

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

Факультет комп'ютерно-інформаційних систем та програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: проектування інформаційної системи реєстрації та зберігання ЕЕГ  
когнітивних сигналів керування елементами опорно-рухового механізму  
пацієнтів з використанням об'єктно орієнтованих архітектур

Виконав(ла): студент(ка) 4 курсу, групи СП-42  
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Ямко В. О.

(підпис)

(прізвище та ініціали)

Керівник

Петрик М. Р.

(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  
2023

## АНОТАЦІЯ

Кваліфікаційна робота бакалавра на тему «Проектування інформаційної системи реєстрації та зберігання ЕЕГ когнітивних сигналів керування елементами опорно-рухового механізму пацієнтів з використанням об'єктно орієнтованих архітектур» написана Ямко Владиславом Олексійовичем, студентом Тернопільського національного технічного університету імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-42.

Відомості про обсяг: сторінок – 59, рисунків – 29, таблиць – 0, частин – 4, додатків – 1, посилань – 15, формул – 10.

Метою наукової роботи є створення апаратно-програмного комплексу для автоматизованої діагностики есенціального тремору. Для цього будуть використовуватись методи математичного та комп'ютерного моделювання для обробки вхідних даних за розробленими математичними моделями та алгоритмами. Бібліотека алгоритмів та програмне забезпечення дозволять створювати комп'ютерні моделі та візуалізувати результати досліджень, а також забезпечать зручність виконання порівняння результатів та оцінки стану пацієнтів.

У результаті роботи було створено систему, яка аналізує неврологічні рухи відповідно до встановлених шаблонів. Для досліджень було зібрано базу записів пацієнтів та діагностичних даних. Зібрано та валідовано набори даних для аналізу ефективності моделей. Дана бібліотека містить набір алгоритмів для визначення ступеня захворювання на есенціальний тремор у вигляді модуля програмної системи. Набір інструментів для розробки прикладного програмного забезпечення включає: мову програмування JavaScript та її бібліотеки, середовище розробки Visual Studio Code IDE для моделювання та розробки, методологію гнучкої (Agile) розробки програмного забезпечення та об'єктно-орієнтований підхід до проектування та реалізації.

## ABSTRACT

Bachelor's qualification work on the topic "Design of an Information System for Registration and Storage of EEG Cognitive Control Signals for Patients' Support-Motor Mechanism Elements Using Object-Oriented Architectures" was written by Yamko Vladyslav Oleksiyovych, a student of Ivan Puluj Ternopil National Technical University, Faculty of Computer and Information Systems and Software Engineering, Department of Software Engineering, group SP-42.

Information about the scope: pages - 59, figures - 29, tables - 0, sections - 4, appendices - 1, references - 15, formulas - 10. The aim of the research work is to create a hardware-software complex for automated diagnostics of essential tremor. To achieve this, methods of mathematical and computer modeling will be used for processing input data using developed mathematical models and algorithms. The algorithm library and software will enable the creation of computer models and visualization of research results, as well as facilitate the comparison of results and assessment of patients' condition.

As a result of the work, a system was created that analyzes neurological movements according to established patterns. A database of patient records and diagnostic data was collected. Data sets for the analysis of model effectiveness were collected and validated.

This library contains a set of algorithms for determining the degree of essential tremor disease as a module of the software system.

The set of tools for developing application software includes the JavaScript programming language and its libraries, JavaScript, the Visual Studio Code development environment for modeling and development, the Agile software development methodology, and an object-oriented approach to design and implementation.

## Перелік умовних позначень, символів, скорочень і термінів

ПЗ – програмне забезпечення.

ПС – програмна система, комплекс програмного забезпечення.

ПК – персональний комп'ютер, робоча машина для розробки та виконання програм.

ВДТ – візуальний дисплейний термінал.

HTML — стандартизована мова розмітки документів для перегляду веб-сторінок у браузері.

CSS — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.

JS – (JavaScript) динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript.

React (React.js) — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Meta і спільнотою індивідуальних розробників.

Node.js — платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Засновником платформи є Раян Дал.

ООП – (Об'єктно-орієнтований програмування) парадигма програмування, в якій основою є класи та об'єкти, які між собою взаємодіють.

UML – (Unified Modeling Language) уніфікована мова графічного представлення та об'єктного моделювання в області розробки програмного забезпечення парадигми об'єктно-орієнтованого програмування.

Алгоритм – набір інструкцій, які описують порядок виконання дій, що дозволяють досягти результату за скінченну кількість кроків.

Програмна бібліотека – пакет підпрограм або об'єктів, класів, що використовуються в розробці програмного забезпечення.

ЕТ – есенціальний тремор.

FFT – (Fast Fourier Transform) швидке перетворення Фур'є.

Т-об'єкт – Тремор-об'єкт.

АНР – аномальні неврологічні рухи.

КГМ – кора головного мозку.

ЛПЗ – лікувально профілактичний заклад

НС – надзвичайна ситуація

## ЗМІСТ

Перелік умовних позначень, символів, скорочень і термінів.....	6
ВСТУП.....	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ ТА ТЕХНІЧНЕ ЗАВДАННЯ .....	10
1.1. Аналіз предметної області .....	10
1.2. Засоби та методи вирішення проблеми .....	11
2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ПРОГРАМНОГО ПРОДУКТУ .....	13
2.1. Розробка архітектури програмного продукту.....	13
2.2. Аналіз інструментальних засобів розробки .....	15
2.3 Пошук акторів та варіантів використання .....	15
2.4 Опис варіантів використання .....	18
2.5 Шаблон розробки застосунку .....	19
2.6 Абстрактний рівень системи .....	21
2.7 Проектування бази даних.....	24
2.8 Діаграма розгортання застосунку .....	26
3. РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	28
3.1 Вибір мови програмування та технологій розробки.....	28
3.2. Розробка функцій об'єктів проектованого програмного забезпечення .....	32
3.3. Розробка вбудованих процедур обробки даних .....	36
3.4. Ілюстрація роботи створеного програмного забезпечення .....	39
3.5 Тестування програмного забезпечення та оцінка якості .....	49
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ.....	51
4.1 Вимоги до виробничих приміщень для експлуатації ВДТ.....	51
4.2 Основні елементи підвищення стійкості функціонування лікувального закладу в умовах надзвичайних ситуацій.....	53
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	60
ДОДАТОК А – Публікація у науковому виданні .....	61
ДОДАТОК Б – Лістинг коду інформаційної системи .....	71
ДОДАТОК В – Диск із кваліфікаційною роботою бакалавра .....	77

## ВСТУП

Актуальність теми - застосування новітніх інформаційних технологій в медицині може істотно підвищити якість діагностування різних захворювань. Це досягається завдяки отриманню додаткової інформації про патологічні процеси. Одним з найпоширеніших порушень опорно-рухового механізму людини є тремор, який може мати різну етіологію та патогенез. Актуальною проблемою на сьогодні є своєчасне виявлення даної хвороби якою займаються фахівці нейро-спрямованої галузі медицини. Найпоширенішим проявом тремору є тремтіння рук. Якщо зсилатися на всесвітні організації по охороні здоров'я то в Україні зі 100 людей 4 будуть страждати тремором у віці до 40 років, та 6 зі 100 у віці від 65 років.

Мета роботи – створення, автоматизування та налагодження інформаційної системи реєстрації та зберігання сигналів зібраних з кори головного мозку за допомогою ЕЕГ.

Завдання полягає у наданні можливостей отримання даних сигналів ЕЕГ з кори головного мозку для подальшого їх аналізу завдяки математичним перетворенням котрі допоможуть отримати кодові нотації для подальшої їх інтерполяції на комп'ютері.

Об'єктом дослідження є хід проектування інформаційної системи реєстрації ЕЕГ когнітивних сигналів керування елементами опорно-рухового апарату пацієнта та їх зберігання

Предметом дослідження є можливість створення комплексної системи оцінювання стану пацієнтів з порушеннями функціонування опорно-рухового апарату через тремо, яка включатиме інтерактивний браузерний застосунок для збору даних сигналів ЕЕГ їх аналіз та репрезентація.

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ ТА ТЕХНІЧНЕ ЗАВДАННЯ

## 1.1. Аналіз предметної області

Одним із найпоширеніших захворювань опорно-рухового апарату являється тремор. Він проявляється у вигляді імпульсивних мимовільних м'язових скорочень, інколи у ритмічній формі, вони візуально проявляються як сіпання певних частин тіла. Найбільше страждають на часткову втрату координації кінцівки, обличчя та голосові зв'язки від тремору. Каталізатором тремору можуть бути, як і зовнішні подразники, наприклад морози, так і внутрішні або неврологічні [1].

Діагностування даного захворювання опорно-рухового механізму починається від ретельного оцінювання його проявів, подразників, характеристик та інших медичних показань. Поділяється тремор на два стани: акційний, тобто той що виникає у наслідок фізичної активності пацієнта, навіть самі базові котрі вимагають контролю та точності як налити воду чи написати слово. В свою чергу – спокою виникає в наслідок уже присутніх неврологічних порушень або ж ідіопатичних захворювань пацієнта, що змушують тіло перебувати у стані постійного стресу, який посилюється в стресових ситуаціях.

Тремор дії, або ж акційний, теж поділяється на два підвиди кінетичний який виникає в наслідок добровільного руху та той що виникає в наслідок утримування позиції в супротив протидійним силам, тобто – постральний. Причиною ж для обох є тремор есенції. Тремор фізіологічний – низько частотний тремор, який практично не можливо уловити неозброєним оком, проте неврологічні показники пацієнтів із даним захворювання не відхиляються від нормальних. Сам по собі тремор може існувати як частинка більш комплексного захворювання, як паркінсон або склероз, або існувати самостійним захворюванням [2].



Наразі наявна складність у ідентифікуванні захворювань що негативно впливають на опорну рухову систему людини серед яких присутній тремор, все це через відсутність адекватної методології діагностування та лікування. Дана недосконалість спричинена вакуумом у сфері новітнього медичного обладнання котре б могло удосконалити та полегшити роботу стосовно цих захворювань.

Запропонованою методикою, котра використовується лікарями та дослідниками протягом певного часу є зчитування когнітивних сигналів кори головного мозку під час активності та пасивності пацієнта за допомогою електроенцефалографа.

## 1.2. Засоби та методи вирішення проблеми

Після проведення аналізу даної предметної області у висновку було визначено що за допомогою електроенцефалограми можливо знайти аномальні когнітивні хвилі у певних частинах мозку для ідентифікування змін у ньому. Багаточисельні дослідження науковців вказують на те що пацієнти у котрих діагностовано есенціальний тремор при замірах ЕЕГ мали сильні коливання між частотами 4Гц та 12Гц. Пропорційно при більш видимих ознаках захворювання опорно-рухового механізму даною проблемою, збільшувалась і частота коливань у піддослідних.

Когнітивні сигнали кори головного мозку отримані завдяки електроенцефалограмі можна встановити як орієнтир на шляху до зменшення ступеня тремору або навіть зменшити частоту передачі сигналів поміж лаячними волокнами та клітинами Пуркін'є, використовуючи такі методи нейромодуляції як:

- Транскраніальна стимуляція постійним струмом
- Транскраніальна магнітна стимуляція

Електроенцефалографічні дослідження також допомагають визначити, оглянути та розрізнити між собою інші опорно-рухові захворювання апарату як наприклад міоклонія, яка являє собою короткочасні спазмуючі скорочення м'язів або груп м'язів яка в основному виникає під час стану спокою та сну, обох бистрої що триває від 10 до 20 хвилин та повільної, що настає практично одразу після засипання та її тривалість сягає від півтора години стадій а також в русі. Відмінності в основному проявляються в синхронних скороченнях м'язів та нерегулярному ритмі діаграми когнітивних сигналів. Також присутні і інші захворювання опорно-рухового механізму котрі ми можемо спостерігати та аналізувати за допомогою графіків сигналів електроенцефалограми.

## 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ПРОГРАМНОГО ПРОДУКТУ

### 2.1. Розробка архітектури програмного продукту

Для даного проекту було вибрано раціональний уніфікований процес розробки застосунку.

З плюсів обраної методології можна визначити наявність поетапності та ітеративності розробки ПЗ. В силу того, що кроки розробки є доволі маленьким ми в короткий проміжок часу після початку розробки можна спостерігати результати зробленої роботи, це дозволяє на ходу моделювати та підлаштовувати програму відповідно до висунутих вимог а також швидких змін.

RUP визначив чотири етапи, які мають проходити під час життєвого циклу проекту. Цей етап дозволяє представити процес на високому рівні, як проект у стилі «водоспад», але важливість процесу полягає в ітераціях розробки, які містяться в усіх етапах [3].

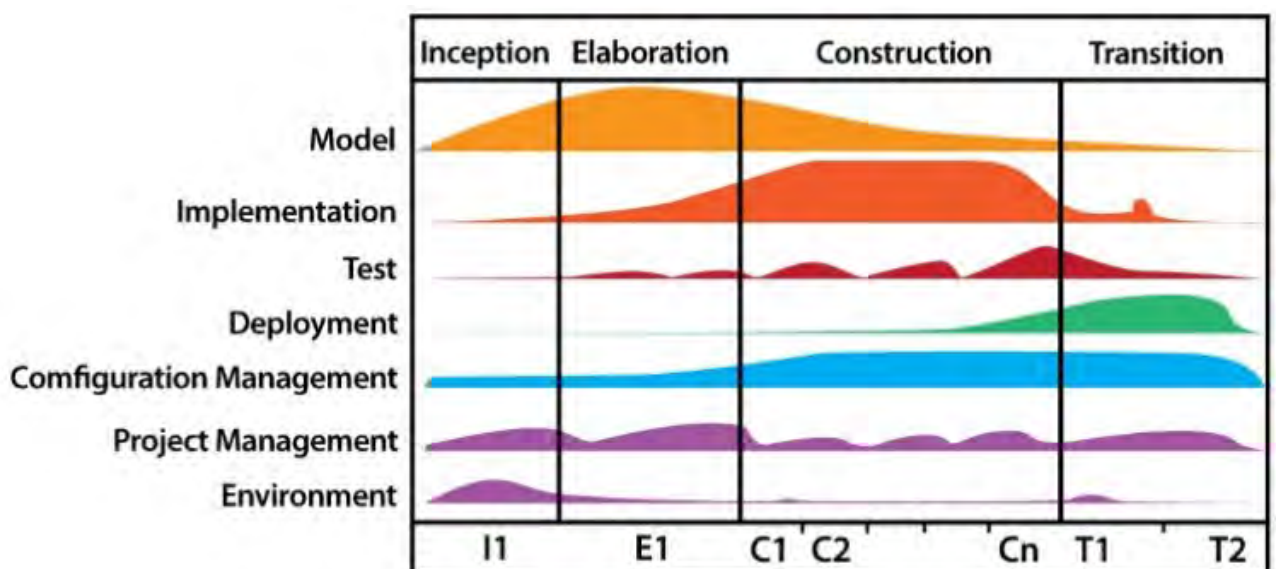


Рисунок 2.1 – Модель RUP

Задля кращого розуміння вище сказаного детальніше розглянемо всі 4 процеси обраного процесу розробки:

Початкова фаза – головною метою є належне обмеження системи, щоб можна було перевірити початкові витрати та бюджет. У цій фазі розробляється бізнес-логіка, яка складається з опису компанії, факторів успіху (таких як очікуваний дохід і визнання на ринку) і фінансової проєкції.

Фаза уточнення – головною метою є зменшення основних факторів ризику, які були виявлені на даний момент досліджень. Проєкт набуває форми під час стадії уточнення. В цій фазі проводиться вивчення області проблеми, архітектура проєкту набуває своєї основної форми.

Фаза конструювання - створення програмної системи є головною метою. Створення компонентів системи та інших функцій є основним акцентом цієї фази. Більшість програмування виконується на цьому етапі. У великих проєктах можуть бути встановлені кілька ітерацій будівництва, зусиллями для розбиття випадків використання на керовані фрагменти, щоб створити прототипи, які можна продемонструвати.

Фаза впровадження - головною метою є «перехід» системи з розробки до впровадження, щоб кінцевий користувач міг її використовувати та розуміти. Завдання цієї фази включають навчання кінцевих користувачів та адміністраторів системи її використанню, а також бета-тестування системи, щоб переконатися, що вона відповідає їх очікуванням. Кожен розробник, який не виконує відповідну роботу, замінюється або вилучається протягом оцінювальної фази системи. Порівнюється також з встановленим під час фази Заснування стандартом якості.

## 2.2. Аналіз інструментальних засобів розробки

Для розробки потрібно обрати зручне середовище програмування з необхідними засобами. Для мови програмування JavaScript доступні такі середовища, як VS Code, WebStorm та Netbeans, які практикуються у багатьох проектах. В даному випадку було вирішено скористатися середовищем програмування VS Code. Воно пропонує високу гнучкість налаштувань, можливості розширення та має доступний інтерфейс для користувача. NetBeans можна безкоштовно завантажити з офіційного сайту.

Вибір мови та середовища є ключовим для розробки додатку, що підтримує науковий експеримент. Він впливає на ефективність та швидкість розробки, а також на трудомісткість та затрати. Важливо враховувати побудовану та обґрунтовану архітектуру системи при виборі. Для нашої архітектури підходять мови, що підтримують ООП, такі як C#, JavaScript, Java, Python та C++. Java, Python та C# є найбільш поширеними для такого типу розробок, але наш вибір впав на мову програмування JavaScript з причини її універсальності та багатоплатформеності підтримки. До переваг вибору JavaScript можна віднести наявність багатьох інтернет ресурсів з прикладами та бібліотеками, а також її стандартизованість та доступність.

## 2.3 Пошук акторів та варіантів використання

Провівши ретельний аналіз предметної області було зіставлено ряд вимог котрим повинна підпорядковуватися система а також проблеми які вона повинна вирішувати. Засилаючись на дані питання можна виокремити та розподілити набори дій всіх акторів програми та їх варіанти використання.

Отримані результати представляють двох акторів:

- Незареєстрований користувач
- Авторизований користувач

Серед них лише авторизований користувач може являти собою головного актора. Адже незареєстрованому користувачеві не будуть надані привілегиї даної програми, він може бути лише піддослідним або спосетрігачем у даній процедурі. Проте жоден із перерахованих акторів не має права чи можливості провотиди наукові досліди чи брати участь у розробці ключової логіки системи, при відсутності відповідних кваліфікацій та сертифікацій. Нижче наведено діаграму варіантів використання під проведення діагностики (рис. 2.2).

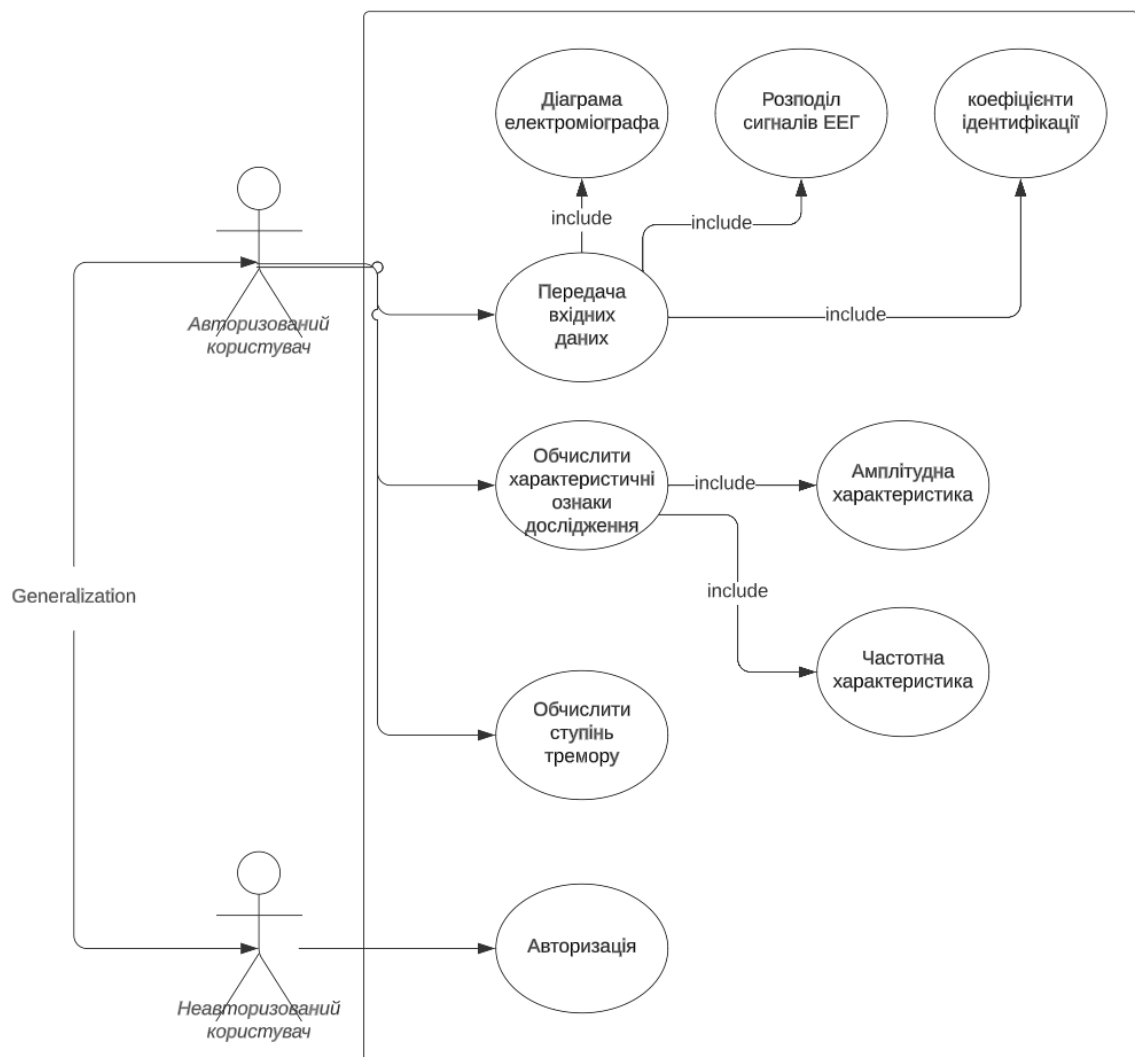


Рисунок 2.2 – Діаграма варіантів використання

Аби користувач міг дізнатися чи належить він до системи він проходить процедуру авторизації та аутентифікації, яка ґрунтується провадженням логіну та паролю. За невдалої спроби або ухилення від спроби проходження даної процедури користувач знаходиться поза межами системи тобто неавторизований користувачем. Єдиним варіантом дій даного актора є авторизація. Напротив авторизований користувач є явним актантом даної систем якому представлено безліч можливостей, серед котрих є і лог-аут тобто добровільне перестосування в неавторизованого користувача. Ось набір дій представлений авторизованому користувачеві:

- Лог-аут
- Обчислення ступення тремору
- Амплітудна характеристика
- Частотна характеристика
- Розгортка спіралей по осям
- Вирахування характерних ознак досліджень
- Введення вхідних параметрів

З наведеного вище можна зробити логічний висновок, всі можливі ключові функції здатен використовували лише один актор. Даний застосунок планується до застосування в науково-дослідницьких лабораторіях. Програма буде реалізована як веб-застосунок для отримання результатів розрахунків на основі прогнозних моделей, а деякі з її модулів будуть реалізовані як бібліотеки, відповідно до вимог. Застосунок націлений на вузькоспеціалізований сегмент всієї аудиторії, тому користувачів не повинно бути багато, оскільки система розрахована на багатокористувацьке використання, що вимагає розробки існуючих архітектурних рішень, таких як:

- Авторизація
- Хешування
- Зберігання вхідних та вихідних даних

## 2.4 Опис варіантів використання

Задля кращого розуміння представленої раніше діаграми варіантів використання (рис. 2.2), буде ретельніше розглянуто ключові ВВ:

- Передача вхідних даних – тут користувач надсилає на сервер всі необхідні дані над котрими відбуваються розрахунки та аналізи. Самі ж дані представлені у вигляді файлів котрі містять у собі ненормалізовані числові значення отримані від електроенцефалограми. Після надання файлів користувач мусить задати роздільну здатність та порт виведення сигналів КГМ котрі повинні відображатися на діаграмах.
- Авторизація – даний варіант використання дозволяє змінити стан актора з «неавторизованого користувача» на «авторизованого користувача». Процес виглядає наступним чином, користувач вводить у відповідні поля логін та пароль та при внесенні їх у коректному форматі надсилає дані на сервер де проводиться їх перевірка на наявність та правильність. У випадку вдалої авторизації користувач переадресовується на головну сторінку застосунку.
- Обчислити ступінь тремору – використовується згодом при отриманні проміжних вичислень дає можливість обчислити значення для визначення неврологічних показників за допомогою математичних методів перетворення та статистично вибраних змінних.



## 2.5 Шаблон розробки застосунку

На сьогодні ключовою нішею в житті людей став інтернет, що сприяє виникненню інформаційних вакуумів серед веб-застосунків через нові потреби котрі виникають у користувачів мережі інтернет.

Веб-застосунки представляються у вигляді клієнт-серверних програм, роль клієнта виконується веб-браузером. Клієнт надсилає запити на сервер, котрий у свою чергу обробляє їх та з часом відправляє результат назад клієнту. Схему їхньої роботи можна побачити на рисунку нижче (рис. 2.3).

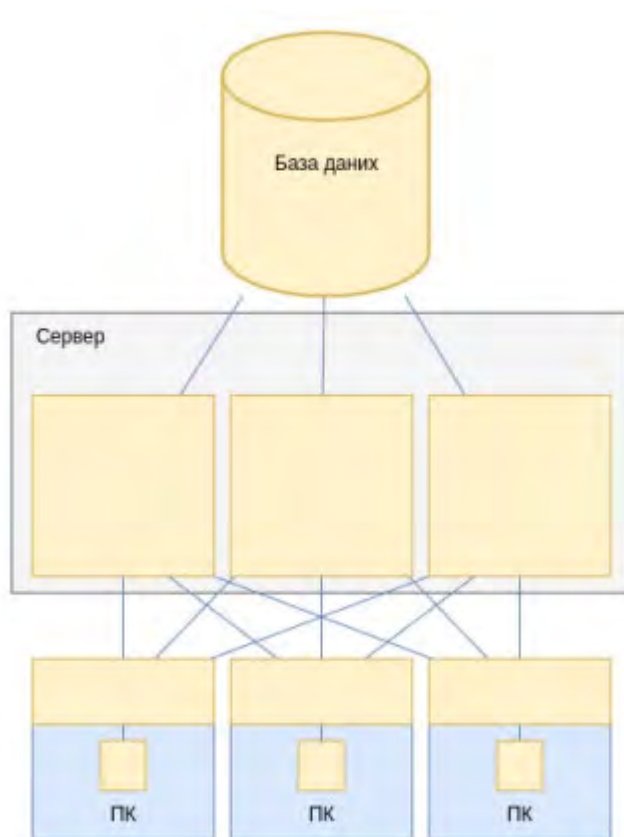


Рисунок 2.3 – Схема роботи веб-програми

Серед плюсів використання веб-браузеру у ролі клієнта можна навести такі:

1. Універсальність – браузери зустрічаються у будь-якому типі сучасної техніки будь то планшет, телефон чи персональний комп'ютер. Також в сутність того що

технології розвиваються веб-браузери можна зустріти також на побутовій техніці як холодильники. Тому відпадає необхідність до окремої програми клієнта котру потрібно інстальовувати так як є ймовірність того що вона на просто не буде підтримуватись операційною системою певного застосунку

2. Комплексність – браузері здатні забезпечити додатковий функціонал на котрий вони не були розраховані за допомогою складних клієнтів, аплетів.

3. Безпека – браузері здатні забезпечити безпеку при взаємодії з інтернетом.

4. Багаторівневність - веб-застосунки виходять за рамки парадигми клієнт-сервер. Веб-програми не обмежуються взаємодією між браузерами та серверами. Коли вони взаємодіють з іншими внутрішніми серверами, такими як бази даних і застарілі системи, веб-сервери виступають як «клієнти». Таким чином, більшість веб-додатків мають багаторівневу архітектуру. Багаторівневі програми, як випливає з назви, складаються з кількох рівнів, кожен з яких може виконувати ролі як клієнта, так і сервера під час обміну даними зі своїми сусідами. З іншого боку, простіші клієнт-серверні програми складаються з лише двох рівнів, у яких виконуються різні функції, такі як клієнт і сервер. Кожен рівень представлений у вигляді прикладного рівня. Прикладні рівні включають браузер, веб-сервер, сервер прикладних програм, базу даних/застарілу систему та інші компоненти. Веб-клієнт нижнього рівня (зазвичай браузер або інтелектуальний агент) починає обробку, надсилаючи запит на веб-сервер. Кожен набір суміжних рівнів містить пару серверів і клієнтів, і залежно від того, з ким із сусідів він взаємодіє, кожен проміжний рівень може діяти як сервер або клієнт. Наприклад, у той самий спосіб, як браузер підключається до веб-сервера для виконання запиту, частина програми, що виконується на веб-сервері, може підключитися до рівня бізнес-логіки або моделі даних і виступати як клієнт для цих служб.

Отже розглянувши вище сказані переваги було обрано даний принцип проектування.

## 2.6 Абстрактний рівень системи

Обраною парадигмою програмування було обрано структурну а саме об'єктно-орієнтовану. Серед переваг даного підходу можна визначити її модульність, яка забезпечується завдяки основним концепціям ООП: інкапсуляції, успадкуванню, поліморфізму та абстракції.

Дана парадигма наразі є однією з найбільш поширених, адже вона допомагає вибудувати міцний каркас майбутнього додатку. Нижче буде перераховано ключові властивості ООП через які і було її обрано:

1. Приватність – контролювання звернень до полів та методів класу за допомогою інтерфейсів. Це забезпечує зниження ризиків непередбачуваних мутацій змінних та функцій. Також приватні елементи не можуть бути використані поза межами класу це дозволяє не хвилюватись що до можливих незапланованих змін роботи логіки пов'язаних методів чи класів під час маніпуляцій над ними, адже вони не мають до них доступу.

2. Лаконічність – об'єкти завжди представлені мінімально можливим набором полів та методів. Завдяки цьому економиться значна кількість ресурсів машини.

3. Мультизадачність – цей параметр теж дозволяє зекономити ресурс пам'яті на машині так як ми маємо можливість перекоористати існуючий програмний код у новому функціоналі з мінімальними його модифікаціями за потреби.

Задля зображення у візуальній формі обраного підходу буде використано UML діаграми.

UML – це мова візуального моделювання загального призначення, яка використовується для визначення, візуалізації, конструювання та адміністрування артефактів програмної системи [4]. Своє використання дана мова знаходить у написанні та проектуванні інформації про цю систему. В ній містяться різний помічний функціонал як семантику, вказівки та нотації. UML є майже не замінною

в проектуванні ООП систем, адже завдяки ній можна описати всі класові зв'язки та функціонали, цикли життя та інші види діаграм.

Першою та основною діаграмою побудованою на UML є діаграма класів. Вона містить взаємозв'язки класів та їх відносини. Вона малюється в абстрактний спосіб. На рисунку нижче зображено базова архітектура веб-застосунку (рис. 2.4).

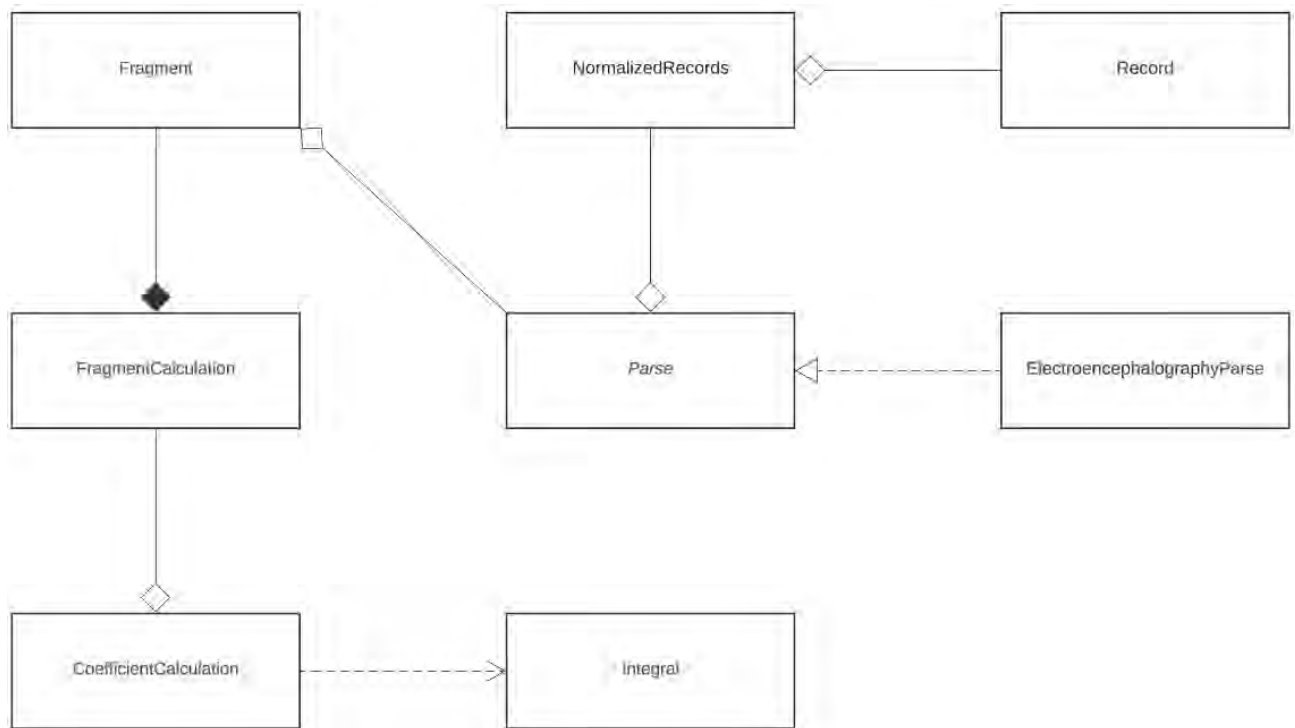


Рисунок 2.4 – Діаграма класів застосунку

На даній вище діаграмі продемонстровано всі ключові класи програми. Інтерфейс `Parse` являє собою базовий клас через котрий проходять всі вхідні дані. За допомогою нього представлено фундаментальні види взаємодії з отримуваними даними. З даного інтерфейсі всі похідні класи наслідують один метод за для нормалізації даних та фільтрації. Задля зняття надмірного навантаження з даного методу імплементовано допоміжні методи.

В середині системи представлені такий клас як `ElectroencephalographyParse`. Він реалізовує метод `NormaliseInput` та інші допоміжні методи що сприяють кращі нормалізації вхідних даних та читабельності коду. Даний клас можна назвати підготовчим. Вони переформатовують дані у більш зручний вид для проведення

подальших робіт з ними. Написаний `ElectroencephalographyParse` клас за допомогою Singleton патерну, який надає кожній змінній лише один екземпляр представлення, тобто імплементовані класи будуть глобальними що надасть зручний і швидкий доступ до змінних.

`NormalisedRecords` масштабує та нормалізує вхідні дані. Створюється певний екземпляр класу в котрому містяться роздільна здатність масштабування та екстрималі поля серед усіх значень.

`FragmentCalculation` та `Integral` класи реалізують математичні алгоритми. Вони реалізують допоміжний функціонал у перетвореннях та обчисленнях отриманих даних, створюючи власні екземпляри та отримуючи атрибути що придатні для розробки алгоритмів.

`CoefficientCalculation` реалізує методи математичних алгоритмів задля оцінювання інтенсивності сигналів КГМ. Він знаходить своє використання в середині двох попередньо описаних класів `FragmentCalculation` та `Integral`.

Також у даній системі реалізовано систему авторизації яка дає користувачу змогу користуватися основним функціоналом даної програми. На рисунку нижче зображено діаграму класів процесу авторизації (рис. 2.5).

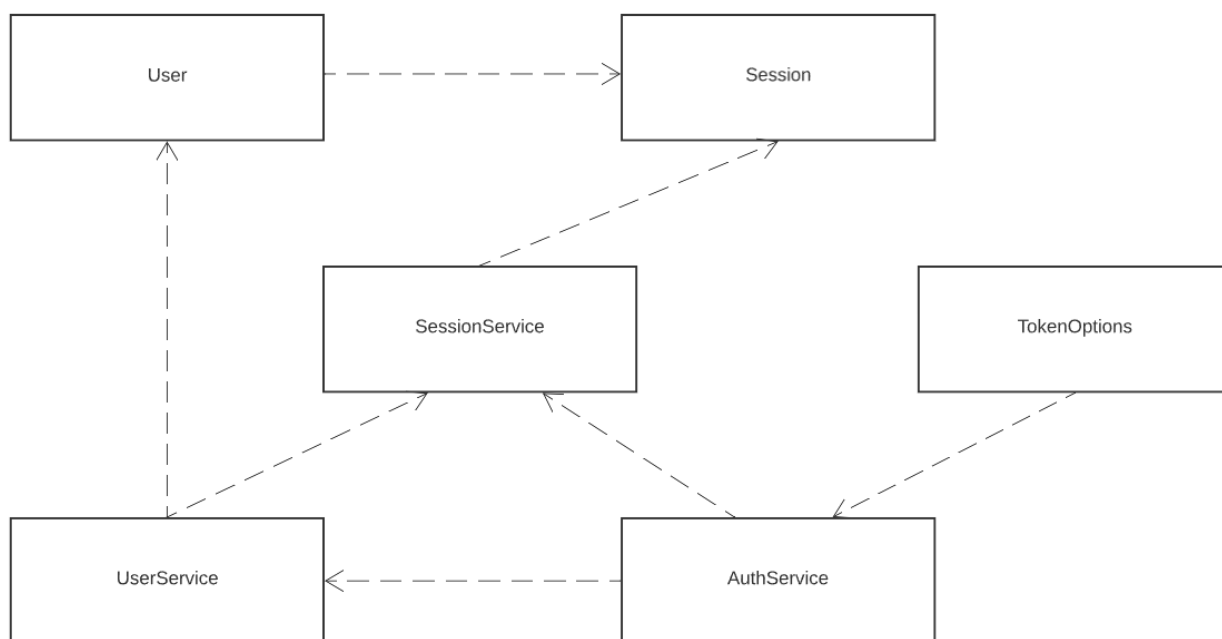


Рисунок 2.5 – Діаграма класів авторизації

Задля кращого розуміння наведеної вище діаграми розібрано відповідальності лише ключових класів.

User несе відповідальність за базове представлення користувача в системі. Він передбачає створення нових екземплярів користувача то отримання інформації про нього. Це забезпечує вищий рівень безпеки при взаємодії з БД.

UserService перехідний рівень для підвищення безпеки. Він розподіляє доступи серед користувачів. Дозволяє створення редагування чи видалення даних користувача.

Session створюється коли користувач входить в систему. Під час поточної сесії отримує набір user значень, при завершенні сеансу дані анулюються. В зворотному випадку чинності сесії та її валідності, Session буде пере використано.

AuthService використовується при обох процесах авторизації та автентифікації. В його розпорядженні два інших сервісних класи SessionService та UserService для логіну, сайн-іну та лог-ауту з системи. Надає можливість зміни облікових даних адже не поодинокі випадки втрачання або забування паролі, попри поширеність менеджерів особистих даних та паролів.

## 2.7 Проектування бази даних

База даних необхідна за для зберігання користувацьких даних та даних ЕЕГ. Для їх зберігання було обрано документо-орієнтовану систему. Вибір даної БД зумовлений її швидкодією, масштабільністю та масштабованістю. Перевагою над реляційними БД є уникнення дорогіших викликів операції з'єднання завдяки зберігання інформації в межах одного документу.

Після проведення аналізу вимог інформаційної системи було окреслено діаграму баз даних наведену в приклад нижче (рис. 2.6).

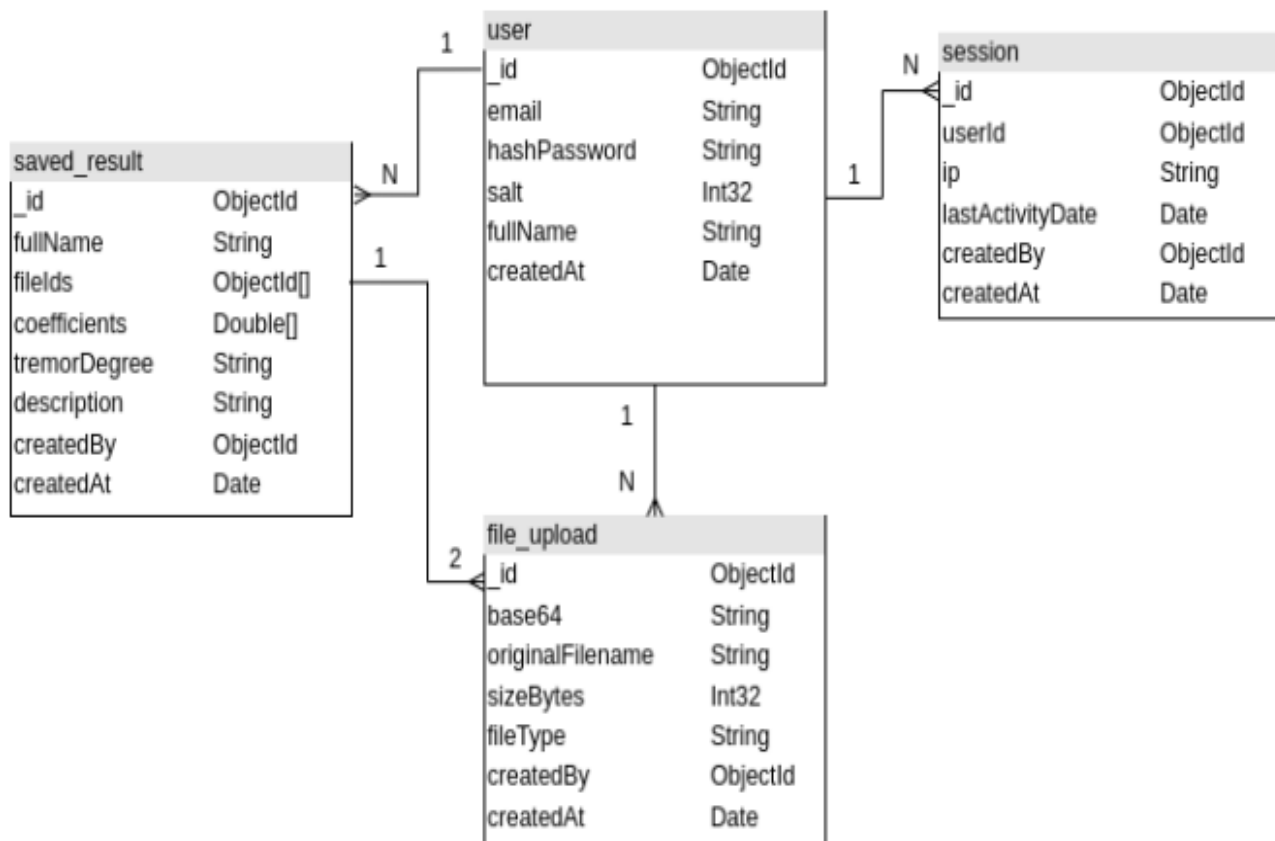


Рисунок 2.6 – Діаграма баз даних

На цій діаграмі зображено колекції або ж сутності та їхні взаємозв'язки на котрих виведено відношення поміж ними. Розбір відношень сутностей такий: сутність `user` містить 3 унікальних відношення у форматі один до багатьох. Перше з них до сутності `session` що може мати декілька унікальних сесій користувачів але на одного користувача одна сесія. Другий зв'язок з сутністю `saved_result`, де одик користувач може мати декілька результатів, але результат може мати лише одного користувача. Та третій зв'язок із сутністю `file_upload` із ідентичним принципом роботи до попередніх, один користувач здатен здійснити кідькісні завантаження, проте кожен файл містить лише одного користувача.

Сутність `saved_result` містить відношення багатьох до багатьох з сутністю `file_upload`. Результат містить два завантажених файли однак файл належить лише одному результату.

## 2.8 Діаграма розгортання застосунку

У даній діаграмі зображено та описано процес доставки ПЗ та її взаємодії із апаратною системою під час виконання власного функціоналу. Приклад даної діаграмо зображено на рисунку нижче (рис. 2.7).

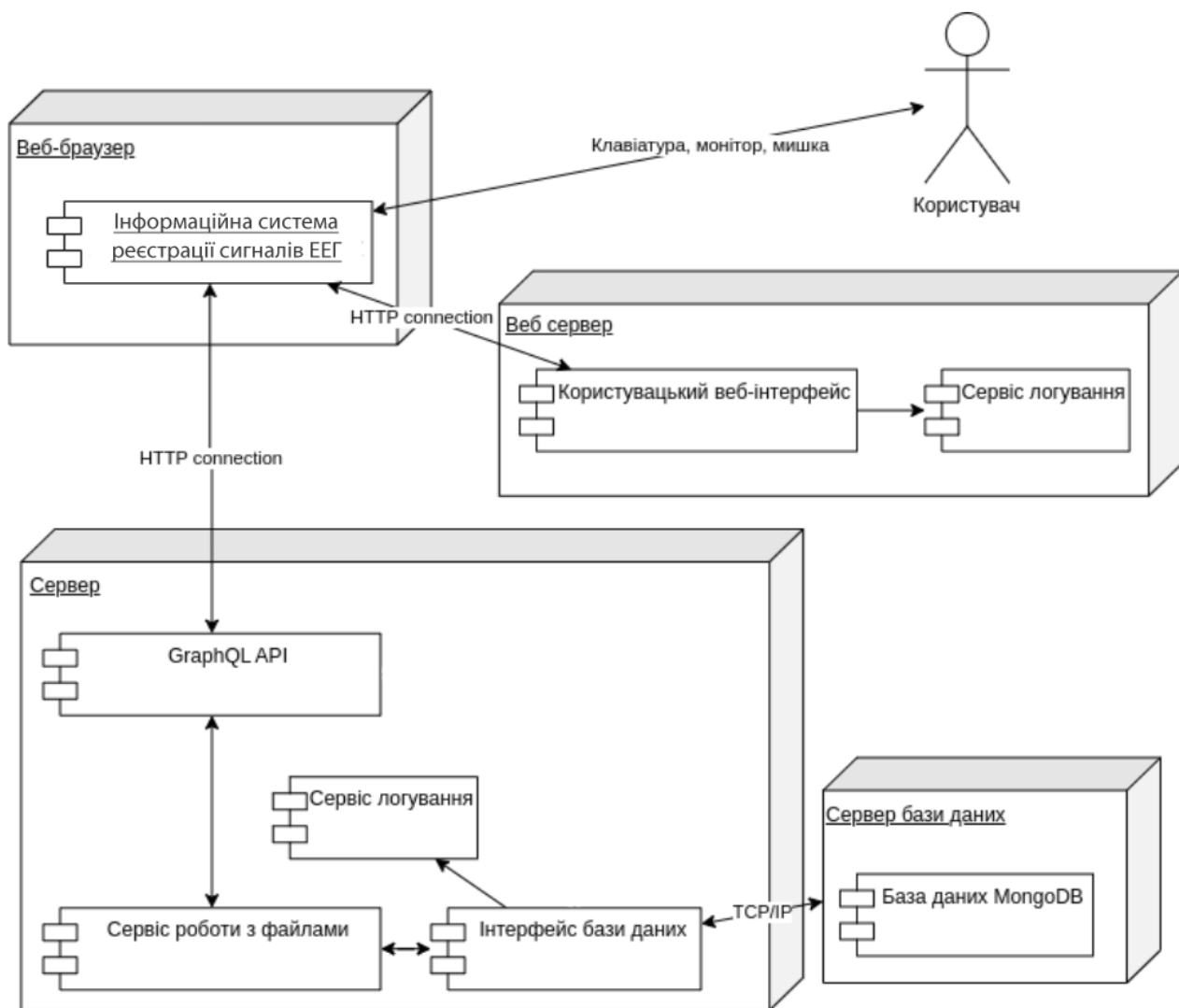


Рисунок 2.7 – Діаграма розгортання системи

На рисунку вище зображена багато-вузлева система робота у котрій відбувається наступним чином:



1. Користувач заходить у веб-браузер на ПК з використанням периферійних пристроїв як клавіатура, монітор та миша.
2. Браузер надсилає запит до веб-сервера у відповідь отримуючи скомпільований код розмітки та програми.
3. Веб-сервер логує та зберігає дані під час прогонки пересланого коду.
4. При рендерингу контенту браузер надсилає запити до сервера для отримання необхідних даних. Система робить запити при кожній зміні інтерфейсу котра вимигає авторизації користувача. Спілкування із сервером відбувається за допомогою GraphQL. Він використовує внутрішні сервісні вузли для виконання операцій. Серед вузлів є авторизація, логування та роботи з файлами які при необхідності можуть звертатися до інтерфейсу баз даних.
5. Інтерфейс баз даних керує запитами що надсилаються до самих власне баз даних щоб їх змінити отримати додати або видалити.

### 3. РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

#### 3.1 Вибір мови програмування та технологій розробки

Під час вибору інструментів для реалізації програми, яка забезпечує оцінку ступеня тремору, важливо звернути увагу на існуючі бібліотеки, які дозволяють ефективно та точно обробляти та працювати з вхідними параметрами. Зважаючи на те що під час фази проектування перевагу було надано веб-застосунку, то для початку необхідно визначити, які інструменти будуть використані для реалізації рівня клієнта. Клієнт являє собою застосунок з мульти платформовою підтримкою оскільки є веб сторінкою. Для його реалізації використовувався ряд мов програмування, стилів та розміток. Базовий рівень складається із мови HTML для задання каркасу застосунку поверх котрого було накладено стилі за допомогою CSS.

При проектуванні програмного забезпечення для зчитування сигналів ЕЕГ було використано об'єктно-орієнтований підхід та мову програмування JavaScript. Для правильної реалізації системи було проведено аналіз основних класів та сутностей, використовуючи обрані рішення та постановки задачі. Розроблення виконувалося шляхом реверс-інжинірингу, що передбачає створення ключових елементів та їх взаємодій, а потім - побудову UML-діаграми класів. Такий підхід було обрано через специфіку предметної області та необхідність системних доопрацювань, а також через складність оцінити весь масштаб та структуру системи на ранніх стадіях розробки.

Основним рушієм сторінки став JavaScript, а саме фреймворк React.js, що дозволило зекономити велику кількість людино-годин для розробки даного проекту. Даний фреймворк дозволяє створення багатосторінкових та що саме основне одно сторінкових або SPA сайтів, та спрощеними маніпуляціями DOM деревом застосунку. Також однією із причин вибору даної мови програмування стала можливість написання програм на об'єктно орієнтованій архітектурі.

React відображає компоненти у їхніх життєвих циклах. Також реакт надає дві варіації написання коду: класову та функціональну. Класова система написання коду є більш знайома для програмістів котрі привикли писати програми за допомогою таких мов програмування як C, C++, Java і тд. Проте вона є застарілою у новіших версіях реакту та більше не підтримується її розробниками, тому вибір скоротився лише до 1 варіанту, а саме функціонального підходу, хоч він здається не об'єктно орієнтованим насправді це не так. Плюси даної системи є наявність великої кількості синтаксичного цукру котрий в разі полегшує написання коду багато компонентних застосунків.

Слід розглянути технологію, яка забезпечить високу продуктивність, швидку роботу з великими масивами даних і можливості реалізації об'єктно-орієнтованого підходу програмування, оскільки вся логіка розрахунків оцінки ступеня тремору буде виконуватися на сервері. Таким чином, на серверній частині програми буде використано Node.js. Node.js — це безкоштовне серверне середовище що генерує динамічний контент сторінок та інші базові функції, що підтримується безліччю платформ, саме ж використовує мову JS, двигуном котрої є V8 розроблений компанією Chrome [5]. Зі сторони сервера виконується перебазування виконання коду JS до серверу, надаючи змогу практично всім операційним системам підтримувати даний фреймворк. Код у ньому копіюється на льоту. Однією із особливостей Node.js є можливість підтримки інших вузькоспеціалізованих фреймворків як Коа, Socket, Nest та безліч інших. Node.js здатен до імплементації ПЗ за допомогою об'єктно орієнтованої парадигми оскільки мовою до його написання є JavaScript.

З переваг даного фреймворку можна перелічити:

- Поліморфізм
- Швидкодія
- Можливість реалізації об'єктно орієнтованої парадигми написання коду
- Приватність

- Гнучкість
- Кросплатформеність
- Ресурсо-бережне кешування
- Технології стеку є надійними
- Продуктивність
- Відсутність потреби у кількох мовах при реалізації клієнт-серверних застосунків

Гнучкість є основною ефективною перевагою Node.js, оскільки при необхідності внесення правок у логіку роботи певного вузла, відсутня необхідність до корегування суміжних або залежних компонентів програмного забезпечення, буде достатньою лише заміна даної ноди або ж вузла. Також Node.js надає АПІ котрі дають змогу напряду взаємодіяти з функціоналом двигуна V8, модулі управління пов'язані з операційною системою та можливості написання інтернаціоналізованих програм. Що до безпеки – вона забезпечується серверними протоколами які обмежують доступ до конфіденційних даних.

Задля забезпечення документо-орієнтованої бази даних буде використовуватися NOSQL застосунок MongoDB [14]. Вона зберігає дані у гнучкому JSON-подібному форматі, тому поля документу можуть різнитись та змінюватись з часом. Документна модель веде до об'єктів в коді програми. MongoDB в силу своєї розподіленості, легко масштабується по горизонталі. Однією із ключевих характеристик можна виділити ізоляцію робочого навантаження, завдяки цьому аналітичні процеси над локальними даними відбуваються на вториних вузлах. Користувач має доступ до витратоефективних правил архівування в міру того як розростається його база даних. Доступ до самої ж БД здатен відбутися з будь-якої мови з допомогою терміналу. Сама ж база даних здатна буди у вигляді безсерверної технології для економії ресурсів.

Керований сервіс Atlas API даних від MongoDB гарантує захищену роботу із збереженими даними завдяки стандартним протоколам HTTPS. На відміну від прямого з'єднання із базою даних, сервіс надає повністю керований сервіс

проміжного програмного забезпечення що знаходиться поміж клієнтом і його кластером. Atlas API від MongoDB надає доступ до БД з будь-якої платформи котра підтримує протокол HTTPS:

- Веб-браузер
- Веб-сервер
- Конвеєри CI/CD
- Безсерверні та кордонні середовища опчислення
- Мобільні аплікації
- Гаджети з простору інтернету речей

Для кожного API запиту аутентифікується адресант, включно із валідацією токена доступу, вхід в систему за допомогою з заголовками облікових даних або ж на пряму привласнюючи унікальний ран-тайм користувачеві опираючись на обрану конфігурацію. Далі авторизується запит. Це забезпечує валідність надісланих користувачем форм і надання дозволів до виконання операцій базуючись на схемі авторизації кінцевої точки. Подальшим кроком буде виконання запрошеної операції, задіяючи зчитування або записування даних в Atlas з згенерованою кінцевою точкою або викликавши власний функціонал. Фіналом даного процесу є повернення HTTPS відповіді адресанту. Відповідь включає результати згенерованої кінцевої операції або ж довільні дані. Формат відповіді може бути як у стилі JSON так і власний формат від MongoDB – EJSON.

Також важливу роль у написанні застосунку відіграє обрання зручного середовища розробки. Оскільки в минулі роки відбувався бум розвитку серед інтегрованих середовищ розробок від безлічі компаній, вибір конкретного в першу чергу тепер зводиться виключно до особистих преференцій користувача одними із найпопулярніших редакторів коду наразі є JetBrains WebStorm, IntelliJ IDEA та Visual Studio Code. В силу обраної мови програмування JavaScript та попередніх навичок роботи у середовищах Visual Studio та VS Code, вибір зупинився на Visual Studio Code. На відміну від вище перелічених інтегрованих середовищ розробок (IDE) та багатьох інших, VS Code здатен підтримувати безліч мов програмування.

Також дане IDE є модальним, тобто найкраще застосунок відкриває себе у зв'язці із розширеннями та аддонами котрі полегшую роботу програмісту під час написання коду, надаючи йому заготовлені наперед сніппети та можливість їх створення, покращених підсвіток тексту а також безліч інших корисних функцій [7].

Серед переваг даного продукту можна теж винести IntelliSense – загальний термін що поозначає різноманітні функції редактора коду як доповнення коду, додаткова інформація про параметр та список учасників. Під час написання JS коду це економить величезну кількість сил та людино-годин. Є можливість його конфігурування за допомогою `jsconfig`у` або ж підсилити ефект з допомогою JSDoc котрий суворіше визначає вхідні та вихідні типи параметрів.

При роботі з React працює покращена підсвітка синтаксису. А для дебагінгу можна використовувати встроїні JS дебагери. Їх теж можна конфігурувати під власні потреби. За допомогою брейкпоінтів буде виконуватись построчне виконання коду що дає змогу краще зрозуміти в якій послідовності виконуються методи та в разі помилки точно встановити її та локалізувати.

Перелічині вище технології, мови, IDE і решта ретельно підібрані під вимоги поставленої задачі та архітектури, з врахуванням власних потребностей. Це дозволить цілком відворити запланований застосунок для приємного користування користувачем.

### 3.2. Розробка функцій об'єктів проектованого програмного забезпечення

У програмі реалізовані методи різної складності котрі зчитують данні, займаються їх перетвореннями та інтерпретаціями.

Також для системи репрезентації даних сигналів ЕЕГ які поступають від КГМ було спроектовано архітектуру браузерного застосунку котрий перетворює набуті діаграми у кодовий варіант.

Перш за все необхідно створити метод реінтерпритації сигналів ЕЕГ котра проставляє часові відмітки з зачитаними та перетвореними даними. На рисунку нижче зображено лістинг коду що до даного завдання (рис. 3.1).

```
const data: any = useMemo(() => {
  if (!electromyographRecords?.length) return undefined;
  const startTime = produceTime(electromyographRecords[0].time);
  return electromyographRecords.map((record) => ({
    ...record,
    time: new Date(),
    absoluteTime: +(produceTime(record.time) - startTime).toFixed(3),
  }));
}, [electromyographRecords]);
```

Рисунок 3.1 – Програмний код даних для ЕЕГ

Для наглядності даних використовуються стороння бібліотека котра спеціалізується на відображенні діаграм та чартів різних видів. На рисунку нижче зображено лістинг коду що до даного завдання (рис. 3.2).

```
<Chart
  name="electromyographRecords"
  options={electromyographOptions}
  datasets={[
    {
      label: t(`calculation.series.electromyograph`),
      backgroundColor: 'blue',
      data,
      type: 'line',
      pointRadius: 0,
      borderColor: 'blue',
      borderWidth: 0.1,
    },
  ]}
/>
```

Рисунок 3.2 – Програмний код відображення діаграм ЕЕГ

Наступною стадією є створення та налаштування сервера. Саме він буде відповідальним за реєстрацію даних. Задля забезпечення коректності результатів в продовж цілої роботи сервера на ньому будуть виконуватися чисельні перевірки різних типів.

Перш за все необхідно реалізувати завантаження вмісту файлів з даними енцефалограм та їх подальший парсинг. Тому принцип роботи даного функціоналу такий: спочатку відбувається зчитування даних файлу, при успішному результаті даного методу відбувається парсинг. Після цього вони поміщаються в середину класу. Та в кінці якщо відсутності будь яких помилок або збоїв програми дані заносяться в БД. На рисунку нижче зображено лістинг коду що до даного завдання (рис. 3.3).

```

async normalizeData(base64: string, signalPort: ElectromyographPort):
Promise<ElectromyographRecord[], Date> {
  const ssv = base64ToString(base64);
  const { data, errors }: { data: unknown[][]; errors: unknown[] } = Papa.parse(ssv,
{ delimiter: '\t' });

  const headers = data.filter((row) => row.length === 1).flat() as string[];
  const startDate = getStartDate(headers);
  const signalPortIndex = ElectromyographPortIndex[signalPort as keyof typeof
ElectromyographPortIndex];

  if (!signalPortIndex) throw new Error(BackendErrors.UNAVAILABLE_SIGNAL_PORT);
  if (errors.length) throw new Error(BackendErrors.TABLET_FILE_INCORRECT_FORMAT);

  const records = data
    .map((row) => {
      const time = this.parseTime(row[0] as string, startDate);
      const signal = +(row[signalPortIndex] as number);
      if (Number.isFinite(time.getTime()) && Number.isFinite(signal)) {
        const record: ElectromyographRecord = {
          time,
          relativeTime: startDate.getTime() - time.getTime(),
          signal,
        };
        return record;
      }
    })
    .filter(Boolean) as ElectromyographRecord[];

  return [records, startDate];
}

```

Рисунок 3.3 – Програмний код запису та парсингу даних ЕЕГ



Результати діагностування з електроенцефалографа можна спостерігати на рисунку наведеному вище. Задля дотримання принципів парадигми програмування SOLID та зведення надлишкового навантаження функцій одне на одну до нуля, парсування файлу було винесено в межі власної функції. Реалізація необхідного функціоналу представлена на наступному зображенні лістингу (рис. 3.4).

```
function parseData(data, startDate, signalPortIndex) {
  const records = data
    .map((row) => {
      const time = this.parseTime(row[0] as string, startDate);
      const signal = +(row[signalPortIndex] as number);
      if (Number.isFinite(time.getTime()) && Number.isFinite(signal)) {
        const record: ElectromyographRecord = {
          time,
          relativeTime: startDate.getTime() - time.getTime(),
          signal,
        };
        return record;
      }
      return null;
    })
    .filter(Boolean) as ElectromyographRecord[];
  return records;
}
```

Рисунок 3.4 – Програмний код парсингу завантажених даних ЕЕГ

Наступним кроком буде нормалізація отриманих даних. Тобто їхнє приведення до діапазону від -1 до 1. Дана логіка закладена у відповідну функцію, що дозволяє спросити дані для подальшої роботи та розуміння їх. На рисунку нижче зображено лістинг коду що до даного завдання (рис. 3.5).

```

constructor(data: T[], name: keyof T) {
  const maxValue = Math.max(...data.map((row) => Math.abs(row[name] as
number)));
  this.maxValue = maxValue;
  this.scaledRecords = data.map((row) => ((row[name] as number) /
maxValue) as unknown as typeof row);
}

```

Рисунок 3.5 – Програмний код нормалізації значень

Ключову роль в даній програмі відіграє метод, котрий здійснює обрахунки коефіцієнтів. Даному методу на вхід приходять нормалізовані дані з діаграми ЕЕГ, після чого над ними відбуваються маніпуляції. По завершенню дії методу він повертає перетворені дані.

### 3.3. Розробка вбудованих процедур обробки даних

Для візуалізації вхідних даних перш за все необхідно створити методи їх перетворення що допоможуть у виявленні аномальностей при перевірці опорно-рухового апарату пацієнта під час проходження трас на визначення аномальних неврологічних рухів. Під час даного тесту зчитуються когнітивні сигнали МГМ.

Задля зв'язування когнітивних сигналів ЕЕГ із рухами пацієнта на АНР трасі використовуються формули  $j = \overline{1, n_1 + 1}$  що визначають мілкіші сегменти траси та їх активності. Відстань траси може різнитись в залежності від обраної роздільної здатності. Апроксимувавши отримані дані ми отримаємо взаємо-зворотні впливи зв'язків одне на одну [8, 9]. На рисунку нижче наведено очний приклад співвідношень даних зв'язків одне між одним (рис. 3.6).

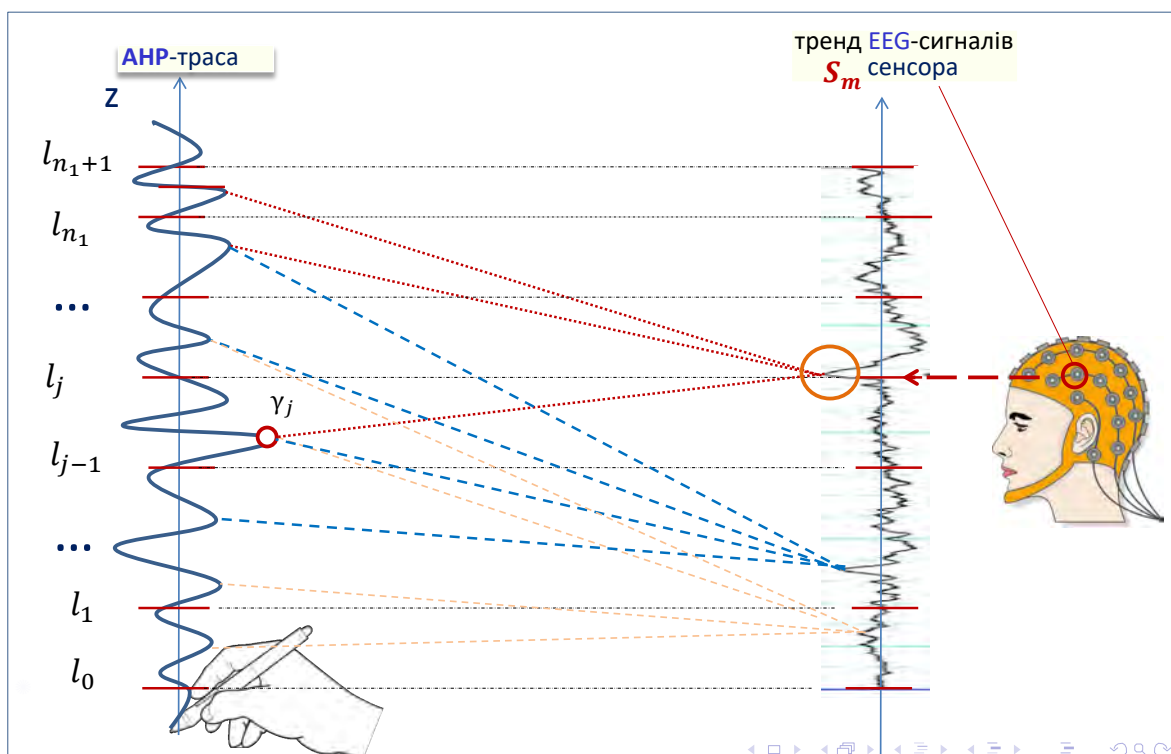


Рисунок 3.6 Зв'язки зворотних впливів АНР-траси та когнітивних сигналів ЕЕГ

Вичисливши необхідні початкові дані підставляємо їх у змінні функції визначення коливальних компонентів з знаходженням швидкості розповсюдження хвилі у фазі,  $\beta_k^2$ , де  $k = \overline{(1, n_1 + 1)}$ . Розв'язавши дану функцію отримуємо систему крайових значень у котрих вхідні вимоги однорідні для кожної точки траси [10, 11]:

$$\frac{\partial^2}{\partial t^2} u_k(t, z) = b_k^2 \frac{\partial^2}{\partial z^2} u_k + S_k^*(t, z) \quad (1)$$

далі:

$$u_k(t, z)|_{t=0} = 0, \quad \frac{\partial u_k}{\partial t}|_{t=0} = 0, \quad k = \overline{1, n_1 + 1} \quad (2)$$

Визначаємо крайні умови:

$$u_{k-1}(t, z)|_{z=l_{k-1}} = U_{L_{k-1}}, \quad u_k(t, z)|_{z=l_k} = U_{l_k}, \quad k = \overline{1, n_1 + 1} \quad (3)$$

Виводимо незв'язну функцію крайовій задачі, у котрій невизначеними функціями в залежності часу являються коливальні компоненти інтенсивності

розповсюдження фазових коливань  $b$ ,  $k = \overline{1, n_1 + 1}$ , в той час як точки обсервації на фракції траси  $\gamma_k \subset \Omega_k$ ,  $k = \overline{1, n_1 + 1}$ ,  $u_k(t, z)|_{\gamma_k} = U_{l_k}(t, z)|_{\gamma_k}$  визначені. (4)

Вирішення стартової крайової задачі передбачає визначення перехідної точки для кожного  $k$ -го елемента на трасі аномального неврологічного руху, та записується таким чином  $b_k \in D$ , де  $D = \left\{ v(t, z) : v|_{\Omega_{k_1 T}} \in C(\Omega_{k_1 T}), v > 0, k = \overline{1, n_1 + 1} \right\}$ .

Наступний запис описує незв'язність та відвітвленість від початкових шляхів на  $\gamma_{k_1} \in \Omega_{k_1}$  має такий вигляд [10, 11]:

$$J_k(b_{kk}) = \frac{1}{2} \int_0^T \left( \|u_k(t, z, b_k) - U_k^*\|^2 \right) dt \quad (5)$$

Математично обґрунтовані формули Фур'є для знаходження кінчених інтегральних перетворень наддадуть змогу знайти розв'язок постановленій задачі [11].

Виконаємо інтегральні перетворення:

$$\begin{aligned} F[u_k(t, z)] &= \int_{k-1}^{l_k} u_k(t, z) V_m(\beta_m, z) dz \equiv U_{km}(t), \\ F^{-1}[U_{km}(t)] &= \sum_{m=0}^{\infty} U_{km}(t) \frac{V_m(\beta_m, z)}{\|V_m(\beta_m, z)\|^2} \equiv u_k(t, z), \\ F\left[\frac{\partial^2}{\partial z^2} u_k(t, z)\right] &= -\beta_m^2 U_{km}(t) + \beta_m U_{l_{k-1}} - \beta_m (-1)^m U_{l_k}, \end{aligned} \quad (6)$$

$$V_m(\beta_m, z) = \sin \beta_m (z - l_{k-1}), \quad \beta_m = \frac{m\pi}{\Delta h}, \quad \|V_m\|^2 = \int_{l_{k-1}}^{l_k} [V_m(\beta_m, z)]^2 dz = \frac{\Delta l}{2},$$

В результаті отримаємо:

$$\frac{d^2}{dt^2} U_{km}(t, z) = -b_k^2 \beta_m^2 U_{km}(t) + b_k^2 \beta_m (U_{l_{k-1}} - (-1)^m U_{l_k}) + S_{km}^*(t), \quad (7)$$

$$u_{km}(t, z)|_{t=0} = 0, \quad \frac{\partial u_{km}}{\partial t} \Big|_{t=0} = 0, \quad k = \overline{1, n_1 + 1}. \quad (8)$$

Дана задача є подібною до задачі Коші, а отже і розв'язкам її буде:

$$U_{km}(t) = \int_0^t \frac{\sin b_k \beta_m (t-\tau)}{b_k \beta_m} \left[ S_{km}^*(\tau) + b_k^2 \beta_m \left( U_{l_{k-1}} - (-1)^m U_{l_k} \right) \right] d\tau \quad (9)$$

Вертаємось до початкових функцій, в класичній формулу отримується один допустимий розв'язок [11]:

$$u_k(t, z) = \int_0^t \int_{l_{k-1}}^{l_k} \mathbf{H}_k^1(t-\tau, z, \xi) S_k^*(\tau, \xi) d\xi d\tau + \int_0^t \left( \mathbf{H}_k^{21}(t-\tau, z, l_{k-1}) U_{l_{k-1}} - \mathbf{H}_k^{22}(t-\tau, z, l_k) U_{l_k} \right) d\tau. \quad (10)$$

Вектора отриманих рівнянь мають такий вигляд:

$$\mathbf{H}_k^1(t-\tau, z, \xi) = \frac{2}{\Delta h} \sum_{m=0}^{\infty} \frac{\sin b_k \beta_m (t-\tau)}{b_k \beta_m} \sin \beta_m (\xi - l_{k-1}) \sin \beta_m (z - l_{k-1})$$

$$\mathbf{H}_k^{21}(t, z, l_{k-1}) = \frac{2b_k}{\Delta h} \sum_{m=0}^{\infty} \sin b_k \beta_m t \sin \beta_m (z - l_{k-1})$$

$$\mathbf{H}_k^{22}(t, z, l_k) = \frac{2b_k}{\Delta h} \sum_{m=0}^{\infty} \sin(b_k \beta_m t) (-1)^m \sin \beta_m (z - l_{k-1}).$$

Отож у результаті отримуємо дане рівняння що є розв'язок необхідної нам задачі:

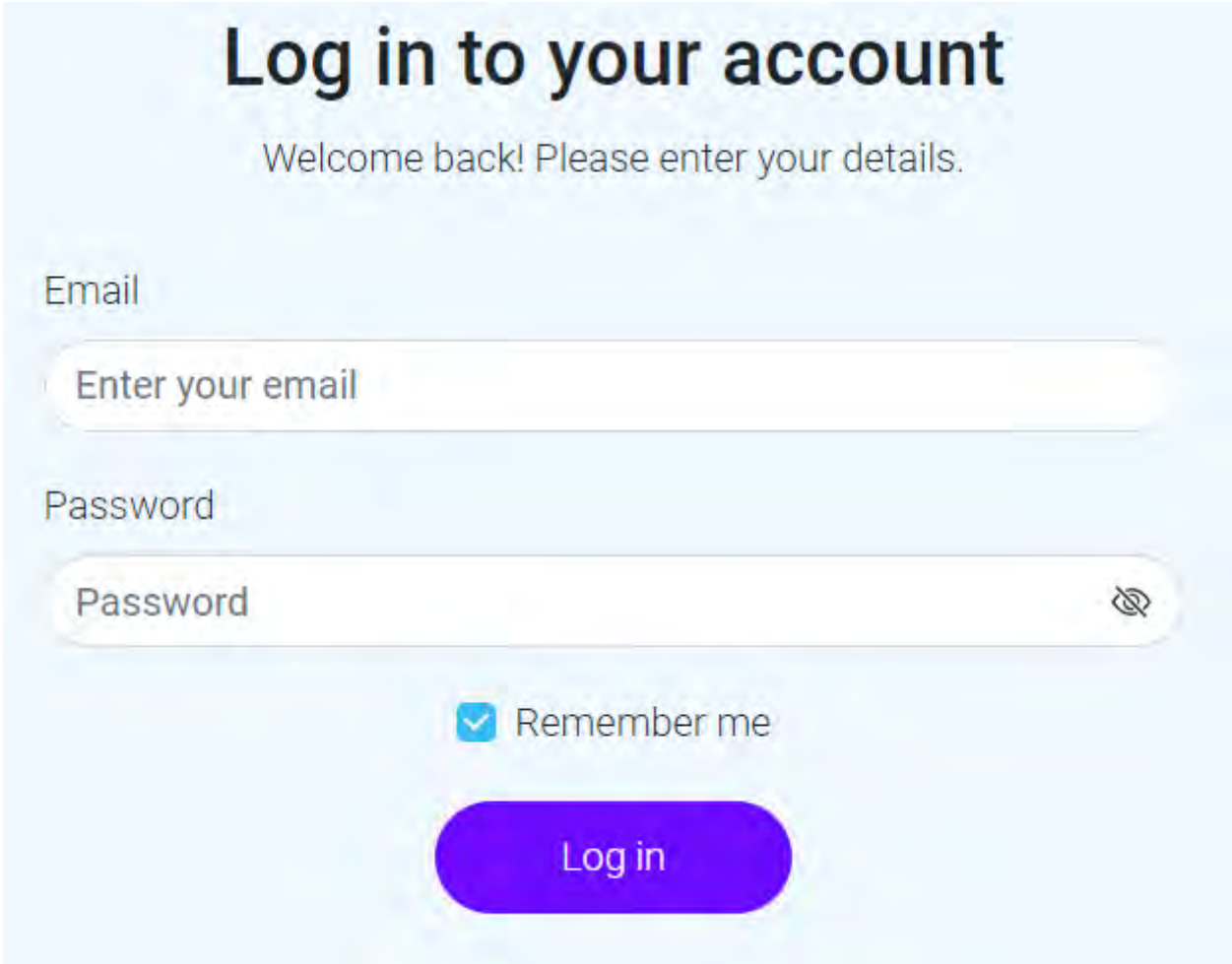
$$u_k(t, z) = \frac{2}{\Delta h} \sum_{m=0}^{\infty} \frac{1 - \cos(b_k \beta_m t)}{\beta_m} \sin \beta_m (z - l_{k-1}) \left( S_k^* \frac{1}{(b_k \beta_m)^2} \left( (-1)^m - 1 \right) + U_{l_{k-1}} \left( 1 - (-1)^m \frac{U_{l_k}}{U_{l_{k-1}}} \right) \right). \quad (11)$$

### 3.4. Ілюстрація роботи створеного програмного забезпечення

В результаті проведеної роботи на виході отримано клієнт-серверний застосунок, з простим для розуміння користувацьким інтерфейсом, метою, якого є інтеракції з логічними методами вирахування коефіцієнтів та презентації їх даних у вигляді графіків та діаграм.

Першим елементом даного застосунку з котрим зіткнеться користувач буде вікно авторизації. Виключенням стане варіант якщо користувач уже мав вдалу спробу входу в аккаунт та не здійснив вихід з нього, тоді програма здійснить перевірку на наявність в локальних або куки файлах на наявність ключа

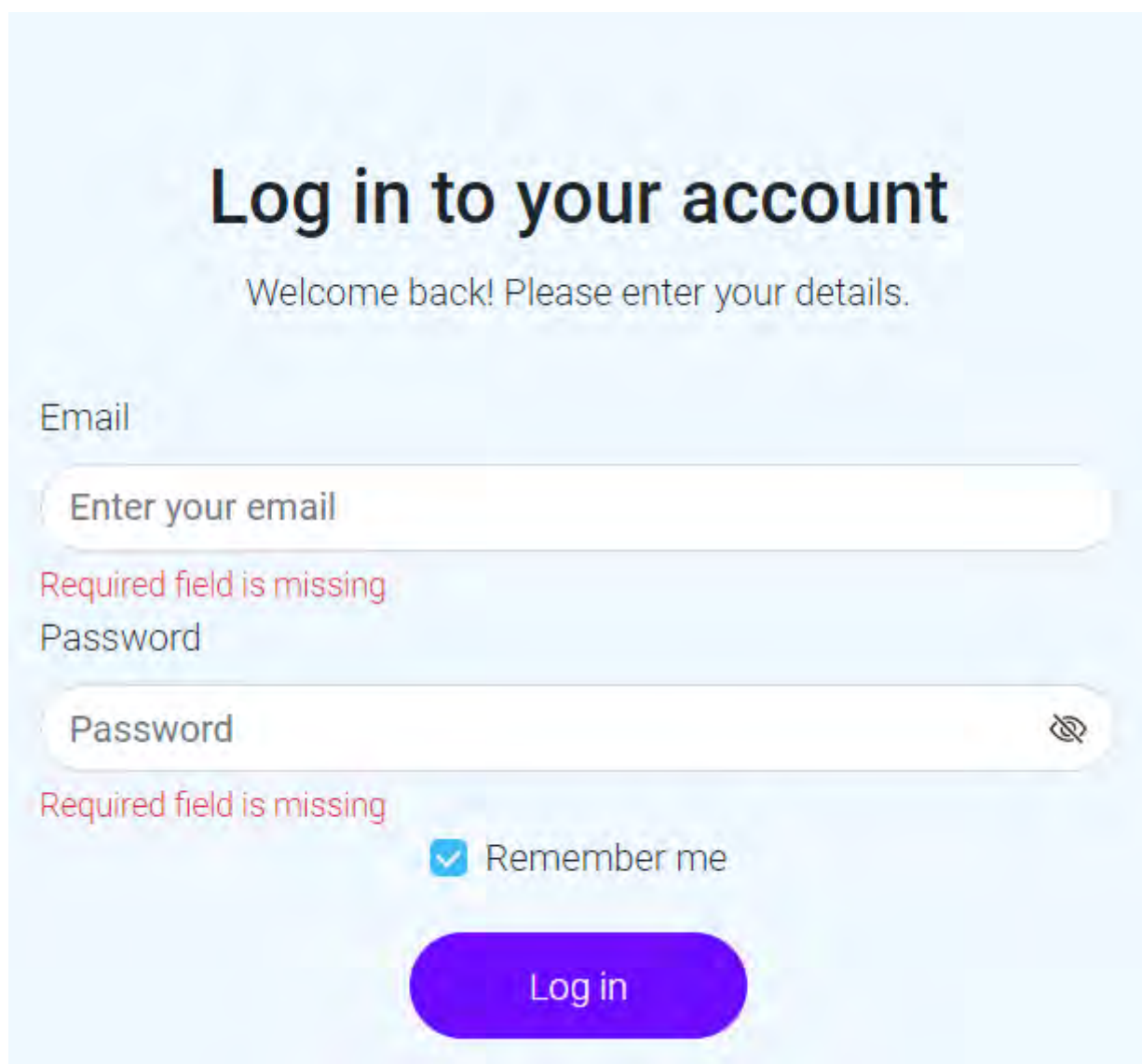
авторизації і при його наявності та коректності дане вікно пропуститься самою програмою.



The image shows a login form with a light blue background. At the top, the title "Log in to your account" is displayed in a large, bold, black font. Below the title, a subtitle reads "Welcome back! Please enter your details." in a smaller, grey font. The form contains two input fields: "Email" and "Password". The "Email" field has a placeholder text "Enter your email". The "Password" field has a placeholder text "Password" and a small icon of an eye with a slash through it, indicating a toggle for password visibility. Below the password field, there is a checkbox labeled "Remember me" which is checked. At the bottom of the form, there is a prominent purple button with the text "Log in" in white.

Рисунок 3.7 – Вікно авторизації користувача

У випадку пропущених або невірно креденційних даних користувач побачить індикатори котрі вказують на помилки котрі він здійснив у полях вводу електронної адреси пошти чи паролю. Також на рисунках представлено селектори Remember me або ж Запам'ятати мене. Даний селектор необхідний для надання згоди користувачем на те що у сесійні або кукіс файли програмою запишуться ключу доступу авторизації, завдяки яким зникне необхідність в повторних введеннях даних при наступних спробах входу в аккаунт.



**Log in to your account**

Welcome back! Please enter your details.

Email

Enter your email

Required field is missing

Password

Password

Required field is missing

Remember me

Log in

The image shows a login form on a light blue background. At the top, the title 'Log in to your account' is displayed in a large, bold, black font. Below the title is a subtitle 'Welcome back! Please enter your details.' in a smaller, grey font. The form consists of two input fields: 'Email' and 'Password'. The 'Email' field contains the placeholder text 'Enter your email' and has a red error message 'Required field is missing' below it. The 'Password' field contains the placeholder text 'Password' and has a red error message 'Required field is missing' below it. To the right of the password field is a toggle icon for visibility. Below the password field is a checked checkbox labeled 'Remember me'. At the bottom of the form is a large, rounded, purple button labeled 'Log in'.

Рисунок 3.8 – Індикатори у випадку пропущених полів при спробі входу

В разі успішної авторизації роутер перенаправить користувача зі сторінки авторизації на сторінку вирахування коефіцієнтів. На рисунку нижче зображено лістинг коду що до даного завдання (рис. 3.9).

# Calculation

Spiral data

Вибрати файл    Файл не вибрано

EEG data

Вибрати файл    Файл не вибрано

Points range

30

EEG port

A0 × ▼

Submit

Рисунок 3.9 – Вікно вирахування коефіцієнтів

Як можна побачити на рисунку вище дана програма запрошує дані у файлових форматах з уже зчитаними сигналами та імпульсами КГМ зчитаними за допомогою електроенцефалограми. Користувач може задати необхідну роздільну здатність діаграми та обрати необхідний йому порт частину КГМ. Також надано можливість споглянути електроміограф у не нормалізованій формі, що зображено нижче на рисунку 3.10



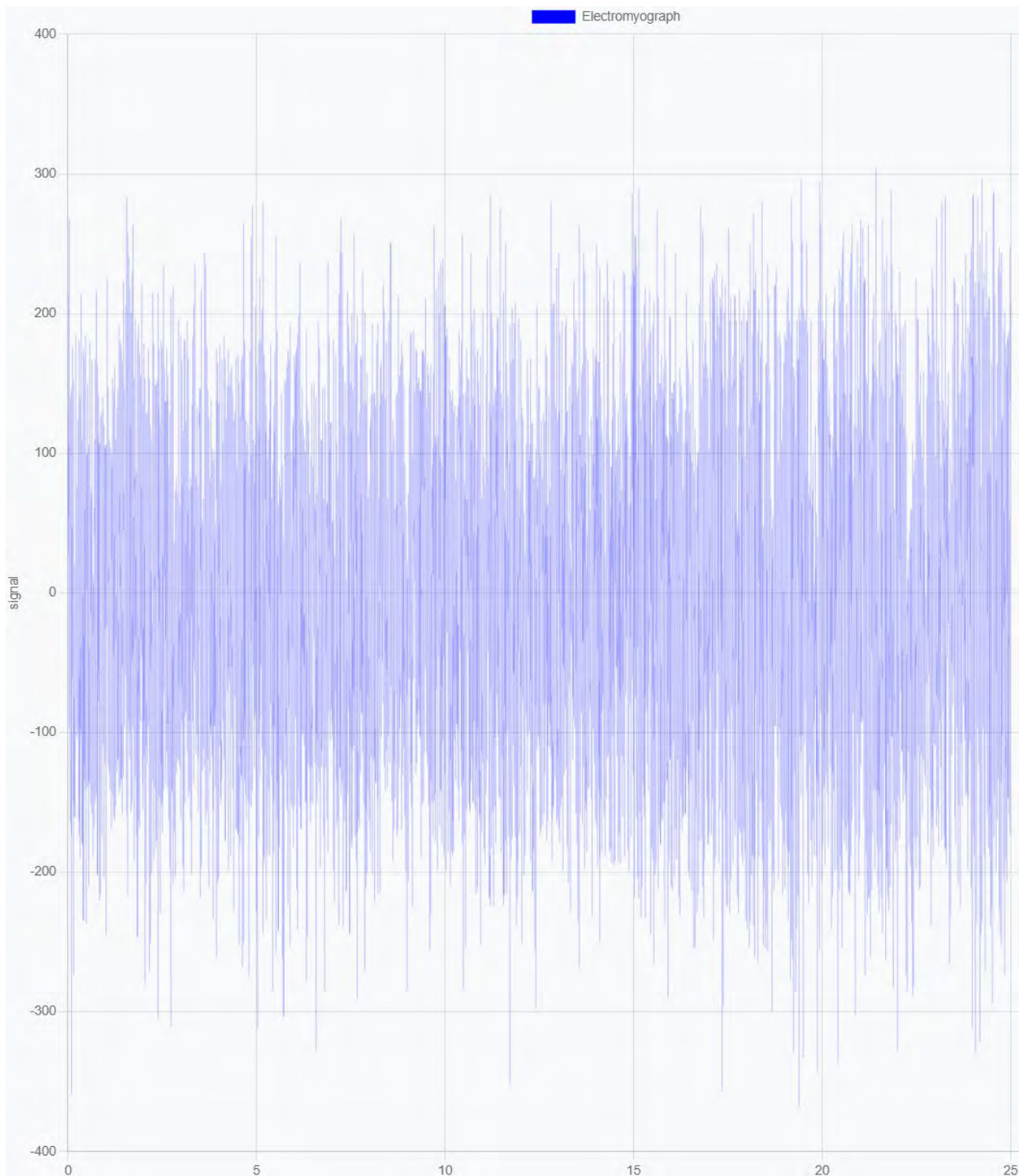


Рисунок 3.10 – діаграма електроміографа КГМ пацієнта на каналі С3

Також користувачу надано можливість оглянути діаграму котра відображає результати перевірок математичної моделі.

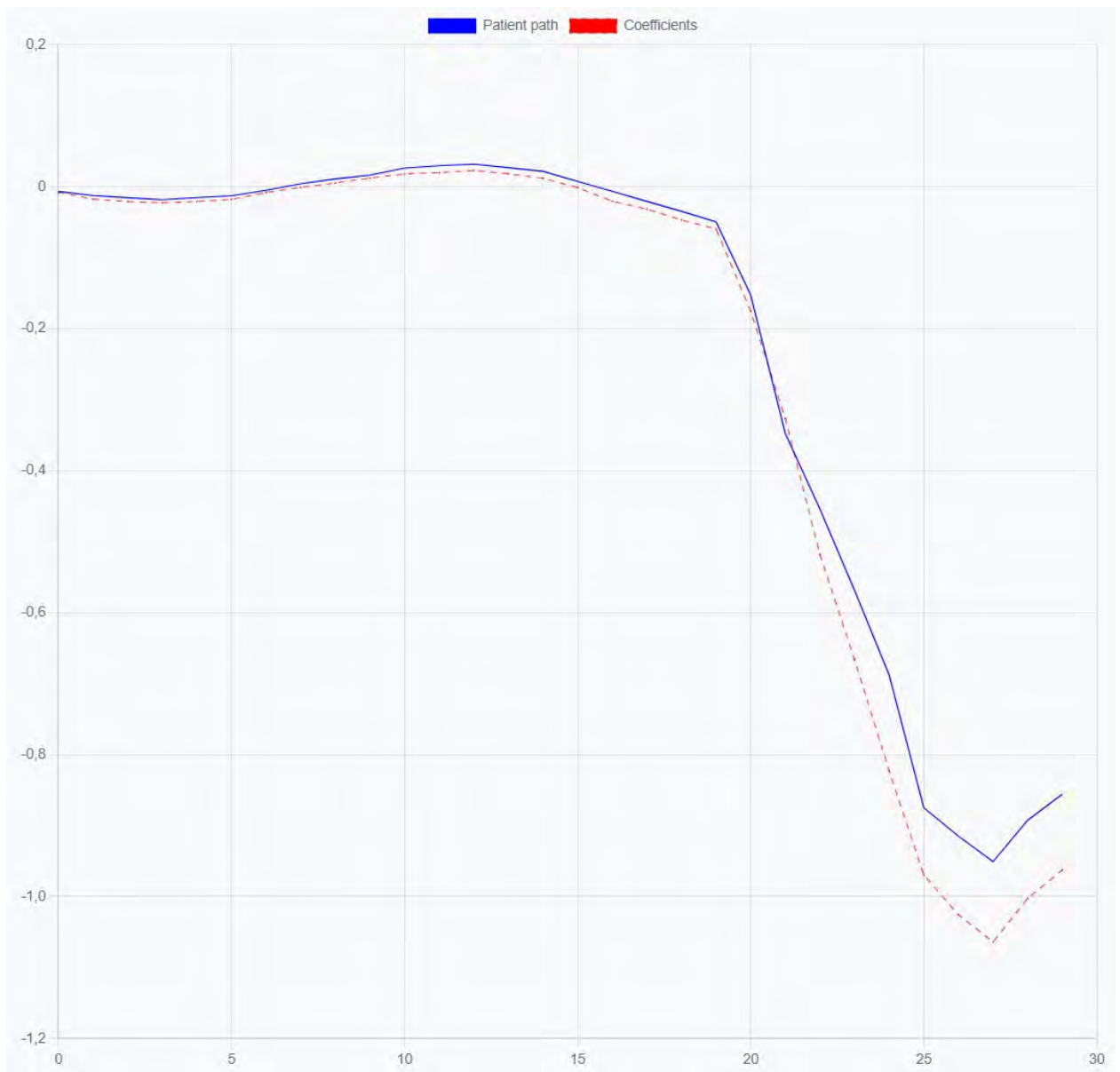


Рисунок 3.11 – Діаграма коефіцієнтів

Наступними на черзі є розгортки нейронних сигналів електроенцефалографа зчитаних із кори головного мозку, на цих трасах відображається коливання та частоти котрі спостерігаються під час тесту опорно-рухового апарату пацієнта. Тестова модель зображена на рис. 3.12

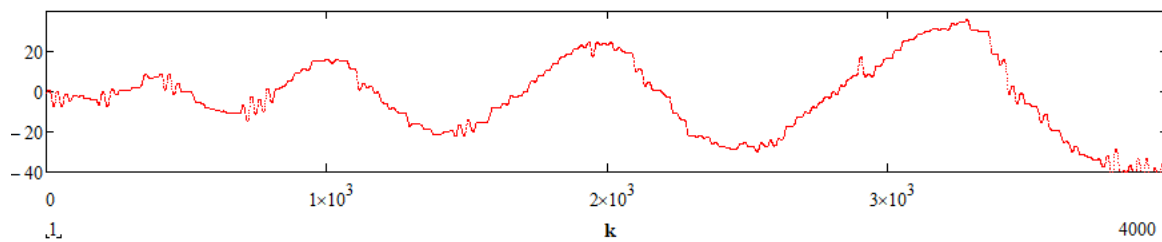


Рисунок 3.12 Розгорнута АНР-траса

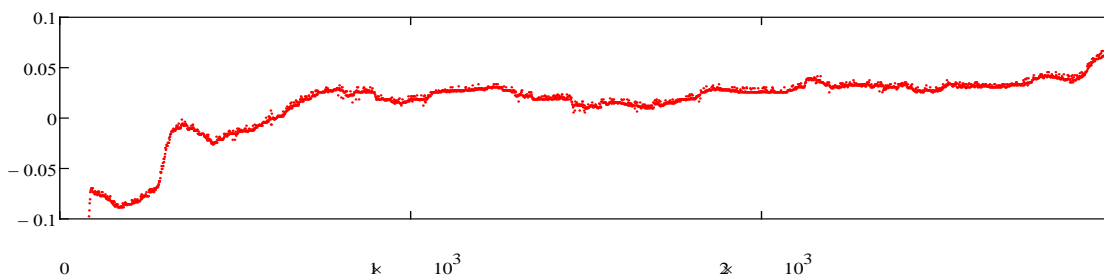


Рисунок 3.13 Тестовий розподіл сигналів ЕЕГ

На рисунку 3.13 представлено розподіл ЕЕГ-сигналів, та силу їхнього впливу на руховий апарат пацієнта під час проходження тестування.

Задля отримання вище зображених результатів необхідно виконати матричні перетворення амплітудних сигналів на кожній ділянці траси аномальних неврологічних рухів котру проходив пацієнт під час сичтування ЕЕГ сигналів. По даному перетворюванню буді відбуватись налаштовування та корегування моделі для зведення похибки при інтерпритації зчитаних даних до нуля, як зображено на рисунку 3.14.

$$\begin{bmatrix} u_1(t, z) \\ \dots \\ u_j(t, z) \\ \dots \\ u_{n+1}(t, z) \end{bmatrix} = \int_0^t \begin{bmatrix} \int_{l_0}^{l_k} Hh_1(t-\tau, z, \xi) & \dots & \int_{l_n}^{l_{n+1}} Hh_{1,n+1}(t-\tau, z, \xi) \\ \dots & \dots & \dots \\ \int_{l_0}^{l_k} Hh_j(t-\tau, z, \xi) & \dots & \int_{l_n}^{l_{n+1}} Hh_{j,n+1}(t-\tau, z, \xi) \\ \dots & \dots & \dots \\ \int_{l_0}^{l_k} Hh_{n+1}(t-\tau, z, \xi) & \dots & \int_{l_n}^{l_{n+1}} Hh_{n+1,n+1}(t-\tau, z, \xi) \end{bmatrix} \cdot \begin{bmatrix} S_1(\tau, \xi) \\ \dots \\ S_i(\tau, \xi) \\ \dots \\ S_n(\tau, \xi) \end{bmatrix} d\xi d\tau$$

Рисунок 3.14 Формула матричних перетворювань ЕЕГ сигналу

На рисунку наведеному нижче можна спостерігати результати тестованої траси. Роздільна здатність даної траси для початку була обрана мала для кращого профілювання амплітудних значень.

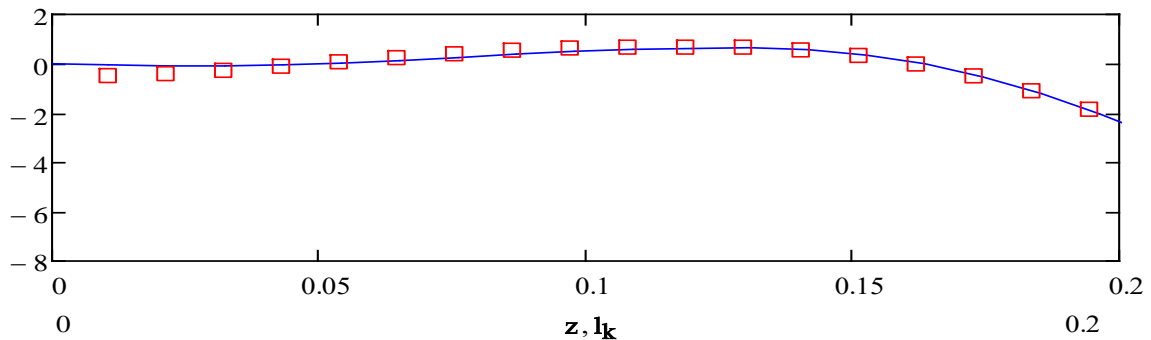


Рисунок 3.15 Діаграма порівняння проходження АНР-траси(червона) з моделлю (синя) з роздільною здатністю 0.2

Можемо спостерігати похибка під час проходження даної траси пацієнтом під час тестування сягає менше 2% у порівнянні із ідеальною моделлю на роздільній здатності 0.2 або ж 20 початкових точок обсервації. А також спираючись на методи перетворення матриці котрі було створенно раніше для кожної ділянки траси та знаходження їх зав'язків, не у вакуумі, отримано високу точність наставлення двох шляхів положених одне на одну. Наступними кроками будуть часткові збільшення роздільної здатності трас (рис. 3.16).

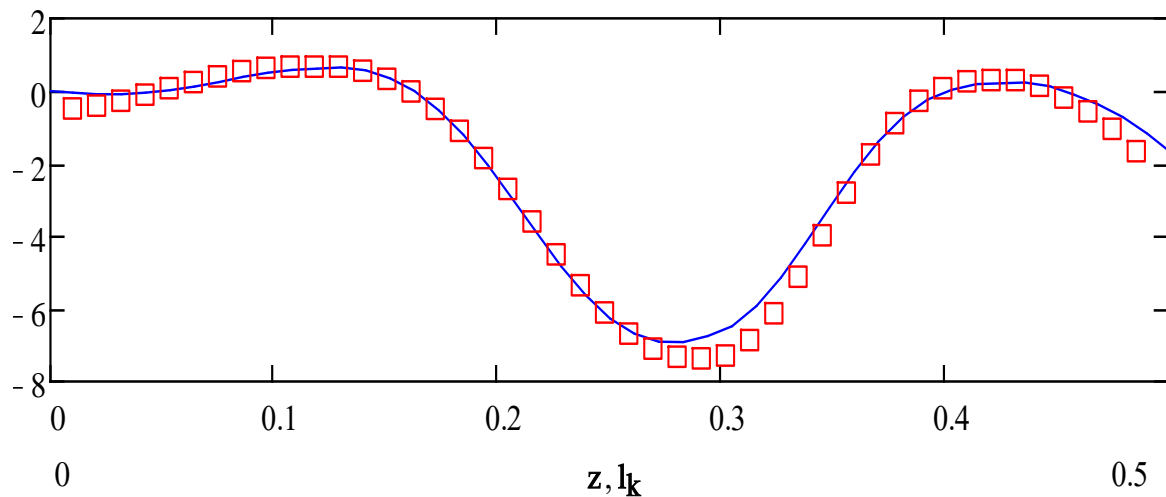


Рисунок 3.16 Діаграма порівняння проходження АНР-траси(червона) з моделлю (синя) з роздільною здатністю 0.5

Проаналізувавши хвильові функції з початковим значенням роздільної здатності в 1 та подальшим її збільшенням до 5, 10, 20 та 40 було визначено відхилення кривих від ідеальних моделей АНР траси (див. рис. 3.17 - 3.21).

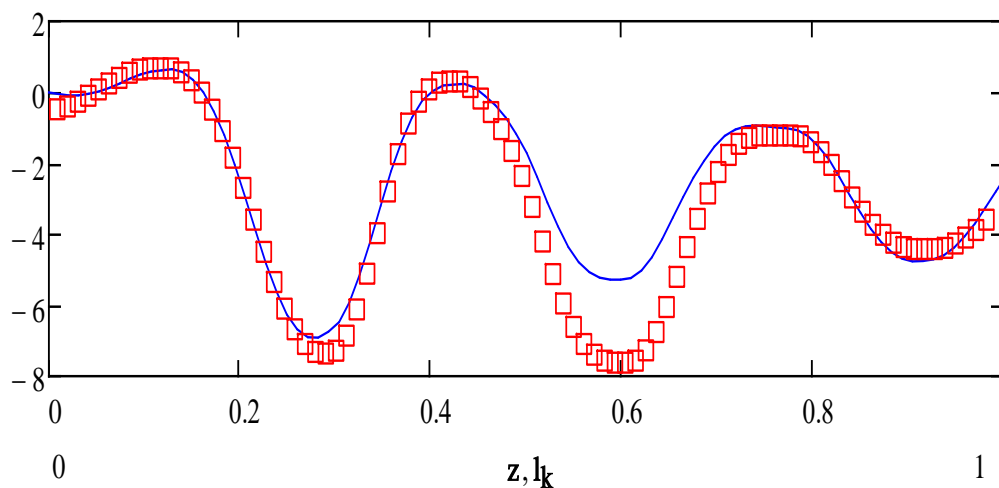


Рисунок 3.17 Діаграма порівняння проходження АНР-траси(червона) з моделлю (синя) з роздільною здатністю 1

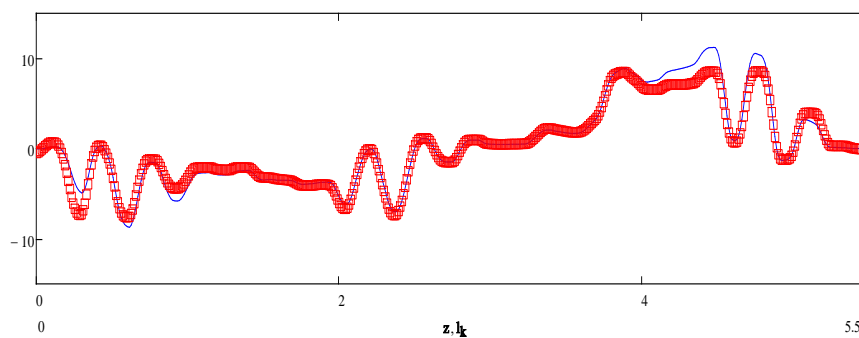


Рисунок 3.18 Діаграма порівняння проходження АНР-траси(червона) з моделлю (синя) з роздільною здатністю 5

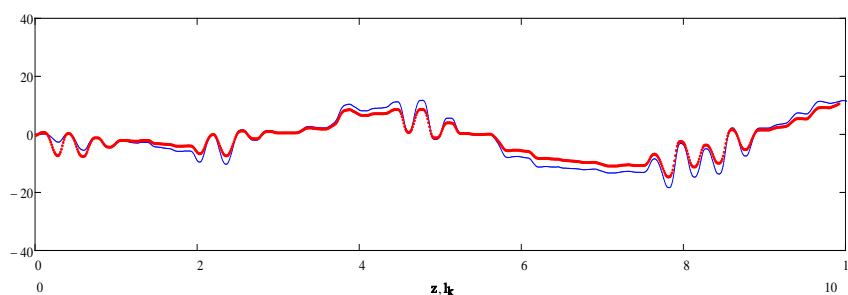


Рисунок 3.19 Діаграма порівняння проходження АНР-траси(червона) з моделлю (синя) з роздільною здатністю 10

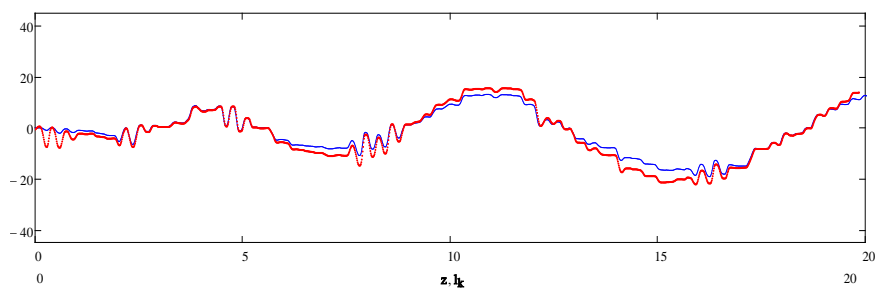
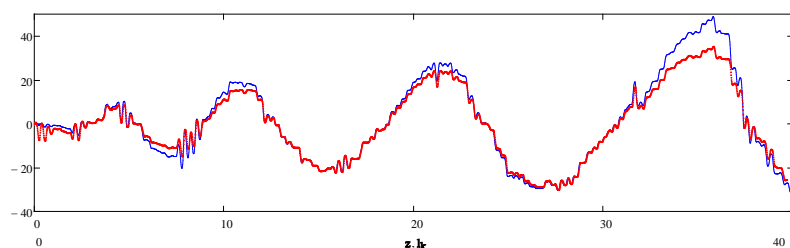
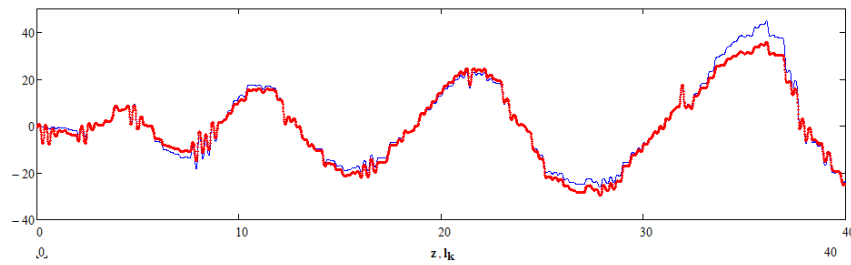


Рисунок 3.20 Діаграма порівняння проходження АНР-траси(червона) з моделлю (синя) з роздільною здатністю 20



a)



b)

Рисунок 3.21 Діаграма порівняння проходження АНР-траси(червона) з моделлю (синя) з роздільною здатністю 40

На рисунку наведеному вище представлено дві діаграми з роздільною здатністю 40 проте з різними циклічностями параметрів визначення. За мету даного розгалуження було поставлено знаходження найбільш коректного представлення діаграм перед користувачем. Важливо виділити те що на обох діаграмах спостерігається ретельне слідування ідеальній моделі траси а коливання шляху пройденого пацієнтом практично ідентичне коливанням зазначеним у ідеальній моделі даної траси.

Отож у результаті усіх вище перелічених задач, функцій методів діаграм користувачеві надається повна розгортка як і зчитаних з допомогою електроенцефалографа сигналів кори головного мозку так і розгортки пройдених тестів на визначення треморних захворювання у опорно-руховому апараті.

### 3.5 Тестування програмного забезпечення та оцінка якості

Як відомо, програм без помилок не існує. Дане програмне забезпечення потребує тестування, щоб переконатися, що воно відповідає стандартам розробки та що алгоритми, які було реалізовано, вірні.

Перш за все програма була перевірена на стійкість до помилок, тобто випробувано внесення неправильних параметрів або відсутності вхідних даних, а також внесення неправильних або некоректних вхідних даних під час використання. Коли система внесла дані з невідповідними значеннями формату представлення даних, система викинула помилку в лог, що означає, що причину помилки ми знаємо, але система поки що не реагує з точки зору інформативності.

У комп'ютерному програмуванні модульне тестування ПЗ використовуються для перевірки одного або більше модулів програми, методів класів або частин програми для визначення, чи вся бізнес-логіка правильна та відповідає задуманим цілям і результатам. Автоматизовані функціональні тести перевіряють код програми на правильну роботу, а розробник може сам проганяти їх, коли це необхідно.

```

WARNING in ./src/styles/index.sass (./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[7].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[7].use[2]!./node_modules/resolve-url-loader/index.js??ruleSet[1].rules[1].oneOf[7].use[3]!./node_modules/sass-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[7].use[4]!./src/styles/index.sass)
Module Warning (from ./node_modules/postcss-loader/dist/cjs.js):
Warning
(2583:3) autoprefixer: Replace color-adjust to print-color-adjust. The color-adjust shorthand is currently deprecated.
@ ./src/styles/index.sass 8:6-387 22:17-24 26:7-21 58:25-39 59:36-47 59:50-64 63:6-73:7 64:54-65 64:68-82 70:42-53 70:56-70 72:21-28 83:0-357 83:0-357 84:22-29 84:33-47 84:50-64
61:4-74:5
@ ./src/App.tsx 11:0-29
@ ./src/index.tsx 6:0-24 9:38-41

webpack 5.74.0 compiled with 1 warning in 7797 ms
No issues found.

```

Рисунок 3.22 Перевірка тестів на наявність помилок у кодї

На розглянутому вище рисунку можна побачити відсутність будь яких важливих або критичних помилок або збоїв роботи системи, єдине попередження про можливо застарілість використаного параметру.

Загалом тестування можна вважати успішним. Моделі потребують ретельної перевірки у використанні на практиці в дослідженнях і потребуватимуть виправлень і змін під час розвитку та дослідження проекту. Система продемонструвала, що проект працює та має право на існування.



## 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Вимоги до виробничих приміщень для експлуатації ВДТ

Для того, щоб забезпечити якісну та ефективну роботу працівників, необхідно дотримуватися основних правил користування комп'ютерною технікою, основних норм і правил охорони праці. Це пов'язано з тим, що проектування та розробка інформаційної системи, яка ідентифікує параметри аномальних неврологічних рухів людини, відбувалося за допомогою персональних комп'ютерів.

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності [12].

Кодекс законів про працю, Закон України «Про охорону праці» та інші законодавчі акти регулюють охорону праці, зокрема охорону праці офісних працівників. Відповідно до ст. 153 Кодексу законів про працю України та ст. 6 Закону України «Про охорону праці» на всіх підприємствах повинні бути забезпечені безпечні та нешкідливі умови праці [12], [13]. Власник або уповноважений ним орган повинен нести відповідальність за забезпечення

Робочі місця працівників, обладнані комп'ютерами, повинні відповідати вимогам ДСанПІН 3.3.2.007-98 «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-7.15-18 «Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Робочі місця з екранними пристроями мають бути спроектовані та розраховані таким чином, щоб працівники могли рухатися та змінювати положення на робочому місці, щоб забезпечити безпеку та захист їхнього здоров'я. Гранично

допустимий рівень впливу на людину факторів довкілля, таких як шум, вібрація, забруднювачі та температура, має бути знижений.

Усі компоненти та розташування робочого місця з екранами повинні відповідати ергономічним, антропологічним і психофізіологічним вимогам, а також характеру виконуваних робіт [14].

На робочому місці, де виконуються роботи сидячи, необхідно дотримуватися загальних ергономічних вимог з урахуванням характеру та особливостей трудової діяльності. Ці вимоги включають відстань 1,2 м між бічними поверхнями ВДТ і 2,5 м від тильної поверхні ВДТ до екрана ВДТ. Ці відстані розраховуються з урахуванням характеристик стін і перегородок.

Коли користувач працює за комп'ютером, слід дотримуватися наступних рекомендованих правил під час роботи: підшва на підлозі або підставці для ніг; стегна горизонтально; передпліччя вертикально; лікті під кутом 70–90 градусів до вертикальної поверхні; зап'ястя зігнуті під кутом 20 градусів відносно горизонтальної поверхні; і нахил голови 20 градусів відносно вертикальної поверхні. Оскільки інноваційні технології розвиваються, деякі галузі вимагають використання широкоформатних моніторів. У цих випадках відстань у 60 см між користувачем і дисплеєм є дуже небезпечним, оскільки це порушує медичні норми роботи з обчислювальними пристроями. Згідно з цими нормами оптимальна відстань між користувачем і дисплеєм повинна становити від півтора до двох діагоналей.

Клавіатура має бути розташована на відстані 100–300 мм від краю столу, звернутого до працівника. У конструкції клавіатури має бути опорний пристрій, який дозволяє змінювати кут нахилу поверхні [15].

Робочий стіл оператора обчислювальної машини повинен бути оснащений наступними компонентами: сидінням, спинкою та стаціонарними або змінними підлокітниками. Стілець також може бути оснащений аксесуарами, які покращують умови роботи, наприклад підголовником і підставкою для ніг.

#### 4.2 Основні елементи підвищення стійкості функціонування лікувального закладу в умовах надзвичайних ситуацій

Важлива роль у виконанні завдань медичного забезпечення населення в умовах НС належить об'єктам охорони здоров'я: лікарням, поліклінікам, санепідстанціям, станціям переливання крові, аптекам і аптечним складам. Вони є базою створення закладів і формувань державної служби медицини катастроф, беруть участь у виконанні лікувально-евакуаційних, санітарно-гігієнічних і протиепідемічних заходів, забезпечують об'єкти охорони здоров'я і службу медицини катастроф ЛЗ. Від стійкості функціонування об'єктів охорони здоров'я, організації взаємодії між ними залежить вирішення завдань з медичного забезпечення населення в умовах НС.

Підвищення стійкості функціонування об'єкта охорони здоров'я в умовах НС – це завчасне проведення комплексу адміністративно-організаційних, інженерно-технічних, матеріально-економічних, санітарно-протиепідемічних заходів, спрямованих на підготовку об'єкта до роботи в умовах НС мирного і воєнного часу і виконання ним завдань з надання медичної допомоги і лікування постраждалого населення.

Для забезпечення стійкості функціонування ЛПЗ в умовах НС, вони повинні відповідати певним медико-технічним вимогам, які поділяються на загальні і спеціальні. До загальних медико-технічних вимог відносяться вимоги, що специфічні для закладів охорони здоров'я і реалізуються у всіх проектах. До спеціальних відносяться вимоги, які залежать від природних чинників (сейсмічність, низькі ґрунтові води), від регіону забудови (близькість АЕС, хімічно небезпечних об'єктів, вибухо- і пожежонебезпечних об'єктів), від типу ЛПЗ (лікарня, поліклініка, станція переливання крові). Для деяких категорій лікарень при їх плануванні необхідно передбачити майданчик для посадки вертольотів

(літаків), обов'язково забезпечити можливість роздільного в'їзду і виїзду автотранспорту, що прибуває в ЛПЗ, а також обладнати пристосування для вивантаження хворих. Вирішення усіх цих питань входить у перелік підвищення стійкості функціонування ЛПЗ. Крім того, виконується ряд технічних та інших вимог.

При визначенні системи надійності енергопостачання та електроосвітлення закладів охорони здоров'я повинні передбачатися варіанти аварійного освітлення за допомогою пересувних електростанцій, які забезпечують підключення мереж у будівлі. Пересувні електростанції використовуються насамперед для освітлення операційних (пологових), перев'язочних, палат інтенсивної терапії, а також для підключення переносних електроламп у приймальному відділенні, палатах і коридорах за допомогою запасних штепсельних розеток. Особлива увага звертається на технічні пристрої підключення до електромережі, здатність їх протистояти різким коливанням при землетрусі (у сейсмонебезпечних зонах).

Аварійне теплопостачання забезпечується створенням запасів газу в балонах та інших видів палива (для котельних або печей) на період відновлення основного джерела теплопостачання, а також можливістю подачі газу від зовнішніх мереж за допомогою спеціальних пристроїв.

Водопостачання в НС забезпечується шляхом створення запасів питної води з розрахунку 2 л/добу на хворого (постраждалого) і технічної води — по 10 л/добу на ліжку. Аварійні ємкості встановлюються у верхній частині будівлі або в окремій водонапірній башті (басейні). Передбачається можливість подачі води за допомогою трубопроводів (гнучких шлангів) від зовнішніх мереж або пересувних засобів із застосуванням спеціальних приєднувальних конструкцій.

Каналізаційна система в лікувальних закладах, які за планом призначені для приймання постраждалих з територій, забруднених радіоактивними речовинами, повинна забезпечувати проведення дезактивації з урахуванням безпеки для персоналу і довкілля (спеціальні відстійники у системі очисних споруд).

Для захисту будівель лікувальних закладів від радіоактивних та хімічних речовин, задимленості й інших шкідливих чинників створюється максимально можлива герметичність внутрішніх приміщень при закритих вікнах; система вентиляції повинна за необхідності створювати підпір повітря у палатах, операційних і процедурних і мати систему фільтрів у місцях забору повітря.

Система внутрішньої лікарняної безпеки від уражуючих чинників (пожежо- і вибухонебезпечні речовини, пристрої і матеріали; отруйні і радіоактивні речовини; патогенні для людини бактерії, віруси і гриби) повинна плануватися і створюватися так, щоб хворі і постраждалі не мали контактів з перерахованими чинниками. Це досягається раціональним розподілом потоків хворих і обслуговуючого персоналу, а також раціональним розміщенням і устаткуванням відповідних приміщень лікарні, створенням системи вентиляції і шлюзів, що перешкоджають поширенню шкідливих чинників за межі робочих приміщень.

Для захисту хворих у стаціонарних ЛПЗ повинно бути передбачено будівництво захисних споруд — сховищ або протирадіаційних укриттів. Засоби зв'язку в лікарні повинні забезпечувати постійну можливість подачі сигналу тривоги у всі приміщення, де знаходяться хворі і персонал, через радіомережу або іншу систему гучного зв'язку. Чергова зміна на чолі з керівництвом лікарні повинна бути оснащена портативними переносними засобами зв'язку для роботи усередині будівлі і поза нею у межах чутності. Ці ж засоби зв'язку можуть використовуватися при виїзді медичних бригад в осередок НС.

У великих ЛПЗ необхідно мати автоматизовану систему реєстрації постраждалих і банк даних про історії хвороби для їх швидкої статистичної обробки. Система екстреної евакуації хворих повинна бути доповнена індивідуальними рятувальними пристроями, які можуть використовуватися при порушеннях евакуації звичайним порядком: через вікна на першому поверсі, а починаючи з другого і вище — з використанням трапів, запасних сходів, спеціальних мереж або інших пристроїв, що дозволяють опустити людину на безпечний майданчик.

Найважливішим елементом стійкості роботи закладів охорони здоров'я є резерви ЛЗ і медичного майна, які створюються на випадок НС. До медичного майна відносять: лікарські засоби, антидоти, радіопротектори, вироби медичного призначення, медичну техніку, дезінфекційні засоби й інші витратні матеріали, засоби транспортування, автономні джерела електроживлення до приладів тощо. Їх накопичення здійснюється за табелями оснащення медичних формувань, що створюються на період НС.

Для станцій швидкої медичної допомоги, станцій переливання крові, санепідстанцій обов'язково передбачаються складські приміщення з холодильниками, ємкість яких визначається потребою у зберіганні препаратів, що вимагають дотримання температурного режиму.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи бакалавра було розроблено веб-застосунок для реєстрації та зберігання ЕЕГ когнітивних сигналів керування елементами опорно-рухового механізму пацієнтів з використанням об'єктно орієнтованих архітектури. В даній системі реалізовано користувацький інтерфейс та процедуру автентифікації користувача.

Провівши аналіз предметної області було визначено що даний застосунок внесе свій вклад в допомогу діагностування захворювання есенціального тремору та споріднених до нього захворювань. Також було проведено аналіз методів за якими відбувалась реєстрація сигналів КГМ Т-об'єктів.

В ході роботи виконувалось зчитування вхідних даних з подальшої їх обробкою та візуалізацією. Ця система дозволить отримати більш наглядні співставлення рухів пацієнтів з сигналами нейро-вузлів котрі їм передає головний мозок при проходженні АНР-трас котрі в подальшому було розв'язано з допомогою формул перетворення гібридних інтегралів Фур'є. Оскільки подібних систем не існувало або не було у відкритому доступі медичним працівникам доводилось в ручну співставляти дані показники із реальними рухами що залишало місце для людської помилки або неточностей котрі в подальшому могли переростати в хибні твердження що до діагнозу.

Також система передбачає збереження результатів аналізів. Ця можливість забезпечує повторне використання даних для аналізу чи маніпуляцій над ними.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Elble R.J., Brilliant M., Leffler K., Higgins C. Quantification of Essential Tremor in Writing and Drawing. *Movement Disorders*. 1996; II(1):70-8.
2. Матвієнко Ю. О. Есенціальний тремор - поширений руховий розлад нервової системи / Ю. О. Матвієнко. – 2014. – С. 18.
3. Rational unified process [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Rational\\_unified\\_process](https://en.wikipedia.org/wiki/Rational_unified_process).
4. Документація UML [Електронний ресурс] – Режим доступу: <http://www.uml.org/>.
5. Серверна мова програмування Node.js [Електронний ресурс] – Режим доступу: <https://nodejs.org/en/>.
6. Документація бази даних MongoDB [Електронний ресурс] – Режим доступу: <https://www.mongodb.com/home>
7. Редактор коду Visual Studio Code [Електронний ресурс] – Режим доступу: <https://code.visualstudio.com/>.
8. Rajaraman V., Jack D., Adamovich S.V., Hening W., Sage J., Poizner H. A novel quantitative method for 3D measurement of Parkinsonian tremor. *Clinical neurophysiology*, 11(2), 187-369 (2000)
9. Мудрик І. Я. Автоматизовані системи діагностування стану пацієнтів, хворих на есенціальний тремор / Мудрик Іван Ярославович, 2021.
10. Перетворення Фур'є, Лапласа: узагальнення та застосування / Г. П. Лопушанська, А. О. Лопушанський, О. М. М'яус. – Львів, 2014.
11. Перетворення Фур'є [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F\\_%D0%A4%D1%83%D1%80%27%D1%94](https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F_%D0%A4%D1%83%D1%80%27%D1%94).



12. Закон України «Про охорону праці» [Електронний ресурс] // № 2695 – XII від 14.10.1992 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12>.
13. Кодекс законів про працю України [Електронний ресурс] // № 322 – VIII від 10.12.1971 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/322-08>.
14. Наказ України «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [Електронний ресурс] 80 // № 207 від 14.02.2018 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18>.
15. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Електронний ресурс] // № 7 від 10.12.1998 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18>.

ДОДАТКИ

УДК 004.942: 616-71

М.Р. ПЕТРИК, І.Я. МУДРИК, М.В. БАЧІНСЬКИЙ, І.Я. СТАДНИК,  
 М.І. ПІДГУРСЬКИЙ, В.О. ЯМКО  
 Тернопільський національний технічний університет ім. І. Пулюя

### ВИСОКОПРОДУКТИВНІ МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ МОДЕЛЮВАННЯ ТА ІДЕНТИФІКАЦІЇ ПАРАМЕТРІВ АНОРМАЛЬНИХ РУХІВ ПІД ДІЄЮ ЗВОРОТНИХ КОГНІТИВНИХ ВПЛИВІВ

*Нові методи моделювання використовуються для забезпечення підходу до проектування цифрових діагностичних систем здоров'я для пацієнтів із неврологічними захворюваннями. Актуальним завданням є створення нових програмно-апаратних рішень для медицини та автоматизованих діагностичних систем для ідентифікації нових явищ організму та здоров'я людини. Важливим є дослідження нейро-біо-систем зі feedback-зв'язком, пов'язаних з аналізом стану та поведінки T-об'єктів (пацієнтів з ознаками тремору) під когнітивним впливом нейронних вузлів кори головного мозку.*

*В сучасних умовах особлива увага приділяється новим цифровим системам діагностики та лікування в медичному застосуванні. Створена математична модель наномедицинської системи орієнтована на визначення параметрів аномальних рухів пацієнтів із симптомами тремору (T-об'єктів), викликаних негативними впливами певного набору нервових вузлів кори головного мозку. Визначення параметрів цих впливів окреслить шляхи вирішення проблеми.*

*Авторами описано моделі сигналів, отриманих в якості вхідних даних для опрацювання (рисунок пацієнта тесту спіралі Архімеда), проаналізовано точність та ефективність методів комп'ютеризованого аналізу ступеню тремору. Описано основні результати моделювання, отримано частотні характеристики, амплітуди коливання, відхилення від норми та інші показники. Розроблено високоефективну інформаційну технологію для оцінки неврологічних рухів людини на основі гібридної моделі аналізу хвильового сигналу з урахуванням когнітивного зворотного зв'язку нейровузлів кори головного мозку.*

*З використанням гібридних перетворень Фур'є реалізовано швидкісне аналітичне рішення моделі у векторній формі, що дозволяє визначати елементи рухів на кожному відрізку складної спіральної траєкторії, що виконуються пацієнтам за допомогою електронного пера на цифровому планшеті, та ідентифікували параметри досліджуваних нейросистем зі зворотним зв'язком.*

*Ключові слова: неврологічні рухи, тремор, математичне моделювання, когнітивні зворотні нейрозв'язки, аномальний рух, графічний планшет, електроенцефалограма, діагностика, гібридне перетворення Фур'є, апаратне та програмне забезпечення, комп'ютерне моделювання.*

M. PETRYK, I. MUDRYK, M. BACHYNSKYI, I. STADNYK, M. PIDGURSKYI, V. YAMKO  
 Ternopil Ivan Puluj National Technical University

### HIGH-PRODUCT METHODS AND INFORMATION TECHNOLOGIES FOR MODELING AND IDENTIFICATION OF THE PARAMETERS OF ABNORMAL MOVEMENTS UNDER THE INFLUENCE OF REFUSED COGNITIVE INFLUENCES

*New modeling techniques are used to provide an approach to the design of digital health diagnostic systems for patients with neurological diseases. An urgent task is the creation of new software and hardware solutions for medicine and automated diagnostic systems for identifying new phenomena of the body and human health. It is important to study neuro-bio-systems with feedback, related to the analysis of the state and behavior of T-objects (patients with signs of tremor) under the cognitive influence of neural nodes of the cerebral cortex.*

*In modern conditions, special attention is paid to new digital systems of diagnosis and treatment in medical applications. The designed mathematical model of the nanomedical system is focused on determining the parameters of abnormal movements of patients with tremor symptoms (T-objects), caused by the negative effects of a certain set of nerve nodes of the cerebral cortex. Determining the parameters of these influences will outline ways to solve the problem.*

*Authors described the models of the signals received as input data for processing (drawing of the patient of the Archimedes-spiral test), analyzed the accuracy and efficiency of the methods of computerized analysis of the degree of tremor. The main modeling results are described, frequency characteristics, oscillation amplitudes, deviations from the norm and other indicators are obtained. A highly effective information technology has been developed for the evaluation of neurological movements of a person based on a hybrid model of wave signal analysis taking into account the cognitive feedback of cerebral cortex neuronodes.*

<https://doi.org/10.32782/mathematical-modelling/2022-5-1-9>

*With the use of hybrid Fourier transformations, a fast analytical solution of the model in vector form was implemented, which allows determining the elements of movements on each segment of a complex spiral trajectory performed by the patient using an electronic pen on a digital tablet. . and identified the parameters of the studied neurosystems with feedback.*

*Keywords: neurological movements, tremor, mathematical modeling, cognitive feedback, abnormal movement, graphics tablet, electroencephalogram, diagnosis, Fourier-hybrid transform, hardware and software, computer simulation.*

### Постановка проблеми

У силу того, що наразі відсутні методи прогресивного та швидкого тестування та аналізу даних для діагностування захворювань на тремор, зацікавленість до систем даного типу зі сторони медичних закладів є досить високою. Також системи даного спрямування дають розширені можливості для оцінки отриманих результатів діагностики не тільки в межах даного захворювання, а й стану здоров'я пацієнта загалом.

З метою подання математичного розв'язку у формі реалізації процедури функціональної ідентифікації амплітудних компонентів і фазової швидкості поширення хвилі аномального невралгічного руху (АНР) як функції часу в рамках декомпозиції моделі з урахуванням умов, що відомі сліди (дані спостереження у формі цифрових даних АНР-рухів пацієнта) розв'язку для кожного сегмента з побудовою систем початково-крайових задач (мікромоделей) для послідовних сегментів кривої руху.

### Аналіз стану досліджень проблеми аномальних неврологічних рухів

Новітні інформаційні технології та методи моделювання в розробці комп'ютерної діагностичної системи покращують вирішення проблеми лікування критичних захворювань у світі, особливо людей, уражених неврологічними захворюваннями, такими як аномальні неврологічні рухи (АНР) або тремтіння та їх екстремальні форми, у вигляді хвороб Альцгеймера, Паркінсона [1]. АНР - небажані коливальні рухи певної частини тіла (рук, органів мови, очних яблук), що виникають внаслідок мимовільного скорочення м'язів людини [2]. Ознаками порушень регуляції рухів людини є збільшення їх амплітуди, зміна частоти і форми коливань. Аналіз цих параметрів АНР має вирішальне значення для розуміння ролі дисфункції зворотного зв'язку (feedback) в нейронних вузлах кори головного мозку (КГМ) у процесах когнітивного контролю рухів людини та раннього виявлення нейромоторних розладів. Складність ідентифікації АНР полягає в недосконалості існуючих методів діагностики, їх низькій точності, а також у відсутності математичних і програмних засобів ідентифікації нервово-зворотного впливу КГМ-вузлів на їх поведінку [2]. Дослідження нейросистем, пов'язані з аналізом поведінки пацієнтів із симптомами тремору (Т-об'єкти), проводилися рядом дослідників, таких як Pullman S.L., Legrand A.-P., Vidailhet M. (ESPCI Paris Tech, ICeM CNRS), Ван Дж.-С., Луї Е., Хаубенбергер Д., Каловіц Д. та інші. [2-5]. Тут основну увагу було приділено аналізу параметрів відносно нормального стану та поведінки пацієнтів за допомогою класичних методів цифрової обробки на основі перетворення Фур'є [5-12]. Однак такі методи на сьогодні вже вичерпали себе і не дозволяють аналізувати аномальні стани зі складною, важкопрогнозованою поведінкою, яка притаманна реальним Т-об'єктам з високим ступенем тремору. Через недосконалість таких методів відбувається втрата 60-80% важливої інформації з опису реального стану пацієнтів, що де-факто зумовлює низьку якість такого аналізу.

### Мета дослідження

Для конфігурації моделі ідентифікації використовували фрагменти траєкторії АНР спірального типу досліджуваного Т-об'єкта, здійснювану ним на цифровому графічному планшеті. Моделювання та ідентифікацію параметрів рухів АНР проводили в рамках завдання ідентифікації параметрів когнітивних feedback- впливів ЕЕГ-вектора сигналів КГР на сегменти траєкторії АНР, розробленого за допомогою гібридної моделі АНР. Для побудови ідентифікаційної моделі використовувався фрагмент сліду АНР досліджуваного Т-об'єкта, виконаний ним електронним пером на інтерактивному цифровому планшеті.

Метою даного дослідження є розробка гібридної аналітичної моделі, яка забезпечуватиме глибоку декомпозицію системи без порушення її цілісності та зв'язків, що не здійснюється класичними методами обробки сигналів, що приводять до втрати 60-80% інформації про реальний стан об'єкту. Методами машинного навчання необхідно попередньо уточнити вагові коефіцієнти впливу наборів цифрових записів сигналів когнітивних впливів нейровузлів кори головного мозку пацієнта у ході виконання ним тестових прикладів рисування спіральних траєкторій на цифровому планшеті.

#### **Пропоновані підходи авторів до вирішення проблеми аналізу АНР під дією зворотних когнітивних впливів основного дослідження**

Авторами запропоновано високопродуктивну інформаційну технологію дослідження АНР, побудовану на основі гібридної моделі аналізу сигналів нейросистеми, яка описує стан і поведінку 3D елементів траєкторій АНР Т-об'єкта з урахуванням когнітивні ефекти нервового зворотного зв'язку ідентифікованих вузлів КГМ з використанням комп'ютерного тестування рухів Т-об'єкта, що робить спроби відтворення шаблонних тестових траєкторій (у вигляді спіралі Архімеда та ін. тестів). За допомогою методів гібридних перетворень Фур'є побудовано швидкісні аналітичні розв'язки моделі аналізу АНР у вигляді векторних функцій, які визначають елементи тестових траєкторій рухів Т-об'єкта на кожному сегменті АНР [13-16]. На їх основі запропоновано високопродуктивні алгоритми ідентифікації параметрів АНР для покомпонентної оцінки ефектів нейро-зворотного зв'язку, які дозволяють розпаралелювати обчислення.

**Спосіб збору даних АНР за допомогою графічного планшета.** Якісне оцінювання характеристик АНР шляхом та кількісного покращення схем комп'ютерного тестування АНР вимагає знання точного положення електронного пера Т-об'єкта, що проходить тест на цифровому планшеті та тиску пера на поверхню планшета як функції часу впродовж усієї тривалості експерименту.

Для визначення координат  $X$  і  $Y$ , та  $Z$  як функцій часу цього необхідне використання графічного планшета зі спеціально адаптованим програмним забезпеченням, що забезпечує високу частоту і точність збору даних. З цих причин був обраний графічний планшет HUION KAMVAS PRO 16. Його активна зона (відповідає 345 x 194 мм) сумісна із загальноприйнятими моделями шкали оцінки тремору Фана-Толози-Маріна (FTRS) [4]. Рідер має роздільну здатність введення пера 5080 lpi, точність 0,25 мм і швидкість розпізнавання до 266 точок в секунду (згідно зі специфікаціями виробника). Крім того, даний планшет дозволяє відображати тиск пера під час малювання на площині та вимірювати рухи на відстані до 10 мм над поверхнею пера, що дозволяє візуалізувати рухи пацієнта в просторі та силу натиснення пера на поверхню планшета.



**Рис 1.** Візуальне представлення використання графічного планшета HUION KAMVAS PRO 16 в тесті

Для дослідження електроенцефалографічних (ЕЕГ) сигналів КГМ обрано комп'ютерний електроенцефалограф НЕЙРОКОМ – п'яте покоління розроблених комплексів



комп'ютерної електроенцефалографії. Енцефалограф в комплексі призначений для реєстрації, поглибленого аналізу та інтерпретації ЕЕГ і викликаних потенціалів (ВП), проведення різноманітних аналізів для наукових досліджень.

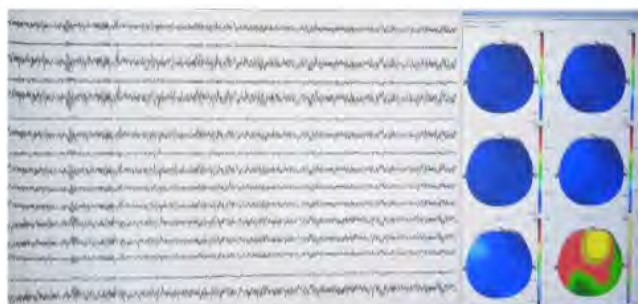


Рис 2. Візуальне представлення процесу збору даних від електроенцефалографічного комплексу NeuroCom

Шолом для аналізу когнітивних сигналів кори головного мозку із встановленою апаратно-програмною платформою виробника забезпечує 16-канальний відбір енцефалограм і передачу їх на персональний комп'ютер за відповідним протоколом. Програмне кондиціювання сигналів ЕЕГ і постобробка відбувається на ПК. Дані зберігаються як у необробленому тексті, так і у візуалізованому представленні в кожен момент часу. Дані зчитуються з кожного каналу відведення з інтервалом 2 мс (частота 500 Гц).

Гібридна модель забезпечує кількісні амплітудно-частотні характеристики АНР. Для врахування когнітивних feedback-впливів на поведінку АНР системи використовуються отримані цифрові набори індикаторів ЕЕГ-сигналів, які синхронно з рухом електронного пера Т-об'єкта надходять від 16-ти елементного вектора часових послідовностей сигналів, що поступають від ЕЕГ-сенсорів, встановлених на визначених на нейровузлах КГМ. Вектор ЕЕГ-сигналів в цілому визначає динаміку АНР для кожного  $j$ -го сегмента складної АНР- траси, де  $j=1, n1$  – номер поточного сегмента АНР,  $n1$  – кількість сегментів розбиття усїєї траси АНР (рис. 4). У моделі розбиття може бути встановлено автоматично будь-яким способом, з будь-якою кінцевою кількістю сегментів. Їх довжина також може бути різною залежно від вибору рівня деталізації ділянок руху. Слід зауважити, що згідно концепції запропонованої гібридної моделі, когнітивні feedback-впливи вектор ЕЕГ-сигналів впливають тільки на поточні сегменти АНР- траси, що синхронізовані по часу проходження зі значеннями компонентів ЕЕ-вектора та обумовлюють визначену спадну еспоненціальну післядію на майбутні ділянки коливного руху (на пройдені ділянки руху ці впливи відсутні, що природньо).



Рис 3. Ілюстрація використання дослідницького обладнання 3. Гібридна математична модель аналізу Т-об'єкта АНР з урахуванням когнітивних нейро-зворотних впливів вузлів КГМ

Гібридний модельний аналіз АНР, побудований на основі концепції поширення хвильового сигналу, визначає поелементний опис елементів траєкторій АНР з урахуванням матриці когнітивних feedback-впливів вектора сигналів нейровузлів КГМ на сегментах АНР [13]. В основі реалізації лежить метод визначення положення електронного пера, з допомогою якого Т-об'єкт відтворює траєкторію руху шаблону (спіралі Архімеда) на екрані інтерактивного планшета [14].

Visualisation de l'influence des composants du vecteur de neurosignaux  $S(t)$  sur les éléments de MNA-trace  $(l_{j-1}, l_j), j = 1, n_1$

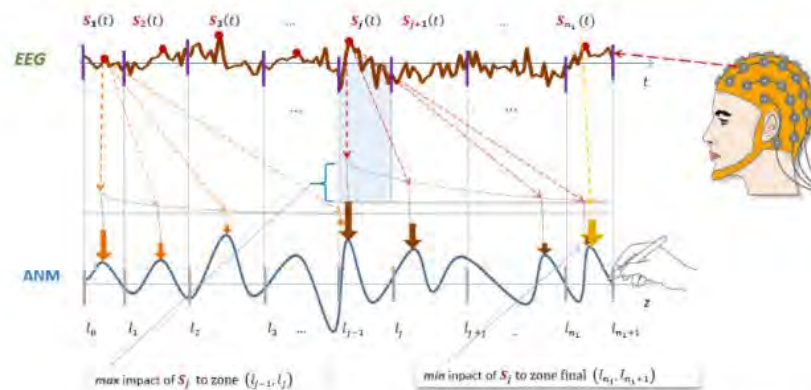


Рис. 4. Схематизація та візуалізація поточових когнітивних feedback-впливів компонентів вектора зв'язків EEG-сигналів нейровузлів КГМ на окремі елементи АНР-траєкт  $(l_{j-1}, l_j), j = 1, n_1$  в рамках гібридної моделі АНР-аналізу

Відхилення траєкторії руху пера від шаблону має складну форму (рис. 4) і дає оцифровану інформацію для визначення неврологічного стану Т-об'єкта. Слід руху пера розбивається на простіші елементи з метою декомпозиції складних АНР-рухів у схематизації моделі. Кількість поділок залежно від складності зображення АНР можна вибирати довільно.

#### Опис математичної моделі функціональної ідентифікації АНР

Для представлення математичного розв'язку у вигляді реалізації процедури функціональної ідентифікації амплітудних складових і фазової швидкості розповсюдження хвилі АНР як функцій часу в рамках декомпозиції моделі, враховуючи умови, що відомі траси (дані спостереження у вигляді цифрових даних АНР-рухів Т-об'єкта) розв'язку для кожного  $k$ -го сегмента, отримана наступна система прямих початково-крайових задач (мікромоделей) для послідовних сегментів АНР [13]:

$$\frac{\partial^2}{\partial t^2} u_k(t, z) = b_k^2 \frac{\partial^2}{\partial z^2} u_k + S_k^*(t, z) \quad (1)$$

з початковими умовами:

$$u_k(t, z)|_{t=0} = 0, \quad \frac{\partial u_k}{\partial t}|_{t=0} = 0, \quad k = \overline{1, n_1 + 1} \quad (2)$$

Крайові умови на кожному з тонких сегментів АНР по  $z$ :

$$u_{k-1}(t, z)|_{z=l_{k-1}} = U_{l_{k-1}}, \quad u_k(t, z)|_{z=l_k} = U_{l_k}, \quad k = \overline{1, n_1 + 1} \quad (3)$$

**Вибір функціонала-нев'язки.** Вважаємо, що компоненти фазової швидкості поширення хвилі АНР  $b_k, k = \overline{1, n_1 + 1}$  крайової задачі (3.3)-(3.5) є невідомими функціями від часу. При відомих значеннях положення пера  $u_k(t, z)$  в точках спостереження на сегментах АНР  $\gamma_k \subset \Omega_k, k = \overline{1, n_1 + 1}$ :

$$u_k(t, z)|_{\gamma_k} = U_k(t, z)|_{\gamma_k}, \quad (4)$$

де початково-крайова задача (6.1)-(6.4) може бути розглянута для кожної точки  $z$  для кожного тонкого  $k_i$ -го сегмента праси АНР і полягатиме в знаходженні функцій

$$b_k \in D, \text{ де } D = \left\{ v(t, z): v|_{\Omega_{k_i}} \in C(\Omega_{k_i}), v > 0, k = \overline{1, n_1 + 1} \right\}.$$

Функціонал-нев'язка відхилення розв'язку від його слідів на  $\gamma_{k_i} \in \Omega_{k_i}$ , згідно [14] запишеться у вигляді:

$$J_k(b_k) = \frac{1}{2} \int_0^T \|u_k(t, z, b_k) - U_k^*\|^2 dt. \quad (5)$$

**Методика розв'язання прямої крайової задачі ідентифікації.** Побудова та математичне обґрунтування розв'язку задачі здійснюється шляхом використання скінченного інтегрального перетворення Фур'є. Застосувавши до задачі (6.1)-(6.4) інтегральні оператори [13]:

$$\begin{aligned} F[u_k(t, z)] &= \int_{k-1}^{k} u_k(t, z) V_m(\beta_m, z) dz \equiv U_{km}(t) \\ F^{-1}[U_{km}(t)] &= \sum_{m=0}^{\infty} U_{km}(t) \frac{V_m(\beta_m, z)}{\|V_m(\beta_m, z)\|^2} \equiv u_k(t, z) \end{aligned} \quad (6)$$

$$F\left[\frac{\partial^2}{\partial z^2} u_k(t, z)\right] = -\beta_m^2 U_{km}(t) + \beta_m U_{k_{i-1}}, \left[1 - (-1)^m \frac{U_{k_i}}{U_{k_{i-1}}}\right] = -\beta_m^2 U_{km}(t) + \beta_m U_{k_{i-1}} - \beta_m (-1)^m U_{k_i}$$

$$V_k(\beta_m, z) = \sin \beta_m (z - l_{k-1}) \quad \beta_m = \frac{m\pi}{\Delta h} \quad \|V_k\|^2 = \int_{l_{k-1}}^{l_k} [V_k(\beta_m, z)]^2 dz = \frac{\Delta h}{2} \quad \sin \beta(l_k - l_{k-1}) = 0$$

Розв'язок задачі (6.1)-(6.4) після низки перетворень одержано у формі зручній і ефективній для числових ітераційних розрахунків для використання в процедурах ідентифікації параметрів:

$$u_k(t, z) = \frac{2}{\Delta l} \sum_{m=0}^{\infty} \frac{1 - \cos(b_k \beta_m t)}{\beta_m} \sin \beta_m (z - l_{k-1}) \left( S_k^* \frac{1}{(b_k \beta_m)^2} ((-1)^m - 1) + U_{k_{i-1}} \left( 1 - (-1)^m \frac{U_{k_i}}{U_{k_{i-1}}} \right) \right) \quad (7)$$

Метод розв'язання спряженої крайової задачі АНР

У відповідності з вихідною прямою початково-крайовою задачею (1.1)-(1.5), слідуючи [14] для кожного наближення розв'язку, вводимо до розгляду спряжену часово-крайову задачу:

$$\frac{\partial^2}{\partial t^2} \phi_k(t, z) + b_k \frac{\partial^2}{\partial z^2} \phi_k(t, z) = (u_{k_i}^* - U_k^*)|_{z=\gamma_k} \delta(z - \gamma_k), \quad k = \overline{1, n_1 + 1} \quad (8)$$

$$\text{умовами при } t = T \quad \phi_k(t, z)|_{t=T} = 0 \quad (9)$$

та крайовими умовами першого роду для кожного наближення  $D_{\varphi k}^n$ , розв'язку:

$$\phi_k(t, z)|_{z=l_{k-1}} = 0; \quad \phi_k(t, z)|_{z=l_k} = 0, \quad (10)$$

Аналітичний розв'язок спряженої часово-крайової задачі (6.10)-(6.10) отримуємо у вигляді [14]:

$$\phi_k(t, z) = \int_{l_{k-1}}^{l_k} \int_0^T \bar{H}_k(t - \tau, z, \xi) (u_{k_i}^* - U_k^*)|_{z=\gamma_k} \delta(z - \gamma_k) d\xi d\tau, \quad (11)$$

де:

$$\bar{H}_k(t - \tau, z, \xi) = \frac{2}{\Delta h} \sum_{m=0}^{\infty} \frac{\text{sh } b_k \beta_m (t - \tau)}{b_k \beta_m} \sin \beta_m (\xi - l_{k-1}) \sin \beta_m (z - l_{k-1}).$$

Розв'язок (11) спряженої задачі (8)-(9) після інтегрування і низки перетворень конвертуємо у формі зручній і ефективній для числових ітераційних розрахунків для використання в процедурах ідентифікації параметрів:



$$\phi_k(t, z) = \frac{2}{\Delta h} \sum_{m=0}^{\infty} \frac{1 - \operatorname{ch}(b_k \beta_m (T-t))}{(b_k \beta_m)^2} \sin \beta_m \gamma_k \sin \beta_m (z - l_{k-1}) (U_k^* - u_k^n), \quad k = \overline{1, n_1 + 1} \quad (12)$$

Формули аналітичних виразів градієнтів функціонала-нев'язки

Вирази компонентів градієнтів функціонала-нев'язки для необхідних коефіцієнтів  $b_k$ :

$$\nabla J_{b_k} = \int_0^T \int_{l_{k-1}}^{l_k} \phi_k(t, z) \frac{\partial^2}{\partial z^2} u_k(t, z) dz dt \quad (13)$$

Для задач функціональної ідентифікації формули для компонентів градієнтів функціоналу невязки як функції від часу або координати, отримуємо, знімаючи відповідний інтеграл по  $t$  чи  $z$  у (13)

$$\nabla J_{b_k}(t) = \int_{l_{k-1}}^{l_k} \phi_k(t, z) \frac{\partial^2}{\partial z^2} u_k(t, z) dz, \quad (14)$$

$$\nabla J_{b_k}(z) = \int_0^T \phi_k(t, z) \frac{\partial^2}{\partial z^2} u_k(t, z) dt. \quad (15)$$

Тут подані точні аналітичні вирази компонентів підінтегрального виразів (14), (15):

$$\phi_k(t, z) = \frac{2}{\Delta h} \sum_{m=0}^{\infty} \frac{1 - \operatorname{ch}(b_k \beta_m (T-t))}{(b_k \beta_m)^2} \sin \beta_m \gamma_k \sin \beta_m (z - l_{k-1}) (U_k^* - u_k^n), \quad k = \overline{1, n_1 + 1}$$

$$\frac{\partial^2}{\partial z^2} u_k(t, z) = -\frac{2}{\Delta l} \sum_{m=0}^{\infty} \beta_m (1 - \cos(b_k \beta_m t)) \sin \beta_m (z - l_{k-1}) \left( S_k^* \frac{1}{(b_k \beta_m)^2} ((-1)^m - 1) + U_{l_{k-1}} \left( 1 - (-1)^m \frac{U_{l_k}}{U_{l_{k-1}}} \right) \right)$$

**Регуляризаційні вирази** для  $n+1$ -го кроку визначення ідентифікуючої залежності. З використанням методу мінімальних похибок для визначення залежності функціональної ідентифікації компоненти амплітудного коефіцієнта коливного руху сигналу  $b_k^{n+1}$  як константу (параметр) на відрізку або у вигляді функціональної залежності від часу  $t$  чи  $z$ , або від двох змінних для кожного  $k$ -го елемента АНР  $k = \overline{1, n_1 + 1}$ . Зокрема, для функціональної залежності від часу  $t$  маємо такий регуляризаційні вирази:

$$b_k^{n+1}(t) = b_k^n(t) - \nabla J_{b_k}^n(t) \frac{\|u_k(t, \gamma_k, b_k) - U_k^*\|^2}{\|\nabla J_{b_k}^n(t)\|_{l_k}^2}, \quad t \in (0, T), \quad k = \overline{1, n_1}. \quad (16)$$

Матричний алгоритм ідентифікації вектора адаптивних коефіцієнтів впливу матриці когнітивних EEG сигналів на вектор сегментів траєкторії АНР

Формула зв'язку вектор сегментів траєкторії коливного руху електронного пера  $[u_j(t)]$ ,  $j = \overline{1, n+1}$  з матрицею матриці сигналів сенсорів EEG  $[S_{ij}(t)]$ ,  $i = \overline{1, m}$ ;  $j = \overline{1, n+1}$ , де  $m$  – кількість сенсорів EEG (в нашому випадку  $m = 16$ ),  $n+1$  – загальна кількість сегментів траєкторії руху пера вздовж координати  $z$ :

$$\begin{bmatrix} u_1(t, z) \\ u_2(t, z) \\ \dots \\ u_{n+1}(t, z) \end{bmatrix} = \begin{bmatrix} S_{11}(t) & S_{12}(t) & \dots & S_{1m}(t) \\ S_{21}(t) & S_{22}(t) & \dots & S_{2m}(t) \\ \dots & \dots & \dots & \dots \\ S_{n+1,1}(t) & S_{n+1,2}(t) & \dots & S_{n+1,m}(t) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{bmatrix}. \quad (17)$$

Звідси знаходимо:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{bmatrix} = \left( \begin{bmatrix} S_{11}(t) & S_{12}(t) & \dots & S_{1m}(t) \\ S_{21}(t) & S_{22}(t) & \dots & S_{2m}(t) \\ \dots & \dots & \dots & \dots \\ S_{n+1,1}(t) & S_{n+1,2}(t) & \dots & S_{n+1,m}(t) \end{bmatrix}^T \right)^{-1} \begin{bmatrix} u_1(t, z) \\ u_2(t, z) \\ \dots \\ u_{n+1}(t, z) \end{bmatrix},$$

Або:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} S_{11}(t) & S_{12}(t) & \dots & S_{1,n+1}(t) \\ S_{21}(t) & S_{22}(t) & \dots & S_{2,n+1}(t) \\ \dots & \dots & \dots & \dots \\ S_{m,1}(t) & S_{m,2}(t) & \dots & S_{m,n+1}(t) \end{bmatrix}^{-1} \begin{bmatrix} u_1(t, z) \\ u_2(t, z) \\ \dots \\ u_{n+1}(t, z) \end{bmatrix}. \quad (18)$$

Перемножуючи в правій частині (18) елементи матриць за правилами множення, отримуємо вектор значень коефіцієнтів адаптивного впливу  $[S_{ij}(t)]$ ,  $i = \overline{1, m}$ ;  $j = \overline{1, n+1}$  на  $[u_j(t)]$ ,  $j = \overline{1, n+1}$ :

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \tilde{S}_{11}(t) \cdot u_1(t, z) + \tilde{S}_{12}(t) \cdot u_2(t, z) + \dots + \tilde{S}_{1,n+1}(t) \cdot u_{n+1}(t, z) \\ \tilde{S}_{21}(t) \cdot u_1(t, z) + \tilde{S}_{22}(t) \cdot u_2(t, z) + \dots + \tilde{S}_{2,n+1}(t) \cdot u_{n+1}(t, z) \\ \dots \\ \tilde{S}_{m,1}(t) \cdot u_1(t, z) + \tilde{S}_{m,2}(t) \cdot u_2(t, z) + \dots + \tilde{S}_{m,n+1}(t) \cdot u_{n+1}(t, z) \end{bmatrix}, \quad (19)$$

де  $\tilde{S}_{ij}(t)$ ,  $i = \overline{1, m}$ ;  $j = \overline{1, n+1}$  - елементи оберненої матриці  $(s^T)^{-1}$ , індекс  $T$  - знак транспонування матриці.

### Висновки

Інформаційну технологію розроблено на основі гібридної моделі АНР-аналізу нейро-системи, яка описує стан і поведінку Т-об'єктів з урахуванням когнітивних feedback- впливів КГМ. За допомогою методу гібридних перетворень Фур'є побудовано аналітичний розв'язок моделі у векторному вигляді, визначає елементи траєкторій на кожному сегменті АНР. На основі цього запропоновано високоефективні алгоритми ідентифікації параметрів досліджуван-них систем зворотного зв'язку. На відміну від загальноприйнятого класичного статистичного підходу обробки сигналів, запропонована гібридна модель орієнтована на глибоку декомпози-цію системи без порушення її цілісності та всіх важливих зв'язків. Крім того, завдання цього спостереження полягає в тому, щоб ретельно перевірити механізми та вдосконалити методи збору та аналізу даних, отриманих із спіральних малюнків пацієнта. Такий підхід дає можли-вість більш якісно описати складні приховані механізми АНР з великою кількістю внутрішніх зв'язків і когнітивних feedback- впливів КГМ, забезпечити високий ступінь повноти даних.

### Список використаної літератури:

1. Electroencephalography complex NEUROKOM, NEUROLAB. Instructions for medical application AINC.941311.001 П U 33.1-02066769-001-2002.
2. Haubenberger D., Kalowitz D., Nahab F. B., Toro C., Ippolito D., Luckenbaugh D.A., Wittevrongel L., Hallett M. Validation of Digital Spiral Analysis as Outcome Parameter for Clinical Trials in Essential Tremor. *Movement Disorders*, 26 (11), 2073-2080, (2011).
3. Rajaraman V., Jack D., Adamovich S.V., Hening W., Sage J., Poizner H. A novel quantitative method for 3D measurement of Parkinsonian tremor. *Clinical neurophysiology*, 11(2), 187-369 (2000).
4. Legrand A.P., Rivals I., Richard A., Apartis E., Roze E., Vidailhet M., Meunier S., Hainque E. New insight in spiral drawing analysis methods – Application to action tremor quantification. *J Clinical Neurophysiology*, 128 (10), 1823–1834, (2017).
5. Wang J.-S., Chuang F.-C. An Accelerometer-Based Digital Pen with a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition. *IEEE Transactions on Industrial Electronics*, 59(7), 2998-3007 (2012).
6. Louis E. D., Gillman A., Böschung S., Hess C. W., Yu Q., Pullman S. L. High width variability during spiral drawing: Further evidence of cerebellar dysfunction in essential tremor. *Cerebellum*, 11, 872-879 (2012).

7. Xie H., Wang Z. Mean frequency derived via Huang-Hilbert transform with application to fatigue EMG signal analysis. *Comput Meth Progr Biomed*, 2006; 82:114–20.
8. Bhidayasiri R., Mari Z. Digital phenotyping in Parkinson's disease: Empowering neurologists for measurement-based care. *Parkinsonism Relat Disord*. 2020 Nov;80:35-40. DOI: 10.1016/j.parkreidis.2020.08.038, (2020).
9. Viviani P., Burkhard P.R., Chiuvè S.C., dell'Acqua C.C., Vindras P. Velocity control in Parkinson's disease: a quantitative analysis of isochrony in scribbling movements. *Exp Brain* 2009;194:259–83, (2009)
10. Salarian A., Russmann H., Wider C., Burkhard P.R., Vingerhoets F.J., Aminian K. Quantification of tremor and bradykinesia in Parkinson's disease using a novel ambulatory monitoring system. *Biomedical Engineering, IEEE Transactions on*, 54. Jg., Nr. 2, pp. 313- 22, (2007).
11. Lo G., Suresh A. R., Stocco L., González-Valenzuela S., and Leung V. C. A wireless sensor system for motion analysis of Parkinson's disease patients, (PERCOM Workshops), 2011 IEEE International Conference on. IEEE, pp. 372-375, (2011).
12. Petryk M., Gancarczyk T., Khimich O. Methods of Mathematical Modeling and Identification of Complex Processes and Systems on the basis of High-performance Calculations (neuro- and nanoporous feedback cyber systems, models with sparse structure data, parallel computations). Scientific Publishing University of Bielsko-Biala.:Bielsko-Biala, Polska), (2021), 194 p.
13. Khimich A.N., Petryk M.R., Mykhalyk D.N., Boyko I.V., Popov A.V., Sydoruk. V.A. Methods for mathematical modeling and identification of complex processes and systems based on visoproductive computing (neuro- and nanoporous cyber-physical systems with feedback, models with sparse structure data, parallel computing). Monograph, Kiev: National Academy of Sciences of Ukraine. Glushkov Institute of Cybernetics. 2019. - 176 p. ISBN: 978-966-02-9188-1, (2019).
14. Mudryk I., Petryk M. Hybrid artificial intelligence systems for complex neural network analysis of abnormal neurological movements with multiple cognitive signal nodes. 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP) : Conference, Lviv, 21-25 August 2020. P. 108–111, (2020).

#### References

1. Electroencephalography complex NEUROKOM. NEUROLAB. Instructions for medical application AINC.941311.001 11 U 33.1-02066769-001-2002.
2. Haubenberger, D., Kalowitz, D., Nahab, F. B., Toro, C., Ippolito, D., Luckenbaugh, D.A., Wittevrongel, L., & Hallett, M. (2011). Validation of Digital Spiral Analysis as Outcome Parameter for Clinical Trials in Essential Tremor. *Movement Disorders*, **26** (11), 2073-2080.
3. Rajaraman, V., Jack, D., Adamovich, S.V., Hening, W., Sage, J., & Poizner, H. (2000). A novel quantitative method for 3D measurement of Parkinsonian tremor. *Clinical neurophysiology*, **11** (2), 187-369.
4. Legrand, A.P., Rivals, I., Richard, A., Apartis, E., Roze, E., Vidailhet, M., Meunier, S., & Hainque, E. (2017). New insight in spiral drawing analysis methods. Application to action tremor quantification. *J Clinical Neurophysiology*, **128** (10), 1823–1834.
5. Wang, J.-S., & Chuang, F.-C. (2012). An Accelerometer-Based Digital Pen with a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition. *IEEE Transactions on Industrial Electronics*, **59** (7), 2998-3007.
6. Louis, E. D., Gillman, A., Böschung, S., Hess, C. W., Yu, Q., & Pullman, S. L. (2012). High width variability during spiral drawing: Further evidence of cerebellar dysfunction in essential tremor. *Cerebellum*, **11**, 872-879.
7. Xie, H., & Wang, Z. (2006). Mean frequency derived via Huang-Hilbert transform with application to fatigue EMG signal analysis. *Comput Meth Progr Biomed*; 82:114–20

8. Bhidayasiri, R., & Mari, Z. (2020). Digital phenotyping in Parkinson's disease: Empowering neurologists for measurement-based care. *Parkinsonism Relat Disord.* 2020 Nov; 80:35-40. DOI: 10.1016/j.parkreldis.2020.08.038.
9. Viviani, P., Burkhard, P.R., Chiuvé, S.C., dell'Acqua, C.C., & Vindras, P. (2009). Velocity control in Parkinson's disease: a quantitative analysis of isochrony in scribbling movements. *Exp Brain* 2009;194:259–83.
10. Salarian, A., Russmann, H., Wider, C., Burkhard, P.R., Vingerhoets, F.J., & Aminian, K. (2007). Quantification of tremor and bradykinesia in Parkinson's disease using a novel ambulatory monitoring system, *Biomedical Engineering, IEEE Transactions on*, 54, Jg., (2), 313-322.
11. Lo, G., Suresh, A. R., Stocco, L., González-Valenzuela, S., & Leung V. C. (2011). A wireless sensor system for motion analysis of Parkinson's disease patients, (PERCOM Workshops), *2011 IEEE International Conference on. IEEE*, 372-375.
12. Petryk, M., Gancarczyk, T., & Khimich, O. (2021). *Methods of Mathematical Modeling and Identification of Complex Processes and Systems on the basis of High-performance Calculations (neuro- and nanoporous feedback cyber systems, models with sparse structure data, parallel computations)*. Scientific Publishing University of Bielsko-Biala.:Bielsko-Biala, Polska.
13. Khimich, A.N., Petryk, M.R., Mykhalyk, D.N., Boyko, I.V., Popov, A.V., & Sydoruk, V.A. (2019). *Methods for mathematical modeling and identification of complex processes and systems based on visoproductive computing (neuro- and nanoporous cyber-physical systems with feedback, models with sparse structure data, parallel computing)*. Monograph, Kiev: National Academy of Sciences of Ukraine. Glushkov Institute of Cybernetics.
14. Mudryk, I., & Petryk, M. (2020). Hybrid artificial intelligence systems for complex neural network analysis of abnormal neurological movements with multiple cognitive signal nodes. *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP) : Conference, Lviv, 21-25 August 2020*, 108–111.

Петрик Михайло Романович – д.ф.-м.н., професор кафедри програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя, e-mail: Mykhaylo\_petryk@tntu.edu.ua, ORCID: 0000-0001-6612-7213,

Мудрик Іван Ярославович – PhD, старший викладач кафедри програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя, e-mail: ilmudryk@ukr.net, ORCID: 0000-0002-4305-1911,

Бачинський Михайло Володимирович – к.т.н., доцент кафедри біотехнічних систем Тернопільського національного технічного університету імені Івана Пулюя, e-mail: m.bachynskyy@gmail.com, ORCID: 0000-0003-4139-7633.

Стадник Ігор Ярославович – д.т.н., проф. кафедри обладнання харчових технологій Тернопільського національного технічного університету імені Івана Пулюя, e-mail: igorstadnykk@gmail.com, ORCID: 0000-0003-4126-3256.

Підгурський Микола Іванович – д.т.н., проф. кафедри будівельної механіки Тернопільського національного технічного університету імені Івана Пулюя, e-mail: kaf\_zv@tu.edu.ua, ORCID: 0000-0001-8831-4810.

Ямко Владислав Олексійович - студент кафедри програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя, спеціальність 121 – інженерія програмного забезпечення, e-mail: vebleb01@gmail.com .

## ДОДАТОК Б – Лістинг коду інформаційної системи

## Лістинг коду 1 – Код відображення графіків

```

import React, { useLayoutEffect, useRef } from 'react';
import { Chart as ChartJS, ChartDataset, ChartOptions, ChartTypeRegistry,
registerables } from 'chart.js';
import { merge } from 'lodash';

export type Coordinate = {
  x: number;
  y: number;
};

ChartJS.register(...registerables);

export type ChartOptionProps = ChartOptions<keyof ChartTypeRegistry>;
export type ChartDatasetProps<T> = ChartDataset<keyof ChartTypeRegistry,
T>;

const Chart = <T,>(props: {
  label?: string;
  name: string;
  coordinates?: Coordinate[];
  options?: ChartOptionProps;
  datasets?: ChartDatasetProps<T>[];
}) => {
  const { coordinates, name, options, label, datasets = [] } = props;
  const ref = useRef<HTMLDivElement | null>(null);

  useLayoutEffect(() => {
    if (!ref.current) return;
    while (ref.current.firstChild) {
      ref.current.removeChild(ref.current.firstChild);
    }
    const canvas = document.createElement('canvas');
    canvas.setAttribute('id', 'name_' + Math.random());
    canvas.setAttribute('width', '400');
    canvas.setAttribute('height', '400');
    ref.current.appendChild(canvas);
    new ChartJS(canvas, {
      type: 'scatter',
      data: {
        datasets: [
          coordinates
            ? {
              label,
              data: coordinates as any,
              backgroundColor: 'red',
              borderColor: 'red',
              pointRadius: 0.4,
            }
          : []
        ]
      }
    });
  }, [ref, coordinates, label, datasets]);
};

```

```

        type: 'line',
        borderWidth: 0.2,
        borderDash: [5, 5],
      }
      : (undefined as any),
      ...datasets,
    ].filter(Boolean),
  },
  options: merge({ interaction: { mode: 'nearest' } }, options),
});
}, [coordinates, label, name, options, datasets]);

return <div ref={ref} />;
};

export default Chart;

```

## Лістинг коду 2 – Код розкладання вхідних даних

```

import React from 'react';
import { useTranslation } from 'react-i18next';
import Chart, { ChartOptionProps } from '../components/Chart';
import { TabletRecord } from 'graphql/types.generated';
import Width from '../components/Width';
import s from './Chart.module.sass';

const YAndTimeOptions: ChartOptionProps = {
  scales: {
    yAxisKey: { title: { text: 'y', display: true } },
    xAxisKey: { title: { text: 'time', display: true } },
  },
  parsing: { xAxisKey: 'time' },
};

const XAndTimeOptions: ChartOptionProps = {
  scales: {
    yAxisKey: { title: { text: 'x', display: true } },
    xAxisKey: { title: { text: 'time', display: true } },
  },
  parsing: { xAxisKey: 'time', yAxisKey: 'x' },
};

const PressureAndTimeOptions: ChartOptionProps = {
  scales: {
    yAxisKey: { title: { text: 'pressure', display: true }, beginAtZero: true },
    xAxisKey: { title: { text: 'time', display: true } },
  },
  parsing: { xAxisKey: 'time', yAxisKey: 'pressure' },
};

const DecomposeTabletChart = ({ tabletRecords }: { tabletRecords?: TabletRecord[] }) => {
  const { t } = useTranslation();

```

```

if (!tabletRecords) return null;
return (
  <>
  <div>
    <div className={s.half}>
      <Chart
        name="decomposeTabletByYAndTime"
        coordinates={tabletRecords}
        label={t('calculation.series.tablet')}
        options={YAndTimeOptions}
      />
    </div>
    <Width data={tabletRecords} parse={{ x: 'time', y: 'y' }} />
  </div>
  <div>
    <div className={s.half}>
      <Chart
        name="decomposeTabletByXAndTime"
        coordinates={tabletRecords}
        label={t('calculation.series.tablet')}
        options={XAndTimeOptions}
      />
    </div>
    <Width data={tabletRecords} parse={{ x: 'time', y: 'x' }} />
  </div>
  <div>
    <div className={s.half}>
      <Chart
        name="decomposeTabletByPressureAndTime"
        coordinates={tabletRecords}
        label={t('calculation.series.tablet')}
        options={PressureAndTimeOptions}
      />
    </div>
    <Width data={tabletRecords} parse={{ x: 'time', y: 'pressure' }} />
  </div>
</>
);
};

export default DecomposeTabletChart;

```

### Лістинг коду 3 – Код парсингу файлу ЕЕГ

```

import { parse } from 'date-fns';
import Papa from 'papaparse';
import { errors as BackendErrors } from '../constants/errors';
import { ElectromyographRecord } from '../classes/ElectromyographRecord';
import { ElectromyographPort, ElectromyographPortIndex } from
'../constants';
import { base64ToString, getStartDate } from './utils';

class ElectromyographCalculationService {

```

```

    parseTime(dateStr: string, referenceDate: Date) {
      const parsedDate = parse(dateStr, 'H:m:s.SSS', referenceDate);
      return parsedDate;
    }

    async normalizeData(base64: string, signalPort: ElectromyographPort):
    Promise<[ElectromyographRecord[], Date]> {
      const ssv = base64ToString(base64);
      const { data, errors }: { data: unknown[][]; errors: unknown[] } =
    Papa.parse(ssv, { delimiter: '\t' });

      const headers = data.filter((row) => row.length === 1).flat() as
    string[];
      const startDate = getStartDate(headers);
      const signalPortIndex = ElectromyographPortIndex[signalPort as keyof
    typeof ElectromyographPortIndex];

      if (!signalPortIndex) throw new
    Error(BackendErrors.UNAVAILABLE_SIGNAL_PORT);
      if (errors.length) throw new
    Error(BackendErrors.TABLET_FILE_INCORRECT_FORMAT);

      const records = data
      .map((row) => {
        const time = this.parseTime(row[0] as string, startDate);
        const signal = +(row[signalPortIndex] as number);
        if (Number.isFinite(time.getTime()) && Number.isFinite(signal)) {
          const record: ElectromyographRecord = {
            time,
            relativeTime: startDate.getTime() - time.getTime(),
            signal,
          };
          return record;
        }
      })
      .filter(Boolean) as ElectromyographRecord[];

      return [records, startDate];
    }
  }

export default new ElectromyographCalculationService();

```

#### Лістинг коду 4 – Код сервісу авторизації

```

import { DocumentType } from '@typegoose/typegoose';
import { JWTTokenParams, JWTTokenPayload, LoginUserProps,
LoginWithRefreshTokenProps } from './AuthService.types';
import jwt from 'jsonwebtoken';
import { AuthenticationError, ForbiddenError } from 'apollo-server-
express';
import SessionModel from '../models/SessionModel';

```



```

import UserModel from '../..//user/models/UserModel';
import SessionsService from './SessionsService';
import { TokenType } from './AuthService.types';
import { jwtOptionsByTokenType, jwtSecretByTokenType } from '../constants';
import { User } from '../..//user/classes/User';
import { errors } from '../..//constants/errors';

class AuthService {
  authenticateWithEmailAndPassword = async ({ email, password }:
Pick<LoginUserProps, 'email' | 'password'>) => {
    const userWithOnlyServices = await
UserModel.findByUsername(email).select('services');
    if (!userWithOnlyServices) {
      throw new Error(errors.INCORRECT_CREDENTIALS);
    }

    const auth = await userWithOnlyServices.authenticate(password);
    if (!auth.user) throw new Error(auth.error);

    const user = await UserModel.findByUsername(email);
    return user;
  };

  createUserSession = async ({ user, rememberMe, ip }: { user:
DocumentType<User>; rememberMe: boolean; ip: LoginUserProps['ip'] }) => {
    const userId = user.id;
    const session = await SessionModel.createSession({ userId, ip });
    const sessionId = session.id;
    const jwtPayload = { sub: userId, email: user.email, sessionId };
    const accessToken = this.generateToken(jwtPayload, TokenType.access);
    const refreshToken = rememberMe ? this.generateToken(jwtPayload,
TokenType.refresh) : undefined;
    return { accessToken, refreshToken };
  };

  login = async ({ email, rememberMe, password, ip }: LoginUserProps) => {
    const user = await this.authenticateWithEmailAndPassword({ email,
password });
    if (user.deactivated) throw new ForbiddenError('You are not allowed to
access this application');

    const { accessToken, refreshToken } = await this.createUserSession({
user, rememberMe, ip });
    return {
      accessToken,
      refreshToken,
      user,
    };
  };

  // rename this method
  loginWithRefreshToken = async ({ refreshToken, ip }:
LoginWithRefreshTokenProps) => {
    const tokenData = await this.verifyToken(refreshToken,
TokenType.refresh);

```

```

    await SessionModel.updateSession(tokenData.sessionId as string, { ip,
lastActivityDate: new Date() });

    const { sub, email, sessionId } = tokenData;
    return this.generateToken({ sub, email, sessionId }, TokenType.access);
};

logout = async (accessToken: string) => {
    await SessionsService.logout(accessToken);
};

getUnixExpirationOfToken(token: string) {
    const decoded = jwt.decode(token) as JWTTokenParams;
    if (!decoded || !decoded.exp) return null;
    return decoded.exp * 1000;
}

generateToken = (payload: JWTTokenPayload, type: TokenType) => {
    const secret = jwtSecretByTokenType[type];
    const options = jwtOptionsByTokenType[type];
    return jwt.sign(payload, secret, options);
};

verifyToken = async (token: string, type: TokenType = TokenType.access)
=> {
    const secret = jwtSecretByTokenType[type];
    const tokenData = jwt.verify(token, secret) as JWTTokenParams;
    const session = await SessionModel.findById(tokenData.sessionId);
    const skipSessionCheck = TokenType.resetPassword === type;
    if (!skipSessionCheck && !session) {
        throw new AuthenticationError(errors.TOKEN_EXPIRED);
    }
    return tokenData;
};

changePassword = async (token: string, currentPassword: string,
newPassword: string): Promise<boolean> => {
    const { email } = await this.verifyToken(token, TokenType.access);
    try {
        const user = await this.authenticateWithEmailAndPassword({ email,
password: currentPassword });
        await user.setPassword(newPassword);
        await user.save();

        return true;
    } catch (error: any) {
        if (error.message.includes(errors.INCORRECT_CREDENTIALS)) {
            throw new Error(errors.INCORRECT_PASSWORD);
        }
        throw error;
    }
};
}

export default new AuthService();

```

ДОДАТОК В – Диск із кваліфікаційною роботою бакалавра