробка клієнтської частини для взаємодії користувачів з додатком. Цей крок буде описано в останньому розділі кваліфікаційної роботи.

4 РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ

В рамках роботи однією з задач є розробка користувальницького інтерфейсу, що буде слугувати способом, за допомогою якого користувачі отримують послуги та комунікують із системою.

4.1 Опис візуального інтерфейсу

Візуальний інтерфейс був розроблений на основі фреймворку React.JS та складається з компонентів, кожен з яких представляє певну сторінку інтерфейсу. Особлива увага в розробці frontend частини приділяється дотриманню безпеки кожної сторінки шляхом контролю доступу до них, що виконується шляхом перевірки ролі з JWT, який сервер повертає при вході в систему, як було описано в підрозділі 2.3.3. Для забезпечення цього механізму та механізму включення токену в заголовок запиту використовується власний екземпляр бібліотеки axios, яка дозволяє відправляти запити до сервера. Власний екземпляр цієї сутності було налаштовано таким чином, що при отриманні з сервера відповіді з кодом помилки 403 Forbidden, тобто відказано в доступі, користувача перенаправляє на головну сторінку, а при отриманні коду помилки 401 Unauthorized, коли користувач не надає JWT в запиті, його перенаправляє на сторінку входу. Цей механізм наведено в лістингу 4.1.

Лістинг 4.1 – Механізм захисту запитів

```
error => {
        const status = error.response?.status;

        if (status === 401) {
            localStorage.removeItem('token');
            history.push('/login');
        } else if (status === 403) {
```

Продовження лістингу 4.1

```
            history.push('/main');
        }
        return Promise.reject(error);
    }
);
```

Також з JWT, що надає сервер, надходить інформація про роль користувача у системі: пацієнт, лікар або адміністратор, та на основі цього реалізовано механізм умовного рендерінгу для відображення відповідного інтерфейсу. В залежності від ролі, отриманої з токена, користувачу демонструється сторінка, що відповідає його повноваженням та задачам у системі. Цей процес на прикладі головної сторінки продемонстрований у лістингу 4.2

Лістинг 4.2 – Відображення сторінки в залежності від ролі

```
const roles = getUserRoles();
if (roles.includes('PATIENT')) {
    return (<PatientMain/>)
}
if (roles.includes('DOCTOR')) {
    return (<DoctorMain/)
}
if (roles.includes('ADMIN')) {
    return (<AdminMain/>)
}
```

Такий підхід створює персоналізований користувацький досвід, де кожен бачить лише релевантну для нього інформацію та функціональність. Наочні приклади таких спеціалізованих головних сторінок для пацієнта та лікаря наведені на рисунках 4.1 та 4.2 відповідно.

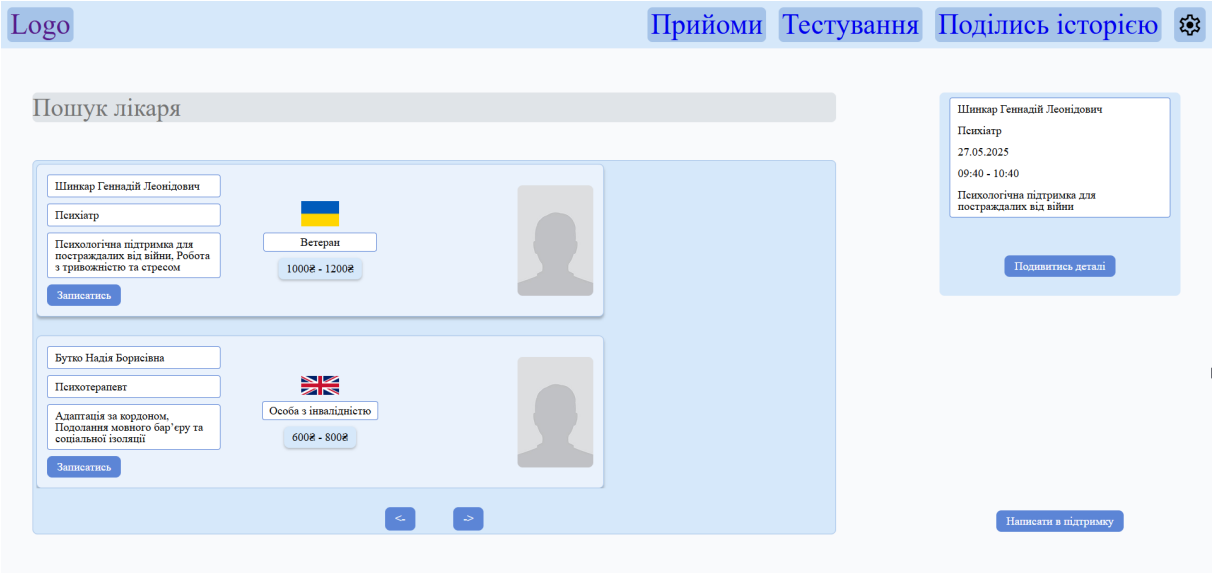


Рисунок 4.1 – Головна сторінка пацієнта

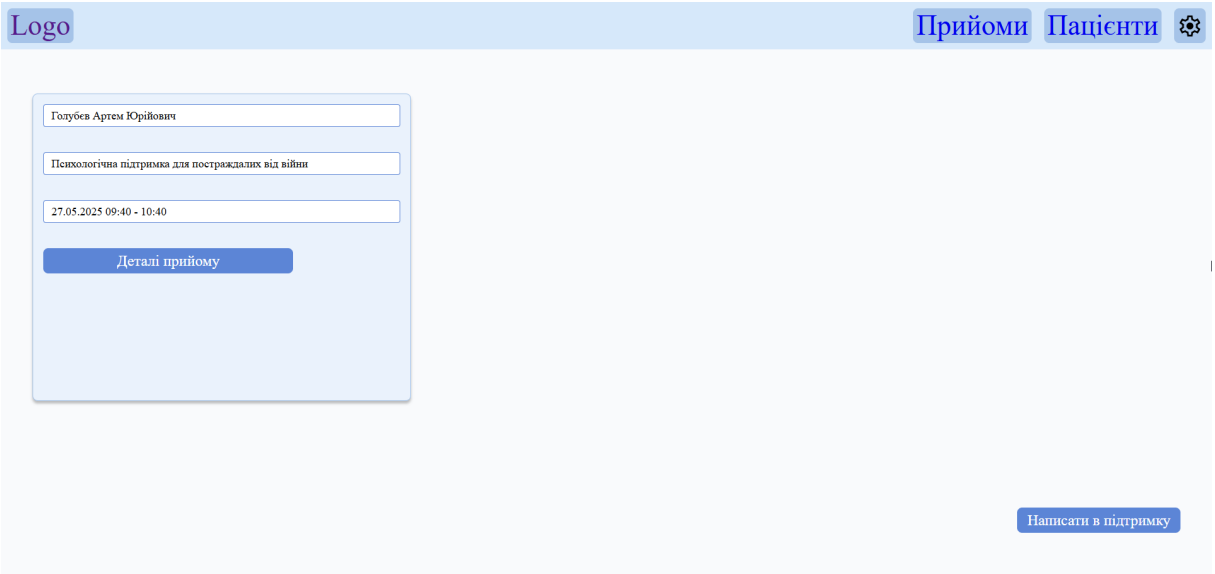


Рисунок 4.2 – Головна сторінка лікаря

Одним з ключових компонентів розробленого програмного застосунку є модуль інтелектуального тестування, що доступний для пацієнтів. Цей інструмент збирає даних, необхідні для подальшого аналізу психоемоційного стану. Процес тестування складається з кількох етапів, а його початковою та частиною є запис голосу пацієнта. Ця екранна форма, представлена на рисунку 4.3.

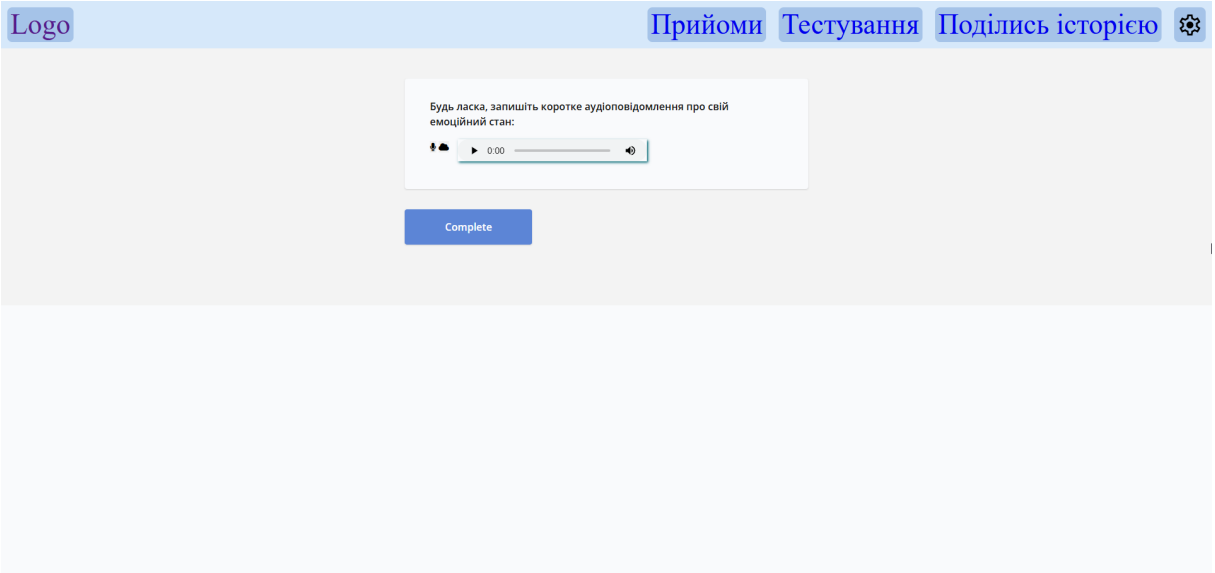


Рисунок 4.3 – Інтерфейс запису голосу для тестування

Після запису фрагменту голосу він відправляється на сервер та оброблюється за допомогою моделі, після чого модуль повертає значення ймовірності емоцій у вигляді, що наведений у лістингу 4.3.

Лістинг 4.3 – Результат психологічного тестування

```
{
  "ang": 0.00012796473400100692,
  "exc": 0.0006043648379463829,
  "fru": 0.008876536406506085,
  "hap": 0.00022653714144552873,
  "neu": 0.03192611891334602,
  "sad": 0.958238477966755
}
```

В даному випадку модель надає емоції «сум» з ймовірністю 96%, проте якщо різниця між двома емоціями з найбільшою ймовірністю складає менше 10 відсотків – як результат надаються обидві емоції, але та, яка йде другою, отримає статус додаткової та пацієнту пропонується звернути на це увагу. Приклад результатів тестування наведений на рисунку 4.4.

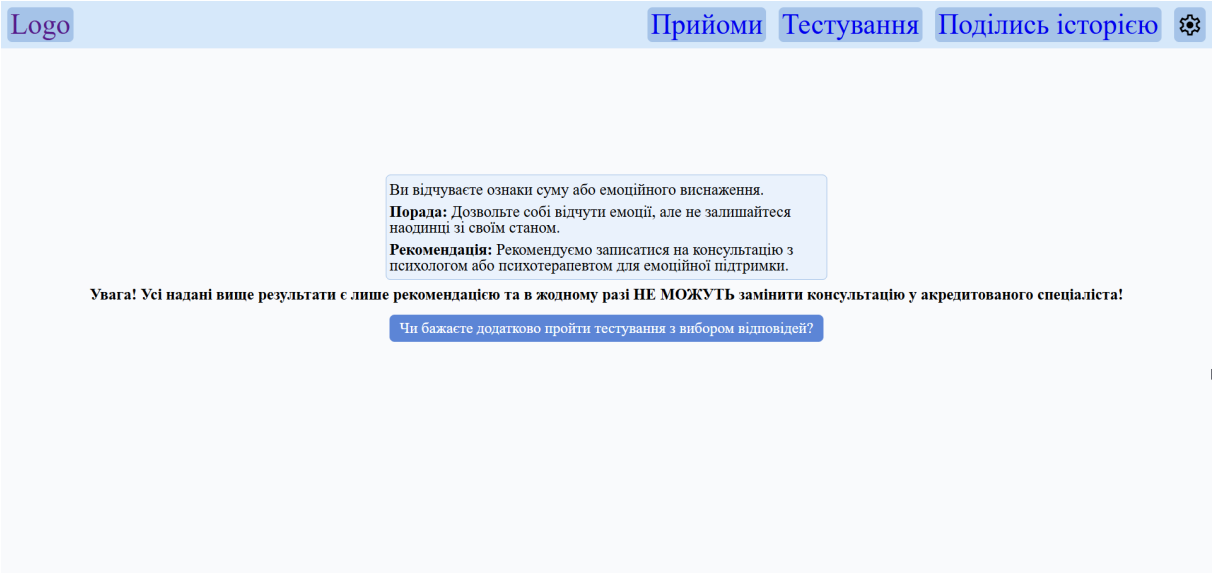


Рисунок 4.4 – Результати тестування за голосом

Окрім того, пацієнту пропонується пройти традиційний вид тестування, з причин які були описані в другому розділі, так само як і формула отримання фінального результату. Приклад запитання на цьому тесті наведений на рисунку 4.5.

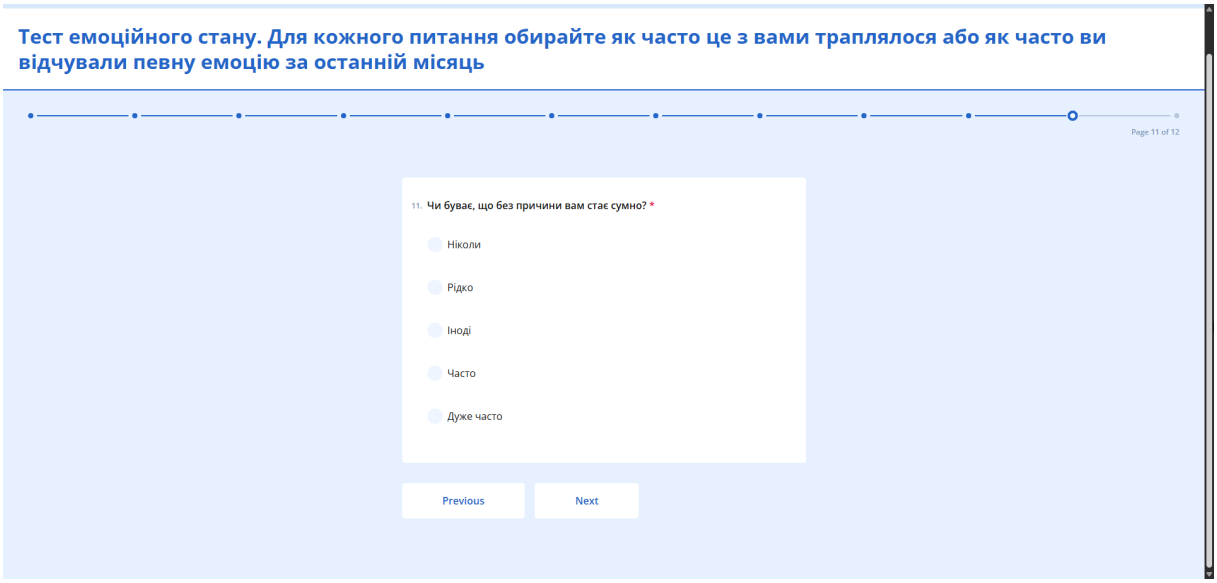


Рисунок 4.5 – Процес проходження традиційного тестування

Після комбінації двох видів тестування отримано результат, що наведений в лістингу 4.4.

Лістинг 4.4 – Результат комбінованого тестування

```
{
  "exc": 0.060423055386562466,
  "hap": 0.036158575999011865,
  "sad": 0.7427669345767284,
  "fru": 0.06621357548455425,
  "ang": 0.0360895753138007,
  "neu": 0.058348283239342214
}
```

З представленого результату видно, що ймовірність емоції «сум», що переважала до комбінації результатів тепер трохи знизилась, що більше відповідає дійсності, бо емоційний спектр людини переважно не складається з однієї емоції та завжди більш багатогранний. Загалом інтеграція моделі в застосунок та використання її за допомогою візуального інтерфейсу пройшли вдало, усі задачі були виконані.

4.2 Висновки до розділу

В цьому розділі було описано розробку користувальницького інтерфейсу. Основну увагу приділено безпеці клієнтської частини, що досягається за допомогою використання JWT токену та власного екземпляру axios, налаштованого для запитів до серверної частини. Було продемонстровано головні сторінки для пацієнтів та лікарів, проходження комплексного тестування, а саме процес запису голосу, отримання результатів та проходження розширеного тесту. Окремі, доволі стандартні, функції застосунку не було показано в деталях, щоб не перевантажувати пояснювальну записку. Але з ними можна ознайомитися в репозиторії.

ВИСНОВКИ

В результаті даної кваліфікаційної роботи було проведено аналіз предметної галузі, актуальність якої визначила напрям досліджень. Проаналізовано існуючі рішення для обраної предметної галузі.

Спроектовано та розроблено веб-застосунок для психологічного тестування з використанням штучного інтелекту. Для реалізації ключової функціональної компоненти застосунку, було натреновано модель нейронної мережі. В якості частини, що пов'язує користувача з даною компонентою, виступає класичний веб-застосунок, спроектований модульним чином для подальшої еволюції при необхідності. Виходячи із специфіки задач, що вирішуються, два класи СУБД було використано. Моделювання в ході проектування виконувалося з використанням сучасних нотацій та засобів.

У порівнянні з аналогічними системами ця розробка використовує більш інноваційні методи психологічного тестування, що дозволяє розширити доступність використання для різних категорій користувачів, окрім того цей застосунок враховує контекст українського суспільства.

В частині розробки нейронної мережі було досягнуто результатів, що за точністю на даний час не поступаються іншим моделям, що виконують аналогічну задачу. Нейронні мережі такого типу для вирішення поставленої задачі раніше не були впроваджені в проаналізованих застосунках. В подальшому для вдосконалення роботи застосунку може використовуватися розбивання застосунку з монолітної на мікросервісну архітектуру, що дозволить підвищити відмовостійкість та швидкодію роботи застосунку. Окрім того для вдосконалення нейронної мережі, за дотримання конфіденційності є можливим збір набору даних, який буде містити записи голосу українською мовою задля дотренування моделі. В подальшому доцільно розгортати застосунок на ринки різних країн,

підтримка різних мов та тренування моделей нейронної мережі для кращого розпізнавання різних мов.

Дана кваліфікаційна робота містить пояснення та порівняння різних моделей нейронних мереж в якості feature extractor для аналізу аудіо, що може використовуватись як навчальний матеріал.

Усі задачі що ставились при плануванні роботи були виконані в повному обсязі, з доведеним позитивним результатом за кожним пунктом.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) Mental health in the 21st century – the center for brain, mind and society. *The Center for Brain, Mind and Society*. URL: https://brainmindsociety.org/posts/mental-health-in-the-21st-century (date of access: 24.04.2025).

2) Increases in poor mental health, mental distress, and depression symptoms among U.S. adults, 1993--2020 / N. S. Udupa et al. *Journal of mood and anxiety disorders*. 2023. Vol. 2. P. 100013. URL: https://doi.org/10.1016/j.xjmad.2023.100013.

3) *Інститут поведінкових досліджень*. URL: https://ibs.auk.edu.ua/uploads/files/shares/66d9600c2b658.pdf (дата звернення: 24.04.2025).

4) Swargiary K. Integrating artificial intelligence in branches of psychology. Scholars' Press, 2024. 533 p. URL: https://www.researchgate.net/publication/385812794_Integrating_Artificial_Intelligence_in_Branches_of_Psychology (date of access: 24.04.2025).

5) Emotion in the human face. Elsevier, 1972. URL: https://doi.org/10.1016/c2013-0-02458-9 (date of access: 24.04.2025).

6) Does training improve the detection of deception? A meta-analysis / V. Hauch et al. *Communication research*. 2014. Vol. 43, no. 3. P. 283–343. URL: https://doi.org/10.1177/0093650214534974 (date of access: 24.04.2025).

7) Sociable machines - Kismet, the robot. *Home Page | MIT CSAIL*. URL: http://www.ai.mit.edu/projects/sociable/baby-bits.html (date of access: 24.04.2025).

8) Emotion recognition from speech using wav2vec 2.0 embeddings. *arXiv.org*. URL: https://arxiv.org/abs/2104.03502 (date of access: 24.04.2025).

9) Програма 7П | центр "коло сім'ї". *Центр здоров'я та розвитку "Коло сім'ї"*. URL: https://k-s.org.ua/resources/7p/ (дата звернення: 24.04.2025).

10) Українці відчувають стрес і тривогу, але обирають конструктивні копінгові стратегії реагування на ці стани. *Програма ментального здоров'я | Ти як?*. URL: https://howareu.com/news/ukraintsi-vidchuvaiut-stres-i-tryvohu-ale-obyraiut-konstruktyvni-kopinhovi-stratehii-reahuvannia-na-tsi-stany (дата звернення: 24.04.2025).

11) Аналітичний звіт чому люди уникають психологічної допомоги бар'єри та стереотипи. *American University Kyiv*. URL: https://er.auk.edu.ua/server/api/core/bitstreams/b5ed4503-b76f-4947-9e40-718c3b6a6e41/content (дата звернення: 24.04.2025).

12) Vries A. D. The growing energy footprint of artificial intelligence. *Joule*. 2023. URL: https://doi.org/10.1016/j.joule.2023.09.004.

13) Artificial intelligence market size, share | industry report, 2030. *Market Research Reports & Consulting | Grand View Research, Inc.* URL: https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-market (date of access: 24.04.2025).

14) Speech emotion recognition using machine learning / A. K. Saw et al. *International journal of health sciences*. 2022. URL: https://doi.org/10.53730/ijhs.v6ns1.8662 (date of access: 30.04.2025).

15) Vaj T. Spectral features of speech signals: unveiling vocal characteristics. *Medium*. URL: https://vtiya.medium.com/spectral-features-of-speech-signals-unveiling-vocal-characteristics-ee2754a3d0c6 (date of access: 28.05.2025).

16) 3.11. zero-crossing rate – introduction to speech processing. *Introduction to Speech Processing – Introduction to Speech Processing*. URL: https://speechprocessingbook.aalto.fi/Representations/Zero-crossing_rate.html (date of access: 22.05.2025).

17) Energy. *musicinformationretrieval.com | Instructional notebooks on music information retrieval.* URL: https://musicinformationretrieval.com/energy.html (date of access: 28.05.2025).

18) Schmitt M., Ringeval F., Schuller B. At the border of acoustics and linguistics: bag-of-audio-words for the recognition of emotions in speech. *Interspeech 2016*. 2016. URL: https://doi.org/10.21437/interspeech.2016-1124 (date of access: 28.05.2025).

19) About openSMILE – openSMILE Documentation. *audeering GitHub Pages*. URL: https://audeering.github.io/opensmile/about.html (date of access: 22.05.2025).

20) Antoniadis P. Neural networks: difference between conv and FC layers. *Baeldung*. URL: https://www.baeldung.com/cs/neural-networks-conv-fc-layers (date of access: 28.05.2025).

21) What is an LSTM neural network?. *dida Machine Learning*. URL: https://dida.do/what-is-an-lstm-neural-network (date of access: 28.05.2025).

22) Vision transformer architecture and applications in digital health: a tutorial and survey / K. Al-hammuri et al. *Visual computing for industry, biomedicine, and art*. 2023. Vol. 6, no. 1. URL: https://doi.org/10.1186/s42492-023-00140-9 (date of access: 28.05.2025).

23) Home. *C4 model*. URL: https://c4model.com/ (date of access: 08.05.2025).

24) 10 best programming languages for software development in 2024. *Software Development Company | Netguru*. URL: https://www.netguru.com/blog/best-programming-language-for-software-development (date of access: 08.05.2025).

25) Top 7 databases for web applications development of the year. *Softude - Software Product Development & Solutions Company*.

URL: https://www.softude.com/blog/top-databases-for-developing-web-applications-of-the-year (date of access: 08.05.2025).

26) Top 7 nosql databases: which is right for you?. *Jelvix*. URL: https://jelvix.com/blog/top-7-nosql-databases (date of access: 08.05.2025).

27) Team ThoughtSpot. Conceptual vs logical vs physical data models. *ThoughtSpot*. URL: https://www.thoughtspot.com/data-trends/data-modeling/conceptual-vs-logical-vs-physical-data-models (date of access: 14.05.2025).

28) WebRTC. *WebRTC*. URL: https://webrtc.org/ (date of access: 18.05.2025).

29) Getting started with peer connections | WebRTC. *WebRTC*. URL: https://webrtc.org/getting-started/peer-connections (date of access: 18.05.2025).

30) ICE - MDN web docs glossary: definitions of web-related terms | MDN. *MDN Web Docs*. URL: https://developer.mozilla.org/en-US/docs/Glossary/ICE (date of access: 18.05.2025).

31) Jones M., Bradley J., Sakimura N. JSON web token (JWT). RFC Editor, 2015. URL: https://doi.org/10.17487/rfc7519 (date of access: 19.05.2025).

32) Architecture :: spring security. *Spring | Home*. URL: https://docs.spring.io/spring-security/reference/servlet/architecture.html (date of access: 28.05.2025).

33) Lesson: annotations (the java™ tutorials > learning the java language). *Moved*.
URL: https://docs.oracle.com/javase/tutorial/java/annotations/ (date of access: 19.05.2025).

34) Livingstone S. R., Russo F. A. The ryerson audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in north american english. *Plos one*. 2018. Vol. 13, no. 5.

P. e0196391. URL: https://doi.org/10.1371/journal.pone.0196391 (date of access: 21.05.2025).

35) IEMOCAP: interactive emotional dyadic motion capture database / C. Busso et al. *Language resources and evaluation*. 2008. Vol. 42, no. 4. P. 335–359. URL: https://doi.org/10.1007/s10579-008-9076-6 (date of access: 21.05.2025).