

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

Горохівська Людмила Михайлівна

Програмна система попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача / Software System for Preliminary Assessment of the Risk of Psychoemotional Disorders Based on User Digital Markers

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи КН-43
Л. М. Горохівська

Науковий керівник:
к.т.н., доцент, О.Р. Осолінський

Кваліфікаційну роботу допущено до
захисту

«___» _____ 2026 р.

Завідувач кафедри

_____ Н.В. Дзюбановська

Тернопіль – 2026

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

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Н.В. Дзюбановська

«_____» _____ 202 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ
ГОРОХІВСЬКІЙ Людмилі Михайлівні
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програмна система попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача / Software System for Preliminary Assessment of the Risk of Psychoemotional Disorders Based on User Digital Markers

керівник роботи О.Р. Осолінський, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджений наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

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

4. Основні питання, які потрібно розробити:

– предметної області та підходів до моніторингу ризику депресії та тривожності;

– проаналізувати огляди на депресію і тривожність як об'єкти моніторингу та цифрові маркери активності, сну та голосу;

– провести огляд і аналіз існуючих рішень;

– сформулювати постановку задачі;

– розробити загальну структуру проектованої системи;

– розробити алгоритмічне забезпечення системи;

– розробити інформаційне забезпечення;

– провести вибір та аналіз програмно-апаратних засобів для модуля моніторингу;

– реалізувати програмно-технологічне забезпечення модуля;

– провести тестування модуля.

5. Перелік графічного матеріалу в роботі:

– розширена архітектура модуля;

– діаграма послідовності програмного модуля.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студентка _____ Л. М. Горохівська _____
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ О.Р. Осолінський _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи: 78 с., 24 рис., 3 таблиці, 34 джерела, 4 додатки.

Метою роботи є розробка програмної системи попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача.

В роботі застосовано методи системного аналізу, логічного структурування, алгоритмічного моделювання, машинного навчання, обробки числових і аудіоданих, аналізу голосових сигналів та проектування інформаційних потоків вебзастосунку. Для програмної реалізації використано мову програмування Python, PHP, JavaScript, HTML і CSS, а також бібліотеки NumPy, pandas, CatBoost, joblib, librosa та Chart.js.

Запропоновано структуру програмної системи, що включає вебінтерфейс для лікаря, модуль керування профілями пацієнтів, підсистему завантаження та вибору активних файлів, серверну логіку з API-ендпоінтами, модуль аналізу структурованих фізіологічних і поведінкових даних, окремий голосовий модуль, підсистему візуалізації результатів і засоби відображення історичних змін у часі.

Розроблено програмну систему, яка дозволяє створювати та редагувати профілі пацієнтів, завантажувати табличні файли щоденних вимірювань і WAV-аудіофайли, обирати активний файл для аналізу, виконувати обчислення середніх значень, похідних індикаторів і часових тенденцій, а також формувати прогнози ймовірності класів «норма», «тривога», «депресія» та «ПТСР». Для структурованих даних використано багатокласову модель CatBoost, а для голосового каналу — виділення ознак MFCC, RMS та ZCR з подальшим класифікаційним оцінюванням.

Проведено тестування основних функцій програмної системи. Перевірено створення профілю пацієнта, завантаження, вибір, видалення і вивантаження файлів, відображення таблиці щоденних даних, обчислення середніх значень та похідних індикаторів, запуск прогнозування, відображення ймовірностей і пояснення, аналіз історичних змін у часі, а також окремий сценарій голосового аналізу з відтворенням аудіо та обробкою помилкових дій.

Ключові слова: ПСИХОЕМОЦІЙНІ ПОРУШЕННЯ, ЦИФРОВІ МАРКЕРИ, ЦИФРОВИЙ МОНІТОРИНГ, ВЕБЗАСТОСУНОК, АКТИВНІСТЬ, СОН, ГОЛОСОВІ ХАРАКТЕРИСТИКИ, САТВОOST, МАШИННЕ НАВЧАННЯ.

ANNOTATION

Explanatory note to the qualification thesis: 78 pages, 24 figures, 3 tables, 34 references, 4 appendices.

The aim of the work is to develop a software system for preliminary assessment of the risk of psychoemotional disorders based on user digital markers.

The work applies methods of system analysis, logical structuring, algorithmic modelling, machine learning, numerical and audio data processing, voice signal analysis, and design of information flows for a web application. The implementation uses Python, PHP, JavaScript, HTML and CSS together with NumPy, pandas, CatBoost, joblib, librosa, and Chart.js.

A structure of the software system is proposed, including a doctor-oriented web interface, patient profile management, upload and active-file selection workflows, backend API logic, a structured-data analysis module, a separate voice-analysis module, probability visualisation, and historical trend presentation.

A software system has been developed that supports creating and editing patient profiles, uploading structured daily measurement files and WAV recordings, selecting an active file for analysis, computing averages, derived indicators and temporal trends, and generating class probabilities for normal state, anxiety, depression, and PTSD. The structured-data branch uses a multiclass CatBoost model, while the voice branch uses MFCC, RMS and ZCR features followed by classifier-based estimation.

Testing of the main functions of the software system was carried out. The workflows of patient profile creation, file upload, active-file selection, file deletion and download, daily-data display, average and derived-indicator calculation, prediction launch, explanation rendering, historical trend display, voice upload and playback, and error handling were verified.

Keywords: PSYCHOEMOTIONAL DISORDERS, DIGITAL MARKERS, DIGITAL MONITORING, WEB APPLICATION, ACTIVITY, SLEEP, VOICE CHARACTERISTICS, CATBOOST, MACHINE LEARNING.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної області та підходів до попереднього оцінювання психоемоційних порушень	12
1.1 Психоемоційні порушення як об'єкт цифрового спостереження та цифрові маркери активності, сну і голосу.....	12
1.2 Огляд і аналіз існуючих рішень.....	15
1.3 Постановка задачі	21
2 Алгоритмічне та інформаційне забезпечення програмної системи	23
2.1 Загальна структура проектованої системи	23
2.2 Алгоритмічне забезпечення системи	26
2.3 Інформаційне забезпечення	30
3 Програмно-технологічне забезпечення системи попереднього оцінювання психоемоційних порушень	34
3.1 Вибір та аналіз програмно-апаратних засобів для програмної системи.....	34
3.2 Програмно-технологічне забезпечення системи	36
3.3 Формування та обґрунтування навчального набору даних	43
3.4 Тестування системи	46
Висновки	50
Список використаних джерел	52
Додаток А Розширена архітектура модуля	56
Додаток Б Діаграма послідовності програмного модуля	57
Додаток В Код програмно апаратного модуля.....	58
Додаток Г Апробація отриманих результатів	69

ВСТУП

В сучасних умовах цифрові технології активніше проникають у сферу охорони здоров'я та персонального моніторингу. Смартфони або носимі пристрої та програмні засоби аналізу даних дають змогу фіксувати окремі фізіологічні чи поведінкові показники, виявляти зміни, що можуть бути пов'язані з психоемоційним станом людини [1, 2, 11, 12]. Особливої актуальності це набуває в задачах раннього виявлення депресії та тривожності, оскільки такі стани впливають на якість життя, працездатність, соціальну активність і загальне функціонування людини, але часто залишаються непоміченими на початкових етапах.

Традиційні підходи до оцінювання депресивних і тривожних проявів переважно ґрунтуються на клінічних інтерв'ю, опитувальниках та самооцінці користувача. Не зважаючи на їх значущість, такі методи здебільшого відображають стан людини лише в конкретний момент часу та значною мірою залежать від суб'єктивності відповідей.

Сучасні підходи цифрового фенотипування психоемоційного стану орієнтовані на використання цифрових слідів повсякденної активності людини, що формуються під час користування смартфоном або носимими пристроями [1, 3, 11, 12]. Це відкриває можливість переходу від одноразового оцінювання до більш регулярного та менш нав'язливого моніторингу психоемоційного стану у звичних умовах життя.

Серед цифрових маркерів, які мають практичну цінність в задачах такого моніторингу, особливе місце посідають показники активності, характеристики сну та голосові ознаки. Зміни рівня рухової активності, порушення режиму сну, зростання малорухомості, а також зміни темпу мовлення, гучності, інтонації чи спектральних характеристик голосу можуть виступати непрямими індикаторами погіршення психоемоційного стану [3, 5, 6, 9, 10]. При цьому ні один з цих показників окремо не є універсально достатнім для висновку, тому найбільш доцільним є їх спільне використання в межах багатфакторної системи оцінювання ризику

Аналіз існуючих рішень показує, що в цій сфері сьогодні використовуються кілька основних підходів: а) мобільні застосунки; б) wearable-рішення; г) системи digital phenotyping та засоби аналізу мовлення.

Мобільні застосунки вирізняються доступністю та масштабованістю, але спираються на непрямі поведінкові ознаки. Носимі пристрої забезпечують більш стабільний збір даних про фізіологію, активність і сон, проте потребують додаткового обладнання.

Підходи digital phenotyping є найбільш повними, тому що поєднують кілька потоків даних, але водночас ускладнюють технічну реалізацію і не закривають питання приватності.

Системи аналізу мовлення є перспективним напрямом, але їх можна розглядати як допоміжний канал в складі комплексного програмного рішення [5, 6, 9].

Тому актуальність теми полягає в потребі створення програмної системи, яка забезпечує цілісний процес попереднього оцінювання ризику психоемоційних порушень — від завантаження й підготовки цифрових маркерів користувача до їх аналізу, формування прогностичних імовірностей, подання пояснення та відображення динаміки змін у зручній для лікаря формі.

Така система має враховувати непрямий характер цифрових маркерів, потребу у поєднанні кількох джерел даних, необхідність накопичення історії спостережень, вимоги до конфіденційності персональної інформації та потребу в персоналізованішій інтерпретації вимірювань з урахуванням віку, статі, індексу маси тіла й типу навантаження користувача.

Метою даної роботи є проектування та розробка програмної системи попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача, що забезпечує завантаження, обробку та аналіз структурованих фізіологічних, поведінкових і голосових даних, формування прогнозу та подання результатів для подальшого клінічно орієнтованого використання.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

- провести аналіз предметної області цифрового спостереження за психоемоційним станом і визначити основні цифрові маркери, придатні для використання в системі;
- виконати огляд існуючих підходів і рішень у сферах моніторингу пристроїв зчитування, цифрового фенотипу та аналізу мовлення;
- спроектувати загальну структуру веборієнтованої програмної системи та розробити алгоритмічне забезпечення збору й обробки даних;
- розробити модель формування прогнозних імовірностей для структурованих і голосових даних, а також логіку відображення історичних змін у часі;
- спроектувати інформаційне забезпечення системи з урахуванням профілів пацієнтів, файлових сутностей та аналітичних артефактів;
- реалізувати програмно-технологічну частину вебсистеми та провести тестування її основних сценаріїв.

Об’єктом дослідження є процес цифрового моніторингу психоемоційного стану людини за поведінковими та голосовими даними.

Предметом дослідження є методи, алгоритми та програмні засоби обробки показників активності, сну, серцевого ритму, похідних табличних ознак і голосових характеристик для попереднього оцінювання ризику психоемоційних порушень.

Практичне значення одержаних результатів полягає у створенні веборієнтованої програмної системи з інтерфейсом для лікаря, файловим керуванням даними пацієнтів, окремими модулями аналізу структурованих і голосових даних, відображенням прогнозних імовірностей, поясненням результатів та переглядом динаміки спостережень у часі.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків. Основні положення кваліфікаційної роботи апробовано у тезах конференції «Програмна система попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача» [33]. Структуру та оформлення кваліфікаційної роботи виконано

відповідно до методичних рекомендацій кафедри інформаційно-обчислювальних систем і управління для спеціальності 122 «Комп'ютерні науки» [34].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПІДХОДІВ ДО ПОПЕРЕДНЬОГО ОЦІНЮВАННЯ ПСИХОЕМОЦІЙНИХ ПОРУШЕНЬ

1.1 Психоемоційні порушення як об'єкт цифрового спостереження та цифрові маркери активності, сну і голосу

В сучасних умовах дедалі більшої уваги набувають підходи, пов'язані з цифровим моніторингом психоемоційного стану людини. Це пояснюється тим, що депресія і тривожні розлади є поширеними станами, які суттєво впливають на якість життя, працездатність, соціальну активність і загальне функціонування людини, однак часто залишаються своєчасно не виявленими. Одним із перспективних напрямів є використання цифрових даних, які можуть збиратися за допомогою смартфона та носимих пристроїв, для виявлення поведінкових змін, пов'язаних із ризиком погіршення психічного стану. В науковій літературі такий підхід розглядається в межах цифрового фенотипування, тобто кількісного опису поведінки людини на основі даних, які формуються в процесі повсякденного користування персональними цифровими пристроями [1, 11, 12].

На відміну від методів традиційної клінічної оцінки, цей модуль орієнтований на виявленні прихованих цифрових закономірностей. Аналізуючи щоденну активність, система визначає непрямі ознаки, що можуть свідчити на можливі відхилення у психоемоційному стані. Йдеться про три групи показників, таких як активність користувача, характеристики сну та голосові параметри мовлення. Вибір цих даних обґрунтовується тим, що депресія і тривожність часто проявляються не лише через суб'єктивні скарги, а і через зміни у звичній поведінці, тобто зменшення рухової активності, порушення режиму сну, зниження варіативності щоденної мобільності, зміни у темпі мовлення, інтонації, гучності або емоційній виразності голосу. Тому програмний модуль розглядає ці сигнали як цифрові маркери, що можуть свідчити про підвищений ризик депресивних або тривожних проявів [1, 3, 4, 7, 10].

Серед них важливе місце посідають показники активності. В межах цифрового моніторингу активність може описуватися різними способами,

наприклад кількістю кроків, інтенсивністю руху, загальною мобільністю, тривалістю періодів малорухомості, частотою виходу з дому, кількістю відвіданих локацій, різноманітністю маршрутів та загальною динамікою повсякденного пересування. Систематичний огляд досліджень цифрового фенотипування показує, що GPS, акселерометр та інші пасивні сенсори є одними з найвживаніших джерел для фіксації таких параметрів [2, 7, 8, 10, 13]. При цьому в багатьох роботах виявлено, що для людей з вираженішими депресивними, тривожними або стресовими проявами характерні менша кількість відвіданих місць, нижча варіативність пересування та більша частка сидячої або пасивної поведінки.

Також в частині досліджень було зафіксовано зв'язок між зростанням малорухомості та посиленням депресивної симптоматики [13, 14]. З технічної точки зору такі ознаки є цінними, тому що їх можна збирати відносно непомітно для користувача протягом тривалого часу. Це дає змогу перейти від одноразового опитування до безперервного спостереження за поведінковими патернами.

В той самий час низький рівень активності сам по собі не є достатньою підставою для висновку про депресію чи тривожний розлад, оскільки може пояснюватися стилем життя, професією, станом здоров'я, погодними умовами, навчальним навантаженням або іншими зовнішніми чинниками. Тому такі системи розглядають активність в поєднанні з іншими цифровими ознаками [1; 2].

Однією з таких ознак є характеристики сну, які відіграють важливу роль в цифровому моніторингу психічного стану, тому що його порушення можуть одночасно бути і симптомом і фактором ризику. В наукових роботах зазначається, що вони є одним із клінічних проявів психічних розладів і пов'язані з підвищеним ризиком розвитку депресії та тривожності. Зокрема, безсоння може впливати на тяжкість, тривалість і повторюваність депресивних епізодів [3].

Огляд цифрового фенотипування також показує, що гірша якість сну та менша його тривалість часто корелюють з вираженими депресивними симптомами, вищим рівнем стресу та емоційною нестабільністю [3, 7, 10]. В практичному сенсі це означає, що для програмного модуля інформативними можуть бути такі характеристики, як загальна тривалість сну, час засинання, кількість нічних

пробуджень, пробудження після засинання, час перебування в ліжку, нерегулярність режиму сну та зміщення циркадного ритму.

Важливою перевагою цифрового збору таких даних є те, що смартфони та носимі пристрої дають змогу оцінювати сон у природних умовах, а не лише в межах лабораторного обстеження. В літературі зазначається, що полісомнографія залишається стандартом дослідження сну, проте її застосування не завжди є зручним і доступним у повсякденній практиці [3].

Водночас використання смартфонів для оцінювання сну має суттєві обмеження, оскільки споживчі мобільні пристрої не оснащені сенсорами медичного класу, а повноцінне визначення стадій сну лише на їх основі наразі є неможливим. Тому дані про сон в межах такого модуля доцільно розглядати як допоміжні поведінкові індикатори, а не як засіб повноцінної сомнологічної діагностики.

Ще одним важливим джерелом цифрових ознак, яке доцільно аналізувати разом із показниками сну, є голосові характеристики [5, 6, 9]. Інтерес до голосу пояснюється тим, що емоційний стан, психомоторна загальмованість, тривога, виснаження або зниження мотивації можуть відображатися в особливостях мовлення.

В сучасних дослідженнях аналізують такі параметри, як висота тону, гучність, темп мовлення, паузи, спектральні характеристики, варіативність голосу, лексичне багатство та емоційне забарвлення мовлення [5, 6, 9]. Аналіз наявних даних показує, що окремі акустичні характеристики, зокрема пов'язані з тоном і гучністю, статистично відрізняють мовлення людей з депресивними проявами від здорових учасників, а також дають змогу розмежовувати різні рівні вираженості депресивного стану.

Це свідчить про перспективність голосу як джерела цифрових маркерів, але також вказує на потребу обережного використання таких ознак через наявні обмеження досліджень і невеликої вибірки.

Тобто голос може виступати додатковим джерелом оцінювання ризику, особливо якщо він аналізується разом з активністю та сном. Якщо система

одночасно фіксує зниження мобільності, порушення сну та появу нетипових голосових характеристик, ймовірність того, що йдеться саме про психоемоційне погіршення, зростає. Також голос є дуже чутливим до сторонніх чинників, таких як застуда, втома, шумне оточення, мови спілкування, індивідуальних особливостей дикції, якості мікрофона, та контексту розмови [4, 5, 6, 9].

Також тре враховувати, що навіть широко вживані інструменти оцінювання психоемоційного стану, зокрема PHQ-9 і GAD-7, застосовуються переважно для попереднього обстеження, а не для встановлення остаточного клінічного діагнозу. Крім того, точність їх використання залежить від коректної мовної та культурної адаптації, оскільки без цього зростає ризик неточного трактування результатів. [17, 18, 19].

Не менш важливими є і практичні та етичні обмеження такого підходу [15, 16]. Дослідження цифрового фенотипування вказують на проблеми приватності, тому що дані про локацію, використання телефону, дзвінки, мікрофон або контекст спілкування можуть бути чутливими для користувача. Крім цього, між людьми існують значні індивідуальні відмінності, через що однакові правила не завжди однаково добре працюють в різних групах. Слід також враховувати, що частина наявних досліджень виконана на невеликих або неклінічних вибірках, тому узагальнювати їх результати потрібно обережно.

Тому під час проектування програмного модуля потрібно передбачити добровільність збору даних, мінімізацію чутливої інформації, локальну або захищену обробку, можливість пояснення отриманого результату та використання адаптивних порогів замість жорстких універсальних правил [15, 16].

1.2 Огляд і аналіз існуючих рішень

В сучасних дослідженнях цифровий моніторинг депресії та тривожності розвивається в кількох основних напрямках [7, 8, 10, 16]. Найчастіше йдеться про мобільні застосунки для смартфонів, рішення на основі носимих пристроїв, підходи digital phenotyping, в яких поєднуються кілька потоків цифрових даних а також

системи, що додатково використовують мовлення або аудіосигнали. Усі ці підходи мають спільну мету переходу від одноразового опитування до більш регулярного і менш нав'язливого спостереження за змінами психоемоційного стану людини в її звичних умовах життя.

Одним із найбільш доступних напрямів є мобільні застосунки. Їхня головна перевага полягає в тому, що смартфон вже є повсякденним пристроєм для більшості користувачів, тому окреме спеціальне обладнання часто не потрібне. Такі системи можуть поєднувати активний збір даних, наприклад короткі опитування і самооцінки, з пасивним збором інформації про рухову активність, мобільність, освітленість, режим користування телефоном, взаємодію з екраном, дзвінки, повідомлення або інші поведінкові сигнали [1, 2, 12, 13, 14].

В наукових роботах показано, що сенсори смартфона дають змогу виявляти поведінкові патерни, пов'язані зі стресом, тривожністю та депресивними проявами, тому можуть використовуватися як технічна основа для раннього виявлення змін психічного стану [2, 7, 8, 12, 13]. На рисунку 1.1 зображено загальну роль смартфона як засобу збору цифрових маркерів.

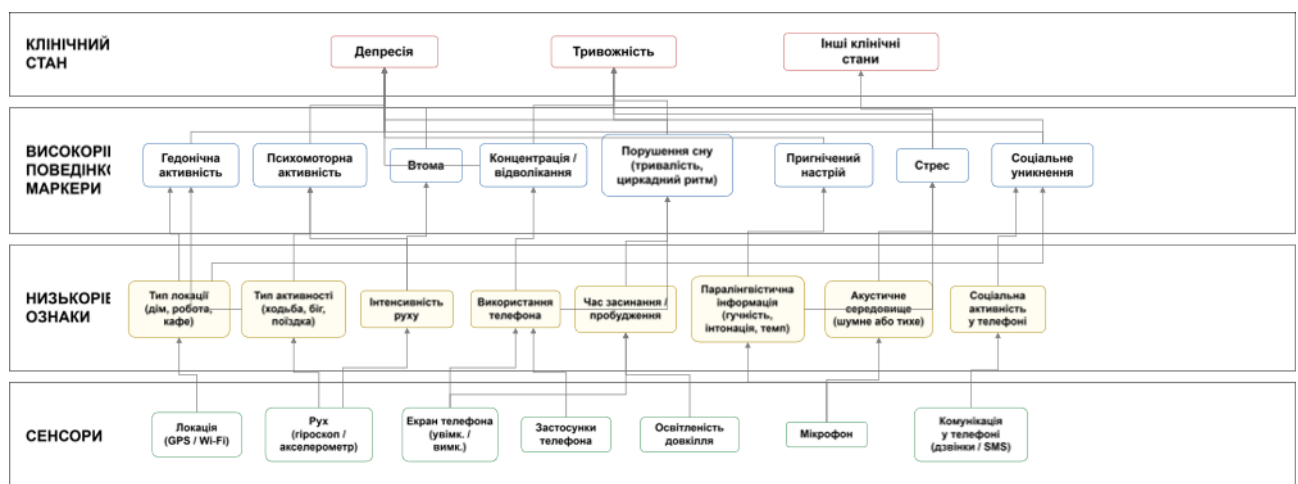


Рисунок 1.1 - Смартфон для збору цифрових маркерів

Важливим фактором для впровадження таких систем є наявність стабільної та розгалуженої сенсорної бази. Сучасні смартфони вже протягом тривалого часу оснащені стандартним набором модулів, таких як акселерометри, GPS, Bluetooth, мікрофони, камери, Wi-Fi та засоби стільникового зв'язку. Саме ці апаратні

компоненти є найбільш універсальними джерелами даних для побудови кросплатформних систем цифрового моніторингу [1, 2, 12, 14].

Розробка мобільних програмних рішень у цій сфері стикається з низкою технологічних викликів. Надійність збору даних безпосередньо залежить від наданих користувачем дозволів, особливостей фонової роботи процесів, безпекових політик операційних систем та стабільності доступу до сенсорів у реальному часі. Крім того, так як більшість отримуваних параметрів є непрямими показниками стану людини, їхня обробка та інтерпретація в межах моделей оцінювання ризику потребують особливої точності та врахування багатьох контекстуальних факторів [12].

Ще одним важливим напрямом є носимі-рішення, тобто системи на основі смарт годинників, фітнес-браслетів та інших носимих пристроїв. На відміну від мобільних застосунків, вони краще адаптовані для безперервного збору фізіологічних та поведінкових сигналів. На рисунку 1.2 зображено Wearable-пристрої для моніторингу активності, сну та фізіологічних показників.



Рисунок 1.2 - Wearable-пристрої для моніторингу

Зокрема, це стосується даних про кількість кроків, загальну рухову активність, параметри сну, серцевий ритм та похідні від нього показники. Wearable AI на сьогодні демонструють найвищу ефективність саме у сфері виявлення ознак

депресії та тривожних станів [8; 10]. Основний вектор застосування таких рішень спрямований на автоматизацію діагностичних та скринінгових задач, що дозволяє виявляти ризики на ранніх етапах, хоча вони й не є повноцінною заміною терапевтичних методів лікування [10].

Разом з тим, результати роботи wearable-рішень потребують зваженого трактування. Попри високу перспективність використання штучного інтелекту в носимих пристроях, цей технологічний напрям перебуває на етапі становлення. Поточний рівень зрілості систем не дозволяє використовувати їх як єдиний інструмент для повноцінної клінічної діагностики. Тому такі програмні модулі мають застосовуватися як допоміжні засоби у поєднанні з іншими методами оцінювання та верифікації стану користувача [8, 9, 10].

Найбільш комплексним підходом на сьогодні є digital phenotyping [1, 3, 4, 11, 12]. Його сутність полягає в тому, що психоемоційний стан оцінюється не за одним окремим сигналом, а за сукупністю поведінкових і фізіологічних показників, які збираються зі смартфона, носимого пристрою та, за потреби з коротких самооцінок користувача.

Аналіз сучасної науково-технічної бази дозволяє виділити базовий набір ознак, що мають найвищу прогностичну цінність, до якого належать акселерометр, кількість кроків, серцевий ритм і сон [10]. Саме ці показники виявилися найбільш стабільними та інформативними для прогнозування розладів настрою на різних типах пристроїв. На рисунку 1.3 зображено загальну схему підходу digital phenotyping в задачах моніторингу депресії та тривожності.

Практичну реалізацію такого підходу ілюструють сучасні прототипи, в яких застосунок який працює на смартфоні поєднується з носимим пристроєм і психометричними шкалами. В таких системах одночасно збираються активні дані через PHQ-9, GAD-7 та інші опитувальники, а також пасивні дані про фізичну активність, місцеперебування, освітленість, режим користування телефоном, сон і фізіологічні реакції [14, 17, 18, 19]. Це показує, що сучасні рішення поєднують моніторинг, аналітику і зворотний зв'язок для користувача. На рисунку 1.3 зображено загальну схему підходу digital phenotyping.



Рисунок 1.3 - Загальна схема підходу digital phenotyping

Окремий напрям становлять системи з аналізом мовлення [5, 6, 9]. Вони базуються на припущенні, що емоційний стан, психомоторна загальмованість, тривога, виснаження або зниження мотивації можуть відображатися в темпі мовлення, висоті тону, паузах, гучності та інших акустичних характеристиках.

В дослідженнях показано, що мовленнєві моделі мають предиктивний потенціал для виявлення депресивних і тривожних проявів, але їхня ефективність залежить від якості запису, контексту мовлення та особливостей вибірки [5, 6, 9]. Тому аналіз голосу розглядається не як окреме універсальне рішення, а як додатковий канал в складі багатофакторної системи. На рисунку 1.4 зображено використання голосових характеристик у системах цифрового моніторингу психоемоційного стану.



Рисунок 1.4 - Використання голосових характеристик

Проведений аналіз існуючих рішень показує те, що кожен із розглянутих підходів ефективно розв'язує лише окремі частини моніторингу психоемоційного стану. Ключові функціональні особливості та технічні відмінності між існуючими сервісами та методами цифрового моніторингу наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння підходів до цифрового моніторингу ризику депресії та тривожності

Підхід	Основні дані	Переваги	Обмеження
Мобільні застосунки	активність, GPS, використання смартфона, опитування	доступність, масштабованість, відсутність потреби в окремому пристрої	залежність від ОС, дозволів і непрямий характер частини ознак
Wearable-рішення	кроки, сон, серцевий ритм, рухова активність	безперервний збір фізіологічних і поведінкових сигналів	потреба в додатковому пристрої, різноманітність платформ
Digital phenotyping	поєднання смартфонних, wearable і self-report даних	найбільш повний підхід, краща персоналізація	вища технічна складність, питання приватності
Аналіз мовлення	темп мовлення, висота тону, паузи, гучність	додатковий канал оцінювання емоційного стану	чутливість до шуму, контексту та якості запису

Розглянуті рішення які представлені в таблиці, показують, що жоден із підходів не є універсальним. Мобільні застосунки є найдоступнішими і найпростішими з точки зору масштабування, але вони часто спираються на непрямі поведінкові ознаки.

Wearable-рішення дають більш стабільні дані про фізіологію, рух і сон, але потребують додаткового обладнання і ще не мають достатньої клінічної зрілості для самостійного використання.

Підходи digital phenotyping є найбільш повними, тому що поєднують кілька потоків даних і дозволяють будувати персоналізовані моделі, але вони також є найбільш складними з технічної та етичної точок зору.

Системи з аналізом мовлення мають значний потенціал, однак на сучасному етапі доцільніше розглядати їх як допоміжне доповнення до даних про активність, сон та використання пристрою [5, 6, 9].

1.3 Постановка задачі

У сучасних умовах зростає потреба в інструментах, що дають змогу вчасно виявляти зміни психоемоційного стану людини та реагувати на них ще до переходу у більш виражені форми порушень. Використання цифрових даних про активність, сон, фізіологічні показники та голосові характеристики відкриває можливість побудови програмних рішень, орієнтованих на попереднє оцінювання ризику на основі мультимодальних цифрових маркерів.

Актуальність задачі полягає в необхідності організувати процес цифрового моніторингу як цілісну систему: від збирання та завантаження даних пацієнта до їх агрегування, прогнозування, візуалізації та збереження для повторного аналізу. Такий підхід має враховувати непрямий характер цифрових маркерів, потребу в поєднанні кількох джерел даних, а також вимоги до конфіденційності персональної інформації та коректного трактування результатів.

Метою даної роботи є проєктування та розробка програмної системи попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача.

Для досягнення цієї мети необхідно вирішити такі завдання:

1. Провести аналіз предметної області цифрового спостереження за психоемоційним станом та визначити цифрові маркери, придатні для використання у програмній системі.

2. Виконати огляд існуючих підходів і рішень, зокрема мобільних застосунків, девайсів спостереження, підходів цифрового фенотипування та засобів аналізу мовлення, і сформулювати перелік функціональних вимог до системи.

3. Спроєктувати загальну структуру веборієнтованої системи та визначити її основні компоненти і взаємодію між ними.

4. Розробити алгоритмічне забезпечення завантаження, підготовки та аналізу структурованих і голосових даних.
5. Розробити модель формування прогнозних імовірностей та відображення змін стану у часових вікнах спостереження.
6. Розробити інформаційне забезпечення системи та визначити структуру даних для профілів пацієнтів, файлів спостереження та результатів аналізу.
7. Реалізувати програмно-технологічну частину системи та забезпечити її роботу з табличними файлами щоденних вимірювань і WAV-записами мовлення.
8. Провести тестування працездатності системи та перевірити коректність основних сценаріїв завантаження, обробки, прогнозування та візуалізації даних.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОЇ СИСТЕМИ

2.1 Загальна структура проектованої системи

Програмна система функціонує як інтегрований вебзастосунок, у межах якого поєднано інтерфейс лікаря, серверну логіку, файлову підсистему для роботи з даними пацієнтів, модулі аналітичної обробки та засоби візуалізації результатів. На відміну від ранньої локальної реалізації, фінальний варіант проєкту орієнтований на роботу через сторінку пацієнта, де всі основні сценарії аналізу виконуються в межах єдиного web workflow.

У загальному вигляді система складається з кількох взаємопов'язаних блоків: блоку автентифікації та доступу лікаря, блоку керування профілями пацієнтів, підсистеми зберігання і вибору активних файлів, API-рівня для повернення структурованих JSON-відповідей, модуля аналізу структурованих табличних даних, окремого голосового модуля та блоку подання результатів.

На верхньому рівні структури знаходиться лікар або інший авторизований користувач, який взаємодіє із системою через вебінтерфейс. Через інтерфейс здійснюється вибір пацієнта, перегляд профілю, завантаження нових табличних і голосових файлів, вибір активного файлу для аналізу, запуск прогнозування та перегляд збережених результатів.

Однією з перших складових загальної структури є блок керування профілями пацієнтів. У межах цього блоку виконується створення та редагування записів, у яких зберігаються анкетні дані, вік, стать, антропометричні показники, професійне або навчальне навантаження та прив'язка до відповідального лікаря.

Наступним елементом є підсистема зберігання файлів спостереження. Вона формує основний вхідний потік для всієї системи, оскільки саме з завантажених табличних файлів зчитуються щоденні показники активності, сну та серцевого ритму, а з WAV-файлів — голосові дані. Важливою функцією цієї підсистеми є можливість вибору активного файлу, який надалі використовується в аналізі.

Центральним елементом структури є аналітичне ядро системи. Воно отримує дані з активного файлу, виконує первинну перевірку коректності, обчислює середні

значення, похідні індикатори та часові вікна спостереження, а далі передає підготовлені ознаки до відповідних моделей. Для структурованих табличних даних використовується багатокласова модель CatBoost [27], а для голосового каналу — окремий модуль виділення MFCC, RMS і ZCR з подальшим оцінюванням. Загальну архітектуру програмної системи подано на рисунку 2.1.

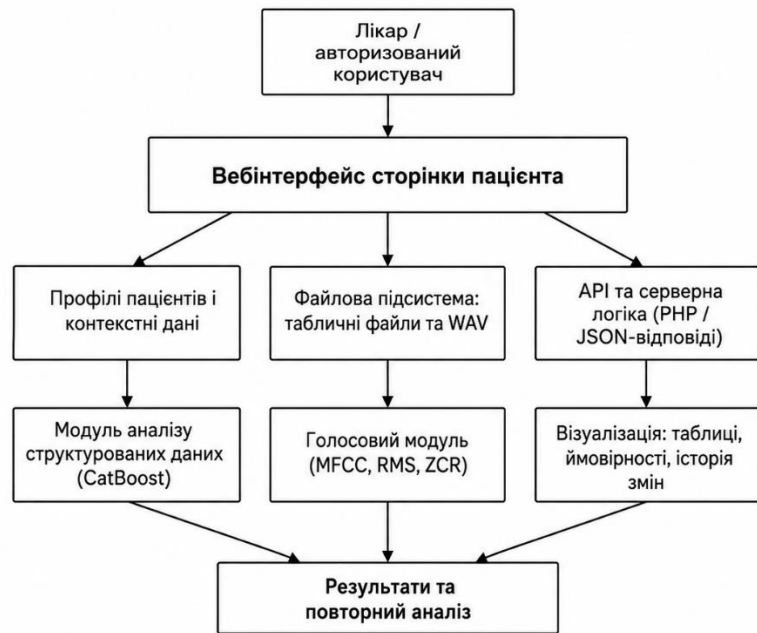


Рисунок 2.1 – Загальна архітектура програмної системи попереднього оцінювання психоемоційних порушень

Після завантаження даних система переходить до етапу попередньої обробки. Для табличного каналу виконується нормалізація назв стовпців, перетворення даних у числовий формат, розрахунок агрегованих характеристик і формування набору ознак, придатного для подальшого аналізу. Для голосового файлу використовується окремий алгоритм обробки, у межах якого з аудіосигналу виділяються інформативні голосові характеристики.

Особливістю фінальної версії системи є те, що для структурованих фізіологічних і поведінкових даних не використовується набір окремих незалежних локальних модулів із подальшим ручним об'єднанням оцінок. Натомість застосовується єдина багатокласова модель, що працює з агрегованими числовими ознаками та контекстними параметрами користувача.

Така побудова є більш придатною для практичного використання, оскільки дозволяє враховувати взаємозв'язок між показниками сну, активності, калорій, пульсу та контекстними атрибутами пацієнта. Завдяки цьому однакові сирі вимірювання не обов'язково інтерпретуються однаково для різних профілів користувачів.

Окремий голосовий модуль працює з параметрами мовлення і формує додатковий канал оцінювання. Після завантаження WAV-файлу система виконує перевірку формату, виділяє вектор ознак і повертає розподіл імовірностей класів разом із графічним і текстовим поясненням результату.

Після завершення аналітичної обробки система переходить до блоку подання результатів. У цьому блоці відображаються таблиця щоденних даних, середні значення, похідні індикатори, прогнозні імовірності, пояснення та історія змін у часі. Така організація дає змогу не лише отримати поточний прогноз, а й простежити динаміку стану пацієнта.

Таким чином, у структурі системи поєднуються профіль пацієнта, файлове керування даними, серверна аналітика, окремий голосовий канал і засоби візуального подання результатів. На рисунку 2.2 подано узагальнену взаємодію цих функціональних блоків.

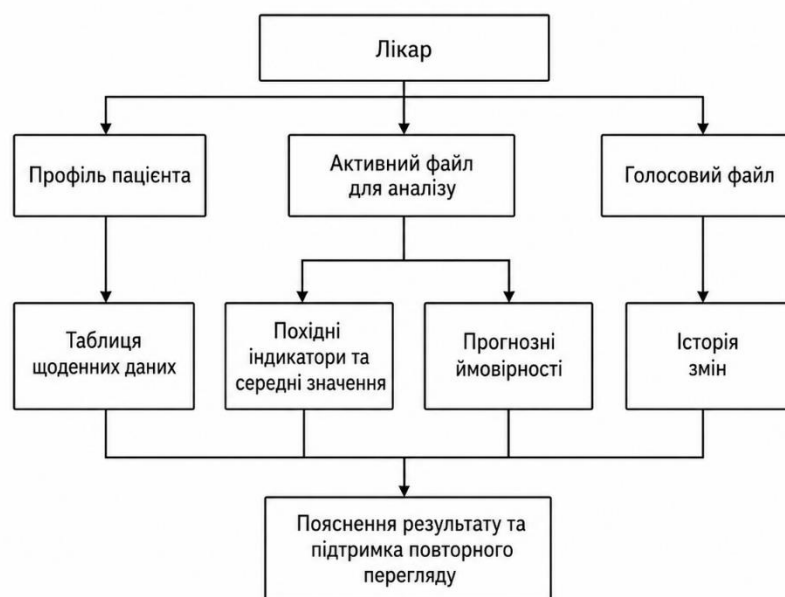


Рисунок 2.2 – Взаємодія функціональних блоків веборієнтованої системи

Після формування результату система зберігає службову інформацію про вибраний файл, пацієнта та аналітичні артефакти, що забезпечує відтворюваність обробки. У фінальній реалізації важливим є не тільки збереження профілю пацієнта, а й контроль того, який саме файл використовується в поточному аналізі.

Окремим елементом структури є блок перегляду історії. Його призначення полягає у відображенні раніше сформованих часових вікон спостереження, середніх значень і змін стану, які інтерпретуються як *baseline*, *stable*, *improved* або *worsened*.

Завершальним функціональним елементом є блок візуалізації ризику, який подає результат у зрозумілому для користувача вигляді: у формі діаграми імовірностей, пояснювального тексту та таблиці динаміки.

2.2 Алгоритмічне забезпечення системи

Робота системи попереднього оцінювання базується на послідовному процесі завантаження вхідних даних, перевірки їх коректності, обробки, формування ознак та повернення результату у вебінтерфейс. Алгоритмічна логіка реалізована навколо серверних сценаріїв і *endpoint*-ів, які забезпечують відтворюваний конвеєр аналізу для кожного пацієнта.

Після вибору пацієнта система визначає активний табличний або голосовий файл. Якщо файл відсутній або має некоректний формат, аналіз не запускається і користувачу повертається повідомлення про помилку. Якщо файл доступний, виконується його читання, нормалізація та підготовка даних до подальшої обробки. Загальний алгоритм роботи системи наведено на рисунку 2.3.

Алгоритм починається з вибору пацієнта та відповідного файлу для аналізу. У межах цього блоку система визначає контекст користувача, перевіряє доступ лікаря та обирає саме той файл, який позначено як активний. Це усуває неоднозначність у випадках, коли для одного пацієнта збережено кілька файлів спостереження.

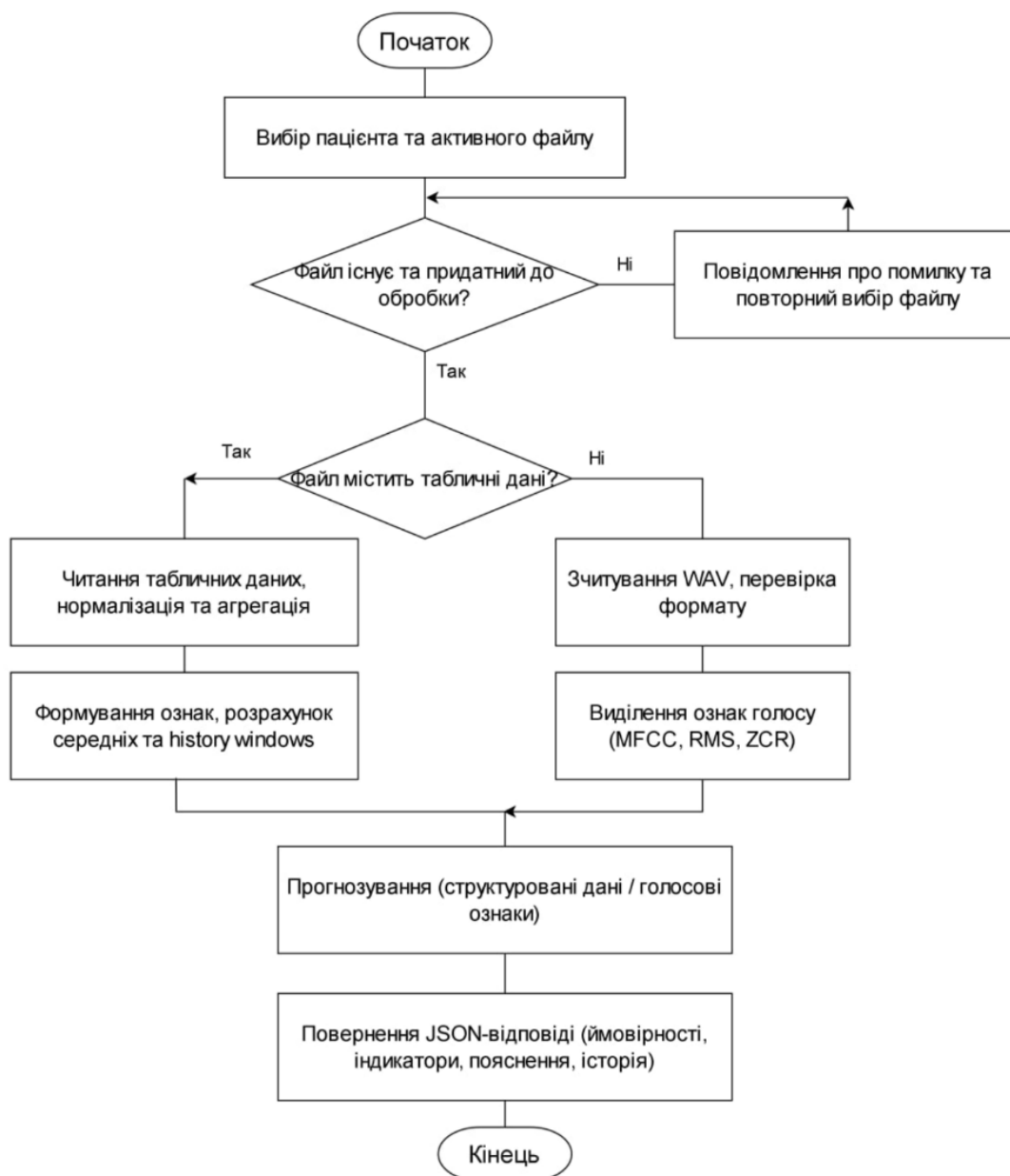


Рисунок 2.3 – Загальний алгоритм роботи системи

Далі система переходить до етапу читання структурованих даних або аудіофайлу. Для табличного каналу вхідними змінними є щоденні значення кроків, калорій, середнього пульсу, тривалості сну, кількості пробуджень і REM-відсотка. Для голосового каналу вхідним є WAV-файл мовлення.

Якщо перевірка виявляє неповні або некоректні дані, алгоритм повертає користувача до попереднього етапу й формує повідомлення про помилку. Якщо ж усі значення придатні до обробки, система переходить до формування ознак.

На цьому етапі для табличних даних обчислюються середні значення, похідні індикатори, контекстні ознаки користувача та часові вікна спостереження. Далі

підготовлений вектор ознак подається на вхід моделі CatBoost, яка повертає розподіл імовірностей між класами «норма», «тривога», «депресія» та «ПТСР».

Після одержання ймовірностей система відображає результат у вигляді діаграми, текстового пояснення та таблиці історичних змін. Завдяки цьому користувач отримує не лише кінцевий прогноз, а і контекст його формування.

Після загального алгоритму роботи системи окремо розглядається етап обробки голосових даних. Він забезпечує завантаження WAV-файлу, перевірку формату, виділення акустичних ознак та побудову окремого прогнозу на основі мовленнєвого сигналу. Алгоритм отримання та первинної перевірки аудіофайлу наведено на рисунку 2.4.

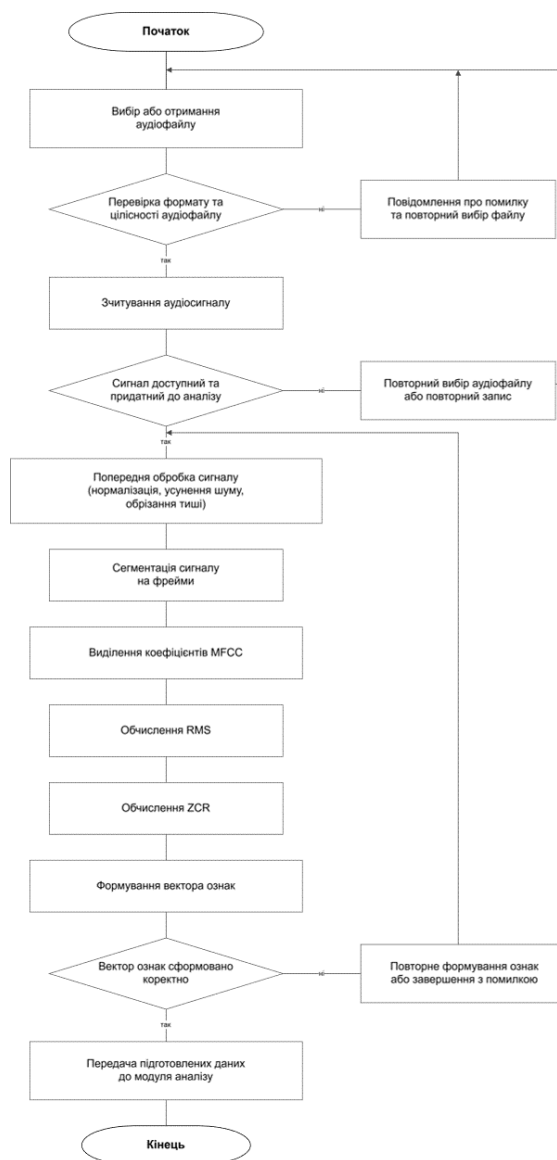


Рисунок 2.4 – Алгоритм отримання та перевірки аудіофайлу

Робота цього алгоритму починається з вибору аудіофайлу, який містить голосові дані для подальшого аналізу. На цьому етапі система перевіряє наявність файлу, його розширення та можливість зчитування сигналу.

Якщо аудіофайл не вибрано або його неможливо обробити, алгоритм не переходить до наступних кроків і повертає повідомлення про помилку. Якщо файл доступний, виконується завантаження аудіосигналу з приведенням до уніфікованого формату.

Наступним етапом є попередня обробка сигналу та формування вектора акустичних ознак. Призначення цього блоку полягає в тому, щоб перетворити сирий сигнал у числове представлення, придатне для роботи класифікаційної моделі.

Після цього система переходить до обчислення основних голосових характеристик. На рисунку 2.5 подано узагальнений алгоритм виділення ознак і формування результату голосового аналізу.

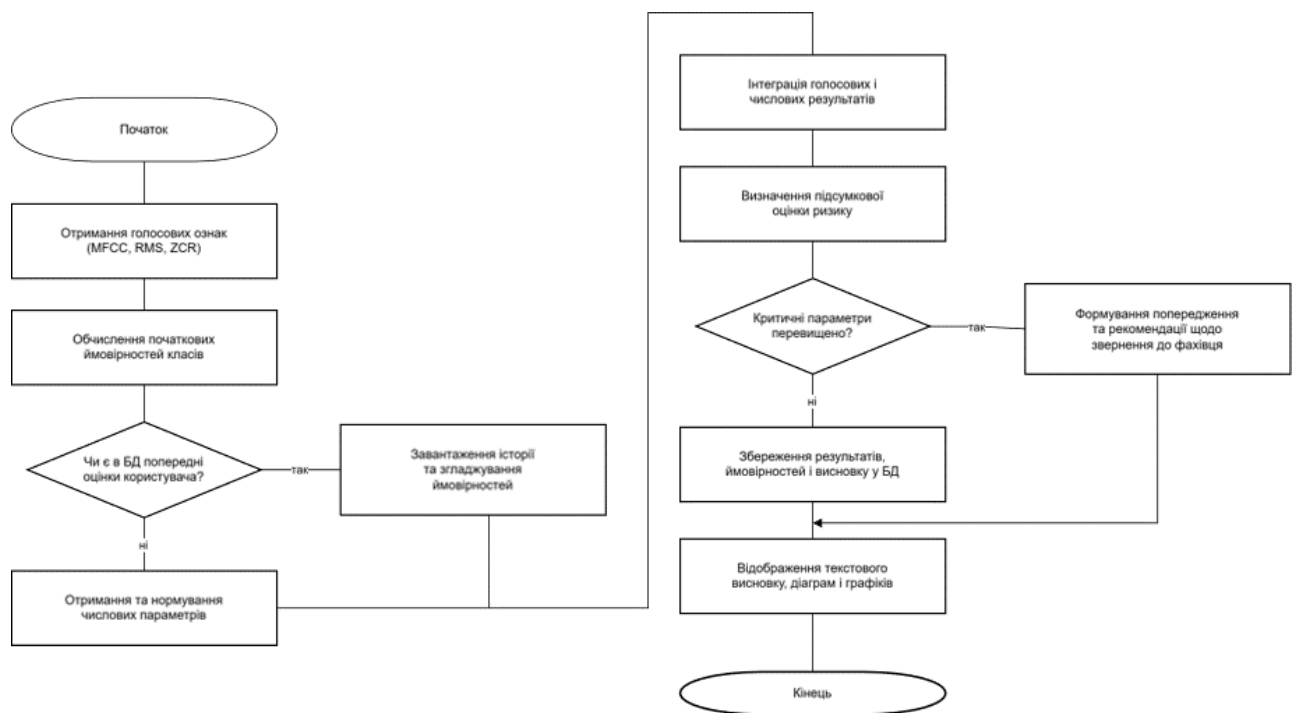


Рисунок 2.5 – Алгоритм голосового аналізу та формування результату

Далі в алгоритмі послідовно розміщено три процесні блоки виділення основних ознак.

Першим із них є блок виділення MFCC, у якому система обчислює мел-частотні кепстральні коефіцієнти, що використовуються для представлення спектральної структури мовлення.

Другим є блок обчислення RMS, у якому визначається енергія сигналу та враховується його інтенсивність.

Третім є блок обчислення ZCR, у якому визначається частота переходів сигналу через нуль. Після обчислення окремих характеристик система формує підсумковий вектор ознак і передає його на вхід збереженої моделі голосового аналізу.

Після формування голосових ознак система виконує прогнозування та повертає імовірності класів разом із хвильовою формою сигналу для візуалізації. Отриманий результат відображається у тій самій сторінці пацієнта, що забезпечує єдність роботи з обома аналітичними каналами.

На основі голосових ознак та даних структурованого каналу система формує два незалежні результати, які можуть використовуватися як взаємодоповнювальні джерела попередньої оцінки стану. Така побудова підвищує інформативність системи, але не замінює клінічного висновку.

Підсумкова логіка системи передбачає обов'язкову валідацію вхідних даних, обробку помилок і подання результату як допоміжної аналітичної інформації, а не як остаточного медичного діагнозу.

2.3 Інформаційне забезпечення

Для забезпечення стабільної роботи системи необхідно визначити інформаційні потоки, якими вона оперує. Робота вебзастосунку базується не лише на анкетних даних пацієнта, а і на файлових сутностях, службових зв'язках між ними та структурованих JSON-відповідях, що передаються між серверною й клієнтською частинами.

Оскільки система працює з кількома типами даних одночасно, інформаційне забезпечення повинно охоплювати профілі пацієнтів, табличні файли щоденних

вимірювань, WAV-файли голосу, карту активних файлів, артефакти моделей і результати аналітичної обробки.

Вхідна інформація системи охоплює сукупність даних, які надходять до неї перед початком аналізу. На першому етапі це ідентифікаційні та описові дані пацієнта: ім'я, стать, вік, антропометричні показники, професійне або навчальне навантаження, а також дані про відповідального лікаря.

Частиною вхідного потоку є табличні параметри поточного спостереження: кроки, витрати калорій, середній пульс, тривалість сну, кількість пробуджень, REM-відсоток та інші пов'язані показники. Окремим вхідним потоком є WAV-аудіофайл для голосового аналізу.

Вихідна інформація формується системою для подальшого перегляду та повторного використання. Основними елементами вихідного потоку є прогнозні імовірності класів, текстове пояснення, похідні індикатори, таблиця змін у часі, а також службові дані про активний файл і джерело результату.

Окремо зберігаються імена файлів, часові мітки завантаження, ознака активного вибору, технічні параметри обробки та службова інформація про версію моделі. Це забезпечує відтворюваність аналізу та спрощує аудит дій у системі. Формат вхідної та вихідної інформації узагальнено в таблиці 2.1.

Таблиця 2.1 – Формат вхідної та вихідної інформації системи

№	Назва поля	Тип даних	Призначення
1	patient id	ціле	ідентифікатор пацієнта
2	full name	текст	ПІБ або відображуване ім'я пацієнта
3	age	ціле	вік пацієнта
4	sex	текст	стать пацієнта
5	height cm	дійсне	ріст пацієнта, см
6	weight kg	дійсне	маса тіла пацієнта, кг
7	bmi	дійсне	індекс маси тіла
8	workload	текст	тип професійного або навчального навантаження
9	assigned doctor	текст	лікар, що працює з профілем пацієнта
10	measurement_file	текст	назва табличного файлу щоденних вимірювань

Продовження Таблиці 2.1

11	voice_file	текст	назва WAV-файлу голосового аналізу
12	is_active	логічне	ознака активного файлу для поточного аналізу
13	upload_time	дата/час	час завантаження файлу до системи
14	steps	дійсне	кількість кроків за день
15	calories	дійсне	витрати калорій
16	bpm_avg	дійсне	середній пульс
17	sleep_total_h	дійсне	загальна тривалість сну
18	awakenings	ціле	кількість нічних пробуджень
19	rem_pct	дійсне	частка REM-фази сну
20	steps_mean	дійсне	середнє значення кроків у вікні спостереження
21	calories_mean	дійсне	середнє значення витрат калорій
22	bpm_avg_mean	дійсне	середнє значення пульсу
23	sleep_total_h_mean	дійсне	середня тривалість сну
24	awakenings_mean	дійсне	середня кількість пробуджень
25	rem_pct_mean	дійсне	середній REM-відсоток
26	prob_normal	дійсне	ймовірність класу «норма»
27	prob_anxiety	дійсне	ймовірність класу «тривога»
28	prob_depression	дійсне	ймовірність класу «депресія»
29	prob_ptsd	дійсне	ймовірність класу «ПТСР»
30	trend_summary	текст	пояснення, історичний тренд і службова інформація

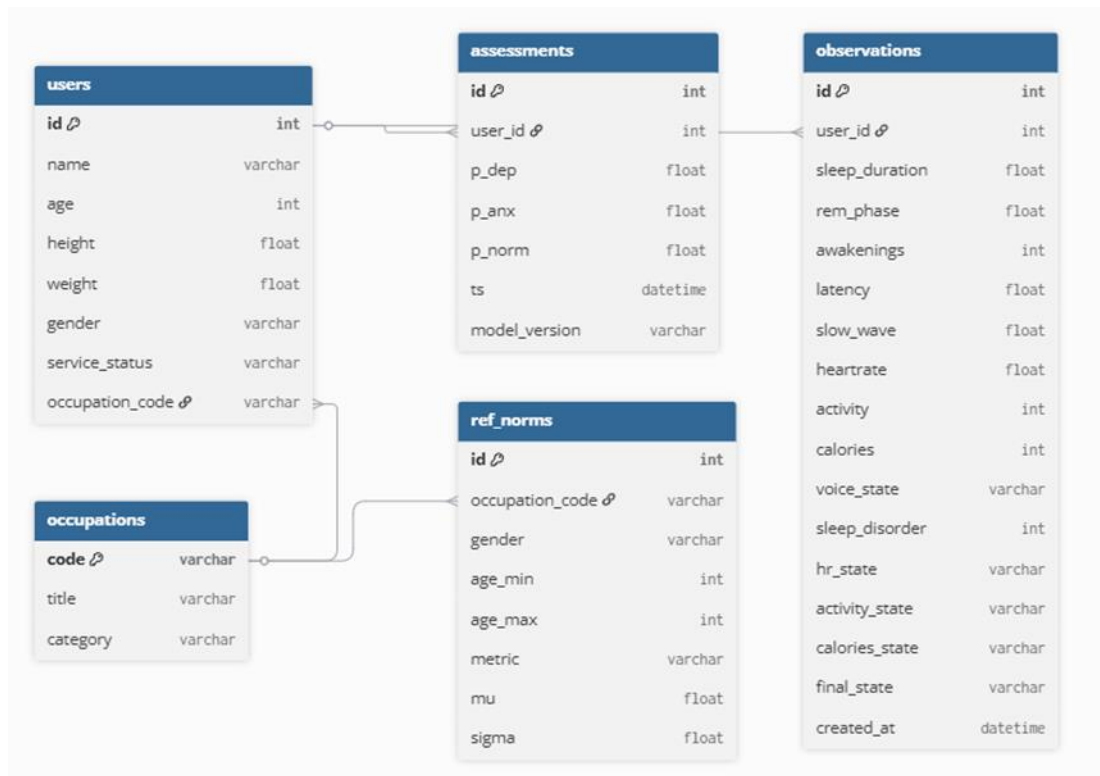
Для забезпечення надійності та цілісності інформації в системі реалізовано логічну структуру зберігання, яка поєднує профілі пацієнтів, файлові каталоги та аналітичні результати. Ключовими сутностями є:

- дані пацієнта (Patient);
- файли структурованих спостережень (PatientFile);
- голосові файли (VoiceFile);
- карта активних файлів та їх вибору для аналізу (ActiveFileMap);
- аналітичні результати та модельні артефакти (Assessment/Artifacts).

Взаємозв'язки між цими сутностями, що відображають логіку збереження та обробки даних, наведено на рисунку 2.6.

Центральною сутністю є профіль пацієнта, до якого прив'язуються табличні файли спостереження та голосові записи. Для кожного пацієнта може існувати

кілька файлів одного типу, але система завжди фіксує, який саме файл позначений як активний для аналізу. Результати прогнозування повертаються у форматі JSON та використовуються клієнтською частиною для побудови таблиць, графіків і пояснювальних блоків. Взаємозв'язки між цими сутностями наведено на рисунку 2.6.



Рисунк 2.6 – Логічна структура зберігання профілів, файлів та результатів аналізу

Така структура дозволяє накопичувати результати у часі, повторно використовувати раніше завантажені файли, контролювати активний вибір джерела для аналізу та забезпечувати коректне відображення динаміки спостережень.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПОПЕРЕДНЬОГО ОЦІНЮВАННЯ ПСИХОЕМОЦІЙНИХ ПОРУШЕНЬ

3.1 Вибір та аналіз програмно-апаратних засобів для програмної системи

Під час реалізації програмної системи попереднього оцінювання психоемоційних порушень було обрано такі основні програмні засоби: PHP для серверної логіки, HTML, CSS і JavaScript для клієнтського інтерфейсу, Python для аналітичних сценаріїв, а також бібліотеки NumPy, pandas, CatBoost, joblib, librosa та Chart.js [24, 26, 27, 28, 29].

Загальна логіка системи передбачає взаємодію вебінтерфейсу, backend API, модулів обробки табличних і голосових даних, а також файлової підсистеми. Тому програмні засоби повинні забезпечувати зручну інтеграцію серверної логіки, клієнтських сценаріїв і зовнішніх Python-скриптів аналізу. Взаємодію основних програмних засобів системи подано на рисунку 3.1.

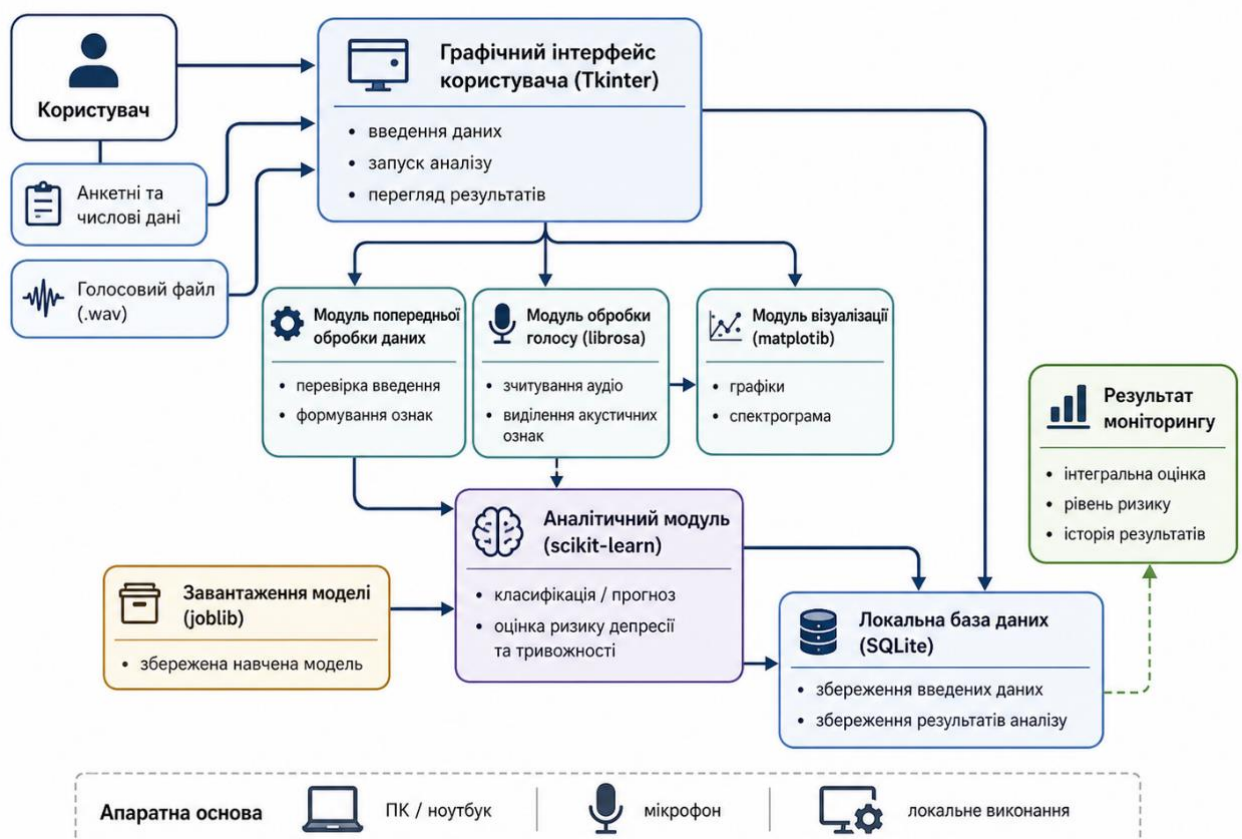


Рисунок 3.1 – Основні програмні засоби системи та їх взаємодія

З боку апаратної реалізації система не потребує спеціалізованого високопродуктивного обладнання. Основним середовищем запуску є персональний комп'ютер або ноутбук, на якому розгортається локальний вебсервер і виконуються Python-скрипти аналітичної обробки.

Практична розробка та тестування здійснювалися у локальному середовищі ХАМРР, що забезпечувало запуск Apache, роботу PHP-сторінок і взаємодію з файловою структурою проєкту [25]. Така конфігурація є достатньою для розгортання та перевірки основних сценаріїв системи.

Окремим джерелом даних у системі виступають табличні файли експорту носимих пристроїв і голосові записи у форматі WAV. Смартфон або мікрофон не виконують роль основного обчислювального вузла, а використовуються як засоби отримання й перенесення даних до системи.

Для коректної роботи голосового модуля використовуються аудіозаписи у форматі WAV, оскільки цей формат підтримується засобами обробки аудіо та дає змогу стабільно виділяти акустичні ознаки [21, 28].

Для роботи системи достатньо ноутбука середнього рівня з багатоядерним процесором, оперативною пам'яттю від 8 ГБ та вільним дисковим простором для зберігання файлів пацієнтів, артефактів моделей і службових логів.

Рекомендовані апаратні характеристики для реалізації та тестування системи наведено в таблиці 3.1.

Таблиця 3.1 – Рекомендовані апаратні характеристики для роботи системи

Компонент	Мінімальні характеристики	Рекомендовані характеристики	Призначення в системі
Процесор	2 ядра, частота від 1,8 ГГц	4 ядра, частота від 2,4 ГГц	Запуск локального вебсервера, PHP-сторінок і Python-скриптів аналізу
Оперативна пам'ять	4 ГБ	8 ГБ і більше	Обробка табличних файлів, робота бібліотек CatBoost, NumPy, pandas і librosa

Продовження Таблиці 3.1

Накопичувач	SSD або HDD, від 2 ГБ вільного місця	SSD, від 10 ГБ вільного місця	Зберігання файлів пацієнтів, голосових записів і модельних артефактів
Операційна система	Windows 10 / Linux	Windows 10/11 або сучасний Linux-дистрибутив	Підтримка локального вебсередовища, PHP та Python
Вебсервер	Apache у складі XAMPP	Apache у складі XAMPP	Обслуговування сторінок вебсистеми та API-ендпоінтів
Інтернет-з'єднання	необов'язкове для локального запуску	бажане для оновлення залежностей і документації	Локальна робота системи можлива без постійного доступу до Інтернету
Пристрій введення	клавіатура, миша	клавіатура, миша, мікрофон	Керування вебінтерфейсом і підготовка голосових записів
Аудіопристрої	базовий мікрофон	якісний мікрофон або смартфон для запису WAV	Отримання голосових даних для окремого модуля аналізу
Дисплей	екран від 1366×768	Full HD або вище	Комфортна робота зі сторінкою пацієнта, таблицями та графіками

Отже, програмно-апаратна основа системи побудована за принципом веборієнтованого виконання на персональному комп'ютері або ноутбукі з використанням локального серверного середовища, окремих Python-скриптів аналізу та засобів клієнтської візуалізації.

3.2 Програмно-технологічне забезпечення системи

Програмно-технологічне забезпечення розробленої системи реалізовано як інтегрований вебзастосунок, у межах якого поєднано інтерфейс лікаря, серверну логіку, файлову підсистему для роботи з даними пацієнтів, модулі аналітичної обробки та засоби візуалізації результатів. На відміну від ранньої локальної

реалізації, фінальний варіант проєкту орієнтований на роботу через сторінку пацієнта, де всі основні сценарії аналізу виконуються в межах єдиного web workflow. Загальну організацію програмних компонентів системи наведено на рисунку 3.2.

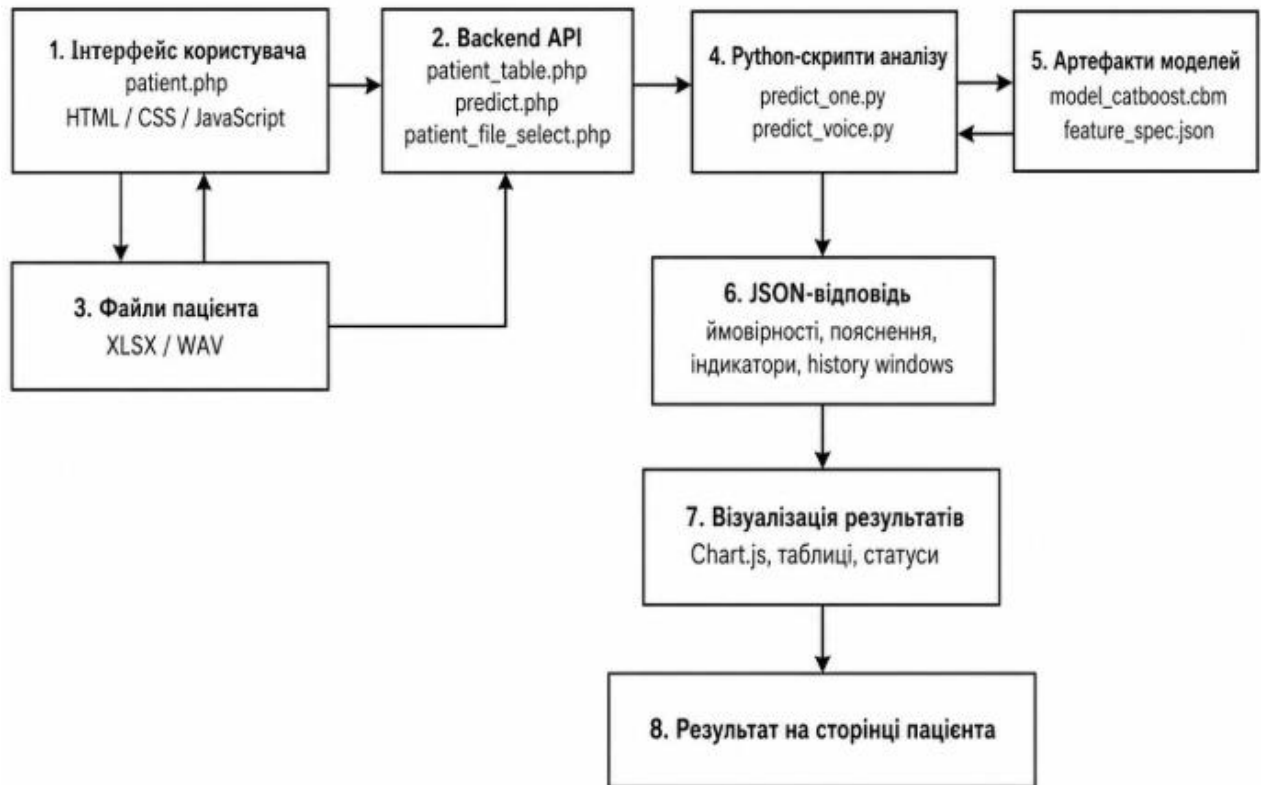


Рисунок 3.2 – Загальна організація програмних компонентів системи

У загальному вигляді система складається з кількох взаємопов’язаних блоків: блоку автентифікації та доступу лікаря, блоку керування профілями пацієнтів, підсистеми зберігання і вибору активних файлів, API-рівня для повернення структурованих JSON-відповідей, модуля аналізу структурованих табличних даних, окремого голосового модуля та блоку подання результатів. На верхньому рівні структури знаходиться лікар або інший авторизований користувач, який взаємодіє із системою через вебінтерфейс. Через інтерфейс здійснюються перегляд профілю пацієнта, завантаження файлів спостереження, вибір активного файлу, запуск аналізу та перегляд результатів. Приклад організації робочої сторінки пацієнта подано на рисунку 3.3.

```

$head = array_map(function($h) {
    $h = trim((string)$h);
    $h = str_replace(['%', ' '], ['pct', '_'], $h);
    $h = preg_replace('/^[^A-Za-z0-9_]/', '', $h);
    return strtolower($h);
}, $rows[0]);

$out['steps']      = $idx['steps']      !== null ? $num($r[$idx['steps']]) : null;
$out['calories']   = $idx['calories']   !== null ? $num($r[$idx['calories']]) : null;
$out['bpm_min']    = $idx['bpm_min']    !== null ? $num($r[$idx['bpm_min']]) : null;
$out['bpm_max']    = $idx['bpm_max']    !== null ? $num($r[$idx['bpm_max']]) : null;
$out['bpm_avg']    = $idx['bpm_avg']    !== null ? $num($r[$idx['bpm_avg']]) : null;
$out['sleep_total'] = $sleepHours;
$out['awakenings'] = $idx['awakenings'] !== null ? (int)$num($r[$idx['awakenings']]) : null;
$out['rem_pct']    = $idx['rem_pct']    !== null ? $percent($r[$idx['rem_pct']]) : null;

```

Рисунок 3.3 – Фрагмент коду обробки структурованих табличних даних пацієнта

Після ініціалізації вебсторінки система звертається до backend endpoint-ів для отримання потрібних даних. Для табличного каналу сторінка пацієнта викликає endpoint читання активного файлу, а для прогнозування — endpoint, який повертає одну JSON-відповідь із ймовірностями класів, похідними індикаторами, поясненням та історичними вікнами. Завдяки цьому логіка інтерфейсу, файлового зберігання та аналітичної обробки не змішується в одному місці, а розподіляється між окремими модулями системи. Приклад завантаження і відображення табличних даних пацієнта наведено на рисунку 3.4.

```

<form action="patient_upload.php" method="post" enctype="multipart/form-data">
    <input type="hidden" name="id" value="<?= htmlspecialchars($id) ?>">
    <input id="pfile" type="file" name="file" accept=".xlsx" required hidden>
    <label for="pfile" class="file-btn"><?= t('choose_file') ?></label>
    <button class="btn" type="submit"><?= t('upload') ?></button>
</form>

<table class="table" id="pt-table">
    <thead><tr><th><?= t('th.date') ?></th><th><?= t('th.steps') ?></th><th><?= t('th.calories') ?></th></tr></thead>
    <tbody></tbody>
</table>

```

Рисунок 3.4 – Фрагмент коду завантаження і відображення табличних даних пацієнта

У межах програмної реалізації структурованого каналу аналізу навчання моделі виконується окремим Python-скриптом `train_catboost.py`. На цьому етапі із підготовленого набору агрегованих ознак завантажуються вхідні дані, формується навчальна і тестова вибірки, створюються об'єкти `Pool` бібліотеки `CatBoost` і виконується навчання багатокласової моделі класифікації. Після завершення навчання система зберігає отриманий артефакт моделі `model_catboost.cbm` та файл специфікації ознак `feature_spec.json` для подальшого використання під час прогнозування. Фрагмент коду навчання багатокласової моделі `CatBoost` для аналізу структурованих даних наведено на рисунку 3.5.

```
feature_cols = [
    "steps_mean", "calories_mean", "bpm_avg_mean",
    "sleep_total_h_mean", "awakenings_mean", "rem_pct_mean",
]
X = df[feature_cols].copy()
y = df["label"].astype("category")

Xtr, Xte, ytr, yte = train_test_split(
    X, y, test_size=float(cfg["training"]["test_size"]),
    random_state=int(cfg["training"]["random_state"]), stratify=y
)

train_pool = Pool(Xtr, ytr)
test_pool = Pool(Xte, yte)
model = CatBoostClassifier(loss_function="MultiClass", auto_class_weights="Balanced")
model.fit(train_pool, eval_set=test_pool)
model.save_model(str(art_dir / "model_catboost.cbm"))
```

Рисунок 3.5 – Фрагмент коду навчання багатокласової моделі `CatBoost` для аналізу структурованих даних

Наведений фрагмент демонструє основну логіку офлайнового етапу побудови моделі: зчитування даних, вибір інформативних ознак, стратифікований поділ вибірки, навчання багатокласового класифікатора та збереження результатів у вигляді повторно використовуваних артефактів. Такий підхід забезпечує відокремлення етапу навчання від етапу експлуатації вебсистеми, у якій модель використовується вже у готовому вигляді без повторного перенавчання.

Після навчання моделі її використання у вебсистемі реалізується окремим скриптом прогнозування `predict_one.py`. Скрипт приймає файл із табличними даними пацієнта, виконує нормалізацію назв колонок, зчитує потрібні показники з першого аркуша Excel, обчислює агреговані середні значення та формує набір ознак у форматі, сумісному з навченою моделлю. Після цього завантажується артефакт `model_catboost.cbm`, виконується обчислення ймовірностей належності до класів «норма», «тривога», «депресія» та «ПТСР», а також формується коротке текстове пояснення результату. Такий сценарій дозволяє використовувати модель безпосередньо у вебсистемі після завантаження активного файлу пацієнта. Фрагмент коду прогнозування психоемоційного стану на основі агрегованих показників із завантаженого файлу наведено на рисунку 3.6.

```
raw_df = map_first_sheet_to_raw_measures(Path(args.file))
feats = aggregate_means(raw_df)

model = CatBoostClassifier()
model.load_model(str(MODEL_PATH))
spec = json.loads(SPEC_PATH.read_text(encoding="utf-8"))

class_order = spec.get("class_order", ["anxiety", "depression", "normal", "ptsd"])
feature_cols = spec.get("numericals", [
    "steps_mean", "calories_mean", "bpm_avg_mean",
    "sleep_total_h_mean", "awakenings_mean", "rem_pct_mean"
])

pool = Pool(feats[feature_cols], cat_features=[])
proba = model.predict_proba(pool)[0]
probs = {cls: float(round(p * 100, 2)) for cls, p in zip(class_order, proba)}
```

Рисунок 3.6 – Фрагмент коду прогнозування психоемоційного стану на основі агрегованих показників із завантаженого файлу

Таким чином, скрипт прогнозування виступає проміжною ланкою між файловою підсистемою вебзастосунку та аналітичним ядром моделі. Він забезпечує перетворення реальних завантажених вимірювань у набір ознак, узгоджений зі

специфікацією моделі, і повертає результат у структурованому вигляді для подальшого відображення у вебінтерфейсі. Приклад візуалізації результатів структурованого аналізу доцільно залишити на рисунку 3.7, а приклад таблиці похідних індикаторів або середніх значень — на рисунку 3.8. Відображення історичних змін у часових вікнах спостереження надано на рисунку 3.9.

```
const data = await fetchJson('./api/predict.php?id=' + encodeURIComponent(id));
const order = ['normal', 'anxiety', 'depression', 'ptsd'];
const values = order.map(k => Number(data.probs?.[k] ?? 0));
buildChart(values);

explEl.innerHTML = [
  `${TR.normal}: ${Number(data.probs.normal ?? 0).toFixed(2)}%`,
  `${TR.anxiety}: ${Number(data.probs.anxiety ?? 0).toFixed(2)}%`,
  `${TR.depression}: ${Number(data.probs.depression ?? 0).toFixed(2)}%`,
  `${TR.ptsd}: ${Number(data.probs.ptsd ?? 0).toFixed(2)}%`
].map(x => `- ${escapeHtml(x)}</li>`).join('');

```

Рисунок 3.7 – Фрагмент коду візуалізації результатів структурованого аналізу

```
const rows = [
  [TR.avg_steps, mean(items, 'steps'), TR.unit_steps],
  [TR.avg_calories, mean(items, 'calories'), TR.unit_kcal],
  [TR.avg_hr_avg, mean(items, 'bpm_avg'), TR.unit_bpm],
  [TR.avg_sleep_total, mean(items, 'sleep_total'), TR.unit_hour_short],
  [TR.avg_awakenings, mean(items, 'awakenings'), TR.unit_times],
  [TR.avg_rem, mean(items, 'rem_pct'), '%']
];

avgBody.innerHTML = rows.map(([label, value, unit]) => `
  <tr>
    <td>${label}</td>
    <td>${value === null ? TR.dash : `${Number(value).toFixed(2)} ${unit}`}</td>
  </tr>
`).join('');
```

Рисунок 3.8 – Фрагмент коду формування таблиці похідних індикаторів і середніх значень

```

historyBody.innerHTML = items.map(item => `
  <tr class="${trendClass(item.trend)}">
    <td>${escapeHtml(`${item.period_start || TR.dash} → ${item.period_end || TR.dash}`)}</td>
    <td>${escapeHtml(item.top || TR.dash)}</td>
    <td>${escapeHtml(fmtNum(item.top_prob, 2, '%'))}</td>
    <td>${escapeHtml(fmtNum(item.averages?.sleep_total, 2, ' ' + TR.unit_hour_short))}</td>
    <td>${escapeHtml(fmtNum(item.averages?.steps, 0, ''))}</td>
    <td>${escapeHtml(fmtNum(item.averages?.bpm_avg, 2, ' ' + TR.unit_bpm))}</td>
    <td>${escapeHtml(trendLabel(item.trend))}</td>
    <td>${escapeHtml(translateSummary(item.summary))}</td>
  </tr>
`).join('');

```

Рисунок 3.9 – Фрагмент коду відображення історичних змін у часових вікнах спостереження

Окремою важливою складовою програмної реалізації є файлова підсистема пацієнта. Вона забезпечує збереження завантажених табличних файлів, формування списку доступних записів, вибір активного файлу для поточного аналізу та очищення активної прив'язки після видалення. Саме ця логіка робить роботу з кількома наборами даних керованою та відтворюваною, оскільки система завжди точно визначає, який саме файл використовується в поточному сеансі аналізу. Фрагмент коду підсистеми керування файлами пацієнта надано на рисунку 3.10.

```

function list_patient_files(string $projectRoot, string $patientId): array {
  $dir = patient_files_dir($projectRoot, $patientId);
  if (!is_dir($dir)) return [];

  $files = [];
  foreach (['*.xlsx', '*.xls', '*.csv'] as $pattern) {
    $files = array_merge($files, glob($dir . DIRECTORY_SEPARATOR . $pattern) ?: []);
  }

  usort($files, fn($a, $b) => filemtime($b) <=> filemtime($a));
  return array_map(fn($f) => [
    'filename' => basename($f),
    'path' => $f,
    'modified_at' => date('Y-m-d H:i:s', filemtime($f) ?: time()),
    'ext' => strtolower(pathinfo($f, PATHINFO_EXTENSION)),
  ], $files);
}

```

Рисунок 3.10 – Фрагмент коду підсистеми керування файлами пацієнта

Наведений фрагмент має демонструвати створення шляху до директорії пацієнта, зчитування списку файлів, роботу з картою активних файлів та функцію визначення поточного активного запису. Це дає змогу узгодити файлову організацію з аналітичним workflow вебсистеми.

3.3 Формування та обґрунтування навчального набору даних

Для побудови моделі структурованих даних у роботі використано контрольований синтетичний набір спостережень, сформований спеціально під логіку цього проєкту. Таке рішення було прийнято не тому, що реальні дані є неважливими, а тому, що на момент розроблення не існувало доступної відкритої вибірки, яка б одночасно містила потрібний склад ознак: багатоденні дані про кроки, сон, частоту серцевих скорочень, витрату калорій, кількість пробуджень, частку REM-сну, контекстні атрибути користувача та узгоджену розмітку саме для класів «норма», «тривога», «депресія» та «ПТСР». Наявні відкриті набори зазвичай охоплюють лише окремі групи цифрових маркерів або не містять достатньо надійної діагностичної прив'язки [2–5].

Додатковим обмеженням була і власна пілотна вибірка. У межах апробації добровольці носили пристрій Xiaomi Smart Band 9 Pro у режимі 24/7 щонайменше 10 днів, що дозволило отримати реалістичні повсякденні спостереження. Проте кількість волонтерів була обмеженою, а не всі учасники мали підтверджений спеціалістом клінічний діагноз. За таких умов використання лише реальних записів як основи для навчання багатокласової моделі було б ненадійним: вибірка залишалася б малою, незбалансованою і недостатньо коректно розміченою. Тому волонтерські дані застосовувалися насамперед для перевірки працездатності інтерфейсу, файлового workflow, коректності обчислень та стрес-тестування конвеєра оброблення, зокрема на випадках пропущених днів, порожніх значень і шумів у даних.

Синтетичний файл навчальних даних було сформовано у вигляді таблиці `mental_state_dataset_400p_10d.csv`. Він містить 4000 рядків, що відповідають 400

профілям спостереження по 10 днів на кожен профіль, та 53 інформативні колонки. У наборі збалансовано представлено чотири класи: normal, anxiety, depression і ptsd — по 1000 щоденних записів для кожного класу. Таблиця включає демографічні та контекстні параметри користувача, сирі щоденні показники, очікувані базові значення для конкретного профілю, а також похідні співвідношення та відхилення. Таким чином модель отримує не просто окремі випадкові числа, а структурований опис поведінки користувача у часі.

Формування синтетичних профілів відбувалося за принципом AI-assisted, але rule-guided генерації. Це означає, що ШІ не створював дані довільно, а працював у межах заданих людиною критеріїв. Спочатку для кожного віртуального пацієнта визначалися вік, стать, тип навантаження, зріст, вага, індекс маси тіла, наявність змінного режиму та базові очікувані показники. Далі задавалися характерні відхилення для певного класу стану, після чого модель генерації створювала 10-денне щоденне спостереження з невеликими коливаннями, правдоподібним шумом і похідними показниками. Такий підхід дозволив поєднати гнучкість генеративного інструмента з контролем дослідника над логікою набору.

Критерії генерації спиралися на узагальнення з наукових джерел, а не на інтуїтивне припущення. Із систематичних оглядів digital phenotyping та wearable AI було взято головну ідею, що для попереднього оцінювання психоемоційного стану найбільш інформативними є саме показники активності, сну та серцевого ритму, причому вони мають аналізуватися не ізольовано, а в поєднанні [2–5]. З цих джерел для генерації було запозичено сам принцип побудови патернів: знижена активність і порушення сну частіше асоціюються з депресивними проявами, підвищені показники пульсу та більш фрагментований сон є типовішими для тривожних станів, а поєднання коротшого сну, більшої кількості пробуджень і вищого фізіологічного напруження є характерним для ПТСР. Отже, літературні джерела використовувалися не для копіювання конкретних чисел, а для задання правдоподібних напрямів і меж відхилення ознак.

Наприклад, один із базових текстових сценаріїв генерації для класу normal описував профіль так: «Сформууй 10 днів спостережень для жінки\чоловіка з

сидячим або змішаним навантаженням. Дані мають бути стабільними, без різких провалів активності, із відносно регулярним сном, помірною кількістю кроків, низькою кількістю пробуджень і фізіологічними показниками в межах умовної норми». Для класу *depression* сценарій формулювався інакше: «Згенеруй профіль жінки\чоловіка із нижчою повсякденною активністю, частішими ознаками малорухомості, помірним зниженням частки REM-сну, більшою варіативністю сну та показниками, що не виходять за фізіологічно правдоподібні межі». Для класу *anxiety* акцент зміщувався на більш високий середній пульс, частіші пробудження та нестабільність, а для *ptsd* — на коротший сон, більшу кількість нічних пробуджень, вищу фізіологічну напругу та виразніші відхилення від очікуваного профілю.

Відповідно, у відповідях генеративної моделі очікувався не вільний текст, а таблицеподібний набір значень із чітко заданими полями. Умовно відповідь мала вигляд: дата, кроки, калорії, *active_minutes*, *sleep_total_min*, *awakenings*, *rem_pct*, *hr_rest*, *hr_avg*, *hr_max*, *hr_min*, а також контекстні атрибути профілю. Після цього згенеровані записи додатково перевірялися, очищувалися і збагачувалися похідними ознаками — *expected_steps*, *expected_sleep_total_min*, *steps_ratio*, *sleep_ratio*, *hr_avg_ratio*, *steps_gap*, *sleep_gap_min* та іншими. Отже, фінальний CSV є не сирим продуктом генерації, а результатом кількох послідовних етапів: сценарного опису, генерації, фільтрації, структуризації та перетворення у формат, придатний для машинного навчання.

Суттєвою перевагою такого підходу стало те, що синтетичні дані виявилися достатньо близькими до реальних за формою та логікою розподілу. Вони не копіювали реальних людей, але відтворювали типові співвідношення між активністю, сном і серцевим ритмом. Наприклад, у класі *depression* середні значення кроків і REM були нижчими, ніж у класі *normal*; у класах *anxiety* та *ptsd* виявлялися вищі значення пульсу і більша фрагментація сну; для *normal*-профілів залишався більш збалансований і стабільний патерн. Саме така внутрішня правдоподібність робила набір корисним для відпрацювання логіки класифікації.

Окремо слід підкреслити, що модель, побудована на цьому наборі, не позиціонується як клінічна. На поточному етапі основною метою було створити систему, здатну коректно аналізувати, структурувати та класифікувати цифрові дані у межах дослідницько-демонстраційного сценарію. Тобто йшлося насамперед про перевірку архітектури, логіки ознак, роботи аналітичного конвеєра та форми подання результату, а не про побудову остаточно валідованого медичного інструмента.

Таким чином, використання генеративних даних було найкращим рішенням саме для цього етапу роботи: воно дозволило отримати достатній обсяг спостережень, збалансувати класи, закласти в набір логіку, підтверджену літературою, і водночас уникнути некоректного перенавчання на дуже малій і слабо розміченій реальній вибірці. У подальшому цей проєкт може бути розвинений у напрямі накопичення більшого масиву реальних даних, зібраних у стандартизованих умовах та доповнених клінічно підтвердженою розміткою. Саме такий наступний крок дасть змогу підвищити точність моделі, посилити її зовнішню валідність і наблизити систему до більш практично значущого використання.

3.4 Тестування системи

Після завершення реалізації основних компонентів системи було проведено перевірку її працездатності в ключових користувацьких сценаріях. Тестування охоплювало не лише базовий запуск аналізу, а й усю логіку веборієнтованого workflow з файлами пацієнтів та окремим голосовим каналом.

Спочатку виконувалася перевірка створення та редагування профілю пацієнта. На цьому етапі оцінювалося коректне збереження анкетних даних, відкриття сторінки пацієнта та відображення пов'язаних із ним аналітичних секцій. Приклад додавання та відкриття профілю пацієнта наведено на рисунку 3.11.

Edit patient

×

Name

Anastasia Horok.

Date of birth

19.03.2011

📅

Sex

Female

▼

Weight (kg)

55

Height (cm)

160

Profession

▼

Save

Рисунок 3.11 – Додавання та відкриття профілю пацієнта

Далі виконувалася перевірка завантаження вхідних даних. Для голосового каналу перевірялося завантаження WAV-файлу та його поява у відповідному списку. Приклад завантаження табличного файлу подано на рисунку 3.12.

Daily data

Choose file

No file chosen

Upload

DATE	STEPS	CALORIES	BPM (MIN)	BPM (MAX)	BPM (AVG)	SLEEP (TOTAL)	AWAKENINGS	REM %
2025-01-17	24188	888	49	194	62.3	0	0	0%
2025-01-18	8440	328	49	168	61.5	0	0	0%
2025-01-19	24899	954	45	121	61.6	8.58	0	8.7%
2025-01-20	12424	581	46	135	63.8	7.6	0	14.9%
2025-01-21	15941	684	47	129	62	6.47	0	23.5%
2025-01-22	2771	130	49	141	63.3	8.55	0	26.7%
2025-01-23	18484	746	50	134	70.1	16.5	1	17.4%
2025-01-24	10134	405	49	133	65.1	6.22	0	15.8%
2025-01-25	5997	256	45	131	60.1	10.03	0	27.7%

Patient files

FILE NAME	TYPE	MODIFIED	SIZE	ACTIVE	ACTIONS
anastasia_1777038958.xlsx	XLSX	2026-04-24 15:55:58	4.3 KB	✓	Download Use for analysis Delete
my_watch_data_1777038939.xlsx	XLSX	2026-04-24 15:55:39	28.3 KB	—	Download Use for analysis Delete
Ideal_Normal_20yo_Female_1776950417.xlsx	XLSX	2026-04-23 15:20:17	27.6 KB	—	Download Use for analysis Delete

Рисунок 3.12 – Завантаження табличного файлу

Після завантаження даних проводилася перевірка основного сценарію аналізу. Для структурованого каналу оцінювалися обчислення середніх значень, похідних індикаторів і ймовірностей класів, а також побудова пояснювального блоку. Для історичної частини перевірялося формування часових вікон і коректне відображення тенденцій зміни стану. Результат аналізу структурованих даних наведено на рисунку 3.13.

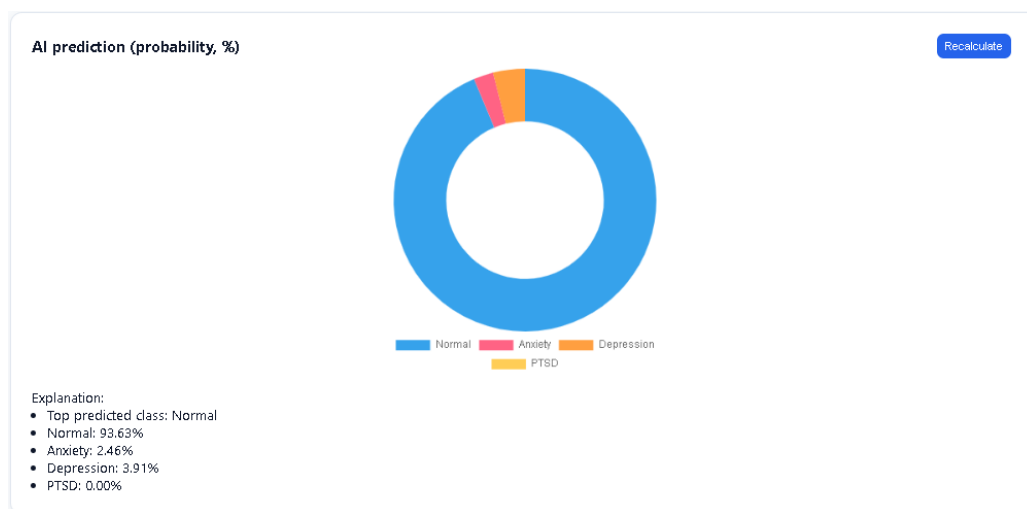


Рисунок 3.13 – Результат аналізу структурованих даних

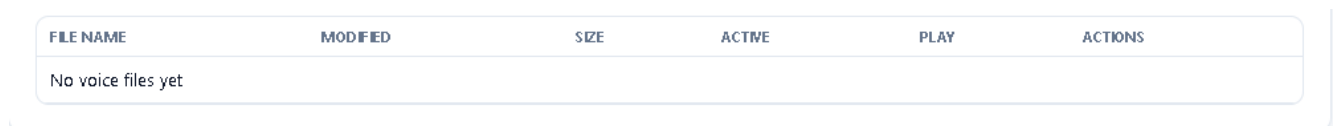
Окремо виконувалася перевірка керування файлами пацієнта: зміна активного файлу, вивантаження файлу на пристрій користувача, видалення непотрібного файлу та повторне оновлення сторінки після виконаної дії. Це було критично важливо для відтворюваності аналітичного процесу. Керування файлами пацієнта та активним вибором показано на рисунку 3.14.

Patient files

FILE NAME	TYPE	MODIFIED	SIZE	ACTIVE	ACTIONS
anastasia_1777038958.xlsx	XLSX	2026-04-24 15:55:58	4.3 KB	☒	Download Use for analysis Delete
my_watch_data_1777038939.xlsx	XLSX	2026-04-24 15:55:39	28.3 KB	☐	Download Use for analysis Delete
Ideal_Normal_20yo_Female_1776950417.xlsx	XLSX	2026-04-23 15:20:17	27.6 KB	☐	Download Use for analysis Delete

Рисунок 3.14 – Керування файлами пацієнта та активним вибором

Також перевірялася реакція системи на помилкові дії: спроба запуску аналізу без активного файлу, завантаження файлу некоректного формату, неповні або пошкоджені дані, а також помилки під час голосового аналізу. У всіх цих випадках система повинна була повертати зрозуміле повідомлення без аварійного завершення роботи. Приклад перевірки обробки помилкових сценаріїв наведено на рисунку 3.15.



FILE NAME	MODIFIED	SIZE	ACTIVE	PLAY	ACTIONS
No voice files yet					

Рисунок 3.15 – Перевірка обробки помилкових сценаріїв

У результаті проведеного тестування було підтверджено працездатність основних функцій розробленої системи. Перевірка показала, що реалізовані компоненти коректно підтримують профілі пацієнтів, файловий workflow, аналіз структурованих і голосових даних, візуалізацію імовірностей та відображення динаміки змін у часі.

ВИСНОВКИ

1. Проведено аналіз предметної області цифрового спостереження за психоемоційним станом людини. Для цього було розглянуто цифрові маркери активності, сну, серцевого ритму та голосових характеристик, а також узагальнено сучасні підходи digital phenotyping. У результаті визначено, що для попереднього оцінювання ризику психоемоційних порушень доцільно поєднувати кілька непрямих маркерів, а не покладатися лише на один показник.

2. Виконано огляд існуючих рішень у сферах мобільних застосунків, wearable-пристроїв, систем аналізу мовлення та цифрового фенотипування. Аналіз показав, що наявні підходи або орієнтовані на окремі групи ознак, або не забезпечують поєднання пацієнтського профілю, керування файлами, прогнозування та перегляду динаміки у межах одного інтерфейсу. Це стало підставою для вибору веборієнтованої архітектури з окремими структурованим і голосовим каналами аналізу.

3. Спроектовано загальну структуру програмної системи та алгоритм її функціонування. Для цього визначено склад основних компонентів, зокрема вебінтерфейс для лікаря, підсистему профілів пацієнтів, файловий workflow, backend API, модулі аналітичної обробки та засоби візуалізації результатів. У результаті отримано цілісну архітектуру, яка забезпечує послідовний шлях даних від завантаження файлу до подання прогнозу, пояснення й історії змін у часі.

4. Розроблено модель формування прогнозних імовірностей для структурованих і голосових даних. Для структурованого каналу використано багатокласову модель CatBoost, а для голосового каналу — виділення ознак MFCC, RMS та ZCR з подальшим класифікаційним оцінюванням. У результаті система формує розподіл імовірностей між класами «норма», «тривога», «депресія» та «ПТСР», а також подає пояснення й допоміжні індикатори для інтерпретації лікарем.

5. Спроектовано інформаційне забезпечення системи з урахуванням профілів пацієнтів, файлових сутностей і аналітичних артефактів. Для цього реалізовано

структуру збереження табличних і голосових файлів, механізм вибору активного файлу та формування JSON-відповідей для фронтенду. Це забезпечило відтворюваність аналітичного процесу, зменшило ризик помилкового аналізу не того файлу та дало змогу організовано працювати з історією спостережень.

6. Реалізовано програмно-технологічну частину вебсистеми та проведено тестування її основних сценаріїв. У межах реалізації створено сторінку пацієнта, модулі завантаження, вибору, видалення та вивантаження файлів, засоби побудови таблиць, графіків, пояснень і часових вікон. Тестування підтвердило коректну роботу системи у ключових сценаріях попереднього оцінювання та показало придатність розробленого рішення як допоміжного інструменту для скринінгу, динамічного спостереження та підготовки до подальшого професійного розгляду стану пацієнта.

Основні положення кваліфікаційної роботи апробовано у науковій роботі «Програмна система попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача» [33], а підтверджувальні матеріали наведено в Додатку Г.

Отже, у кваліфікаційній роботі було розроблено, реалізовано та протестовано програмну систему попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача. Запропоноване рішення має дослідницько-практичний характер і може бути використане як допоміжний інструмент для демонстрації принципів цифрового моніторингу психоемоційного стану та подальшого розвитку систем такого типу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mohr D. C., Zhang M., Schueller S. M. Personal sensing: understanding mental health using ubiquitous sensors and machine learning. *Annual Review of Clinical Psychology*. 2017. Vol. 13. P. 23–47. DOI: 10.1146/annurev-clinpsy-032816-044949.
2. Ben-Zeev D., Scherer E. A., Wang R., Xie H., Campbell A. T. Next-generation psychiatric assessment: using smartphone sensors to monitor behavior and mental health. *Psychiatric Rehabilitation Journal*. 2015. Vol. 38, № 3. P. 218–226. DOI: 10.1037/prj0000130.
3. Kamath J., Leon Barriera R., Jain N., Keisari E., Wang B. Digital phenotyping in depression diagnostics: integrating psychiatric and engineering perspectives. *World Journal of Psychiatry*. 2022. Vol. 12, № 3. P. 393–409. DOI: 10.5498/wjp.v12.i3.393.
4. Maatoug R., Oudin A., Adrien V. та ін. Digital phenotype of mood disorders: a conceptual and critical review. *Frontiers in Psychiatry*. 2022. Vol. 13. Art. 895860. DOI: 10.3389/fpsyt.2022.895860.
5. Kappen M., Vanderhasselt M.-A., Slavich G. M. Speech as a promising biosignal in precision psychiatry. *Neuroscience and Biobehavioral Reviews*. 2023. Vol. 148. Art. 105121. DOI: 10.1016/j.neubiorev.2023.105121.
6. Low D. M., Bentley K. H., Ghosh S. S. Automated assessment of psychiatric disorders using speech: a systematic review. *Laryngoscope Investigative Otolaryngology*. 2020. Vol. 5, № 1. P. 96–116. DOI: 10.1002/lio2.354.
7. Choi A., Ooi A., Lottridge D. Digital phenotyping for stress, anxiety, and mild depression: systematic literature review. *JMIR mHealth and uHealth*. 2024. Vol. 12. Art. e40689. DOI: 10.2196/40689.
8. Beames J. R., Han J., Shvetsov A. та ін. Use of smartphone sensor data in detecting and predicting depression and anxiety in young people (12–25 years): a scoping review. *Heliyon*. 2024. Vol. 10, № 15. Art. e35472. DOI: 10.1016/j.heliyon.2024.e35472.
9. Shin J., Bae S. M. Use of voice features from smartphones for monitoring depressive disorders: scoping review. *Digital Health*. 2024. Vol. 10. Art. 20552076241261920. DOI: 10.1177/20552076241261920.

10. Jung H. W., Kim D. Y., Lee I. та ін. Key features of digital phenotyping for monitoring mental disorders: systematic review. *Journal of Medical Internet Research*. 2025. Vol. 27. Art. e77331. DOI: 10.2196/77331.

11. Insel T. R. Digital phenotyping: technology for a new science of behavior. *JAMA*. 2017. Vol. 318, № 13. P. 1215–1216. DOI: 10.1001/jama.2017.11295.

12. Onnela J.-P., Rauch S. L. Harnessing smartphone-based digital phenotyping to enhance behavioral and mental health. *Neuropsychopharmacology*. 2016. Vol. 41. P. 1691–1696. DOI: 10.1038/npp.2016.7.

13. Saeb S., Zhang M., Karr C. J., Schueller S. M., Corden M. E., Kording K. P., Mohr D. C. Mobile phone sensor correlates of depressive symptom severity in daily-life behavior: an exploratory study. *Journal of Medical Internet Research*. 2015. Vol. 17, № 7. Art. e175. DOI: 10.2196/jmir.4273.

14. Wang R., Chen F., Chen Z., Li T., Harari G., Tignor S., Zhou X., Ben-Zeev D., Campbell A. T. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2014. P. 3–14. DOI: 10.1145/2632048.2632054.

15. Martinez-Martin N., Greely H. T., Cho M. K. Ethical development of digital phenotyping tools for mental health applications: Delphi study. *JMIR mHealth and uHealth*. 2021. Vol. 9, № 7. Art. e27343. DOI: 10.2196/27343.

16. Torous J., Bucci S., Bell I. H., Kessing L. V., Faurholt-Jepsen M., Whelan P., Carvalho A. F., Keshavan M., Linardon J., Firth J. The growing field of digital psychiatry: current evidence and the future of apps, social media, chatbots, and virtual reality. *World Psychiatry*. 2021. Vol. 20, № 3. P. 318–335. DOI: 10.1002/wps.20883.

17. Kroenke K., Spitzer R. L., Williams J. B. W. The PHQ-9: validity of a brief depression severity measure. *Journal of General Internal Medicine*. 2001. Vol. 16, № 9. P. 606–613. DOI: 10.1046/j.1525-1497.2001.016009606.x.

18. Spitzer R. L., Kroenke K., Williams J. B. W., Löwe B. A brief measure for assessing generalized anxiety disorder: the GAD-7. *Archives of Internal Medicine*. 2006. Vol. 166, № 10. P. 1092–1097. DOI: 10.1001/archinte.166.10.1092.

19. Plummer F., Manea L., Trepel D., McMillan D. Screening for anxiety disorders with the GAD-7 and GAD-2: a systematic review and diagnostic metaanalysis. *General Hospital Psychiatry*. 2016. Vol. 39. P. 24–31. DOI: 10.1016/j.genhosppsych.2015.11.005.
20. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O. та ін. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.
21. McFee B., Raffel C., Liang D., Ellis D. P. W., McVicar M., Battenberg E., Nieto O. librosa: audio and music signal analysis in Python. *Proceedings of the 14th Python in Science Conference*. 2015. P. 18–25. DOI: 10.25080/Majora-7b98e3ed-003.
22. Hunter J. D. Matplotlib: a 2D graphics environment. *Computing in Science & Engineering*. 2007. Vol. 9, № 3. P. 90–95. DOI: 10.1109/MCSE.2007.55.
23. Harris C. R., Millman K. J., van der Walt S. J., Gommers R., Virtanen P., Cournapeau D. та ін. Array programming with NumPy. *Nature*. 2020. Vol. 585. P. 357–362. DOI: 10.1038/s41586-020-2649-2.
24. The PHP Group. PHP Documentation. Official documentation. URL: <https://www.php.net/docs.php> (дата звернення: 01.06.2026).
25. Apache Friends. XAMPP. Official website. URL: <https://www.apachefriends.org/index.html> (дата звернення: 01.06.2026).
26. Chart.js contributors. Chart.js Documentation. Official documentation. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 01.06.2026).
27. CatBoost developers. CatBoost Documentation. Official documentation. URL: <https://catboost.ai/docs/en/> (дата звернення: 01.06.2026).
28. Librosa development team. librosa Documentation. Official documentation. URL: <https://librosa.org/doc/latest/index.html> (дата звернення: 01.06.2026).
29. NumPy developers. NumPy Documentation. Official documentation. URL: <https://numpy.org/doc/> (дата звернення: 01.06.2026).
30. Островерхов В. М., Біловус Л. І., Возьний К. З., Луцишин О. О., Монастирський Г. Л., Надвиничний С. А., Питель С. В., Шандрук С. К. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання

кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

31. Pezoulas V. C., Karanasiou G. S., Kanterakis G., Bartsocas C., Cavouras D. Synthetic data generation methods in healthcare: a review on open challenges, quality control and utility evaluation. *Journal of Biomedical Informatics*. 2024. Vol. 155. Art. 104544. DOI: 10.1016/j.jbi.2024.104544.

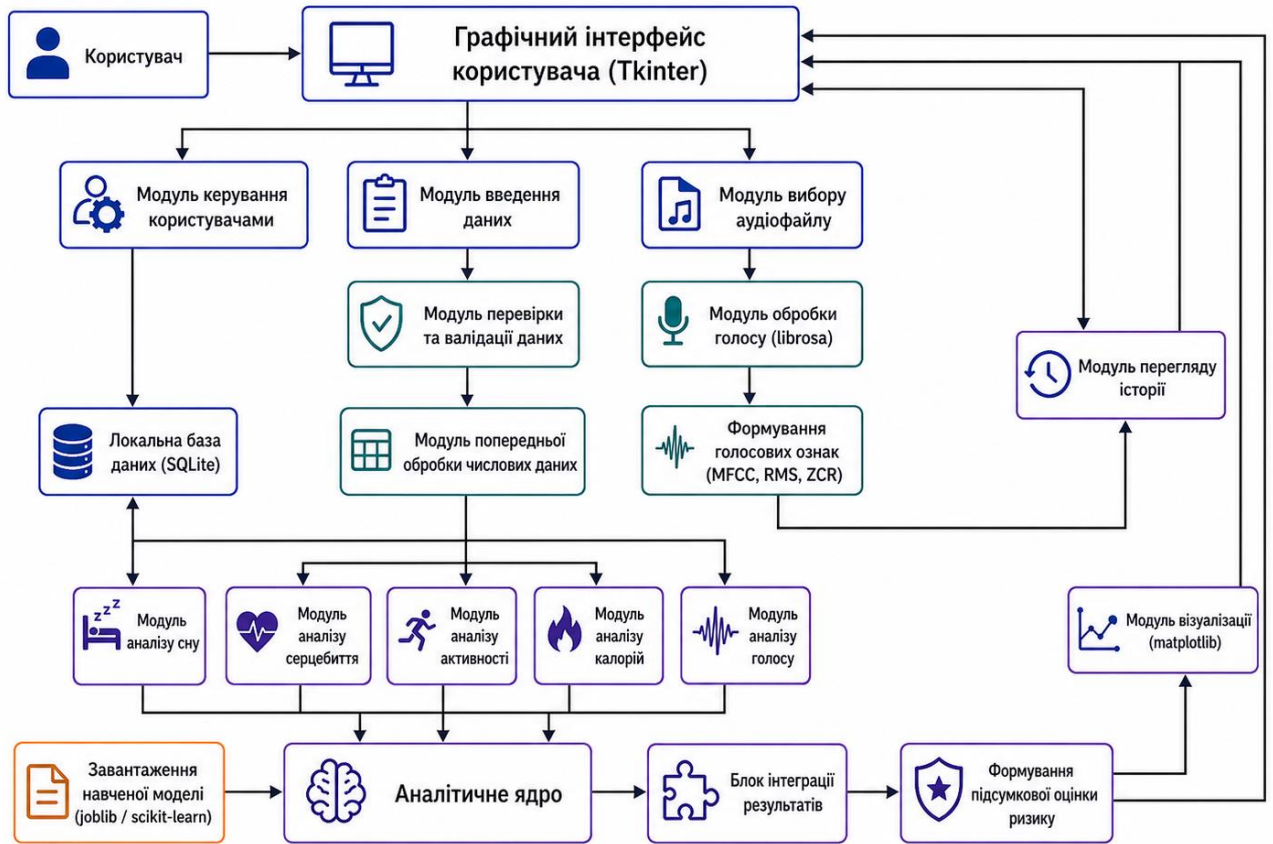
32. Mwigereri D. G., et al. Synthetic data generation of health and demographic tabular data using open-source large language models: comparative quality evaluation. *Frontiers in Artificial Intelligence*. 2025. Vol. 8. Art. 1539027. DOI: 10.3389/frai.2025.1539027.

33. Горохівська Л. М. Програмна система попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача // Теорія модернізації в контексті сучасної світової науки : матеріали VII Міжнародної наукової конференції, 12.06.2026, Чернігів, Україна. Чернігів, 2026. С. 323–328. DOI: 10.62731/mcnd-12.06.2026. URL: <https://archives.mcnd.org.ua/index.php/conference-proceeding/issue/view/12.06.2026>.

34. Комар М. П., Саченко А. О., Васильків Н. М., Гладій Г. М., Коваль В. С., Ліп'яніна-Гончаренко Х. В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

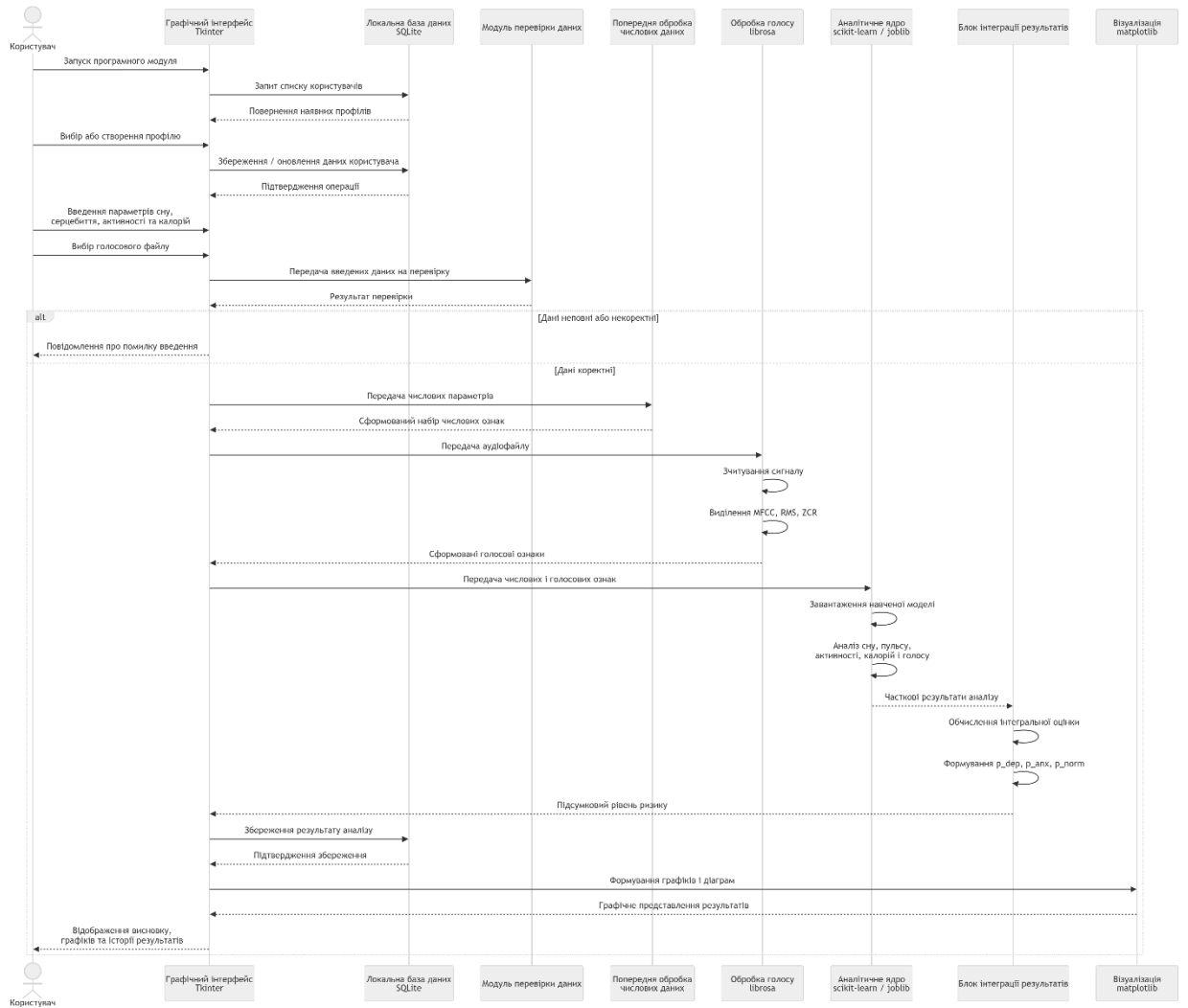
Додаток А

Розширена архітектура модуля



Додаток Б

Діаграма послідовності програмного модуля



Додаток В

Код програмно апаратного модуля

```
1 import os
2 import numpy as np
3 import pandas as pd
4 import librosa
5 import sqlite3
6 import tkinter as tk
7 from tkinter import filedialog, messagebox
8 import joblib
9 import matplotlib.pyplot as plt
10 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
11 from matplotlib.figure import Figure
12 import matplotlib.gridspec as gridspec
13 from tkinter import ttk
14
15 BASE_DIR = os.path.dirname(os.path.abspath(__file__))
16 audio_file_path = None
17 import warnings
18 warnings.filterwarnings("ignore", category=UserWarning, module="sklearn")
19 warnings.filterwarnings("ignore", message=".*model_persistence.*")
20
21 DB_PATH = os.path.join(BASE_DIR, "mental_analysis.db")
22
23 def _conn():
24     return sqlite3.connect(DB_PATH)
25
26 def _column_exists(table, col):
27     with _conn() as conn:
28         c = conn.cursor()
29         c.execute(f"PRAGMA table_info({table})")
30         cols = [r[1] for r in c.fetchall()]
31     return col in cols
32
33 def init_db():
34     with _conn() as conn:
35         c = conn.cursor()
36         # users
37         c.execute("""
38             CREATE TABLE IF NOT EXISTS users(
39                 id INTEGER PRIMARY KEY AUTOINCREMENT,
40                 name TEXT NOT NULL,
41                 age INTEGER,
42                 height REAL,
43                 weight REAL,
44                 gender TEXT
```

```

44         gender TEXT
45     );
46 """
47 # Міграція users: додаємо service_status, occupation_code якщо їх ще немає
48 if not _column_exists("users", "service_status"):
49     c.execute("ALTER TABLE users ADD COLUMN service_status TEXT") # 'military'/'civilian'
50 if not _column_exists("users", "occupation_code"):
51     c.execute("ALTER TABLE users ADD COLUMN occupation_code TEXT") # FK → occupations.code
52
53 # observations
54 c.execute("""
55     CREATE TABLE IF NOT EXISTS observations (
56         id INTEGER PRIMARY KEY AUTOINCREMENT,
57         user_id INTEGER,
58         sleep_duration REAL,
59         rem_phase REAL,
60         awakenings INTEGER,
61         latency REAL,
62         slow_wave REAL,
63         heartrate REAL,
64         activity INTEGER,
65         calories INTEGER,
66         voice_state TEXT,
67         sleep_disorder INTEGER,
68         hr_state TEXT,
69         activity_state TEXT,
70         calories_state TEXT,
71         final_state TEXT,
72         FOREIGN KEY(user_id) REFERENCES users(id)
73     );
74 """)
75
76 # assessments — історія ймовірностей
77 c.execute("""
78     CREATE TABLE IF NOT EXISTS assessments (
79         id INTEGER PRIMARY KEY AUTOINCREMENT,
80         user_id INTEGER,
81         ts TEXT,
82         p_dep REAL,
83         p_anx REAL,
84         p_norm REAL,
85         model version TEXT,
86         FOREIGN KEY(user_id) REFERENCES users(id)
87     );
88 """)
89
90 # occupations — довідник професій/ролей (військових і цивільних)
91 c.execute("""
92     CREATE TABLE IF NOT EXISTS occupations (
93         code TEXT PRIMARY KEY,
94         name TEXT NOT NULL,
95         is_military INTEGER DEFAULT 0
96     );
97 """)
98
99 # ref_norms — еталонні середні та SD для підгруп
100 c.execute("""
101     CREATE TABLE IF NOT EXISTS ref_norms (
102         occupation_code TEXT,
103         sex TEXT, -- 'чоловік'/'жінка' або NULL
104         age_min INTEGER, -- межі віку, напр. 18, 25, 35 ...
105         age_max INTEGER, -- виключна верхня межа (наприклад, 25, 35, 45 ...)
106         metric TEXT, -- 'sleep_duration', 'awakenings', 'latency', 'slow_wave', 'heartrate', 'activity', 'calories'
107         mu REAL,
108         sigma REAL,
109         PRIMARY KEY (occupation_code, sex, age_min, age_max, metric),
110         FOREIGN KEY (occupation_code) REFERENCES occupations(code)
111     );
112 """)
113
114 conn.commit()
115
116 seed_occupations()
117
118 def seed_occupations():
119     """Базовий перелік — можете редагувати під свої задачі."""
120     base = [
121         ("mil_infantry", "Військовий — Піхота (польові умови)", 1),
122         ("mil_logistics", "Військовий — Логістика/Водій", 1),
123         ("mil_medic", "Військовий — Медик", 1),
124         ("mil_staff", "Військовий — Штаб/Офіс", 1),
125         ("civ_office", "Цивільний — Офіс", 0),
126         ("civ_manual", "Цивільний — Фізична праця", 0),
127         ("civ_driver", "Цивільний — Водій/Зміни", 0),
128     ]

```

```

127         ("civ_driver", "Цивільний – Водій/Зміни", 0),
128     ]
129     with _conn() as conn:
130         c = conn.cursor()
131         c.execute("SELECT COUNT(*) FROM occupations")
132         if c.fetchone()[0] == 0:
133             c.executemany("INSERT INTO occupations(code, name, is_military) VALUES (?, ?, ?)", base)
134             conn.commit()
135
136     def get_occupations():
137         with _conn() as conn:
138             c = conn.cursor()
139             c.execute("SELECT code, name FROM occupations ORDER BY is_military DESC, name")
140             return c.fetchall()
141
142     # ---- CRUD для users ----
143     def add_user(name, age=None, height=None, weight=None, gender=None,
144                 service_status="civilian", occupation_code=None):
145         with _conn() as conn:
146             c = conn.cursor()
147             c.execute(
148                 "INSERT INTO users(name, age, height, weight, gender, service_status, occupation_code) "
149                 "VALUES (?, ?, ?, ?, ?, ?, ?)",
150                 (str(name),
151                  None if age is None else int(age),
152                  None if height is None else float(height),
153                  None if weight is None else float(weight),
154                  None if gender is None else str(gender),
155                  str(service_status),
156                  None if occupation_code in (None, "", "-") else str(occupation_code))
157             conn.commit()
158             return c.lastrowid
159
160     def delete_user(user_id):
161         with _conn() as conn:
162             c = conn.cursor()
163             c.execute("DELETE FROM observations WHERE user_id=?", (int(user_id),))
164             c.execute("DELETE FROM assessments WHERE user_id=?", (int(user_id),))
165             c.execute("DELETE FROM users WHERE id=?", (int(user_id),))
166             conn.commit()
167
168     def get_users():
169         with _conn() as conn:
170
169     def get_users():
170         with _conn() as conn:
171             c = conn.cursor()
172             c.execute("SELECT id, name, age, height, weight, gender, service_status, occupation_code FROM users ORDER BY id")
173             return c.fetchall()
174
175     def get_user_data(user_id):
176         with _conn() as conn:
177             c = conn.cursor()
178             c.execute("SELECT name, age, height, weight, gender, service_status, occupation_code FROM users WHERE id=?", (int(user_id),))
179             row = c.fetchone()
180             if not row:
181                 return {}
182             name, age, height, weight, gender, service_status, occupation_code = row
183             occ_name = occupation_code
184             if occupation_code:
185                 with _conn() as conn:
186                     c = conn.cursor()
187                     c.execute("SELECT name FROM occupations WHERE code=?", (occupation_code,))
188                     occ_row = c.fetchone()
189                     if occ_row: occ_name = occ_row[0]
190             return {
191                 "ПІБ": name,
192                 "Вік": age,
193                 "Ріст (см)": height,
194                 "Вага (кг)": weight,
195                 "Стать": gender,
196                 "Статус служби": "Військовий" if (service_status or "").lower().startswith("mil") else "Цивільний",
197                 "Професія/Роль": occ_name or "-"
198             }
199
200     def save_observation(
201         user_id, sleep_duration, rem_phase, awakenings, latency, slow_wave,
202         heartrate, activity, calories,
203         voice_state, sleep_disorder, hr_state, activity_state, calories_state, final_state
204     ):
205         with _conn() as conn:
206             c = conn.cursor()
207             c.execute("""
208                 INSERT INTO observations (
209                     user_id, sleep_duration, rem_phase, awakenings, latency, slow_wave,
210                     heartrate, activity, calories, voice_state, sleep_disorder, hr_state,
211                     activity_state, calories_state, final_state

```

```

212         activity_state, calories_state, final_state
213     ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
214
215     """
216     (
217         int(user_id),
218         float(sleep_duration), float(rem_phase), int(awakenings),
219         float(latency), float(slow_wave), float(heartrate),
220         int(activity), int(calories),
221         str(voice_state), int(sleep_disorder), str(hr_state),
222         str(activity_state), str(calories_state), str(final_state)
223     )
224     conn.commit()
225
226 def get_user_observations(user_id):
227     with _conn() as conn:
228         c = conn.cursor()
229         c.execute("""
230             SELECT sleep_duration, rem_phase, awakenings, latency, slow_wave,
231                 heartrate, activity, calories, voice_state, sleep_disorder,
232                 hr_state, activity_state, calories_state, final_state
233             FROM observations
234             WHERE user_id=?
235             ORDER BY id DESC
236             """, (int(user_id),))
237         return c.fetchall()
238
239 def save_assessment(user_id, p_dep, p_anx, p_norm, model_version="step2"):
240     from datetime import datetime
241     with _conn() as conn:
242         c = conn.cursor()
243         c.execute("""
244             INSERT INTO assessments (user_id, ts, p_dep, p_anx, p_norm, model_version)
245             VALUES (?, ?, ?, ?, ?, ?)
246             """, (int(user_id), datetime.utcnow().isoformat(timespec="seconds"),
247                 float(p_dep), float(p_anx), float(p_norm), str(model_version)))
248         conn.commit()
249
250 def get_last_probs(user_id):
251     with _conn() as conn:
252         c = conn.cursor()
253         c.execute(
254             "SELECT p_dep, p_anx, p_norm FROM assessments WHERE user_id=? ORDER BY id DESC LIMIT 1",
255             (int(user_id),)
256         )
257         row = c.fetchone()
258         return (np.array(row, dtype=float) if row and all(v is not None for v in row) else None)
259
260 # ---- ймовірнісні помічники (15/25/60 + згладжування) ----
261 def _softmax(x):
262     x = np.asarray(x, dtype=float); x = x - np.max(x)
263     e = np.exp(x); return e / e.sum()
264
265 def compute_tri_probs(prior=(0.15, 0.25, 0.60),
266                       voice_state=None, sleep_disorder=None, hr_state=None,
267                       activity_state=None, calories_state=None,
268                       available_modules=5, temperature=1.0):
269     prior = np.asarray(prior, dtype=float); prior = prior / prior.sum()
270     s_dep = 0.0; s_anx = 0.0; s_norm = 0.0
271
272     if voice_state:
273         vs = str(voice_state).lower()
274         if vs in ("depression", "депресія"): s_dep += 1.0
275         elif vs in ("anxiety", "тривога"): s_anx += 1.0
276         elif vs == "normal": s_norm += 0.5
277
278     if sleep_disorder is not None:
279         if int(sleep_disorder) == 1: s_dep += 0.6; s_anx += 0.4
280         else: s_norm += 0.3
281
282     if hr_state:
283         hs = str(hr_state).lower()
284         if hs in ("depression", "низька hrv", "ріпотонус"): s_dep += 0.6
285         elif hs in ("anxiety", "тахікардія"): s_anx += 0.6
286         elif hs == "normal": s_norm += 0.3
287
288     if activity_state:
289         ac = str(activity_state).lower()
290         if ac in ("depression", "низька активність"): s_dep += 0.5
291         elif ac in ("anxiety", "гіперактивність"): s_anx += 0.4
292         elif ac == "normal": s_norm += 0.3
293
294     if calories_state:
295         cs = str(calories_state).lower()
296         if cs in ("depression", "значний дефіцит"): s_dep += 0.3
297         elif cs in ("anxiety", "значний надлишок"): s_anx += 0.3
298         elif cs == "normal": s_norm += 0.2
299
300     m = max(1, int(available_modules))
301     T = temperature * (5.0 / m); T = max(1.0, min(T, 3.0))
302     logits = np.array([s_dep, s_anx, s_norm]) / T + np.log(prior + 1e-9)
303     return _softmax(logits)

```

```

297     return _softmax(logits)
298
299 def smooth_probs(prev, curr, beta=0.8):
300     prev = np.asarray(prev, dtype=float); curr = np.asarray(curr, dtype=float)
301     log_prev = np.log(prev + 1e-9); log_curr = np.log(curr + 1e-9)
302     log_post = beta * log_prev + (1.0 - beta) * log_curr
303     post = np.exp(log_post); post /= post.sum()
304     return post
305
306 # ===== END INLINE DB + PROB HELPERS =====
307
308 # Ініціалізація БД
309 init_db()
310
311 #2 Завантаження натренованих моделей
312 model = joblib.load(os.path.join(BASE_DIR, "sleep", "sleep_model.pkl"))
313 hr_model = joblib.load(os.path.join(BASE_DIR, "heartrate", "heartrate_model.pkl"))
314 hr_encoder = joblib.load(os.path.join(BASE_DIR, "heartrate", "heartrate_label_encoder.pkl"))
315 activity_model = joblib.load(os.path.join(BASE_DIR, "activity", "activity_model.pkl"))
316 activity_encoder = joblib.load(os.path.join(BASE_DIR, "activity", "activity_label_encoder.pkl"))
317 calories_model = joblib.load(os.path.join(BASE_DIR, "calories", "calories_model.pkl"))
318 calories_encoder = joblib.load(os.path.join(BASE_DIR, "calories", "calories_label_encoder.pkl"))
319 voice_model = joblib.load(os.path.join(BASE_DIR, "voice", "voice_model.pkl"))
320 meta_model = joblib.load(os.path.join(BASE_DIR, "meta", "meta_model.pkl"))
321 meta_encoder = joblib.load(os.path.join(BASE_DIR, "meta", "meta_label_encoder.pkl"))
322
323 #3 Функція для вилучення ознак з аудіофайлу
324 def extract_features(file_path):
325     y, sr = librosa.load(file_path, sr=None)
326     mfccs = np.mean(librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13).T, axis=0)
327     rms = np.mean(librosa.feature.rms(y=y).T, axis=0)
328     zcr = np.mean(librosa.feature.zero_crossing_rate(y=y).T, axis=0)
329     features = np.hstack([mfccs, rms, zcr])
330     feature_names = [f"mfcc{i+1}" for i in range(13)] + ["rms", "zcr"]
331     return pd.DataFrame([features], columns=feature_names)
332
333 #4. Функція відкриття аудіофайлу
334 def open_file():
335     global audio_file_path
336     audio_file_path = filedialog.askopenfilename(
337         filetypes=[("Audio Files", "*.wav *.mp3")]
338     )
339     if audio_file_path:
340         filename = os.path.basename(audio_file_path)
341
342         filename = os.path.basename(audio_file_path)
343         audio_status_var.set(f"Вибрано файл: {filename}")
344         messagebox.showinfo("Файл завантажено", "Аудіо успішно завантажено.")
345     else:
346         audio_status_var.set("Файл не вибрано")
347
348 #5. Переглянути дані користувача
349 def show_user_data():
350     user_id = get_selected_user_id()
351     if not user_id:
352         messagebox.showwarning("Увага", "Оберіть користувача для перегляду.")
353         return
354
355     user_data = get_user_data(user_id)
356     if not user_data:
357         messagebox.showinfo("Інформація", "Немає збережених даних для цього користувача.")
358         return
359
360     view_window = tk.Toplevel(root)
361     view_window.title("Інформація про користувача")
362     view_window.geometry("700x500")
363
364     tk.Button(view_window, text="📄 Показати середні значення", font=("Arial", 12),
365              command=lambda: show_user_stats(user_id)).pack(pady=10)
366     tk.Button(view_window, text="🗑️ Видалити останній аналіз", font=("Arial", 12),
367              command=lambda: delete_last_observation(user_id, view_window)).pack(pady=5)
368
369     style = ttk.Style()
370     style.theme_use("clam")
371     style.configure("Treeview", rowheight=25, font=("Arial", 11), borderwidth=1, relief="solid")
372     style.configure("Treeview.Heading", font=("Arial", 12, "bold"))
373     style.map("Treeview", background=[("selected", "#c0c0c0")])
374
375     tree = ttk.Treeview(view_window, columns=("Поле", "Значення"), show="headings", style="Treeview")
376     tree.heading("Поле", text="Поле")
377     tree.heading("Значення", text="Значення")
378     tree.column("Поле", anchor="w", width=250)
379     tree.column("Значення", anchor="w", width=400)
380     tree.tag_configure("bold_row", font=("Arial", 14, "bold"))
381
382     scrollbar = ttk.Scrollbar(view_window, orient="vertical", command=tree.yview)
383     tree.configure(yscroll=scrollbar.set)

```

```

381
382 tree.pack(side="left", fill="both", expand=True)
383 scrollbar.pack(side="right", fill="y")
384
385 for key, value in user_data.items():
386     tree.insert("", "end", values=(key, value))
387
388 observations = get_user_observations(user_id)
389 if observations:
390     tree.insert("", "end", values=("", ""))
391     tree.insert("", "end", values=("--- Дані аналізів ---", ""))
392     for i, obs in enumerate(observations, 1):
393         tree.insert("", "end", values=(f"Аналіз #{i}", ""), tags=("bold_row",))
394         labels = [
395             "Тривалість сну", "REM-фаза", "Пробудження", "Час засинання", "Глибина сну",
396             "Серцебиття", "Активність", "Калорії", "Голос", "Сон",
397             "Стан серцебиття", "Стан активності", "Стан калорій", "Фінальний стан"
398         ]
399         for label, value in zip(labels, obs):
400             tree.insert("", "end", values=(label, value))
401
402 tree.pack(fill="both", expand=True)
403
404 def delete_last_observation(user_id, parent_window):
405     with _conn() as conn:
406         c = conn.cursor()
407         c.execute("""
408             DELETE FROM observations
409             WHERE id = (SELECT id FROM observations WHERE user_id = ? ORDER BY id DESC LIMIT 1)
410             """, (int(user_id),))
411         conn.commit()
412     messagebox.showinfo("Успіх", "Останній аналіз було видалено.")
413     parent_window.destroy()
414     show_user_data()
415
416 def show_user_stats(user_id):
417     data = get_user_observations(user_id)
418     if not data:
419         messagebox.showinfo("Немає даних", "Немає жодних аналізів для цього користувача.")
420         return
421
422 stats_window = tk.Toplevel(root)
423 stats_window.title("Середні значення параметрів користувача ")
424
425 stats_window.title("Середні значення параметрів користувача ")
426 stats_window.geometry("900x600")
427
428 numeric_data = np.array(data)[:, :8].astype(float)
429 averages = np.mean(numeric_data, axis=0)
430
431 def scale(values):
432     return [v / 1000 if v > 1000 else v for v in values]
433
434 averages_scaled = scale(averages)
435
436 labels = [
437     "Сон (год)", "REM (%)", "Пробудження", "Засинання (хв)", "Глибина сну (%)",
438     "Серцебиття", "Активність", "Калорії"
439 ]
440
441 fig = Figure(figsize=(8, 5))
442 ax = fig.add_subplot(111)
443 ax.bar(labels, averages_scaled, color='mediumseagreen')
444 for i, value in enumerate(averages_scaled):
445     ax.annotate(f"{averages[i]:.0f}",
446               xy=(i, value), xytext=(0, 5), textcoords="offset points",
447               ha='center', fontsize=10, color="black")
448 ax.set_title("Середні значення параметрів (значення >1000 поділено на 1000)")
449 ax.set_ylabel("Значення (оригінальні)")
450 ax.set_yticks([])
451 ax.tick_params(axis='x', rotation=30)
452 fig.tight_layout()
453
454 canvas = FigureCanvasTkAgg(fig, master=stats_window)
455 canvas.draw()
456 canvas.get_tk_widget().pack(fill="both", expand=True)
457
458 def format_pct_factory(proba):
459     def format_pct(pct):
460         if max(proba) >= 99.9:
461             return f"{pct:.1f}%" if abs(pct - 100) < 0.1 else ""
462         return f"{pct:.1f}%" if pct >= 5 else ""
463     return format_pct
464
465 #5. Функція для аналізу стану (основна логіка)
466 def analyze():
467     global audio_file_path

```

```

465 global audio_file_path
466 if audio_file_path is None:
467     messagebox.showerror("Помилка", "Будь ласка, завантажте аудіофайл.")
468     return
469
470 for entry in entries:
471     if not entry.get().strip():
472         messagebox.showerror("Помилка", "Будь ласка, заповніть усі поля перед запуском аналізу.")
473         return
474
475 try:
476     # 1) Голос
477     features_df = extract_features(audio_file_path)
478     voice_pred = voice_model.predict(features_df)[0]
479     voice_state = {0: "normal", 1: "anxiety", 2: "depression"}.get(int(voice_pred), "normal") \
480                 if isinstance(voice_pred, (int, np.integer)) else str(voice_pred).lower()
481
482     # 2) Ввідні
483     sleep_duration = float(entries[0].get())
484     rem_phase = float(entries[1].get())
485     awakenings = int(entries[2].get())
486     latency = float(entries[3].get())
487     slow_wave = float(entries[4].get())
488     heartrate = float(entries[5].get())
489     activity = int(entries[6].get())
490     calories = int(entries[7].get())
491
492     # 3) Користувач / стать
493     user_id = get_selected_user_id()
494     user = next((u for u in get_users() if u[0] == user_id), None)
495     if not user:
496         messagebox.showerror("Помилка", "Користувача не знайдено.")
497         return
498     gender = (user[5] or "").lower()
499     gender_female = 1 if gender == "жінка" else 0
500     gender_male = 1 if gender == "чоловік" else 0
501
502     # 4) Підмоделі
503     sleep_input = pd.DataFrame({
504         "duration": sleep_duration, "rem": rem_phase, "awakenings": awakenings,
505         "latency": latency, "slow_wave": slow_wave,
506         "gender_female": gender_female, "gender_male": gender_male
507     })
508
509     sleep_disorder = int(model.predict(sleep_input)[0])
510
511     activity_input = pd.DataFrame({
512         "activity": activity, "gender_female": gender_female, "gender_male": gender_male
513     })
514     activity_pred = activity_model.predict(activity_input)[0]
515     activity_state = str(activity_encoder.inverse_transform([activity_pred])[0]).lower()
516
517     hr_input = pd.DataFrame({
518         "heartrate": heartrate, "gender_female": gender_female, "gender_male": gender_male
519     })
520     hr_pred = hr_model.predict(hr_input)[0]
521     hr_state = str(hr_encoder.inverse_transform([hr_pred])[0]).lower()
522
523     calories_input = pd.DataFrame({
524         "calories": calories, "gender_female": gender_female, "gender_male": gender_male
525     })
526     calories_pred = calories_model.predict(calories_input)[0]
527     calories_state = str(calories_encoder.inverse_transform([calories_pred])[0]).lower()
528
529     # 5) Ймовірності 15/25/60
530     _modules = [voice_state, sleep_disorder, hr_state, activity_state, calories_state]
531     available_modules = sum(int(x is not None and x != "") for x in _modules)
532
533     proba = compute_tri_probs(
534         prior=(0.15, 0.25, 0.60),
535         voice_state=voice_state, sleep_disorder=sleep_disorder, hr_state=hr_state,
536         activity_state=activity_state, calories_state=calories_state,
537         available_modules=available_modules, temperature=1.0
538     )
539     labels = np.array(["Депресія", "Тривожний стан", "Здорова норма"])
540
541     # 6) Згладжування + логування
542     prev = get_last_probs(user_id)
543     if prev is not None:
544         proba = smooth_probs(prev, proba, beta=0.8)
545     save_assessment(user_id, proba[0], proba[1], proba[2], model_version="step2")
546
547     # 7) Фінальне формулювання
548     idx_max = int(np.argmax(proba))
549     if labels[idx_max] == "Здорова норма":
550         final_state = "Ймовірно здорова особа"

```

```

549         final_state = "Ймовірно здорова особа"
550     elif labels[idx_max] == "Тривожний стан":
551         final_state = "Невизначено – потрібен моніторинг"
552     else:
553         final_state = "Можливе фізіологічне або раннє психічне порушення"
554
555     # 8) Збереження спостереження
556     save_observation(
557         user_id, sleep_duration, rem_phase, awakenings, latency, slow_wave,
558         heartrate, activity, calories,
559         str(voice_state), int(sleep_disorder),
560         str(hr_state), str(activity_state), str(calories_state), str(final_state)
561     )
562
563     for entry in entries:
564         entry.delete(0, tk.END)
565
566     # 9) Текст
567     details_text = (
568         f"Фінальний психічний висновок: {final_state}\n\n"
569         f"Деталі: голос – {voice_state}, сон – {sleep_disorder}, "
570         f"серцебиття – {hr_state}, активність – {activity_state}, "
571         f"калорії – {calories_state}\n\n"
572     )
573     if final_state == "Високий ризик психічного розладу":
574         details_text += ("⚠️ Високий ризик. Зверніться до фахівця.\n")
575     elif final_state == "Ймовірно здорова особа":
576         details_text += ("✅ Поточний стан стабільний.")
577
578     # 10) Вікно результатів
579     result_window = tk.Toplevel(root)
580     result_window.title("Результат аналізу")
581     result_window.geometry("1200x900")
582
583     warnings_list = check_critical_parameters(heartrate, sleep_duration, rem_phase, slow_wave)
584
585     text_box = tk.Text(result_window, wrap="word", font=("Arial", 14), width=40)
586     text_box.tag_config("warning", foreground="red", font=("Arial", 12, "italic"))
587     text_box.insert("1.0", details_text)
588     if warnings_list:
589         text_box.insert("end", "\n\n⚠️ Критичні зауваження:\n", "warning")
590         for w in warnings_list:
591             text_box.insert("end", f"{w}\n", "warning")
592
593     text_box.insert("end", f"{w}\n", "warning")
594     text_box.config(state="disabled")
595     text_box.pack(side="left", fill="y", expand=False, padx=10, pady=10)
596
597     fig = Figure(figsize=(9, 10))
598     gs = gridspec.GridSpec(3, 1, height_ratios=[2, 1, 1])
599
600     color_mapping = {
601         "Ймовірно здорова особа": "#2ecc71",
602         "Високий ризик психічного розладу": "#e74c3c",
603         "Можливе фізіологічне або раннє психічне порушення": "#9b59b6",
604         "Невизначено – потрібен моніторинг": "#95a5a6",
605         "Соматична активність без вираженого психологічного стресу": "#3498db",
606         "Тимчасовий емоційний стрес": "#f39c12",
607         "Депресія": "#9b59b6",
608         "Тривожний стан": "#f39c12",
609         "Здорова норма": "#2ecc71",
610     }
611
612     # 🍷 Пай-чарт
613     pie_ax = fig.add_subplot(gs[0])
614     format_pct = format_pct_factory(proba)
615     colors = [color_mapping.get(label, "#cccccc") for label in labels]
616     wedges, _, autotexts = pie_ax.pie(proba, autopct=format_pct, startangle=140, colors=colors)
617     pie_ax.set_title("Ймовірності психічного стану")
618     pie_ax.axis("equal")
619     pie_ax.legend(wedges, labels, title="Стан", loc="center left", bbox_to_anchor=(1.05, 0.5), fontsize=10)
620
621     # 📈 Лінійний графік: поточні vs середні
622     parameter_labels = ["Сон (год)", "REM (%)", "Пробудження", "Засинання (хв)", "Глибина сну (%)",
623         "Серцебиття", "Активність", "Калорії"]
624     current_values = [sleep_duration, rem_phase, awakenings, latency, slow_wave, heartrate, activity, calories]
625
626     rows = get_user_observations(user_id)
627     if rows:
628         arr = np.array([r[0], r[1], r[2], r[3], r[4], r[5], r[6], r[7]] for r in rows, dtype=float)
629         average_values = np.nanmean(arr, axis=0).tolist()
630     else:
631         average_values = current_values[:]
632
633     def scale_values(values):
634         return [v / 1000 if v > 1000 else v for v in values]

```

```

633
634     current_values_scaled = scale_values(current_values)
635     average_values_scaled = scale_values(average_values)
636
637     line_ax = fig.add_subplot(gs[1])
638     x = np.arange(len(parameter_labels))
639     line_ax.plot(x, current_values_scaled, marker="o", label="Поточні")
640     line_ax.plot(x, average_values_scaled, marker="s", linestyle="--", label="Середні")
641     line_ax.set_xticks(x)
642     line_ax.set_xticklabels(parameter_labels, rotation=30, ha='right')
643     line_ax.set_ylabel("Значення")
644     line_ax.set_title("Поточні та середні значення (значення >1000 поділено на 1000)")
645     line_ax.legend()
646     line_ax.grid(True)
647
648     all_scaled = np.concatenate((current_values_scaled, average_values_scaled))
649     min_val, max_val = np.min(all_scaled), np.max(all_scaled)
650     padding = (max_val - min_val) * 0.1 if max_val > min_val else 10
651     line_ax.set_ylim([max(0, min_val - padding), max_val + padding])
652
653     # 📊 Барчарт ознак голосу
654     bar_ax = fig.add_subplot(gs[2])
655     bar_ax.bar(features_df.columns, features_df.values[0], color='skyblue')
656     bar_ax.set_title("Ознаки голосу (MFCC, RMS, ZCR)")
657     bar_ax.tick_params(axis='x', labelrotation=45)
658     bar_ax.set_ylabel("Середнє значення")
659     bar_ax.set_ylim(0, max(features_df.values[0]) * 1.1)
660
661     fig.tight_layout()
662     canvas = FigureCanvasTkAgg(fig, master=result_window)
663     canvas.draw()
664     canvas.get_tk_widget().pack(side="right", fill="both", expand=True)
665
666 except Exception as e:
667     messagebox.showerror("Помилка аналізу", str(e))
668
669 def check_critical_parameters(heartrate, sleep_duration, rem_phase, slow_wave):
670     warnings_list = []
671     if heartrate < 50 or heartrate > 100:
672         warnings_list.append("⚠ Серцебиття поза нормою. Зверніться до лікаря або телефонуйте 103.")
673     if sleep_duration < 5 or sleep_duration > 10:
674         warnings_list.append("⚠ Незвична тривалість сну. Варто звернутися до лікаря.")
675     if rem_phase < 15 or rem_phase > 30:
676         warnings_list.append("⚠ Порушення REM-фази сну. Варто звернутися до лікаря.")
677     if slow_wave < 10:
678         warnings_list.append("⚠ Знижена глибина сну. Варто звернутися до лікаря.")
679     return warnings_list
680
681 #6. Форма для додавання користувача (оновлена – статус і професія)
682 def open_user_form():
683     add_window = tk.Toplevel(root)
684     add_window.title("Додати користувача")
685
686     tk.Label(add_window, text="ПІБ:").grid(row=0, column=0, padx=5, pady=5, sticky="w")
687     name_entry = tk.Entry(add_window); name_entry.grid(row=0, column=1, padx=5)
688
689     tk.Label(add_window, text="Вік:").grid(row=1, column=0, padx=5, pady=5, sticky="w")
690     age_entry = tk.Entry(add_window); age_entry.grid(row=1, column=1, padx=5)
691
692     tk.Label(add_window, text="Ріст (см):").grid(row=2, column=0, padx=5, pady=5, sticky="w")
693     height_entry = tk.Entry(add_window); height_entry.grid(row=2, column=1, padx=5)
694
695     tk.Label(add_window, text="Вага (кг):").grid(row=3, column=0, padx=5, pady=5, sticky="w")
696     weight_entry = tk.Entry(add_window); weight_entry.grid(row=3, column=1, padx=5)
697
698     tk.Label(add_window, text="Стать:").grid(row=4, column=0, padx=5, pady=5, sticky="w")
699     gender_var = tk.StringVar(value="Жінка")
700     tk.Radiobutton(add_window, text="Жінка", variable=gender_var, value="Жінка").grid(row=4, column=1, sticky="w")
701     tk.Radiobutton(add_window, text="Чоловік", variable=gender_var, value="Чоловік").grid(row=4, column=2, sticky="w")
702
703     # Нове: статус служби
704     tk.Label(add_window, text="Статус:").grid(row=5, column=0, padx=5, pady=5, sticky="w")
705     status_var = tk.StringVar(value="military")
706     tk.Radiobutton(add_window, text="Військовий", variable=status_var, value="military").grid(row=5, column=1, sticky="w")
707     tk.Radiobutton(add_window, text="Цивільний", variable=status_var, value="civilian").grid(row=5, column=2, sticky="w")
708
709     # Нове: професія/роль
710     tk.Label(add_window, text="Професія/Роль:").grid(row=6, column=0, padx=5, pady=5, sticky="w")
711     occs = get_occupations()
712     occ_names = [name for _, name in occs] or [""]
713     occ_codes = [code for code, _ in occs] or [None]
714     occ_var = tk.StringVar(value=occ_names[0])
715     occ_menu = tk.OptionMenu(add_window, occ_var, *occ_names)
716     occ_menu.grid(row=6, column=1, padx=5, pady=5, sticky="w")
717

```

```

717
718 def save_user():
719     try:
720         name = name_entry.get().strip()
721         age = int(age_entry.get()) if age_entry.get().strip() else None
722         height = float(height_entry.get()) if height_entry.get().strip() else None
723         weight = float(weight_entry.get()) if weight_entry.get().strip() else None
724         gender = gender_var.get()
725         service_status = status_var.get()
726         # мапимо вибрану назву → код
727         occ_name = occ_var.get()
728         occ_code = None
729         for code, name_ in occs:
730             if name_ == occ_name:
731                 occ_code = code
732                 break
733
734         add_user(name, age, height, weight, gender, service_status, occ_code)
735         messagebox.showinfo("Успішно", "Користувача додано!")
736         add_window.destroy()
737         update_user_menu()
738     except ValueError:
739         messagebox.showerror("Помилка", "Будь ласка, введіть коректні дані.")
740
741 tk.Button(add_window, text="Додати", command=save_user).grid(row=7, column=0, pady=8)
742 tk.Button(add_window, text="Повернутись", command=add_window.destroy).grid(row=7, column=1, pady=8)
743
744 #7. Функція для отримання ID вибраного користувача
745 def get_selected_user_id():
746     selection = selected_user_var.get()
747     if not selection or not selection[0].isdigit():
748         return None
749     return int(selection.split(".")[0])
750
751 #8. Функція для оновлення випадяючого меню користувачів
752 def update_user_menu():
753     global user_menu
754     menu = user_menu["menu"]
755     menu.delete(0, "end")
756     new_users = get_users()
757     new_names = [{"u[0]": u[1]} for u in new_users]
758     for name in new_names:
759         menu.add_command(label=name, command=lambda value=name: selected_user_var.set(value))
760
761     menu.add_command(label=name, command=lambda value=name: selected_user_var.set(value))
762     if new_names:
763         selected_user_var.set(new_names[-1])
764
765 def is_first_observation(user_id):
766     observations = get_user_observations(user_id)
767     return len(observations) == 0
768
769 def determine_final_state(final_state, user_id):
770     if final_state == "Невизначено – потрібен моніторинг" and not is_first_observation(user_id):
771         return "Можливе фізіологічне або раннє психічне порушення"
772     return final_state
773
774 # 9. GUI – Ініціалізація головного вікна
775 root = tk.Tk()
776 root.title("Аналіз параметрів сну за моделлю")
777 root.geometry("800x760")
778 root.resizable(False, False)
779 root.configure(bg="#f4f6f8")
780
781 style = ttk.Style()
782 style.theme_use("clam")
783
784 style.configure("Main.TFrame", background="#f4f6f8")
785 style.configure("Card.TLabelframe", background="white", padding=12)
786 style.configure("Card.TLabelframe.Label", font=("Arial", 12, "bold"), foreground="#1f2d3d")
787 style.configure("Title.TLabel", background="#f4f6f8", font=("Arial", 18, "bold"), foreground="#1f2d3d")
788 style.configure("Section.TLabel", background="white", font=("Arial", 11), foreground="#2c3e50")
789 style.configure("Info.TLabel", background="white", font=("Arial", 10), foreground="#5b6570")
790 style.configure("TButton", font=("Arial", 11), padding=6)
791 style.configure("Accent.TButton", font=("Arial", 12, "bold"), padding=8)
792
793 def delete_selected_user():
794     user_id = get_selected_user_id()
795     if user_id is None:
796         messagebox.showerror("Помилка", "Користувача не вибрано.")
797         return
798     confirm = messagebox.askyesno("Підтвердження", "Видалити цього користувача?")
799     if confirm:
800         delete_user(user_id)
801         update_user_menu()
802         messagebox.showinfo("Готово", "Користувача видалено.")
803

```

```

801
802 main_frame = ttk.Frame(root, style="Main.TFrame", padding=18)
803 main_frame.pack(fill="both", expand=True)
804
805 title_label = ttk.Label(
806     main_frame,
807     text="Модуль моніторингу ризику депресії та тривожності",
808     style="Title.TLabel"
809 )
810 title_label.pack(pady=(0, 16))
811
812 # ===== Блок користувача =====
813 user_frame = ttk.LabelFrame(main_frame, text="Користувач", style="Card.TLabelframe")
814 user_frame.pack(fill="x", pady=(0, 14))
815
816 ttk.Label(user_frame, text="Оберіть користувача для аналізу:", style="Section.TLabel").grid(
817     row=0, column=0, sticky="w", padx=4, pady=(0, 8)
818 )
819
820 selected_user_var = tk.StringVar()
821 users = get_users()
822 user_names = [f"{user[0]}. {user[1]}" for user in users]
823 selected_user_var.set(user_names[0] if user_names else "Немає користувачів")
824
825 user_menu = tk.OptionMenu(user_frame, selected_user_var, *(user_names or ["Немає користувачів"]))
826 user_menu.config(font=("Arial", 11), width=22)
827 user_menu.grid(row=1, column=0, sticky="w", padx=4, pady=(0, 8))
828
829 user_btns = ttk.Frame(user_frame, style="Main.TFrame")
830 user_btns.grid(row=2, column=0, sticky="w", padx=0, pady=(4, 0))
831
832 ttk.Button(user_btns, text="Додати особу", command=open_user_form).pack(side="left", padx=(0, 8))
833 ttk.Button(user_btns, text="Переглянути дані", command=show_user_data).pack(side="left", padx=(0, 8))
834 ttk.Button(user_btns, text="Видалити користувача", command=delete_selected_user).pack(side="left")
835
836 # ===== Блок параметрів =====
837 params_frame = ttk.LabelFrame(main_frame, text="Параметри аналізу", style="Card.TLabelframe")
838 params_frame.pack(fill="x", pady=(0, 14))
839
840 fields = [
841     "Тривалість сну, год",
842     "REM-фаза, %",
843     "Пробудження за ніч",
844     "Пробудження за ніч",
845     "Час засинання, хв",
846     "Глибокий сон, %",
847     "Пульс у спокої, уд/хв",
848     "Активність, кроки",
849     "Калорії, ккал"
850 ]
851
852 entries = []
853
854 for i, field in enumerate(fields):
855     ttk.Label(params_frame, text=field, style="Section.TLabel").grid(
856         row=i, column=0, sticky="w", padx=(4, 16), pady=6
857     )
858     entry = ttk.Entry(params_frame, width=20)
859     entry.grid(row=i, column=1, sticky="w", pady=6)
860     entries.append(entry)
861
862 # ===== Блок аудіо та запуску =====
863 action_frame = ttk.LabelFrame(main_frame, text="Аудіофайл і запуск аналізу", style="Card.TLabelframe")
864 action_frame.pack(fill="x", pady=(0, 8))
865
866 audio_status_var = tk.StringVar(value="файл не вибрано")
867
868 ttk.Button(action_frame, text="Завантажити аудіо", command=open_file).grid(
869     row=0, column=0, sticky="w", padx=4, pady=(4, 8)
870 )
871
872 ttk.Label(action_frame, textvariable=audio_status_var, style="Info.TLabel").grid(
873     row=1, column=0, sticky="w", padx=4, pady=(0, 10)
874 )
875
876 ttk.Button(action_frame, text="Запустити аналіз", command=analyze, style="Accent.TButton").grid(
877     row=2, column=0, sticky="w", padx=4, pady=(0, 4)
878 )
879
880 root.mainloop()

```

Додаток Г

Апробація отриманих результатів

ЗБІРНИК НАУКОВИХ ПРАЦЬ

З МАТЕРІАЛАМИ VII МІЖНАРОДНОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ

12 ЧЕРВНЯ 2026 РІК

М. ЧЕРНІГІВ, УКРАЇНА

**«ТЕОРІЯ МОДЕРНІЗАЦІЇ В КОНТЕКСТІ
СУЧАСНОЇ СВІТОВОЇ НАУКИ»**



ЗБІРНИК НАУКОВИХ
ПРАЦЬ З МАТЕРІАЛАМИ
VII МІЖНАРОДНОЇ
НАУКОВОЇ КОНФЕРЕНЦІЇ



ТЕОРІЯ МОДЕРНІЗАЦІЇ В КОНТЕКСТІ СУЧАСНОЇ СВІТОВОЇ НАУКИ

| 12 червня 2026 рік
м. Чернігів, Україна

Вінниця, Україна
«UKRLOGOS Group»
2026

УДК 082:001
Т 33



Організація, від імені якої випущено видання:

ГО «Міжнародний центр наукових досліджень»

Номер запису організації в Єдиному реєстрі громадських об'єднань: 1499141.

Голова оргкомітету: Сотник С.Г.

Верстка: Білоус Т.В.

Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 22 від 11.06.2026 року.



Конференцію зареєстровано Державною науковою установою у сфері управління Міністерства освіти і науки «Український інститут науково-технічної експертизи та інформації» в базі даних науково-технічних заходів України на поточний рік та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення № 174 від 26.01.2026).

Збірник наукових праць з матеріалами конференції видано офіційно суб'єктом видавничої справи зі **Свідоцтвом ДК № 7860 від 22.06.2023**.

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Теорія модернізації в контексті сучасної світової науки:
Т 33 збірник наукових праць з матеріалами VII Міжнародної наукової конференції, м. Чернігів, 12 червня, 2026 р. / Міжнародний центр наукових досліджень. — Вінниця: ТОВ «УКРЛОГОС Груп», 2026. — 484 с.

ISBN 978-617-8582-52-4

DOI 10.62731/mcnd-12.06.2026

Викладено матеріали учасників VII Міжнародної наукової конференції «Теорія модернізації в контексті сучасної світової науки», яка відбулася 12 червня 2026 року у місті Чернігів.

УДК 082:001

© Колектив учасників конференції, 2026

© ГО «Міжнародний центр наукових досліджень», 2026

ISBN 978-617-8582-52-4

© ТОВ «УКРЛОГОС Груп», 2026

ПРОГРАМНА СИСТЕМА ПОПЕРЕДНЬОГО ОЦІНЮВАННЯ РИЗИКУ ПСИХОЕМОЦІЙНИХ ПОРУШЕНЬ НА ОСНОВІ ЦИФРОВИХ МАРКЕРІВ КОРИСТУВАЧА Горохівська Л.М.	323
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОШУКУ ЕТИЧНИХ БРЕНДІВ ОДЯГУ З ПОЯСНУВАННЯМ SUSTAINABILITY-СКОРИНГОМ Бабіч Б.В.	329
РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОЦІНЮВАННЯ БАЛАНСУ МІЖ НАВЧАННЯМ, РОБОТОЮ ТА ВІДПОЧИНКОМ У ДІЯЛЬНОСТІ JUNIOR-РОЗРОБНИКА Цирин О.М.	334
СУЧАСНІ ТЕНДЕНЦІЇ МОДЕРНІЗАЦІЇ ПРОГРАМНОЇ ІНЖЕНЕРІЇ: ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ В DEVOPS-ПРАКТИКИ Кязімов Рафаель Анар огли	339
ЦИФРОВІЗАЦІЯ ДЕРЖАВНИХ ПОСЛУГ У КОНТЕКСТІ ТЕОРІЇ МОДЕРНІЗАЦІЇ Титаренко В.А.	342

СЕКЦІЯ XV. ФІЛОЛОГІЯ ТА ЖУРНАЛІСТИКА

"LEGAL NIHILISM" IN THE LINGUISTIC WORLDVIEW: HOW CRIMINAL JARGON AND ARGO ENTER EVERYDAY SPEECH Ibrahimov I.M., Oliinyk V.O.	345
ГЕРМАНО-БАЛТО-СЛОВ'ЯНСЬКІ ІЗОГЛОСИ Хоменко Т.А.	348
МОДЕРНІЗАЦІЯ МЕДІАКОМУНІКАЦІЙ У КОНТЕКСТІ ЗАХИСТУ ПРАВ ЛЮДИНИ: ВИКЛИКИ ЦИФРОВОГО СУСПІЛЬСТВА Фурса І.З.	350
ХУДОЖНЄ ОСМИСЛЕННЯ АНТИКОЛОНІАЛІЗМУ У ПОЕМІ Т. ШЕВЧЕНКА «СОН» Хомякова В.С.	353

СЕКЦІЯ XVI. ПЕДАГОГІКА ТА ОСВІТА

АІ-ГРАМОТНІСТЬ МАЙБУТНЬОГО ВЧИТЕЛЯ ІНФОРМАТИКИ ЯК СКЛАДОВА ПРОФЕСІЙНОЇ ПІДГОТОВКИ В УМОВАХ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ОСВІТИ Ібрагімова Л.А.	357
ВИКОРИСТАННЯ СУЧАСНИХ ЦИФРОВИХ РЕСУРСІВ У ЛОГОПЕДИЧНІЙ ПРАКТИЦІ З ДІТЬМИ ДОШКІЛЬНОГО ВІКУ: ТЕОРЕТИЧНІ ТА ПРАКТИЧНІ АСПЕКТИ Федірко К.Ю.	363

ПРОГРАМНА СИСТЕМА ПОПЕРЕДНЬОГО ОЦІНЮВАННЯ РИЗИКУ ПСИХОЕМОЦІЙНИХ ПОРУШЕНЬ НА ОСНОВІ ЦИФРОВИХ МАРКЕРІВ КОРИСТУВАЧА

Горохівська Людмила Михайлівна

Бакалавр спеціальності «Комп'ютерні науки»

Західноукраїнський національний університет, Україна

Науковий керівник: Осолінський Олександр Романович

канд.техн.наук

доцент кафедри інформаційно-обчислювальних систем і управління

Західноукраїнський національний університет, Україна

Вступ

Психічні розлади залишаються однією з найважливіших медико-соціальних проблем сучасності. За даними Всесвітньої організації охорони здоров'я, у 2021 році майже кожна сьома людина у світі жила з психічним розладом, а найбільш поширеними були тривожні та депресивні розлади [1]. Поряд із високою поширеністю проблемою залишається недостатній доступ до своєчасної допомоги, тому особливого значення набувають засоби раннього виявлення ризикових змін і дистанційного моніторингу стану людини.

Традиційне оцінювання психоемоційного стану ґрунтується переважно на клінічній бесіді, анкетуванні та спостереженні спеціаліста. Такі підходи залишаються необхідними, однак мають обмеження: вони не завжди дозволяють безперервно відстежувати зміни у повсякденному житті, значною мірою залежать від суб'єктивних відповідей пацієнта та часто фіксують стан лише в окремий момент часу. Саме тому в останні роки активно розвиваються цифрове фенотипування та цифрові біомаркери психічного здоров'я, які поєднують дані зі смартфонів, носимих пристроїв та інших цифрових джерел [2], [3].

Особливий інтерес для попереднього оцінювання ризику психоемоційних порушень становлять цифрові маркери, що відображають сон, фізичну активність, серцевий ритм, витрати енергії та мовні характеристики. Систематичні огляди показують, що дані з носимих пристроїв і методи машинного навчання є перспективними для

виявлення ознак депресії, тривоги та пов'язаних станів, хоча такі системи не повинні замінювати клінічне рішення [4], [5]. Окремий напрям становить аналіз голосу: акустичні та просодичні параметри мовлення дедалі частіше розглядаються як потенційні цифрові маркери емоційного напруження та депресивних проявів [6], [7].

Метою роботи є розроблення програмної системи попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача, яка поєднує оброблення фізіологічних, поведінкових і голосових даних, надає інтерпретовані результати у зручному вебінтерфейсі та підтримує повторний аналіз даних у динаміці.

Опис програмної системи

Розроблена система реалізована як веборієнтований інструмент для лікаря або іншого авторизованого користувача. Архітектурно вона складається з рівня керування пацієнтами, модуля завантаження та вибору активних файлів, модуля аналітичної обробки табличних фізіологічних даних, голосового модуля та рівня візуалізації результатів. Така побудова дає змогу не лише отримувати прогноз, а й організовано працювати з історією спостережень, окремими файлами та поясненням отриманого результату.

My Project

Home

Patients

Messages

Settings

Luna

Age 26

Sex Female

Weight 55 kg

Height 160 cm

Profession Sedentary

26

Daily data

Choose file

File file already exists

Upload

DATE	STEPS	CALORIES	BPM (min)	BPM (max)	BPM (avg)	SLEEP (min)	SLEEP (max)	AROUSAL (%)	BPM %
2025-01-01	8865	488.6	53	129	64.8	9.6	1	21.3%	
2025-01-02	9294	522.6	57	111	64.6	8.07	0	23.6%	
2025-01-03	9138	514.2	53	101	63.9	9.62	0	23.8%	
2025-01-04	9895	551.5	53	112	62.3	8.73	1	24.7%	
2025-01-05	9439	502.7	53	107	66.3	8.67	1	26.2%	
2025-01-06	10769	473.4	57	101	71.6	8.88	0	21.7%	
2025-01-07	8446	474.3	58	120	74.8	8.38	0	30%	
2025-01-08	9239	482.6	54	108	66.3	8.77	1	25.8%	
2025-01-09	8205	498	54	116	71.3	9.22	1	25.2%	

Patient files

FILE NAME	TYPE	SIZE	DATE	ACTIONS
khal_Patient_2025_Female_1714890208.xlsx	XLSX	27.6 KB	2025-01-09 16:17:16	Download Use for analysis Delete

Рис. 1. Інтерфейс розробленої системи для роботи з цифровими маркерами пацієнта

У системі передбачено створення профілю пацієнта, завантаження структурованих таблиць із щоденними вимірюваннями та аудіофайлів у форматі WAV, вибір активного файлу для поточного аналізу, перегляд вихідних значень, обчислених середніх показників, похідних індикаторів і часових тенденцій. Результати подаються у вигляді імовірностей класів, короткого текстового пояснення та таблиць змін у часі, що робить систему зручною для попереднього скринінгу та подальшого клінічного розгляду.

Цифрові маркери та модель оцінювання

Базу інформаційних ознак системи становлять цифрові маркери, які можуть бути отримані з носимих пристроїв і супровідних цифрових джерел. До основних фізіологічних і поведінкових показників належать кількість кроків, витрати калорій, середня, мінімальна та максимальна частота серцевих скорочень, загальна тривалість сну, кількість пробуджень і частка фази REM. Для підвищення інформативності система також використовує агреговані характеристики за інтервал спостереження, похідні співвідношення, часові тенденції та контекстні ознаки користувача.

З урахуванням практичних вимог до персоналізації в модель структурованих даних включено не лише фізіологічні показники, а й контекстні атрибути користувача: вік, стать, індекс маси тіла, тип професійного або навчального навантаження, а також похідні категоріальні ознаки, сформовані на їх основі. Такий підхід дає змогу зменшити ризик однакової інтерпретації для різних профілів користувачів із подібними сирими вимірами та краще враховувати індивідуальні особливості норми.

Для класифікації структурованих табличних даних застосовується модель градієнтного бустингу CatBoost у багатокласовому режимі. На вхід моделі подаються агреговані числові ознаки та категоріальні контекстні параметри, а на виході формується розподіл імовірностей між чотирма класами: «норма», «тривога», «депресія» та «ПТСР». Обрана модель придатна для роботи зі змішаними табличними ознаками, підтримує оброблення категоріальних змінних і забезпечує зручне повторне використання після офлайнного навчання.

Окремою аналітичною гілкою є голосовий модуль. Після завантаження WAV-файлу аудіосигнал перетворюється на вектор ознак, що включає MFCC-коефіцієнти, RMS-енергію та zero-crossing rate. Далі ці

характеристики подаються на вхід попередньо навченої моделі голосового аналізу, а результат також відображається у вигляді імовірностей класів і короткого пояснення. Мультимодальний підхід підвищує гнучкість системи, оскільки поєднує поведінкові, фізіологічні та мовні сигнали в єдиному інформаційному середовищі.

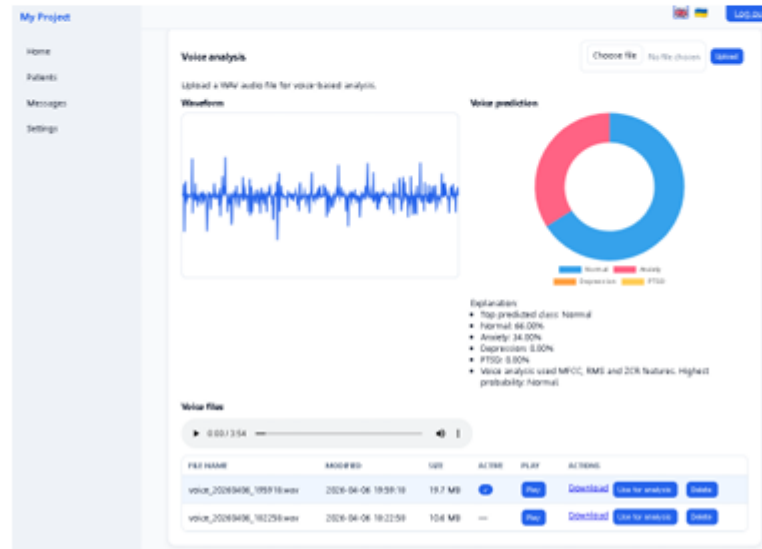


Рис. 2. Модуль голосового аналізу та подання результатів прогнозування

Практична реалізація та апробація

Програмну систему реалізовано як інтегрований вебзастосунок із фронтендом для взаємодії з користувачем, серверною логікою для керування даними пацієнтів і файловою підсистемою для роздільного зберігання таблиць та аудіозаписів. У межах сценарію використання лікар може переглянути пацієнта, вибрати активний файл для аналізу, отримати підсумкові середні значення, похідні індикатори, таблицю змін у часі та прогнозні імовірності. Така організація забезпечує відтворюваність аналітичного процесу й знижує ризик помилкового аналізу не того файлу.

На етапі розроблення та налагодження для структурованої моделі використовувався синтетичний набір даних, побудований за логікою

типових профілів спостереження, а для практичної перевірки працездатності інтерфейсу та конвеєра оброблення було залучено реальні дані носимого пристрою, зібрані в пілотному форматі від добровольців. Це дозволило перевірити коректність завантаження, обчислення агрегованих характеристик, формування прогнозу та відображення результатів у вебінтерфейсі.

Отримані результати підтвердили, що запропонована система придатна для попереднього аналізу цифрових маркерів користувача та може бути використана як допоміжний інструмент у сценаріях скринінгу, динамічного спостереження і підготовки до подальшого клінічного розгляду. Водночас її результати повинні трактуватися саме як підтримка прийняття рішення, а не як самостійний медичний висновок.

Висновки

У роботі запропоновано програмну систему попереднього оцінювання ризику психоемоційних порушень на основі цифрових маркерів користувача. Система поєднує збір і організацію даних пацієнта, оброблення фізіологічних і поведінкових показників, окремий модуль голосового аналізу та наочне подання результатів у вебінтерфейсі.

Практична цінність запропонованого рішення полягає в тому, що воно дає можливість перейти від розрізнених сирих показників до структурованого, повторюваного та частково персоналізованого попереднього оцінювання. Включення контекстних характеристик користувача, таких як вік і тип професійного навантаження, підвищує чутливість системи до індивідуальних відмінностей між профілями спостереження.

Подальший розвиток роботи доцільно спрямувати на розширення вибірки реальних даних, клінічну валідацію моделей, удосконалення мультимодального злиття ознак і поглиблення механізмів пояснюваності прогнозу. Це створює передумови для побудови гнучких цифрових інструментів раннього виявлення ризикових психоемоційних змін у повсякденному житті.

Список використаних джерел:

1. Mental disorders. World Health Organization. 2025. URL: <https://www.who.int/news-room/fact-sheets/detail/mental-disorders> (дата звернення: 26.05.2026).
2. Bufano P., Casiglia S., Di Girolamo G., Rossi A., Comparelli A., De Carolis A., et al. Digital Phenotyping for Monitoring Mental Disorders: Systematic Review. *Journal of Medical Internet Research*. 2023. Vol. 25. e46778.

3. McGinnis E. W., Cherian J., McGinnis R. S. The State of Digital Biomarkers in Mental Health. *Digital Biomarkers*. 2024. Vol. 8, No. 1. P. 210–217.
4. Abd-Alrazaq A., AlSaad R., Shuweihdi F., Ahmed A., Aziz S., Sheikh J. Systematic review and meta-analysis of performance of wearable artificial intelligence in detecting and predicting depression. *npj Digital Medicine*. 2023. Vol. 6. Art. 84.
5. Abd-alrazaq A., Alajlani M., Ahmad R., AlSaad R., Aziz S., Ahmed A., et al. Wearable Artificial Intelligence for Anxiety and Depression: Scoping Review. *Journal of Medical Internet Research*. 2023. Vol. 25. e42672.
6. Donaghy P., McNeill A., O'Brien S., et al. A Review of Studies Using Machine Learning to Detect Voice Biomarkers for Depression. *Journal of Technology in Behavioral Science*. 2024.
7. Riad R., Wac K., Salah A. A., et al. Automated Speech Analysis for Risk Detection of Depression, Anxiety, Insomnia, and Fatigue: Algorithm Development and Validation Study. *Journal of Medical Internet Research*. 2024. Vol. 26. e58572.