

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка програмного забезпечення для інтеграції даних ЕЕГ-
звітів вимірювань акселерометра у пацієнтів з тремором кисті руки

Виконав(ла): студент(ка) IV курсу, групи СП-43
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Шмирко Д.Т.

Керівник

Бревус В.М.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Стоянов Ю.М.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Петрик М.Р

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2025

АНОТАЦІЯ

Кваліфікаційна робота бакалавра. Тернопільський національний технічний університет імені Івана Пулюя, кафедра програмної інженерії, спеціальність 121 Інженерія програмного забезпечення ТНТУ, 2025, 59 сторінок, 13 рисунків, 4 таблиці, презентація, 25 джерел.

Ключові слова: біомедичні сигнали, ЕЕГ, акселерометрія, планшет, тремор, синхронізація даних, HL7, TimescaleDB, FastAPI, візуалізація, REST API, інтеграція даних.

Тема: Розробка програмного забезпечення для збору, інтеграції та аналізу біомедичних сигналів ЕЕГ та графічного планшета для дослідження тремору.

Об'єкт дослідження – процеси збору, синхронізації, обробки та візуалізації біомедичних сигналів у неврологічних тестах із використанням різних сенсорних пристроїв.

Предмет дослідження – методи та програмні технології інтеграції та аналізу даних ЕЕГ і акселерометричних (графічних планшетів) вимірювань для виявлення та кількісної оцінки тремору.

Мета роботи – створити мікросервісне програмне забезпечення, яке забезпечує збір, валідацію, синхронізацію, зберігання й візуалізацію інтегрованих біомедичних даних із ЕЕГ та планшета, із застосуванням сучасних стандартів медичного обміну та гнучкого веб-інтерфейсу для дослідників і лікарів.

Методи дослідження: системний аналіз предметної області, об'єктно-орієнтоване проектування, методи цифрової обробки сигналів, реляційне моделювання даних, REST API архітектура, інтеграційне тестування.

Наукова новизна роботи полягає у комплексному підході до інтеграції біомедичних сигналів, що передбачає синхронізацію даних із різних джерел, автоматичну обробку пропусків, застосування стандартів HL7 та IEEE 11073, а також впровадження ефективної архітектури для зберігання й візуалізації багатоканальних часових рядів.

Практичне значення – розроблене ПЗ може використовуватись у клінічних, дослідницьких та освітніх цілях для швидкої підготовки, інтеграції та аналізу даних біомедичних досліджень, а також для впровадження у центрах неврології чи фізіотерапії.

ABSTRACT

Bachelor's thesis. Ternopil Ivan Puluj National Technical University, Department of Software Engineering, specialty 121 "Software Engineering", TNTU, 2025, 59 pages, 13 figures, 4 tables, presentation, 25 references.

Keywords: biomedical signals, EEG, accelerometry, tablet, tremor, data synchronization, HL7, TimescaleDB, FastAPI, visualization, REST API, data integration.

Title: Development of software for the collection, integration, and analysis of EEG and tablet biomedical signals for tremor research.

Object of research: processes of collecting, synchronizing, processing, and visualizing biomedical signals in neurological tests using various sensor devices.

Subject of research: methods and software technologies for integrating and analyzing EEG and accelerometric (tablet) measurements for tremor detection and quantitative assessment.

Purpose of the thesis: to develop microservice-based software that provides collection, validation, synchronization, storage, and visualization of integrated biomedical data from EEG and tablet sources, applying modern standards of medical data exchange and a flexible web interface for researchers and clinicians.

Methods: system analysis of the subject area, object-oriented design, digital signal processing techniques, relational data modeling, REST API architecture, integration testing.

Scientific novelty: the research offers a comprehensive approach to integrating biomedical signals by synchronizing data from different sources, automated handling of missing values, applying HL7 and IEEE 11073 standards, and implementing an efficient architecture for storing and visualizing multichannel time series.

Practical significance: the developed software can be used in clinical, research, and educational contexts for fast preparation, integration, and analysis of biomedical research data, and is suitable for deployment in neurology or physiotherapy centers.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕГРАЦІЇ БІОМЕДИЧНИХ ДАНИХ 1.1	
Основи акселерометрії та аналіз рухових сигналів	10
1.1.1 Структура та характеристики даних акселерометрів	11
1.1.2 Основні методи аналізу акселерометричних даних	11
1.1.3 Методи візуалізації та інтерпретації багатомодальних даних.....	13
1.2 Огляд існуючих методів та програмних рішень інтеграції ЕЕГ і акселерометра.....	14
1.2.1 Основні підходи до інтеграції ЕЕГ і акселерометрії.....	14
1.2.2 Програмні рішення	16
РОЗДІЛ 2. АНАЛІЗ ВИМОГ І ПРОЄКТУВАННЯ СИСТЕМИ.....	18
2.1. Функціональні та нефункціональні вимоги, сценарії використання.....	18
2.1.1. Основні задачі та очікуваний функціонал системи	18
2.1.2. Функціональні вимоги.....	19
2.1.3. Нефункціональні вимоги	21
2.1.4. Основні сценарії використання	22
2.2 Оцінка стандартів обміну медичними даними (HL7, IEEE 11073).....	24
2.2.1 Стандарт HL7: принципи та застосування	24
2.2.2 Стандарт IEEE 11073: медичні пристрої та формати даних.....	25
2.3 Архітектура програмного забезпечення	26
2.3.1 Призначення і взаємодія основних модулів	26
2.4 Моделювання даних: структура ЕЕГ-звітів і вимірювань акселерометра	28
2.5 Інтерфейси для збору, обробки та візуалізації даних.....	31
2.5.1 Веб-інтерфейс для користувача	32
2.5.2 Модуль візуалізації даних та звітності	33
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО МОДУЛЯ	35

3.1 Вибір технологій (мова програмування, фреймворки, СУБД).....	35
3.2. Розробка основних модулів.....	38
3.2.1 Збір даних із пристроїв.....	38
3.2.2 Попередня обробка сигналів.....	40
3.2.3 Зберігання і управління базою даних	40
3.3 Інтеграційне тестування та верифікація коректності даних.....	42
3.4. Оцінка продуктивності та надійності системи	46
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	47
4.1 Долікарська допомога при кровотечах	47
4.2 Заходи, що покращують умови праці оператора ПК.....	49
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	57
ДОДАТОК А – Лістинг частини модулів.....	58
ДОДАТОК Б – Диск з кваліфікаційною роботою бакалавра	59

ВСТУП

В умовах розвитку цифрової медицини та зростання ролі біомедичних сенсорних технологій особливої актуальності набуває питання інтеграції різномірних даних, отриманих у процесі медичних досліджень та діагностики. Поєднання електроенцефалографічних записів (ЕЕГ) із руховими сигналами, що фіксуються акселерометрами або графічними планшетами, відкриває нові можливості для комплексного аналізу стану пацієнтів із руховими порушеннями, такими як тремор, хвороба Паркінсона, інші нейродегенеративні захворювання.

Мета даної дипломної роботи — розробити мікросервісне програмне забезпечення для збору, обробки та інтеграції даних з ЕЕГ-звітів та акселерометричних/графічних планшетів, орієнтоване на дослідження тремору кисті руки. Система має забезпечувати зручний інтерфейс для завантаження та валідації даних, підтримку сучасних стандартів зберігання й обміну медичною інформацією, а також можливість комплексної візуалізації та аналізу результатів.

У межах цієї роботи поставлені такі завдання: здійснити огляд теоретичних основ акселерометрії та методів аналізу рухових сигналів, проаналізувати сучасні методи і програмні рішення для інтеграції даних ЕЕГ та акселерометрів, сформулювати вимоги до програмного комплексу з урахуванням як функціональних, так і нефункціональних, а також інтероперабельних характеристик, обґрунтувати вибір стандартів представлення та обміну біомедичними даними, зокрема таких як HL7 та IEEE 11073, спроектувати архітектуру програмної системи, структуру сховища даних і інтерфейси користувача, реалізувати основні модулі, що забезпечують парсинг, валідацію, очищення, синхронізацію та зберігання даних, а також провести інтеграційне тестування та здійснити оцінку продуктивності й надійності запропонованого рішення.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕГРАЦІЇ БІОМЕДИЧНИХ ДАНИХ

1.1 Основи акселерометрії та аналіз рухових сигналів

Акселерометрія займає важливе місце в сучасній біомедичній інженерії та неврології, оскільки дає змогу об'єктивно реєструвати та аналізувати рухову активність людини у різних станах, зокрема при патологіях, що супроводжуються руховими розладами (тремор, паркінсонізм, порушення координації). Застосування акселерометрів у медичній практиці дає змогу автоматизувати процеси моніторингу та діагностики, підвищити точність і об'єктивність оцінки стану пацієнта, а також розширити можливості наукових досліджень у галузі аналізу рухів.

Акселерометр — це сенсорний пристрій, який вимірює прискорення об'єкта (тобто зміну швидкості) вздовж однієї або кількох осей. У біомедичній практиці найчастіше використовують три осьові акселерометри (X, Y, Z), які дозволяють отримати повну інформацію про рухові характеристики кінцівок, тулуба чи інших частин тіла.

Основними фізичними принципами роботи акселерометрів є:

- П'єзоелектричний ефект — виникнення електричного заряду на поверхні кристалу під впливом механічного навантаження.
- Ємнісний ефект — вимірювання зміни електричної ємності між електродами внаслідок переміщення внутрішньої маси сенсора.
- Резистивний ефект — використовується рідше, оснований на зміні електричного опору при деформації елемента сенсора.

Завдяки розвитку мікроелектромеханічних систем (MEMS) сучасні акселерометри мають компактні розміри, низьке енергоспоживання, високу точність і доступну вартість. Це дозволяє вбудовувати їх у різноманітні носимі пристрої — фітнес-браслети, смартфони, медичні трекери, спеціалізовані сенсорні платформи для наукових експериментів тощо.

1.1.1 Структура та характеристики даних акселерометрів

Дані акселерометрів представляють собою часові ряди — послідовності числових значень прискорення за певними осями, кожне з яких асоційоване з відповідною часовою міткою. Основними параметрами акселерометричних даних є:

- Частота дискретизації (sampling rate) — визначає, з якою періодичністю фіксується значення прискорення (часто від 50 Гц до 1000 Гц для медичних досліджень).
- Діапазон вимірювань — максимальні та мінімальні значення прискорення, які може фіксувати сенсор (зазвичай від $\pm 2g$ до $\pm 16g$).
- Точність вимірювання — залежить від роздільної здатності АЦП (аналогово-цифрового перетворювача), зазвичай від 8 до 16 біт.

Таблиця 1.1 Типовий вигляд структурованого файлу акселерометрії:

timestamp	X	Y	Z
00:00:00.0	0.015	-0.002	0.998
00:00:00.1	0.017	-0.003	1.002
...	...	...	...

Ще може бути стовпець Pressure — якщо це планшет або стилус.

1.1.2 Основні методи аналізу акселерометричних даних

Для ефективного аналізу рухових сигналів, отриманих з акселерометрів, у біомедичних дослідженнях застосовують низку алгоритмів і підходів:

- Попередня обробка:

- Фільтрація сигналів для видалення шуму та артефактів (часто використовують низькочастотні фільтри, фільтр Баттерворта, медіанні фільтри).
- Нормалізація даних для приведення значень до єдиного масштабу.
- Сегментація руху:
 - Виділення окремих епізодів (кроки, фази тремору) на основі порогового або кластерного аналізу.
 - Використання методів машинного навчання для автоматичної класифікації типів рухів.
- Статистичний аналіз:
 - Обчислення середніх, стандартних відхилень, пікових значень для кожної осі.
 - Оцінка симетрії руху (для правої та лівої руки).
- Частотний аналіз:
 - Швидке перетворення Фур'є (FFT) для визначення основних частот тремору чи інших ритмічних рухів.
 - Аналіз спектру для виявлення характерних патологічних патернів (тремор у діапазоні 4–6 Гц при хворобі Паркінсона).
- Візуалізація:
 - Побудова часових графіків та спектрограм рухових сигналів, що дозволяє візуально ідентифікувати епізоди тремору чи аномалії.

У пацієнтів із тремором (при хворобі Паркінсона або есенціальному треморі) акселерометр дозволяє кількісно оцінити такі характеристики тремору, як частота, амплітуда, тривалість епізодів, їх динаміку під впливом лікування чи в різних життєвих ситуаціях [1]. Сучасні дослідження показують, що об'єднання акселерометричних даних із даними ЕЕГ підвищує діагностичну цінність та допомагає розрізняти типи рухових розладів.

1.1.3 Методи візуалізації та інтерпретації багатомодальних даних

Візуалізація біомедичних сигналів — ключовий етап інтегрованого аналізу, що дозволяє ефективно ідентифікувати закономірності, аномалії, кореляції та синхронізовані події між різними типами даних (наприклад, ЕЕГ і акселерометрія). Сучасні інструменти візуалізації значно полегшують інтерпретацію великих масивів складної інформації, роблять її доступною не лише для дослідників, а й для клініцистів.

Види графіків для біомедичних даних:

- Часові графіки (Time-series plots):

Найбільш поширений тип, що дозволяє відобразити динаміку сигналу у часовому розрізі. Для аналізу інтегрованих даних часто використовують багатоканальні часові графіки — коли, наприклад, на одному малюнку відображається одразу ЕЕГ (кілька каналів) та акселерометр (X, Y, Z чи Pressure).

Приклад:

- Графік зміни амплітуди тремору (вісь Y акселерометра) та потужності певного діапазону ЕЕГ у реальному часі.
- Спектральні графіки (Spectrograms, Power Spectrum):
Дозволяють аналізувати частотні характеристики сигналів, що особливо важливо для виявлення патологічних ритмів 4–6 Гц.
 - Спектрограма: двовимірна карта, де по одній осі — час, по іншій — частота, а кольором показано потужність/амплітуду сигналу.
 - Гістограми частотної потужності: для порівняння спектральних характеристик різних сигналів (наприклад, ЕЕГ та руху) за однакові часові інтервали.
- 3D-візуалізації:
Дозволяють поєднувати кілька параметрів у одному просторі. Наприклад, для

акселерометрії — побудова траєкторії руху кінцівки у тривимірному просторі (X, Y, Z), або накладання траєкторії руху на карту активності мозку.

- Кореляційні діаграми та спільні графіки (Joint plots): Дозволяють відслідковувати взаємозв'язок між різними сигналами — наприклад, амплітудою руху й інтенсивністю певних ритмів ЕЕГ. Це може бути:
 - Двовимірний скаттер-плот (scatter plot), де по одній осі — параметр руху, по іншій — характеристика мозкової активності.
 - Лінійні або багатолінійні графіки для наочності співвідношень у часі.

1.2 Огляд існуючих методів та програмних рішень інтеграції ЕЕГ і акселерометра

Інтеграція даних електроенцефалографії (ЕЕГ) та акселерометрії — одне з ключових завдань у сучасних біомедичних дослідженнях для аналізу рухових розладів. Такий підхід дає змогу поєднати об'єктивні маркери руху з нейрофізіологічними показниками активності головного мозку. На практиці це дозволяє точніше діагностувати різні стани пацієнта, виявляти кореляції між руховою та мозковою активністю, а також розробляти інтелектуальні системи підтримки прийняття рішень у медицині.

1.2.1 Основні підходи до інтеграції ЕЕГ і акселерометрії

Синхронізоване записування

Більшість сучасних досліджень проводиться за схемою, коли дані ЕЕГ і акселерометра записуються одночасно. Це може бути досягнуто шляхом

використання багатоканальних медичних реєстраторів, до яких під'єднуються як ЕЕГ-електроди, так і акселерометри, або шляхом окремого запису із подальшою синхронізацією по часових мітках.

Апаратна синхронізація:

В обох пристроях (ЕЕГ і акселерометр) використовується загальний тактовий генератор або сигнал-маркер (наприклад, “trigger”), що гарантує ідентичність часових відліків.

- Переваги: Максимальна точність інтеграції.
- Недоліки: Дорожчі системи, складніше налаштування.

Програмна

синхронізація:

Якщо пристрої не мають апаратного зв'язку, час синхронізується програмно — по часових мітках у файлах. Використовується ресемплінг, інтерполяція, кореляційний аналіз для суміщення даних.

- Переваги: Гнучкість, простота реалізації.
- Недоліки: Може виникати невелике зміщення (латентність) через різну точність годинників.

Алгоритми інтеграції та аналізу:

Попередня обробка:

- Нормалізація часових осей (все зводиться до єдиного формату, наприклад, мілісекунди з початку експерименту).
- Фільтрація та видалення артефактів окремо для кожного сигналу.

Об'єднання даних:

- Inner join по часових мітках (з певною точністю, наприклад, ± 10 мс).
- Інтерполяція сигналів із меншою частотою дискретизації (наприклад, акселерометр зазвичай має вищу частоту, ніж ЕЕГ).

Комплексний аналіз:

- Кореляційний аналіз: визначення зв'язку між руховою та мозковою активністю.
- Виявлення епізодів руху/тремору на основі акселерометрії й вивчення відповідної динаміки ЕЕГ.

- Мультимодальний машинний аналіз (класифікація/кластеризація з використанням обох типів сигналів) [2].

1.2.2 Програмні рішення

З програмних рішень можна виділити комерційні та відкриті:

Відкриті інструменти:

- MNE-Python.
Відкритий фреймворк для аналізу нейрофізіологічних сигналів, особливо EEG і MEG. Підтримує імпорт акселерометричних даних та їх інтеграцію з EEG (через `mne.io.RawArray`, збереження часової) [3].
- BioSPPy (Biomedical Signal Processing in Python).
Дає змогу попередньо обробляти, фільтрувати, аналізувати біомедичні сигнали, у тому числі акселерометрію.
- EEGLAB (MATLAB).
Плагіни для імпорту акселерометричних даних і спільної обробки з EEG, аналізу артефактів руху.
- Lab Streaming Layer (LSL).
Система синхронізації потоків біомедичних даних у реальному часі, широко застосовується у сучасних дослідницьких лабораторіях.

Комерційні та спеціалізовані системи:

- Natus Neuroworks, Micromed BrainQuick, Compumedics Graef — професійні платформи, які можуть мати вбудовану підтримку акселерометрії поряд із EEG, проте інтеграція обмежена закритими форматами.
- Wearable-трекери (наприклад, Empatica E4, Shimmer) — деякі підтримують одночасний запис EEG і акселерометрії, відкритий експорт даних у CSV/EDF.

Інтеграція багатомодальних біомедичних сигналів, таких як ЕЕГ та акселерометрія, супроводжується низкою характерних технічних викликів, які впливають на якість обробки й інтерпретації даних.

Часто виникають проблеми з різною частотою дискретизації та форматом часових міток.

Наприклад, ЕЕГ-системи зазвичай мають частоту вибірки 256–1024 Гц, тоді як акселерометри або інші сенсори можуть працювати з іншими параметрами (50–100 Гц або навіть нижче/вище), і кожен пристрій зберігає час у власному форматі (секунди, мілісекунди, інкрементальний лічильник). Це ускладнює об'єднання даних “по часу” та потребує попереднього узгодження часової бази: ресемплінгу, інтерполяції, або приведення до єдиного формату.

Також різниця в точності апаратних годинників (clock drift) може призвести до “поповзання” часових рядів при довготривалих записах, навіть якщо експеримент стартував синхронно. Без корекції такі розбіжності у таймінгу здатні суттєво вплинути на результати аналізу й виявлення синхронних подій.

РОЗДІЛ 2. АНАЛІЗ ВИМОГ І ПРОЄКТУВАННЯ СИСТЕМИ

2.1. Функціональні та нефункціональні вимоги, сценарії використання

2.1.1. Основні задачі та очікуваний функціонал системи

Система збору, інтеграції та аналізу біомедичних даних повинна забезпечити повний цикл обробки мультимодальних сигналів — від прийому сирих даних до надання інтерпретованих результатів користувачу (лікаря, досліднику, інженеру). Враховуючи специфіку роботи з ЕЕГ та акселерометричними звітами планшета, основні функції системи визначаються так:

Завантаження та валідація сирих файлів. Користувач має змогу завантажити два незалежних звіти (ЕЕГ та планшет), отримати зворотний зв'язок про їхню структурну і часову коректність, а також за потреби — рекомендації щодо усунення помилок.

Зберігання та організація даних. Всі завантажені дані мають бути надійно збережені як у сирому, так і в обробленому вигляді, із прив'язкою до сеансу, часу, апаратури, параметрів експерименту.

Синхронізація даних по часу. Ключова задача — забезпечення коректного об'єднання даних із різних пристроїв, попри різну частоту дискретизації, формат часових міток і наявність пропусків.

Попередня обробка сигналів. Реалізація базових алгоритмів очищення, нормалізації, фільтрації, ресемплінгу для отримання коректної вибірки для подальшого аналізу.

Інтегрований аналіз і візуалізація. Система повинна надавати базову візуалізацію спіралі, а також можливості обчислення метрик (тривалість, точність, плавність рухів, спектральний аналіз тремору).

Зберігання результатів у БД та можливість подальшого доступу для перегляду/експорту.

Безпека, автентифікація й обмеження доступу (мінімум на рівні бази/інтерфейсу).

2.1.2. Функціональні вимоги

Перелік основних вимог наведено нижче:

- Завантаження файлів (ЕЕГ, планшет) у форматах .txt, підтримка UTF-8 та Windows-1251.
- Валідація структурних, часових та контентних параметрів файлів.
- Автоматична перевірка сумісності двох файлів (одна сесія, близькі часові інтервали).
- Можливість очищення даних, фільтрації артефактів, нормалізації значень.
- Автоматичне вирівнювання та інтеграція сигналів з урахуванням різних частот.
- Збереження як сирих, так і синхронізованих даних у реляційній БД (PostgreSQL, TimescaleDB) з можливістю швидкої вибірки.
- Генерація метаданих сесії: дата/час, тип пристрою, оператор, параметри обробки.
- Підтримка REST API для інтеграції з іншими сервісами або розширення функціоналу.
- Протоколювання ключових дій користувача й системних подій.
- Побудова інтерактивних графіків і діаграм (спіраль Архімеда).

Нижче наведений приклад рисунку спіралі.

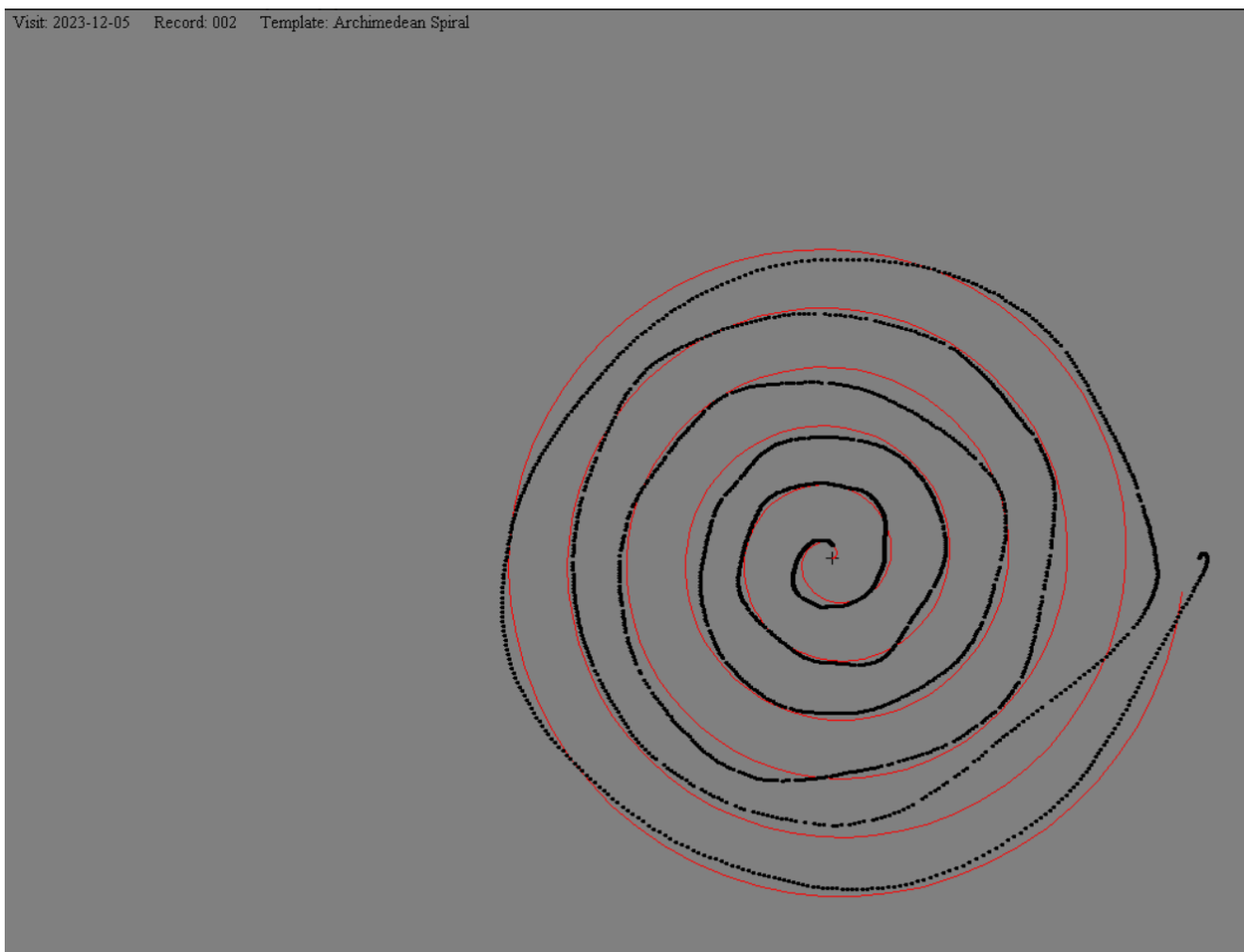


Рисунок 2.1.2.1. Приклад рисунку Архімедової спіралі у пацієнта з тремором

На рисунку 2.1.2.1 наведено типовий приклад виконання тесту малювання архімедової спіралі пацієнтом із тремором. Подібні тести широко використовуються в неврологічній практиці для оцінки ступеня моторних порушень, характерних для хвороби Паркінсона, есенціального тремору та інших патологічних станів центральної нервової системи.

Зображення містить дві основні компоненти.

Червона лінія — це математично ідеальна архімедова спіраль, яка використовується як еталон для порівняння. Вона генерується програмно, відповідає строго визначеним параметрам та задає форму, яку пацієнт має повторити.

Чорна пунктирна лінія — це траєкторія руху стилуса або пера на графічному планшеті, що реєструється під час тесту. Дана лінія відображає реальні рухи пацієнта, які часто є нерівномірними, містять мікроколивання, зсуви та відхилення, характерні для тремору.

Як видно на зображенні, траєкторія пацієнта значно відхиляється від ідеальної спіралі. Можна помітити нерівномірність лінії, наявність перехресть, ламаних відрізків та періодичних коливань навколо еталону. Такі відхилення слугують ключовим джерелом для кількісної оцінки тремору:

- визначаються амплітуда і частота коливань відносно ідеальної кривої;
- аналізується плавність і стійкість рухів у різних ділянках спіралі;
- розраховується час виконання завдання, а також стабільність сили натискання.

Отримані дані використовуються для подальшого чисельного аналізу, побудови спектрограм, кореляції з електроенцефалографічною активністю, а також моніторингу динаміки стану пацієнта протягом лікування.

Таким чином, тест малювання спіралі виступає простим, але інформативним інструментом для візуалізації моторних порушень та подальшого кількісного

2.1.3. Нефункціональні вимоги

Продуктивність: система має швидко обробляти навіть великі масиви даних (до кількох мільйонів точок), забезпечуючи час синхронізації не більше 30 секунд для одного сеансу на звичайному ПК.

Масштабованість: можливість розширення для паралельної обробки кількох сесій, збереження історії для багатьох користувачів.

Стійкість до помилок: система повинна коректно обробляти неповні або пошкоджені файли, сигналізувати про помилки та пропонувати варіанти виправлення.

Безпека даних: усі медичні дані повинні бути ізольовані, унеможливлювати несанкціонований доступ.

Можливість розгортання у контейнерах (Docker).

Використання відкритих стандартів для сумісності з іншими медичними системами (HL7, IEEE 11073) [4, 5].

Зручність інтерфейсу: інтерфейс повинен бути інтуїтивно зрозумілим навіть для не-ІТ користувача.

2.1.4. Основні сценарії використання

Нижче наведено основні сценарії, які реалізує система:

- Завантаження сесії: користувач завантажує файли ЕЕГ та планшета.
- Перевірка цілісності: система автоматично перевіряє структуру, формати, час.
- Синхронізація: система інтегрує дані в єдину часову сітку, формує спільний датасет.
- Зберігання: усі дані зберігаються в базі даних, доступні для перегляду/аналізу.
- Візуалізація: користувач обирає сесію, переглядає інтерактивні графіки спіралі та часових рядів.
- Експорт результатів: за потреби, можливий експорт даних у JSON форматі.
- Адміністрування: адміністратор може переглядати журнали дій, видаляти некоректні сесії.

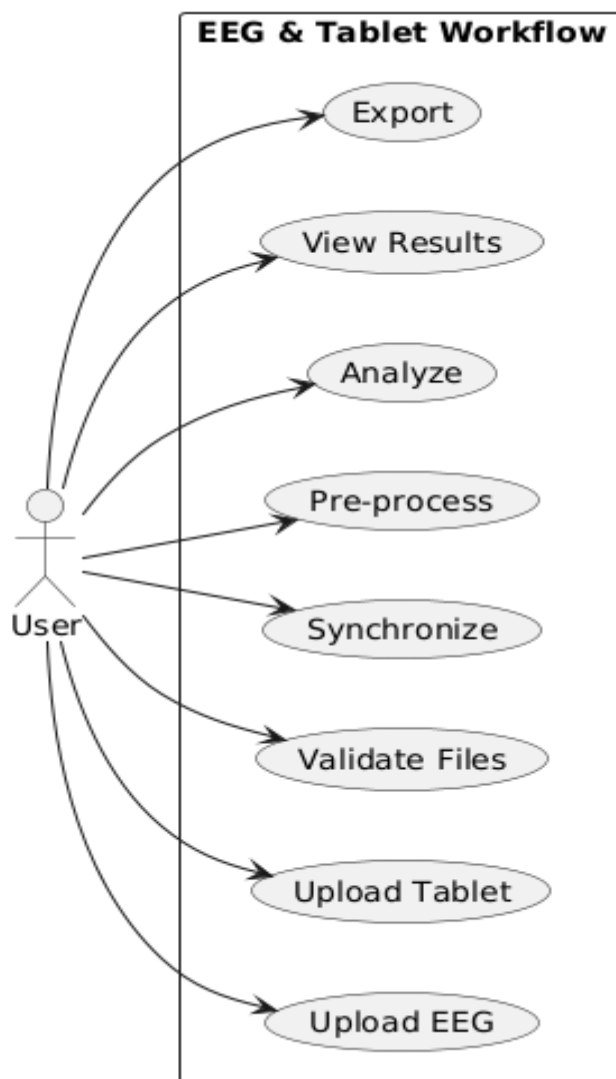


Рисунок 2.1.4.1. Діаграма використання актора User

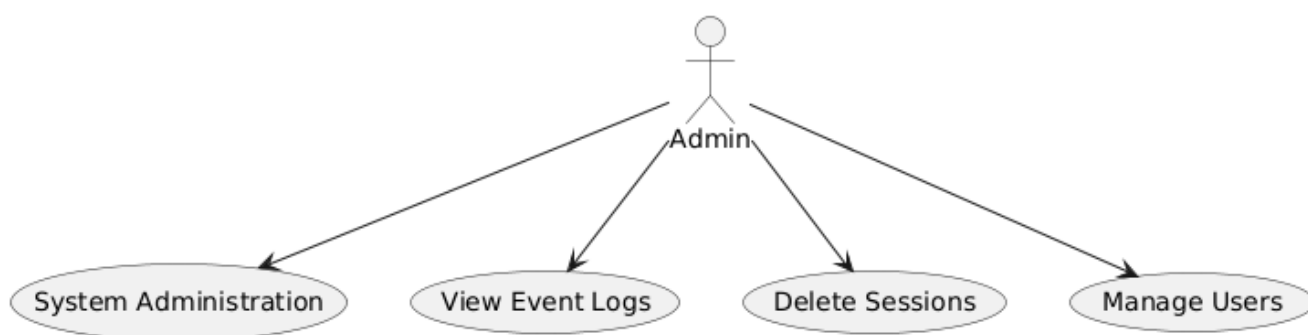


Рисунок 2.1.4.2. Діаграма використання актора Admin

2.2 Оцінка стандартів обміну медичними даними (HL7, IEEE 11073)

Інтеграція біомедичних даних із різних джерел — одне з ключових завдань сучасної медичної інформатики. Для забезпечення коректного обміну, сумісності та захисту медичної інформації в галузі застосовують міжнародні стандарти, серед яких найбільш поширеними є HL7 (Health Level 7) та IEEE 11073. Дотримання цих стандартів дозволяє створювати програмні комплекси, здатні інтегруватися із зовнішніми електронними медичними системами (наприклад, лікарняними інформаційними системами, лабораторними сервісами, пристроями збору даних тощо) [4, 5].

2.2.1 Стандарт HL7: принципи та застосування

HL7 (Health Level Seven International) — це сукупність міжнародних стандартів для обміну, інтеграції, зберігання та відтворення електронної медичної інформації. HL7 регламентує, як медичні дані повинні формуватися, описуватися і передаватися між інформаційними системами.

Основні особливості HL7:

- Визначає формат повідомлень для різних типів медичних подій (наприклад, госпіталізація, діагностика, передача результатів лабораторних досліджень тощо).
- Підтримує використання XML, JSON та інших сучасних форматів для опису структурованих даних.
- Дає змогу забезпечити інтероперабельність між різними програмними продуктами та апаратними пристроями.

- Забезпечує стандартизований підхід до обміну не лише медичними звітами, але й зображеннями, сигналами, даними з сенсорів тощо.

У контексті розробки програмного комплексу для аналізу ЕЕГ та акселерометрії використання стандарту HL7 актуальне для передачі структурованих результатів обстеження, синхронізованих звітів, метаданих експерименту до зовнішніх електронних карток пацієнтів або на платформу медичного моніторингу.

2.2.2 Стандарт IEEE 11073: медичні пристрої та формати даних

IEEE 11073 — міжнародний стандарт для комунікації між медичними пристроями та зовнішніми інформаційними системами. Основна мета — забезпечити уніфіковане подання вимірювань, метаданих, профілів пристроїв, а також протоколів взаємодії.

Ключові риси IEEE 11073:

- Описує моделі пристроїв, ієрархію атрибутів, формати обміну для різних категорій медичних пристроїв (від акселерометрів до моніторів ЕЕГ).
- Підтримує реальний час, що є критичним для потокових біомедичних сигналів.
- Дає змогу стандартизувати обмін структурованими вимірюваннями (часова мітка, значення, одиниці вимірювання, якість сигналу).

У випадку побудови мультимодальних систем збору даних, як у поточній роботі, IEEE 11073 визначає вимоги до формату передачі й опису даних з акселерометрів і ЕЕГ-реєстраторів, а також до сумісності апаратного й програмного забезпечення.

2.3 Архітектура програмного забезпечення

Розроблена система збору та інтеграції біомедичних даних побудована на основі сучасних принципів мікросервісної архітектури, що передбачає чітке розділення бізнес-логіки, модулів обробки даних та шарів зберігання інформації. Це підвищує масштабованість, стійкість до збоїв і полегшує супровід та розширення функціоналу.

Серверна частина реалізована із використанням FastAPI — сучасного асинхронного Python-фреймворка для побудови високопродуктивних веб-сервісів [6]. Зберігання даних організовано у PostgreSQL/TimescaleDB — реляційній СУБД, оптимізованій для великих часових рядів і потоків сенсорних даних [4,5].

Користувачський інтерфейс представлений як веб-додаток, що взаємодіє з API через стандартизовані HTTP(S)-запити.

Модульність архітектури дозволяє при необхідності інтегрувати зовнішні сервіси (наприклад, системи лікарняного обліку, лабораторні інформаційні системи) за допомогою стандартів HL7, FHIR або REST API.

2.3.1 Призначення і взаємодія основних модулів

Модуль завантаження та валідації файлів — забезпечує прийом даних від користувача, перевірку формату, структурної цілісності й метаданих, у відповідності до кращих практик обробки медичних даних [3, 7].

Модуль обробки сигналів — реалізує алгоритми очищення, фільтрації та ресемплінгу даних, що базуються на сучасних наукових підходах до цифрової обробки біомедичних сигналів [1].

Модуль синхронізації та інтеграції — відповідає за точне вирівнювання часових рядів, що критично важливо при аналізі мультимодальних даних у нейроінженерії.

Модуль збереження даних — організує стале зберігання, вибірку й оновлення даних у TimescaleDB, забезпечуючи продуктивну роботу із великими обсягами часових рядів.

Модуль візуалізації та звітності — генерує інтерактивні графіки та метрики для медичного і дослідницького використання [8].

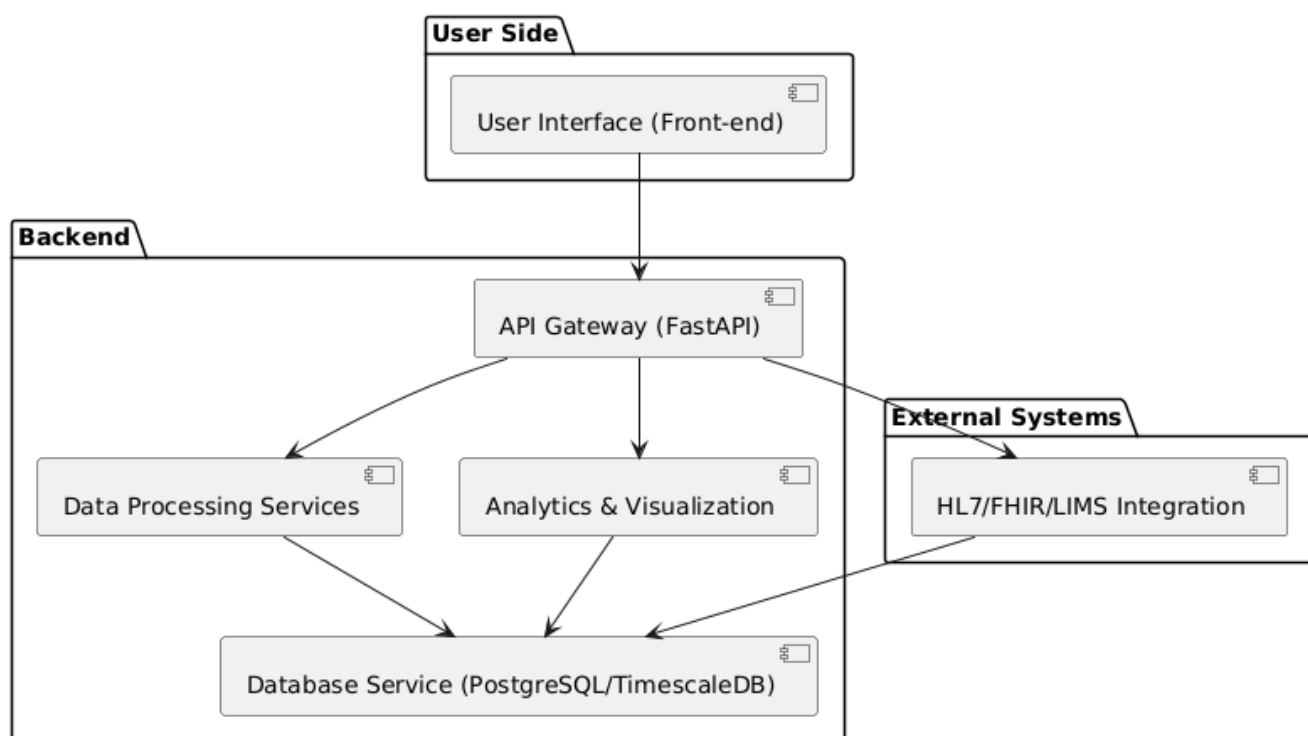


Рисунок 2.3.1.1. Діаграма компонентів

Користувач взаємодіє з веб-інтерфейсом (браузер, мобільний пристрій), надсилає файли на сервер, отримує зворотний зв'язок і візуалізації у реальному часі.

Всі обробки виконуються на сервері асинхронно, що відповідає сучасним підходам до розробки.

Внутрішні модулі системи мають чітко визначені API для взаємодії між собою, що значно спрощує тестування та розширення.

Зовнішні системи (електронна медична картка або лабораторний сервер) можуть інтегруватися через REST API, HL7/FHIR або прямий експорт даних у відповідних форматах.

2.4 Моделювання даних: структура ЕЕГ-звітів і вимірювань акселерометра

Збирання та аналіз біомедичних сигналів вимагає чітко структурованого підходу до організації даних. Всі вимірювання мають зберігатися у вигляді часових рядів, із обов'язковим фіксуванням точних часових міток, ідентифікаторів сесії, каналів і додаткових атрибутів.

Такий підхід відповідає міжнародним рекомендаціям щодо моделювання медичних даних для подальшої інтеграції та аналізу [4,5].

У моїй системі моделі даних сформовані з урахуванням:

- багатоканальної природи ЕЕГ;
- акселерометрії (координат планшета);
- наявності метаданих (інформація про сеанс, пристрій, оператора);
- вимог до коректного зв'язку між різними джерелами даних.

ЕЕГ-звіт (electroencephalography report) зберігає послідовність цифрових значень, отриманих з різних каналів протягом усього сеансу експерименту.

Таблиця 2.4.1. Типова структура одного запису ЕЕГ

Field	Description	Data Type
timestamp_ms	Часова мітка, мілісекунди	Integer
channel_fp1	Значення каналу Fp1	Float

Продовження таблиці 2.4.1

channel_fp2	Значення каналу Fp2	Float
...	...	...
channel_o2	Значення каналу O2	Float
quality_flag	Ознака якості/артефактності	Boolean

Зазвичай у файлі або БД зберігається 17 (або більше) каналів, кожен з яких має своє унікальне ім'я (Fp1, Fp2, F3, F4, ...), що відповідає міжнародній системі 10-20 [9].

```
CREATE TABLE eeg_raw_data (
    id SERIAL PRIMARY KEY,
    timestamp_ms BIGINT NOT NULL,
    session_id UUID REFERENCES eeg_raw_metadata(session_id),
    channel_fp1 REAL,
    channel_fp2 REAL,
    channel_f3 REAL,
    -- ... всі інші канали
    channel_o2 REAL,
    quality_flag BOOLEAN DEFAULT TRUE
);
```

Лістинг 2.4.1 Фрагмент таблиці БД для ЕЕГ

Планшетний звіт містить часові ряди з даними руху пера або акселерометра:

Таблиця 2.4.2. Структура планшетного звіту

Field	Description	Data Type
timestamp_ms	Часова мітка, мілісекунди	Integer
x_mm	Координата X (або прискорення X)	Float
y_mm	Координата Y (або прискорення Y)	Float
z_mm	Координата Z (акселерометр)	Float
pressure	Сила натискання пера	Float
nan_flag	Ознака пропуску даних	Boolean

У планшетних звітах часто трапляються пропуски (наприклад, коли перо відірвано від поверхні) — це фіксується через спеціальний прапорець (`nan_flag`) та/або значення NaN у координатах.

Кожна сесія зберігає унікальні метадані:

- дата і час проведення тесту;
- тип і серійний номер пристрою;
- ПІБ оператора/пацієнта (анонімізовано);

параметри обробки (частота, кількість каналів, версія прошивки пристрою).

Ці дані записуються у відповідні таблиці (наприклад, `eeg_raw_metadata`, `tablet_raw_metadata`) і зв'язуються по `session_id`.

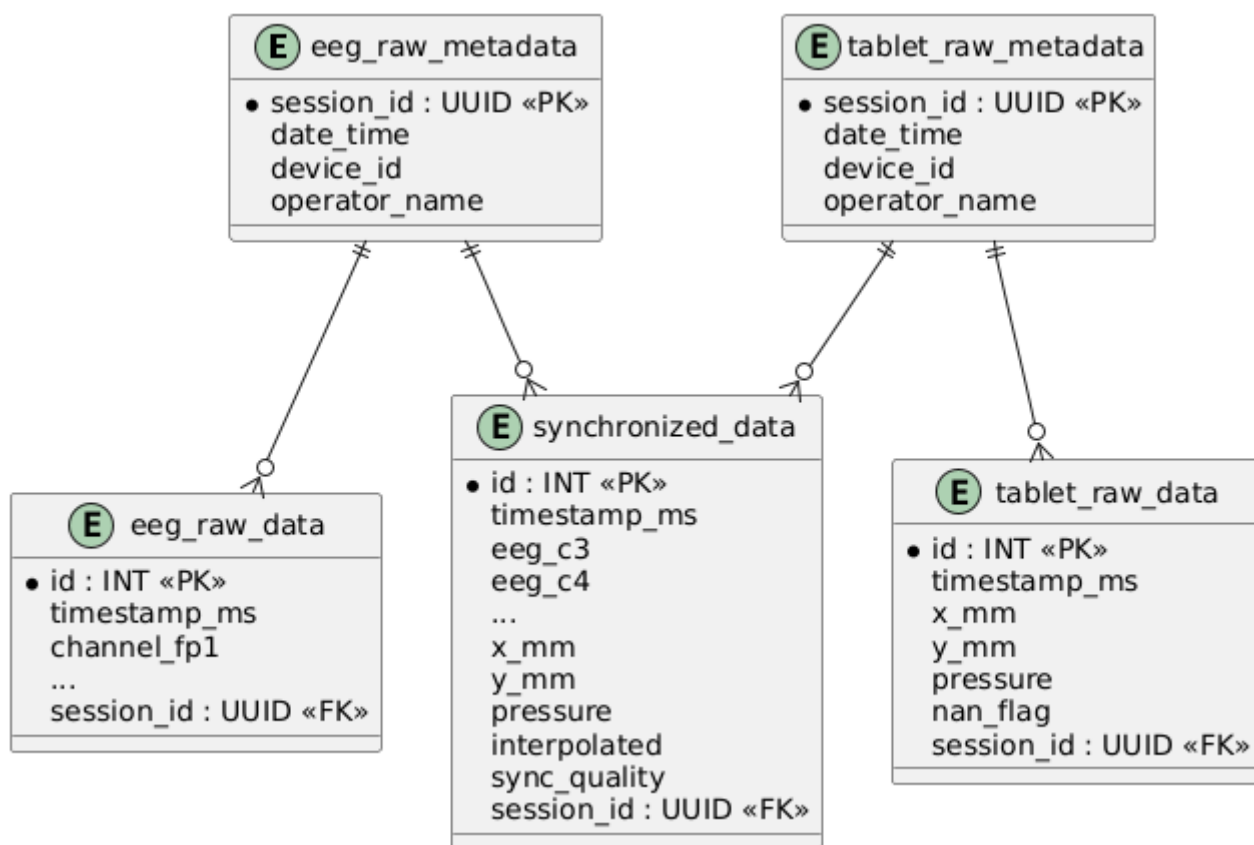


Рисунок 2.4.1. ER-діаграма структури даних

2.5 Інтерфейси для збору, обробки та візуалізації даних

Сучасні біомедичні інформаційні системи висувають жорсткі вимоги до інтерфейсів — як на рівні програмної взаємодії (API), так і у частині кінцевого користувацького досвіду (UX/UI) [1,2]. Інтерфейс має забезпечувати:

- максимальну простоту завантаження даних;
- прозору індикацію етапів обробки;
- зручний та зрозумілий доступ до результатів аналізу;
- підтримку стандартів захисту даних та аутентифікації користувача.

Основним програмним інтерфейсом системи є RESTful API, реалізований за допомогою FastAPI [6].

API надає всі необхідні кінцеві точки (endpoints) для взаємодії із системою на кожному етапі життєвого циклу даних:

/api/upload/eeg — завантаження сирого файлу ЕЕГ;
/api/upload/tablet — завантаження файлу планшета/акселерометра;
/api/validate — перевірка цілісності та структурної відповідності файлів;
/api/synchronize — об'єднання і синхронізація даних;
/api/store/raw — збереження сирих даних у БД;
/api/sessions — отримання списку синхронізованих сесій;
/api/session/{session_id}/spiral — отримання даних для побудови спіралі для візуалізації;
/api/raw-sessions — список сирих сесій у БД;
/api/files/status — моніторинг статусу файлів;
/api/status/database — перевірка статусу БД.

Особливості реалізації:

Усі ключові операції (завантаження, перевірка, обробка) асинхронні, що дозволяє працювати із файлами великого розміру [6,10].

API підтримує повернення структурованих відповідей (JSON) із детальними повідомленнями про помилки, статус, статистику обробки.

Для кожного етапу передбачено відповідний статус/response model — це спрощує інтеграцію з іншими системами (додатками для статистичного аналізу чи електронними медкартами).

```
POST /api/upload/eeg
Content-Type: multipart/form-data

file: <eeg_data_file.txt>
```

Лістинг 2.5.1 Приклад HTTP POST запиту для завантаження файлу:

2.5.1 Веб-інтерфейс для користувача

Користувацький інтерфейс побудований на основі шаблонів Jinja2 (HTML, CSS, Bootstrap) та JavaScript.

Завантаження та синхронізація даних

(upload.html) — користувач обирає файли, бачить покрокову інструкцію та індикатори прогресу.

Всі етапи (upload → validate → sync) мають наочне відображення статусу, можливість очищення файлів та отримання підказок у разі помилки [11].

Аналіз та візуалізація

(analysis.html) — користувач обирає синхронізовану сесію, отримує інтерактивну візуалізацію спіралі (накладання ідеальної та реальної траєкторії), ключові метрики виконання тесту.

Підтримується можливість перегляду попередніх сесій, деталізації окремих точок малюнка, інтерактивної побудови графіків.

Перегляд та експорт сирих даних

(raw_data.html) — перегляд метаданих, структури сесії, завантаження даних для зовнішнього аналізу.

2.5.2 Модуль візуалізації даних та звітності

Інтерфейс підтримує кілька типів візуалізації (на базі Chart.js та кастомного JS):

- Часові графіки руху (Time Series Plots): дозволяють відобразити траєкторію пера/стилуса у часовому розрізі, виявити фрагменти тремору, аналізувати пропуски [12].
- Спіраль Архімеда: головний тест — накладання реальної траєкторії пацієнта на ідеальну спіраль, підрахунок відхилень, плавності, стабільності тиску пера [1].
- Статистичні метрики: час виконання, кількість пропусків, індекс тремору, середня швидкість руху — подаються у вигляді таблиці/графіка з порівнянням різних сесій.
- Збереження й експорт результатів: для подальшого аналізу або додавання у медичну картку.
- Особливості:
 - Динамічне оновлення графіків при зміні сесії або під час завантаження нових даних.
 - Можливість інтеграції з зовнішніми аналітичними модулями через API.
 - Підтримка експорту у стандартних форматах.

Нижче наведено діаграму взаємодій:

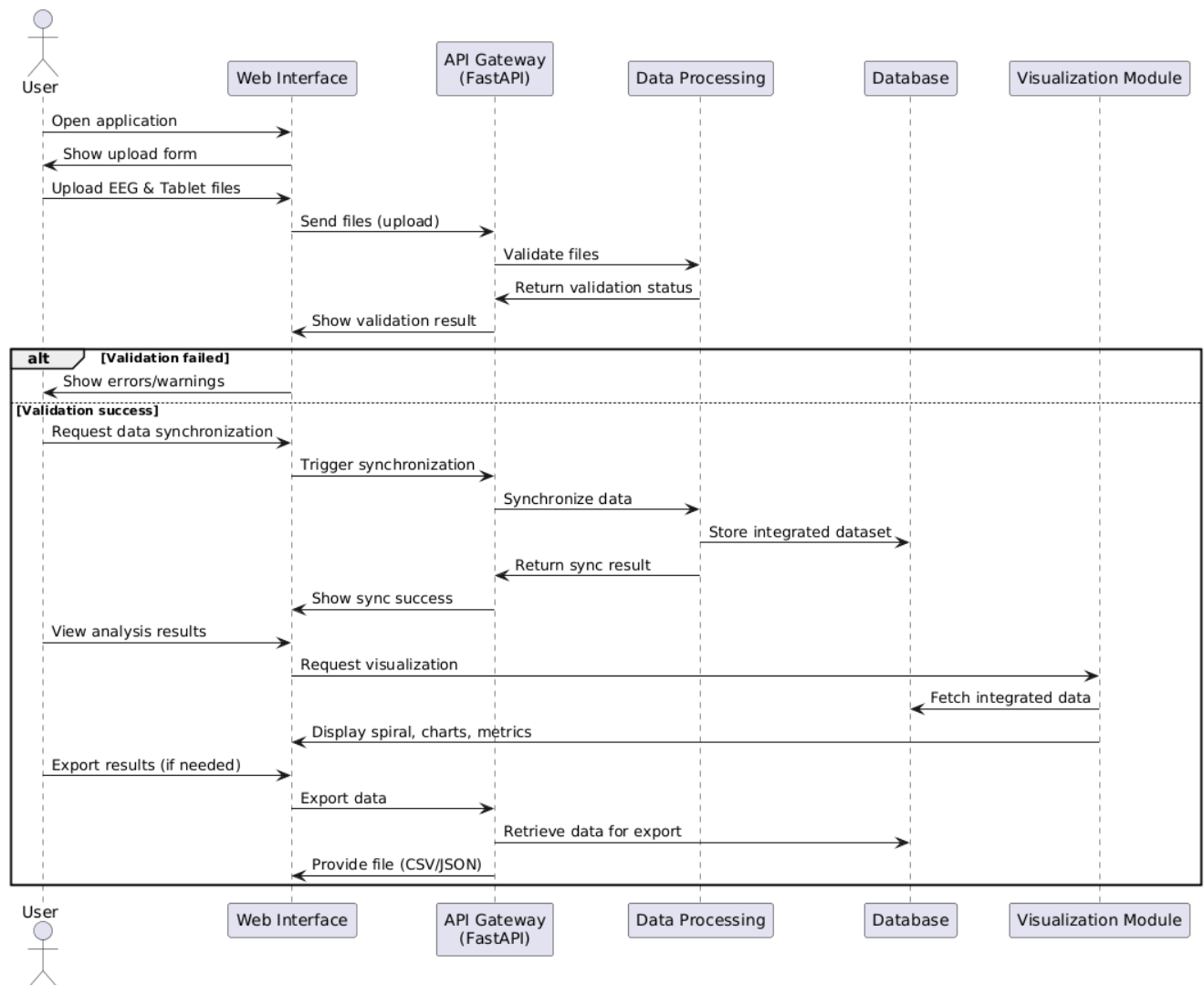


Рисунок 2.5.2.1. Діаграма взаємодій системи

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО МОДУЛЯ

3.1 Вибір технологій (мова програмування, фреймворки, СУБД)

Обраний стек технологій дозволяє забезпечити всі необхідні властивості для ефективної роботи з великими часовими рядами, забезпечити інтеграцію із зовнішніми системами, а також створити зручний користувацький інтерфейс.

Мова програмування:

В основі бекенду лежить Python — одна з найпопулярніших мов у сфері обробки наукових та біомедичних даних.

Python має розвинену екосистему бібліотек для роботи із сигналами, статистики, (NumPy, SciPy, mne) [3,13].

Для реалізації серверної частини використано FastAPI — сучасний асинхронний веб-фреймворк для Python, який забезпечує високу продуктивність, простий опис API, інтеграцію з OpenAPI/Swagger, а також підтримку асинхронної обробки файлів великого розміру

Для зберігання сирих та синхронізованих даних використовується PostgreSQL із розширенням TimescaleDB.

Це дозволяє ефективно працювати з великими масивами часових рядів, виконувати швидкі запити на часових інтервалах, масштабувати зберігання при зростанні кількості сесій.

Інтерфейс реалізовано за допомогою HTML5, CSS (Bootstrap), шаблонів Jinja2, а також JavaScript. Для інтерактивної візуалізації даних застосовуються бібліотеки Chart.js та кастомний JS.

Для управління середовищем — venv, pip, Docker (при необхідності деплою на сервері), для тестування — unittest.

Весь проєкт побудовано за принципом розділення на логічні модулі, що полегшує розширення, тестування і рефакторинг коду.

Кожен модуль відповідає за окрему частину бізнес-логіки:

- обробка й валідація даних;

- синхронізація сигналів;
- зберігання у БД;
- робота з API та інтерфейсом користувача.

Структура каталогу app/:

- data_processor/ — обробка та очищення даних;
- models/ — ORM-моделі для бази даних;
- services/ — сервіси для роботи з raw/sync даними;
- routes.py — опис REST API;
- database.py — налаштування БД;
- templates/ — HTML-шаблони;
- static/ — стилі, скрипти.

Таблиця 3.1.1. таблиця використаних бібліотек

№	Категорія	Бібліотека/Модуль	Призначення
1	Веб-фреймворк	fastapi	Основний сервер для REST API, бекенд логіка
2	ASGI сервер	uvicorn	Запуск і асинхронне обслуговування FastAPI
3	Веб-ядро	starlette	Інфраструктура для FastAPI, обробка запитів/відповідей

Продовження таблиці 3.1.1

4	Шаблони та форми	jinja2, MarkupSafe	HTML-шаблони, динамічні сторінки
5	Робота з файлами	python-multipart, chardet	Завантаження, парсинг, коректне визначення кодування
6	Валідація/типізація	pydantic, annotated-types, typing_extensions, typing-inspection, pydantic_core	Опис та перевірка структур даних, валідація API
7	База даних	psycopg2, sqlalchemy, greenlet	З'єднання з PostgreSQL, ORM (SQLAlchemy), підтримка асинхронної роботи
8	Обробка та аналіз даних	numpy, scipy	Обробка, фільтрація, аналіз сигналів
9	CLI та утиліти	click, tqdm, colorama	Зручні інструменти для командного рядка, кольоровий вивід
10	HTTP та асинхронні інструменти	h11, anyio, sniffio, idna	Базові мережеві й асинхронні залежності

3.2. Розробка основних модулів

3.2.1 Збір даних із пристроїв

У сучасних медичних інформаційних системах збір біомедичних сигналів здійснюється за допомогою спеціалізованих пристроїв, таких як електроенцефалографи (ЕЕГ-реєстратори), графічні планшети чи акселерометри.

У рамках проєкту передбачена можливість завантаження файлів, що містять результати вимірювань з обох типів пристроїв, через веб-інтерфейс системи.

Користувач завантажує сирі текстові звіти із ЕЕГ-апарату та планшета за допомогою інтерфейсу на сторінці “Завантаження та синхронізація”.

Після завантаження файли зберігаються на сервері для подальшої обробки (див. фрагмент із routes.py):

```
@api_router.post("/upload/eeg",  
response_model=FileUploadResponse)  
async def upload_eeg_file(file: UploadFile = File(...)):  
    # Збереження та валідація файлу EEG
```

Лістинг 3.2.1.1. Фрагмент коду для завантаження даних

Система підтримує перевірку формату, структури і кодування завантажених файлів. (Рекомендовано для майбутнього: реалізувати прямий збір даних із пристроїв через API або драйвери у реальному часі, що значно підвищить автоматизацію та швидкість збору інформації.)

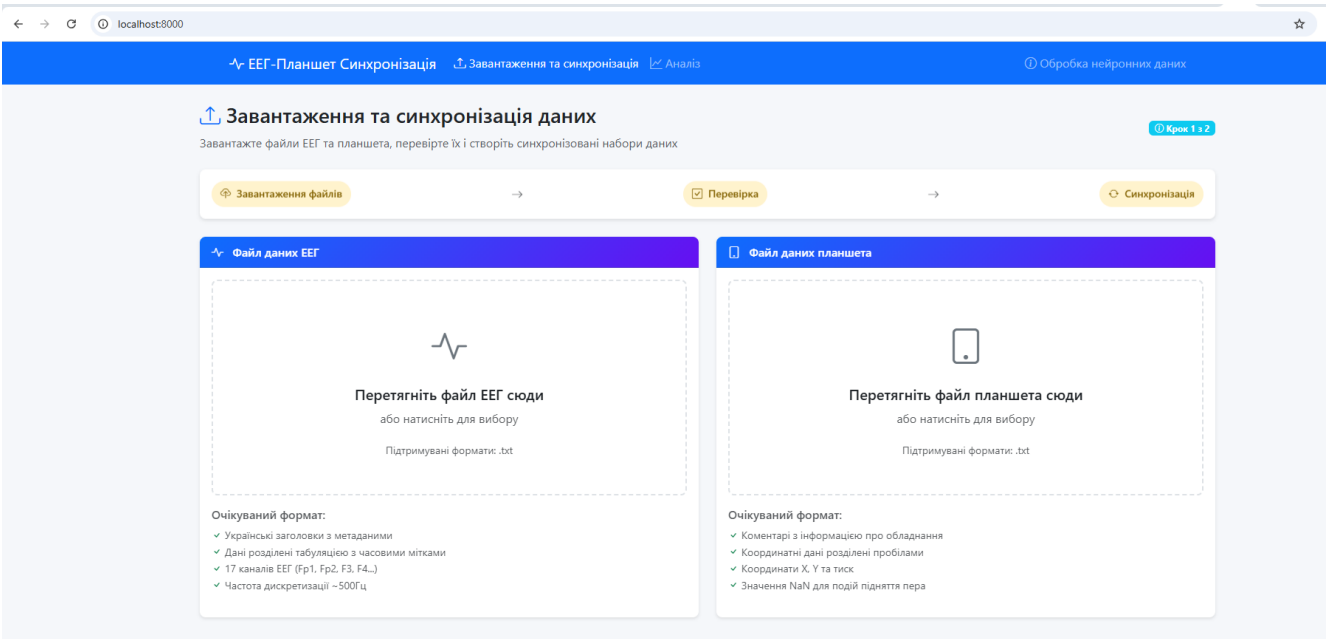


Рисунок 3.2.1.1. Приклад інтерфейсу користувача

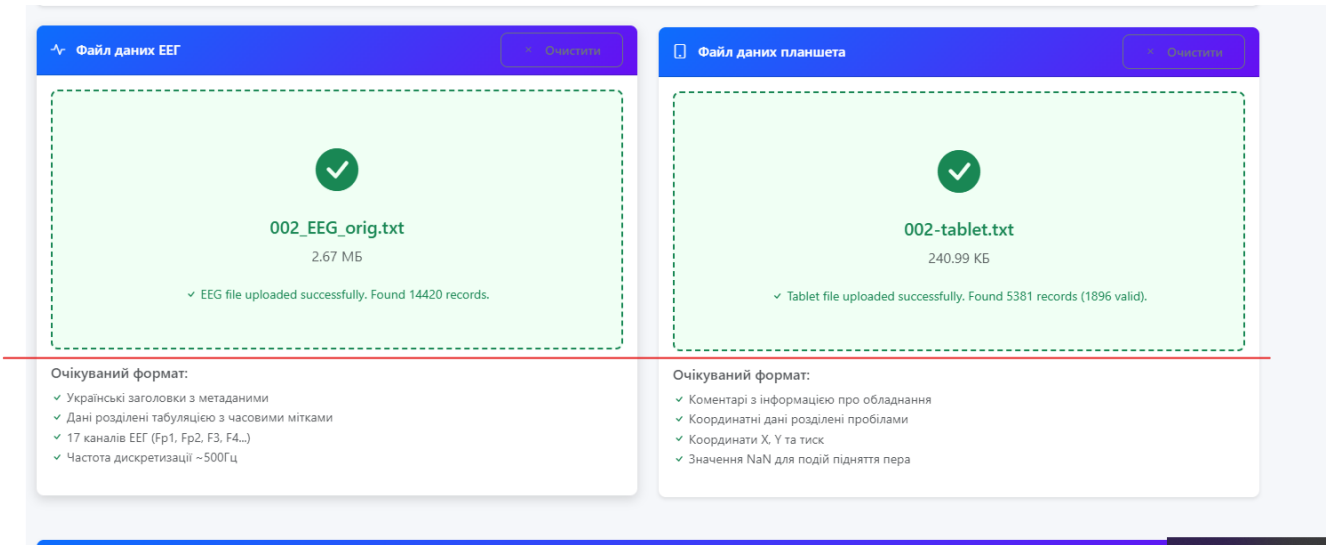


Рисунок 3.2.1.2. Завантажені файли

Впроваджено у кодї:

Дані завантажуються користувачем через веб-інтерфейс (форма upload.html).

На бекенді використовується абстрактний клас парсера, який визначає, як саме відкривати файли, зчитувати кодування, знаходити початок і кінець метаданих і основних записів.

3.2.2 Попередня обробка сигналів

Очищення від артефактів, фільтрація, інтерполяція пропусків, нормалізація, видалення шуму(ЕЕГ).

Впроваджено у коді:

- Детекція артефактів по амплітуді та градієнту.
- Фільтрація каналів (highpass, lowpass, notch).
- Інтерполяція (linear, spline) і заміна на медіану для проблемних ділянок.

Фрагмент реалізації прикріплений у додатку А.

Планшет:

```
# tablet.py
if config.tablet_handle_nan:
    cleaned_data, nan_stats = self.handle_nan_values(cleaned_data,
config).
```

Лістинг 3.2.2.1. Фрагмент реалізації для планшету

Виявлення та обробка NaN, згладжування координат, пошук викидів (за швидкістю, прискоренням, виходом за межі області).

Впроваджено у коді:

Обрізання початкових/кінцевих NaN, інтерполяція коротких ділянок, видалення довгих сегментів NaN.

3.2.3 Зберігання і управління базою даних

Для забезпечення надійності зберігання, швидкого доступу до великих масивів біомедичних часових рядів та можливості масштабування, у системі використовується реляційна база даних PostgreSQL із розширенням TimescaleDB.

Це дозволяє ефективно оперувати як з невеликими, так і з дуже великими обсягами даних, які виникають при тривалому зборі сигналів ЕЕГ та рухів із планшета.

Система організована за принципом розділення на кілька основних типів таблиць:

Таблиці метаданих — містять інформацію про кожну сесію (id сесії, дату, оператора, параметри пристрою, список каналів та інше).

Наприклад: `eeg_raw_metadata`, `tablet_raw_metadata`.

Таблиці сирих даних — зберігають усі значення сигналів для кожної сесії окремо, з деталізацією по кожному каналу, координаті або параметру.

Наприклад: `eeg_raw_data`, `tablet_raw_data`.

Таблиці синхронізованих/інтегрованих даних — містять результати поєднання сигналів з різних джерел у єдину часову сітку, а також позначки якості, інтерполяції тощо.

Наприклад: `synchronized_data`

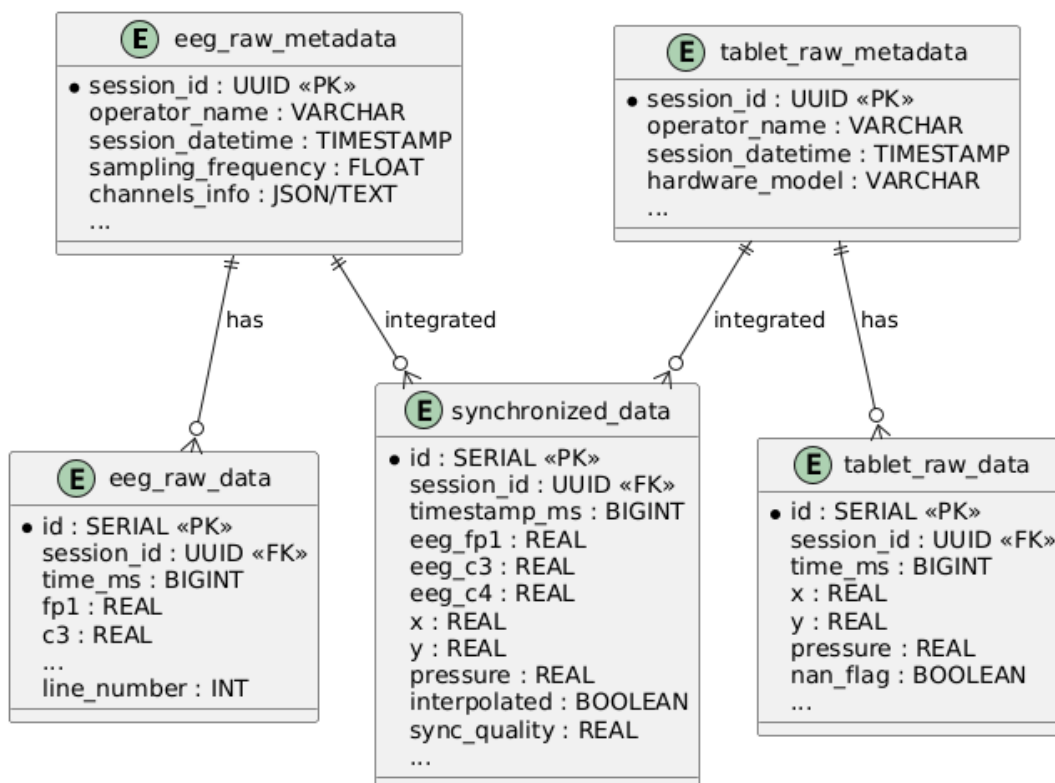


Рисунок 3.2.3.1. Діаграма зв'язків між таблицями

3.3 Інтеграційне тестування та верифікація коректності даних

Інтеграційне тестування системи забезпечує впевненість у тому, що всі основні компоненти (збір, обробка, синхронізація, зберігання та візуалізація біомедичних даних) взаємодіють коректно, а результат відповідає очікуванням користувача.

Такий підхід дозволяє виявити “слабкі місця” саме на стику різних модулів і переконатися, що дані не втрачаються та не спотворюються на жодному етапі.

Нижче у роботі наведено кілька сценаріїв інтеграційної перевірки:

Сценарій 1:

Користувач заходить у веб-інтерфейс, обирає файл із сирими даними ЕЕГ та файл із планшета/акселерометра, завантажує їх через форму.

Що перевіряється:

- Відображення статусу завантаження (успіх/невдача)
- Перевірка коректності формату, кодування, наявності метаданих
- Система повертає детальний звіт про стан кожного файлу (кількість каналів, точок, дата експерименту)

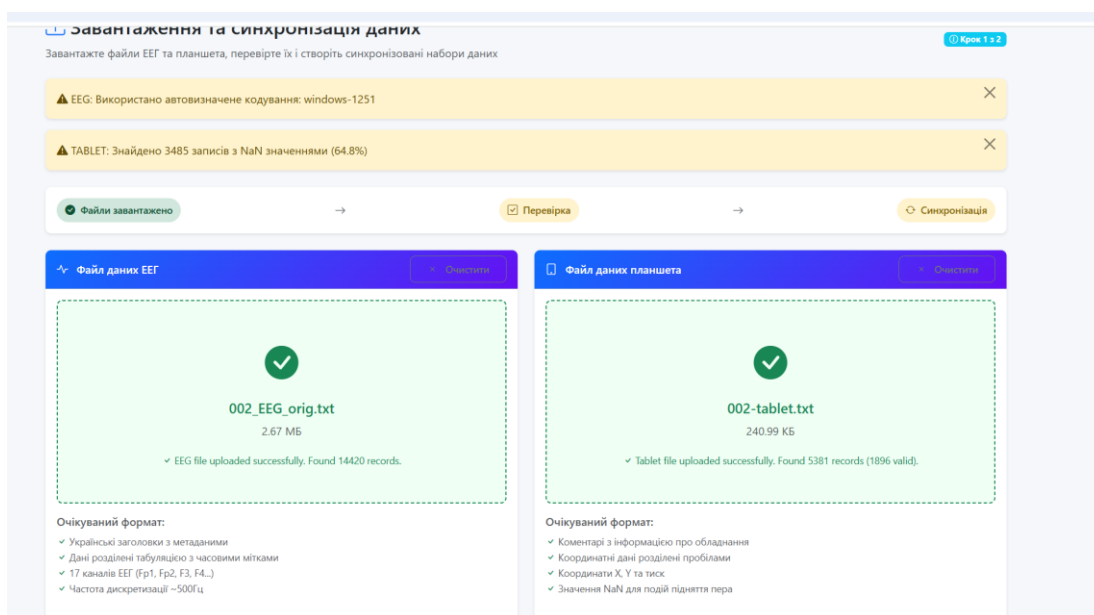


Рисунок 3.3.1. Завантаженні дані з пристроїв

Сценарій 2:

Після завантаження файлів система автоматично очищає дані, видаляє артефакти, проводить інтерполяцію пропусків, вирівнює сигнали по часу.

Що перевіряється:

- Якість очищення (наприклад, зменшення кількості пропусків у даних)
- Коректність вирівнювання часових рядів (часові мітки співпадають)
- Відсутність втрат сигналу або появи некоректних значень

Приклад використання:

- На вкладці “Синхронізація” відображається звіт про кількість інтерпольованих точок, розподіл “чистих” і “проблемних” даних.
- Можна візуально порівняти графік “до” та “після” обробки.

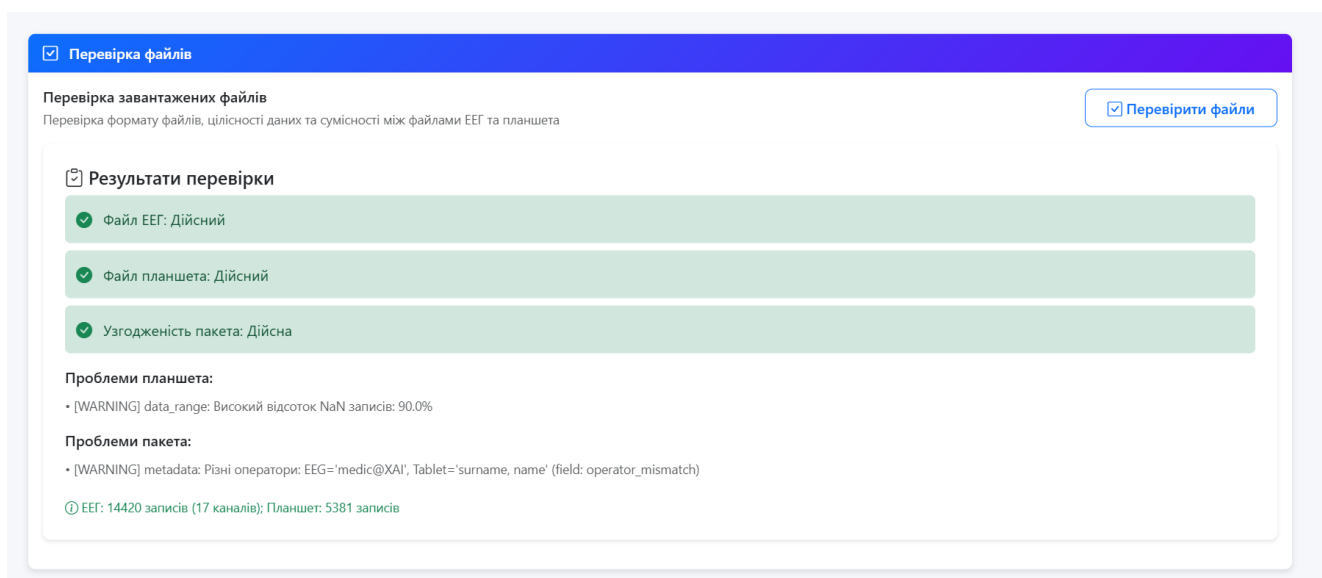


Рисунок 3.3.2. Результат перевірки файлів

Синхронізація даних

Синхронізація даних ЕЕГ та планшета

Синхронізувати дані

ⓘ Потребує перевірених файлів

Процес включає очищення даних, часове вирівнювання та передискретизацію до уніфікованої частоти. Це може зайняти кілька хвилин залежно від розміру файлів.

Назва сесії

Сесія 24.06.2025 14:10:08

Опціонально: дайте вашій сесії зрозумілу назву для легкої ідентифікації

Кроки обробки:

① Очищення даних (видалення артефактів, фільтрація)

② Вирівнювання часових міток (взаємна кореляція)

③ Передискретизація до уніфікованої частоти (250Гц)

✓ Синхронізацію завершено

Synchronization completed successfully. Quality: 0.80. Stored to database.

14420

Записи ЕЕГ

5381

Записи планшета

3444

Синхронізовано

80%

Оцінка якості

36900мс

Час обробки

Попередження обробки:

• Removed 0 long NaN segments

• Removed 0 NaN records from long segments

• Removed 0 long NaN segments

• Removed 0 NaN records from long segments

✓ Переглянути аналіз

Рисунок 3.3.3. Результат синхронізації

Згідно результатів інтеграції було синхронізовано 3444 точок за часовими мітками. Також функціонал дозволяє експортувати дані у JSON форматі на локальну машину, нижче наведений приклад з UI.

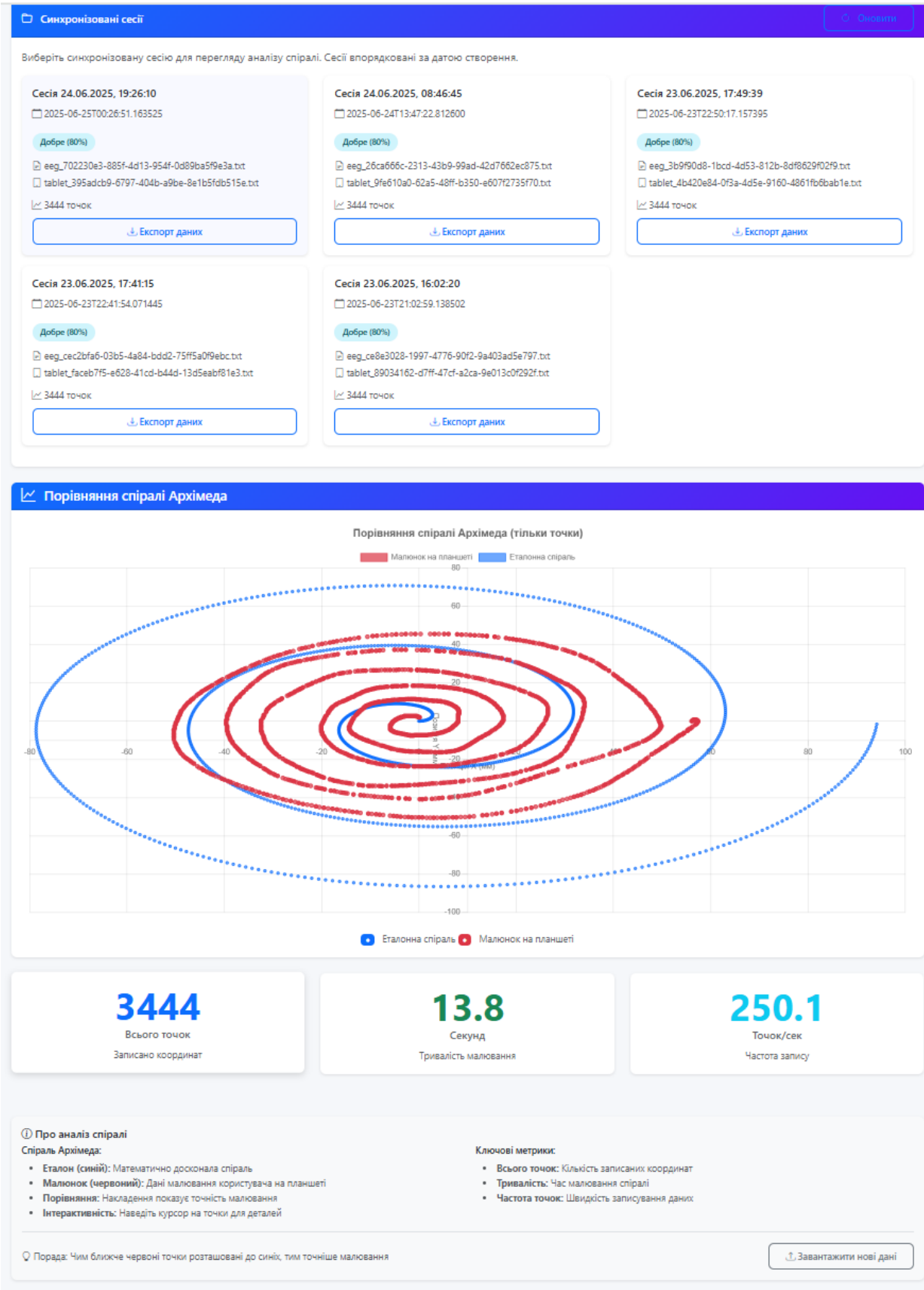


Рисунок 3.3.4. Результат виконання на спіралі Архімеда

3.4. Оцінка продуктивності та надійності системи

Завдяки використанню TimescaleDB система демонструє високу швидкість доступу до даних, навіть при збереженні десятків або сотень сесій. Запити на вибірку за `session_id` чи часовим інтервалом виконуються за долі секунди.

Індексація по ключових полях (`session_id`, `time_ms`, `timestamp_ms`) гарантує масштабованість при збільшенні кількості записів.

Використання векторизованих операцій NumPy та функцій SciPy дозволяє значно скоротити час виконання основних етапів — очищення сигналу, інтерполяції, фільтрації, ресемплінгу.

Перевага полягає у тому, що більшість математичних операцій виконуються на рівні C-модуля всередині бібліотек, що зменшує навантаження на процесор.

Завдяки асинхронній обробці FastAPI та можливості використання кількох воркерів у uvicorn, система справляється із паралельним обслуговуванням декількох користувачів — одночасне завантаження кількох сесій різними користувачами не призводить до суттєвого зниження швидкодії.

На етапі завантаження та попередньої обробки система проводить глибоку валідацію структури даних (перевіряє кодування, формат, кількість каналів, коректність часових міток). Виявлені помилки не призводять до “падіння” системи, а повертаються користувачу у вигляді детальних попереджень чи списку помилок для виправлення.

Всі критичні операції з даними (запис у базу, обробка великих файлів) супроводжуються веденням журналу подій (logging).

При виникненні помилок, транзакції у БД автоматично відкочуються (rollback), що захищає від часткової втрати інформації.

Система підтримує повторну обробку файлів, які не пройшли перевірку, без необхідності повторного завантаження. Для кожної сесії генерується унікальний ідентифікатор, що гарантує відсутність “перезапису” або змішування даних різних тестів.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при кровотечах

Якщо поруч із вами знаходиться людина з кровотечею, а фахівці ще не наспіли, необхідно надати потерпілому першу допомогу. Для цього важливо визначити характер кровотечі, адже при різних травмах заходи долікарської допомоги будуть відрізнятися.

Кровотеча — це витікання крові із кровоносних судин при порушенні їхньої цілісності, відповідно існують різні види кровотеч, при кожній з яких надаються різні типи долікарської допомоги. Розрізняють капілярну, венозну і артеріальну кровотечі [18].

Капілярна кровотеча виникає навіть при незначному пораненні, оскільки кров по капілярах тече повільно і під невеликим тиском, то такі кровотечі не призводять до значної втрати крові і легко зупиняються. Пошкоджені капіляри швидко закриваються тромбом, який утворюється при зсіданні крові. Першою допомогою при капілярних кровотечах є знезараження місця поранення перекисом водню або йодною настоякою, а також накладання на нього чистої пов'язки.

Венозна кровотеча виникає при ушкодженні поверхневих вен, і особливо коли ушкоджені великі вени, зсідання крові не здатне швидко зупинити кровотечу, через що за короткий час можуть бути значні крововтрати. При венозних кровотечах кров витікає рівномірно і має темний колір, і для її зупинки достатньо накладати компресійну пов'язку. Перед цим рану потрібно знезаразити, а для зменшення кровотечі тимчасово притиснути пошкоджену судину.

Артеріальна кровотеча особливо небезпечна для життя, вона має вигляд пульсуючого струменя, подібного до фонтану, і кров має яскраво-червоний колір. У цьому разі необхідно діяти негайно: перш за все вище місця поранення швидко притискають пальцями ушкоджену судину у точках, де прощупується пульс, і накладають джгут. Щоб не пошкодити нерви і шкіру, джгут накладають поверх одягу, хустки або іншої м'якої тканини. Якщо джгута немає, можна скористатися

ременем або зробити закрутку з будь-якого шнура чи тканини. Для цього між тканиною і тілом вставляють міцну палицю і закручують тканину до зупинки кровотечі, а потім палицю прибинтовують до тіла. Джгут залишають на кінцівках не більше ніж на 1,5–2 години, а в холодну пору року — на 1 годину, інакше може настати омертвіння тканини. Щоб цього не сталося, до джгута приколують записку з точним зазначенням часу його накладання. Якщо потерпілого за цей час неможливо доставити до лікарні, через 1,5–2 години джгут послаблюють на 1–2 хвилини, а якщо кровотеча продовжується, джгут знову затягують.

Існують також зовнішня і внутрішня кровотечі. При зовнішній кровотечі кров витікає через рану в шкірі чи у видимих слизових оболонках або з порожнин, а при внутрішній кровотечі кров виливається в тканини й органи тіла, що називається крововиливом. Швидка і значна втрата крові є дуже небезпечною, оскільки супроводжується зниженням кров'яного тиску, порушенням кровопостачання мозку, серця і всіх інших органів, тому це може стати причиною загибелі людей, яких ще можна було б врятувати, надавши своєчасну першу допомогу.

Внутрішні кровотечі в черевну порожнину, порожнину грудей чи черепа є надзвичайно небезпечними, і встановити їх наявність можна лише за зовнішнім виглядом людини: вона стає блідою, у неї виступає холодний піт, а пульс частішає і слабне. У такому разі потрібно негайно викликати швидку допомогу, а до її прибуття потерпілого кладуть або напівсаджають і не рухають з місця, а до ймовірного місця кровотечі (живота, грудей, голови) прикладають холодний компрес — мішечок із льодом чи снігом, грілку або пляшку з холодною водою [20].

Отже, знаючи, які є типи кровотеч і як з ними боротися, можна надати необхідну долікарську допомогу при різних пораненнях тіла, а ця інформація дуже важлива, адже така ситуація може трапитися з кожним, тому важливо знати, як надавати правильну допомогу вчасно.

4.2 Заходи, що покращують умови праці оператора ПК.

Діяльність більшості працівників сучасних професій у виробничій сфері пов'язана з використанням комп'ютерної техніки, яка для сучасної людини є такою ж необхідністю, як телевізор або холодильник [21]. Побутові прилади ми використовуємо, не задумуючись про їх шкідливість або нешкідливість, усвідомлюючи лише переваги їх наявності, тоді як про комп'ютери існує багато інформації як про їх безпечність, так і шкідливість.

Осіб, які працюють з комп'ютерами, поділяють на такі групи: розробники програм (інженери-програмісти) мають справу переважно з відеотерміналами, їх робота характеризується інтенсивною розумовою творчою працею з підвищеним напруженням зору, концентрацією уваги на фоні нервово-емоційного напруження, вимушеною робочою позою, загальною гіподинамією, а також періодичним навантаженням на кисті рук; оператори електронно-обчислювальних машин виконують роботу, пов'язану з обліком інформації, одержаної з візуального дисплейного терміналу за попереднім запитом або тієї, що самостійно надходить з нього, яка супроводжується перервами різної тривалості, пов'язана з виконанням іншої роботи й характеризується як робота з напруженням зору, невеликими фізичними зусиллями, нервовим напруженням середнього ступеня та виконується у довільному темпі; оператори комп'ютерного набору займаються одноманітною за характером роботою з документацією та клавіатурою, під час якої нечасто та ненадовго переключають погляд на екран дисплея, вводять дані з високою швидкістю, а така робота характеризується як фізична праця з підвищеним навантаженням на кисті рук на фоні загальної гіподинамії, з напруженням зору (фіксація зору переважно на документи) та нервово-емоційним напруженням.

Для забезпечення комфортної роботи оператора ПК необхідно звернути увагу на такі основні вимоги: стан виробничих приміщень; характеристики та стан техніки (ПК, монітори, принтери, наявність заземлення тощо); організація робочого місця; дотримання режимів праці та відпочинку; моніторинг стану здоров'я

працівника (стан органів зору, показники захворювань із тимчасовою втратою працездатності тощо); профілактичні заходи.

Важливим елементом комфортної роботи є тиша. Так, нормативними значеннями еквівалентного рівня звуку є: 50 дБА для програмістів, 65 дБА для операторів у залах оброблення інформації, 75 дБА для операторів у приміщеннях, де розташовані гучні агрегати. За результатами гігієнічних досліджень, еквівалентні рівні шуму на робочих місцях працівників офісних приміщень знаходяться у межах 48–59 дБА еквівалентно, і зазвичай перевищення рівня шуму зумовлене телефонними розмовами співробітників, особливо у приміщеннях із великою кількістю робочих місць.

Для комфортної роботи важливо, щоб мікроклімат у приміщенні відповідав стандартам, адже температура та вологість повітря впливають на самопочуття, стан слизових оболонок очей, верхніх дихальних шляхів і шкіри персоналу офісів. Згідно зі стандартами мікроклімату: у теплий період року температура повітря має бути у межах 22–25°C, швидкість руху повітря — до 0,1 м/с, відносна вологість повітря — 40–60%; у холодний період року температура повітря може коливатися у межах 21–24°C, швидкість руху повітря — до 0,1 м/с, вологість повітря - 40–60%.

Не менш важливою умовою комфортної роботи є правильний режим праці та відпочинку, оскільки оператор персонального комп'ютера працює з монітором, а найбільше втомлюються саме очі, тому напруженість роботи прямо пропорційна з напруженістю очей і тривалістю зосередження уваги, яка може становити понад 75% часу. Тому очі найбільше страждають під час роботи з комп'ютером. Велике значення мають відстань до екрана, шрифт, розмір тексту на моніторі, наявність чи відсутність мерехтіння, яскравість екрану, освітлення робочого місця, а також наявність перерв у роботі. Ігнорування таких, на перший погляд, простих речей часто призводить до погіршення зору та хвороб очей [19].

Для збереження здоров'я працівників, запобігання професійним захворюванням і підтримки працездатності слід дотримуватись регламентованих перерв для відпочинку: для розробників програм — 15 хвилин через кожну годину

роботи за комп'ютером, для операторів ЕОМ — 15 хвилин через кожні дві години, для операторів комп'ютерного набору — 10 хвилин після кожної години роботи.

Оскільки на напруженість очей впливає також неправильне положення при роботі, працівникам слід дотримуватись правильної робочої пози, адже неправильна поза шкідливо впливає на скелетно-м'язову систему. Для забезпечення правильної пози тіла при роботі необхідно мати правильно організоване робоче місце.

Для правильної організації робочого місця можна звернутися до нормативного документа ДСТУ 8604:2015, у якому описано, що робоче місце працівника має відповідати таким умовам: світло повинно падати зліва для правшів і навпаки для лівшів; при розміщенні декількох робочих місць в одному приміщенні мінімальна площа для одного місця має бути достатньою; екран монітора має бути на оптимальній відстані від очей користувача (60–70 см, але не ближче 50 см); екран розташовується для забезпечення комфортного зорового спостереження у вертикальній площині під кутом $+30^\circ$ до нормальної лінії погляду; поверхня клавіатури має бути з антистатичними властивостями; під час роботи обов'язково потрібно робити перерви для розвантаження очей.

Отже, дотримуючись цих правил, можна уникнути небажаних професійних хвороб, покращити умови праці користувача персонального комп'ютера й уникнути небажаних недуг.

ВИСНОВКИ

Результатом роботи є проведений глибокий теоретичний аналіз основ акселерометрії, електроенцефалографії та специфіки рухових тестів для діагностики тремору. Встановлено, що інтеграція даних від різних сенсорних пристроїв, зокрема ЕЕГ та акселерометрів/графічних планшетів, дозволяє отримати значно більше інформації для об'єктивної оцінки стану пацієнта.

Огляд сучасних методів, алгоритмів та програмних рішень засвідчив, що найбільш перспективним напрямком є побудова мультимодальних платформ, які поєднують збір, обробку та візуалізацію сигналів у єдиному цифровому просторі. Було доведено доцільність застосування комплексного аналізу моторної активності спільно із електрофізіологічними показниками мозкової діяльності.

Було виконано системний аналіз вимог до програмного комплексу — визначено функціональні та нефункціональні характеристики, окреслено сценарії взаємодії користувачів із системою, обґрунтовано вибір стандартів медичного обміну (HL7, IEEE 11073).

Запропоновано модульну мікросервісну архітектуру, яка забезпечує масштабованість, простоту підтримки та можливість подальшої інтеграції з зовнішніми медичними системами.

Ретельно змодельовано структуру даних: сформовано формат представлення сирих і оброблених сигналів, організовано систему метаданих і забезпечено підтримку часових рядів великої щільності.

Описані програмні інтерфейси для збору, обробки, синхронізації та візуалізації даних є достатньо гнучкими для подальшого розвитку функціоналу платформи, у тому числі для додавання нових типів сенсорів чи медичних протоколів.

На практиці реалізовано програмний модуль для збору, інтеграції та аналізу біомедичних сигналів. Обґрунтовано вибір сучасних технологій — Python, FastAPI,

TimescaleDB/PostgreSQL, Chart.js — що забезпечило баланс між швидкістю розробки, продуктивністю й гнучкістю архітектури.

Впроваджено алгоритми парсингу, валідації, очищення, синхронізації й об'єднання даних, які довели свою ефективність під час інтеграційного тестування. Система забезпечила високу швидкодію при роботі з великими масивами даних, а також стійкість до помилок завдяки поетапній валідації й веденню журналу подій.

Практичні приклади експлуатації довели, що комплекс дозволяє якісно і надійно зберігати, обробляти й аналізувати біомедичні дані, а також надавати медичному персоналу наочний інструмент для візуалізації результатів тестів і прийняття клінічних рішень.

Поставлені у роботі завдання виконано повністю: створено програмне забезпечення, яке поєднує сучасні алгоритми обробки біомедичних даних, зручний інтерфейс, підтримку міжнародних стандартів та ефективне управління даними.

Розроблена система може бути використана як у наукових, так і у клінічних цілях, із можливістю подальшого розширення — підключення нових модулів обробки, розширення бази пристроїв, інтеграції із зовнішніми медичними платформами.

Завдяки високій продуктивності, надійності й структурованості розв'язання, результати дипломної роботи можуть бути застосовані для впровадження у практичну медицину, а також слугувати основою для подальших досліджень у галузі медичної інформатики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Patel S., Park H., Bonato P., Chan L., Rodgers M. A review of wearable sensors and systems with application in rehabilitation // Journal of NeuroEngineering and Rehabilitation. 2009. Vol. 6, Article number: 21. Режим доступу: <https://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-6-21> (дата звернення: 23.05.2025).
2. Zhao Y., Huang X., Li C., et al. A Systematic Review of Machine Learning Methods for Multimodal EEG Data in Clinical Application [Електронний ресурс]. 2024. Режим доступу: https://www.researchgate.net/publication/388067986_A_Systematic_Review_of_Machine_Learning_Methods_for_Multimodal_EEG_Data_in_Clinical_Application (дата звернення: 24.05.2025).
3. MNE: Python бібліотека для аналізу нейрофізіологічних даних [Електронний ресурс]. Режим доступу: <https://mne.tools/stable/index.html> (дата звернення: 23.05.2025).
4. HL7 International. Official website [Електронний ресурс]. Режим доступу: <https://www.hl7.org/> (дата звернення: 24.05.2025).
5. IEEE 11073 Standard. Health informatics – Medical / health device communication standards [Електронний ресурс]. Режим доступу: <https://standards.ieee.org/> (дата звернення: 28.05.2025).
6. FastAPI Documentation [Електронний ресурс]. Режим доступу: <https://fastapi.tiangolo.com/> (дата звернення: 27.06.2025).
7. ISO 18308:2011. Health informatics – Requirements for an Electronic Health Record Architecture.
8. Shamlo N., Youn C., Singh G., et al. Data visualization for medical research // BMC Medical Informatics and Decision Making. 2020.
9. Klem G.H., Lüders H.O., Jasper H.H., Elger C. The ten-twenty electrode system of the International Federation. The International Federation of Clinical

Neurophysiology // Electroencephalogr Clin Neurophysiol Suppl. 1999. Vol. 52. С. 3–6.
PMID: 10590970;

10. Async File Uploads in FastAPI [Електронний ресурс]. Режим доступу: <https://fastapi.tiangolo.com/advanced/custom-request-and-route/> (дата звернення: 01.06.2025);

11. Bootstrap Documentation [Електронний ресурс]. Режим доступу: <https://getbootstrap.com/> (дата звернення: 04.06.2025);

12. Chart.js Documentation [Електронний ресурс]. Режим доступу: <https://www.chartjs.org/docs/> (дата звернення: 10.06.2025);

13. Python Official Site [Електронний ресурс]. Режим доступу: <https://www.python.org/> (дата звернення: 20.05.2025);

14. ДСТУ 7234:2011. Дизайн і ергономіка. Обладнання виробниче. Загальні вимоги дизайну та ергономіки;

15. Бедрій І.Я., Нечай В.Я. Безпека життєдіяльності: навчальний посібник. Львів: Манголія, 2006. 499 с;

16. ДСТУ EN ISO 9241-5:2024. Ergonomics of human-system interaction – Part 5: Workstation layout and postural requirements [Електронний ресурс]. Режим доступу: <https://www.iso.org/standard/86222.html> (дата звернення: 29.05.2025);

17. Андрейчук Н.І., Кіт Ю.В., Шибанов С.В., Шерстньова О.В. Охорона праці: навчальний посібник. Львів: Видавництво Львівська політехніка, 2021. 276 с.

18. Желібо Є.П., Зацарний В.В. Безпека життєдіяльності: підручник. Київ: Каравела, 2023. 344 с;

19. Жидецький В.Ц. Охорона праці: підручник. Львів: Афіша, 2020. 176 с;

20. Книга Безпека життєдіяльності. Навчальний посібник Павло Атаманчук, 2017. 276 с;

21. Грибан В.Г. Охорона праці: навч. посібник / В.Г. Грибан, О.В. Негодченко. – Київ: Центр учбової літератури, 2018. 280 с.

22. Мартін Р. С. (Дядечко Боб). Чиста архітектура: мистецтво розробки програмного забезпечення / Р. С. Мартін. – Київ : Фабула, 2019. – 368 с.

23. Дистанційний курс «Кваліфікаційні роботи бакалаврів» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: URL: <https://dl.tntu.edu.ua/bounce.php?course=5329>;
24. Multi-sensor analysis of cognitive signals for neurological disorders and diseases [Petryk, M.](#), [Brevus, V.](#), [Bachynskyi, M.](#), [Legrand, A.P.](#), [Zaiarnyi, M.](#) Ceur Workshop Proceedings, 2024, 3742, pp. 304–315;
25. Leveraging graphics tablet and JPen library to detect essential tremor [Osiichuk, I.](#), [Brevus, V.](#), [Bishchak, D.](#), [Mashtaliar, Y.](#), [Mudryk, I.](#) Ceur Workshop Proceedings, 2024, 3742, pp. 111–126;

ДОДАТКИ

ДОДАТОК А – Лістинг частини модулів

Автодетекція кодування, розбір секцій файлу:

```
# file_parsers.py
def _detect_file_encoding(self, file_path: Path) -> str:
    with open(file_path, 'rb') as f:
        raw_data = f.read(10000)
        detection = chardet.detect(raw_data)
        detected_encoding = detection.get('encoding', 'utf-8')
        # Перелік можливих кодувань для українських файлів
        encodings_to_try = [detected_encoding, 'cp1251', 'utf-8', ...]
        for encoding in encodings_to_try:
            try:
                with open(file_path, 'r', encoding=encoding) as f:
                    test_content = f.read(1000)
                    if 'ЕЕГ' in test_content or 'Huion' in test_content:
                        return encoding
            except (UnicodeDecodeError, UnicodeError):
                continue
        return detected_encoding if detected_encoding else 'cp1251'
```

Фрагмент реалізації детекції артефактів та фільтрація для ЕЕГ.

```
# eeg.py
if config.eeg_apply_filters:
    cleaned_data = self.apply_filters(cleaned_data, config)

if config.eeg_process_artifacts and config.eeg_detect_artifacts:
    artifact_segments = self.detect_artifact_segments(cleaned_data,
config)
    cleaned_data, processing_stats = self.process_artifact_segments(
        cleaned_data, artifact_segments, config
    )
```

ДОДАТОК Б – Диск з кваліфікаційною роботою бакалавра