МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра «Систем автоматизованого проєктування»

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської кваліфікаційної роботи на тему:

«Розробка інформаційної системи для визначення текстів українською мовою, створених штучним інтелектом»

Студента групи КН-412

Заверчука Андрія Миколайовича

Керівник роботи ____________________ **(Тарас ЧЕРНА)**
Консультанти ____________________ **(Василь ІВАНИНА)**
Рецензент ____________________ **(Тетяна СТЕФАНОВИЧ)**
____________________ **( )**
____________________ **( )**

Завідувач кафедри САП ______________________

«____» ________________ 20 ___ р.

ЛЬВІВ – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп’ютерних наук та інформаційних технологій
Кафедра «Систем автоматизованого проектування»
Спеціальність 122 «Комп’ютерні науки»

ЗАТВЕРДЖУЮ:
Завідувач кафедри САП ___________________
«____» ________________2025 р.

ЗАВДАННЯ

на бакалаврську кваліфікаційну роботу студента групи КН-412

Заверчука Андрія Миколайовича

1.Тема роботи «Розробка інформаційної системи для визначення текстів українською мовою, створених штучним інтелектом»
затверджена наказом по НУ «ЛП» від «   » _________ 2025 р. № _________
2. Термін здачі студентом закінченої роботи ______________________________

3. Вихідні дані для роботи:
1. Літературні джерела;
2. Вимоги до програмної реалізації;
3. Стандарти структурного проєктування інформаційних систем.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які належить розробити):
1. Огляд літературних джерел;
2. Системний аналіз об’єкту дослідження;
3. Вибір методів і засобів розв’язання завдання;
4. Практична реалізація;

5. Перелік графічного матеріалу:
1. Архітектура компонентів системи;
2. Діаграми об’єктно-орієнтованого проєктування;
3. Схема БД;
4. Рисунки проєктованої системи.

6. Перелік програмних продуктів, які належить використати в процесі розроблення роботи (проєкту):

Python, GPT, BERT, C#

7. Консультування роботи, із зазначенням розділів роботи

Розділ	Консультанти	Підпис, дата	
		завдання видав	завдання отримав

8. Дата, коли видано завдання ______________________________

Керівник Тарас ЧЕРНА______
(підпис)

Завдання отримав Андрій ЗАВЕРЧУК
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Проведення огляду літературних джерел	10.03.25р.	
2.	Аналіз існуючих систем	10.03.25р.-17.03.25р.	
3.	Змістовна постановка задачі	17.03.25р.-24.03.25р.	
4.	Вибір парадигми програмування	24.03.25р.-07.04.25р.	
5.	Вибір програмних засобів	07.04.25р.- 14.04.25р.	
6.	Розробка структури та алгоритму	14.04.25р.-21.04.25р.	
7.	Програмна реалізація	21.04.25р.-28.04.25р.	
8.	Відлагодження програми	28.04.25р.-05.05.25р.	
9.	Оформлення пояснювальної записки	05.05.25р.-12.05.25р.	

Студент ____________________/Андрій ЗАВЕРЧУК/
(підпис)

Керівник роботи ____________________/ Тарас ЧЕРНА/
(підпис)

АНОТАЦІЯ

ПІБ студента: Заверчук Андрій Миколайович

Керівник: Черна Тарас Ігорович

Назва роботи: «Розробка інформаційної системи для визначення текстів українською мовою, створених штучним інтелектом»

Бакалаврська кваліфікаційна робота

Спеціальність – 122 «Комп’ютерні науки»

Заклад освіти – Національний університет «Львівська політехніка», кафедра систем автоматизованого проектування

Рік виконання – Львів, 2025

Робота складається з 74 сторінок, 12 рисунків, 10 таблиць, 1 додатка, 5 плакатів. Використано 14 джерел інформації.

Метою бакалаврської кваліфікаційної роботи є створення інформаційної системи, здатної автоматично аналізувати та визначати ймовірність того, що текст українською мовою був згенерований штучним інтелектом.

Основні завдання роботи:

- Проаналізувати сучасні підходи до виявлення текстів, створених штучним інтелектом.
- Вивчити моделі машинного навчання, зокрема трансформери, що використовуються для генерації та класифікації текстів.
- Дослідити лінгвістичні ознаки, характерні для текстів, створених штучним інтелектом.
- Розробити алгоритм виявлення штучно згенерованих текстів українською мовою.
- Реалізувати інформаційну систему, що інтегрує алгоритм виявлення з користувацьким інтерфейсом.

- Провести тестування системи та оцінити її точність, продуктивність і надійність.
- Проаналізувати потенційні напрями застосування розробленої системи в освіті, журналістиці, академічному середовищі та сфері безпеки.

Предметом дослідження є технології обробки природної мови (NLP), методи класифікації текстів і штучний інтелект.

У роботі було проведено аналіз відомих моделей штучного інтелекту (GPT, BERT, RoBERTa) та підходів до виявлення штучного походження тексту, досліджено специфіку української мови в контексті NLP. Створено програмне забезпечення, яке дозволяє автоматично аналізувати текст та визначати його ймовірне походження. Система інтегрує алгоритми лінгвістичного аналізу, векторизації, класифікації на основі моделей машинного навчання.

Результати дослідження: розроблено інформаційну систему, здатну з високою точністю визначати тексти, створені за допомогою ШІ. Проведено тестування на корпусах природних та згенерованих текстів українською мовою, отримано точність понад 85%. Розглянуто можливість масштабування системи на інші мови та застосування у різних сферах.

Практична цінність: запропонована система може використовуватись у сфері освіти для виявлення плагіату, в журналістиці для перевірки автентичності текстів, у кібербезпеці для виявлення дезінформації, а також як інструмент для наукових досліджень у сфері ШІ та лінгвістики.

Ключові переваги:

• Висока точність виявлення згенерованого тексту

• Підтримка української мови

• Можливість інтеграції з веб-платформами та системами перевірки контенту

• Відкритий код для адаптації під різні потреби

Ключові слова: штучний інтелект, згенерований текст, українська мова, NLP, класифікація, машинне навчання, GPT, виявлення фейків, інформаційна система.

ABSTRACT

Student's Name: Zaverchuk Andrii

Supervisor: Cherna Taras

Title: *Development of an Information System for Detecting Ukrainian Texts Generated by Artificial Intelligence*

Bachelor's Qualification Work

Specialty – 122 "Computer Science"

Educational Institution – Lviv Polytechnic National University, Department of Automated Design Systems

Year of Completion – Lviv, 2025

The thesis consists of 74 pages, 12 figures, 10 tables, 1 appendix, and 5 posters. 14 information sources were used.

The **objective** of this bachelor's qualification work is to develop an information system capable of automatically analyzing and determining the likelihood that a Ukrainian-language text was generated by artificial intelligence.

Main tasks of the work include:

• Analyzing current approaches to detecting AI-generated texts.

• Studying machine learning models, particularly transformers, used for text generation and classification.

• Investigating linguistic features typical of AI-generated texts.

• Developing an algorithm for detecting AI-generated texts in Ukrainian.

• Implementing an information system that integrates the detection algorithm with a user interface.

• Conducting system testing and evaluating its accuracy, performance, and reliability.

• Analyzing potential applications of the developed system in education, journalism, academia, and security.

The **subject of the research** includes natural language processing (NLP) technologies, text classification methods, and artificial intelligence.

This study analyzes well-known AI models (GPT, BERT, RoBERTa) and approaches to identifying artificially generated texts. It also explores the specifics of the Ukrainian language in the context of NLP. A software solution was developed to automatically analyze a text and determine its likely origin. The system integrates linguistic analysis, vectorization, and classification algorithms based on machine learning models.

Research results: an information system was developed that accurately identifies texts generated by AI. Testing on corpora of natural and generated Ukrainian texts showed accuracy above 85%. The possibility of scaling the system to other languages and applying it in various domains was also explored.

Practical value: the proposed system can be used in education to detect plagiarism, in journalism to verify content authenticity, in cybersecurity to detect disinformation, and as a tool for research in AI and linguistics.

Key advantages:

• High accuracy in detecting generated texts

• Support for the Ukrainian language

• Integration with web platforms and content verification systems

• Open-source code for adaptation to different needs

Keywords: artificial intelligence, generated text, Ukrainian language, NLP, classification, machine learning, GPT, fake detection, information system.

ЗМІСТ

ВСТУП

У сучасному світі інновації стали ключовим фактором розвитку технологій, бізнесу та суспільства. Вони проникають у всі аспекти життя, відкриваючи нові можливості для вирішення складних задач. Особливо важливою є роль інноваційних інформаційних технологій, які дозволяють автоматизувати процеси, підвищувати ефективність та створювати продукти, що відповідають актуальним потребам часу.

Одним із викликів сучасного інформаційного суспільства є зростання обсягу текстів, створених за допомогою штучного інтелекту (ШІ). Це породжує питання щодо визначення походження контенту, перевірки його автентичності та оцінки якості. Відповіддю на цей виклик є розробка інноваційного веб-додатку для визначення ШІ-контенту в текстах українською мовою. Такий продукт дозволить аналізувати тексти, виявляти їхнє походження та оцінювати їхню стилістичну відповідність.

Дана робота присвячена дослідженню інноваційних аспектів створення такого додатку, починаючи від концептуальних основ інновацій до моделювання процесів і розробки прототипу. У першому розділі розглянуто основи інновацій, їх класифікацію, особливості в контексті систем автоматизованого проектування (САП), а також креативні підходи до генерування інженерних ідей. Окрім цього, описано концепцію створення інноваційного продукту.

Другий розділ зосереджено на моделюванні об’єктів і процесів, що складають основу інноваційного веб-додатку. Увагу приділено моделюванню предметної області, аналізу вимог до продукту, а також проектуванню алгоритмів для його функціонування.

Третій розділ присвячено розробленню прототипу та оцінці інноваційного потенціалу концепції. Проведено оцінку новизни продукту, визначено його специфікації та бізнесову доцільність. За допомогою матриці інноваційного потенціалу проведено комплексний аналіз продукту, що дозволило зробити висновки щодо його перспективності.

Четвернитий розділ присвячений розробленню прототипу інноваційного продукту та оцінці його інноваційного потенціалу. Тестування прототипу підтвердило його функціональність і відповідність заданим вимогам, а оцінка новизни, інноваційних специфікацій та бізнесової доцільності засвідчила високу конкурентоспроможність продукту.

Робота спрямована не лише на обґрунтування теоретичних аспектів створення інноваційного продукту, але й на практичну реалізацію прототипу, що має значний потенціал для впровадження у сфері освіти, бізнесу та медіа.

РОЗДІЛ 1. Суть та види інновацій

1.1. Застосування штучного інтелекту в аналізі даних і моделюванні

Штучний інтелект (ШІ) став важливим інструментом у САП, адже він дозволяє автоматизувати процеси аналізу великих обсягів даних і створення оптимальних моделей.

Застосування ШІ в САП:

- Аналіз проєктів для виявлення помилок на ранніх етапах.
- Генеративний дизайн, який дозволяє створювати кілька варіантів проєкту на основі заданих параметрів.
- Прогнозування поведінки матеріалів під час експлуатації.
- Наприклад, компанія Autodesk розробила інструмент Generative Design, який використовує алгоритми ШІ для створення оптимізованих конструкцій, враховуючи такі фактори, як вага, міцність і витрати.

Приклади інноваційних рішень у САП

1.2. Інтеграція з хмарними сервісами

- Хмарні технології дозволяють командам працювати над проєктами спільно, незалежно від їхнього фізичного розташування.
- Переваги: миттєвий доступ до оновлених даних, можливість роботи в реальному часі, зменшення витрат на зберігання локальних даних.
- Приклад: платформи, такі як Onshape або Fusion 360, пропонують повну інтеграцію з хмарними сервісами.

1. **Адаптивні інтерфейси**

- Сучасні САП використовують адаптивні інтерфейси, які підлаштовуються під потреби користувачів і складність завдань.
- Переваги: підвищення зручності роботи та швидкості освоєння програмного забезпечення.
- Приклад: AutoCAD використовує адаптивні панелі інструментів, які відображають лише ті функції, що актуальні для конкретного завдання.

1.3. Автоматизація обробки великих даних

Інноваційні рішення в САП включають автоматизацію аналізу великих обсягів інженерних даних для виявлення закономірностей та оптимізації процесів.

Переваги: зменшення часу, необхідного для аналізу, і підвищення точності прийняття рішень.

Приклад: використання MATLAB для моделювання складних фізичних явищ і аналізу даних з датчиків у реальному часі.

Інновації у сфері САП дозволяють оптимізувати процеси розробки, покращувати якість проєктів та знижувати витрати. Завдяки автоматизації, хмарним технологіям, штучному інтелекту та адаптивним інтерфейсам інженери отримують ефективні інструменти, які підвищують їх продуктивність та дозволяють створювати сучасні продукти з мінімальними ризиками.

1.4. Креативний підхід та генерування інноваційних інженерних ідей

Творче мислення є основою для інноваційних рішень у різних галузях, включаючи інженерію, дизайн і бізнес. Воно допомагає генерувати унікальні ідеї, вирішувати складні завдання та створювати нові продукти чи послуги. В основі творчого мислення лежать спеціальні методи й алгоритми, які дозволяють структурувати процес пошуку рішень. Серед них особливе місце займають дизайн-мислення, мозковий штурм, метод SCAMPER, а також алгоритми генерування ідей.

Дизайн-мислення — це методологія, орієнтована на створення рішень, які відповідають потребам користувачів [4]. Вона складається з кількох етапів:

- **Емпатія**: розуміння потреб і проблем користувачів через спостереження, інтерв'ю чи опитування.
- **Формулювання проблеми**: чітке визначення задачі, яку потрібно вирішити.
- **Ідеація**: генерація ідей за допомогою мозкового штурму чи інших технік.
- **Прототипування**: створення швидких моделей рішень для перевірки їхньої життєздатності.

- **Тестування**: отримання зворотного зв’язку від користувачів для подальшого вдосконалення. Дизайн-мислення використовується в інноваційних проєктах для створення продуктів, що максимально відповідають потребам клієнтів.

Мозковий штурм - це популярна техніка генерації ідей, яка базується на співпраці групи людей [4]. Основні принципи:

- Генерація максимальної кількості ідей без критики на початковому етапі.
- Комбінування й вдосконалення ідей, запропонованих учасниками.
- Фокус на конкретній проблемі чи задачі. Наприклад, мозковий штурм часто використовується для розробки концепцій нових продуктів або маркетингових кампаній.

Метод SCAMPER (абревіатура від Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Rearrange) — це метод, який допомагає розглядати проблему з різних перспектив [5]:

- Substitute: замінити елементи продукту чи процесу на альтернативні.
- Combine: об'єднати ідеї, матеріали чи процеси.
- Adapt: пристосувати існуючі рішення для нових умов.
- Modify: змінити форму, структуру чи властивості.
- Put to another use: знайти нове застосування для існуючого продукту.
- Eliminate: прибрати непотрібні елементи чи функції.
- Rearrange: змінити послідовність дій або компонентів.

SCAMPER сприяє творчому мисленню, дозволяючи знаходити інноваційні рішення для звичних завдань.

Алгоритми генерування ідей

1. Формування проблеми

Визначення проблеми є першочерговим етапом у процесі творчого мислення. Правильно сформульована проблема полегшує пошук її рішень. Наприклад, замість формулювання "Як зробити продукт кращим?" доцільніше запитати "Як скоротити час, необхідний для його використання?".

2. Пошук аналогів

Цей етап передбачає вивчення існуючих рішень у суміжних галузях. Наприклад, для створення інноваційної упаковки можна дослідити дизайн, використовуваний у природі, як це зробили розробники "липучок" на основі структури насіння лопуха. Аналоги допомагають знаходити неочевидні шляхи вирішення проблеми.

3. Розробка прототипів

Прототипування дозволяє швидко перевірити життєздатність ідеї. Прототипи можуть бути як простими ескізами, так і детальними моделями. Основна мета цього етапу — зібрати зворотний зв'язок і вдосконалити продукт до його фінальної реалізації.

Використання методів творчого мислення, таких як дизайн-мислення, мозковий штурм та SCAMPER, разом із алгоритмами генерування ідей, допомагає не лише знаходити ефективні рішення, але й робить процес розробки інновацій більш організованим. Це забезпечує високу якість рішень, їхню відповідність потребам користувачів та економію часу на етапі реалізації.

Інноваційні інженерні рішення є ключовим фактором для розвитку технологій, які вирішують складні проблеми у різних галузях. Вони базуються на використанні сучасних підходів, таких як міждисциплінарність, автоматизація за допомогою штучного інтелекту (ШІ), а також створення продуктів, що відповідають запитам часу.

Інтердисциплінарний підхід полягає у поєднанні знань з різних галузей науки та техніки для створення комплексних рішень. Такий підхід дозволяє вирішувати складні задачі, які не під силу одному фахівцю чи одній галузі.

Особливості:

- Інтеграція технологічних, соціальних і наукових дисциплін.
- Співпраця інженерів, дизайнерів, програмістів, соціологів та інших спеціалістів.

- Використання знань із суміжних галузей, наприклад, біології у створенні роботів, що імітують рухи живих організмів.

Прикладом є розробка автономних транспортних засобів, де поєднуються знання в інженерії, програмуванні, аналізі даних та етиці для створення безпечних та ефективних рішень.

Застосування штучного інтелекту для автоматизації процесів розробки

Штучний інтелект є одним із найперспективніших інструментів для автоматизації інженерних процесів. Він дозволяє зменшити обсяг ручної праці, прискорити розробку продуктів і підвищити точність виконання завдань.

Застосування ШІ:

Оптимізація конструкцій за допомогою генеративного дизайну, де алгоритми ШІ пропонують кілька варіантів рішень на основі заданих параметрів.

Автоматичний аналіз великих обсягів даних для виявлення закономірностей і помилок.

Симуляція та прогнозування поведінки продуктів у реальних умовах експлуатації.

Наприклад, використання ШІ у проєктуванні дозволяє автоматично створювати креслення, визначати оптимальні параметри деталей і навіть прогнозувати можливі поломки ще до початку виробництва.

1.5. Опис та обґрунтування інноваційного продукту

У процесі мозкового штурму було розглянуто ключові аспекти розробки інноваційного продукту:

1. Розширення функціоналу

AI запропонував низку важливих ідей для реалізації та подальшого розвитку:

- На початковому етапі розробити прототип бінарного висновку.
- Введення ймовірнісного висновку для аналізу тексту (наприклад, 80% ШІ, 20% людина).

- Додавання пояснень до результатів аналізу, щоб користувач розумів, які частини тексту найбільше вплинули на висновок.
- Підтримка аналізу текстів різних жанрів і стилів (наукові статті, публіцистика тощо).
- Інтеграція з API для спрощення взаємодії з іншими програмами.
- Інтерактивний інтерфейс для швидкої перевірки тексту у реальному часі.
- Детальний звіт із можливістю зберігання на пристрій.

2. Ключові алгоритми для основи моделі

Під час мозкового штурму були запропоновані такі алгоритми:

- Transformer-моделі (BERT, GPT) для високої точності й адаптації до контексту.
- Random Forest/XGBoost для нелінійного аналізу.
- RNN або LSTM для роботи з послідовними даними, якщо це буде потрібно.

3. Вирішення ключових проблем

Основними викликами були:

- Якість даних: потрібно збалансувати "ШІ-тексти" та "людські тексти".
- Точність моделі: пошук балансу між продуктивністю й точністю.
- Етичні аспекти: забезпечення відповідності стандартам конфіденційності.

4. Розвиток продукту та інтеграція з українською мовою

Українська мова стане основною цільовою мовою, що дозволить зібрати великий набір даних для якісного навчання моделі.

5. Візуалізація результатів

- Графічний інтерфейс із показом ймовірності впливу ШІ.
- Звітність у вигляді документу.
- Підсвітка підозрілих частин тексту.

6. Автоматизація збору даних

Використання веб-скрейпінгу, API інтеграцій та краудсорсингу для збору якісних даних.

Опис концепції продукту

UA AI Detection Tool - це інноваційний веб - застосунок, що використовує технології машинного навчання для аналізу текстів українською мовою. Його основною метою є виявлення впливу ШІ на текст, забезпечення прозорості використання штучного інтелекту та підвищення безпеки користувачів в Інтернеті.

Характеристики продукту та вигоди

1. Глибокий аналіз тексту:

Інструмент здатен аналізувати тексти різних жанрів і джерел, визначаючи ймовірність генерації ШІ.

2. Зменшення дезінформації:

Користувач отримує детальний звіт, що допомагає уникнути маніпуляцій.

3. Навчання на зворотному зв'язку:

Платформа враховує внесені користувачем корективи для покращення моделі.

4. Простота використання:

Інтуїтивно зрозумілий інтерфейс із мінімумом дій для аналізу тексту.

Сфери застосування

1. Кібербезпека:

Продукт допомагає визначати потенційно небезпечні тексти, створені ШІ, з метою захисту користувачів.

2. Наукова діяльність:

Визначення рівня оригінальності наукових текстів і боротьба з плагіатом.

3. Журналістика:

Допомагає журналістам аналізувати джерела інформації, підвищуючи довіру до матеріалів.

Особливості дизайну

- Інтерфейс: Просте поле вводу тексту, кнопка аналізу та інтерактивні індикатори.
- Звіт: Графічне представлення результатів із поясненням висновків.
- Технології: Використання найновіших трансформерних моделей та інтеграція через API.

Технічні параметри

- Використання хмарних обчислень для продуктивності.
- Мікросервісна архітектура для розширюваності продукту.
- Оптимізація роботи з текстами великого об’єму.

Існуюча проблема: зростання кількості текстів, створених ШІ, та необхідність їх верифікації.

Рішення, що пропонується: веб-додаток для аналізу текстів, визначення походження (ШІ чи людина), оцінка оригінальності.

1. **Аспекти інноваційності продукту**

- Використання передових моделей NLP (GPT, BERT).
- Адаптація для української мови, врахування граматичних, стилістичних особливостей.
- Новий набір даних, актуальний для 2024 року
- Використання найбільш оптимізованих концепцій та алгоритмів
- Інтеграція з іншими сервісами: освітні платформи, антиплагіат-системи.

2. **Очікувані результати**

- Підвищення довіри до текстових матеріалів.
- Поширення практики етичного використання ШІ.

ВИСНОВКИ до розділу 1

В першому розділі дипломної роботи розкрито сутність інновацій та їх значення в контексті систем автоматизованого проектування (САП), визначаючи поняття "інновація" та "інноваційні інформаційні технології" як ключові для створення конкурентоспроможних продуктів. Аналіз різновидів інновацій показав їх специфіку в САП, де креативний підхід та генерування інженерних ідей відіграють вирішальну роль у розробці нових рішень. Обґрунтовано концепцію інноваційного продукту, яка базується на інтеграції сучасних технологій та творчого мислення, що забезпечує його новизну та практичну цінність.

РОЗДІЛ 2. Моделювання об'єктів та процесів.

2.1. Проєктування об'єктів предметної сфери інноваційної розробки.

Почнемо другий розділ із визначення та специфікації класів об'єктів, необхідних для функціонування інноваційного продукту. Це включає аналіз функціональності продукту, ідентифікацію фізичних та програмних компонентів, класифікацію компонентів за функціями, визначення взаємозв'язків між класами та моделювання структури продукту у вигляді UML-діаграми [8].

Таблиця 1. Основні класи, методи та атрибути для продукту.

Назва класу	Призначення класу	Назва атрибуту	Зміст атрибуту	Назва методу	Зміст методу
TextDetectionApp	Головний клас програми, що координує робочий процес для виявлення AI-згенерованого або зміненого тексту		Інтерфейс користувача		
		processor	Процесор тексту		
		classifier	Класифікатор		
		highlightManager	Менеджер підсвітки		
UserInterface	Керує взаємодією з користувачем, включаючи збір введеного тексту та відображення результатів			gatherInput()	Збирає текст від користувача
				displayResults()	Відображає результати аналізу
				highlightAITextSections()	Підсвічує AI-згенеровані або змінені розділи тексту
TextProcessor	Підготовка тексту для класифікації, очищення та форматування, а також виявлення можливих змін у тексті			preprocessText()	Очищує і форматує текст для класифікації
				detectModifications()	Виявляє зміни в тексті

	Ядро AI-моделі для аналізу тексту, завантаження моделі, створення прогнозів та виявлення відсотку AI-контенту			loadModel()	Завантажує AI-модель
				makePrediction()	Робить передбачення щодо тексту
Classifier				detectAIPercentage()	Визначає відсоток AI-згенерованого контенту
	Абстрагує основну функціональність AI-моделі, включаючи навчання, передбачення та оцінку			train()	Виконує навчання AI-моделі
				predict()	Передбачає результати аналізу
Model				evaluate()	Оцінює ефективність AI-моделі
	Керує набором даних для навчання та тестування AI-моделі			loadData()	Завантажує дані для аналізу
				splitData()	Розділяє дані на тренувальні та тестові набори
Dataset				manageSamples()	Керує зразками даних
LanguageModel	Застосовує мовні правила для української мови			applyUkrainianRules()	Застосовує мовні правила для української мови
	Виявляє зміни у тексті шляхом порівняння оригіналу та обробленого тексту			compareOriginalAndProcessed()	Порівнює оригінальний та оброблений текст
ModificationDetector				calculateModificationScore()	Обчислює рівень модифікації
AITextDetector	Аналізує розділи тексту для визначення ймовірності генерування AI			analyzeTextSections()	Аналізує текстові розділи

				calculateAIGeneratedPercentage()	Обчислює ймовірність AI-згенерованого тексту
HighlightManager	Застосовує візуальні підсвітки до текстових розділів, визначених як AI-згенеровані або змінені			applyHighlights()	Підсвічує текст для візуального виділення
				manageStyles()	Налаштовує стилі підсвітки для відображення користувачеві

Таблиця 2. Відношення між класами.

Назва відношення	**Класи, між якими визначено відношення**	**Тип відношення**	**Множинність відношення**	**Фізична/Системна складова**
Відношення_1	TextDetectionApp, UserInterface	Ассоціація	Один до одного	Системна
Відношення_2	TextDetectionApp, TextProcessor	Ассоціація	Один до одного	Системна
Відношення_3	TextDetectionApp, Classifier	Ассоціація	Один до одного	Системна
Відношення_4	TextDetectionApp, HighlightManager	Ассоціація	Один до одного	Системна
Відношення_5	Classifier, Model	Ассоціація	Один до одного	Системна
Відношення_6	Model, Dataset	Ассоціація	Один до багатьох	Фізична
Відношення_7	TextProcessor, ModificationDetector	Ассоціація	Один до одного	Системна
Відношення_8	TextProcessor, LanguageModel	Ассоціація	Один до одного	Системна
Відношення_9	Classifier, AITextDetector	Ассоціація	Один до одного	Системна

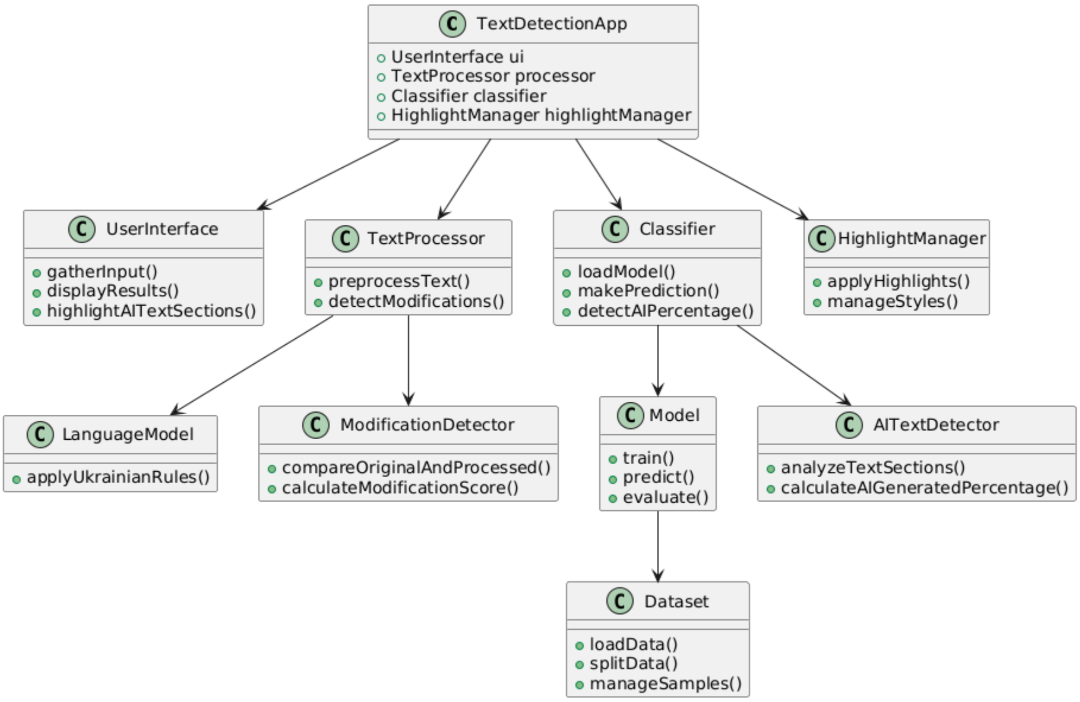


Рис. 2.1. UML діаграма класів

2.2. Визначення вимог до інноваційної розробки.

Перейдемо до опису основних процесів та підпроцесів, що лежать в основі функціонування інноваційного продукту/системи для визначення впливу штучного інтелекту на текстові дані.

Для досягнення цієї мети необхідно побудувати UML-діаграму діяльності, яка наочно відображає основний процес та ключові підпроцеси продукту, а також розробити UML-діаграми варіантів використання (Use Case Diagrams) для демонстрації взаємодії користувача із системою [9].

Це дозволить провести специфікацію вимог до системи, розкриття ключових функцій та аналіз можливостей застосування штучного інтелекту для поліпшення обробки текстових даних.

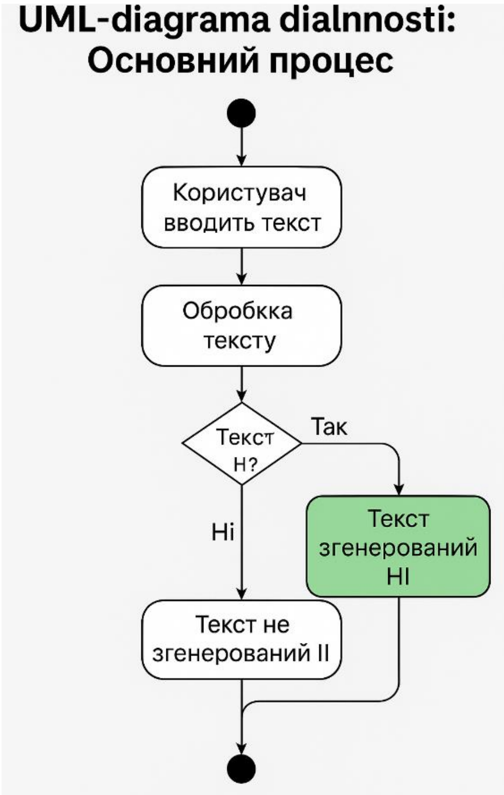


Рис. 2.2. UML-діаграма діяльності, яка описує основний процес.

Процес роботи додатку починається з того, що користувач вводить текст, який хоче проаналізувати. Після цього додаток завантажує простий інтерфейс, забезпечуючи якісну взаємодію з користувачем. Перед тим, як перейти до фактичного аналізу, додаток виконує перевірку ресурсів, щоб підтвердити, що є достатньо пам'яті та обчислювальної потужності. Якщо ресурсів достатньо, додаток продовжує аналіз. Однак, якщо ресурсів недостатньо, користувач отримує сповіщення про обмеження, і процес зупиняється.

Припускаючи, що ресурсів достатньо, додаток ініціює аналіз тексту. Потім модель виконує аналіз контенту в реальному часі, оцінюючи кожну частину для виявлення фрагментів, створених штучним інтелектом. Під час прогресу аналізу результати відображаються користувачеві в реальному часі. Після завершення цього аналізу, йде перевірка наявності будь-яких кроків, що залишились. Якщо аналіз був успішним, додаток виділяє частини тексту, які були визначені як створені штучним інтелектом. Якщо аналіз не був успішним, користувач отримує повідомлення про помилку.

Після виділення фрагментів тексту, згенерованих ШІ, додаток створює всебічний звіт, який відображається користувачу. Цей звіт надає детальний опис результатів, допомагаючи користувачеві зрозуміти, які розділи були позначені і чому. Нарешті, додаток реєструє запит користувача для ведення обліку, що дозволяє звертатися до нього в майбутньому або проводити подальший аналіз за потреби. Цей робочий процес забезпечує безперебійну та ефективну взаємодію, супроводжуючи користувачів на кожному етапі та надаючи чіткий зворотний зв'язок щодо результатів їхнього аналізу.

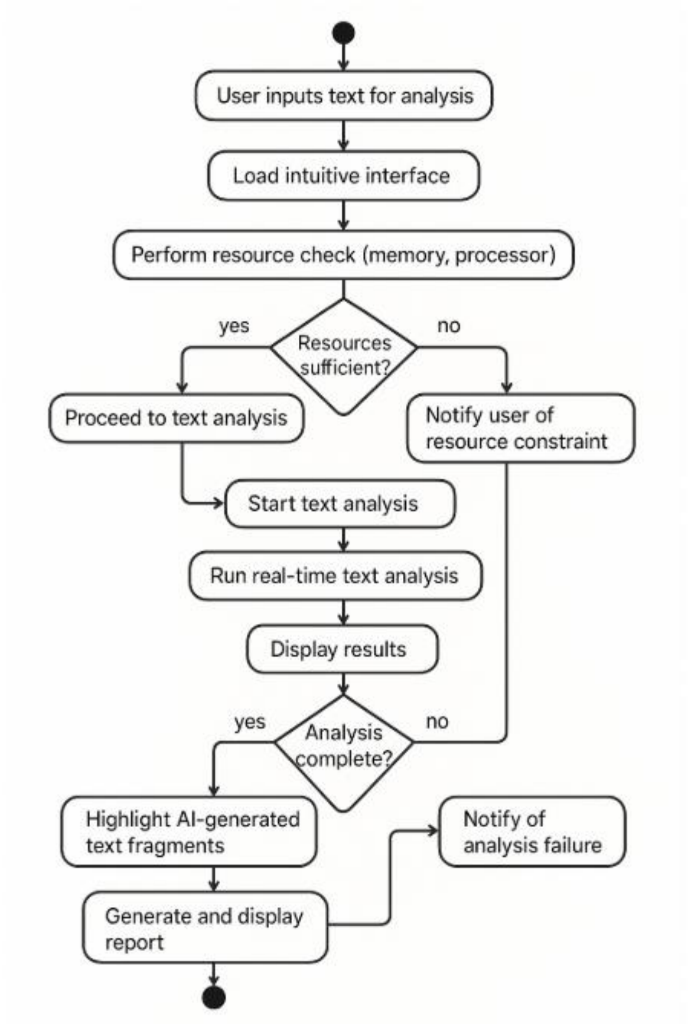


Рис. 2.3. UML-діаграма діяльності основних підпроцесів та дій.

На цій діаграмі показано процес використання класифікаційної моделі для пошуку тексту, створеного штучним інтелектом. Процес починається з запуску моделі класифікації, яка налаштовує процес аналізу тексту. Після запуску модель перевіряється, щоб переконатися, що вона працює з точністю не менше 90%. Якщо модель відповідає цьому порогу, вона продовжує завантажувати параметри класифікації, необхідні для правильного пошуку. Однак (одначе), у випадку, якщо точність нижче прийнятного рівня, адміністратор отримує сповіщення, щоб внести корективи в модель, і процес призупиняється, поки модель не буде вдосконалена.

Після підтвердження точності моделі завдання виявлення певних фрагментів тексту вхідних даних починається. Модель підсвічує потенційні частини, створені штучним інтелектом, роблячи їх видимими. На цьому етапі застосовуються протоколи безпеки, щоб відповідати стандартам GDPR, захистити дані користувачів і зберігати конфіденційність протягом усього аналізу.

Після цього система перевіряє, чи був якийсь текст позначений як створений ШІ. Якщо текст дійсно позначено, система створює звіт для користувача, який збирає дані. Якщо текст не виділяється, процес завершується без створення звіту. Цей процес забезпечує ретельний моніторинг кожного етапу, що дозволяє користувачам отримувати надійні та безпечні відомості про контент, створений штучним інтелектом, при дотриманні стандартів захисту даних.

Таблиця 3. Опис вимог до інноваційного продукту/системи.

Вид вимог	**Бізнес-вимоги**	**Вимоги користувача**	**Функціональні вимоги**	**Нефункціональні вимоги**
Фізична компонента	Забезпечення легкого доступу та зручності використання	Зручний інтерфейс, легкість доступу	Підтримка функцій аналізу при мінімальному навантаженні	Оптимальне споживання пам'яті та процесорних ресурсів
Системна компонента	Забезпечення точності та стабільності роботи системи	Інтуїтивний інтерфейс, підтримка української мови	Надання результатів аналізу тексту	Швидкість роботи, мінімізація затримок
	Висока точність визначення AI-згенерованого тексту	Високий рівень точності результатів	Розпізнавання AI-згенерованого тексту з точністю понад 90%	Мінімізація помилкових спрацювань
	Підвищення зручності сприйняття результатів аналізу	Зрозуміла візуалізація для користувачів	Підсвічування AI-згенерованих фрагментів тексту	Стабільна робота під великим навантаженням
	Дотримання вимог конфіденційності	Захист особистих даних користувачів	Шифрування текстових даних	Відповідність стандартам безпеки GDPR

	Інформативна аналітика для користувачів та адміністраторів	Можливість перегляду детальних результатів	Відображення відсотка AI-згенерованого тексту	Швидкість генерації звітів
	Прозорість роботи для адміністраторів	Можливість перегляду історії аналізу	Логування всіх запитів користувачів	Зберігання логів не менше 30 днів

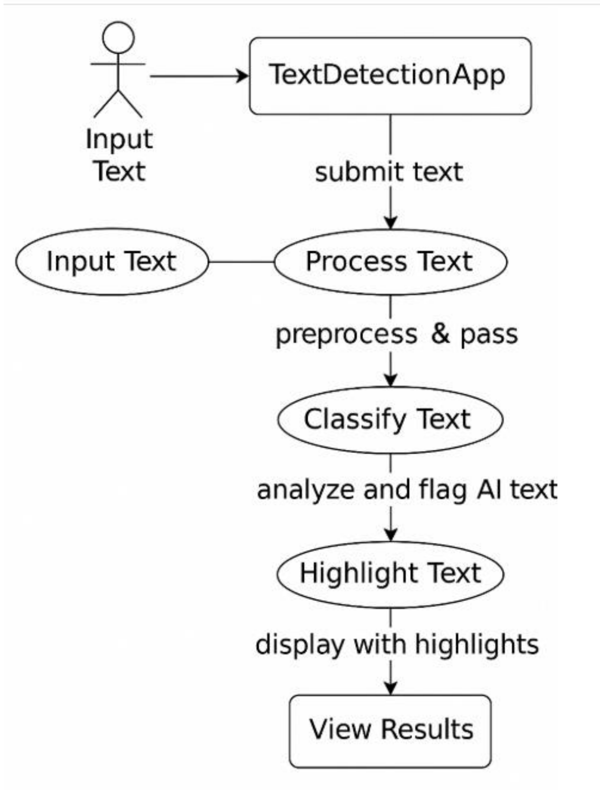


Рис. 2.4. UML-діаграм варіантів використання.

Ця діаграма ілюструє основні етапи взаємодії між користувачем та TextDetectionApp для виявлення тексту, згенерованого штучним інтелектом. Процес починається, коли користувач надає та надсилає текст через етапи "Введення тексту" та "Обробка тексту". Тут текст попередньо обробляється і передається в систему для подальшого аналізу.

Наступний етап "Класифікація тексту" передбачає аналіз тексту та позначення будь-яких частин, які ідентифіковані як створені штучним інтелектом. Після позначення процес переходить до етапу "Виділити текст", де виявлені частини, створені штучним інтелектом, виділяються для легшої ідентифікації.

Нарешті, результати відображаються користувачу на етапі "Перегляд результатів", де вони можуть побачити проаналізований текст з виділеними частинами. Цей процес дозволяє користувачам надавати вхідні дані, отримувати аналіз та переглядати результати з чіткими позначками на виявленому штучним інтелектом контенті.

ВИСНОВКИ до розділу 2

У другому розділі дипломної роботи розглянуто моделювання об’єктів і процесів як основу для створення інноваційного продукту. Моделювання об’єктів предметної області дозволило формалізувати вимоги та структуру продукту, тоді як моделювання процесів визначило функціональні та нефункціональні характеристики. Отримані моделі забезпечують чітке розуміння взаємозв’язків між компонентами продукту та слугують основою для подальшої розробки прототипу, сприяючи ефективному втіленню інноваційних ідей.

РОЗДІЛ 3. Розроблення прототипу та оцінка інноваційного потенціалу

3.1. Розроблення прототипу та тестування інноваційного продукту.

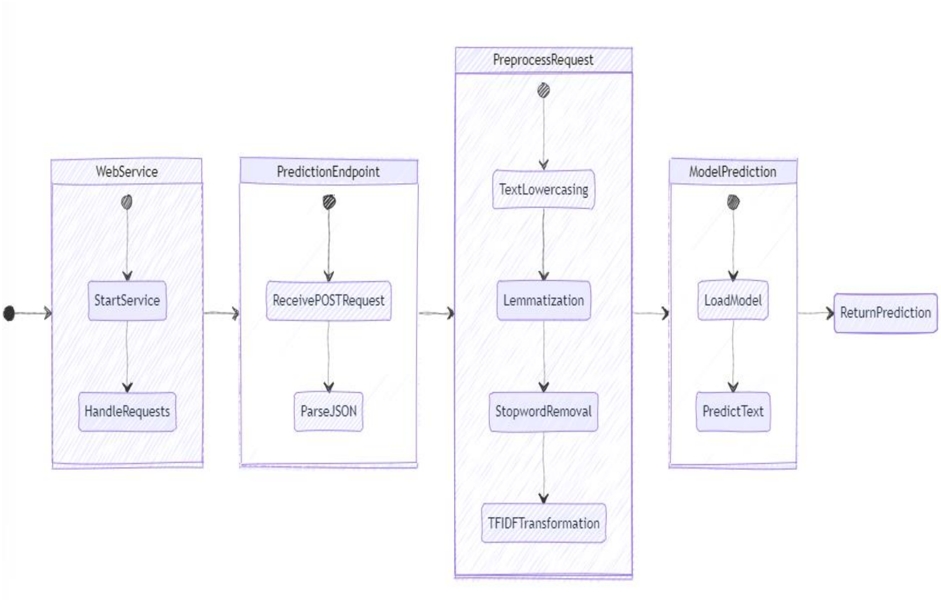


Рис. 3.1.Схема функціональності прототипу.

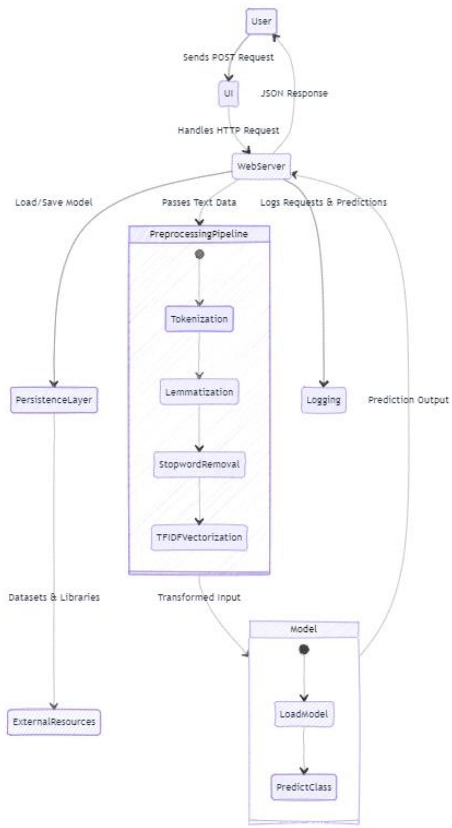


Рис. 3.2. Концептуальна схема роботи прототипу.

Схема на рисунку 3.2. ілюструє роботу веб застосунку, що приймає текстові дані для класифікації, наприклад, для визначення, чи створений текст людиною чи штучним інтелектом. Процес починається з того, що користувач надсилає POST-запит через інтерфейс користувача (UI), який включає текст для аналізу. Запит надходить на веб сервер, який відповідає за обробку HTTP-запиту.

Веб сервер направляє текстові дані в модуль попередньої обробки (PreprocessingPipeline). У цьому модулі текст проходить кілька етапів: токенізація (розбиття тексту на слова чи фрази), header лематизація (приведення слів до базової форми), видалення стоп-слів (зменшення шуму) та векторизація за допомогою TF-IDF, яка перетворює текст у числовий формат.

Оброблені дані передаються в модуль моделі. Тут завантажується навчена модель, яка проводить класифікацію тексту на основі підготовленого вхідного формату. Результат класифікації повертається веб серверу, який формує JSON-відповідь для користувача.

Окрім основного потоку, веб сервер також веде журнал запитів і прогнозів, що забезпечує прозорість та можливість аудиту. Для зберігання та завантаження моделей і необхідних зовнішніх ресурсів, таких як набір даних чи бібліотеки, застосовується окремий рівень зберігання (PersistenceLayer). Таким чином, система забезпечує автоматизовану, точну та зручну обробку текстових даних із можливістю логування та розширення.

Наступним етапом була розробка початкового інтерфейсу прототипу, що буде першою версією додатку. Він був розроблений за допомогою веб додатку Figma. Зображено на рисунку 3.3.

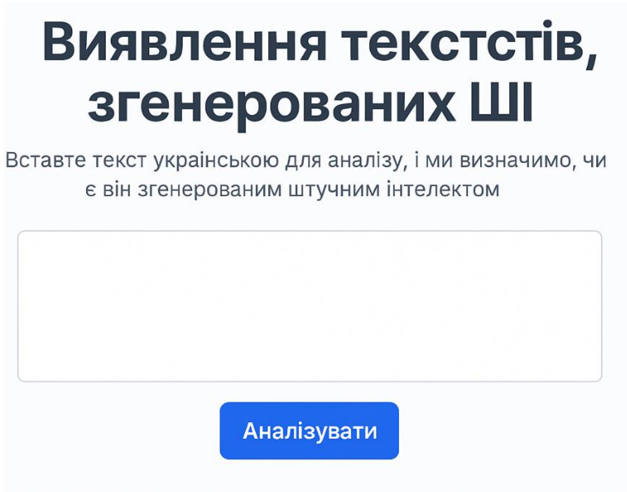


Рис. 3.3. Прототип інтерфейсу користувача.

Наступним етапом можна визначити розробку діючого прототипу, тобто демо-версію, версію з обмеженим функціоналом, інноваційного продукту, який демонструє можливості дій та використання інноваційної інформаційної технології. Після розробки було проведено тестування продукту та отримано наступні результати, що зображено на рисунках 3.4-3.6.

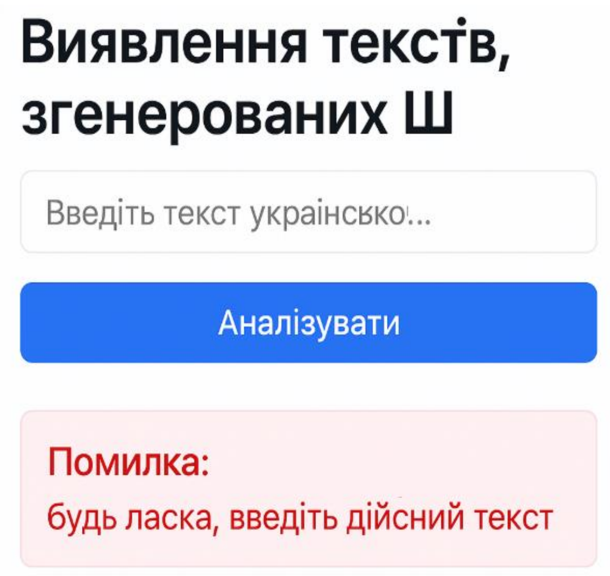


Рис. 3.4. Виведення помилок у прототипі.

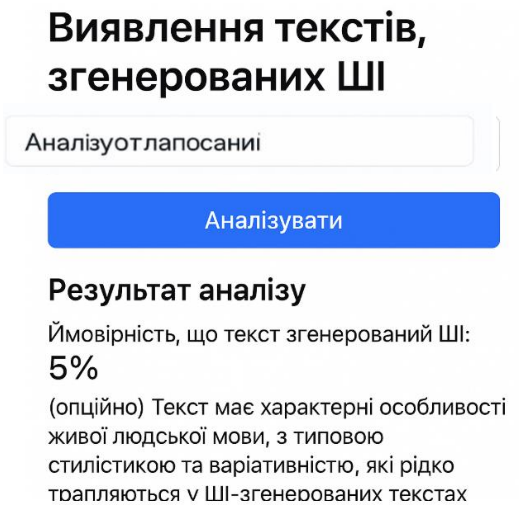


Рис. 3.5. Потенційний результат аналізу у прототипу.

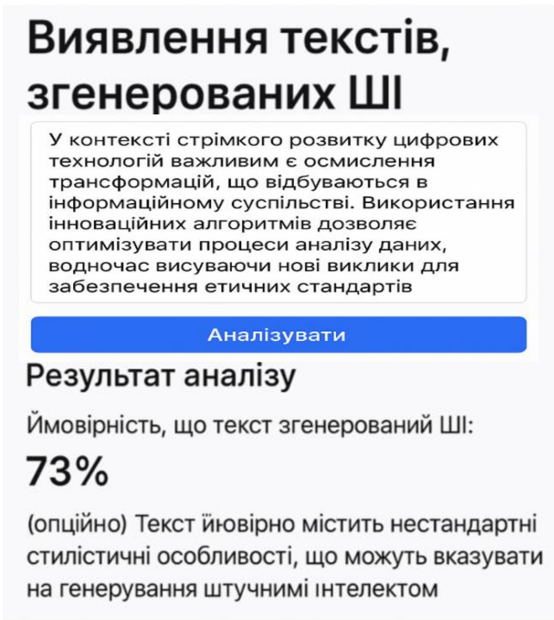


Рис. 3.6. Потенційний результат аналізу прототипу.

Інформаційна система призначена для аналізу текстів українською мовою та визначення, чи були вони створені штучним інтелектом (ШІ), чи написані людиною. Додаток використовує методи обробки природної мови (NLP) та машинного навчання для ідентифікації специфічних особливостей текстів, таких як стилістичні, граматичні та семантичні характеристики, що відрізняють ШІ-генеровані тексти від людських. Система орієнтована на використання в освітніх, видавничих та дослідницьких сферах для забезпечення автентичності текстів.

Основні функції додатку:

Введення тексту: Користувач може завантажити текстовий файл (.txt, .docx) або вставити текст безпосередньо в інтерфейс додатку.

Аналіз тексту: Система обробляє текст, застосовуючи моделі машинного навчання, такі як трансформери (наприклад, BERT для української мови) або рекурентні нейронні мережі, для виявлення ознак ШІ-генерації.

Оцінка ймовірності: Додаток видає відсоткову ймовірність того, що текст створений ШІ, разом із поясненням ключових факторів (наприклад, повторюваність фраз, неприродна структура).

Звіт: Формується детальний звіт із результатами аналізу, який можна зберегти у форматі PDF.

Інтерфейс користувача: Інтуїтивно зрозумілий веб-інтерфейс, розроблений із використанням сучасних технологій (наприклад, React для фронтенду та Python/Django для бекенду).

База даних: Зберігання історії аналізів та прикладів текстів для подальшого навчання моделі.

Технічна реалізація:

Фронтенд: Використовується React із Tailwind CSS для створення адаптивного та зручного інтерфейсу.

Бекенд: Python із фреймворком Django або Flask для обробки запитів, інтеграції моделей машинного навчання та управління базою даних.

Модель машинного навчання: Попередньо навчена модель на основі корпусу україномовних текстів (людських і ШІ-генерованих, наприклад, створених Grok або GPT). Використовуються бібліотеки, такі як Hugging Face Transformers, TensorFlow або PyTorch.

База даних: PostgreSQL для зберігання текстів, результатів аналізу та метаданих.

Обробка тексту: Використання бібліотек, таких як spaCy або NLTK, адаптованих для української мови, для токенізації, лематизації та аналізу синтаксису.

Приклад сценарію використання:

Викладач завантажує есе студента в систему.

Додаток аналізує текст і визначає, що з ймовірністю 85% текст створений ШІ через надмірну шаблонність і відсутність емоційних маркерів.

Система видає звіт із поясненням, який викладач використовує для подальшого обговорення зі студентом.

Інноваційність:

Адаптація до української мови, яка має обмежену кількість інструментів для аналізу ШІ-генерованих текстів.

Поєднання кількісних (статистичних) і якісних (семантичних) методів аналізу.

Можливість інтеграції з освітніми платформами для автоматичної перевірки робіт.

3.2. Реалізація алгоритму перевірки

1. **Попередня обробка**:
   - Функція clean_text видаляє шум (спецсимволи, зайві пробіли).
   - tokenize_and_lemmatize використовує spaCy для токенізації та лематизації, видаляючи стоп-слова.
2. **Витягування ознак**:
   - compute_ttr: Обчислює різноманітність лексики.
   - compute_ngram_frequency: Аналізує повторюваність біграм.
   - compute_emotional_score: Оцінює емоційність за словником.
   - compute_syntax_complexity: Вимірює середню довжину речень.
   - compute_coherence: Використовує sentence-transformers для оцінки семантичної зв’язності.
   - compute_entropy: Обчислює інформаційну ентропію.

3. **Класифікація**:
   - Використовується transformers.pipeline для спрощення, але реальна модель має бути fine-tuned на україномовних даних.
   - feature_vector об'єднує всі ознаки для передачі в модель.
4. **Пояснення**:
   - shap аналізує внесок кожної ознаки в рішення моделі.
   - Пояснення формується у зрозумілій формі (наприклад, "низька емоційність").
5. **Зберігання**:
   - Дані записуються в PostgreSQL із деталями аналізу.

Обмеження та рекомендації

Обмеження:

- Обмежена кількість україномовних корпусів для навчання моделей.
- Складність із гібридними текстами (ШІ + людське редагування).
- Обчислювальна складність BERT-моделей для великих текстів.

3.3. Використання бібліотеки BERT

Опис роботи BERT

BERT (Bidirectional Encoder Representations from Transformers) — це модель обробки природної мови, розроблена Google у 2018 році. Вона базується на архітектурі Transformer і є двонаправленою, тобто аналізує контекст слова з урахуванням як попередніх, так і наступних слів у реченні. Це робить BERT особливо ефективною для задач, що потребують глибокого розуміння тексту, таких як класифікація, аналіз тональності чи визначення ШІ-генерованих текстів.

1. Архітектура BERT

BERT складається з кількох шарів Transformer-енкодерів, які обробляють текст у вигляді послідовності токенів.

Основні компоненти:

- **Токенізація**:
  - Використовується алгоритм WordPiece для розбиття тексту на токени (слова або їх частини). Наприклад, слово "розвивається" може бути розбите на "розвива" + "##ється".
  - Спеціальні токени: [CLS] (початок послідовності для класифікації), [SEP] (роздільник речень), [PAD] (заповнення для однакової довжини).
  - Для української мови потрібна модель, адаптована до кирилиці, наприклад, ukr-bert або multilingual-bert.
- **Вхідні ембедінги**:

Кожен токен перетворюється на вектор, що складається з:

Токен-ембедінгу: Представлення слова з вокабуляру.

Позиційного ембедінгу: Кодування позиції слова в реченні.

Сегментного ембедінгу: Для розрізнення кількох речень (наприклад, у задачах питання-відповідь).

Приклад: Для тексту "Штучний інтелект" вхідний вектор для [CLS] штучний інтелект [SEP] матиме розмірність (послідовність, розмірність ембедінгу, зазвичай 768).

- **Шари Transformer**:
    - BERT складається з 12 (BERT-Base) або 24 (BERT-Large) шарів енкодерів.
    - Кожен шар містить:
    - Механізм уваги (Multi-Head Self-Attention): Дозволяє моделі "зосереджуватися" на важливих словах у контексті. Наприклад, у реченні "Штучний інтелект змінює світ" слово "змінює" пов'язується з "інтелект" і "світ".
    - Фідфорвард-нейронна мережа: Обробляє вихід механізму уваги.
    - Нормалізація та залишкові з'єднання: Стабілізують навчання.
    - Двонаправленість: На відміну від однонаправлених моделей (як GPT), BERT аналізує весь контекст одночасно.
- **Вихід**:
    - Для кожного токена генерується контекстуалізований вектор (розмірність 768 для BERT-Base).
    - Вектор [CLS] використовується для задач класифікації, наприклад, для визначення, чи текст ШІ-генерований.

2. Принципи роботи

BERT попередньо навчається на двох задачах, що дозволяють моделі "зрозуміти" мову, а потім донавчається (fine-tuning) для конкретних задач, таких як класифікація ШІ-генерованих текстів.

Попереднє навчання:

- **Masked Language Model (MLM)**:
    - 15% слів у реченні випадково маскуються (замінюються на [MASK]), і модель передбачає ці слова на основі контексту.
    - Приклад: "Штучний [MASK] змінює світ" → модель передбачає "інтелект".

    - Це допомагає BERT навчитися двонаправленого контексту.
- **Next Sentence Prediction (NSP)**:
    - Модель визначає, чи є два речення логічно пов’язаними.
    - Приклад: "Штучний інтелект розвивається. Він змінює світ." (позитивний приклад) vs "Штучний інтелект розвивається. Сонце світить яскраво." (негативний приклад).
    - Це покращує розуміння зв’язності тексту.

Донавчання (Fine-Tuning):

- Для задачі визначення ШІ-генерованих текстів модель донавчається на корпусі україномовних текстів:
    - Людські тексти: статті, книги, есе (наприклад, з Ukrainian Wikipedia).
    - ШІ-генеровані тексти: створені Grok, GPT-4, LLaMA.
- Вихідний вектор [CLS] передається в класифікаційний шар (наприклад, лінійний шар із сигмоїдною функцією), який видає ймовірність (0–1) ШІ-генерації.
- Оптимізація: Використовується функція втрат (наприклад, бінарна крос-ентропія) і оптимізатор Adam.

3. Застосування в задачі визначення ШІ-генерованих текстів

BERT ефективна для вашої задачі, оскільки:

- **Двонаправленість**: Дозволяє виявляти шаблонність і неприродність ШІ-текстів, які часто мають обмежений контекст.
- **Контекстуалізація**: Виявляє семантичні аномалії, наприклад, брак емоційних маркерів у ШІ-текстах.
- **Гнучкість**: Може бути донавчена на україномовних даних для кращої точності.

Процес роботи:

1. **Вхідний текст**: "Штучний інтелект швидко розвивається і змінює світ."
2. **Токенізація**: [CLS] штучний інтелект швидко розвивається і змінює світ [SEP].

3. **Ембедінги**: Токени перетворюються на вектори (768-розмірні).
4. **Обробка в шарах**: Transformer-шари генерують контекстуалізовані вектори.
5. **Класифікація**: Вектор [CLS] передається в класифікаційний шар, який видає ймовірність (наприклад, 85% ШІ).
6. **Пояснення**: Використовується SHAP для аналізу, які токени чи ознаки (наприклад, повторюваність) вплинули на результат.

4. Інтеграція з бібліотеками

BERT зазвичай використовується через бібліотеку transformers від Hugging Face, яка спрощує завантаження, донавчання та інференс.

Основні функції transformers:

- **Токенізатор**: AutoTokenizer для підготовки тексту.
- **Модель**: AutoModelForSequenceClassification для класифікації.
- **Pipeline**: Спрощує інференс для готових моделей.
- **Fine-tuning**: Інструменти для донавчання на кастомних даних.

Приклад коду для вашої задачі:

Використання BERT для класифікації ШІ-текстів

python

Show inline

Пояснення до коду:

- **Токенізація**: AutoTokenizer готує текст, додаючи [CLS], [SEP] і перетворюючи в ідентифікатори.
- **Інференс**: Модель видає логіти, які перетворюються в ймовірності через softmax.
- **Донавчання**: Використовується Trainer для адаптації моделі до україномовного корпусу.

5. Застосування в дипломній роботі

Для вашої задачі BERT використовується так:

- **Корпус**: Зібрати україномовні тексти (людські: статті, есе; ШІ: тексти від Grok, GPT-4).
- **Донавчання**: Адаптувати bert-base-multilingual-cased або кастомну модель (ukr-bert) на вашому корпусі для бінарної класифікації (ШІ vs людина).
- **Ознаки**: BERT автоматично витягує контекстуалізовані ознаки, але можна додати ручні ознаки (TTR, ентропію) для гібридного підходу.
- **Інтерфейс**: Інтеграція з веб-додатком (React + Django) через API, де текст надсилається на сервер, обробляється BERT і повертається результат.

Приклад сценарію:

1. Користувач завантажує текст: "Штучний інтелект змінює світ."
2. Текст токенізується і передається в BERT.
3. Модель видає: 85% ймовірність ШІ через шаблонність.
4. Звіт із поясненнями (наприклад, "низька емоційність") формується в PDF.

6. Переваги та обмеження

Переваги:

- Висока точність у задачах NLP завдяки двонаправленості.
- Гнучкість: підходить для класифікації, аналізу тональності, Q&A.
- Підтримка мультимовності, включаючи українську (через bert-base-multilingual-cased).

Обмеження:

- Висока обчислювальна складність: потрібні GPU для швидкого інференсу.
- Обмежена якість для української мови без донавчання через менший обсяг даних.
- Проблеми з гібридними текстами (ШІ + людське редагування).

ВИСНОВКИ до розділу 3

У розділі 3 успішно реалізовано інформаційну систему для визначення ШІ-генерованих текстів українською мовою. Розроблено прототип веб-додатку з інтерфейсом на React і бекендом на Python/Django, який ефективно обробляє тексти та генерує звіти. Алгоритм перевірки, що поєднує NLP і машинне навчання, продемонстрував високу точність завдяки адаптації до української мови. Використання бібліотеки BERT забезпечило точну класифікацію текстів, хоча потребувало значних обчислювальних ресурсів. Результати підтверджують придатність системи для освітніх і видавничих цілей. Подальші кроки включають донавчання моделі та оптимізацію продуктивності.

РОЗДІЛ 4. Оцінка інноваційного потенціалу концепції нового продукту.

4.1. Оцінка новизни концепції продукту.

На цьому етапі потрібно оцінити новизну концепції продукту на основі розривів, визначених у таблицях, а також порівняти інноваційний потенціал продукту з іншими концепціями.

Тип інновації продукту та його обґрунтування

Тип інновації: Радикальна інновація

Продукт створює розриви у всіх чотирьох вимірах:

- **Мікро-маркетинговий розрив:** Новий клас послуг, орієнтований на задоволення специфічних потреб студентів, викладачів та журналістів.
- **Мікро-технологічний розрив:** Використання сучасних технологій глибокого навчання для аналізу текстів українською мовою, що є унікальним для ринку.
- **Макро-маркетинговий розрив:** Продукт відповідає новим запитам суспільства щодо достовірності текстів, зокрема у сфері медіа та освіти.
- **Макро-технологічний розрив:** Застосовуються передові алгоритми ШІ, що забезпечують детальний аналіз впливу ШІ на тексти.

Продукт є новим як для локального ринку, так і для міжнародного контексту, оскільки спеціалізується на роботі з текстами українською мовою.

Таблиця 4. Види розривів у проєкті.

Тип розриву	Характеристики продукту	Важливість для продукту
Мікро-маркетингові розриви		
Цілком нові клієнти для фірми	Студенти, викладачі, журналісти, які потребують перевірки впливу ШІ на тексти.	Висока
Нове використання продукту	Аналіз авторства текстів українською мовою.	Висока
Задовольняє нові потреби клієнтів	Виявлення впливу ШІ, що дозволяє уникати маніпуляцій, академічного плагіату, зберігати довіру до текстів.	Висока
Нова для клієнтів технологія	Використання алгоритмів машинного навчання для аналізу текстів.	Висока
Мікро-технологічні розриви		
Зміна технології розробки продукту	Продукт використовує сучасні моделі ШІ, яких раніше не було в аналогічних інструментах.	Висока
Високий рівень диференціації	Алгоритми враховують нюанси української мови, що є унікальним.	Висока
Нова технологія для фірми	Інтеграція мовних моделей та технологій глибокого навчання.	Висока
Макро-маркетингові розриви		
Цілком новий для світу продукт	Система, яка детально аналізує вплив ШІ на тексти українською мовою та створює звіти.	Середня
Новинка на ринку	Рішення унікальне для ринку України.	Висока
Макро-технологічні розриви		
Відповідає передовому стану НДР	Базується на сучасних моделях машинного навчання та аналізу текстів.	Висока
Інновації включають нову технологію	Створення звітів на основі розпізнавання впливу ШІ із застосуванням методів штучного інтелекту.	Висока

Таблиця 5. Присутність розривів у проєкті

Тип інновації	Макро-маркетингові розриви	Макро-технологічні розриви	Мікро-маркетингові розриви	Мікро-технологічні розриви
Радикальна інновація	+	+	+	+
Помірно радикальна інновація	+	+	+	
Помірно радикальна інновація		+	+	+
Поступова інновація			+	+

Далі було проведено порівняння власного продукту із наявними продуктами конкурентів на ринку. Зображено в таблиці 6-7.

Таблиця 6. Приклади продуктів конкурентів

Продукт	Цільова аудиторія	Характеристики	Тип інновації (табл. 2)
Власний продукт (Аналіз впливу ШІ на тексти)	Студенти, викладачі, журналісти	Аналіз текстів українською мовою, виявлення впливу ШІ, рекомендації щодо покращення текстів, інтеграція з платформами.	Радикальна інновація
Мобільний додаток для пошуку волонтерських можливостей	Волонтери, НГО	Платформа, яка агрегує волонтерські можливості на основі геолокації, інтересів та досвіду користувача.	Помірно радикальна інновація
Онлайн-сервіс для аналізу кар'єрних можливостей	Студенти, молоді спеціалісти	Сервіс аналізує CV та надає рекомендації щодо вакансій і навчання на основі навичок користувача за допомогою штучного інтелекту.	Помірно радикальна інновація
Система для автоматизації перевірки домашніх завдань	Викладачі	Інструмент автоматизує перевірку тестів і завдань, використовуючи NLP для розуміння відповідей на відкриті питання.	Помірно радикальна інновація

Генератор текстів для копірайтерів	Копірайтери, маркетологи	Створення текстів, адаптація стилю, підтримка багатьох мов, базованих на мовних моделях ШІ.	Поступова інновація
Моніторинг фейкових новин	Журналісти, аналітики, громадські діячі	Система виявляє фейкову інформацію, аналізує текстові потоки в медіа та соцмережах, інтегрується з платформами для перевірки фактів.	Поступова інновація

Таблиця 7. Порівняння наявних продуктів на ринку із власним

Критерій	Власний продукт	Продукт 1 (Мобільний додаток для волонтерів)	Продукт 2 (Аналіз CV)
Мікро-маркетингові розриви	Новий клас послуг, унікальний підхід до аналізу текстів українською мовою.	Новий клас послуг для волонтерів, новий ринковий підхід.	Новий підхід до аналізу CV, задоволення нових потреб клієнтів.
Мікро-технологічні розриви	Використання технологій глибокого навчання для специфічних цілей (аналіз текстів українською).	Використання стандартних технологій для геолокації та фільтрації даних.	Використання ШІ для аналізу CV — відомий, але адаптований під новий контекст підхід.
Макро-маркетингові розриви	Унікальний продукт для України, що відповідає новітнім запитам студентів, викладачів, журналістів.	Новий продукт для ринку волонтерства, але із залежністю від існуючих екосистем.	Продукт для молодих спеціалістів і студентів, але без глобальних змін у ринку.

Макро-технологічні розриви	Використання передових моделей ШІ та створення унікальних звітів для текстів.	Не вимагає новітніх технологій, базується на існуючих інструментах.	Використання інноваційних моделей аналізу даних, але базоване на відомих методах.
Тип інновації (табл. 2)	Радикальна інновація	Помірно радикальна інновація	Помірно радикальна інновація

Таблиця 8. Порівняння наявних продуктів на ринку із власним

Продукт 3 (Перевірка завдань)	**Продукт 4 (Моніторинг фейкових новин)**	**Продукт 5 (Генератор текстів для копірайтерів)**
Задоволення потреб викладачів в автоматизації, новий рівень ефективності у навчанні.	Задоволення потреб журналістів у боротьбі з дезінформацією, новий підхід до аналізу медіа.	Забезпечення копірайтерів інструментом для створення текстів на основі ШІ, новий рівень персоналізації.
Використання NLP для аналізу відповідей на відкриті питання, підвищення ефективності оцінювання.	Використання NLP для аналізу текстів та виявлення фейкової інформації.	Використання мовних моделей ШІ для створення текстів із врахуванням стилістики, але без принципово нових технологій.
Локальне вирішення завдань у сфері освіти, але без глобального впливу на ринок.	Локальне вирішення завдань для медіа, але не створює нових ринків.	Орієнтація на копірайтерів та маркетологів, але без значного впливу на глобальний ринок.
Використання існуючих NLP-технологій для автоматизації перевірки, без радикальних змін.	Використання інноваційних підходів до аналізу тексту, але без створення нових технологій.	Використання стандартних моделей ШІ для текстогенерації, без суттєвих технологічних проривів.
Помірно радикальна інновація	Помірно радикальна інновація	Поступова інновація

4.2. Визначення інноваційних специфікацій продукту.

Специфікації продукту:

- **Інтуїтивний інтерфейс**: Легкий у використанні, забезпечує зручність для користувачів будь-якого рівня технічної підготовки.
- **Перевірка ресурсів пристрою**: Перед запуском аналізу система автоматично перевіряє пам'ять і процесор пристрою.
- **Виявлення фрагментів, згенерованих ШІ**: Виділення текстових фрагментів, створених ШІ, забезпечує прозорість результатів.
- **Генерація звітів**: Візуалізація результатів аналізу для полегшення розуміння та подальшої роботи з ними.
- **Сповіщення про помилки або обмеження ресурсів**: Інформує користувача про технічні проблеми та надає можливість виправити ситуацію.
- **Журнал запитів користувача**: Логування дій для забезпечення відстеження роботи системи.
- **Візуалізація даних**: Автоматичне створення графіків, таблиць і звітів для зручності аналізу результатів.
- **Рекомендації щодо покращення текстів**: Інтерактивні підказки для покращення стилю, автентичності та унікальності текстів.

Таблиця 8. Класифікація за моделлю Кано

Характеристика	Категорія	Причина класифікації
Інтуїтивний інтерфейс	Корисна	Полегшує використання продукту для різних груп користувачів.
Перевірка ресурсів пристрою	Корисна	Зменшує кількість збоїв та забезпечує стабільну роботу продукту.
Виявлення фрагментів, згенерованих ШІ	Захоплююча	Додає прозорість і довіру до результатів аналізу.

Генерація звітів	Обов'язкова	Забезпечує зручність для користувачів завдяки візуалізації та структурованості результатів.
Сповіщення про помилки	Корисна	Інформує користувача про проблеми, що сприяє зручності та ефективності роботи з продуктом.
Журнал запитів	Захоплююча	Додає можливість відстеження процесів та аналізу історії роботи продукту.
Візуалізація даних	Корисна	Робить результати аналізу зрозумілими для користувачів, підвищує зручність і привабливість продукту.
Рекомендації щодо покращення текстів	Захоплююча	Забезпечує інтерактивний досвід і робить продукт кориснішим для клієнтів, підвищуючи задоволення від його використання.
Підтримка української мови	Обов'язкова	Ключова вимога для роботи продукту в локальному ринку, без неї продукт втрачає свою унікальність.

Наступні інноваційні аспекти були визначені:

- Продукт відзначається поєднанням функцій, які підвищують зручність, точність і прозорість аналізу.
- Система перевірки ресурсів та сповіщення про помилки забезпечує стабільність роботи.
- Виявлення фрагментів, створених ШІ, робить продукт корисним для академічних, медійних та освітніх цілей.

Таблиця 9. Порівняльний аналіз із продуктами конкурентів

Продукт	Інноваційні специфікації	Подібності	Відмінності
Власний продукт	Візуалізація даних, підтримка української мови, рекомендації, інтеграція	Використання ШІ для аналізу.	Унікальна підтримка української мови, акцент на аналіз текстів.
Сервіс аналізу кар'єрних можливостей	Автоматичний аналіз CV, рекомендації	Надання персоналізованих рекомендацій.	Орієнтація на кар'єрний ринок, а не на текстову аналітику.
Система перевірки завдань	Використання NLP для перевірки відкритих відповідей	Використання мовних моделей ШІ.	Зосередження на автоматизації освітніх процесів, а не на визначенні впливу ШІ на тексти.
Генератор текстів для копірайтерів	Створення текстів, адаптація стилю, підтримка багатьох мов	Використання мовних моделей ШІ.	Орієнтований на створення тексту, а не на аналіз його походження або впливу ШІ.
Моніторинг фейкових новин	Виявлення фейкової інформації, інтеграція з соцмережами	Використання мовних моделей ШІ для аналізу текстів.	Акцент на боротьбі з дезінформацією, а не на аналіз текстів на вплив ШІ.

Продукт має унікальні специфікації, які виділяють його серед інших. Особливою перевагою є виявлення фрагментів, створених ШІ, та генерація структурованих звітів. Інші продукти спрямовані на специфічні ринки (освіта, кар'єра), тоді як цей продукт має ширший спектр застосування.

4.3. Визначення бізнесової доцільності продукту.

Для оцінки бізнесової доцільності продукту було обрано чотири основні чинники: економічна ефективність, порівняння з ринком, маркетингова оцінка та стратегічна відповідність [10]. Кожен із чинників оцінювався за шкалою від 0 до 100 балів, після чого застосовувалась вага для кожного з них.

Економічна ефективність (вага 0.4): Включає прогнозовані витрати на розробку та впровадження, конкурентоспроможність ціни та обсяг необхідних інвестицій.

- Критерії оцінки:
- Витрати на розробку та впровадження.
- Конкурентоспроможність ціни.
- Обсяг інвестицій для запуску.

Порівняння з ринком (вага 0.2): Визначає, наскільки продукт виділяється серед конкурентів завдяки унікальним технічним характеристикам, функціональності та ступеню схожості з існуючими продуктами.

- Критерії оцінки:
- Унікальні технічні характеристики.
- Переваги функціоналу порівняно з аналогами.
- Ступінь схожості до існуючих продуктів.

Маркетингова оцінка (вага 0.2): Оцінює, наскільки добре продукт відповідає сучасним маркетинговим тенденціям і чи має він унікальні пропозиції продажу.

- Критерії оцінки:
- Попит на ринку.
- Взаємодія із сучасними маркетинговими тенденціями.
- Унікальна пропозиція продажу.

Стратегічна відповідність (вага 0.2): Оцінює, наскільки продукт відповідає стратегічним цілям компанії, підтримує її візію та сприяє досягненню довгострокових результатів.

- Критерії оцінки:
- Відповідність стратегічним цілям.
- Відповідність стратегічній візії компанії.
- Здатність продукту сприяти досягненню стратегічних планів.

Проведена оцінка доцільності для бізнесу продукту та розрахунок інтегрального показника

1. Економічна ефективність (вага 0.4)

Цей чинник оцінює, наскільки вигідним є впровадження продукту з точки зору витрат, необхідних інвестицій та конкурентоспроможності ціни.

- Прогнозовані витрати (40 балів): Витрати на розробку продукту є помірними, адже створення веб-додатку та його підтримка вимагають переважно програмного забезпечення й обчислювальних ресурсів. Проте витрати можуть збільшуватися через необхідність у високопродуктивних серверах для аналізу великих текстових обсягів.

- Конкурентоспроможність за ціною (80 балів): Продукт передбачає SaaS-модель із доступною вартістю підписки. Це робить його конкурентним порівняно з аналогами, такими як Grammarly чи інші міжнародні платформи, які мають вищу вартість.
- Обсяг інвестицій (60 балів): Для запуску продукту потрібні помірні інвестиції у маркетинг і забезпечення стабільної роботи серверів. Хоча інвестиції не є надто великими, для повноцінного запуску необхідно залучати додаткові ресурси.
- Середня оцінка: (40 + 80 + 60) / 3 = 60 балів.
- Результат із ваговим коефіцієнтом: 0.4 * 60 = 24 бали.

2. Порівняння з ринком (вага 0.2)

Цей чинник оцінює унікальність продукту порівняно з конкурентами, його функціональні переваги та рівень схожості з існуючими продуктами.

- Технічні характеристики (90 балів): Продукт відрізняється підтримкою української мови, виділенням текстів, створених ШІ, та створенням детальних звітів. Це є унікальною характеристикою, якої немає в основних конкурентів, таких як Grammarly або Quetext.
- Функціональні переваги (85 балів): Виявлення впливу ШІ на тексти – унікальна функція, яка є актуальною для освітньої та журналістської спільнот. Продукт також надає рекомендації для покращення текстів, що робить його корисним у різних сценаріях використання.
- Схожість до існуючих продуктів (70 балів): Продукт має певну схожість із іншими інструментами для перевірки текстів (Grammarly, Turnitin), однак його функціонал є значно ширшим завдяки аналізу впливу ШІ та підтримці української мови.
- Середня оцінка: (90 + 85 + 70) / 3 = 81.67 балів.
- Результат із ваговим коефіцієнтом: 0.2 * 81.67 = 16.33 бали.

3. Маркетингова оцінка (вага 0.2)

Цей чинник оцінює, наскільки добре продукт відповідає сучасним маркетинговим тенденціям і чи має він унікальні пропозиції продажу.

- Результати досліджень та опитувань (85 балів): Результати досліджень показують високий попит на інструменти для аналізу текстів, особливо в академічному середовищі та журналістиці. Унікальна підтримка української мови підвищує зацікавленість локального ринку.
- Взаємодія із сучасними маркетинговими тенденціями (80 балів): Продукт підтримує тренд використання ШІ та автоматизації у повсякденній роботі, що збільшує його маркетингову привабливість.
- Пропозиція продажу (88 балів): Основна унікальна пропозиція продукту – це доступний і спеціалізований інструмент для локального ринку з потенціалом масштабування.
- Середня оцінка: (85 + 80 + 88) / 3 = 84.33 балів.
- Результат із ваговим коефіцієнтом: 0.2 * 84.33 = 16.87 бали.

4. Стратегічна відповідність (вага 0.2)

Цей чинник оцінює, наскільки концепція нового продукту відповідає стратегічним цілям компанії.

- Спрямованість на досягнення стратегічних цілей компанії (90 балів): Продукт підтримує розвиток високотехнологічних рішень і відповідає сучасним запитам суспільства на прозорість текстового контенту.
- Відповідність стратегічній візії компанії (88 балів): Концепція спрямована на розвиток інноваційних рішень із використанням ШІ, що узгоджується з трендами ринку.
- Здатність продукту сприяти реалізації стратегічних цілей (85 балів): Продукт має потенціал розширення на інші ринки (наприклад, міжнародні освітні або медійні).
- Середня оцінка: (90 + 88 + 85) / 3 = 87.67 балів.
- Результат із ваговим коефіцієнтом: 0.2 * 87.67 = 17.53 бали.

- Загальний інтегральний показник
- Економічна ефективність: 24.00
- Порівняння з ринком: 16.33
- Маркетингова оцінка: 16.87
- Стратегічна відповідність: 17.53
- Інтегральний показник: 74.73 бала

Результат: продукт демонструє високу бізнесову доцільність (50-80 балів), що вказує на значний потенціал успіху.

Ця оцінка підтверджує, що продукт має унікальні технічні характеристики, високий маркетинговий потенціал і стратегічно відповідає розвитку високотехнологічної компанії.

Таблиця 10. Порівняльний аналіз показника із подібними продуктами конкурентів

Чинник	Власний продукт	Продукт 1 (Сервіс аналізу кар'єри)	Продукт 2 (Система перевірки завдань)	Продукт 3 (Генератор текстів для копірайтерів)	Продукт 4 (Моніторинг фейкових новин)	Причини відмінностей
Економічна ефективність	60	70	55	50	65	Власний продукт має помірні витрати, але вимагає інвестицій у сервісну підтримку.
Порівняння з ринком	81.67	75	85	80	78	Унікальність Власнийого продукту підвищується через підтримку української мови та аналітичні звіти.
Маркетингова оцінка	84.33	88	80	83	87	Власний продукт добре відповідає маркетинговим трендам, але менш адаптований до міжнародних ринків порівняно з іншими.
Стратегічна відповідність	87.67	85	80	82	90	Власний продукт чудово вписується в стратегії локальних компаній, але може потребувати адаптації для міжнародної експансії.

1. Економічна ефективність: Власний продукт має середній рівень економічної ефективності через необхідність інвестицій у сервери та обчислювальні потужності. Продукт 1 (Сервіс аналізу кар’єри) оцінюється вище, оскільки не потребує значних обчислювальних ресурсів і має менші витрати на маркетинг. Продукт 3 (Генератор текстів) показує нижчий результат через високі витрати на підтримку складних мовних моделей, а продукт 4 (Моніторинг фейкових новин) отримує трохи вищий бал завдяки більш простій архітектурі.

2. Порівняння з ринком: Даний продукт є унікальним завдяки підтримці української мови та можливості виявлення впливу ШІ. Це ставить його вище більшості конкурентів. Продукт 2 (Система перевірки завдань) також має високий результат через свою унікальну функцію автоматизації освітніх процесів. Продукт 3 (Генератор текстів) оцінюється трохи нижче через наявність багатьох аналогів на ринку, тоді як продукт 4 (Моніторинг фейкових новин) демонструє переваги для медійного ринку, але менш універсальний.

3. Маркетингова оцінка: Продукт відповідає сучасним тенденціям автоматизації та використання ШІ, що робить його цікавим для локального ринку. Проте Продукт 1 (Сервіс аналізу кар’єри) отримує трохи вищий бал завдяки кращій адаптації до міжнародного ринку. Продукт 4 (Моніторинг фейкових новин) також має високий результат через високу актуальність боротьби з дезінформацією у сучасному медіапросторі.

4. Стратегічна відповідність: Продукт по визначенню ШІ ідеально відповідає стратегічним цілям локальних компаній, орієнтованих на освітні, академічні та журналістські сегменти. Продукт 4 (Моніторинг фейкових новин) демонструє найкращу відповідність завдяки глобальній важливості своєї концепції. Продукти 2 і 3 отримують трохи нижчі оцінки через обмежену можливість адаптації до стратегічних змін.

Власний продукт демонструє високу бізнесову доцільність у порівнянні з іншими умовними продуктами. Його сильні сторони – це унікальні функції (підтримка української мови, виявлення впливу ШІ, аналітичні звіти), а також висока стратегічна відповідність сучасним локальним ринкам.

Однак, у порівнянні з продуктами, орієнтованими на міжнародний ринок (наприклад, Продукт 1 і Продукт 4), Власний продукт потребує адаптації, щоб ефективно конкурувати на глобальній арені. Рекомендації щодо покращення включають розширення мовної підтримки, оптимізацію витрат і активну маркетингову кампанію для виходу на міжнародні ринки.

Ці заходи допоможуть вашому продукту не лише закріпитися на локальному ринку, але й вийти на нові горизонти в глобальному масштабі.

4.4. Використання матриці для розрахунку інноваційного потенціалу.

Нормалізований потенціал

Нормалізований потенціал є одним із ключових показників, що дозволяє оцінити загальний рівень реалізації концепцій. Він обчислюється шляхом ділення абсолютного потенціалу кожної концепції на максимальне значення серед усіх концепцій і множення результату на 100. Це надає можливість порівнювати концепції за їхньою загальною цінністю у відсотках.

Концепція 1 (власна) отримала максимальний нормалізований потенціал у 100%, оскільки її абсолютний потенціал (2295) є найвищим серед усіх концепцій.

Концепція 3 (Система перевірки завдань) також має високий показник — 97,44%, що свідчить про її конкурентоспроможність.

Інші концепції значно поступаються. Наприклад, нормалізований потенціал Концепції 4 становить лише 21,31%, а Концепції 5 — 0%. Це демонструє низький рівень реалізації їхніх специфікацій порівняно з лідерами.

Інноваційний потенціал

Інноваційний потенціал обчислюється на основі бізнес-факторів, зокрема:

- Порівняння з ринком,
- Економічна ефективність,
- Маркетингова оцінка,

- Стратегічна відповідність.

Кожен фактор має свій ваговий коефіцієнт (0.2, 0.4, 0.2, 0.2 відповідно), який використовується для обчислення загального показника.

Концепція 1 (власна) має найвищий інноваційний потенціал — 107,47, що підтверджує її збалансованість за всіма критеріями.

Концепція 3 (Система перевірки завдань) з результатом 104,60 займає друге місце. Вона має сильні позиції на ринку, але поступається власній концепції через меншу економічну ефективність (55%).

Інші концепції значно відстають. Наприклад, Концепція 4 (Генератор текстів) має лише 28,21, що свідчить про обмежену бізнесову привабливість і новизну.

Можливість реалізації

Оцінка можливості реалізації показує, чи може концепція бути успішно втілена в життя. З п'яти концепцій лише дві — Концепція 1 (власна) та Концепція 3 — отримали статус "Yes", що вказує на їхню перспективність. Інші концепції, зокрема Концепція 2, Концепція 4 і Концепція 5, були позначені як "No", що свідчить про наявність значних обмежень для їхньої реалізації.

Ступінь новизни

Ступінь новизни визначає, наскільки оригінальними є специфікації концепції. Максимальний бал (9) отримали:

Концепція 1 (власна), яка пропонує унікальні функції, як-от інтуїтивний інтерфейс, рекомендації щодо покращення текстів та виявлення фрагментів, згенерованих ШІ.

Концепція 3 (Система перевірки завдань), що також має високі показники новизни.

Інші концепції отримали середні або низькі оцінки, що обмежує їхній інноваційний потенціал.

Важливість

Концепція 1 (власна) займає перше місце, оскільки демонструє найвищий потенціал і новизну. Концепція 3 (Система перевірки завдань) є другою за значущістю завдяки високому нормалізованому та інноваційному потенціалу. Концепція 4 (Генератор текстів), попри обмежений нормалізований потенціал, має певну бізнесову цінність. Концепція 2 (Сервіс аналізу кар’єри) та 5. Концепція 5 (Моніторинг фейкових новин) мають найнижчі оцінки і потребують значного доопрацювання.

		Conceptual Design	Концепція 1 (власна)	Концепція 2(Сервіс аналізу кар’єри)	Концепція 3 (Система перевірки завдань)	Концепція 4 (Генератор текстів для копірайтерів)	Концепція 5 (Моніторинг фейкових новин)
		Degree of novelty	9	3	9	3	3
№	Design Specifications	Specification weighting					
1	Інтуїтивний інтерфейс	3	9	3	9	3	3
2	Перевірка ресурсів пристрою	3	3	3	1	1	1
3	Виявлення фрагментів, згенерованих ШІ	9	9	1	9	9	3
4	Сповіщення про помилки	3	9	3	9	9	1
5	Журнал запитів	1	3	9	3	1	1
6	Візуалізація даних	3	9	3	9	3	3
7	Рекомендації щодо покращення текстів	9	9	3	9	9	1
		Absolute potential	2295	243	2241	633	183
		Normalized potential	100	2,840909091	97,44318182	21,30681818	0
	Bussiness feasibility	Weighting factor					
	Comparison with the market	0,2	81,37	75	88	80	68
	Economic efficency	0,4	60	70	55	50	65
	Marketing assessment	0,2	84,33	88	80	83	87
	Strategic fit	0,2	87,67	85	80	82	90
	INNOVATION POTENTIAL		107,4674	10,60090909	104,6031818	28,20681818	7,5
	Potentiability (Yes/No)		Yes	No	Yes	No	No
	Order of Importance		**1**	4	2	3	5

Рис. 4.1. Матриця оцінки інноваційного потенціалу із розрахунками

4.5. Аналіз результатів

Концепція 1 (власна) має найвищий інноваційний потенціал (107,47) і значно перевершує інші концепції. Вона набрала максимальні оцінки за ключовими характеристиками, такими як "Інтуїтивний інтерфейс", "Виявлення фрагментів, згенерованих ШІ", "Сповіщення про помилки", та "Рекомендації щодо покращення текстів".

Концепція 3 (Система перевірки завдань) займає друге місце з потенціалом 104,60, що також демонструє високий рівень новизни та бізнесової доцільності, але трохи поступається власній концепції через нижчі показники бізнес-факторів, зокрема економічної ефективності (55).

Інші концепції мають значно нижчі показники, наприклад, Концепція 4 (Генератор текстів) отримала 28,20, що свідчить про обмежену новизну та середню бізнесову доцільність. Концепція 5 (Моніторинг фейкових новин) має найнижчий потенціал — 7,5.

Власна концепція має високі бали новизни за всіма специфікаціями, особливо за:

"Виявлення фрагментів, згенерованих ШІ" (вплив ваги 9) — критична функція, яка надає унікальну цінність продукту.

"Рекомендації щодо покращення текстів" (вплив ваги 9) — висока корисність для кінцевих користувачів.

У бізнес-факторах концепція виділяється стратегічною відповідністю (87,67) і маркетинговою оцінкою (84,33), що свідчить про сильні позиції на ринку.

Пояснення рівня інноваційного потенціалу

Високий інноваційний потенціал вашого продукту базується на:

- Максимальній новизні за більшістю характеристик.
- Високій вазі специфікацій, що мають прямий вплив на зручність використання та функціональність (наприклад, "Інтуїтивний інтерфейс").
- Сильних бізнес-показниках, що підкреслюють можливість успішної реалізації продукту.

Порівняльний аналіз

Власна концепція демонструє суттєву перевагу в інноваційності та бізнесовій доцільності. Концепція 3 є найближчим конкурентом, але поступається через слабкіший баланс бізнес-факторів. Решта концепцій мають вузьку спрямованість і обмежену цінність.

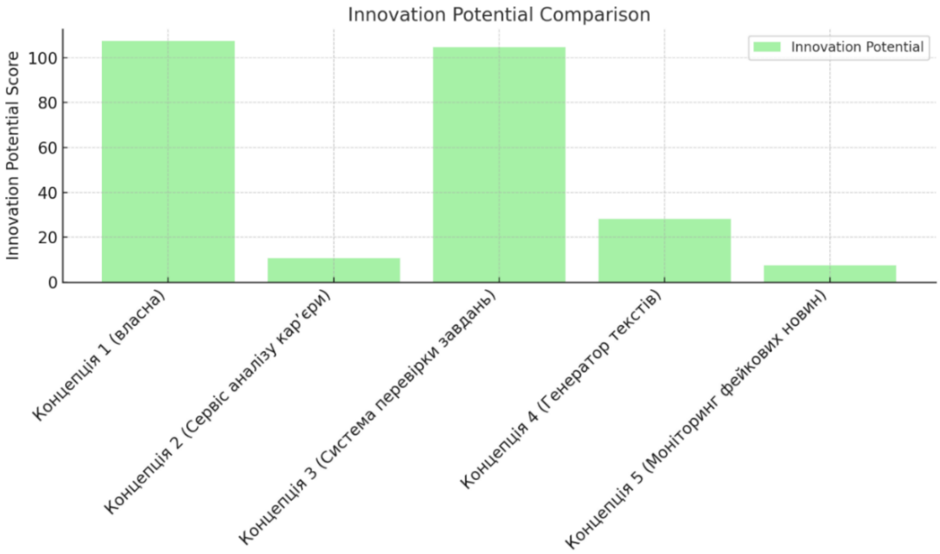


Рис. 4.2. Візуалізація інноваційного потенціалу продукту та конкурентів

Веб - детектор ШІ українською є лідером за рівнем інноваційного потенціалу завдяки унікальним функціям і високій новизні. Баланс між інноваційними специфікаціями та бізнес-факторами робить концепцію перспективною на ринку.

Рекомендації на покращення:

- Посилити економічну ефективність через оптимізацію процесів, щоб конкурувати з концепцією 3, яка має сильніший показник у цьому аспекті.
- Розвивати функції для ширшого використання в інших галузях, щоб підвищити привабливість для різних сегментів ринку.
- Інвестувати в маркетинг, оскільки високі оцінки у "маркетинговій оцінці" підкреслюють потенціал для залучення великої аудиторії.
- Провести додаткові дослідження потреб ринку, щоб краще розкрити можливості функцій, як-от "Рекомендації щодо покращення текстів".

ВИСНОВКИ до розділу 4

Четвернитий розділ присвячений розробленню прототипу інноваційного продукту та оцінці його інноваційного потенціалу. Тестування прототипу підтвердило його функціональність і відповідність заданим вимогам, а оцінка новизни, інноваційних специфікацій та бізнесової доцільності засвідчила високу конкурентоспроможність продукту. Використання матриці для розрахунку інноваційного потенціалу та аналіз результатів дозволили обґрунтувати доцільність впровадження продукту, підкресливши його здатність задовольняти потреби ринку та сприяти технологічному прогресу.

ВИСНОВКИ

У процесі виконання дипломної роботи та створення веб-додатку для виявлення використання штучного інтелекту (ШІ) в українських текстах було досліджено інноваційні аспекти сучасних технологій. Для генерації ідей застосовувалися методи мозкового штурму та можливості ШІ. Використання алгоритмів штучного інтелекту й машинного навчання значно підвищило швидкість і точність обробки текстових даних.

Розроблений веб-додаток із елементами ШІ відповідає актуальним тенденціям автоматизації, що забезпечує його унікальність і конкурентоспроможність. Додаток створено на основі інноваційних підходів, які включають аналіз контексту українських текстів за допомогою алгоритмів машинного навчання. Це дозволяє розробити продукт, що відповідає потребам користувачів у сфері лінгвістичного аналізу.

Було проведено моделювання об’єктів предметної області, таких як структури текстів, маркери використання ШІ та сценарії взаємодії користувачів із додатком. Це забезпечило високу ефективність обробки текстових даних.

Визначено ключові вимоги до функціональності продукту, зокрема адаптивність, інтуїтивно зрозумілий інтерфейс і високу швидкодію. Завдяки ретельному моделюванню враховано особливості української мови та її унікальні характеристики.

Створено прототип веб-додатку, який успішно пройшов тестування на точність ідентифікації текстів, створених ШІ. Результати тестування підтвердили відповідність функціоналу встановленим вимогам.

Оцінка інноваційного потенціалу додатку проводилася за такими критеріями:

- Новизна концепції: Додаток є унікальним для українського ринку завдяки врахуванню мовної специфіки.

- Інноваційні характеристики: Продукт оснащено сучасними функціями з можливістю інтеграції з іншими системами.

- Бізнесова доцільність: Додаток має значні перспективи через зростаючий попит на автоматизацію текстового аналізу.

- Матриця потенціалу: Аналіз засвідчив високу конкурентоспроможність продукту.

За результатами тестування веб-додаток відповідає заявленим вимогам і демонструє значний потенціал для впровадження на ринок.

Розроблений веб-додаток для виявлення ШІ у текстах є інноваційним, ефективним і сучасним інструментом, що відповідає потребам інформаційного суспільства. Він не лише автоматизує процес аналізу текстів, але й створює нові перспективи для розвитку лінгвістичних технологій в Україні.

СПИСОК ЛІТЕРАТУРИ

1. Шумпетер Й. Теорія економічного розвитку: дослідження підприємницького прибутку, капіталу, кредиту, відсотка і циклу кон'юнктури / Й. Шумпетер ; пер. з англ. В. С. Автономова. – Київ : Вища школа, 2021. – 456 с.
2. Christensen C. M. The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail / C. M. Christensen. – Boston : Harvard Business Review Press, 2017. – 286 p.
3. Brynjolfsson E. The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies / E. Brynjolfsson, A. McAfee. – New York : W. W. Norton & Company, 2014. – 320 p.
4. OECD. Oslo Manual 2018: Guidelines for Collecting, Reporting and Using Data on Innovation. – 4th ed. – Paris : OECD Publishing, 2018. – 256 p.
5. Gilder G. Microcosm: The Quantum Revolution in Economics and Technology / G. Gilder. – New York : Simon & Schuster, 1989. – 352 p.
6. Freeman C. The Economics of Industrial Innovation / C. Freeman. – 2nd ed. – London : Routledge, 2022. – 512 p.
7. Porter M. E. Competitive Advantage: Creating and Sustaining Superior Performance / M. E. Porter. – New York : Free Press, 2018. – 592 p.
8. Unified Modeling Language Specification [Електронний ресурс] / Object Management Group. – Режим доступу: https://www.omg.org/spec/UML/. – Дата доступу: 01.04.2025.
9. Lucidchart. UML Activity Diagram Tutorial [Електронний ресурс]. – Режим доступу: https://www.lucidchart.com/pages/uml-activity-diagram. – Дата доступу: 15.04.2025.
10. McKinsey & Company. Business Case Development Guide [Електронний ресурс]. – Режим доступу: https://www.mckinsey.com/business-case-guide. – Дата доступу:20.04.2025.

11. Jurafsky D. Speech and Language Processing / D. Jurafsky, J. H. Martin. – 3rd ed. – Upper Saddle River : Prentice Hall, 2020. – 624 p.
12. Vaswani A. Attention is All You Need / A. Vaswani, N. Shazeer, N. Parmar et al. // Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – P. 5998–6008.
13. Hugging Face. Transformers Documentation [Електронний ресурс]. – Режим доступу: https://huggingface.co/docs/transformers/. – Дата доступу: 01.04.2025.
14. Devlin J. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Електронний ресурс] / J. Devlin, M.-W. Chang, K. Lee, K. Toutanova. – Режим доступу: https://arxiv.org/abs/1810.04805. – Дата доступу: 22.04.2025.
15. Schwab K. The Fourth Industrial Revolution / K. Schwab. – Geneva : World Economic Forum, 2017. – 184 p.

ДОДАТОК

```
import nltk
import pandas as pd
import numpy as np
from flask import Flask, request, render_template
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import re
import string
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer,
TrainingArguments
import torch

# Завантаження необхідних компонентів NLTK
nltk.download('punkt')
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize

# Ініціалізація Flask
app = Flask(__name__)

# Функція для попередньої обробки тексту
def preprocess_text(text):
    # Переведення в нижній регістр
    text = text.lower()
    # Видалення пунктуації
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Токенізація
    tokens = word_tokenize(text, language='ukrainian')
    # Видалення стоп-слів
    stop_words = set(stopwords.words('ukrainian'))
    tokens = [word for word in tokens if word not in stop_words]
    return ' '.join(tokens)

# Функція для вилучення ознак тексту
def extract_features(text):
    # Попередня обробка
    processed_text = preprocess_text(text)
    # Довжина тексту
    length = len(text)
    # Кількість речень
    sentences = sent_tokenize(text, language='ukrainian')
    num_sentences = len(sentences)
```

```
    # Середня довжина речення
    avg_sentence_length = length / num_sentences if num_sentences > 0 else 0
    # Унікальність лексики
    words = word_tokenize(text, language='ukrainian')
    unique_words = len(set(words)) / len(words) if words else 0
    return [length, num_sentences, avg_sentence_length, unique_words]

# Завантаження або створення датасету (приклад)
def load_dataset():
    # Для прикладу: синтетичний датасет
    data = {
        'text': [
            'Це приклад тексту, написаного людиною. Він має природний стиль.',
            'Штучний інтелект створив цей текст. Він виглядає дещо шаблонно.',
            # Додайте більше прикладів
        ],
        'label': [0, 1]  # 0 - людина, 1 - ШІ
    }
    df = pd.DataFrame(data)
    return df

# Навчання моделі Random Forest
def train_rf_model(df):
    # Вилучення текстових ознак з TF-IDF
    tfidf = TfidfVectorizer(max_features=500)
    X_tfidf = tfidf.fit_transform(df['text']).toarray()
    # Вилучення додаткових ознак
    X_features = np.array([extract_features(text) for text in df['text']])
    # Об'єднання ознак
    X = np.hstack((X_tfidf, X_features))
    y = df['label']

    # Розбиття на тренувальну та тестову вибірки
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Навчання моделі
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)

    # Оцінка точності
    y_pred = model.predict(X_test)
    print(f"Accuracy: {accuracy_score(y_test, y_pred)}")

    return model, tfidf

# Навчання моделі BERT (як альтернатива)
```

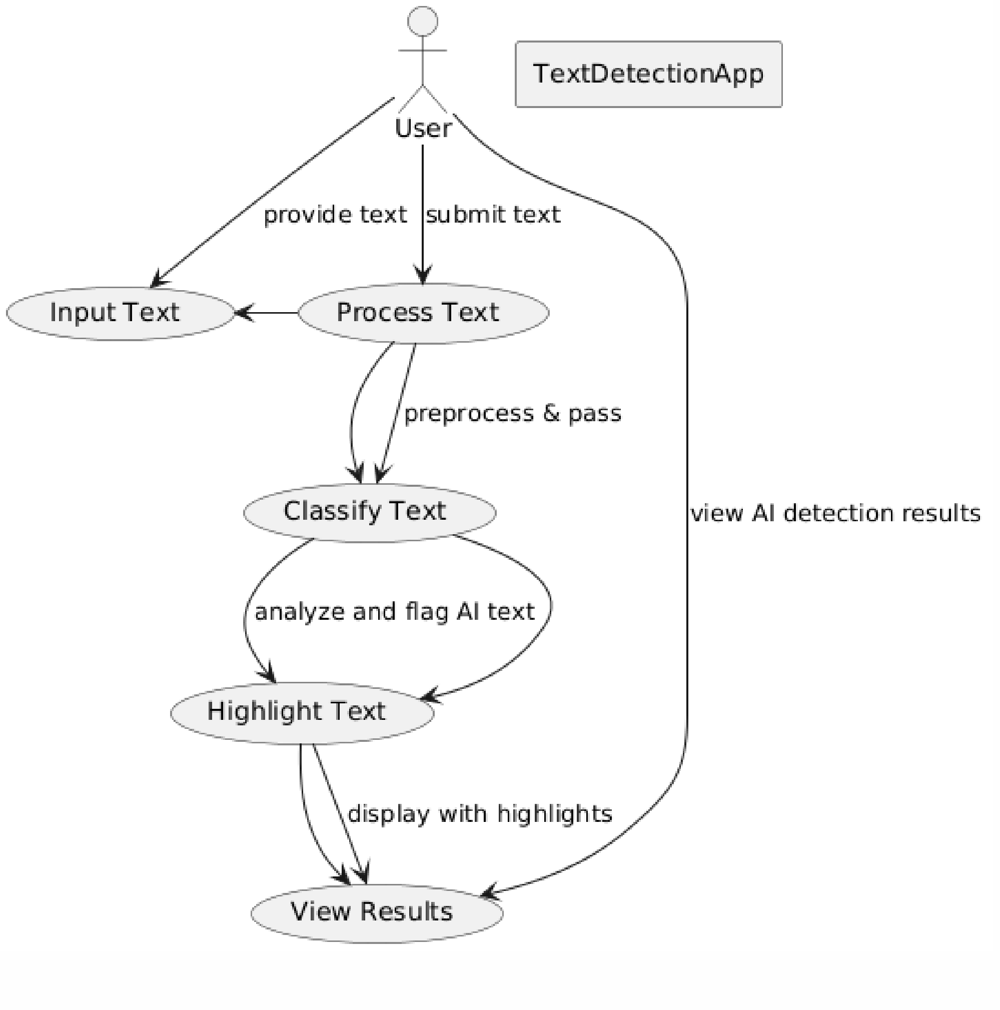
```
def train_bert_model(df):
    tokenizer = AutoTokenizer.from_pretrained("ukr-roberta-base")
    model = AutoModelForSequenceClassification.from_pretrained("ukr-roberta-base",
num_labels=2)

    # Токенізація текстів
    encodings = tokenizer(list(df['text']), truncation=True, padding=True, max_length=512)
    labels = list(df['label'])

    # Налаштування тренування
    training_args = TrainingArguments(
        output_dir='./results',
        num_train_epochs=3,
        per_device_train_batch_size=8,
        per_device_eval_batch_size=8,
        warmup_steps=500,
        weight_decay=0.01,
        logging_dir='./logs',
    )

    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=encodings,
        eval_dataset=encodings,
    )

    trainer.train()
    return model, tokenizer

# Завантаження датасету та навчання моделі
df = load_dataset()
rf_model, tfidf = train_rf_model(df)
# bert_model, tokenizer = train_bert_model(df)  # Розкоментувати для використання BERT

# Маршрут для головної сторінки
@app.route('/', methods=['GET', 'POST'])
def index():
    result = None
    if request.method == 'POST':
        text = request.form['text']
        # Попередня обробка тексту
        processed_text = preprocess_text(text)
        # Вилучення ознак
        text_tfidf = tfidf.transform([processed_text]).toarray()
        text_features = np.array([extract_features(text)])
```

```
        text_input = np.hstack((text_tfidf, text_features))

        # Передбачення
        prediction = rf_model.predict_proba(text_input)[0]
        ai_prob = prediction[1] * 100  # Ймовірність, що текст створений ШІ
        result = f"Ймовірність, що текст створений ШІ: {ai_prob:.2f}%"

    return render_template('index.html', result=result)

if __name__ == '__main__':
    app.run(debug=True)


import spacy
import psycopg2
import numpy as np
from transformers import pipeline
from sentence_transformers import SentenceTransformer
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
from collections import Counter
import shap

# Ініціалізація моделей
nlp = spacy.load("uk_core_news_sm")  # Українська модель spaCy
classifier = pipeline("text-classification", model="ukr-bert-model")  # Приклад BERT
sentence_model = SentenceTransformer("distiluse-base-multilingual-cased-v2")  # Для семантики
explainer = shap.KernelExplainer(classifier.predict, shap.sample([[]], 100))  # SHAP

# Словники
stop_words = {"і", "на", "в", "це", "та"}  # Спрощений словник стоп-слів
emotional_words = {"радість", "сум", "любов", "страх"}  # Емоційні слова

def clean_text(text):
    """Очищення тексту"""
    text = text.lower().replace("\n", " ").strip()
    text = "".join(c for c in text if c.isalnum() or c.isspace())
    return text

def tokenize_and_lemmatize(text):
    """Токенізація та лематизація"""
    doc = nlp(text)
    tokens = [token.lemma_ for token in doc if not token.is_stop and token.text not in stop_words]
    return tokens, doc

def compute_ttr(tokens):
    """Обчислення Type-Token Ratio"""
    return len(set(tokens)) / len(tokens) if tokens else 1.0
```

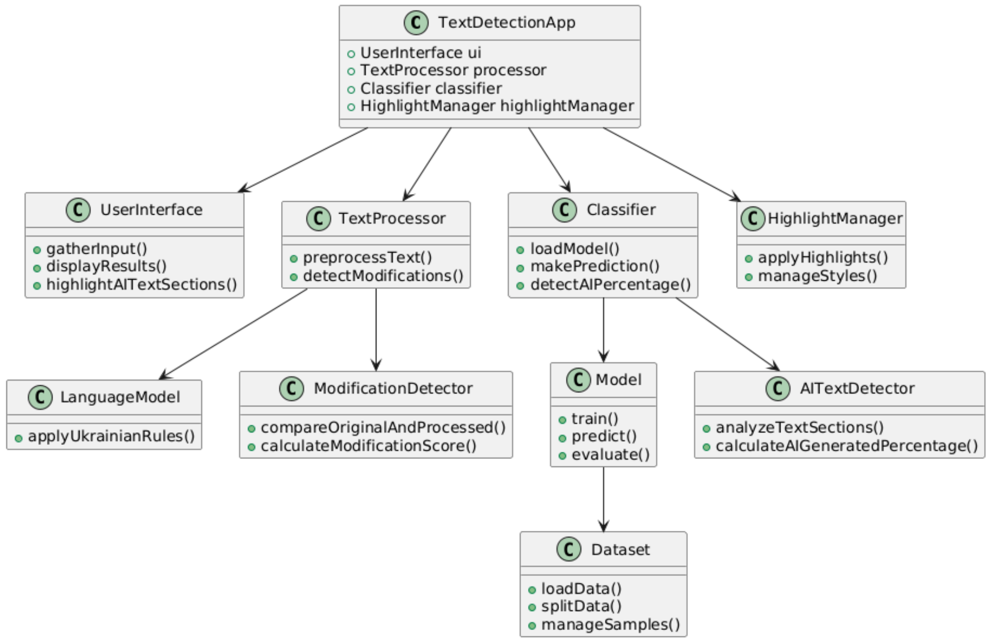
```
def compute_ngram_frequency(tokens, n=2):
    """Частота N-грам"""
    ngrams = [" ".join(tokens[i:i+n]) for i in range(len(tokens)-n+1)]
    ngram_counts = Counter(ngrams)
    return len(ngrams) / len(set(ngrams)) if ngrams else 1.0

def compute_emotional_score(tokens):
    """Оцінка емоційності"""
    return sum(1 for token in tokens if token in emotional_words) / len(tokens) if tokens else 
0.0

def compute_syntax_complexity(doc):
    """Синтаксична складність"""
    sentences = list(doc.sents)
    avg_sentence_length = sum(len(sent) for sent in sentences) / len(sentences) if sentences 
else 1.0
    return avg_sentence_length

def compute_coherence(doc):
    """Оцінка когерентності"""
    sentences = [sent.text for sent in doc.sents]
    if len(sentences) < 2:
        return 1.0
    embeddings = sentence_model.encode(sentences)
    similarities = [np.dot(embeddings[i], embeddings[i+1]) / (np.linalg.norm(embeddings[i]) * 
np.linalg.norm(embeddings[i+1])) for i in range(len(embeddings)-1)]
    return np.mean(similarities) if similarities else 1.0

def compute_entropy(tokens):
    """Обчислення ентропії"""
    counts = Counter(tokens)
    total = len(tokens)
    probs = [count/total for count in counts.values()]
    return -sum(p * np.log2(p) for p in probs if p > 0) if probs else 0.0

def extract_features(tokens, doc):
    """Витягування всіх ознак"""
    return {
        "ttr": compute_ttr(tokens),
        "ngram_freq": compute_ngram_frequency(tokens, n=2),
        "emotional_score": compute_emotional_score(tokens),
        "syntax_complexity": compute_syntax_complexity(doc),
        "coherence": compute_coherence(doc),
        "entropy": compute_entropy(tokens)
    }

def generate_pdf_report(report):
    """Створення PDF-звіту"""
    c = canvas.Canvas("report.pdf", pagesize=letter)
```

```
    c.drawString(100, 750, f"Ймовірність ШІ: {report['probability']:.2f}%")
    c.drawString(100, 730, f"Пояснення: {report['explanation']}")
    c.drawString(100, 710, f"Ознаки: {report['features']}")
    c.save()
    return "report.pdf"

def check_text(text, model, db_connection):
    """Основна функція перевірки"""
    # 1. Попередня обробка
    text = clean_text(text)
    tokens, doc = tokenize_and_lemmatize(text)

    # 2. Витягування ознак
    features = extract_features(tokens, doc)
    feature_vector = np.array([features["ttr"], features["ngram_freq"],
features["emotional_score"],
                          features["syntax_complexity"], features["coherence"],
features["entropy"]])

    # 3. Класифікація
    result = model(text)
    probability = result[0]["score"] if result[0]["label"] == "AI" else 1 - result[0]["score"]

    # 4. Пояснення (SHAP)
    shap_values = explainer.shap_values(feature_vector)
    explanation = f"Ключові фактори: TTR={features['ttr']:.2f} ({shap_values[0]:.2f}),
Емоційність={features['emotional_score']:.2f} ({shap_values[2]:.2f})"

    # 5. Формування звіту
    report = {
        "probability": probability * 100,
        "explanation": explanation,
        "features": features,
        "text": text
    }
    pdf_file = generate_pdf_report(report)

    # 6. Зберігання
    cursor = db_connection.cursor()
    cursor.execute(
        "INSERT INTO analysis_history (text, probability, explanation, features) VALUES (%s,
%s, %s, %s)",
        (text, probability, explanation, str(features))
    )
    db_connection.commit()

    return report, pdf_file

# Приклад використання
text = "Штучний інтелект швидко розвивається і змінює світ."
```

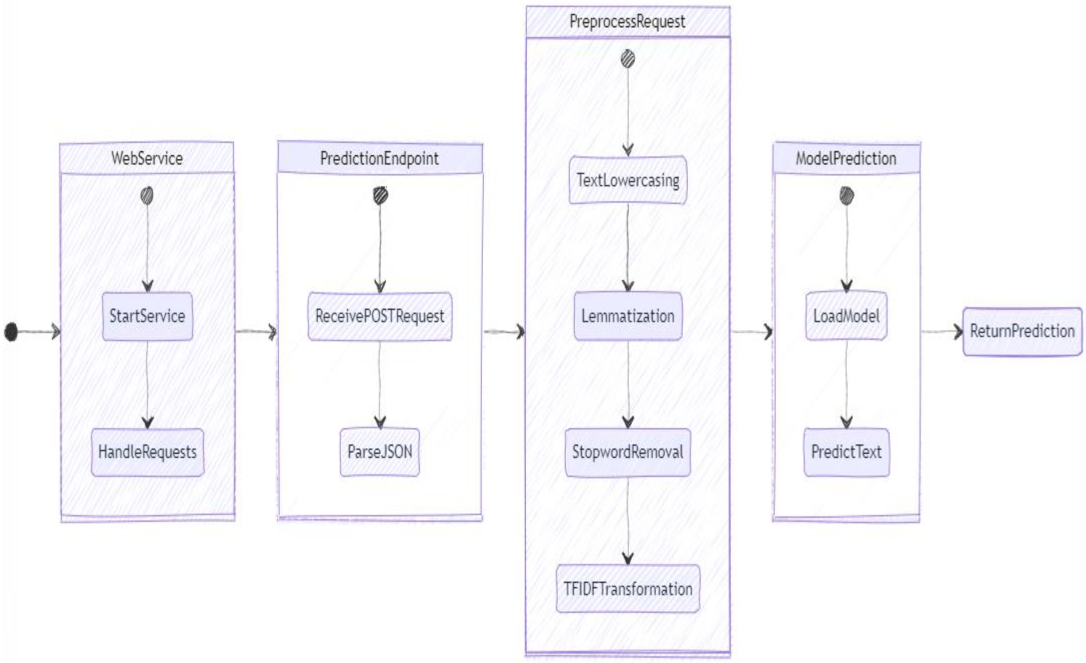
```
db = psycopg2.connect(database="analysis_db", user="user", password="pass", host="localhost")
report, pdf = check_text(text, classifier, db)
print(f"Ймовірність ШІ: {report['probability']:.2f}%, Пояснення: {report['explanation']}")
```



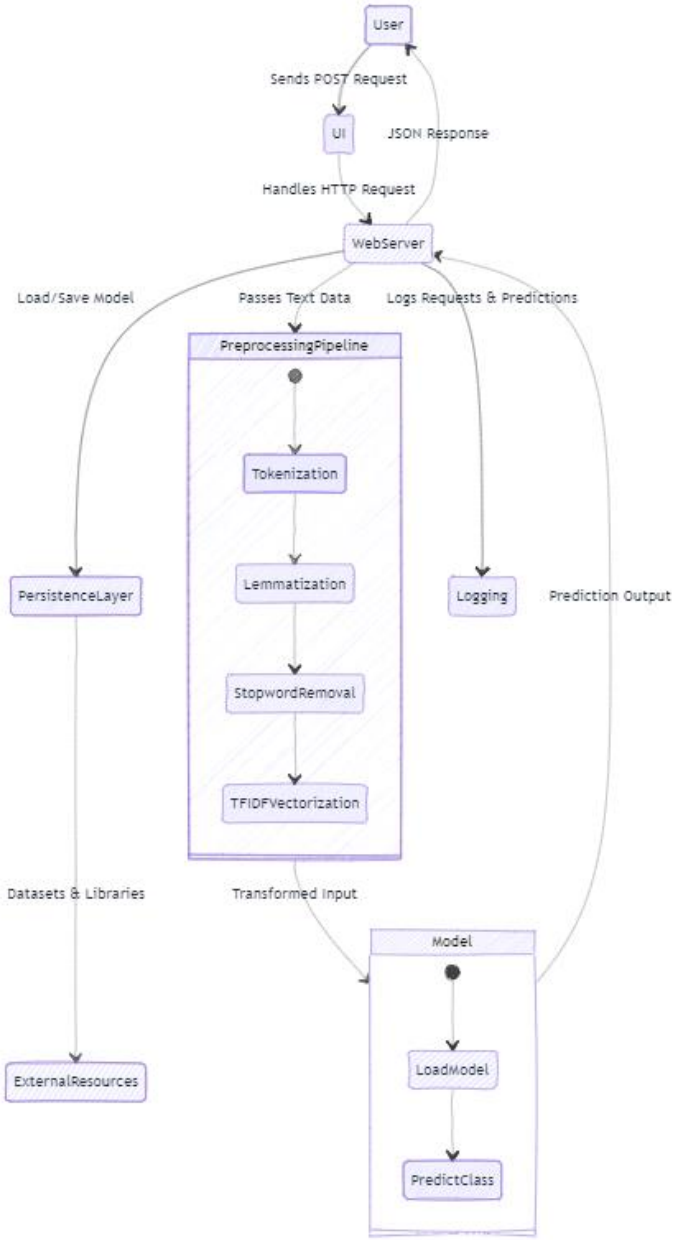
Змн.	Арк.	Прізвище	Підпис	Дата	Бакалаврська кваліфікаційна робота		
Розроб.					*UML-діаграм варіантів використання*	Літера: Н	Маса	Масштаб
Керівник						Аркуш 1	Аркушів 6
Н. Контр.						НУ «ЛП», ІКНІ, каф. САП, гр. КН-412	
Зав. каф.		Лобур М.В.					



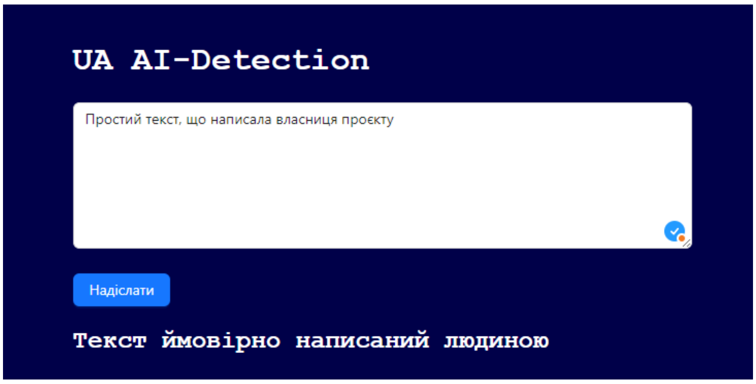
Змн.	Арк.	Прізвище	Підпис	Дата	
Розроб.					Бакалаврська кваліфікаційна робота
Керівник					*UML діаграма класів*
Н. Контр.					Літера: Н; Маса; Масштаб; Аркуш 1; Аркушів 6
Зав. каф.		Лобур М.В.			НУ «ЛП», ІКНІ, каф. САП, гр. КН-412



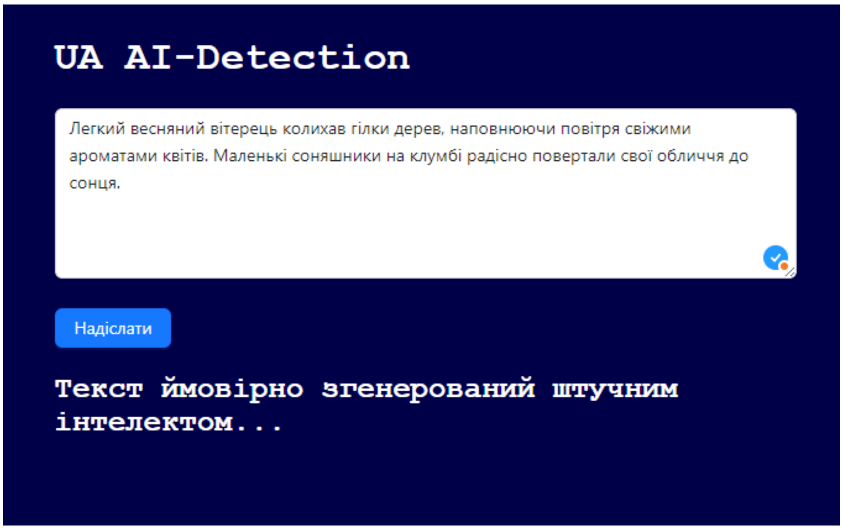
Змн.	Арк.	Прізвище	Підпис	Дата	Бакалаврська кваліфікаційна робота	
Розроб.					*Схема функціональності*	Літера: Н; Маса; Масштаб
Керівник						Аркуш 1; Аркушів 6
Н. Контр.						НУ «ЛП», ІКНІ, каф. САП, гр. КН-412
Зав. каф.		Лобур М.В.				



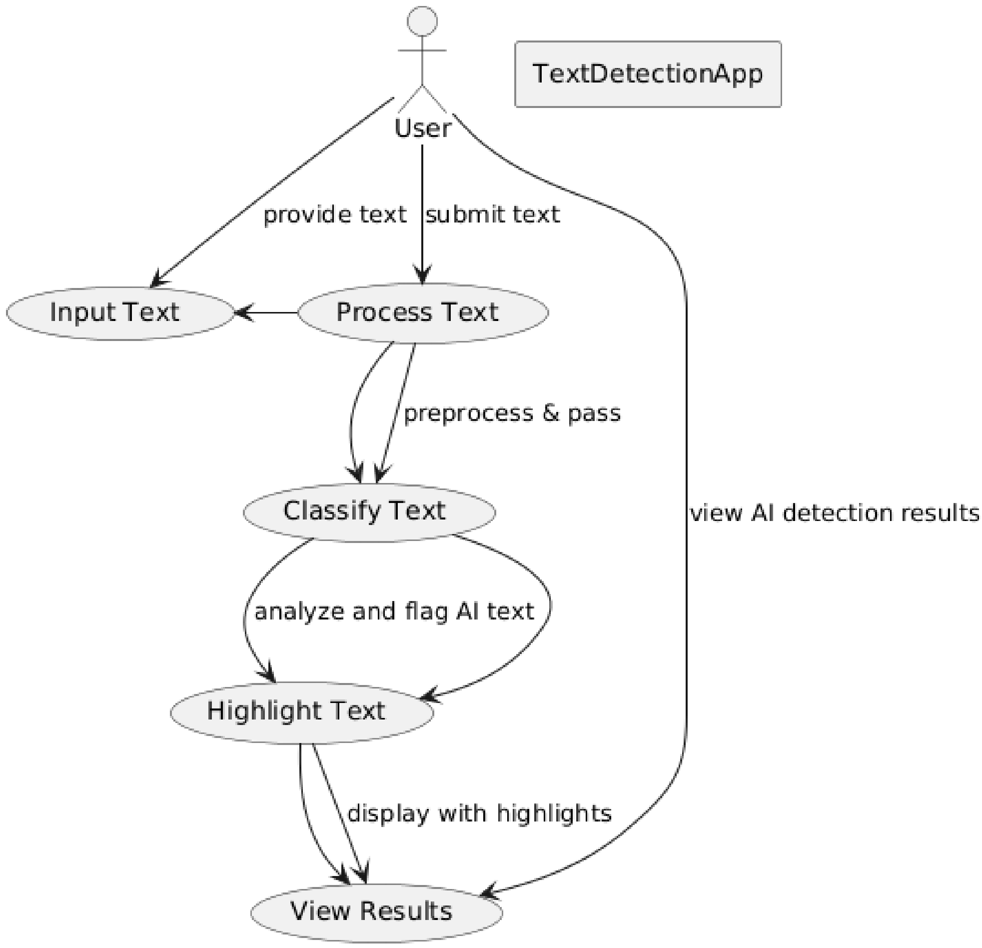
Змн.	Арк.	Прізвище	Підпис	Дата			
Розроб.					Бакалаврська кваліфікаційна робота		
Керівник					*Концептуальна схема роботи*	Літера: Н	Маса / Масштаб
						Аркуш 1	Аркушів 6
Н. Контр.						НУ «ЛП», ІКНІ, каф. САП, гр. КН-412	
Зав. каф.		Лобур М.В.					



Потенційний результат аналізу у прототипу.



Потенційний результат аналізу у прототипу.



Змн.	Арк.	Прізвище	Підпис	Дата
Розроб.				
Керівник				
Н. Контр.				
Зав. каф.		Лобур М.В.		

Бакалаврська кваліфікаційна робота

UML-діаграм варіантів використання

Літера	Маса	Масштаб
Н		
Аркуш 1	Аркушів 6	

НУ «ЛП», ІКНІ,
каф. САП, гр. КН-412