

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення**

ДОПУЩЕНО ДО ЗАХИСТУ


Завідувач кафедри _____ Є. О. Давиденко
підпис

«___» _____ 20__ р.

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОБРОБКИ
ДАНИХ ДЛЯ ОЦІНЮВАННЯ РЕАБІЛІТАЦІЇ
ПОСТ-ІНСУЛЬТНИХ ПАЦІЄНТІВ
Спеціальність «Інженерія програмного забезпечення»**

121 – КРБ.1 – 408.21810826

Студент

 В. Ю. Фінік
підпис

«__» _____ 20__ р.

Керівник д-р техн. наук, проф.

_____ І. М. Журавська
підпис

«__» _____ 20__ р.

Консультант канд. техн. наук

_____ А. О. Алексєєва
підпис

«__» _____ 20__ р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є. О. Давиденко
« ____ » _____ 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи бакалавра

Видано студенту групи 408 факультету комп'ютерних наук
Фініку Владиславу Юрійовичу

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи
«Програмне забезпечення обробки даних для оцінювання реабілітації
пост-інсультних пацієнтів»

Затверджена наказом по ЧНУ від «01» грудня 2021 р. № 314

2. Строк представлення кваліфікаційної роботи « ____ » _____ 2022 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: _____

Очікуваний результат роботи: Програмне забезпечення обробки даних
для оцінювання реабілітації пост-інсультних пацієнтів

Початкові дані: спеціалізовані файли (Data Set), зібрані медичними

зкладами та/або медичними науковими установами, з показниками пост-
інсультних пацієнтів, які знаходяться на реабілітації

4. Перелік питань, що підлягають розробці _____

– огляд прикладної сфери;

– огляд вже існуючих аналогів систем, призначених для оцінювання
реабілітації пост-інсультних пацієнтів;

– вибір інструментів для реалізації застосунку;

– програмна реалізація застосунку.

5. Перелік графічних матеріалів

Презентація

6. Завдання до спеціальної частини
Аналіз робочого місця користувача за усіма вимогами техніки безпеки

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексеева А. О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи проф. кафедри комп'ютерної інженерії, д-р техн. наук, проф.
Журавська Ірина Миколаївна

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Фінік Владислав Юрійович

(прізвище, ім'я, по батькові студента)

Ф. Фінік
(підпис)

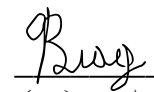
Дата видачі завдання « ____ » _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: «Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів»

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Затвердження завдання на виконання КРБ	16.01	18.01	Виконано
2.	Огляд літератури за темою роботи	18.01	22.01	Виконано
3.	Складання календарного плану КРБ	22.01	25.01	Виконано
4.	Аналіз існуючих аналогів	25.01	02.02	Виконано
5.	Вибір інструментів для реалізації застосунку	03.02	06.02	Виконано
6.	Аналіз характеристик необхідних для збору та аналізу	07.02	14.02	Виконано
7.	Реалізація візуалізації статистики	02.05	10.05	Виконано
8.	Тестування та формування документації	11.05	14.05	Виконано
9.	Розробка спеціальної частини з охорони праці	15.05	17.05	Виконано
10.	Відгук керівника КРБ	18.05	19.05	Виконано
11.	Оформлення КРБ та презентації	19.05	20.05	Виконано
12.	Попередній захист	13.06	15.06	Виконано
13.	Рецензування	21.06	23.06	Виконано
14.	Захист кваліфікаційної роботи	29.06	29.06	Виконано

Розробив студент Фінік Владислав Юрійович
(прізвище, ім'я, по батькові студента)


(підпис)

«___»_____ 2022 р.

Керівник роботи: д-р техн. наук, проф. Журавська Ірина Миколаївна
(прізвище, ім'я, по батькові)

(підпис)

«___»_____ 2022 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Програмне забезпечення обробки даних для оцінювання реабілітації
пост-інсультних пацієнтів»

Студент 408 гр.: Фіник Владислав Юрійович

Керівник: д-р техн. наук, проф. Журавська І. М.

Кваліфікаційна робота бакалавра присвячена створенню програмного забезпечення для оцінювання якості реабілітації пацієнтів після інсульту. Дана робота є актуальною через зменшення фізичної активності внаслідок технічного прогресу та через велику кількість стрес-факторів, спричинених початком повномасштабної війни.

Об'єкт дослідження – процес обробки наборів даних для оцінювання якості реабілітації пост-інсультних пацієнтів.

Предмет дослідження – моделі та засоби обробки Великих даних за допомогою спеціалізованого ПЗ для оцінювання та візуалізації якості реабілітації.

Мета роботи – оцінювання реабілітації шляхом розробки програмного забезпечення для оцінювання реабілітації пост-інсультних пацієнтів шляхом обробки статистичних даних та їх подальшої візуалізації.

У першому розділі було розглянуто актуальність створення системи для оцінювання. У другому розділі продемонстровано моделі і прототипи ПЗ. У третьому розділі наводяться необхідні інструменти та описується архітектура застосунку. В четвертому розділі наведено кроки реалізації застосунку та процеси тестування, а також продемонстрована документація.

В результаті виконаної роботи було реалізовано ПЗ для оцінки реабілітації пост-інсультних пацієнтів, створено документацію до нього, а також інтерфейс користувача.

Фахова частина КРБ викладена на 61 с. (без додатків), містить 24 рис., 6 табл., 3 додатки, 24 джерела в переліку посилань.

Ключові слова: *пост-інсультна реабілітація, статистичний аналіз, тест реакції, набір даних, великі дані.*

ABSTRACT

of the Bachelor's Thesis

"Data processing software for assessing the rehabilitation of post-stroke patients"

Student of group 408: Fynyk Vladyslav Yuriiovich

Supervisor: Dr.Sc. (Eng.), Professor Zhuravska I. M.

The Bachelor's Thesis is intended for the creation the software evaluating the quality of rehabilitation estimations after a stroke. This thesis is relevant due to the reduction of physical activity of technological progress and due to the large number of stressors caused by the beginning of a full-scale war.

The object of the thesis is the process of processing the quality of rehabilitation kits for post-stroke research for evaluation.

The subject of the thesis is models and means of Big Data processing via specialized software for assessment and visualization of rehabilitation quality.

The goal is to evaluate rehabilitation by developing software to evaluate the rehabilitation of post-stroke patients by processing statistical data and their subsequent visualization.

The first section considered the relevance of creating a system for evaluation. The second section demonstrates software models and prototypes. The third section provides the necessary tools and describes the application architecture. The fourth section presents the steps of application implementation and testing processes, as well as the documentation.

As a result of this work, software was constructed to assess the rehabilitation of post-stroke patients, documentation was created, as well as the user interface.

The Bachelor's Thesis contains 61 pages (without appendices), 24 figures, 6 tables, 3 appendices, 24 sources in the reference list.

Keywords: post-stroke rehabilitation, statistical analysis, reaction test, data set, big data.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Загальний опис реабілітаційних процесів після інсульту	8
1.2 Аналіз шкал для оцінювання реабілітації	9
1.3 Огляд та аналіз наявних аналогів обробки даних	13
1.3.1 Вебкалькулятор NIHSS.....	13
1.3.2 Застосунок MoCA Duo.....	14
1.4 Особливості даних, зібраних щодо пост-інсультних пацієнтів.....	15
1.5 Специфікація вимог до створюваного проєкту	16
1.5.1 Призначення та межі проєкту	16
1.5.2 Загальний опис	17
1.5.3 Вимоги до інформаційного забезпечення	17
1.5.4 Вимоги до програмного забезпечення	17
1.5.5 Вимоги до зовнішніх інтерфейсів	17
1.5.6 Властивості програмного забезпечення	18
Висновки до розділу 1	18
2 МОДЕЛЮВАННЯ ТА ПРОТОТИПУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	20
2.1 Ключові особливості обробки Big Data	20
2.2 Основний інструмент моделювання ПЗ.....	21
2.3 Розробка діаграми використання	22
2.4 Розробка дизайну інтерфейсу застосунку.....	28
Висновки до розділу 2	33
3 ТЕХНІЧНЕ ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ...	35
3.1 Пошук набору даних для подальшої обробки	35
3.2 Опис наданих наборів даних	36
3.3 Вибір мови програмування.....	37
3.4 Інструменти, необхідні для реалізації застосунку	38

3.4.1	Бібліотека Pandas.....	38
3.4.2	Бібліотека NumPy.....	39
3.4.3	Бібліотека Matplotlib	39
3.5	Бібліотека створення графічних інтерфейсів Tkinter.....	41
3.6	Розробка архітектури застосунку.....	43
	Висновки до розділу 3	45
4	ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ	46
4.1	Розробка шару роботи з даними	46
4.2	Реалізація сторінок застосунку	48
4.3	Тестування ПЗ.....	53
4.4	Створення документації.....	54
	Висновки до розділу 4	56
	ВИСНОВКИ.....	57
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	59
	ДОДАТОК А Базовий клас для обробників даних	62
	ДОДАТОК Б Код класу для роботи з MoSA шкалою.....	64
	ДОДАТОК В Матеріали апробації роботи	66

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ	– програмне забезпечення
ПК	– персональний комп'ютер
ШІ	– штучний інтелект
BI	– Barthel Scale/Index
GCS	– Glasgow Coma Scale/Score
IDE	– Integrated Development Environment
ML	– Machine Learning
MoCA	– Montreal Cognitive Assessment
mRS –	– Modified Rankin Scale
NIHSS	– National Institutes of Health Stroke Scale
NIST	– National Institute of Standards and Technology
ORM	– Object-related mapping
UML	– Unified Modelling Language

ВСТУП

Незважаючи на те, що людство невпинно розвивається і відбувається поступова автоматизація більшості галузей, у даного розвитку є й велика кількість мінусів, одним з яких є пришвидшення темпу життя і поява більшої кількості стрес-ситуацій. Такий стан справ, вкупі з поширенням малорухомого способу життя, становить серйозну загрозу життю та здоров'ю людини. І навіть при тому, що медицина здійснила доволі великий ривок в порівнянні з минулими століттями, ризик загинути або стати інвалідом внаслідок інсульту залишається доволі великим.

Інсульт – це гострий стан, під час якого вражаються кровоносні судини головного мозку [24]. Він є другою за поширеністю причиною смерті та основною причиною інвалідності дорослих [21]. Порушення нормального функціонування верхніх кінцівок, викликане інсультом, часто приводять до неможливості виконання людиною різних повсякденних справ. Тому якнайшвидше відновлення людини після нього є надзвичайно важливою і актуальною задачею.

Процес відновлення здорового стану організму називають реабілітацією. Вона зазвичай напрямлена на відновлення рухових можливостей людини та її когнітивних здібностей, хоча все частіше люди потребують психічної реабілітації. Задля цього були розроблені спеціальні рухові терапії, що зазвичай проходяться під наглядом медичних працівників. Особливо це актуально через війну, яка почалась узимку 2022 року. Велика кількість постраждалих від неї, можуть мати симптоми, подібні до тих, що наявні в людей, які пережили інсульт.

Тому особливо важливим завданням на даний момент є створення систем для оцінювання якості реабілітації пацієнтів, що дозволило б покращити якість самої реабілітації та пришвидшило розробку нових методик лікування та покращення уже наявних.

Метою даної роботи являється розробка програмного забезпечення для оцінювання реабілітації пост-інсультних пацієнтів шляхом обробки статистичних даних та їх подальшої візуалізації у тому вигляді, що буде зрозумілий людині. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз предметної області та існуючих аналогів;
- обрати інструменти, необхідні для створення застосунку;
- знайти дані для подальшого оброблення;
- визначити та реалізувати основний функціонал застосунку;
- реалізувати відображення результатів обробки;
- створити документацію.

Об'єктом дослідження являється процес обробки даних (Data Set) щодо оцінювання якості реабілітації пост-інсультних пацієнтів.

Предметом дослідження є методи та засоби обробки великих обсягів даних (Big Data) за допомогою спеціалізованого ПЗ для оцінювання та візуалізації якості реабілітації.

Актуальністю розробки даного програмного забезпечення є те, що завдяки йому дослідники з міжнародних організацій зможуть швидше давати оцінку різним реабілітаційним методикам, що дозволить пришвидшити модернізацію загальних відновлювальних процесів, які можна буде проєктувати на інші напрямки медицини, наприклад, післявоєнну реабілітацію.

Розробка даного ПЗ є необхідною як акт удосконалення наявних напрацювань з галузі оцінювання реабілітації та структуризація їх у єдину систему, в якій будуть усі наявні методи оцінки відновлення здоров'я пацієнтів

Серед сфер застосування розроблюваного ПЗ можна виділити такі основні:

- а) загальна медична сфера (використання лікарнями та лікарями);

б) дослідницька сфера (використання міжнародними дослідницькими організаціями;

в) сфера післявоєнної реабілітації.

Робота пройшла апробацію під час Всеукраїнської науково-практичної конференції «Інформаційні технології та інженерія» (Миколаїв, 09–11 лютого 2022 р.).

За результатами бакалаврської роботи опубліковано 1 друковану працю [23].

Робота складається з вступу, чотирьох фахових розділів, висновків, переліку джерел посилання з 24 джерел, 3 додатків та спеціальної частини з питань охорони праці та безпеки життєдіяльності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний опис реабілітаційних процесів після інсульту

Реабілітація – набір втручань, призначених для оптимізації функціонування та зниження інвалідності осіб з захворюваннями у взаємодії з навколишнім середовищем [16]. Тобто завдяки реабілітації людина може бути максимально незалежною у повсякденній діяльності.

Досягається це шляхом усунення основних умов її поганої працездатності (наприклад, болю) і покращення її функціонування у повсякденні, підтримуючи її у боротьбі з труднощами з мисленнями, зором, слухом, спілкуванням, прийомом їжі або пересуванням.

Деякі приклади реабілітації включають:

- вправи для покращення мовлення та спілкування;
- психологічна підтримка;
- фізична підготовка та освіта щодо здорового способу життя
- тренування з використання різних допоміжних засобів (наприклад, тростина, інвалідний візок, тощо) для компенсації ураження рухового апарату.

Реабілітація в значній мірі орієнтована на людину, а це значить, що заходи та підхід, обрані для кожного індивіда, залежать від його цілей та переваг. Часто вона є допоміжним інструментом при хірургічному або медикаментозному втручанні для зменшення та запобігання ускладнень, пов'язаних з багатьма станами здоров'я.

Реабілітація після інсульту поділяється на відновлення когнітивного та фізичного здоров'я. Для оцінювання відновлення організму використовуються різні шкали оцінювання, основними із яких є:

- шкала інсульту Національного інституту охорони здоров'я США (NIHSS);
- модифікована шкала Ренкіна (mRS);
- шкала ком Глазго (GCS);

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

- Монреальська шкала оцінки когнітивних функцій (MoCA);
- індекс Бартеля (BI).

Більшість із них дозволяють проводити оцінку відновлення навіть в домашніх умовах, але лише за наявності зовнішнього спостерігача. Те, що ними можна користуватись без спеціальних знань та вмій позитивно впливає на ступінь їх поширеності серед населення та дозволяє зменшити навантаження на лікарів шляхом зміщення навантаження в роботі з пацієнтами. Але це не відміняє втручання фахівців з області медицини в лікування, оскільки необхідно скрупульозно стежити за всіма змінами в організмі людей, вражених інсультом. З кожним роком з'являється більше досліджень, направлених на визначення ефективності віддаленого лікування хворих [7].

Великою проблемою є те, що немає доступного програмного забезпечення для можливості віддаленого надання інформації про стан пацієнтів до лікарів. Існують певні вебзастосунки, що дозволяють проводити просту оцінку використовуючи згадані вище шкали.

1.2 Аналіз шкал для оцінювання реабілітації

Шкала Національного інституту охорони здоров'я (англ. National Institutes of Health, NIH) зазвичай використовується для оцінювання фізичного здоров'я пацієнта та його когнітивних здібностей (рис. 1.1).

National Institutes of Health Stroke Scale

Score = 0 No stroke
 Score = 1-4 Minor stroke
 Score = 5-15 Moderate stroke
 Score = 15-20 Moderate to severe stroke
 Score = 21-42 Severe stroke

National Institutes of Health Stroke Scale score	
1a. Level of consciousness	0 = Alert; keenly responsive 1 = Not alert, but arousable by minor stimulation 2 = Not alert; requires repeated stimulation 3 = Unresponsive or responds only with reflex
1b. Level of consciousness questions: What is the month? What is your age?	0 = Answers two questions correctly 1 = Answers one question correctly 2 = Answers neither question correctly
1c. Level of consciousness commands: Open and close your eyes. Grip and release your hand.	0 = Performs both tasks correctly 1 = Performs one task correctly 2 = Performs neither task correctly
2. Best gaze	0 = Normal 1 = Partial gaze palsy 2 = Forced deviation
3. Visual	0 = No visual loss 1 = Partial hemianopia 2 = Complete hemianopia 3 = Bilateral hemianopia
4. Facial palsy	0 = Normal symmetric movements 1 = Minor paralysis 2 = Partial paralysis 3 = Complete paralysis of one or both sides
5. Motor arm 5a. Left arm 5b. Right arm	0 = No drift 1 = Drift 2 = Some effort against gravity 3 = No effort against gravity; limb falls 4 = No movement
6. Motor leg 6a. Left leg 6b. Right leg	0 = No drift 1 = Drift 2 = Some effort against gravity 3 = No effort against gravity 4 = No movement
7. Limb ataxia	0 = Absent 1 = Present in one limb 2 = Present in two limbs
8. Sensory	0 = Normal; no sensory loss 1 = Mild-to-moderate sensory loss 2 = Severe to total sensory loss
9. Best language	0 = No aphasia; normal 1 = Mild to moderate aphasia 2 = Severe aphasia 3 = Mute, global aphasia
10. Dysarthria	0 = Normal 1 = Mild to moderate dysarthria 2 = Severe dysarthria
11. Extinction and inattention	0 = No abnormality 1 = Visual, tactile, auditory, spatial, or personal inattention 2 = Profound hemi-inattention or extinction
Total score = 0-42.	

Рисунок 1.1 – Шкала обстеження інсульту Національного інституту охорони здоров'я США

Дана шкала має ухил на визначення відновлення саме когнітивних функцій людини шляхом виконання певних завдань творчих завдань, де необхідно проявити логічне мислення, а також завдяки навантаженню на мозок шляхом запам'ятовування та обробки інформації, яка надається куратором опитування (рис. 1.2).

NAME : _____
Education : _____
Sex : _____ Date of birth : _____
DATE : _____

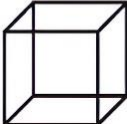
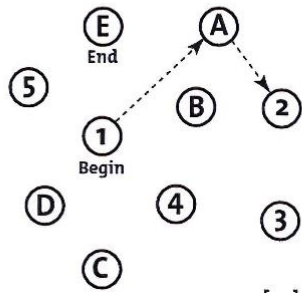
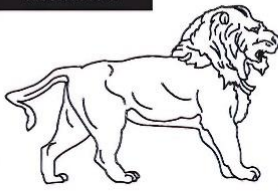
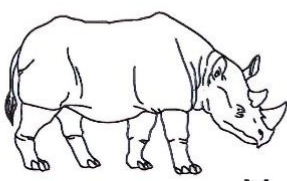
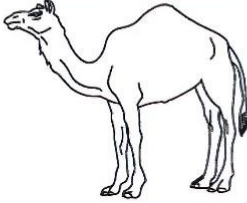
MONTREAL COGNITIVE ASSESSMENT (MOCA)							POINTS																	
VISUOSPATIAL / EXECUTIVE	 Copy cube 	Draw CLOCK (Ten past eleven) (3 points)					<input type="text"/> /5																	
NAMING	 <input type="text"/> []  <input type="text"/> []  <input type="text"/> []						___/3																	
MEMORY	Read list of words, subject must repeat them. Do 2 trials. Do a recall after 5 minutes.	<table border="1" style="margin: auto;"> <tr> <td></td> <td>FACE</td> <td>VELVET</td> <td>CHURCH</td> <td>DAISY</td> <td>RED</td> </tr> <tr> <td>1st trial</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>2nd trial</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>		FACE	VELVET	CHURCH	DAISY	RED	1st trial						2nd trial									No points
	FACE	VELVET	CHURCH	DAISY	RED																			
1st trial																								
2nd trial																								
ATTENTION	Read list of digits (1 digit/ sec.). Subject has to repeat them in the forward order [] 2 1 8 5 4 Subject has to repeat them in the backward order [] 7 4 2						___/2																	
	Read list of letters. The subject must tap with his hand at each letter A. No points if ≥ 2 errors [] FBACMNAAJKLBAFAKDEAAAJAMOF AAB						___/1																	
	Serial 7 subtraction starting at 100 [] 93 [] 86 [] 79 [] 72 [] 65 4 or 5 correct subtractions: 3 pts, 2 or 3 correct: 2 pts, 1 correct: 1 pt, 0 correct: 0 pt						___/3																	
LANGUAGE	Repeat : I only know that John is the one to help today. [] The cat always hid under the couch when dogs were in the room. []						___/2																	
	Fluency / Name maximum number of words in one minute that begin with the letter F [] _____ (N ≥ 11 words)						___/1																	
ABSTRACTION	Similarity between e.g. banana - orange = fruit [] train - bicycle [] watch - ruler						___/2																	
DELAYED RECALL	Has to recall words WITH NO CUE FACE [] VELVET [] CHURCH [] DAISY [] RED []						___/5																	
Optional	Category cue Multiple choice cue						Points for UNCUED recall only																	
ORIENTATION	[] Date [] Month [] Year [] Day [] Place [] City						___/6																	
© Z.Nasreddine MD Version November 7, 2004 www.mocatest.org		Normal ≥ 26 / 30			TOTAL _____ /30 Add 1 point if ≤ 12 yr edu																			

Рисунок 1.2 – Монреальська шкала оцінки когнітивних функцій

Можна помітити, що доволі незручно працювати з результатами даних тестів, оскільки необхідно або використати багато паперу, або створити певну базу даних і працювати з нею. Хоча в науковій і статистичній діяльності зазвичай використовують файли з розширенням `.csv`, з якими можна працювати за допомогою Microsoft Excel, але це є доволі незручним через те, що значення розділені комами, і тому ускладнюється робота з ними як з матрицею. На рис. 1.3 наведено вміст файлу, в який записували координати позицій роботу, що використовувався для збору інформації для покращення фізичної реабілітації, з дослідження університету Торонто [4].

	A
1	-1.262287325959465523e-01,-1.077193039949828846e+00,-3.721644411936243513e-01
2	-5.960892847979269021e-02,-4.493592206643626796e-01,-1.425837878775939038e-01
3	2.233513631821013382e-02,1.570446921636408366e-01,4.798544007738706624e-02
4	2.015739678715347885e-02,5.742013419550905384e-01,5.157155521785213720e-02
5	-5.276533762234666414e-01,-3.907962158685360926e-02,-5.213255847392832092e-03
6	-7.308279559475432396e-01,-5.242200354879061885e-01,-2.259477369605194241e-01
7	-6.757725613866540915e-01,-5.306025635968393273e-01,-8.603463489695104549e-01
8	-5.922144051637963091e-01,-5.250063221191023644e-01,-1.022692624613148693e+00
9	4.611879965408505200e-01,-1.880515480173310938e-01,-4.889766785093919814e-03
10	6.718396967358313265e-01,-6.281308104254044178e-01,-2.026641505361406470e-01
11	7.899493834039318374e-01,-7.734131072699046427e-01,-6.418576339308853518e-01
12	8.116961116698336864e-01,-8.093084893391897516e-01,-7.527967429845534575e-01
13	-2.920509263958914192e-01,-1.008281400870790767e+00,-4.552611752233851061e-01
14	1.110026596818003142e-01,-6.701948024928280345e-01,-1.174914890368647047e+00
15	1.696226196275456982e-02,-1.678547954079365523e+00,-1.552268349825885352e+00
16	2.414151135794619929e-01,-1.315420729106230091e+00,-1.664223027846475800e+00
17	1.412213644970380377e-01,-1.078489678374775051e+00,-4.177978849055593025e-01
18	7.754908743310890973e-01,-7.197392903850644563e-01,-1.061692020790531599e+00
19	6.705381384209839757e-01,-1.726799898085532137e+00,-1.436621724570740000e+00
20	8.658673043519380386e-01,-1.343594663611985318e+00,-1.505179049847015005e+00
21	2.054284785600740244e-02,2.011065254461740042e-02,2.184540472765020610e-02

Рисунок 1.3 – Приклад вмісту csv-файлу

Приклади інструментів для роботи з цим типом файлів будуть наведені в наступному розділі. В даному дослідженні фахівці розробили також робота на основі штучного інтелекту, який використовував машинне бачення для визначення компенсаційних рухів людини, що б дозволило спостерігати за

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів правильністю виконання поставлених завдань та повідомляти про неправильність їх виконання.

1.3 Огляд та аналіз наявних аналогів обробки даних

1.3.1 Вебкалькулятор NIHSS

Вебсайт NIHSS [12] представляє з себе певну форму з можливістю вибору за варіантами (рис. 1.4).

1A: Level of consciousness May be assessed casually while taking history	Alert; keenly responsive	0
	Arouses to minor stimulation	+1
	Requires repeated stimulation to arouse	+2
	Movements to pain	+2
	Postures or unresponsive	+3
1B: Ask month and age	Both questions right	0
	1 question right	+1
	0 questions right	+2
	Dysarthric/intubated/trauma/language barrier	+1
	Aphasic	+2
1C: 'Blink eyes' & 'squeeze hands' Pantomime commands if communication barrier	Performs both tasks	0
	Performs 1 task	+1
	Performs 0 tasks	+2

0 points
 NIH Stroke Scale

Copy Results 📄
Next Steps »»»

Рисунок 1.4 – Інтерфейс вебкалькулятора

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

Переваги:

- 1) доступність;
- 2) автоматичне відображення результатів без необхідності перезавантаження сторінки.

Недоліки:

- 1) відсутність пояснення результатів, а саме ступеня важкості інсульту або відновлення після реабілітації. Надається лише чисельне значення, через що для розуміння усієї картини необхідно самому шукати табличку з поясненнями;
- 2) відсутність ведення статистики.

1.3.2 Застосунок MoCA Duo

Застосунок MoCA Duo розроблений задля проходження MoCA-тесту і доступний для скачування в Play Market (Android) [11] або App Store(iOS). Інтерфейс застосунку наведений на рис. 1.5.

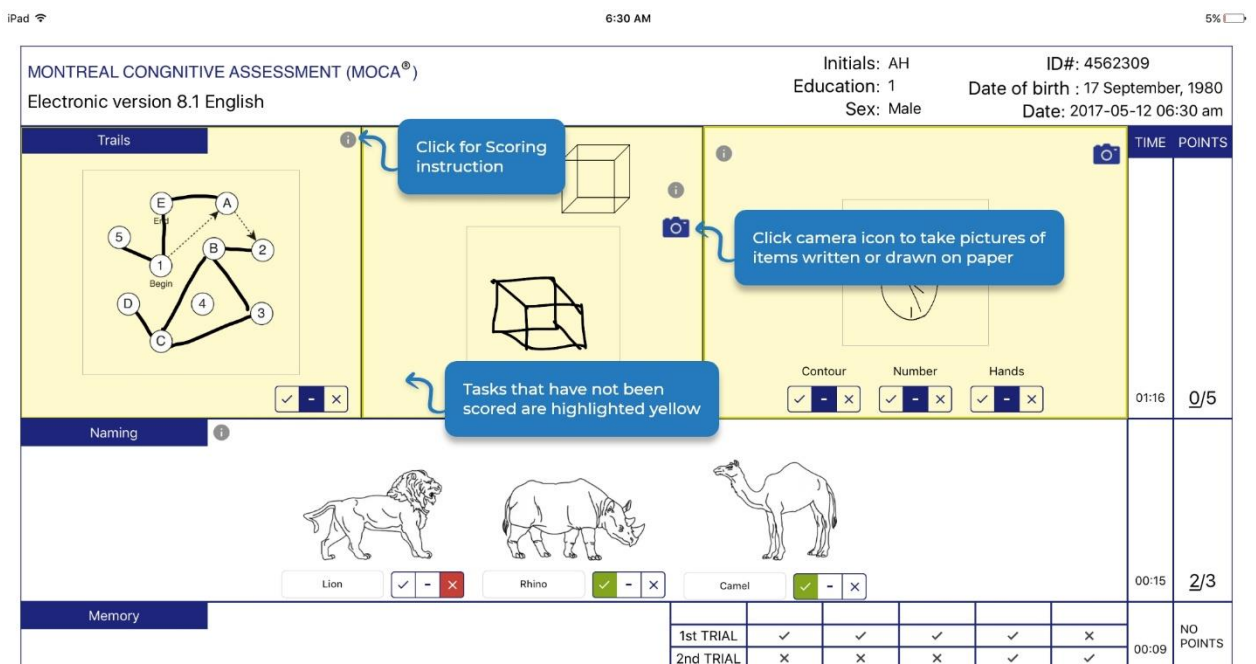


Рисунок 1.5 – Інтерфейс застосунку MoCA Duo

Переваги:

- 1) реалізація потенціалу сенсорних екранів (можна малювати на екрані пальцем або стилусом);
- 2) можливість розпізнавання малюнків на папері шляхом їх фотографування;
- 3) розпізнавання мови в текст (speech to text recognition);
- 4) інструкції на екрані;
- 5) відстеження часу виконання тесту та відповідей на запитання;
- б) ведення статистики.

Недоліки:

- 1) доступність встановлення для невеликої кількості пристроїв;
- 2) відсутність візуалізації статистики.

1.4 Особливості даних, зібраних щодо пост-інсультних пацієнтів

Big Data – це дані, які характеризуються великою різноманітністю та надходять у зростаючих обсягах та з більшою швидкістю [22].

Три особливості великих даних:

- **обсяг** даних має серйозне значення, при роботі з великими даними доводиться обробляти великі обсяги неструктурованих даних з низькою щільністю. Це можуть бути дані невідомої цінності як, наприклад, кліки на вебсторінці або, що відноситься до безпосередньої теми, обладнання з підтримкою датчиків;
- **швидкість** отримання даних також є значно більшою, наприклад, у застосунках, що працюють у режимі реального часу та вимагають відповідних оцінки та дій;
- **різноманітність** відноситься до багатьох типів доступних даних. І якщо традиційні типи даних були структуровані і вписувались в реляційну базу даних, то з появою великих даних дані з'являються в нових неструктурованих типах, наприклад такі як текст, відео та аудіо, що вимагають

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів додаткової попередньої обробки для отримання значення та підтримки метаданих.

Також великі дані використовуються в машинному навчанні, що дозволяє іноді знаходити нові закономірності в даних або дозволяє швидше та якісніше їх обробляти. Яскравим прикладом буде дослідження університету Торонто, де дослідникам для оцінювання довелося реалізувати модель машинного навчання [1].

Дані, зібрані в дослідженнях, вирізняються своєю великою різноманітністю, оскільки інсульт впливає не тільки на фізичний стан організму, через що кількість можливих тестів є надзвичайно великою, а й на когнітивне здоров'я людини, і якщо проводити повний спектр аналізів відновлення організму, то при поєднанні обох кардинально різних методик загальна різноманітність даних стає доволі громіздкою.

1.5 Специфікація вимог до створюваного проєкту

1.5.1 Призначення та межі проєкту

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Система призначена для обробки статистичних даних та їх подальшої візуалізації задля полегшення оцінювання реабілітації пост-інсультних пацієнтів.

Погодження, що ухвалені в програмній документації

Було погоджено, що для роботи з даними будуть використовуватись бібліотеки Pandas та NumPy, оскільки вони надають більш продвинуті структури даних та методи для роботи з ними.

Межі проєкту ПЗ

Проєкт має реалізовувати функції, що були узгоджені при проектуванні ПЗ та вирішувати головні недоліки застосунків-аналогів, а саме: відсутність

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів візуалізації даних та більш глибокого аналізу відновлення стану організму пацієнта.

1.5.2 Загальний опис

Сфера застосування

Створюваний проєкт має використовуватись дослідниками та лікарями.

Загальна структура і склад системи

Система має складатись з прошарку для роботи з даними, а також інтерфейсу, що поєднував би візуал з функціоналом.

1.5.3 Вимоги до інформаційного забезпечення

Джерела і зміст вхідної інформації (даних)

Джерелами є відкриті дослідження з реабілітації, а також надані датасети.

Вимоги до способів організації, збереження та ведення інформації

Інформація має зберігатись в нереляційній базі даних або у відповідних .csv файлах, що використовуються у науковій сфері.

1.5.4 Вимоги до програмного забезпечення

Архітектура програмної системи

Програмна система має складатися з двох ланок: клієнтський застосунок та прошарок роботи з даними.

Мова і технологія розробки ПЗ

Мовою розробки було обрано Python як для розрахунків так і для реалізації програмного інтерфейсу з використанням відповідних бібліотек.

1.5.5 Вимоги до зовнішніх інтерфейсів

Інтерфейс користувача

Користувацький інтерфейс має задовольняти стандартні UI та UX вимоги.

1.5.6 Властивості програмного забезпечення

Доступність

Від ПЗ вимагається доступність для будь-якого користувача, в якого є наявні бібліотеки та дані, що можуть оброблятися системою.

Супроводжуваність

Комплекс застосунків має бути легко супроводжуваним та придатним для подальшої модифікації чи доповнення.

Переносимість

Програмне забезпечення має бути кроссплатформеним. Клієнтські застосунки можуть бути використані будь-яким ПК.

Надійність

Створюваний проєкт має належним чином оброблювати помилки та припиняти свою роботу у разі виникнення нефатальних помилок та демонструвати їх у сприятливій для людського читання формі. Потенційно небезпечні для системи помилки не мають бути показані користувачам.

Безпека

Дані з досліджень мають бути надійно захищені, що наведено в умовах їх отримання.

Висновки до розділу 1

В результаті написання першого розділу було наведено основну інформацію, що необхідна для більш повного та поглибленого розуміння зазначеної теми. Описано загальну інформацію щодо реабілітаційних процесів.

Проведено аналіз наявних шкал для оцінювання реабілітації, робота з якими в подальшому дозволить проводити більш глибокий та інтенсивний

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів
аналіз даних задля їх обробітку та надання відповідних результатів
оцінювання.

Наведено одні з основних систем-аналогів, що в певній мірі реалізують функціонал розроблюваної системи, але при цьому мають низку критично важливих недоліків, вирішення яких є однією з задач дипломної роботи.

Також надано основну інформацію для розуміння того, чим являються «Великі Дані» і чому робота з ними відрізняється від роботи зі звичайними даними.

Сформовано специфікацію вимог до розроблюваного програмного забезпечення задля більш чіткого представлення та опису його поведінки. В ній наведено функціональні і нефункціональні вимоги.

2 МОДЕЛЮВАННЯ ТА ПРОТОТИПУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Ключові особливості обробки Big Data

Оскільки «Великі Дані» зазвичай швидко оновлюються та є неструктурованими, то й інструменти роботи з ними є відмінними від звичайних статистичних систем. Також часто дані є невзаємопов'язаними, що значно ускладнює їх подальшу обробку.

На рис. 2.1 можна побачити архітектуру застосунку для роботи з «Великими Даними», адаптовану для медичної сфери у дослідженні [8].

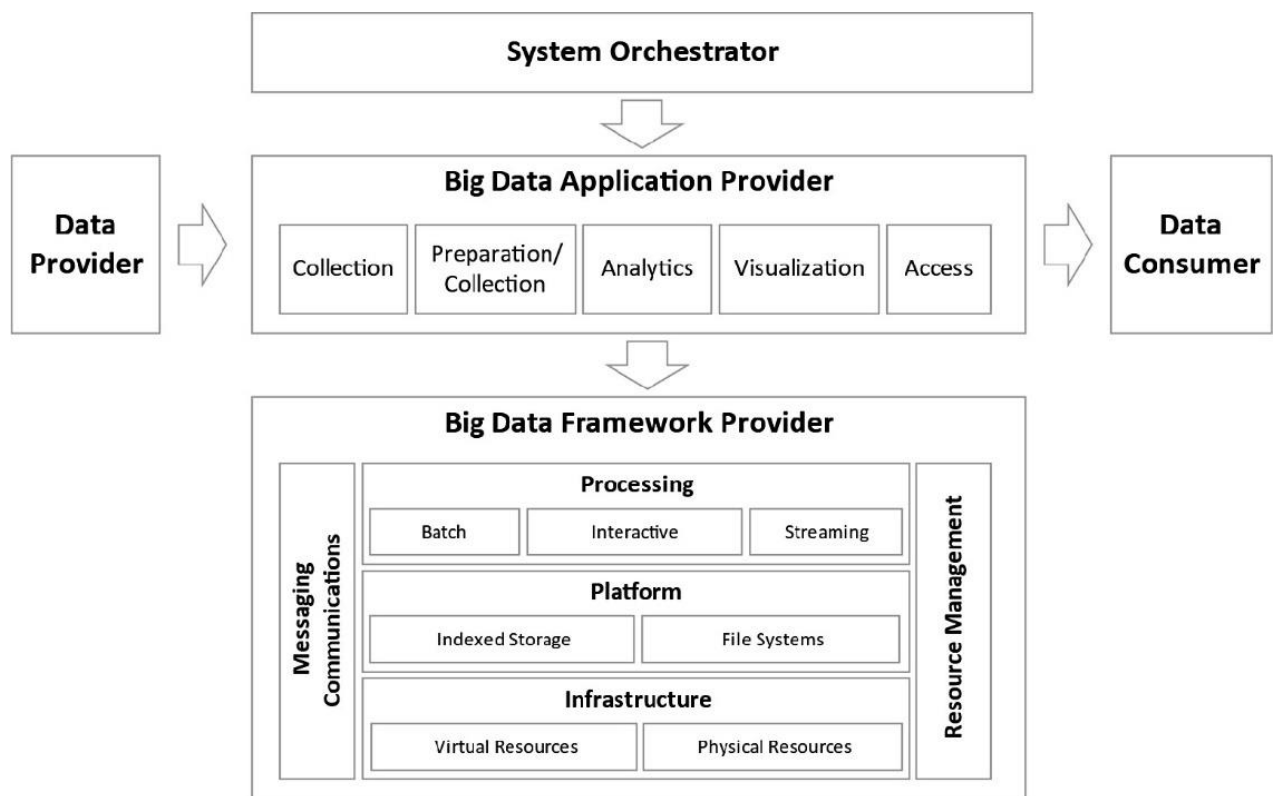


Рисунок 2.1 – Архітектура ПЗ для роботи з Big Data

Пояснюючи дану схему, можна визначити, що:

- System Orchestrator гарантує, що різні програми, дані та компоненти інфраструктури працюють разом;

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

– Data Provider надає нові дані, що надходять з різних джерел, у систему великих даних для перетворення їх у оброблювані Data Set-и;

– Big Data Application Provider містить бізнес-логіку та функціональні можливості, необхідні для перетворення даних у цінні знання за допомогою п'яти основних дій:

- 1) збір;
- 2) підготовка;
- 3) аналітика;
- 4) візуалізація;
- 5) доступ.

– Big Data Framework Provider має ресурси та сервіси для зберігання та обробки даних;

– Data Consumer використовує інтерфейси сервісів, наданих Big Data Application Provider для доступу до необхідної інформації.

Оскільки на даний момент ключовою задачею є саме оцінювання реабілітації, а не збір інформації та її глибока обробка, було обрано рішення зі спрощення архітектури, зазначеної вище, до менш навантаженої, але при цьому такої ж масштабованої. Також в ній відсутнє поєднання з інтерфейсом користувача, реалізація якого також є одним з ключових факторів виконання роботи.

2.2 Основний інструмент моделювання ПЗ

UML – це уніфікована мова моделювання загального призначення для розробки в області інженерії програмного забезпечення, яка призначена для забезпечення стандартного способу візуалізації дизайну системи [2]. Вона використовується розробниками задля [18]:

- надання вказівок щодо порядку діяльності команди;
- демонстрації того, які функції слід розробити;
- спрямування завдань окремих розробників та команди в цілому;

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

– пропонування критеріїв для моніторингу та вимірювання продуктів і діяльності проєкту.

Використання UML при розробці програмного забезпечення є доволі важливим, оскільки основною перевагою є саме стандартизація даної мови, завдяки чому можна легко читати й розуміти створені діаграми та навіть генерувати код з діаграми класів завдяки спеціальним застосункам [6]. Хоча останнє ще потребує подальшого покращення, оскільки до рівня розробника йому ще далеко.

В даній роботі будуть наведені більш конкретизовані випадки UML-діаграм, а саме: діаграми класів, діаграми використання. Використання їх усіх разом дозволить створити більш обширне та чітке представлення щодо створюваного продукту.

2.3 Розробка діаграми використання

Після розгляду поставлених задач та проведених оцінок застосунків-аналогів, було сформовано можливості використання застосунку користувачем, що зазначені на рис. 2.2.

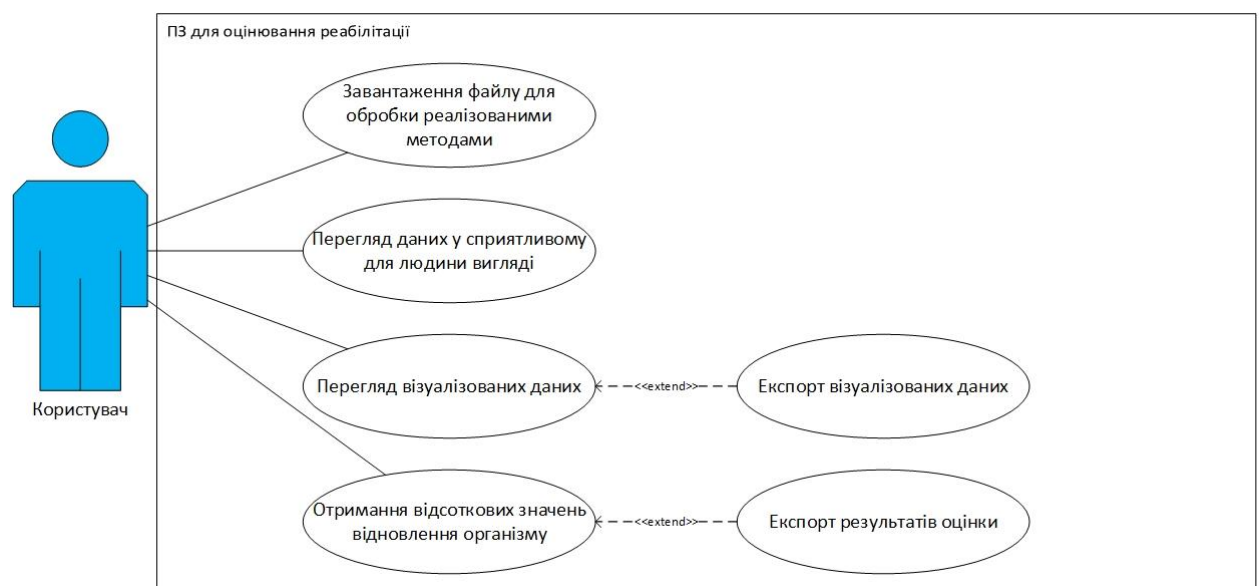


Рисунок 2.2 – Діаграма використання розроблюваного застосунку

Задля надання більш детальної інформації щодо них, було також створено таблиці 2.1-2.6 з описом кожного з прецедентів.

Таблиця 2.1 – Завантаження файлу для обробки

Назва	Завантаження файлу для обробки
ID	1
Короткий опис	Користувач вибирає власний файл, що може бути обробленим реалізованими методами з обробітку даних
Головні актори	Користувач
Другорядні актори	Відсутні
Передумови	Наявність у користувача файлу для обробки програмним забезпеченням
Постумови	Система завантажує обраний файл та проводить відповідні обробки даних.
Основний сценарій	<ol style="list-style-type: none"> 1. Користувач за допомогою відповідного меню обирає файл з власної файлової системи. 2. Система завантажує файл. 3. Система проводить необхідні для подальшої роботи обробки даних.
Альтернативні сценарії	<p>У разі виникнення помилки система повідомляє про неї, інформуючи в залежності від того, на якому етапі помилка виникла. Попередньо передбачені помилки, що можуть виникнути на даному етапі:</p> <ol style="list-style-type: none"> 1. Помилка при завантаженні файлу. 2. Помилка при його обробці.

Таблиця 2.2 – Перегляд даних у сприятливому для людини вигляді

Назва	Перегляд даних у адаптованому для людського перегляду форматі
ID	2
Короткий опис	Користувач переглядає вміст обраного файлу у табличному вигляді
Головні актори	Користувач
Другорядні актори	Відсутні
Передумови	Було обрано файл для перегляду і він успішно завантажений
Постумови	Вміст успішно оброблено для подальшого представлення і користувач його переглянув.
Основний сценарій	<ol style="list-style-type: none"> 1. Користувач за допомогою відповідного меню переходить на сторінку представлення файлу. 2. Вміст файлу приводиться до шаблонного вигляду. 3. Відбувається візуалізація вмісту
Альтернативні сценарії	<p>У разі виникнення помилки система повідомляє про неї, інформуючи в залежності від того, на якому етапі помилка виникла. Попередньо передбачені помилки, що можуть виникнути на даному етапі:</p> <ol style="list-style-type: none"> 1. Помилка при обробці файлу. 2. Помилка під час приведення його до шаблонного вигляду 3. Помилка під час візуалізації

Таблиця 2.3 – Перегляд візуалізованих даних

Назва	Перегляд візуалізованих даних (графіків)
ID	3
Короткий опис	Користувач переглядає вміст обраного файлу у вигляді графіку
Головні актори	Користувач
Другорядні актори	Відсутні
Передумови	Було обрано файл для перегляду і він успішно завантажений
Постумови	Вміст файлу успішно оброблено для подальшого створення графіку та користувач їх переглянув
Основний сценарій	<ol style="list-style-type: none"> 1. Користувач за допомогою відповідного меню переходить на сторінку з графіком. 2. Вміст файлу оброблюється для подальшого графічного представлення. 3. Відбувається візуалізація графіку
Альтернативні сценарії	<p>У разі виникнення помилки система повідомляє про неї, інформуючи в залежності від того, на якому етапі помилка виникла. Попередньо передбачені помилки, що можуть виникнути на даному етапі:</p> <ol style="list-style-type: none"> 1. Помилка при обробці файлу. 2. Помилка під час обробки даних 3. Помилка під час візуалізації

Таблиця 2.4 – Отримання відсоткових значень відновлення організму

Назва	Отримання відсоткових значень відновлення організму
ID	4
Короткий опис	Користувач переглядає результат обробки наданих даних, що являє собою значення відновлення організму у відсотках
Головні актори	Користувач
Другорядні актори	Відсутні
Передумови	Було обрано файл для обробки і він успішно завантажений та оброблений
Постумови	Вміст успішно оброблено для подальшого аналізу та розрахунків. Результати переглянуті користувачем
Основний сценарій	<ol style="list-style-type: none"> 1. Користувач за допомогою відповідного меню переходить на сторінку представлення результатів оцінки. 2. Відбувається обробка даних 3. Відбувається візуалізація оброблених даних
Альтернативні сценарії	<p>У разі виникнення помилки система повідомляє про неї, інформуючи в залежності від того, на якому етапі помилка виникла. Попередньо передбачені помилки, що можуть виникнути на даному етапі:</p> <ol style="list-style-type: none"> 1. Помилка при обробці даних. 2. Помилка під час візуалізації

Таблиця 2.5 – Експорт побудованих графіків

Назва	Експорт побудованих графіків після обробки даних
ID	5
Короткий опис	Користувач експортує візуалізовані графіки
Головні актори	Користувач
Другорядні актори	Відсутні
Передумови	Було обрано файл для обробки і він успішно завантажений, оброблений та візуалізований
Постумови	Користувач успішно експортував візуалізований графік
Основний сценарій	<ol style="list-style-type: none"> 1. Користувач натискає на кнопку експорту графіку 2. Відбувається експорт за вказаним шляхом
Альтернативні сценарії	<p>У разі виникнення помилки система повідомляє про неї, інформуючи в залежності від того, на якому етапі помилка виникла. Попередньо передбачені помилки, що можуть виникнути на даному етапі:</p> <ol style="list-style-type: none"> 1. Помилка під час експорту

Таблиця 2.6 – Експорт результатів оцінки

Назва	Експорт результатів оцінки
ID	5
Короткий опис	Користувач експортує результати оцінки
Головні актори	Користувач
Другорядні актори	Відсутні
Передумови	Було обрано файл для обробки і він успішно завантажений, оброблений та оцінений
Постумови	Користувач успішно експортував результати проведеної оцінки

Кінець таблиці 2.6

Основний сценарій	<ol style="list-style-type: none"> 1. Користувач натискає на кнопку експорту результатів оцінки 2. Відбувається експорт даних за вказаним шляхом
Альтернативні сценарії	<p>У разі виникнення помилки система повідомляє про неї, інформуючи в залежності від того, на якому етапі помилка виникла. Попередньо передбачені помилки, що можуть виникнути на даному етапі:</p> <ol style="list-style-type: none"> 1. Помилка під час експорту

2.4 Розробка дизайну інтерфейсу застосунку

Продуманий дизайн застосунку є дуже важливим його аспектом, оскільки від того, чи буде він лаконічним, простим та зрозумілим залежить і те, наскільки зручно зможуть користуватись ним кінцеві споживачі продукту. Оскільки розробка інтерфейсів не є новим напрямком у інженерії ПЗ, то за цей час уже було узагальнено найкращі практики (англ. Best Practices). Основними з них являються [19; 20]:

1) KISS (Keep it simple, stupid) [3] – якщо розглядати середовище розробки, то це принцип, який стверджує, що більшість систем працюють краще, коли вони являються простими і не ускладнюються. Тобто, якщо зробити проєкцію на дизайн системи, то це буде значити, що вона має бути не сильно навантаженою, щоб користувач не губився у різноманітті кнопок чи текстових полів;

2) кольорова палітра не повинна бути тьмяною: текст повинен бути чітко видимим і вирізнятись, елементи не мають зливатись між собою, щоб користувач міг легко їх відрізнити;

3) елементи повинні бути згруповані за своєю сутністю. Наприклад, функції роботи з файлами мають бути поєднані у спеціальному меню з назвою *File*;

4) інтерфейс повинен бути пристосованим до користувача і до середовища і бути адаптованим під задачі, будь то вебсторінка, гра, чи застосунок з наукового аналізу;

5) елементи повинні виконувати завдання, які від них очікуються. Наприклад, кнопка видалення файлу не має перейменовувати колонки;

6) використовуючи термінологію потрібно бути обережним і насамперед виходити із того, наскільки великий словниковий запас саме користувача, оскільки відсутність розуміння ним певних функцій може призвести до негативного досвіду користуванням ПЗ та зайвою витратою часу на пошук значень слів;

7) шрифти мають бути легкими для читання на всіх типах екранів та пристосованими до різних роздільних здатностей;

8) візуальне розбиття має велику роль при роботі з ПЗ, в якого існує кілька окремих частин в інтерфейсі з різним функціоналом, тому важливим є структурування цих частин і їх відокремлення одне від одного.

Підсумувавши і прийнявши до уваги зазначені принципи, було розроблено мінімалістичні інтерфейси, зображені на рис. 2.3–2.7.

На рис. 2.3 зображено головне вікно застосунку, на якому наявна базова інформація про те, як користуватись застосунком та наведено список реалізованих методик оцінювання реабілітації. Документація буде наведена в декількох екземплярах: одна для користувача і інша для розробників. Оскільки сама по собі інформація, що повинна бути представлена в цих документаціях є доволі відмінною по своїх характеристиках, було обрано саме таке рішення. В головному меню буде наведена документація для користувача, в якій будуть пояснюватись можливості створеного програмного забезпечення та інструкція з використання реалізованих функцій оцінювання, що допоможе швидше зорієнтуватись та почати роботу із оцінювання.

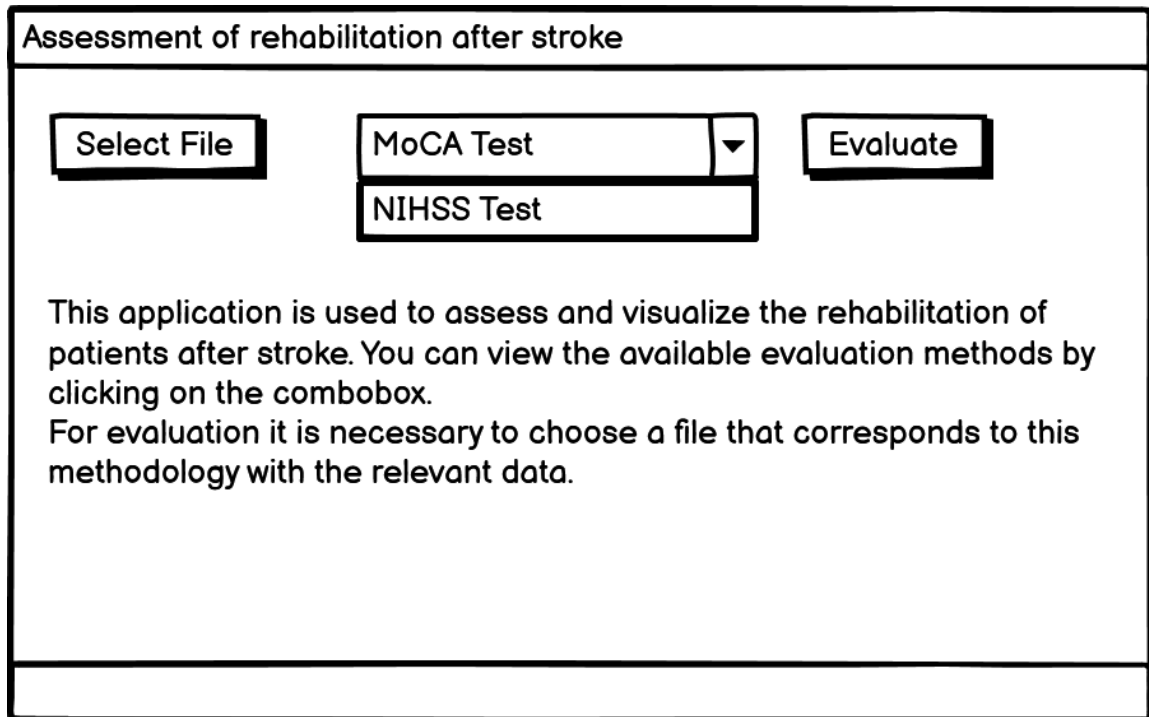


Рисунок 2.3 – Головне вікно застосунку

Оскільки надані дані мережі не можуть бути включеними до файлової системи застосунку згідно з угодою про нерозголошення, то було обрано рішення реалізувати вибір файлу користувачем.

На рис. 2.4 зображена сторінка з табличним представленням даних з обраного файлу. Оскільки роздільником в файлах `.csv` зазвичай є кома, то людині складно сприймати ці дані візуально, через що й було ухвалено рішення з розробки даного інтерфейсу.

Також через використання файлів з великим об'ємом було обрано рішення з реалізації їх відображення з більш малим конкретним обсягом, що буде легким для подальшого рендеру. Задля переходу між обраними проміжками додано 2 кнопки, що будуть використовуватись для зміщення зрізу даних.

Assessment of rehabilitation after stroke				
Back		Visualize data		Path to data
id	characteristic1	characteristic2	...	characteristicN
1	1	3		2
2	0	1		1
3	2	0		1
4	1	1		2
5	3	2		2
6	3	1		3
7	2	2		2
8	3	3		2

[Previous](#)
[Next](#)

Рисунок 2.4 – Сторінка візуалізації даних з файлу

На рис. 2.5 зображена сторінка з представленням даних у вигляді графіків. Лінійний графік в реалізованих методологіях буде відобразити середні та медіанні значення задля більш об'єктивного подання інформації. На ній будуть зображені візуалізовані дані усього масиву даних пацієнтів із прикріпленням до часових проміжків задля полегшення відстеження загальної реабілітаційної тенденції обраної дослідниками групи осіб.

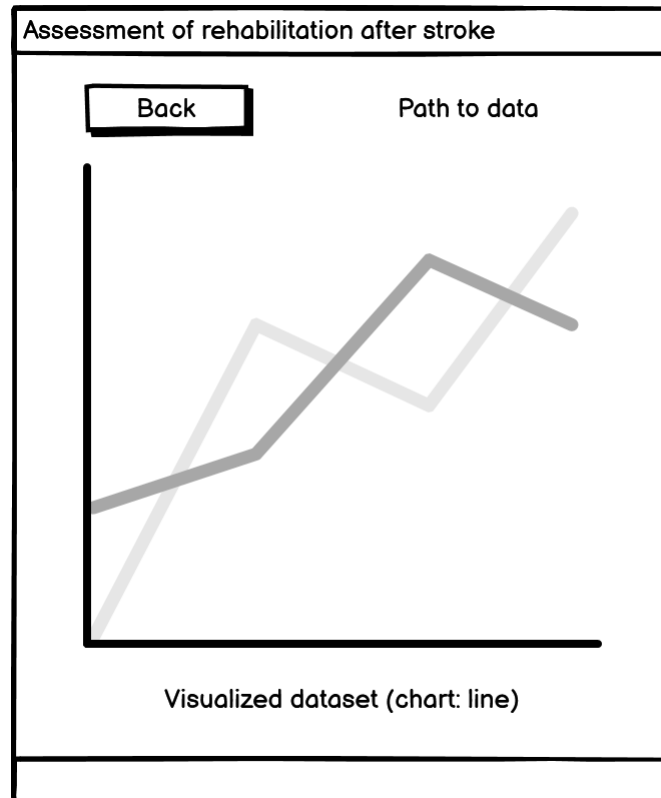


Рисунок 2.5 – Сторінка представлення даних

На рис. 2.6 представлена сторінка кінцевої оцінки реабілітації пацієнта. Вона репрезентує роботу з методологією MoCA Test, через що визначити фізичне відновлення не є можливим Дані будуть представлені у табличному вигляді де це можливо, показуючи відновлення з урахуванням часу реабілітації. Графіки відповідають за візуальну складову оцінки реабілітації, задля спрощення та пришвидшення надання кінцевих висновків. Для цього більш резонно використовувати методологію NIHSS Test, але натомість визначити ментальне здоров'я не є можливим. Задля повної оцінки реабілітації було прийнято рішення об'єднати, де це можливо, пацієнтів із різних наборів даних, оскільки різні тести дозволяють отримати окремо рівень фізичного і когнітивного ураження. Задля цього буде реалізована кнопка з відповідною функцією.

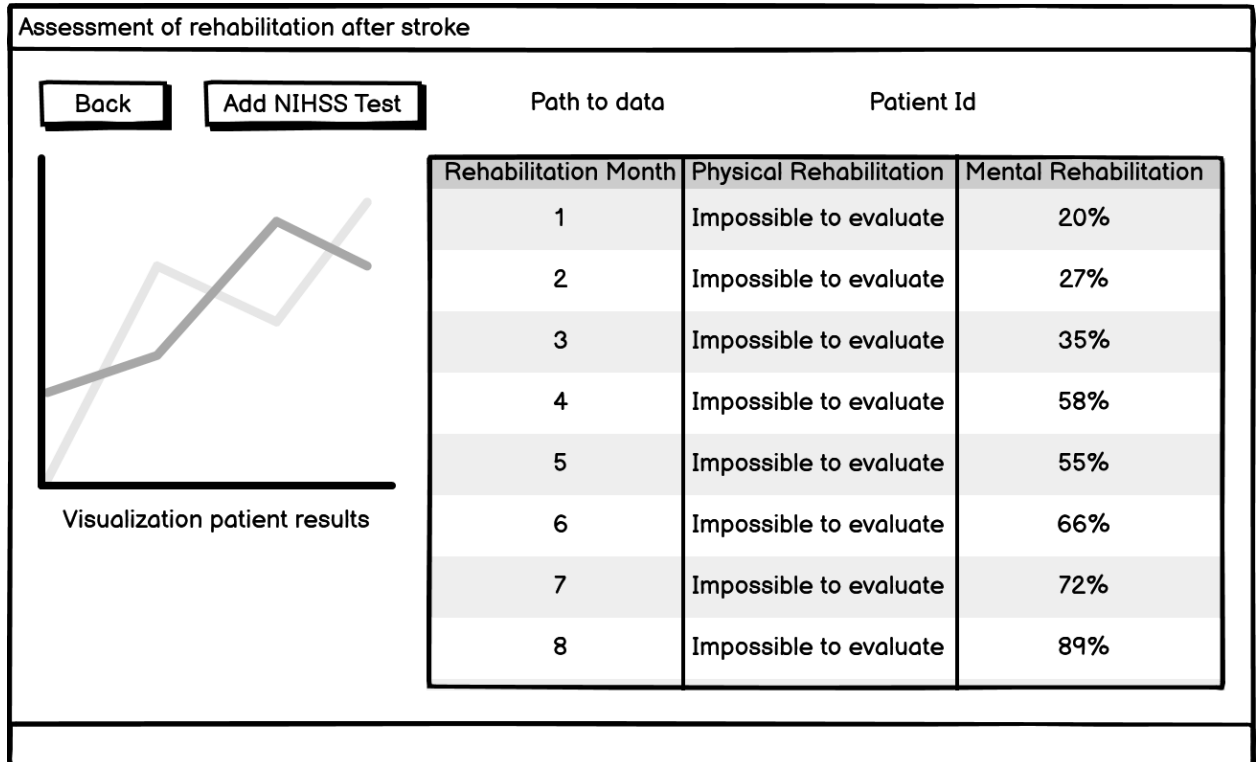


Рисунок 2.6 – Сторінка результатів реабілітації

Наведені вище прототипи майже повністю зображують майбутній інтерфейс застосунку, оскільки деякі інтерфейси, що є майже ідентичними, але з певними відмінностями представлено не було. Такими є інтерфейс оцінювання реабілітації за шкалою NIHSS та спільною NIHSS і MoCA, де відмінністю є можливість визначення різних типів відновлення.

Мова, яка використовувалась при прототипуванні інтерфейсів є англійською, оскільки вона є необхідною для роботи в усіх міжнародних дослідницьких та наукових спільнотах, тому задля поширення використання розроблюваного програмного забезпечення було обрано саме її.

Висновки до розділу 2

В результаті написання другого розділу КРБ було наведено більш докладну інформацію щодо роботи з Big Data поза контекстом конкретної мови та модулів.

Також описано основний інструмент розробки діаграм, що використовується у сфері розробки ПЗ, а саме UML. Наведено створену діаграму використання розроблюваного продукту з детальним описом кожного з прецедентів у вигляді таблиці, в якій подана основна інформація про прецедент, що складається з передумов та постумов, основного та альтернативних сценаріїв.

Наведено основну інформацію, яка є необхідною для створення зручних і легких графічних інтерфейсів Розроблено макети графічного інтерфейсу застосунку згідно з усіма пунктами, зазначеними в відповідному підрозділі. Для кожного із макетів докладно описано його значення та особливості, які необхідно врахувати при його реалізації.

3 ТЕХНІЧНЕ ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Пошук набору даних для подальшої обробки

Одним з ключових завдань, зазначених в першому розділі є пошук якісного набору даних задля реалізації оцінки реабілітації пацієнтів. Задля виконання поставленої задачі було проведено пошук серед відкритих та закритих досліджень з метою отримання даних для подальшої обробки. Основною проблемою стало те, що велика кількість досліджень, які б повністю задовільнили всі вимоги, була відносно застарілою, тому дані могли бути недостатньо актуальними. Іншою ж проблемою стало те, що велика кількість запитів до дослідників була проігнорована і відповіді просто не надійшло.

Після довготривалих пошуків було знайдено дослідження університету Торонто, де сам набір даних був у відкритому доступі, але він складався з координат рухів спеціального механізму, тому без докладної документації обробляти їх було б непрофесійно, оскільки кінцеві висновки могли б виявитись невірними в своїй основі. На жаль, відповідна документація відсутня і в жодному з документів дослідження не згадувалось про щось подібне. Тому

Наступним джерелом потенційного набору даних стала організація що розробляє застосунок MoSA Duo. Було проведено комунікаційну роботу з метою потенційного отримання необхідних наборів даних для подальшої реалізації запланованого застосунку. Робота виявилась успішною і представники даної організації погодились надати наявні в них дані. Також, оскільки вони працюють з шкалою оцінювання, було поставлено ряд ключових питань з метою отримання інформації щодо наявності в них інших даних щодо інших існуючих шкал, наприклад, для доповнення існуючого застосунку або ж створення подібних аналогів. Відповідь виявилась негативною, але натомість було надано інформацію щодо того, що подібні дані

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів можуть бути у організації з якою вони вели співпрацю. Через певний час надійшла інформація про те, що такі дані дійсно є, і пов'язані вони з NIHSS, саме тому вона теж є однією з основних в даній кваліфікаційній роботі.

3.2 Опис наданих наборів даних

Отримані дані є файлами з розширенням `.csv`, один з яких зображений на рис. 3.1.

Id	Trails	Cube	Clock	Naming
1	0	0	1	1
2	1	1	2	1
3	0	1	1	1
4	0	0	1	1
5	1	0	1	2
6	1	0	2	1
7	1	1	1	2
8	1	1	2	2
9	0	1	2	1
10	0	1	2	2
11	1	0	1	1
12	0	0	1	2
13	1	1	2	1
14	0	0	1	2
15	0	0	1	2

Рисунок 3.1 – Вигляд отриманого файлу (MoSA)

Тут зібрані дані 5000 пацієнтів, з періодом запису контрольних тестів в 1 місяць. Сумарно реабілітація проходила максимум 8 місяців в залежності від тяжкості інсульту. Але в основній більшості тут зібрані дані людей з середньою складністю інсульту. Схожим є файл із результатами NIHSS тестів.

Оскільки розголошення особистих даних є забороненим, то замість особистих даних було надано їх хеші, що в подальшому дозволить дослідити наявність пацієнтів, до яких застосовувались обидва типи тестування.

В колонках представлені методики оцінки згідно з відповідними оригінальними тестами, але в певній мірі модифіковані та адаптовані до поставлених дослідниками задач.

3.3 Вибір мови програмування

Майже всі мови програмування можуть в тій чи іншій працювати з «Великими даними», але зручність та швидкість написання програмного забезпечення варіюється в залежності від мови доволі сильно. Найбільш популярними та прогресивними в цьому плані є Scala, Python, R та Go. Кожна з них є потужною у виконанні завдань з обробки даних.

Тому для того, щоб правильно обрати мову необхідно провести аналіз переваг та недоліків кожної з них. В результаті перегляду оцінювань і порівнянь названих мов було прийнято рішення зупинитись на використанні мови програмування Python. Існує декілька ключових особливостей, що зумовили саме такий вибір, а саме:

- а) для неї створено велику кількість модулів, що спрощують процес розробки та тестування;
- б) завдяки використанню існуючих бібліотек можна реалізувати різні види застосунків, наприклад, вебзастосунок або застосунок пристосований безпосередньо до певної операційної системи;
- в) швидкість вивчення базового функціоналу бібліотек є вищою відносно інших мов, тому інтегрування їх в код стає більш швидким, від чого сама розробка прототипу робочого програмного забезпечення потребує значно менше часу;
- г) динамічна типізація, що дозволяє реалізовувати ті конструкції, що є неможливими або складними для написання в мовах зі статичною типізацією. Також за необхідністю можна типізувати необхідні ділянки коду самостійно, що може зменшити кількість проблем із використанням реалізованих функцій;
- д) простота розгортання є великою, оскільки в більшості дистрибутивів, що основані на операційній системі Linux Python наявний за

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів замовчуванням, а в Windows його можна легко встановити за допомогою вбудованого магазину застосунків.

Із недоліків Python можна виділити те, що він є повільнішим в роботі з певними розрахунковими задачами у порівнянні з деякими іншими мовами. І хоча зазвичай це не помітно, при роботі з великою кількістю розрахунків або опрацюванню значних масивів даних це може призвести до певних проблем. Але ці проблеми вирішуються різними модулями, що можна встановити і використовувати в кодї: вони дозволяють значно збільшити швидкість роботи Python завдяки інтеграції C++, C.

3.4 Інструменти, необхідні для реалізації застосунку

3.4.1 Бібліотека Pandas

Pandas – це бібліотека для аналізу, обробки і маніпуляції з даними. Основними перевагами даної бібліотеки є:

- 1) швидкий та ефективний клас DataFrame для роботи з даними;
- 2) інструменти для читання та запису даних у різних форматах, наприклад, .csv, текстові файли, Microsoft Excel, бази даних SQL;
- 3) розумне вирівнювання даних та інтегрована обробка відсутніх даних;
- 4) гнучка зміна та поворот даних;
- 5) легкі вставка та видалення стовбців із структури даних;
- 6) високопродуктивне злиття та поєднання наборів даних;

Окремо необхідно відмітити високу оптимізацію, що зумовлена написанням критичних частин коду на Cython або C [13].

Враховуючи отримані набори даних, дана бібліотека є ідеальним варіантом для зручної, швидкої та простої роботи з ними не тільки через методи обробітку таблиць, а й можливість читання й запису файлів безпосередньо з використанням даної бібліотеки.

3.4.2 Бібліотека NumPy

NumPy – це бібліотека, що дозволяє розширити стандартні структури даних завдяки доданню багатовимірних масивів та матриць, а також великої колекції математичних функцій для роботи з цими масивами [5].

Більшість переваг даної бібліотеки збігається з описаною вище бібліотекою Pandas, оскільки вони добре сумісні між собою і часто використовуються разом для максимальної ефективності і швидкості. Можна легко конвертувати структури даних, надані однією бібліотекою, в структуру даних іншої, що робить код дуже гнучким. Також найбільш потужна частина NumPy, а саме *ndarray*, що представляє з себе багатовимірний масив, часто використовується при тій чи іншій роботі з DataFrame. Використання вбудованих в NumPy математичних функцій дозволяє спростити роботу з таблицями, оскільки в Pandas реалізований режим сумісності, завдяки чому можна легко та швидко обраховувати необхідні показники та застосовувати перетворення до рядків чи стовпчиків таблиць.

3.4.3 Бібліотека Matplotlib

Matplotlib – це потужна бібліотека для створення статичних, анімованих та інтерактивних графіків у Python [10]. Він легко інтегрується з NumPy та різними графічними бібліотеками, такими як Tkinter чи PyQt. Завдяки даній бібліотеці можна побудувати велику кількість різних графіків. Типи графіків, що підтримуються Matplotlib:

- а) базові: звичайний графік (рис. 3.2), стовбуровий, крюковий;

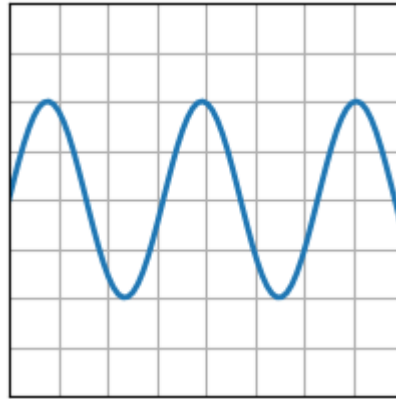


Рисунок 3.2 – Звичайний графік

б) ділянки масивів і полів: контурний (рис. 3.3), зубчатий, сагайдак, сітковий;

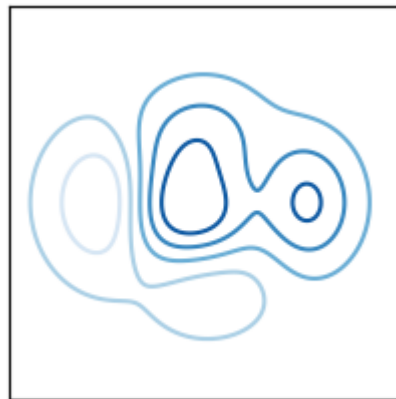


Рисунок 3.3 – Контурний графік

в) графіки статистики: гістограма (рис. 3.4), скрипковий, пиріг;

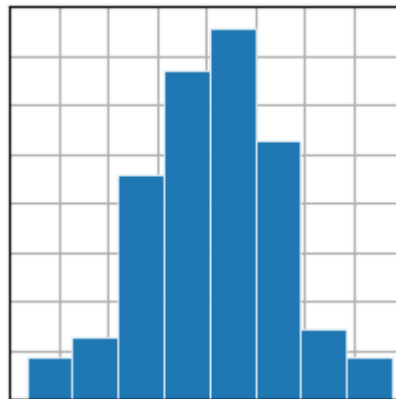


Рисунок 3.4 – Гістограма

г) неструктуровані координати: триконтур (рис. 3.5), приколір.

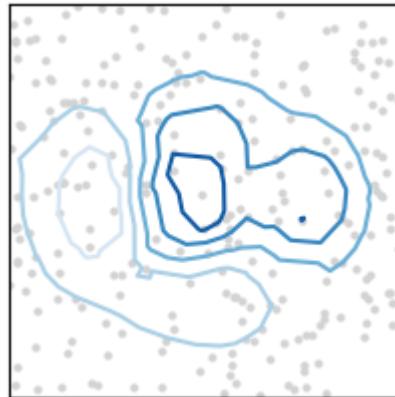


Рисунок 3.5 – Триконтур

Така кількість різноманітних функцій для візуалізації дозволить швидко адаптуватись до вхідних даних без необхідності встановлення сторонніх бібліотек, тому вибір даного модулю є доволі аргументованим.

3.5 Бібліотека створення графічних інтерфейсів Tkinter

Tkinter – стандартний пакет Python для створення настільних графічних інтерфейсів. Він дозволяє створювати кросплатформні застосунки, що будуть працювати в більшості Unix-платформ, macOS та Windows.

Архітектура даного модулю є доволі розрізненою, оскільки він складається з декількох різних модулів, кожен з яких має окремий функціонал і власну офіційну документацію [17].

Tkinter є значно простішим для побудови малих настільних застосунків на відміну від основного його конкурента PyQt, який надає більш просунуті можливості для роботи з оформленням та стилізацією. Попри це для реалізації користувацького інтерфейсу було обрано саме Tkinter, оскільки:

- а) він є безкоштовним для будь-якого комерційного використання і є відкритим вихідним кодом;
- б) його легко зрозуміти та опанувати через невеликий об'єм бібліотеки;
- в) він є вбудованим в Python за замовчуванням.

Це основні переваги даного пакету, але в нього також присутні певні недоліки, але вони є не настільки критичними аби змінити ухвалене рішення. Одним з ключових мінусів Tkinter часто визначають дизайн, але після останніх оновлень він став виглядати краще і тому його можна відкинути. На рис. 3.6 зображено тестовий застосунок, що дозволяє обрати і застосувати тему до графічних елементів. Можна помітити, що даний дизайн є доволі непоганим і мінімалістичним.

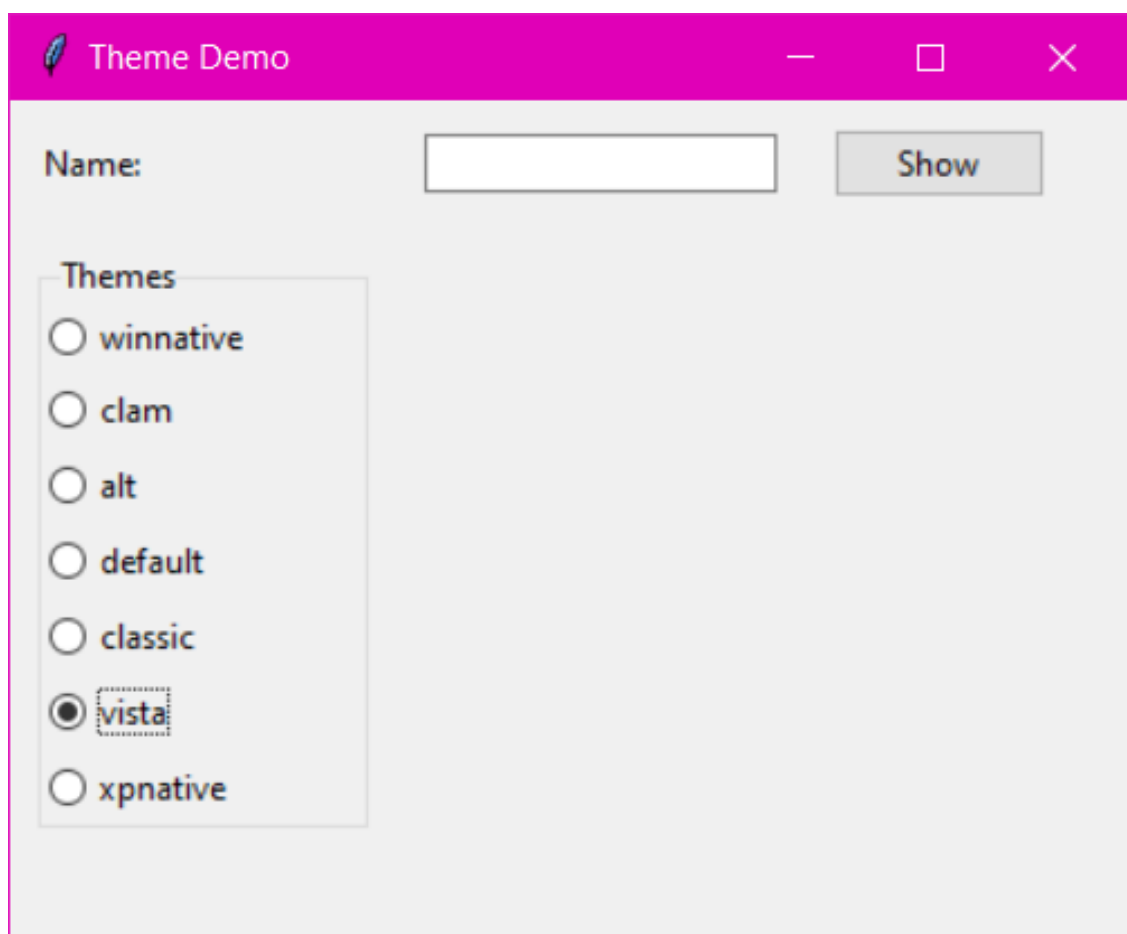


Рисунок 3.6 – Тестовий застосунок з використанням Tkinter

Ще одним недоліком, хоч і незначним, є те, що у Tkinter відсутній інструмент дизайнера, тому всі елементи необхідно прописувати вручну, але натомість існують різні бібліотеки, які можуть перетворити шаблон інтерфейсу Figma в реальний дизайн, що може значно спростити та пришвидшити процес дизайну застосунку.

3.6 Розробка архітектури застосунку

Утримання балансу між простотою і зрозумілістю коду та гнучкістю архітектури є досить складним завданням. Саме для допомоги у вирішенні цих задач, а також низки архітектурних рішень, використовується принцип SOLID [9], що є аббревіатурою. Значення символів аббревіатури:

а) Single responsibility principle (Принцип єдиної відповідальності) – клас має відповідати тільки за одну частину функціональності програми, причому вона повинна бути повністю інкапсульована в ній.

б) Open/closed principle (Принцип відкритості/закритості) – класи мають бути відкритими для розширення, але закритими для зміни, щоб не ламати існуючий код при внесенні змін.

в) Liskov substitution principle (Принцип підстановки Лісков) – підкласи повинні доповнювати, а не заміщати поведінку базового класу, щоб у випадку чого об'єкти могли бути замінені нащадками без зміни коду програми.

г) Interface segregation principle (Принцип поділу інтерфейсу) – краще мати багато спеціалізованих інтерфейсів ніж один універсальний задля майбутньої гнучкості компонентів.

д) Dependency inversion principle (Принцип інверсії залежностей) – класи верхніх рівнів не мають залежати від класів нижніх рівнів; обидва повинні залежати від абстракцій.

Прийнявши до уваги наведені принципи, було розроблено діаграму класів для прошарку роботи з даними, зображений на рис. 3.7.

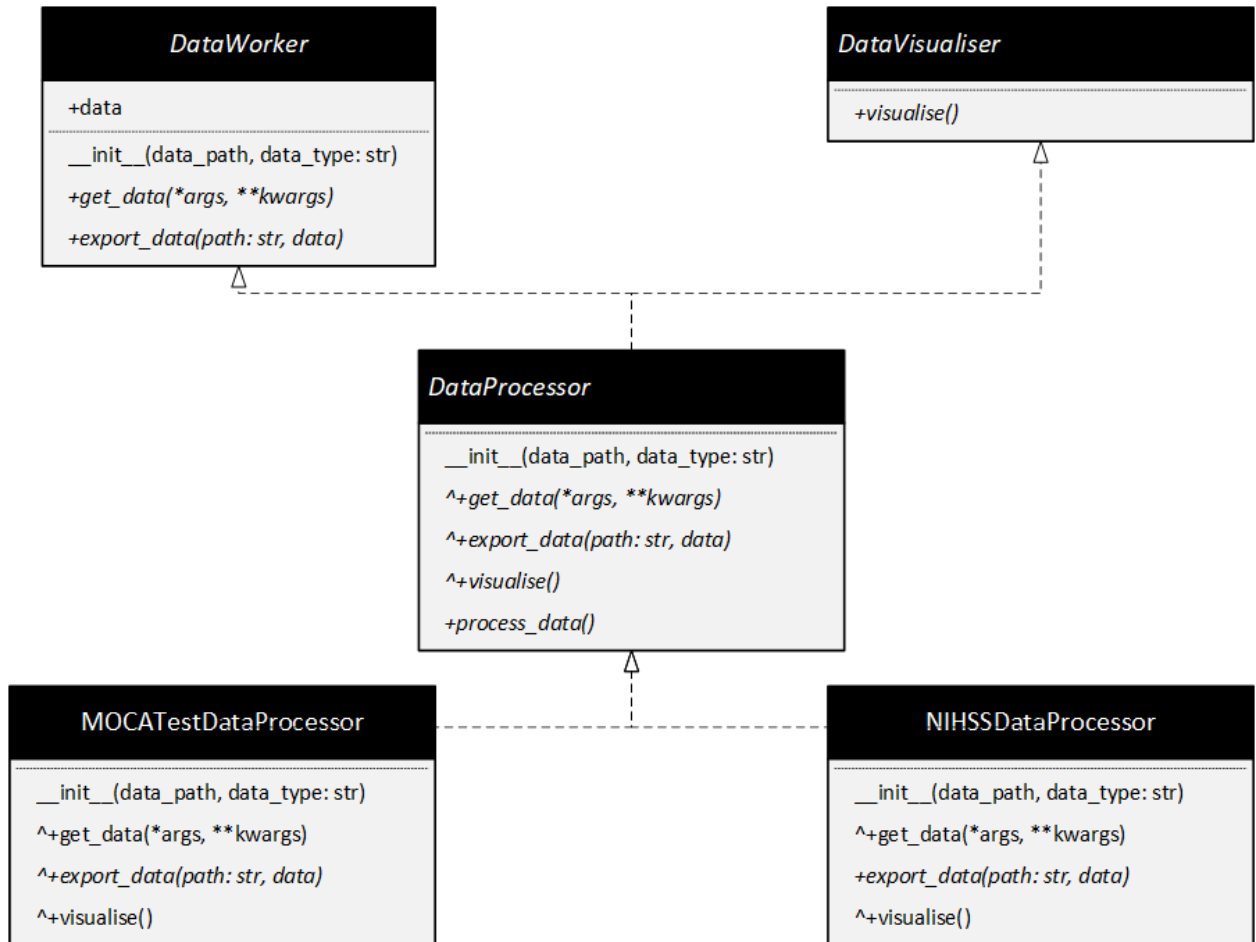


Рисунок 3.7 – Діаграма класів прошарку роботи з даними

Тут окремі необхідні функції розподілені між абстрактними класами: це зумовлено особливостями мови програмування Python в якій немає інтерфейсів. Завдяки цьому відбувається компонування необхідних функцій до вихідних кінцевих класів.

Для розробки інтерфейсу використано реалізацію структурного шаблону MVC (model, view, controller) [15]. Моделі зазвичай це структури даних, які містять в собі поля, необхідні для роботи з ORM задля спрощення роботи з БД. Але в даному випадку замість нього буде окремий шар роботи з даними, який буде використовуватись представленнями.

Контролер зв'язуватиме представлення між собою та дозволить змінювати інтерфейс в залежності від обраних користувачем дій.

Висновки до розділу 3

У результаті виконання роботи над третім розділом кваліфікаційної роботи було детально описано джерело наборів даних та наведено ті проблеми, з якими довелось зіштовхнутись при їх пошуку та запитах на отримання.

Також було описано отримані набори даних та пояснено ключові особливості, які пов'язані з безпекою даних людей, що брали участь у дослідженні, реабілітації і подальшому зібранні даних.

Було наведено інформацію про ключові інструменти, використання яких є необхідним і критично важливим для успішного виконання кваліфікаційної роботи.

Надано основну інформацію щодо бібліотеки, яка буде використовуватись для реалізації графічного інтерфейсу застосунку. Наведено її плюси та мінуси, а також продемонстровано графічні елементи, які в ній використовуються.

Описано шаблон MVC згідно з яким буде розробляться застосунок та наведено певні відмінності в реалізації, що спричинені обраними рішеннями.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

4.1 Розробка шару роботи з даними

Основною задачею даного прошарку є різноманітна робота та маніпуляція даними, аби не ускладнювати розробку інтерфейсу зайвою логікою. Вирішення досягається шляхом створення класів для кожного методу оцінки реабілітації, які б містили усі необхідні методи для читання, запису, перетворення даних тощо. Структура даного шару показана на рис. 3.7, де загальна логіка перенесена в базові класи і перевизначається у дочірніх класах. При цьому можна легко створити різні допоміжні функції всередині похідних класів, які б були локально корисними, без залежності від зовнішніх факторів.

В результаті реалізації розробленої раніше діаграми, було створено класи, що відповідають за кожен окрему методологію оцінки і працюють в залежності від очікуваних даних.

Оскільки шкали MoCA і NIHSS є зворотними за своїми чисельними характеристиками, то і оцінка реабілітації також відрізняється. В шкалі MoCA чим вищий отриманий бал, тим вища оцінка ментального здоров'я людини. Натомість в шкалі NIHSS вища оцінка реабілітації відповідає нижчій оцінці тесту і є максимальною при 0 балів.

Для репрезентативності оцінки реабілітації усіх пацієнтів використовуватимуться два значення: середнє значення і медіана. Їх поєднання дозволить побачити більш об'єктивну картину, що дозволить зменшити імовірність похибки в оцінюваннях до мінімуму.

Першим і основним є клас для читання-запису даних, зображений на лістингу нижче:

```
class DataWorker(ABC):
    """
    Abstract class for data workers.
    """

    @abstractmethod
```

```

def __init__(self, data_path, data_type: str):
    """
    Initializes the data worker.

    :param data_path: Path to the data.
    :param data_type: Type of the data.
    """
    pass

    @abstractmethod
    def get_data(self):
        pass

    @abstractmethod
    def export_data(self, path: str, data):
        pass

```

Він являється загальним інтерфейсом, що містить в собі основні функції для роботи з читанням і записом даних. Оскільки типи даних і їх отримання може бути різним, було обрано рішення не встановлювати тип параметру *data_path*, оскільки це може бути як завантаження локальних файлів на ПК, так і отримання їх через інтернет.

Наступним важливим класом є *DataProcessor*, представлений у додатку А. В ньому наявна головна абстрактна функція, яка дозволяє отримати уже оброблені дані. За відсутністю необхідності обробки можна повернути початкові дані. Також в даному класі реалізовані статичні методи обробки даних, необхідні для візуалізації. Параметр *data* тут є типізованим, а саме класом *DataFrame* модуля *pandas*, оскільки всередині використовуються функції, притаманні даному класу, а тому якби розробники, що працювали б над даним кодом, підставили масив замість екземпляру зазначеного класу, то це призвело б до помилки під час інтерпретації, оскільки IDE не повідомила б про помилку на стадії редагування коду.

Ключовими ж являються кінцеві класи для роботи з шкалами, один з яких зображений у додатку Б. В ньому реалізовані всі необхідні методи для роботи з даною шкалою MoSA, а саме функція оцінки реабілітації, візуалізації

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів даних та їх експорту. Хотілось би зауважити, що попри візуально малу кількість обробників помилок, вони не завжди є доречними, а в тих випадках, де в розробленому коді не використовуються, взагалі непотрібними. Наприклад, на лістингу нижче перевіряти чи є чи існує колонка *Id* або ж чи існує пацієнт з відповідним ідентифікатором не є доцільним, оскільки відповідно до контексту, звідки ця функція викликається, не може бути випадку, коли хоч одне із необхідних значень не є визначеним.

```
patient_data = self.data[self.data['Id'] == patient_id]
```

Клас для роботи з шкалою NIHSS є аналогічним, але з певними незначними особливостями, які настільки ж детально розбирати не є необхідним, оскільки основна важлива інформація була донесена в розділі про `MoCATestDataProcessor`.

4.2 Реалізація сторінок застосунку

Для реалізації основного вікна застосунку був створений клас, що є наслідником від `tkinter.Tk`. Він містить в собі контейнер для відображення сторінок, розміщення елементів в яких відбувається за допомогою сітки (`grid`). Даний клас наведений на лістингу нижче:

```
class MainApp(tk.Tk):
    def __init__(self, *args, **kwargs):
        tk.Tk.__init__(self, *args, **kwargs)

        self.title("Rehabilitation assessment")

        self.container = tk.Frame(self)
        self.container.grid(row=0, column=0, sticky="nsew")

        self.container.grid_rowconfigure(0, weight=1)
        self.container.grid_columnconfigure(0, weight=1)

        self.show_frame(StartPage)
```

```
# to display the current frame passed as parameter
def show_frame(self, cont, *args, **kwargs):
    frame = cont(self.container, self, *args, **kwargs)
    frame.grid(row=0, column=0, sticky="nsew")
    frame.tkraise()
```

Завдяки сітковій розмітці можна легко створити дизайн, який буде відповідати розробленим макетам.

Результатом став фрейм, зображений на рис. 4.1.

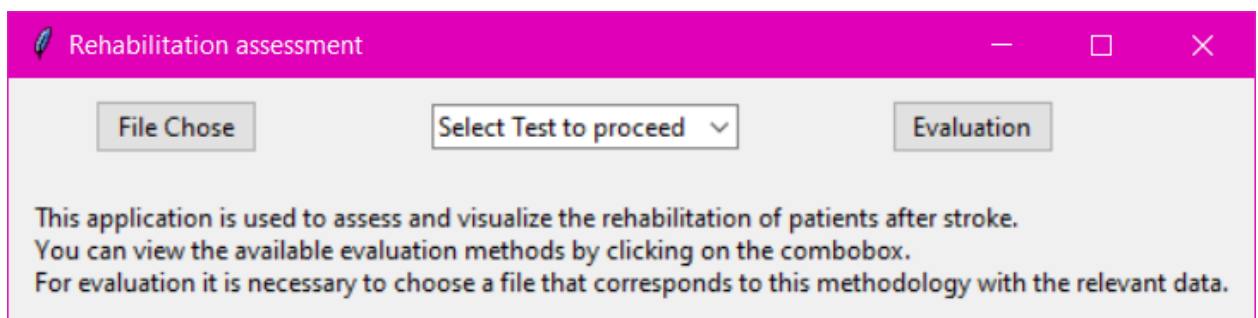


Рисунок 4.1 – Головний фрейм застосунку

Важливим тут є пояснення маршрутизація в застосунку. Перехід на новий фрейм відбувається завдяки виклику в контролері елементу відповідної функції для відображення, передавши необхідні параметри. Приклад кнопки, що реалізує даний перехід наведено на лістингу нижче:

```
self.evaluation_btn = ttk.Button(self, text="Evaluation",
                                command=lambda: controller.show_frame(
                                    get_page_class_by_name(
                                        self.tests_combo.get()),
                                    self.file_path))
```

Задля співставлення назв сторінок із їх класами було реалізовано спеціальну функцію, наведену на лістингу нижче:

```
def get_page_class_by_name(class_name: str):
    classes_with_names = {
        'MoCA Test': MoCATestPage,
        'NIHSS Test': NIHSSTestPage,
    }
```

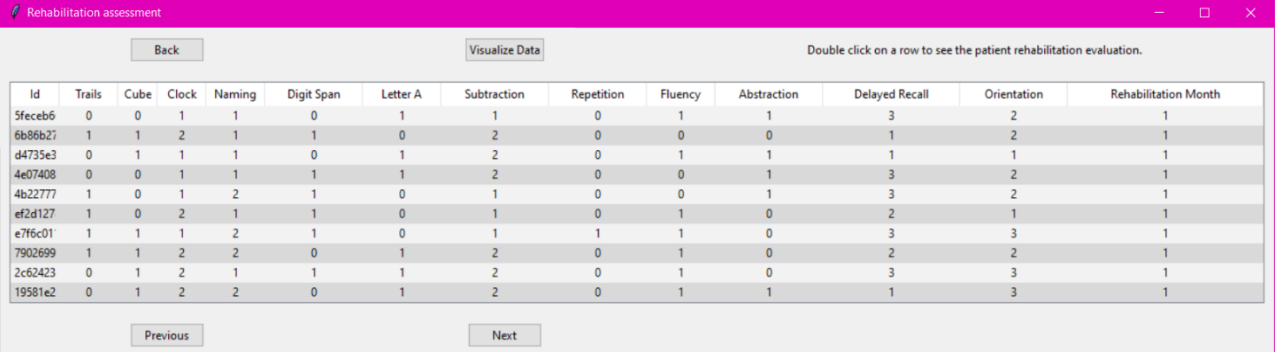
```

try:
    return classes_with_names[class_name]
except KeyError:
    raise ValueError(f'Unknown page class name: {class_name}')

```

Дана функція необхідна для того, щоб можна було за значенням із *combobox*, що використовується для обрання методу оцінки, перейти на необхідну сторінку. Можна було також зробити глобальний словник, але тут є певна проблема, що не дозволить реалізувати все через те, що елементи на початку інтерпретованого файлу нічого не знають про елементи після них. Це можна виправити імпортом анотацій з модулю `__future__`, але це спрацює лише всередині функцій. Поза ними дані маніпуляції будуть неуспішними і IDE буде видавати помилку, оскільки посилання на функцію чи клас є невизначеним.

Наступним фреймом є таблична візуалізація (за можливістю) обраних даних, зображений на рис. 4.2.



Id	Trails	Cube	Clock	Naming	Digit Span	Letter A	Subtraction	Repetition	Fluency	Abstraction	Delayed Recall	Orientation	Rehabilitation Month
5feceb6	0	0	1	1	0	1	1	0	1	1	3	2	1
6b96b27	1	1	2	1	1	0	2	0	0	0	1	2	1
d4735e3	0	1	1	1	0	1	2	0	1	1	1	1	1
4e07408	0	0	1	1	1	1	2	0	0	1	3	2	1
4b22777	1	0	1	2	1	0	1	0	0	1	3	2	1
ef2d127	1	0	2	1	1	0	1	0	1	0	2	1	1
e7f6c01	1	1	1	2	1	0	1	1	1	0	3	3	1
7902699	1	1	2	2	0	1	2	0	1	0	2	2	1
2c62423	0	1	2	1	1	1	2	0	1	0	3	3	1
19581e2	0	1	2	2	0	1	2	0	1	1	1	3	1

Рисунок 4.2 – Фрейм з табличною візуалізацією даних

Кнопка *Visualize Data* реалізовує перехід на фрейм з візуалізацією табличних даних, обраних користувачем. Справа від неї знаходиться елемент *Label*, що відповідає за інформацію, необхідну для користувача. В даному випадку, в ньому реалізовано повідомлення про те, що переглянути оцінку пацієнта можна завдяки подвійному кліку на необхідний рядок. Також через те, що кількість записів є надзвичайно великою і рендер може бути доволі

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів довгим, було обрано рішення реалізувати пагінацію записів. За переміщення відповідають кнопки *Previous* і *Next*.

Наступним фреймом є візуалізація даних з попереднього фрейму, зображений на рис. 4.3. Під зображенням реалізовано меню для роботи з ним, головною функцією в якому є саме можливість експорту зображення.

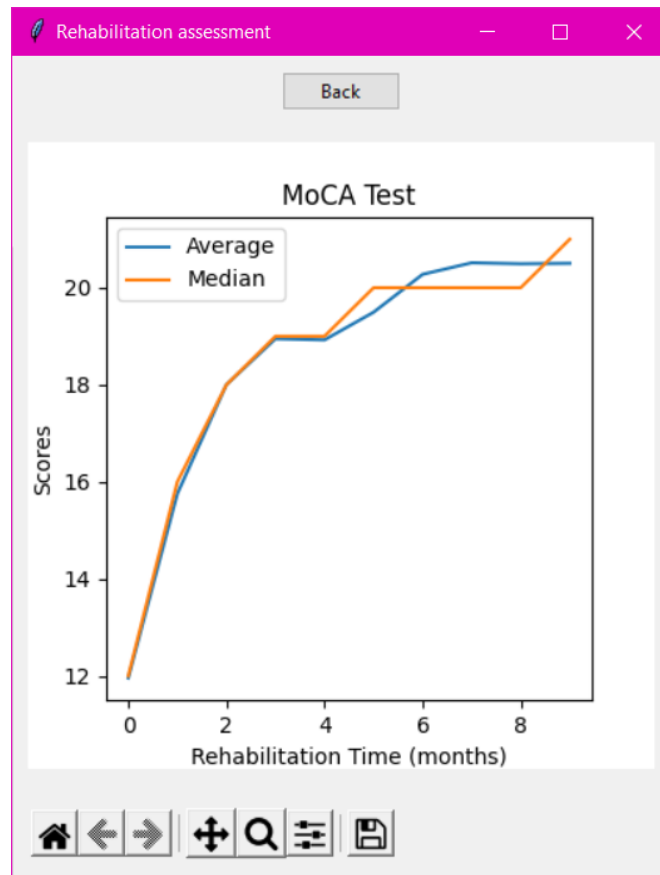


Рисунок 4.3 – Фрейм з візуалізацією даних

Візуалізація відбувається завдяки класам *Figure* та *FigureCanvasTkAgg* бібліотеки *matplotlib*. Перший відповідає за власне фігуру та її зображення, а другий дозволяє отримати віджет для подальшого відображення в фреймі. Приклад суміщення роботи цих класів для отримання результату, зображеного на рис. 4.3 наведено на лістингу нижче. Також на ньому вказано як необхідно додавати зображення до фрейму. Відбувається це через отримання віджету завдяки функції *get_tk_widget()*.


```

fig = Figure(figsize=(4, 4), dpi=100)
plot_line = fig.add_subplot(111)
processed_data = self.data_processor.process_data()
plot_line.plot(

pdw.DataProcessor.get_average_for_visualize(processed_
data))
plot_line.plot(

pdw.DataProcessor.get_median_for_visualize(processed_d
ata))

plot_line.legend(['Average', 'Median'])
plot_line.set_title(plot_name)
plot_line.set_xlabel('Rehabilitation Time (months)')
plot_line.set_ylabel(score_name)

canvas = FigureCanvasTkAgg(fig, self)
canvas.get_tk_widget().grid(row=1, column=0, padx=10,
pady=10)

```

Останнім і головним фреймом є фрейм з результатами оцінювання пацієнта, зображений на рис. 4.4.

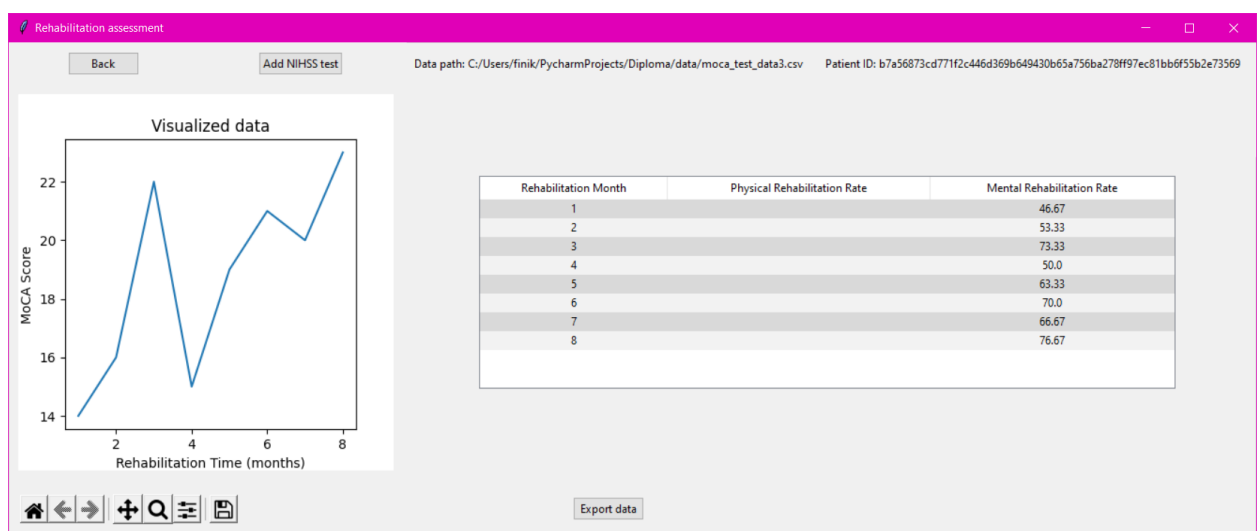


Рисунок 4.4 – Фрейм з результатами оцінювання пацієнта

Гідною уваги тут є кнопка для додання результатів оцінювання за шкалою NIHSS, якщо такі наявні, шляхом вибору файлу з необхідними

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів даними. Тоді відбудеться пошук по *Id* пацієнта, і якщо такого знайдено не буде, з'явиться вікно з відповідним повідомленням, зображене на рис. 4.6. Даний пацієнт існує в файлі з результатами NIHSS тестів, тому результат його фізичного відновлення можна спостерігати на рис. 4.5.

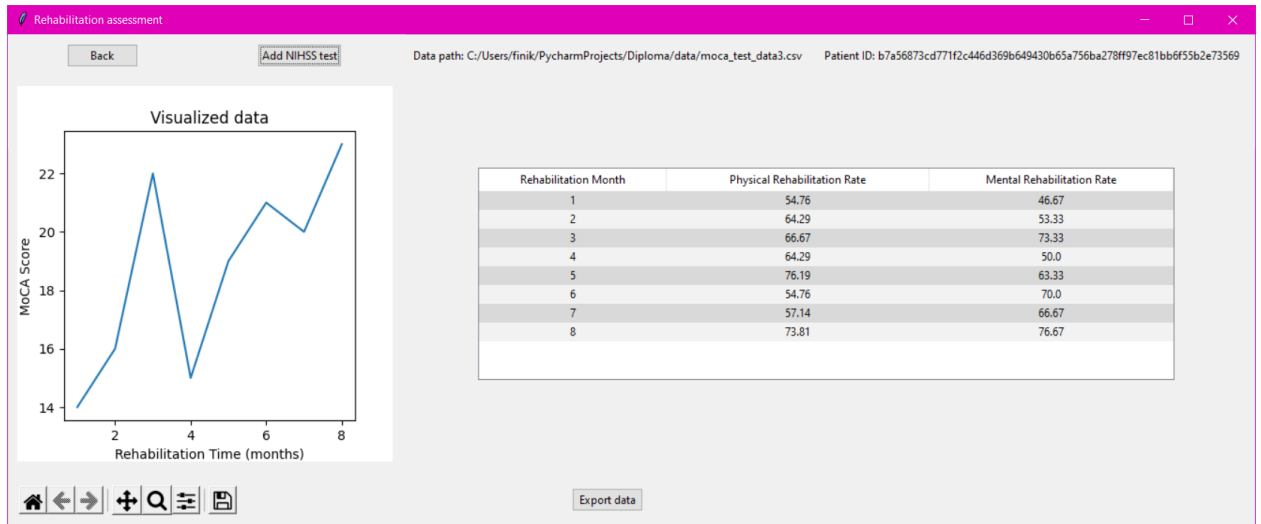


Рисунок 4.5 – Результат додання оцінювання NIHSS шкали

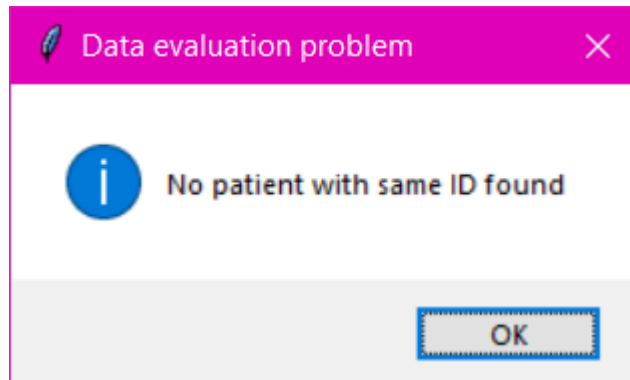


Рисунок 4.6 – Повідомлення про відсутність даних пацієнта

4.3 Тестування ПЗ

Під час тестування програмного забезпечення його працездатність перевірялась, розділено відповідно до тих помилок, які може викликати користувач, і тих, що можуть бути всередині функцій. Зі сторони користувача основною можливою помилкою є обрання файлу. На даному етапі можуть виникнути такі помилки:

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

- а) файл є порожнім;
- б) файл не підходить для обробки через відсутність необхідних колонок;
- в) проблеми з кодуванням;
- г) неправильний тип файлу.

Відповідно до кожної з цих імовірних помилок було реалізовано виведення необхідної інформації про помилку. Запис помилки наведено на лістингу нижче:

```
try:
    self.data = pd.read_csv(data_path)
    self.read_result = 'Success'
except FileNotFoundError:
    self.read_result = 'File not found'
except pd.errors.ParserError:
    self.read_result = 'File is not a valid CSV file'
except UnicodeDecodeError:
    self.read_result = 'File is not a valid CSV file'
```

Відображення помилки в залежності від результатів читання наведено на наступному лістингу:

```
if self.data_processor.read_result != 'Success':
    alert('Error', self.data_processor.read_result,
'error')
    return
```

Зі сторони коду помилковість могла бути пов'язаною з неправильними обрахунками і відображенням елементів, але після пильного огляду та проведення декількох сеансів тестування недоліків виявлено не було.

4.4 Створення документації

Важливими елементами документації коду є правильні назви та позначення типів. Це часто дозволяє одразу зрозуміти що робить функція і не витрачати зайвий час на перегляд текстової документації, яка, зазвичай, є більше доповненням, аніж основою. В Python є таке поняття як *docstring* –

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів рядковий літерал, що пишеться одразу після визначення функції та обрамлюється потрійними лапками, причому неважливо одинарними чи подвійними [14]. Зазвичай в них пишуть загальний опис функції, описують параметри та наводять приклади використання. Завдяки цьому розвинені IDE накрштаут Pycharm дозволяють легко переглядати документацію при наведенні на функцію.

Також важливою частиною гарної документації є коментарі, які мають бути доречними, наприклад, як на лістингу нижче:

```
# adding different backgrounds to the rows of the
table will make it
# easier to perceive them visually
self.data_table.tag_configure('even',
background='#f2f2f2')
self.data_table.tag_configure('odd',
background='#d9d9d9')
```

Це дозволяє легко пояснити важливість того чи іншого коду або принцип його роботи. Часто це необхідно при побудові складних виразів, які людині складно зрозуміти, хоча за можливістю краще розбити їх на більш прості компоненти.

В розробленому застосунку функції мають свої docstring там де це є необхідним для розуміння її задач або ж параметрів. Для кожного необхідного для пояснення обраного рішення рядка написано коментар, що описує причини виконання поставлених задач саме таким чином.

Проведено типізацію даних де це є необхідним задля усунення помилок при роботі з функціями шляхом чітких обмежень параметрів.

Документація користувача є в певній мірі інтегрованою в застосунок шляхом підбору правильних назв кнопок з відповідними функціями та додання інформаційних полів там де це є необхідним

Також створено README.md файл, в якому детально описана інструкція з встановлення розробленого ПЗ на інший ПК.

Висновки до розділу 4

В результаті виконання роботи над четвертим розділом кваліфікаційної роботи бакалавра було описано розробку та реалізацію різних аспектів застосунку, а саме шару роботи з даними та власне інтерфейсу. Надано інформацію про ключові моменти та особливості даних підрозділів.

Наведено необхідні для повного розуміння етапу розробки та кінцевого програмного забезпечення лістинги, а також описано їх важливість та чому вони продемонстровані в розділі, а не в додатках. Пояснено особливості маршрутизації в *Tkinter* та чому було обрано саме такі проєктні рішення.

Продемонстровано кінцевий дизайн застосунку та надано інформацію про кожний із фреймів, які в ньому використовуються

Описано процес успішного тестування програмного забезпечення як зі сторони користувача, так і зі сторони розробника та наведено інформацію про відстеження помилок та повідомлення про них користувачу.

Наведено докладну інформацію про формування документації застосунку як для розробників, так і користувачів. Описано основні інструменти та механізми, які використовувались при її створенні.

ВИСНОВКИ

Результати аналізу досліджень процесу реабілітації пост-інсультних пацієнтів та збору й обробки відповідних даних показують, що впровадження цифрових технологій для оцінки якості життя за різними медико-статистичними методиками (Берга, Фугла-Мейєра, Спілбергера-Ханіна та ін.) може суттєво змінити уявлення про ефективність програм реабілітації пост-інсультних пацієнтів. Отримані результати підтверджують доцільність подальшого розвитку алгоритмів та програмного забезпечення для аналізу масивів великих даних із результатами програм медичної реабілітації для оцінювання впливу реабілітації на якість життя пацієнтів після інсульту.

В ході виконання бакалаврської роботи було розроблено програмне забезпечення для оцінювання реабілітації пост-інсультних пацієнтів шляхом обробки статистичних даних та їх подальшої візуалізації у тому вигляді, що буде зрозумілий людині.

Також описано значущість даної роботи та необхідність розвитку обраної сфери дослідження, яка полягає у збільшенні швидкості оцінювання реабілітації та відповідно створення нових методологій для відновлення організму шляхом подальшого обрання більш результативних методів та вправ.

Для досягнення поставленої мети виконано наступні завдання:

- проведено аналіз предметної області та існуючих аналогів;
- обрано інструменти, необхідні для створення застосунку;
- знайдено та отримано дані для подальшого оброблення;
- визначено та реалізовано основний необхідний функціонал застосунку;
- реалізовано відображення результатів обробки та взаємодію з ними;
- створено документацію.

За результатами спеціального розділу з охорони праці було описано та створено безпечні для людської діяльності умови на робочому місці. Досягнуто це було завдяки докладній обробці питань з шумності приміщення, його освітлення, мікроклімату, опису технік безпеки під-час використання ПК, основних вимог до робочого місця та детального опрацювання питання пожежної безпеки

Робота пройшла апробацію під час Всеукраїнської науково-практичної конференції «Інформаційні технології та інженерія» (Миколаїв, 09–11 лютого 2022 р.).

За результатами бакалаврської роботи опубліковано 1 друковану працю.

Робота складається з вступу, чотирьох фахових розділів, висновків, переліку джерел посилання з 24 джерел, трьох додатків та спеціальної частини з питань охорони праці та безпеки життєдіяльності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Automatic detection of compensation during robotic stroke rehabilitation therapy. 2017. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8214256> (Last accessed: 15.05.2022).
2. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language user guide. 2nd Ed. 2005. 496 p.
3. Dalzell T. The routledge dictionary of modern American slang and unconventional English. 595 p. Also available online, URL: https://books.google.com.ua/books?id=5F-YNZRv-VMC&pg=PA595&redir_esc=y#v=onepage&q&f=false (Last accessed: 11.06.2022).
4. Dolatabadi E., Ye B., Mihailidis A., et al. The Toronto rehab stroke pose dataset to detect compensation during stroke rehabilitation therapy. *Pervasive Health'17* : Proc. of the 11th EAI Intern. Conf., May 23–26, Barcelona, Spain, 2017. DOI:10.1145/3154862.3154925.
5. Harris C. R., Millman K. J., Walt S. J. van der et al. Array programming with NumPy. *Nature*. 2020. Vol. 585, Issue 7825. P. 357–362. DOI:10.1038/s41586-020-2649-2.
6. Instant generator – visual paradigm community circle. URL: <https://circle.visual-paradigm.com/docs/code-engineering/instant-generator/> (Last accessed: 11.06.2022).
7. Ke L., Crotty A.-W. Z., Na L., et al. Telerehabilitation services for stroke (Review). 2020. DOI:10.1002/14651858.CD010255.pub3.
8. León A., Pastor Ó. Enhancing precision medicine: A Big Data-driven approach for the management of genomic data. *Big Data Research*. Vol. 26, 15.11.2021. P. 100253. DOI:10.1016/j.bdr.2021.100253.
9. Martin R. Design principles and design patterns. *Object Mentor*. Issue c.
10. Matplotlib – visualization with Python. URL: <https://matplotlib.org/> (Last accessed: 15.06.2022).

11. MOCA test online. MoCA Duo – Apps on Google Play. MOCA test online. URL: https://play.google.com/store/apps/details?id=com.moca.app&hl=en_CA&gl=US (Last accessed: 20.05.2022).
12. NIH stroke scale/score (NIHSS) – MDCalc. URL: <https://www.mdcalc.com/nih-stroke-scale-score-nihss> (Last accessed: 15.05.2022).
13. Pandas – Python data analysis library. URL: <https://pandas.pydata.org/about/> (Last accessed: 15.06.2022).
14. Python docstrings - GeeksforGeeks. URL: <https://www.geeksforgeeks.org/python-docstrings/> (Last accessed: 22.06.2022).
15. Reenskaug T., Coplien J. O. The DCI Architecture : A new vision of object-oriented programming. *Artima Developer*. 2009.
16. Rehabilitation. URL: <https://www.who.int/news-room/fact-sheets/detail/rehabilitation> (Last accessed: 20.05.2022).
17. tkinter — Python interface to Tcl/Tk — Python 3.10.5 documentation. URL: <https://docs.python.org/3/library/tkinter.html> (Last accessed: 15.06.2022).
18. Unified Modeling Language (UML) description, UML diagram examples, tutorials and reference for all types of UML diagrams - use case diagrams, class, package, component, composite structure diagrams, deployments, activities, interactions, profiles, etc. URL: <https://www.uml-diagrams.org/> (Last accessed: 10.06.2022).
19. User interface – best practices & principles for GUI design. URL: <https://stackoverflow.com/questions/90813/best-practices-principles-for-gui-design> (Last accessed: 11.06.2022).
20. User interface design - some of the best practices to keep in mind -. URL: <https://visualhierarchy.co/blog/user-interface-design-best-practices/> (Last accessed: 11.06.2022).
21. Wafa H. A., Wolfe C. D. A., Emmett E., et al. Burden of stroke in Europe: Thirty-year projections of incidence, prevalence, deaths, and disability-

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів
adjusted life years. *Stroke*. 2020. Vol. 51, Issue 8. P. 2418–2427.
DOI:10.1161/STROKEAHA.120.029606.

22. What is Big Data? | Oracle. URL: <https://www.oracle.com/big-data/what-is-big-data/> (Last accessed: 22.05.2022).

23. Фіник В. Ю., Журавська І. М. Цифрові технології обробки наборів даних для оцінювання реабілітації пост-інсультних пацієнтів. *Інформаційні технології та інженерія : тези Всеукр. наук.-практ. конф., Миколаїв. 9–11 лютого 2022*. С. 140–142.

24. Що таке інсульт та як його розпізнати? URL: <https://moz.gov.ua/article/health/scho-take-insult-ta-jak-jogo-rozpiznati> (Last accessed: 13.05.2022).

ДОДАТОК А

Базовий клас для обробників даних

```

class DataProcessor(DataVisualizer, DataWorker, ABC):
    """
    Base class for data processors.
    """

    @abstractmethod
    def process_data(self):
        pass

    @staticmethod
    def get_median_for_visualize(data: pd.DataFrame, step=None):
        """
        Returns the median of the data.

        :param data: The data.
        :param step: The step.
        :return: The median.
        """
        if step is None:
            step = len(data) // 10
        elif step > len(data) or step < 0:
            raise ValueError(
                "Step must be between 0 and the length of the data.")

        start_index = 0
        end_index = start_index + step

        processed_data = data.apply(np.sum, axis=1)

        medians = []
        while end_index < len(data):
            medians.append(processed_data.iloc[start_index:end_index].median())
            start_index += step
            end_index += step
        medians.append(processed_data.iloc[start_index:].median())

        return medians

    @staticmethod
    def get_average_for_visualize(data: pd.DataFrame, step=None):
        """
        Returns the average of the data.

        :param data: The data.
        :param step: The step.
        :return: The average.

```

```
"""
```

```
if step is None:  
    step = len(data) // 10  
elif step > len(data) or step < 0:  
    raise ValueError(  
        "Step must be between 0 and the length of the data.")  
start_index = 0  
end_index = start_index + step  
  
processed_data = data.apply(np.sum, axis=1)  
  
averages = []  
while end_index < len(data):  
    averages.append(processed_data.iloc[start_index:end_index].mean())  
    start_index += step  
    end_index += step  
averages.append(processed_data.iloc[start_index:].mean())  
  
return averages
```

ДОДАТОК Б**Код класу для роботи з MoCA шкалою**

```

class MoCATestDataProcessor(DataProcessor):
    data = None
    test_min_value = 0
    test_max_value = 30

    def __init__(self, data_path, data_type: str):
        super().__init__(data_path, data_type)
        self.data_path = data_path
        try:
            self.data = pd.read_csv(data_path)
            self.read_result = 'Success'
        except FileNotFoundError:
            self.read_result = 'File not found'
        except pd.errors.ParserError:
            self.read_result = 'File is not a valid CSV file'
        except UnicodeDecodeError:
            self.read_result = 'File is not a valid CSV file'

    def get_data(self):
        return self.data

    def process_data(self):
        return self.data.drop(columns=['Id', 'Rehabilitation Month'])

    def visualize_data(self, step=None):
        if step is None:
            step = len(self.data) // 10
        elif step > len(self.data) or step < 0:
            raise ValueError(
                "Step must be between 0 and the length of the data.")
        processed_data = self.process_data()
        plt.plot(DataProcessor.get_median_for_visualize(processed_data, step),
                 label='Median')
        plt.plot(
            DataProcessor.get_average_for_visualize(processed_data, step),
            label='Average')

        plt.title('MOCA Test Data')

        plt.xlabel('Rehabilitation Time (months)')
        plt.ylabel('Total Score')

        return plt.plot

    def process_patient_data(self, patient_id):
        """

```

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

Processes the data for a specific patient.

:param *patient_id*:

:return: *data to plot and list of rehabilitation rate per month*

"""

```
patient_data = self.data[self.data['Id'] == patient_id]
```

```
rehabilitation_months = patient_data['Rehabilitation Month'].tolist()
```

```
patient_data = patient_data.drop(
    columns=['Id', 'Rehabilitation Month'])
```

```
visualize_dict = dict(zip(rehabilitation_months,
    patient_data.apply(np.sum,
        axis=1).tolist()))
```

```
percents = {rehabilitation_months[i]: round(
    row[1].sum() / self.test_max_value * 100, 2) for i, row in
    enumerate(patient_data.iterrows())}
```

```
visualize_dict = pd.Series(visualize_dict)
```

```
return {
    'data_to_visualize': visualize_dict,
    'rehabilitation_rates': percents
}
```

```
def export_data(self, path: str, data: Union[pd.DataFrame, pd.Series]):
    data.to_csv(path, index=False)
```

Кафедра інженерії програмного забезпечення
Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

ДОДАТОК В
Матеріали апробації роботи

Міністерство освіти і науки України
Чорноморський національний університет
імені Петра Могили



«Інформаційні технології та інженерія»

*Всеукраїнська науково-практична конференція
молодих вчених, аспірантів і студентів*

ТЕЗИ

9–11 лютого 2022 року

Миколаїв – 2022

Розганяєв Д. О., Прищепов О. Ф. Система Electronic Stability Program (ESP) для вітчизняних автомобілів 110

Терещенко Г. Ю., Бєліков О. Є. Автоматизована система видачі лікарських засобів за приписом лікаря з функцією віддаленого сповіщення 111

Тіхоміров К. А., Прищепов О. Ф. Автоматизована система керування безпілотним літаючим апаратом 113

Методи і засоби комп'ютерної інженерії

Важоха В. Б., Крайник Я. М. Комплексна система управління сервісами з використанням засобів контейнеризації 117

Горбуров Л. М., Крайник Я. М. Особливості проектування інтелектуальних систем для дослідження нейронних мереж 118

Дяденко А. С., Бойко А. П. Багатокористувацький серверний застосунок на базі рушія UNREAL ENGINE 120

Методи і засоби програмної інженерії

Голуб Р. Р., Дворецький М. Л. Використання архітектурних-патернів MVVM та MVP при розробці мобільних застосунків мовою Kotlin 123

Гусєва-Божаткіна В. А. Визначення метрик та довірчого інтервалу для побудови регресійного рівняння для оцінювання тривалості розробки програмних застосунків, що розробляються мовою COBOL для платформи MAINFRAME 125

Кандиба І. О., Ткаченко М. П. Аналіз складових телекомунікаційної мережі закладу вищої освіти засобами Python 129

Кожушан М. Г., Кунгурцев О. Б. «Фільтрація тлумачень термінів в електронному словнику 132

Кошовий Р. В., Кірей К. О. Застосунок для оцінки збіжностей файлів із кодом С-подібних мов програмування 134

Тіщенко С. Є., Рудніченко М. Д. Мобільне застосування для обліку особистих коштів 137

Фінік В. Ю., Журавська І. М. Цифрові технології обробки наборів даних для оцінювання реабілітації пост-інсультних пацієнтів 140

Червіцька К. В., Комлева Н. О. Особливості архітектури вебсайту для кафедри системного програмного забезпечення 142

УДК 004.725: 004.056

Фіник В. Ю., Журавська І. М.

*Чорноморський національний університет імені Петра Могили,
м. Миколаїв, Україна*

ЦИФРОВІ ТЕХНОЛОГІЇ ОБРОБКИ НАБОРІВ ДАНИХ ДЛЯ ОЦІНЮВАННЯ РЕАБІЛІТАЦІЇ ПОСТ-ІНСУЛЬТНИХ ПАЦІЄНТІВ

Інсульт є однією з найбільш поширених і тяжких патологій, що характеризується високим ступенем летальності та рівнем інвалідизації населення. Хоч з прогресом медицини у світі рівень смертності внаслідок інсульту стає меншим, але Україна, станом на 2016 рік, посідає 41-е місце за рівнем смертності від інсульту (384 на 100 тис. населення) [1].

Проте, рівень захворюваності на інсульт майже вдвічі перевищує загальний рівень захворюваності в розвинених країнах Європи. При цьому, близько третиною хворих є працездатне населення. Хоча рівень смертності є відносно невеликим, але рівень інвалідизації після перенесення інсульту в Україні становить від 76 % до 85 %, що є надзвичайно сумним, якщо порівнювати з країнами Європи, де даний показник варіюється від 25 % до 30 %.

Така невтішна статистика свідчить про необхідність створення певної системи аналізу, яка могла б оцінювати можливість реабілітації пацієнтів або ж навіть передбачувати ризик інсульту у здорових людей. Наразі не було знайдено прикладів подібного програмного забезпечення, тому резонним буде зробити висновок, що зараз дана відповідальність лежить на плечах лікарів, тож є доцільним створювати засоби полегшення їх роботи в даний час.

Для створення програмного забезпечення із вищезазначеними функціями по-перше необхідно розробити алгоритм для оброблення даних. У якості набору даних у такій розробці використаний DataSet, що зібраний Австралійським відділенням статистики з листопада 2012 р. по червень 2015 р. [3]. Діапазони параметрів, за якими проведений аналіз, наведений у табл. 1. Залежність параметрів руху від швидкості була проаналізована для підгрупи пост-інсультних пацієнтів на реабілітації обсягом $n = 27$ та для підгрупи умовно-здорових людей обсягом $n = 17$.

Таблиця 1. Діапазони параметрів формування DataSet для пост-інсультних пацієнтів на реабілітації

Назва параметру	Середнє значення		Середнє квадратичне відхилення		Діапазон	
	$n = 27$	$n = 17$	$n = 27$	$n = 17$	$n = 27$	$n = 17$
Вік	60,15	54,18	12,08	8,33	28–76	40–74
Обрана швидкість ходьби, м/с	0,73	1,2	0,32	0,19	0,29–1,23	0,75–1,46
Баланс Берга (бали)	47,70	–	6,79	–	25–55	–
Оцінка Фугла-Мейєра	22,85	–	6,95	–	9–34	–
Стать (чол./жін.)	18/9	9/8				

Було запущено процес розробки програмного забезпечення для аналізу даних руху пацієнта і подальшим виведенням медико-статистичних параметрів якості його реабілітації або ж можливості його реабілітації взагалі. Візуалізація результатів аналізу DataSet, що наведена на рис. 1, дозволяє покращити розуміння аспектів, що впливають на можливість реабілітації після інсульту або ж на ризик захворювання.

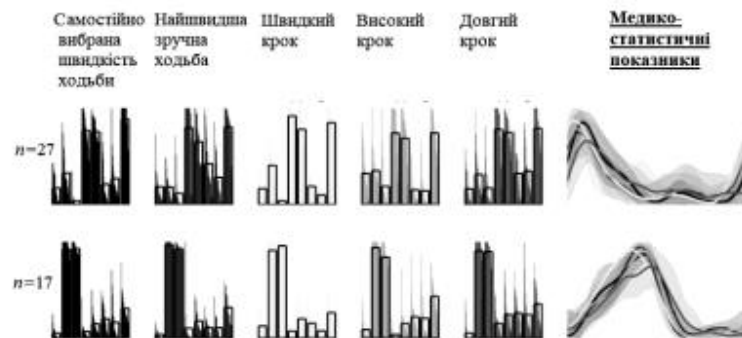


Рисунок 1 – Порівняння медико-статистичних показників пост-інсультних та умовно-здорових пацієнтів в залежності від характеристик руху

Згодом планується вдосконалити систему за допомогою передбачення ризику інсульту шляхом введення необхідних додаткових даних [3].

Таким чином, результати дослідження показують, що впровадження цифрових технологій для оцінки якості життя за різними медико-статистичними методиками (Берга, Фугла-Мейера, Спілбергера-Ханіна та ін.) може суттєво змінити уявлення про ефективність програм реабілітації пост-інсультних пацієнтів. Отримані результати підтверджують доцільність подальшого розвитку алгоритмів та програмного забезпечення для аналізу масивів даних із результатами програм медичної реабілітації для оцінки впливу реабілітації на якість життя пацієнтів після інсульту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Стасенко Т. Інсульт: впровадження успішних стратегій лікування. *Український медичний часопис*. Опубл. 12.01.2017. URL: <https://www.umj.com.ua/article/117650/insult-vprovadzheniya-uspishnih-strategij-likuvannya> (дата звернення: 30.12.2021).
2. Lynch E., Cadilhac D., Hillier S., Luker J. Assessment for rehabilitation after stroke collection / University of South Australia. URL: <https://researchdata.edu.au/assessment-rehabilitation-stroke-collection/966394> (Last accessed: 30.12.2021).
3. Trunov A., Beglytsia V., Gryshchenko G., Ziuzin V., Koshovy V. Methods and tools of formation of general indexes for automation of devices in rehabilitative medicine for post-stroke patients. *Eastern-European Journal of Enterprise Technologies*. 2021. Vol. 4, No. 2 (112). P. 35–46.

УДК 004.42

Червіцька К. В., Комлева Н. О.
Національний університет «Одеська політехніка»,
м. Одеса, Україна

ОСОБЛИВОСТІ АРХІТЕКТУРИ ВЕБСАЙТУ ДЛЯ КАФЕДРИ СИСТЕМНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Наразі будь-який структурний підрозділ у складі закладу вищої освіти повинен мати своє електронне представництво. Це надає можливість користувачам отримувати актуальну інформацію щодо діяльності цього підрозділу, його взаємодії з іншими підрозділами

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

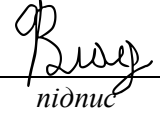
КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОБРОБКИ ДАНИХ ДЛЯ
ОЦІНЮВАННЯ РЕАБІЛІТАЦІЇ ПОСТ-ІНСУЛЬТНИХ
ПАЦІЄНТІВ

СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ
ПИТАННЯ ОХОРОНИ ПРАЦІ РОЗРОБНИКА ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1– 408.21810826

Студент


В. Ю. Фінік
підпис
«__»_____2022р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексєєва
підпис
«__»_____2022р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	3
ВСТУП	4
1 АНАЛІЗ УМОВ ПРАЦІ.....	5
1.1 Аналіз умов праці за шумом.....	5
1.2 Аналіз умов праці за освітленням.....	6
1.3 Аналіз умов праці за мікрокліматом	8
2 ТЕХНІКА БЕЗПЕКИ ПІД ЧАС ВИКОРИСТАННЯ ПК.....	10
3 ОСНОВНІ ВИМОГИ ДО РОБОЧОГО МІСЦЯ	12
4 ПОЖЕЖНА БЕЗПЕКА	13
ВИСНОВКИ.....	14
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	15

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

АУП – автоматична установка пожежогасіння

дБ –децибел

ПК – Персональний комп'ютер

ЦНС – Центральна нервова система

ІТ – Information Technology

ВСТУП

Будь-яке середовище характеризується наявністю потенційних небезпечних і шкідливих чинників, що можуть в тій чи іншій мірі нашкодити людському здоров'ю. Зменшення їх кількості та підвищення загальної як фізичної так і психологічної безпеки людини на виробництві зменшує ризик виникнення критичних ситуацій та сприяє підвищенню загальної продуктивності працівників.

Охорона праці – це наука про збереження здоров'я людини та створення безпечних умов праці у виробничому середовищі, що досягається шляхом виявлення та ідентифікації небезпечних і шкідливих чинників виробничого середовища, розробкою методів і засобів захисту працюючих від їх впливу [3].

Аналіз великого спектру нещасних випадків на різних типах виробництва дозволив зробити висновок, що більшість стається або через невиконання норм безпеки, або через відверто халтурне ставлення до здоров'я працівників працедавцями. Завдяки розвитку людських знань у охороні праці можливо зменшити кількість нещасних випадків.

Попри те, що ІТ вважається однією з найбезпечніших для здоров'я людини галузей, насправді це є не зовсім вірним. Через сидячий спосіб життя та інтенсивну роботу з клавіатурою ризику піддаються одні з найважливіших людських органів та систем: кістково-м'язова система (одним з найбільш поширених захворювань є Тунельний синдром), ЦНС, репродуктивна система, очі. І хоча вплив на організм може бути непомітним довгий час, відновлення після враження цих систем є доволі довгим і складним, а іноді і неможливим. Через це дуже важливим є попередження можливих негативних впливів на організм людини, створення найбільш сприятливих умов для праці, мінімізація шкоди уже наявних чинників.

1 АНАЛІЗ УМОВ ПРАЦІ

1.1 Аналіз умов праці за шумом

Будь-який шум має вплив на продуктивність роботи людини та якість результатів її розумової діяльності. Існують різні види шуму і всі вони по різному можуть впливати на працездатність людини. Наприклад, легка музика або ж звук білого шуму часто може не тільки не відволікати від роботи, а й навіть допомагає працювати більш зосереджено [1]. При цьому не бажано щоб шум перевищував середній допустимий рівень, що складає 65 дБ.

Основними джерелами шуму в ІТ сфері є комп'ютер, кондиціонер, колеги та інші зовнішні чинники.

Таблиця 1.1 – Рівень шуму від різноманітних джерел

Джерело	Рівень шуму
Клавіатура	15
Кондиціонер	35
Миша	10
Охолодження ПК	40
Сторонні чинники	40

Рівень шуму, підраховується за допомогою формули рівня звукового тиску. Формула дає можливість зрозуміти чи є показники допустимими задля роботи у приміщенні.

$$L = 10 \lg \sum_{i=1}^n 10^{0.1 \cdot L_i} \quad (1.1),$$

де L_i – рівень звукового тиску;

n – кількість джерел.

Проведемо усі необхідні підрахунки та визначимо справжній рівень звукового тиску.

$$L = 10 \lg(10^{1.5} + 10^{3.5} + 10^1 + 10^4 + 10^4) = 10 \lg(23203,8) = 43.65 \text{ дБ} \quad (1.2)$$

З результатів розрахунків чітко видно, що рівень шуму не перевищує середнє допустиме значення і відповідає стандартам.

1.2 Аналіз умов праці за освітленням

Освітлення має важливе значення, оскільки завдяки достатній його кількості людина може менше навантажувати очі, через що вони будуть втомлюватись набагато менше. Також недостатня кількість освітлення може призвести до надмірної сонливості та, іноді, збільшити імовірність депресії.

І навпаки: надмірне освітлення теж спричиняє швидку втому очей і може виступати як збудник організму, що може призвести до різних психічних порушень.

Розрахуємо штучне освітлення приміщення у якому було виконано поставлену роботу. Площа приміщення складає 28м^2 , ширина та довжина якої відповідно 4м та 7м , а висота 3м . Skorистаємось відповідною формулою(1.3) [4].

$$F = \frac{E \cdot K \cdot S \cdot Z}{n} \quad (1.3),$$

де F – світловий потік, що розраховується, лм;

E – мінімальна освітленість, лк; за стандартом $E = 300\text{лк}$;

S – площа приміщення (28м^2);

Z – відношення середнього значення освітленості до мінімального ($Z = 1.2$);

K – коефіцієнт запасу, що враховує хибні чинники, що можуть впливати на світловий потік ($K = 1.5$);

n – коефіцієнт використання світлового потоку, що є відношенням світлового потоку яке падає на поверхню; має залежність від усіх характеристик ліхтаря, світильника, лампи, тощо. А також від розмірів приміщення, оздоблення стін і стелі, які можна охарактеризувати спеціалізованими коефіцієнтами відбиття від стін та стелі. $P_{\text{стін}} = 25\%$, $P_{\text{стелі}} = 60\%$.

Також задля аналізу приміщення потрібно обчислити індекс

приміщення за формулою (1.4).

$$I = \frac{S}{h(A+B)} \quad (1.4),$$

де S – площа приміщення (28м²);

h – висота приміщення (3 м);

A – ширина приміщення (4 м);

B – довжина приміщення (7 м);

$$I = \frac{28}{3*(4+7)} = 0.93 \quad (1.5)$$

Спираючись на отримані дані та на табличні значення, можна зазначити, що коефіцієнт становить n = 0,9

Таблиця 1.2 – Значення коефіцієнту використання світлового потоку

СТЕЛЯ	Коефіцієнт відображення, %						Коефіцієнт приміщення i	
	70%				50%			30%
	50%		30%		50%	30%		10%
СТІНИ	30%		10%		30%		10%	КОЕФІЦІЄНТ ВИКОРИСТАННЯ СВІТЛОВОГО ПОТОКУ
ПІДЛОГА	30%	10%	30%	10%	10%			
	0,26	0,25	0,20	0,19	0,17	0,13	0,06	
	0,3	0,28	0,24	0,23	0,2	0,16	0,08	0,6
	0,34	0,32	0,28	0,27	0,22	0,19	0,10	0,7
	0,38	0,36	0,31	0,30	0,24	0,21	0,11	0,8
	0,40	0,38	0,34	0,33	0,26	0,23	0,12	0,9
	0,43	0,41	0,37	0,35	0,28	0,25	0,13	1,0
	0,46	0,43	0,39	0,37	0,30	0,26	0,14	1,1
	0,48	0,46	0,42	0,40	0,32	0,28	0,15	1,25
	0,54	0,49	0,47	0,44	0,34	0,31	0,17	1,5
	0,57	0,52	0,51	0,47	0,36	0,33	0,18	1,75
	0,60	0,54	0,54	0,50	0,38	0,35	0,19	2,0
	0,62	0,56	0,57	0,52	0,39	0,37	0,20	2,25
	0,64	0,58	0,59	0,54	0,40	0,38	0,21	2,5
	0,68	0,60	0,63	0,57	0,42	0,40	0,22	3,0
	0,70	0,62	0,66	0,59	0,43	0,41	0,23	3,5
	0,72	0,64	0,64	0,61	0,45	0,42	0,24	4,0
	0,75	0,66	0,72	0,64	0,46	0,44	0,25	5,0

Наразі є можливість повного розрахунку світлового потоку.

Підставивши необхідні значення до відповідної формули (1.6).

$$F = \frac{E \cdot K \cdot S \cdot Z}{n} = \frac{300 \cdot 1.5 \cdot 28 \cdot 1.2}{0.9} = 16\,800 \quad (1.6)$$

Оскільки лампи які використовуються мають світловий потік 4200Лк. Тому проведемо відповідні розрахунки кількості ламп (1.7).

$$N = \frac{F_1}{F_2} \quad (1.7),$$

де, F_1 – необхідни потік;

F_2 – світловий потік лампи.

$$N = \frac{16\,800}{4200} = 4 \text{ лампи} \quad (1.8)$$

Було проаналізовано умови освітлення приміщення та виявлено, що для відповідної роботи приміщення має містити 4 лампи.

1.3 Аналіз умов праці за мікрокліматом

Мікроклімат виробничих приміщень – це умови внутрішнього середовища цих приміщень, що впливають на тепловий обмін працюючих. Як фактор виробничого середовища він діє певним чином на теплообмін організму людини з цим середовищем і визначає тепловий стан організму людини в процесі праці.

Як правило мікрокліматичні умови характеризуються за такими показниками:

- температура повітря;
- відносна вологість повітря;
- швидкість та інтенсивність теплового опромінювання від поверхонь

Усі показники мікроклімату впливають на продуктивність та працездатність людини, її фізичний та моральний стан. Тривале знаходження в приміщенні з високою температурою та вологістю може призвести до

гіпертемії та надмірної втоми. Також це може слугувати передумовою до більш критичних проблем для людей із слабкою серцево-судинною системою.

Низька температура теж має негативний вплив, оскільки можуть загостритись різні запальовальні процеси, особливо пов'язані з опорно-руховим апаратом.

Оптимальні значення температури повітря, вологості та швидкості руху відображаються у ДСН 3.3.6.042-99. Розглядаючи роботу у сфері інформаційних технологій та її можна віднести до категорій робіт легка 1а та легка 1б.

Таблиця 1.3– Оптимальні значення за ДСН 3.3.6.042-99

Період року	Категорія робіт	Температура повітря	Вологість	Швидкість руху
Холодний	Легка 1а	22-24 °С	60-40 %	0.1 м/с
	Легка 1б	21-23 °С	60-40 %	0.1 м/с
Теплий	Легка 1а	23-25 °С	60-40 %	0.1 м/с
	Легка 1б	22-24 °С	60-40 %	0.2 м/с

Задля вдалого виконання поставлених задач було проведено дослідження середовища приміщення за допомогою різноманітних спеціальних приладів, у результаті було отримано сприятливі значення температури повітря, вологості та швидкості руху. Тим самим було попереджено отримання небажаних хвороб та поганого самопочуття.

2 ТЕХНІКА БЕЗПЕКИ ПІД ЧАС ВИКОРИСТАННЯ ПК

Техніка безпеки – це система організаційних і технічних засобів, які запобігають дії на працівників небезпечних виробничих чинників. У небезпечних зонах постійно діють або періодично виникають чинники, небезпечні для життя і здоров'я людини.

У сьогодення коли ПК став звичним для нас пристроєм, більшість людей нехтують чи навіть не думають наскільки важливою частиною є техніка безпеки при роботі з ПК. При невірному використанні ПК можна отримати травми різного ступеня, що може дуже сильно вплинути на продуктивність та перш за все здоров'я людини. Існують важливі фактори ризику, про які повинні знати користувачі ПК, а саме [5]:

- монотонність процесу;
- навантаження на зоровий аналізатор;
- ураження током;
- погане освітлення;
- занадто яскравий чи блискучий екран;
- підвищений рівень магнітних чи електричних полів.

Перед початком роботи з ПК треба виконувати певні процедури задля техніки безпеки:

- завжди бути впевненим, що усі елементи справні та не можуть привести до небажаних наслідків;
- потрібно бути впевненим у безпечній подачі току до самого ПК та монітору.

Під час роботи з ПК потрібно дотримуватися наступних правил:

- не торкатися мокрими руками працюючий ПК чи монітор;
- не проводити чистку під того, як елементи знаходяться під напругою;

Програмне забезпечення обробки даних для оцінювання реабілітації пост-інсультних пацієнтів

- забороняється класти сторонні предмети на корпус чи екран монітору;
- експлуатувати ПК лише за його призначенням.

Після завершення роботи з ПК потрібно дотримуватися наступних правил:

- вимкнути ПК за усіма інструкціями;
- перевірити чи вимкнувся ПК повністю;
- підготувати ПК до наступного використання, а саме виконати очистку робочого місця від пилу чи іншого бруду.

Також важливими аспектом техніки безпеки при роботі за ПК є наявність комфортних умов роботи. Вони містять у собі наступні правила:

- спиратися ногами на підлогу;
- використання спеціалізованих меблів;
- вірне освітлення.

Важливими рекомендаціями є використання обладнання, а саме окулярів, килимків для миші, підставок для зап'ястя для запобігання тунельного синдрому.

3 ОСНОВНІ ВИМОГИ ДО РОБОЧОГО МІСЦЯ

Будь-яке робоче місце повинно відповідати усім нормам та вимогам чинного законодавства. Антропометричні, фізичні та психологічні вимоги є важливою частиною якості роботи програміста. За рахунок гарного оснащення місця може підвищуватися ефективність роботи від 5% до 25% [2].

Під час організації місця повинні бути виконані наступні умови:

- а) вдало розміщені необхідні пристрої;
- б) наявність достатньої робочої площі;
- в) достатній рівень освітлення;

Головними елементами місця роботи програміста є стіл, стілець та комп'ютер. Оскільки робота зобов'язує більшість часу проводити у положенні сидячи, то слід відповідально відноситися до цього пункту, оскільки погано підібрані елементи можуть приводити не лише до втрати продуктивності, а й до погіршення здоров'я та загального самопочуття працівника. Під час планування робочого місця стіл повинен відповідати наступним умовам:

- а) загальна висота столу повинна бути зручною, щоб була можливість вільно обрати позицію;
- б) під столом програміст не повинен підтискати ноги під себе, загальний обсяг для ніг повинно бути близько 600мм та шириною не менше ніж 500мм;
- в) поверхня столу повинна бути матовою;

Оскільки стілець є доволі важливою частиною робочого місця програміста, то слід зазначити головні вимоги, щодо підбору та його використання. Загальна висота сидіння повинна бути близько 420-550мм. Крісло повинно бути м'яким, а також мати заокруглений кінець та спинку, яка надає змогу регулювання.

4 ПОЖЕЖНА БЕЗПЕКА

У сучасних комп'ютерах дуже висока щільність розміщення елементів електронних систем: в безпосередній близькості один від одного розташовуються сполучні дроти, комунікаційні кабелі, що може посприяти небезпеці різного роду. При цьому можливі оплавлення ізоляції сполучних проводів, їх оголення і, як наслідок, коротке замикання, яке супроводжується іскрінням, яке веде до неприпустимих перевантажень елементів електронних схем. При перенагріванні, вони згорають з розбризкуванням іскор.

Системи вентиляції служать для відводу надлишкового тепла від ЕОМ. Однак ці системи також представляють додаткову пожежну небезпеку для машинного залу та інших приміщень, так як, з одного боку, вентиляція забезпечує подачу кисню, що є окислювачем, в усі приміщення, а з іншого – при виникненні пожежі швидко поширюють вогонь і продукти горіння по всіх приміщеннях і пристроїв, з якими вони пов'язані. Струм до електроустановок подається по кабельних лініях, які становлять особливу пожежну небезпеку. Наявність пального ізоляційного матеріалу, ймовірних джерел запалювання в вигляді електричних іскор і дуг, розгалуженість і важко доступність роблять кабельні лінії місцями найбільш ймовірного виникнення і розвитку пожежі.

У разі пожежі спрацьовує знаходиться в приміщеннях АУП. Найчастіше застосовуються газові АУП, оскільки вони забезпечені світловою та звуковою сигналізацією. Для запобігання поширенню вогню під час пожежі з однієї частини будівлі на іншу влаштовують протипожежні перепони у вигляді протипожежних стін, перегородок, перекриттів, зон, тамбурів-шлюзів, дверей, вікон, люків, клапанів. У будівлі на випадок виникнення пожежі передбачається не менше двох евакуаційних виходів.

ВИСНОВКИ

В результаті написання спеціального розділу з охорони праці кваліфікаційної роботи бакалавра було проаналізовано необхідні умови і норми для комфортної і продуктивної роботи на робочому місці. Виконано дослідження з перевірки відповідності робочого приміщення до різних норм, а саме:

- а) норм шуму;
- б) норм освітлення;
- в) норм мікроклімату.

Було обґрунтовано та продемонстровано те, що приміщення відповідає усім зазначеним нормам та є сприятливим для комфортної та продуктивної роботи.

Описано техніку безпеки під час користуванням ПК, в результаті чого очікується підвищення комфортабельності та безпеки місця роботи.

Також надано інформацію щодо пожежної безпеки в приміщенні та наведено основні вимоги, які є необхідними для обладнання зручного і безпечного робочого місця

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Jeon M. K., Oh J. W. Study on listening to white noise of nursing college students and improvement of concentration. *Medico-Legal Update*. Vol. 19, Issue 1. DOI:10.5958/0974-1283.2019.00061.6.
2. Вимоги до робочого місця програміста - Заходи з охорони праці інженера-програміста. URL: https://vuzlit.com/87295/trebovaniya_rabochemu_mestu_programmista (дата звернення: 15.06.2022).
3. Жидецький В. Ц, Джигирей В. С., Мельников О. В. Основи охорони праці. Львів : Афіша, 2000.
4. Калькулятори розрахунку освітленості приміщення – Ekobil. URL: <http://ekobil.com.ua/kalkulyatori-rozrahunku-osvitlenosti-primishheniya/> (дата звернення: 14.06.2022).
5. Правила безпечної роботи на комп'ютері. URL: <https://pedcollege.kiev.ua/index.php/77-robota-koledzhu/okhorona-pratsi/589-pravyla-bezpechnoi-roboty-na-kompiuteri> (дата звернення: 14.06.2022).