

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ _____ ” _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні управляючі системи
та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Система для аналізу, фільтрації та структурування
контенту».

Виконав:

студент IV курсу, групи ІС-361

Патлай Микита Олегович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

доц., к.т.н. Коган Алла Вікторівна

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

_____ (підпис)

**Консультант з
графічної
документації**

к.т.н., доц. Телишева Тамара Олексіївна

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

_____ (підпис)

Рецензент

к.т.н., доц. Роковий Олександр Петрович

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

_____ (підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(в.і.м'я, прізвище)

(підпис)

“ ___ ” _____ 2020 р.

ЗАВДАННЯ
на дипломний проект студенту

Патлай Микиті Олександровичу
(прізвище, ім'я, по батькові)

1. Тема проекту «Система для аналізу, фільтрації та структурування контенту».

керівник проекту Коган Алла Вікторівна, к.т.н.

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

затверджені наказом по університету від “7”травня 2020 р. №1081-с

2. Термін подання студентом проекту “01”червня 2020 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до

технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна станів системи

2. Схема структурна діяльності

3. Схема структурна варіантів використання

4. Схема структурна компонентів програмного забезпечення

5. Схема структурна класів програмного забезпечення

6. Схема структурна послідовності

7. Схема структурна компонентів програмного забезпечення

6. Консультанти розділів проекту

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

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	13.04.2020	
2.	Аналіз існуючих методів розв'язання задачі	16.04.2020	
3.	Постановка та формалізація задачі	19.04.2020	
4.	Розробка інформаційного забезпечення	23.04.2020	
5.	Алгоритмізація задачі	25.04.2020	
6.	Обґрунтування використовуваних технічних засобів	30.04.2020	
7.	Розробка програмного забезпечення	5.05.2020	
8.	Налагодження програми	10.05.2020	
9.	Виконання графічних документів	12.05.2020	
10.	Оформлення пояснювальної записки	13.05.2020	
11.	Подання ДП на попередній захист	15.05.2020	
12.	Подання ДП на основний захист	01.06.2020	
13.	Подання ДП рецензенту	02.06.2020	

Студент

Микита ПАТЛАЙ

Керівник

Алла КОГАН

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ДП 6206.00.000 ПЗ	Пояснювальна записка	70	
3	A4	ДП 6206.01.000 ТЗ	Технічне завдання	7	
4	A4	ДП 6206.02.000.ССС	Схема структурна станів системи	1	
5	A4	ДП 6206.03.000.ССД	Схема структурна діяльності	1	
6	A4	ДП 6206.04.000.ССВ	Схема структурна варіантів використань	1	
7	A4	ДП 6206.05.000.ССК	Схема структурна компонентів програмного забезпечення	1	
8	A4	ДП 6206.06.000.ССК	Схема структурна класів програмного забезпечення	1	
9	A4	ДП 6206.07.000.ССП	Схема структурна послідовності	1	
10	A4	ДП 6206.08.000.ССК	Схема структурна компонентів програмного забезпечення	1	

ДП 6206 00.000.00

Зм.	Арк.	ПІБ	Підп.	Дата
Розробн.		Патлай М.О.		
Керівн.		Коган А.В.		
Консульт.				
Н/контр.		Телишева Т.О.		
В.о.зав.каф.		Павлов О.А.		

Відомість
дипломного проєкту

Лім.	Лист	Листів
	1	1

КПІ ім. Ігоря Сікорського
Каф. АСОІУ
Гр. ІС-361

**Пояснювальна записка
до дипломного проєкту**

на тему: Система для аналізу, фільтрації та структурування контенту

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з шести розділів, містить 13 рисунків, 2 таблиці, 6 джерел.

Дипломний проект присвячений розробці алгоритму для аналізу, фільтрації та структурування контенту. У дипломному проекті розроблено алгоритм для аналізу контенту у джерелах інформації.

Розділ загальних положень включає в себе опис предметного середовища, огляд наявних аналогів та постановку задачі. Також цей розділ включає опис діяльності та функціональних складових системи.

У розділі інформаційного забезпечення були виявлені вхідні та вихідні дані.

Розділ математичного забезпечення присвячений огляду наявних методів розробки програмної реалізації.

У розділі програмного забезпечення вказано діаграми послідовності, класів, компонентів, специфікацію функцій. Також представлено опис архітектури системи, програмного та апаратного забезпечення.

У технологічному розділі наведено керівництво користувача, опис випробувань програмної реалізації.

АЛГОРИТМ, АНАЛІЗ, КОНТЕНТ, СТРУКТУРУВАННЯ, ФІЛЬТРАЦІЯ,
ТЕКСТ, ГРАФІЧНИЙ КОНТЕНТ

					ДП 6206.00.000 ПЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дат</i>		<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Розробн.</i>		Патлай М.О.			Система для аналізу, фільтрації та структурування контенту		2	
<i>Керівн.</i>		Коган А.В.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
<i>Н/контр.</i>		Телишева Т.О.						
<i>В.о.зав. каф</i>		Павлов О.А.						

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of six sections, contains 13 figures, 2 tables, 6 sources.

Diploma project is dedicated to the development of an algorithm for content analysis, filtering and structuring. In the diploma project the algorithm for the analysis of content in information sources is developed.

The section of general provisions includes a description of the subject environment, an overview of existing analogues and problem statement. This section also includes a description of the activities and functional components of the system.

Input and output data were found in the information support section.

The section of mathematical software is devoted to the review of available methods of software implementation development.

The software section contains sequence diagrams, classes, components, function specifications. There is also a description of the system architecture, software and hardware.

The technological section provides a user manual, a description of software implementation tests.

ALGORITHM, ANALYSIS, CONTENT, STRUCTURING, FILTRATION,
TEXT, GRAPHIC CONTENT

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

В наші дні контент надзвичайно популярна, корисна і не менш важлива частина нашого життя. Люди можуть отримувати свіжі новини з різноманітних джерел, таких як телебачення, газета, журнал, і в даний час найпоширенішим джерелом є Інтернет. Насправді, далеко не всі веб-сайти чи інші ресурси можуть надавати користувачеві правдиву інформацію через різні фактори. На жаль, цей тип джерел має левову частку серед усіх інших.

Більш того, контент сьогодні надзвичайно багато, і далеко не весь контент наразі структурований по різним критеріям, таким, як наприклад “Емоціональне забарвлення тексту”, що є надзвичайно важливим критерієм так як контролюючи емоціональне забарвлення тексту, тобто поділяти його на позитивне або негативне забарвлення, можна досягти більшого балансу між користувачем та ресурсом, запропонувати користувачу більш відповідний для нього результат і так далі. Також, є потреба аналізувати та фільтрувати графічний контент як фотографії або відео.

Дипломний проект присвячений розробці комплексу задач для поліпшення пошуку інформації у мережі Інтернет та у інших джерелах інформації, створення більш безпечного пошуку інформації, більш безпечного перегляду контенту, фільтрації контенту по різним налаштуванням та аналізу контенту по різним параметрам, таким як емоційне забарвлення тексту, виділення відомих для алгоритму людей/об’єктів, у майбутньому вміння розпізнавати фейкову інформацію у різних форматах контенту.

Практичне значення одержаних результатів. Розроблено алгоритм оптимізації, структурування, фільтрації та аналізу контенту.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Проблема фільтрації та аналізу контенту у різних джерелах інформації це справді величезна неприємність для сучасного суспільства. Таким чином засоби масової інформації та уряд можуть поширювати підроблені новини для отримання більшого контролю над людьми. Другий приклад проблеми поширення неправдивої інформації - це, безумовно, марна трата особистого часу. Для прикладу, справедливо сказати, що в назвах відео, статей чи просто публікацій у соціальних медіа є чимало назв-наживок для переходу. Користувач може зрозуміти, що ця стаття або відео підробка лише тоді, коли прочитає або подивиться.

Безумовно цю проблему можна вирішити сучасними комп'ютерними інструментами та парадигмами. Існує три основні поняття, за допомогою яких можна вирішити цю проблему: AI (штучний інтелект), ML (машинне навчання), DL (глибоке навчання) та допоміжні нейронні мережі. Є набір технологій, які реалізують ці поняття. Серед них: TensorFlow, PyTorch, Keras і так далі.

Користуючись мережею Інтернет, користувач може натрапляти на контент, який йому не цікавий або не потрібний. Як приклад можна сказати, що користувач цікавиться політикою та економікою країни та світу, а йому в рекомендації на тематичних сайтах чи ресурсах пропонується контент про рибалку або полювання, що, очевидно, не дуже цікаво користувачу. Більш того, політика та економіка - це ті теми, де надзвичайно багато фейкової інформації, тому є ціль розробити алгоритм, який вмє розпізнавати фейковий контент у відео або тексту.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

Завданням дипломного проекту було розробити систему аналізу, структурування та фільтрації контенту у різних джерелах інформації за допомогою алгоритмів емоціонального забарвлення тексту, лінгвістичного аналізу, алгоритмів визначення виявлення обличчя/обличь та виявлення співпадінь.

В наші дні контент є надзвичайно популярною, корисною і не менш важливою частиною нашого життя. Люди можуть отримувати свіжі новини з різноманітних джерел, таких як телебачення, газета, журнал, і в даний час найпоширенішим джерелом є Інтернет. Насправді, далеко не всі веб-сайти чи інші ресурси можуть надавати користувачеві правдиву інформацію через різні фактори. На жаль, цей тип джерел має левову частку серед усіх інших. Більш того, контента сьогодні надзвичайно багато, і далеко не весь контент наразі структурован по різним критеріям, таким, як наприклад “Емоціональне забарвлення тексту”, що є надзвичайно важливим критерієм так як контролюючи емоціональне забарвлення тексту, тобто поділяти його на позитивне або негативне забарвлення, можна досягти більшого балансу між користувачем та ресурсом, запропонувати користувачу більш відповідний для нього результат і так далі. Також, є потреба аналізувати та фільтрувати графічний контент як фотографії або відео. Виділяти необхідні об’єкти або людей, та структурувати цей контент по відповідних критеріях та демонструвати користувачу те, що він, ймовірно за все воліє.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

1.1.1 Опис процесу діяльності

В деяких ситуаціях користувач не має змоги контролювати та аналізувати вірно контент, тому було прийнято рішення розробити комплекс алгоритмів, які будуть автоматизованими та структурувати контент замість користувача. Основний опір на перших етапах системи є розробка алгоритму аналізу тексту, так як це найбільш розповсюджений спосіб передачі даних через Інтернет. Необхідно правильно та максимально точно розподіляти текст по емоціональним забарвленням. На рисунку 1.1 зображена схема станів.

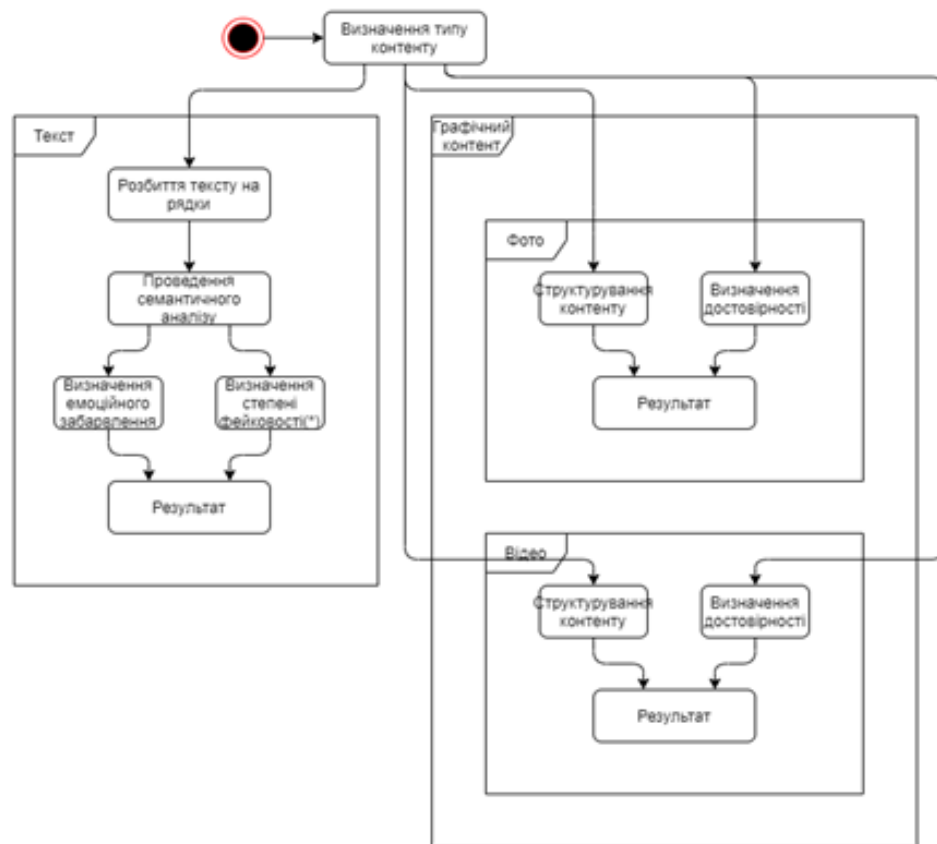


Рисунок 1.1 – Схема структурна станів системи

Змн.	Арк.	№ докум.	Підпис	Дата

На рисунку 1.2 зображена функціональна схема діаграми станів, яка демонструє роботу алгоритму на програмному рівні.

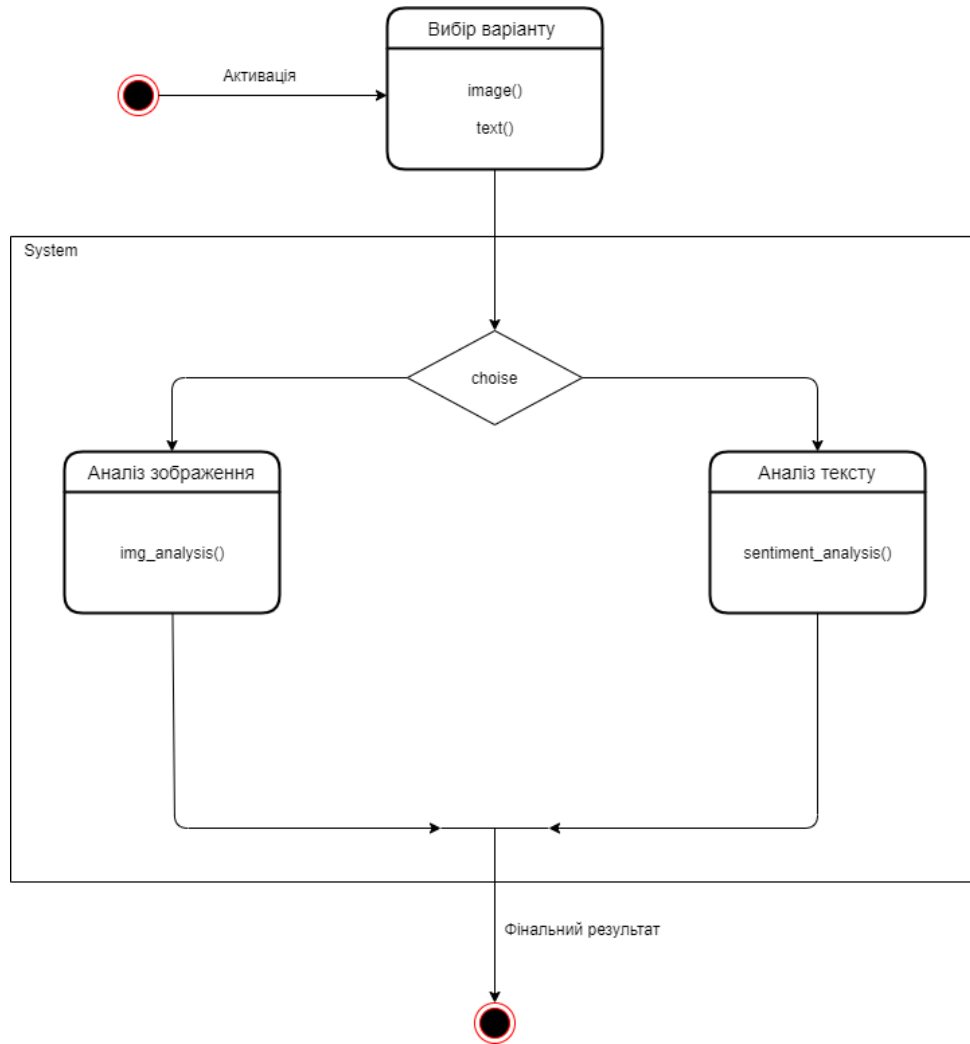


Рисунок 1.2 – Схема структурна станів системи

Змн.	Арк.	№ докум.	Підпис	Дата

Схема активності алгоритму зображена на рисунку 1.3.

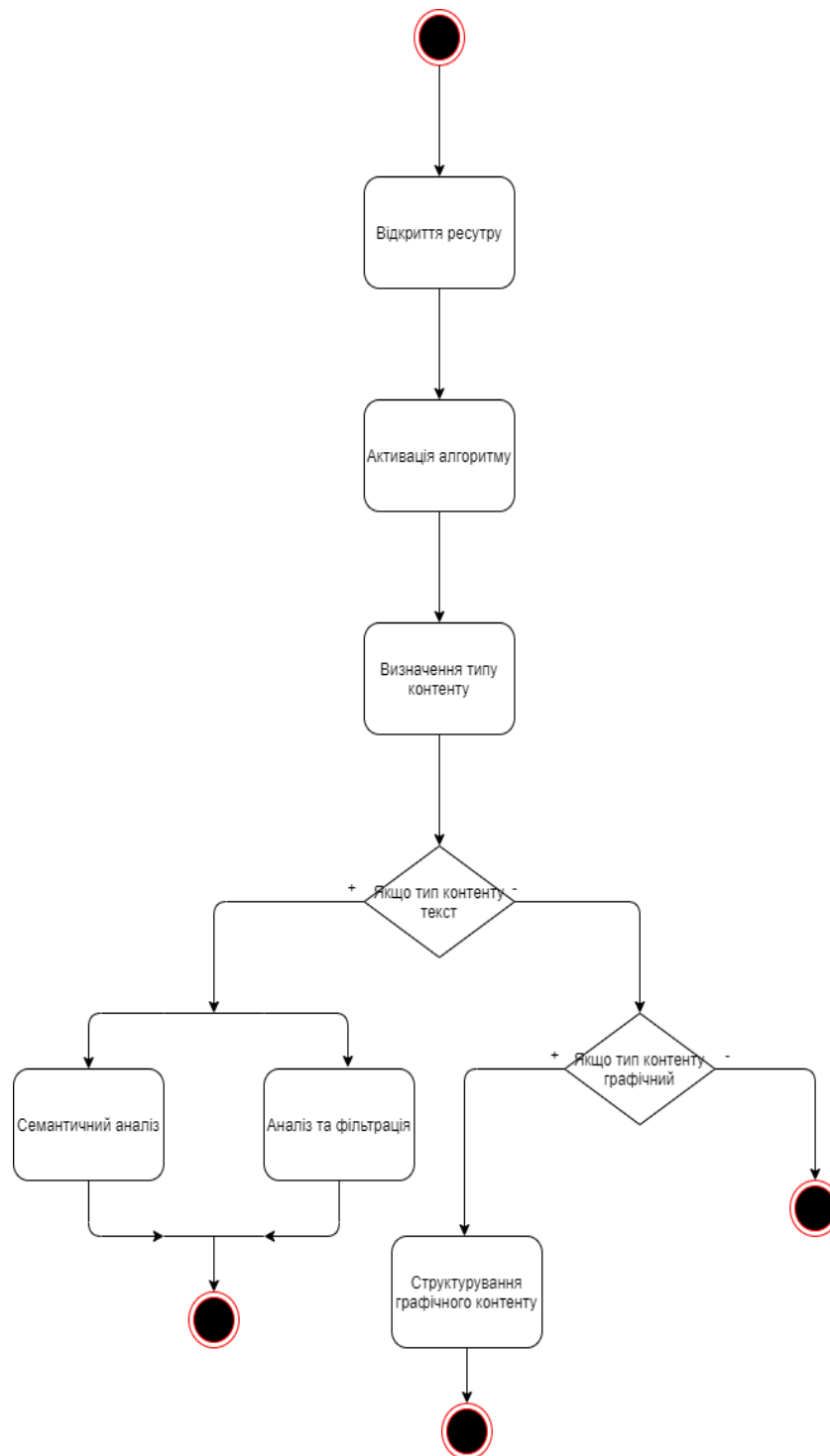


Рисунок 1.3 – Схема структурна діяльності

Змн.	Арк.	№ докум.	Підпис	Дата

1.1.2 Опис функціональної моделі

У системі функціонує лише один актор (дійова особа) - користувач. Користувач може працювати як напряму з системою, так і не напряму, тобто можливе використання офіційного застосунку, та так званого “тіньового” використання, коли користувач взаємодіє з алгоритмом розробленого програмного забезпечення безпосередньо на сайті чи іншому додатку.

Розроблене програмне забезпечення має не монолітну архітектуру, тобто складається з різних модулів, які функціонують в одній загальній системі. Рисунок 1.4 демонструє варіативність використання системи.

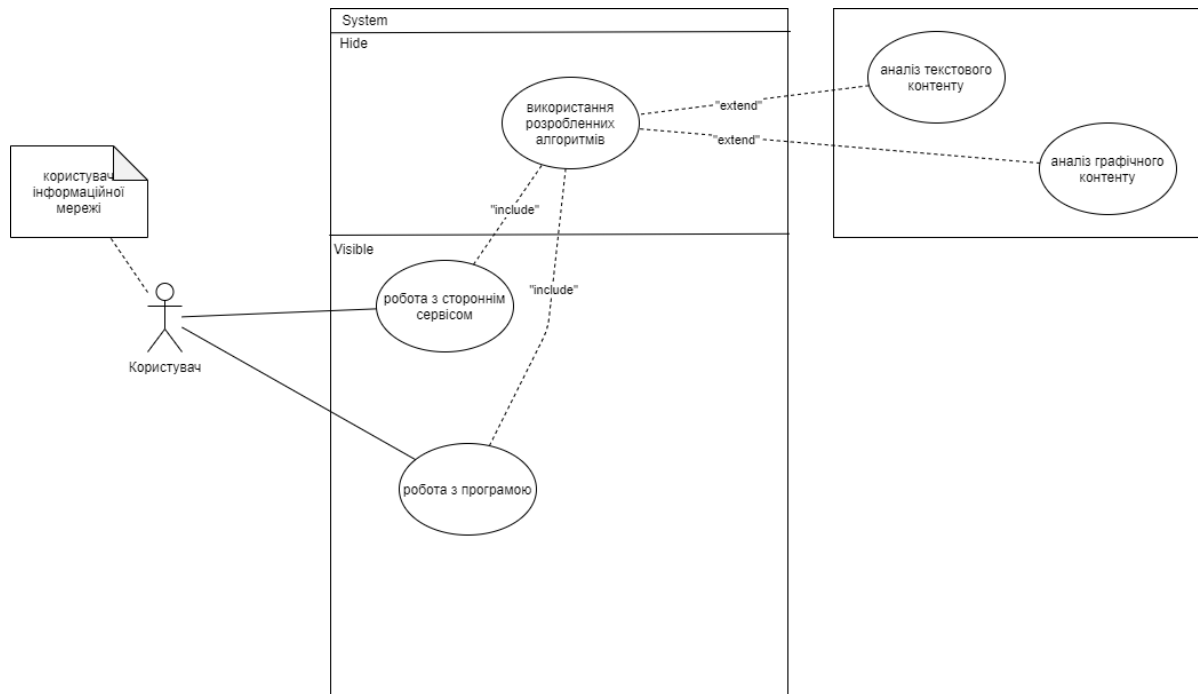


Рисунок 1.4 – Схема структурна варіантів використань

1.2 Огляд наявних аналогів

Зараз на ринку є аналоги від великих та відомих світових корпорацій, таких як Yahoo, Google, YouTube і так далі, але основна відмінність у тому, що ці корпорації використовують схожі алгоритми у внутрішніх реалізаціях їх програмних забезпеченнях, тобто тільки у власному використанні. Схожість алгоритмів у тому, що використовується дуже схожа логіка пошуку та фільтрації контенту, його структурування та аналізу.

Сьогодні, контенту настільки багато, у різних сферах та джерелах, що контролювати та аналізувати його надзвичайно важко. Алгоритми структурування контенту використовуються на таких сервісах, як YouTube, Google, Amazon та Yahoo, однак аналіз потреби користувача там відбувається підбором цікавих йому тем, статей, відео і так далі. Однак, все це відбувається на основі, як правило, одного параметра, це може бути назва статті або відео, заставка або пошук по ключових словах або хештегах. Насправді, інколи такі алгоритми “промахуються” і видають користувачу те, що йому не цікаво або не потрібно. Аналізуючи існуючі рішення розв’язку цієї проблеми, можна виділити кілька основних методів: регресійний аналіз, екстраполяцію та інтерполяцію. На основі цих методів, як правило, і будують відповідні алгоритми. Слід також згадати декілька алгоритмів, які також широко використовуються у фільтрації даних: Фільтр Калмана - це, напевно, найпопулярніший алгоритм фільтрації, який використовується в багатьох областях науки і техніки. Завдяки своїй простоті й ефективності його можна зустріти в GPS-приймачах, обробниках показань датчиків, при реалізації систем управління і т.д.

Основною причиною розробки нових алгоритмів, високорівневих бібліотек та готових рішень є потреба пропонувати клієнтам та користувачам тільки той контент, який йому цікавий або необхідний. Саме для цих потреб великі компанії, такі як Google чи Facebook прийняли свої власні рішення, такі як TensorFlow, Pandas, PyTorch, TextBlob і так далі. Як результат, ці алгоритми та бібліотеки надають можливість розробникам програмного забезпечення впроваджувати новий функціонал у свої програми чи веб-застосунки, а саме функціонал, який так чи інакше пов'язаний з Artificial Intelligence, Machine Learning та Deep Learning. У програмному забезпеченні, яке проектується та створюється мною, використані методи та підходи саме глибинного навчання, яке має свої вагомі плюси, такі як можливість навчати модель. Без всяких сумнівів, цей підхід є більш вигідним у сучасному житті, так як алгоритм має змогу навчатися і покращувати сам себе безпосередньо при його роботі та виконанні.

Розглянемо декілька основних моментів, для чого використовується аналіз, структурування та фільтрація контенту у різних мережах та джерелах інформації:

- існує велика потреба у фільтрації контенту для користувача в тому, щоб пропонувати йому саме те, чим він цікавиться та що йому імовірно потрібно;
- задля покращення зовнішнього та внутрішнього пошуку, можна використовувати спеціальні фільтри, які будуть працювати за допомогою вже написаного алгоритму аналізу, фільтрації та структурування;
- автоматизація внутрішнього процесу фільтрації контенту без “зайвих рухів”.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

1.3 Постановка задачі

1.3.1 Призначення розробки

Система призначена для аналізу, структурування та фільтрації контенту у інформаційних мережах.

Призначенням розробки комплексу реалізацій алгоритмів пошуку, структурування, аналізу та фільтрації є поліпшення та забезпечення безпекою користувача, який користується мережею Інтернет або іншими джерелами.

Система розробляється для додавання відповідного функціоналу у вже готові програмні рішення, такі як веб-сайти або десктопні застосунки.

Найголовнішою ціллю розробки є полегшення сервісу та програмам підбирати контент користувачу та, відповідно, користувачу шукати потрібну йому інформацію.

1.3.2 Цілі та задачі розробки

Цілями створення є:

- аналіз тексту за емоційним забарвленням;
- аналіз графічного контенту;
- підтримка існуючих програмних реалізацій;
- структурування контенту по відповідним тек;
- власне користування.

Безумовно головною ціллю розробки є можливість вибірки відповідного контенту який є потенційно корисним чи цікавим користувачу.

Іншими словами, метою розробки програмного забезпечення є фільтрація контенту по деяким критеріям, таким як пошук інформації, яка є цікавою для користувача, аналізу контенту, мається на увазі глибинний аналіз того чи іншого типу контенту за деякими параметрами, такими як семантичний аналіз тексту або кількість людей на фотографії та структурування контенту за критеріями, які можна встановити самому.

Наприклад створення відповідної директорії куди буде зберігатися інформація за відповідними критеріями, наприклад, тільки позитивний текст або графічний контент лише з однією відповідною персоною. Схожу логіку використовують Yahoo, Google, YouTube та інші світові корпорації.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Висновок до розділу

У цьому розділі була описана мета та ціль розробки проекту, основні аспекти реалізації та проектування. Була визначена предметна область, завдяки якій є змога проаналізувати програмне середовище та програмне забезпечення. Були розроблені діаграми станів (функціональна та логічна), активностей та Use case. Також були описані явні аналоги, які вже існують на ринку.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

На вхід у програмну реалізацію поступають дані у різних форматах, в залежності від того, що необхідно проаналізувати, профільтрувати або структурувати.

Якщо необхідно провести аналіз тексту, то на вхід поступають дані у вигляді рядка, якщо це якийсь сайт в мережі Інтернет чи будь-яке інше джерело інформації, який був спарсенний та після чого була отримана відповідна інформація/дані, яка була записана у внутрішні змінні у програмної реалізації в якості типу даних “рядок”, є можливість використовувати звичайні файли формату .txt, та є можливість використання файлів формату .csv з відповідним роздільником (роздільник необхідно зазначити у самій програмі. Це може бути кома, крапка з комою, слеш або вертикальна риска. Краще за все використовувати найменш очевидний роздільник, такий як вертикальна риска, так як вона не використовується у текстах).

Якщо проводиться аналіз тексту, то дані подаються у вигляді рядків. Це можуть бути однорядкові або багаторядкові рядки, відповідно до того, який текст подається у програму.

Якщо проводиться аналіз графічного контенту, тобто аналіз фотографії або відео, то використовуються різні формати: для картинок це .png, .jpg або .jpeg, а для відео використовуються формати .webm або .mp4. Подаються ці дані після парсингу, явною передачею або з відповідної директорії де можуть зберігатися ці файли.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Якщо треба провести аналіз графічного контенту, то дані подаються у програмну реалізацію у відповідних форматах, які були описані вище.

2.2 Вихідні дані

Вихідні дані з алгоритму можна отримати у декілька форматів: .txt, .csv, у внутрішню змінну у програмній реалізації та у базу даних MongoDB.

Перелік вхідних сигналів:

Таблиця 1.1 – Перелік вхідних сигналів

Призначення	Найменування	Одиниці виміру	Діапазони зміни	Спосіб предст. інформації
Загальні вхідні дані про систему фільтрації, аналізу та структурування контенту.	polarity_range	int	[0, 1]	Графічний інтерфейс, користувачький застосунок, консоль.
	subjectivity_range	int	[0, 1]	
	positive_correct	int	[0, 100]	
	positive_count	int	[0, max int]	
	only_positive_array	array	[0, max int]	
	KNOWN_FACES_DIR	string	[0, max int]	

Продовження таблиці 1.1

Призначення	Найменування	Одиниці виміру	Діапазони зміни	Спосіб предст. інформації
	UNKNOWN_FACES_DIR	string	[0, max int]	
	TOLERANCE	int	[0, 1]	
	FRAME_THICKNESS	int	[0, max int]	
	MODEL	string	[0, max int]	
	known_faces	array	[0, max int]	
	known_names	array	[0, max int]	
	match	array	[0, max int]	
	result	bool	False	

Перелік вихідних документів:

Таблиця 1.2 – Перелік вихідних документів

Призначення	Найменування	Одиниці виміру	Діапазони зміни	Спосіб предст. інформації
Загальні вихідні дані про систему фільтрації, аналізу та структурування контенту.	result	bool	[False, True]	Графічний інтерфейс, користувацький застосунок, консоль, файли.
	match	array	[0, max int]	
	only_positive_array	array	[0, max int]	
	positive_correct	int	[0, 100]	
	positive_count	int	[0, max int]	

2.3 Структура масивів інформації

Так як форматів вхідних і вихідних даних декілька, відповідно і структур масивів інформації буде декілька. Відповідно до того, над чим проводиться аналіз отримується структура масивів інформації.

Вхідні дані:

Аналіз тексту:

1. У формат .txt подаються дані у вигляді рядка, або цілого тексту, де є роздільник у вигляді крапки;
2. У формат .csv подаються дані у вигляді рядка або рядків де є роздільник у вигляді заздалегідь записаного у програму роздільника, це може бути будь-який символ, наприклад “|”;
3. Також, вхідні дані можуть подаватися у вигляді спарсиного сайту або іншого джерела інформації у мережі Інтернет у вигляді великого рядка у програмному забезпеченні.

Аналіз графічного контенту:

1. У форматах .png, .jpg або .jpeg подаються картинки, які можна отримати з відповідних директоріях або після парсингу сайту у мережі Інтернет.
2. У форматах .webm або .mp4 подаються відео, які можна отримати з відповідних директоріях або після парсингу сайту у мережі Інтернет.

Вихідні дані:

Аналіз тексту:

1. У формат .txt записуються дані у вигляді рядка, або цілого тексту, де є роздільник у вигляді крапки;
2. У формат .csv записуються дані у вигляді рядка або рядків де є роздільник у вигляді заздалегідь записаного у програму роздільника, це може бути будь-який символ, наприклад “|”;
3. Вихідні дані можуть записуватися у базу даних MongoDB.

Аналіз графічного контенту:

1. Результат з аналізу картинок у форматах .png, .jpg або .jpeg може зберігатися у файлах форматах .txt або .csv, або результат може записуватися у базу даних MongoDB.
2. Результат з аналізу відео у форматах .webm або .mp4 може зберігатися у файлах форматах .txt або .csv, або результат може записуватися у базу даних MongoDB.

Висновок до розділу

У цьому розділі були розглянуті формати даних, які подаються на вхід, формати даних, які отримуються на виході, структури масивів даних які подаються на вхід або отримуються на виході.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Змістовна постановка задачі для програмної реалізації алгоритму аналізу, структурування та фільтрації інформації у мережі Інтернет та у інших джерелах інформації.

Розробити математичну модель, яка дозволяє мінімізувати час пошуку необхідного контенту для користувача у мережі Інтернет.

Модель повинна дозволяти враховувати загальний час користувача на пошук контенту.

Вихідні дані:

- Загальний час затрачений для пошуку контенту.

3.2 Математична постановка задачі

Використання алгоритму можна поділити на дві частини: на, так зване “сліпе” використання та на явне використання програмного забезпечення.

Під “сліпим” використанням слід розуміти неявне використання програмного забезпечення, тобто алгоритм використовується у внутрішніх системах додатку. Користувач не взаємодіє з алгоритмом напряму. Під час сліпого використання, користувач взаємодіє безпосередньо з застосунком, який має функціонал розробленого алгоритму.

Явне використання застосунку - це використання алгоритму у офіційному графічному застосунку. Це може бути десктопний застосунок або веб-сервіс. Користувач може самостійно вирішувати, що йому треба фільтрувати, аналізувати чи структурувати.

3.3 Обґрунтування методу розв'язання

Для розв'язання поставленої задачі, а саме реалізації алгоритму фільтрації, аналізу та структурування контенту була вибрана ідея розбиття монолітної структури на дві рівні частини, потім дві частини розбиваються рекурсивно навпіл, і так далі. Так зване рекурсивне центральне розбиття.

Якщо взяти, наприклад текст, то маємо наступне: на вхід подається монолітний текст, після парсингу веб-сайту, зчитування файлу або за допомогою графічного застосунку. Після отримання тексту, є декілька шляхів розбиття тексту на секції та слова.

Логіка закладається у тому, що необхідно розбити текст на секції, а потім на слова для того, щоб проаналізувати кожне слово на емоційне забарвлення та послідуєчий запис їх у словник. Після чого можна робити та проводити відповідний аналіз тексту на емоційне забарвлення.

Сам алгоритм виглядає якимось так: створюється функція, яка приймає відповідний рядок та дільник, за замовчуванням довжина рядка ділена на 2. У функції створюється масив, який приймає у себе поділений рядок. Після чого запускається цикл, який працює до кількості елементів (під рядків) у масиві, який ділить елемент на 2 рівні частини та замінює і-ий елемент на ці два нових елементи. Після всіх ітерації новий масив заміняє старий. Так відбувається до тих пір, поки не буде 1 елемент - це 1 слово.

Інша реалізація подібного алгоритму. Створюється така сама функція з такими ж параметрами, але масив за замовчуванням пустий. Запускається цикл який шукає перший пробіл, тобто " ", та якщо знайшов, додає все, що було до пробіла та після пробіла (якщо слово знаходиться десь у центрі рядка), у масив.

Також, мова програмування Python дозволяє розбивати речення на слова за допомогою спеціального методу `.split(' ')` з відповідними роздільниками, в даному прикладі цей роздільник - пробіл.

Якщо ми розглядаємо графічний контент, а саме картинки та фотографії, то у такому випадку логіка алгоритму трішки інакша: після отримання фотографії або будь-якої картинки відбувається так званий поділ картинки чи фотографії на рівні зони. Тобто є фотографія або картинка, вона дробиться на 4 рівні зони, ці 4 рівні зони дробляться ще на 4 рівні зони всередині відповідної зони, тобто маємо вже не 1 велику зону, а 16 маленьких. Це продовжується до тих пір, поки є можливість (поки не досягли 1 пікселя). Тоді відбувається попарне порівняння пікселів і відбувається їх подальше з'єднання. Де колір співпадає з тоном, там ймовірніше за все один об'єкт, де тон відрізняється, значить, ймовірно, там є границя між об'єктом та об'єктом або між об'єктом та фоном.

Приблизно таким же чином працює алгоритм для відео, як і картинки або фотографії. Тільки відео можна перевіряти покадрово, або з інтервалом, який визначається раніше, тобто перевіряти картинку раз у 5 кадрів, наприклад.

3.4 Опис методів розв'язання

Так, як було описано у пункти 3.3, є декілька методів розв'язання поставленої задачі. Інша реалізація подібного алгоритму:

					ДП 6206.00.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

1. Рекурсивне ділення. Логіка заключається у необхідності розбити текст на підтексти, а потім на слова для того, щоб проаналізувати кожне слово на емоційне забарвлення та послідоючий запис їх у словник. Після всього робиться аналіз тексту на емоційне забарвлення. Алгоритм: створюється функція, яка приймає відповідний рядок та дільник, за замовчуванням довжина рядка ділена на 2. У функції створюється масив, який приймає у себе поділений рядок, після чого запускається цикл, який працює до кількості елементів (під рядків) у масиві, який ділить елемент на дві рівні частини та заміняє і-ий елемент на ці два нових елементи, тобто наприклад ['ab', 'cd'] => ['a', 'b', 'cd'] - після першої ітерації. Після всіх ітерації новий масив заміняє старий. Так відбувається до тих пір, поки не буде 1 елемент - це 1 слово.

2. Послідовне вилучення. Створюється та ж сама функція з такими ж параметрами, але масив за замовчуванням пустий. Запускається цикл який шукає перший пробіл, тобто " ", та якщо знайшов, додає все, що було до пробіла та після пробіла (якщо слово знаходиться десь у центрі рядка), у масив.

На рисунку 3.1 зображено код реалізації на мові програмування Python:

```
value = 'hello, World!'
word = ''
result = []

for index in value:
    if (index == ' ' or index == ',' or index == '!') and word != '':
        result.append(word)
        word = ''
    else:
        word = word + index
```

Рисунок 3.1 – Приклад коду на мові програмування Python

1. Розбиття речення на слова за допомоги методу `.split(' ')` з відповідними роздільниками.

2. Рекурсивне ділення графічного контенту. Після отримання фотографії або будь-якої картинки відбувається поділ картинки чи фотографії на рівні зони. Тобто є фотографія або картинка, вона дробиться на чотири рівні зони, ці чотири рівні зони дробляться ще на чотири рівні зони всередині відповідної зони, тобто маємо вже не одну велику зону, тобто початкову картинку або фотографію, а шістнадцять маленьких, які в сумі дають одну велику зону (початкову картинку або фотографію). Це продовжується до тих пір, поки є можливість (поки не досягнули одного пікселя).

Тоді відбувається попарне порівняння пікселів і відбувається їх подальше з'єднання. Де колір співпадає з тоном, там ймовірніше за все один об'єкт, де тон відрізняється, значний, ймовірно, там є границя між об'єктом та об'єктом або між об'єктом та фоном. Таким самим чином працює алгоритм для відео, як і картинки або фотографії. Є можливість перевіряти відео покадрово, або з інтервалом, який визначається раніше, тобто перевіряти картинку раз у п'ять кадрів, наприклад.

Висновок до розділу

У цьому розділі була сформульована математична та змістовна постановка задачі. Задача, для фільтрації аналізу та структурування контенту є досить складною, для описання та описання методів розв'язку цієї задачі. В цьому розділі вдалося в деталях описати методи вирішення цієї задачі та були приведені приклади.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для розробки програмного забезпечення була вибрана мова програмування Python 3.8 як основна, так як вона досить проста для реалізації важких алгоритмів через багату внутрішню структуру, через велику кількість вбудованих методів та класів. Мова програмування Python 3.8 ідеально підходить для реалізації аналізу та маніпуляції з текстом або будь-яким графічним матеріалом. Мова програмування Python 3.8 також чудово підходить для реалізації алгоритмів будь-якої складності, у тому числі і алгоритмів глибинного навчання, штучного інтелекту або машинного навчання.

Також, мова програмування Python 3.8 має велику кількість вже готових реалізацій та бібліотек для реалізації алгоритмів глибинного навчання, штучного інтелекту та машинного навчання від великих корпорацій, таких як Google або Facebook, наприклад, TensorFlow, Pandas, Keras та PyTorch.

Також, для роботи не тільки з CPU а й з GPU було прийнято рішення використовувати мову програмування C++ так як ця мова програмування досить низькорівнева. Також, в доповненні до C++ як основний інструмент роботи з зображеннями та відео була використана бібліотека dlib яка написана на мові програмування C++.

Мова програмування C++ використовується не тільки для роботи з GPU, а й для роботи з зображеннями та відео контентом. Була використана реалізація мови програмування C++17.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Також як допоміжна мова для реалізації внутрішньої логіки була використана мова програмування JavaScript, так як вона є досить універсальна та підходить для широкого ряду задач.

4.2 Вимоги до технічного забезпечення

Даний програмний продукт представлений у вигляді графічної середи або у вигляді API, якщо спосіб використання - це підключення до готового програмного продукту нашу систему.

Для коректної роботи алгоритму необхідний Python 3, C++17, dlib. Для графічного застосунку необхідний PyQt5. API використовує Flask для серверної логіки.

Усі користувачі системи повинні дотримуватися правил експлуатації електронної обчислювальної техніки.

Система повинна адекватно реагувати на помилки застосування та видавати відповідні повідомлення користувачеві, а також записувати їх у log-файл.

4.2.1 Загальні вимоги

Комп'ютер повинен працювати на операційній системі Windows 10 або на Linux, так як є реалізація у вигляді dlib, попередньо необхідно налаштувати систему на коректну роботу. Комп'ютер повинен мати як мінімум 4 Gb ОЗУ, процесор з частотою рівно або більше 2.4 GHz та від 2 Gb вільного місця на комп'ютері

4.3 Архітектура програмного забезпечення.

Архітектура програмного забезпечення зводиться до того, що є 4 основних блока: персональний комп'ютер користувача з графічним застосунком (це може бути як веб-сайт так і десктопний застосунок), Front-end з веб-контейнером для спілкування з Back-end, та дві бази даних з вхідними та вихідними даними. На рисунку 4.1 зображена схема компонентів даного програмного забезпечення.

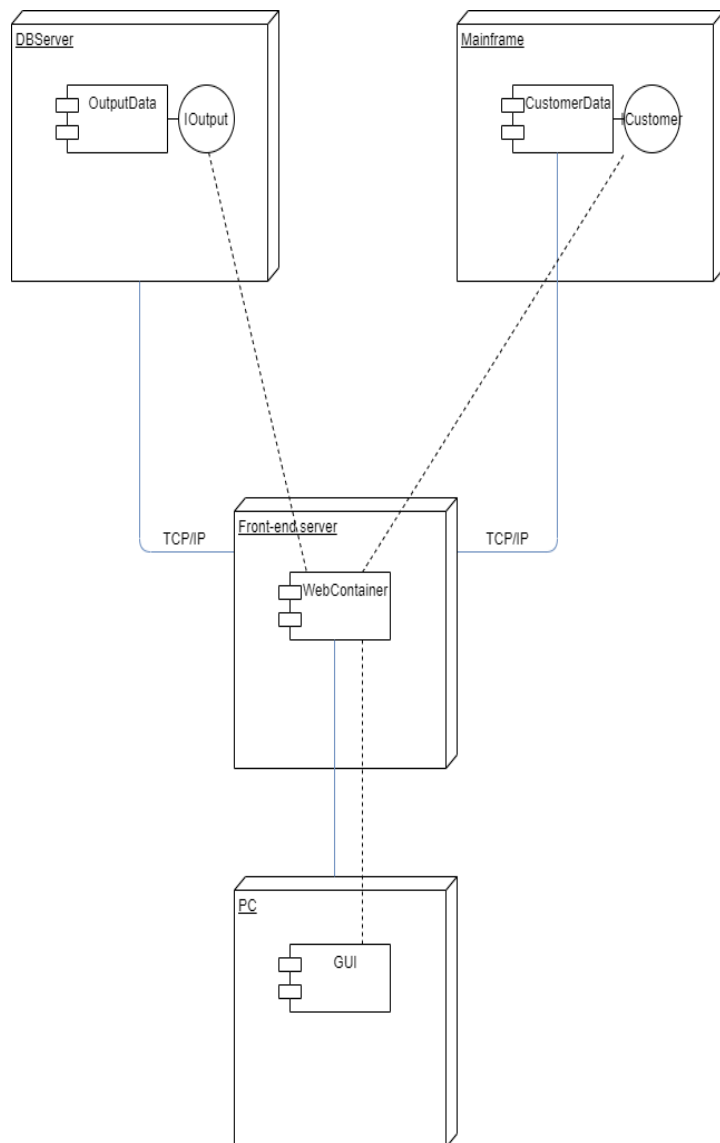


Рисунок 4.1 – Схема структурна компонентів програмного забезпечення

4.3.1 Діаграма класів

На рисунку 4.2 зображена діаграма класів розробленого програмного забезпечення.

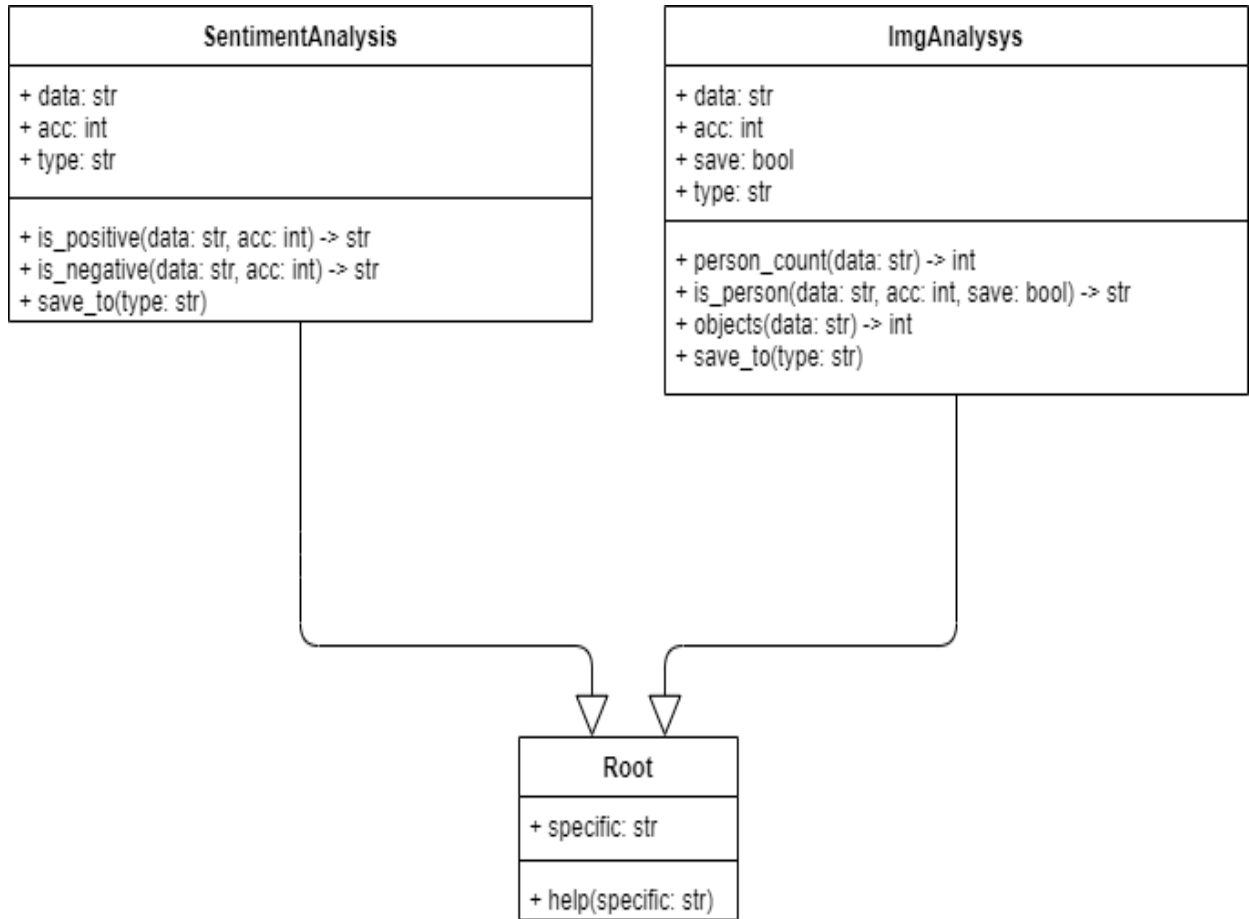


Рисунок 4.2 – Схема структурна класів програмного забезпечення

4.3.2 Діаграма послідовності

На рисунку 4.3 зображена схема послідовності розробленого програмного продукту.

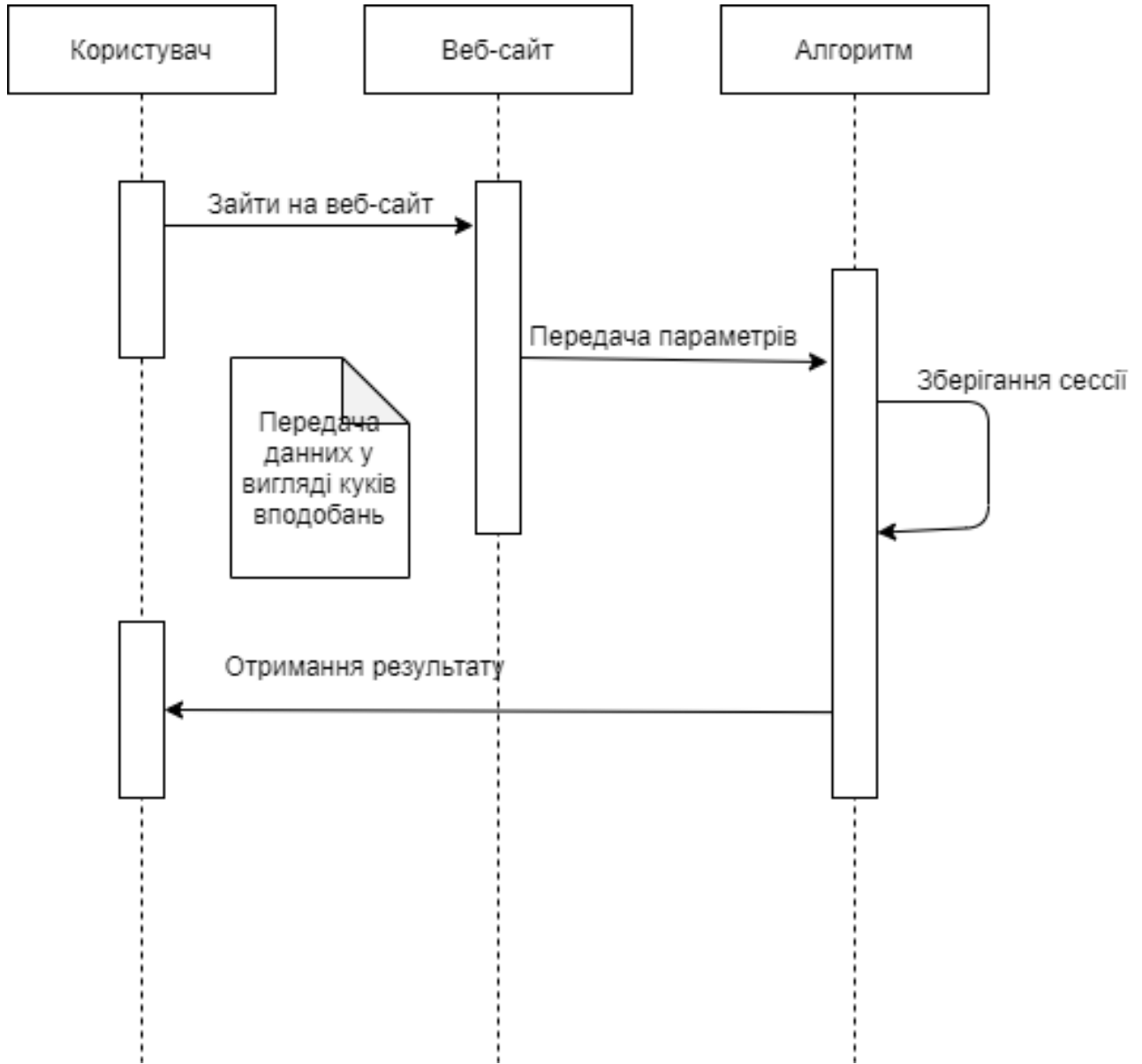


Рисунок 4.3 – Діаграма послідовності

4.3.3 Діаграма компонентів

На рисунку 4.1 зображена схема компонентів даного програмного забезпечення.

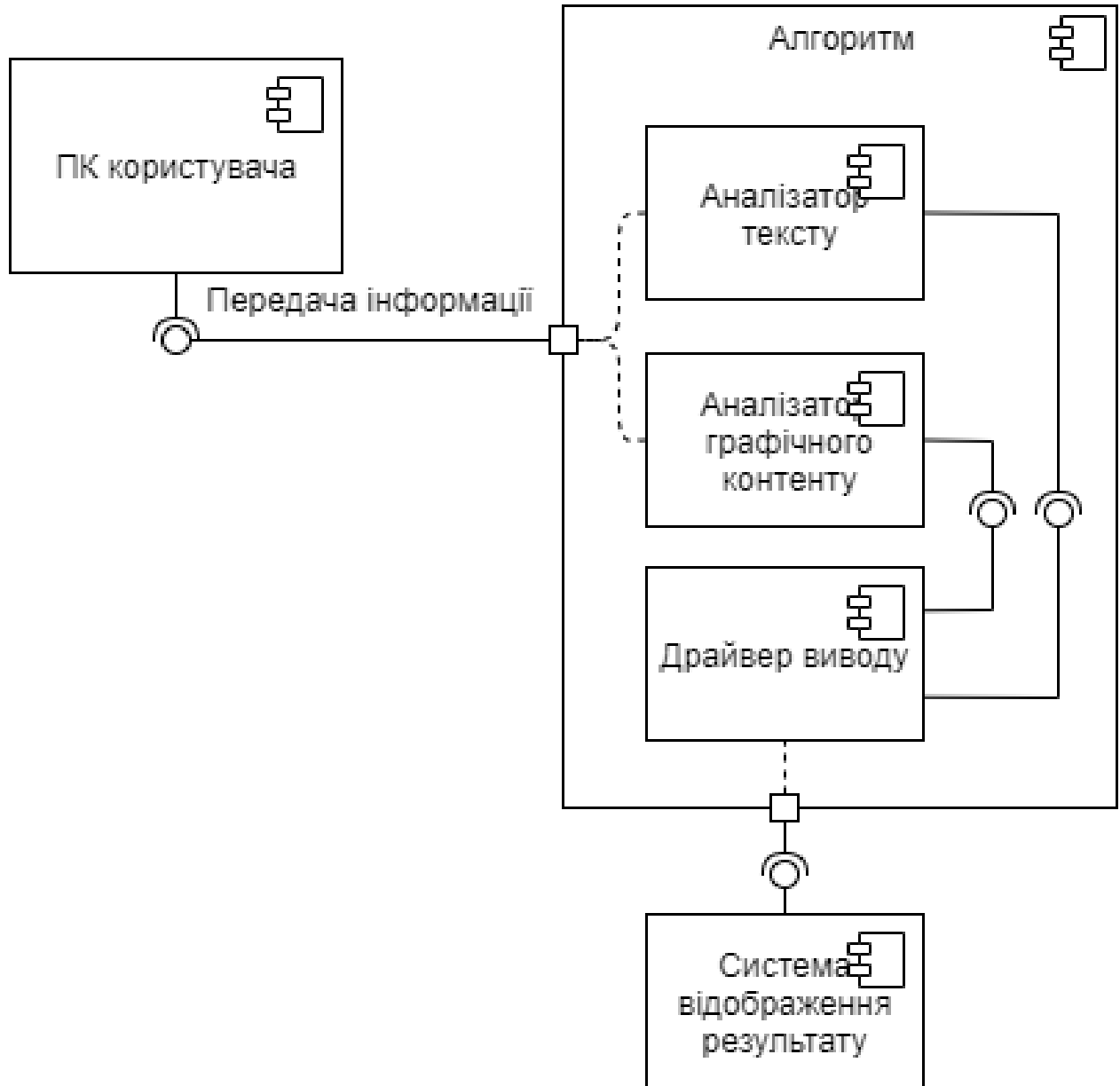


Рисунок 4.4 – Схема структурна компонентів програмного забезпечення

4.3.4 Специфікація функцій

Для роботи з аналізом зображення були використані модулі dlib (написаний на C++, реалізація у Python) та face_recognition.

Для роботи з зображеннями використано модуль cv2.

Для роботи з аналізом даних були використані модулі textblob, vaderSentiment.vaderSentiment та speech_recognition для визначення тону голосу.

Для розробки програмного інтерфейсу була обрана бібліотека PyQt5.

Для розробки API був обран фреймворк Flask.

Програмна реалізація складається з таких основних функцій: fc2 - 'Face detection', sa - 'Sentiment analysis', gui - 'Graphical User Interface' та server.

Функція 'fc2' написана за допомогою низькорівневих модулів face_recognition, dlib та cv2. Реалізація цієї функції дає можливість визначати людей на фотографії, кількість людей на фотографії, якщо створений відповідний dataset з фотографіями відомої людини, то функція також дозволяє перевіряти, чи є на фотографії ця людина, чи ні. Якщо ж людина є, вона поповнює dataset з цією людиною. *У майбутньому планується додати довільний функціонал до функції та визначення відповідних об'єктів на фотографії*.

Функція 'sa' написана за допомогою модулів textblob, vaderSentiment.vaderSentiment та speech_recognition та призначений для визначення тональності тексту, поділу його на позитивний або негативний тон та для подальшого структурування результату у відповідні директорії: позитивний текст та негативний текст.

Висновок до розділу

У цьому розділі були описані засоби розробки, які використовувались у створенні системи для аналізу, структурування та фільтрації контенту. Були наведені плюси і мінуси кожного вибору, переваги для використання того чи іншого рішення. Були описані вимоги до технічного забезпечення, які необхідні для використання комплексу поставлених задач. У цьому розділі описана архітектура додатку, були наведені відповідні діаграми. Були представлені діаграми та структурні схеми класів, послідовностей та компонентів, які описують архітектуру системи та її внутрішніх частин. Для коректної роботи середовища необхідно, щоб всі пункти були виконані, які були описані у розділі.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Для роботи з зображеннями використано модуль cv2.

Для роботи з аналізом даних були використані модулі textblob, vaderSentiment.vaderSentiment та speech_recognition для визначення тону голосу.

Для розробки програмного інтерфейсу була обрана бібліотека PyQt5.

Для розробки API був обран фреймворк Flask.

Програмна реалізація складається з таких основних функцій: fc2 - 'Face detection', sa - 'Sentiment analysis', gui - 'Graphical User Interface' та server.

Функція 'fc2' написана за допомогою низькорівневих модулів face_recognition, dlib та cv2. Реалізація цієї функції дає можливість визначати людей на фотографії, кількість людей на фотографії, якщо створений відповідний dataset з фотографіями відомої людини, то функція також дозволяє перевіряти, чи є на фотографії ця людина, чи ні. Якщо ж людина є, вона поповнює dataset з цією людиною. *У майбутньому планується додати довільний функціонал до функції та визначення відповідних об'єктів на фотографії*.

Функція 'sa' написана за допомогою модулів textblob, vaderSentiment.vaderSentiment та speech_recognition та призначений для визначення тональності тексту, поділу його на позитивний або негативний тон та для подальшого структурування результату у відповідні директорії: позитивний текст та негативний текст.

Змн.	Арк.	№ докум.	Підпис	Дата

Функція 'gui' написана за допомогою модуля PyQt5 та призначена для графічного відображення програмного продукту.

Модуль 'server' написана за допомогою фреймворка Flask для створення API.

На рисунках 5.1-5.4 зображені рисунки роботи програмного забезпечення.

This is a really great movie - is positive

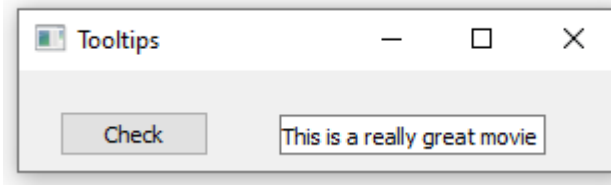


Рисунок 5.1 – Демонстрація екранної форми

Positive accuracy = 70.78810759792354% via 4238 samples

Рисунок 5.2 – Демонстрація роботи програмного забезпечення (відсоток позитивних текстів)

there is 1 people in the image

Рисунок 5.3 – Демонстрація роботи програмного забезпечення (кількість людей на фото)

This is known person

Рисунок 5.4 – Демонстрація роботи програмного забезпечення (результат роботи розпізнавання людей)

5.2 Випробування програмного продукту

Був проведений комплекс відповідних тестів для тестування розробленого програмного забезпечення для переконання у правильності розробки.

Всього було проведено 3 типи тестувань:

1. Ручне тестування. Був проведений комплекс тестів, які були набрані вручну, які показали поверхневий результат, після якого можна переходити на інший тип тестування. Логіка цього типу тестування закладається у тому, що всі вхідні параметри задаються прямо у виконанні програми, у консолі або у графічному застосунку.

2. UNIT тести. Заздалегідь написані тести за допомогою модулів unittest та pdb. Сенс цього типу тестування у тому, що вхідні параметри написані у вигляді функцій та параметрів, які написані у цих функціях, написані заздалегідь.

3. Авто Тести. Сенс авто тестів полягає у тому, що використовується спеціальна для цього бібліотека selenium, яка дозволяє проводити автоматичне тестування, яке можна налаштувати самостійно та виставити саме ті параметри та вхідні дані, які необхідні для подальшого тестування.

Були розроблені загальні алгоритми тестування для розробленої системи фільтрації, структурування та аналізу контенту.

Порядок виконання:

1. Ручні тести - це комплекс тестів, які набираються вручну, які показують поверхневий результат, після якого можна переходити на інший тип тестування. Сенс цього типу тестування закладається у тому, що всі вхідні параметри задаються прямо у виконанні програми, у консолі або у графічному застосунку.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

2. UNIT тести - заздалегідь написані тести. Сенс цього типу тестування у тому, що вхідні параметри написані у вигляді функцій та параметрів, які написані у цих функціях, написані заздалегідь.

3. Авто Тести. Сенс авто тестів полягає у тому, що використовується спеціальна для цього бібліотека selenium, яка дозволяє проводити автоматичне тестування, яке можна налаштувати самостійно та виставити саме ті параметри та вхідні дані, які необхідні для подальшого тестування.

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач системи фільтрації, аналізу та структурування контенту вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Таблиця 5.1 - Проведення випробувань

Номер випробування	Ситуація	Ефект	Фактичний ефект
1	Спроба завантажити графічний контент у аналізатор тексту	При завантаженні графічного контенту у аналізатор тексту, повинне бути виключення з відповідним повідомленням	Співпадає з очікуванням
2	Спроба завантажити текст у графічний контент	При завантаженні тексту у аналізатор графічного контенту, повинне бути виключення з відповідним повідомленням	Співпадає з очікуванням

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 5.1

Номер випробування	Ситуація	Ефект	Фактичний ефект
3	Відключення куків	При відключенні куків, система повинна зберігати всі дані всередині системи	Співпадає з очікуванням
4	Використання неавторизованого користувача у інформаційному джерелі	Система повинна працювати коректно незважаючи на те, авторизован користувач або ні	Співпадає з очікуванням
5	Відключення зберігання результату у системі	Система повинна зберігати дані у ОЗУ, при відключенні внутрішнього зберігання	Співпадає з очікуванням

Висновок до розділу

У розділі було розроблене керівництво користувача, що включає у себе опис дій, які користувач може робити у системі. Були проведені тести для виявлення помилок або небажаної поведінки системи. Всі тести були пройдені успішно, ніяких помилок та небажаних поведінок системи виявлено не було.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

ЗАГАЛЬНІ ВИСНОВКИ

За період розробки дипломного проекту, було вирішено багато проблем та задач. Також була розроблена програмна реалізація системи аналізу, структурування та фільтрації контенту у мережі Інтернет та в інших джерелах інформації. Програмна реалізація була написана на мовах програмування Python, JavaScript та C++.

Була описана предметна область та предметне середовище, були наведені діаграми станів, активності та Use-case. Був у деталях розписаний процес діяльності (діаграми станів та активності), була описана функціональна модель, основні актори та ключові варіанти використання програмної реалізації системи структурування, аналізу та фільтрації контенту. У прикладах були подані наявні аналоги розробленої системи. Були приведені приклади реалізацій Google, YouTube та Yahoo. Була описана постановка задачі та призначення розробки. Були сформульовані цілі та задачі розробки системи.

Були описані вхідні та вихідні дані, їх розширення та конфігурації. Були описані дані, які подаються у цих вхідних даних, були описані вихідні дані, які отримуються користувачем у вихідних даних. Була описана структура масивів даних.

Була описана математична постановка задачі та змістовна постановка задачі. Також були наведені описи та характеристики методів, які використовувалися при розробці програмної реалізації системи аналізу, фільтрації та структурування контенту.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

В деталях були описані засоби розробки, програмне та технічне забезпечення, загальні вимоги та архітектура програмного забезпечення. Була наведена діаграма архітектури програмної реалізації. Також були наведені діаграма класів, діаграма послідовності та діаграма компонентів. Були описані основні специфікації функцій.

Були наведені загальні технічні поняття, а саме керівництво користувача, тестування програмної реалізації, мета цих тестів, загальні положення та результати випробувань.

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Додаток А

Тексти програмного коду

Система для аналізу, фільтрації та структурування контенту

(Найменування програми (документа))

DVD-R

(Вид носія даних)

30 арк, 3 070 Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

Змн.	Арк.	№ докум.	Підпис	Дата

Далі наведено основний код файлів у системі. Весь код наведено на носії.

Код модуля розпізнавання тексту:

"""

Author of the script: Patlay Nikia / Group is IS-z61 / fourth year student.

"""

```
# <ModulesSection modules_count="2">
```

```
from textblob import TextBlob
```

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

"""

The list of high-level instances:

1. textblob - if env is using then pipenv install textblob

otherwise pip3 install textblob

2. vaderSentiment - if env is using then pipenv install vaderSentiment

otherwise pip3 install vaderSentiment

"""

```
# </ModulesSection>
```

```
# +---+ Main script +---+ #
```

```
def is_positive(polarity_range=0, subjectivity_range=1) -> str:
```

```
    if polarity_range >= 1:
```

```
        raise ValueError(f'Polarity range should be between 0 and 1 not inclusive.  
Obtained - {polarity_range}')
```

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

```
positive_correct = 0
```

```
positive_count = 0
```

```
only_positive_array = []
```

```
with open('sentiment_analysis/positive.txt', encoding='utf-8') as file:
```

```
    for line in file.read().split('\n'):
```

```
        analysis = TextBlob(line)
```

```
        if analysis.sentiment.subjectivity < subjectivity_range:
```

```
            if analysis.sentiment.polarity > polariry_range:
```

```
                only_positive_array.append(line)
```

```
                positive_correct += 1
```

```
                positive_count += 1
```

```
with open('sentiment_analysis/only_positive.txt', 'w', encoding='utf-8') as file:
```

```
    for line in only_positive_array:
```

```
        file.write('%s\n' % line)
```

```
    if positive_correct == 0:
```

```
        print(f'There are no possitive messages')
```

```
        with open('sentiment_analysis/local_storage.txt', 'w',  
encoding='utf-8') as file:
```

```
            file.write('There are no possitive messages') #
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

elif positive_correct == 1:
    print(f'{line} - is positive')
    with open('sentiment_analysis/local_storage.txt', 'w',
encoding='utf-8') as file:
        file.write(f'{line} - is positive') #
else:
    print(f'Positive accuracy =
{positive_correct/positive_count*100.0}% via {positive_count} samples')
    # print(is_positive(polariry_range=0, subjectivity_range=1))

def help(methods: bool, about=True) -> str:
    print('The script that returns is the string or set of strings is positive or
negative', end=")

    if methods:
        print('The module has is_positive() method which takes 0 parameters and
checks if the string or set of string is positive and return it, otherwise it returns
negative')
    # analysis = TextBlob('TextBlob sure looks like it has some interestion features!')
    # print(analysis.translate(to='es'))

Код модуля розпізнавання голосу
import speech_recognition as sr

r = sr.Recognizer()

with sr.Microphone() as source:
    print('Speak anything : ')
    audio = r.listen(source)

```

```

try:
    text = r.recognize_google(audio)
    print(f'You said : {text}')
except:
    print('Sorry could not recognize your voice')

```

Код модуля API:

```

# <ModulesSection modules>

from textblob import TextBlob
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# </ModulesSection>

# +---+ Main script +---+ #
def is_positive(polarity_range=0, subjectivity_range=1) -> str:
    if polarity_range >= 1:
        raise ValueError(f'Polarity range should be between 0 and 1 not inclusive.')

    positive_correct = 0
    positive_count = 0
    only_positive_array = []

with open('sentiment_analysis/positive.txt', encoding='utf-8') as file:
    for line in file.read().split('\n'):

```

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

analysis = TextBlob(line)

if analysis.sentiment.subjectivity < subjectivity_range:
    if analysis.sentiment.polarity > polariry_range:
        only_positive_array.append(line)
        positive_correct += 1
        positive_count += 1

with open('sentiment_analysis/only_positive.txt', 'w', encoding='utf-8') as file:
    for line in only_positive_array:
        file.write('%s\n' % line)

if positive_correct == 0:
    print(f'There are no possitive messages')
    with open('sentiment_analysis/local_storage.txt', 'w', encoding='utf-8') as file:
        file.write('There are no possitive messages') #

elif positive_correct == 1:
    print(f'{line} - is positive')
    with open('sentiment_analysis/local_storage.txt', 'w', encoding='utf-8') as file:
        file.write(f'{line} - is positive') #

else:
    print(f'Positive accuracy = {positive_correct/positive_count*100.0}% via
    {positive_count} samples')
# print(is_positive(polariry_range=0, subjectivity_range=1))

def help(methods: bool, about=True) -> str:
    print('The script that returns is the string or set of strings is positive or negative',
    end=")

```

```
if methods:

    print('The module has is_positive() method which takes 0 parameters and checks
    if the string or set of string is positive and return it, otherwise it returns negative')

# analysis = TextBlob('TextBlob sure looks like it has some interestion features!')

# print(analysis.translate(to='es'))

from flask import Flask, redirect, url_for, render_template, request
from textblob import TextBlob

app = Flask(__name__)

@app.route('/', methods=['POST', 'GET'])
def main():

    if request.method == 'POST':

        message = request.form['nm']

        return(message)

    else:

        return render_template('index.html')

if __name__ == '__main__':

    app.run(debug=True)
```

Код графічного застосунку:

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

import sys

from PyQt5 import QtCore, QtWidgets
from PyQt5.QtGui import QFont
from PyQt5.QtWidgets import(
    QWidget,
    QToolTip,
    QPushButton,
    QApplication,
    QLineEdit,
    QVBoxLayout,
    QLabel,
)

sys.path.insert(1, 'sentiment_analysis')
import sentiment

# print(sentiment.is_positive(polarity_range=0, subjectivity_range=1))

class Example(QWidget):

    def __init__(self):
        super().__init__()

        self.initUI()

    def initUI(self):
```

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

```
QToolTip.setFont(QFont('SansSerif', 10))

self.setToolTip('This is a <b>QWidget</b> widget')

self.le = QLineEdit(self)
self.le.move(130, 22)

self.btn = QPushButton('Check', self)
self.btn.move(20, 20)
self.btn.clicked.connect(self.showDialog)

btn = QPushButton('Button', self)
btn.setToolTip('This is a <b>QPushButton</b> widget')
btn.resize(btn.sizeHint())
btn.move(50, 50)
btn.clicked.connect(self.buttonClicked)

self.setGeometry(300, 300, 300, 200)
self.setWindowTitle('Tooltips')
self.show()

def buttonClicked(self):
    sentiment.is_positive(polarity_range=0, subjectivity_range=1)

def showDialog(self):
    with open('sentiment_analysis/positive.txt', 'w', encoding='utf-8') as file:
        file.write('%s\n' % self.le.text())
    sentiment.is_positive(polarity_range=0, subjectivity_range=1)
```

```
# print(self.le.text())
```

```
class ClssDialog(QtWidgets.QDialog): #  
    def __init__(self, parent=None):  
        super(ClssDialog, self).__init__(parent)  
        self.verticalLayout = QtWidgets.QVBoxLayout(self)  
        self.verticalLayout.setObjectName("verticalLayout")  
        self.pushButton = QtWidgets.QPushButton(self)  
        self.pushButton.setObjectName("pushButton")  
        self.pushButton.clicked.connect(self.btnClosed)  
        self.verticalLayout.addWidget(self.pushButton)  
        self.setWindowTitle("Dialog")  
        self.pushButton.setText("Close Dialog")
```

```
def btnClosed(self):  
    self.close()
```

```
if __name__ == '__main__':
```

```
    app = QApplication(sys.argv)  
    ex = Example()  
    sys.exit(app.exec_())
```

Код модуля розпізнавання об'єктів та облич:

```
import face_recognition  
import os  
import cv2
```

```
KNOWN_FACES_DIR = 'known'
UNKNOWN_FACES_DIR = 'unknown'
TOLERANCE = 0.6
FRAME_THICKNESS = 3
FONT_THINCKESS = 2
MODEL = 'cnn'

print('loading known faces')

known_faces = []
known_names = []

for name in os.listdir(KNOWN_FACES_DIR):
    print('ey')
    for filename in os.listdir(f'{KNOWN_FACES_DIR}/{name}'):
        print('eys')
        image =
face_recognition.load_image_file(f'{KNOWN_FACES_DIR}/{name}/{file_name}')
encoding = face_recognition.face_encodings(image)[0]
        known_faces.append(encoding)
        known_names.append(name)

print('processing unknowns faces')

for filename in os.listdir(UNKNOWN_FACES_DIR):
    print(filename)
    image =
face_recognition.load_image_file(f'{UNKNOWN_FACES_DIR}/{file_name}')
```

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

```
locations = face_recognition.face_locations(image, model=MODEL)
encodings = face_recognition.face_encodings(image, locations)
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

for face_encoding, face_location in zip(encodings, locations):
    results = face_recognition.compare_faces(known_faces, face_encoding,
TOLERANCE)
    match = None

    if True in results:
        match = known_names[results.index(True)]
        print(f'Match found: {match}')

    top_left = (face_location[3], face_location[0])
    bottom_right = (face_location[1], face_location[2])

    color = [0, 255, 0]

cv2.rectangle(image, top_left, bottom_right, color, FRAME_THICKNESS)
top_left = (face_location[3], face_location[0])
bottom_right = (face_location[1], face_location[2]+22)
```

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

```
cv2.rectangle(image, top_left, bottom_right, color, cv2.FILLED)
```

```
    cv2.putText(image, match, (face_location[3]+10, face_location[2]+15),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (200, 200, 200), FONT_THICKNESS)
```

```
cv2.imshow(filename, image)
```

```
cv2.waitKey(0)
```

```
# cv2.destroyAllWindows(filename)
```

```
import face_recognition
```

```
image = face_recognition.load_image_file('./known/bci.jpg')
```

```
face_locations = face_recognition.face_locations(image)
```

```
print(f'there is {len(face_locations)} people in the image')
```

```
import face_recognition
```

```
image_of_known =
```

```
face_recognition.load_image_file('./known/ad6c5200477bd377f69a8d3dc9a7f3cd.png
')
```

```
known_encoding = face_recognition.face_encodings(image_of_known)[0]
```

```
image_of_unknown =
```

```
face_recognition.load_image_file('./unknown/om_nom_nom.jpg')
```

```
unknown_encoding = face_recognition.face_encodings(image_of_unknown)[0]
```

```
result = face_recognition.compare_faces([known_encoding], unknown_encoding)
```

```
if result[0]:
```

```
    print('This is known person')
```

```
else:
```

```
    print('Nah')
```

Код графічного додатку (Веб):

```
import React from "react";
```

```
import ReactDOM from "react-dom";
```

```
import { BrowserRouter } from "react-router-dom";
```

```
import App from "./App";
```

```
import "./index.css";
```

```
ReactDOM.render(  
  <BrowserRouter>
```

```
    <App/>
```

```
  </BrowserRouter>,  
  document.getElementById("root"),
```

```
);
```

```
import React from "react";
```

```
import './node_modules/bootstrap/dist/css/bootstrap.min.css';
```

```

const Template = ({ label,
    input_one,
    input_one_label,
    input_one_placeholder,
    input_two,
    input_two_label,
    input_two_placeholder,
    input_three,
    input_three_label,
    input_three_placeholder,
    input_four,
    input_four_label,
    input_four_placeholder,
    input_five,
    input_five_label,
    input_five_placeholder,
    button,
    button_text,
}) => {
const form_style = {
    border: '2px solid',
    width: '300px',
    marginLeft: '40%',
    marginTop: '15%',
},
input_style = {
    borderRadius: '0px',

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

    height: '20px',
    border: '2px solid',
  },
  button_style = {
    backgroundColor: 'lightblue',
    marginTop: '30px',
    marginBottom: '10px',
    borderRadius: '30px',
    height: '30px',
    width: '100px',
    marginLeft: '60%',
  };
return(
  <form style={form_style}>
    <h2>{label}</h2>

    {input_one == 'True' ? (
      <div className="form-group">
        <label>{input_one_label}</label>
        <input className="form-control" placeholder={input_one_placeholder}
style={input_style}/>
      </div>) : (<span/>)}

    {input_two == 'True' ? (
      <div className="form-group">
        <label>{input_two_label}</label>

```

```

    <input type="password" className="form-control"
placeholder={input_two_placeholder} style={input_style}/>
    </div> : (<span/>)}

    {input_three == 'True' ? (
    <div className="form-group">
    <label>{input_three_label}</label>
    <input type="password" className="form-control"
placeholder={input_three_placeholder} style={input_style}/>
    </div> ) : (<span/>)}

    {input_four == 'True' ? (
    <div className="form-group">
    <label>{input_four_label}</label>
    <input type="password" className="form-control"
placeholder={input_four_placeholder} style={input_style}/>
    </div> ) : (<span/>)}

    {input_five == 'True' ? (
    <div className="form-group">
    <label>{input_five_label}</label>
    <input type="password" className="form-control"
placeholder={input_five_placeholder} style={input_style}/>
    </div> ) : (<span/>)}

```

```
{ button == 'True' ? (<button type="submit" className="btn btn-primary btn-block" style={button_style}>{button_text}</button>) : (<span/>)}
  </form>
);
}; export default Template;
```

```
</form>
```

```
);
```

```
}; export default Template;
```

```
// TODO: Rewrite these inputs to one kind
```

```
import React, { Component } from "react";
```

```
import Navbar from './components/navbar.react';
```

```
export default class Main extends Component {
```

```
  render() {
```

```
    return(
```

```
      <Navbar/>
```

```
    );
```

```
  }
```

```
}
```

```
import React, { Component } from "react";
```

```
import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
```

					ДП 6206.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

```

class Textarea extends Component {
  render() {
    return(
      <textarea
        class={this.props.textarea_class}
        name={this.props.textarea_name}
        defaultValue={this.props.textarea_value}
        placeholder={this.props.textarea_title}
        key={this.props.textarea_key}
        onChange={this.props.on_change}
        id={this.props.textarea_id}/>
    );
  }
} export default Textarea;

import React, { Component } from "react";
import '../node_modules/bootstrap/dist/css/bootstrap.min.css';

import Form from '../components/form.react';

class TitleEditField extends Component {
  state = {
    initialData: this.props.initialData,
  };
};

```

```

hidden_style = {
  opacity: '0',
};

pOnClick = e => {
  e.preventDefault();
  let div = document.getElementById(this.props.DIV_id),
      p = document.getElementById(this.props.P1_id);

  div.style.opacity = '1';
  p.style.opacity = '0';

  if(this.props.title_is_visible) {
    let p2 = document.getElementById(this.props.P2_id);
    p2.style.opacity = '1';
  }
}

buttonOnClick = e => {
  e.preventDefault();
  let p = document.getElementById(this.props.P1_id),
      div = document.getElementById(this.props.DIV_id);

  p.style.opacity = '1';
  div.style.opacity = '0';

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

if(this.props.title_is_visible) {
    let p2 = document.getElementById(this.props.P2_id);
    p2.style.opacity = '0';
}

let name = document.getElementById(this.props.textarea_id).value;
this.setState({ initialState: name, });
console.log(name);
}

funcOnChange = () => {
    console.log('onChange works');
}

render() {
    return(
        <div key={this.props.initialData}>
            <p id={this.props.P1_id}
onClick={this.pOnClick}>{this.state.initialData}</p>
            {
                this.props.title_is_visible ? (
                    <p id={this.props.P2_id}
style={this.hidden_style}>{this.props.title}</p>
                ) : (null)
            }
        )
    )
}

```

```

<div id={this.props.DIV_id} style={this.hidden_style}>
  <Form textarea={true}
    textarea_id={this.props.textarea_id}
    on_click={this.buttonOnClick}
    textarea_title={this.props.initialData}
    textarea_value={this.props.initialData}/>
</div>
</div>
);
}
} export default TitleEditField;
import React, { Component } from "react";
import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
class Input extends Component {
  render() {
    return(
      <input
        name={this.props.input_name}
        value={this.props.input_value}
        onChange={this.props.on_change}
        placeholder={this.props.title}
        type={this.props.input_type}
        key={this.props.input_key}
        class={this.props.class_name}
        style={this.props.input_style}/>
    )
  }
}

```

```

    );
  }
} export default Input;

export default class Users extends Component {
  constructor(props) {
    super(props);
    this.state = {
      loading: true,
      data: [],
      group_name: "",
    }
  }

  componentDidMount() {
    if(!localStorage.getItem('access_token'))
      location.replace('http://localhost:3000/');

    axios.post(`https://api-support.ligataxi.su/api/v1/jsonrpc`,
      JSON.stringify({
        method: "user.AllUser",
        id: 0,
        params: {},
      })), {
      dataType: 'json',
    },
  },

```

```

{
  headers: { access_token: localStorage.getItem('access_token'), }
})
.then(res => {
  this.setState({
    loading: false,
    data: res.data.result.data,
  });
})
}

render() {
  let data = [],
      data_dict = { data: this.state.data },
      columns = [{ name: 'Surname', selector: 'surname', sortable: true, },
                  { name: 'Name', selector: 'name', sortable: true, },
                  { name: 'E-mail', selector: 'email', sortable: true, },
                  { name: 'Delete', selector: 'delete', right: true, },];
      console.log(data_dict)

  for(let index in data_dict.data) {
    data.push({
      surname:
        <Link class="nav-link text-dark h6"
to={`/user/${data_dict.data[index].id}`}>{data_dict.data[index].last_name}</Link>,

```

name:

```
<span class="nav-link text-dark
h6">{data_dict.data[index].first_name}</span>,>
```

email:

```
<span class="nav-link text-dark
h6">{data_dict.data[index].email}</span>,>
```

delete:

```
<Link class="btn btn-danger" to={`/remove-
user/${data_dict.data[index].id}`}>Delete</Link>,>
```

```
},);>
```



```
if(this.state.loading) {>
```

```
return(>
```

```
<React.Fragment>>
```

```
<Navbar/>>
```

```
<p class="mt-5 text-dark h2">Loading...</p>>
```

```
</React.Fragment>>
```

```
);>
```



```
createTheme('solarized', {>
```

```
background: {>
```

```
default: 'rgba(0,0,0,.08)',>
```



```
});>
```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації і управління

УЗГОДЖЕНО

Керівник проекту

_____ Алла КОГАН
(підпис) (вл. ім'я, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“14” квітня 2020 р.

**СИСТЕМА ДЛЯ АНАЛІЗУ, ФІЛЬТРАЦІЇ ТА СТРУКТУРУВАННЯ
КОНТЕНТУ**

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 6206.01.000 ТЗ*

на 7 сторінках

Київ – 2020 року

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	3
1.1 Повне найменування системи та її умовне позначення.....	3
1.2 Найменування організації-замовника та організації-учасників робіт.....	3
1.3 Перелік документів, на підставі яких створюється система.....	3
1.4 Планові терміни початку і закінчення роботи зі створення системи.....	3
2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ.....	4
2.1 Призначення системи.....	4
2.2 Цілі створення системи.....	4
3 ХАРАКТЕРИСТИКА ОБ'ЄКТУ АВТОМАТИЗАЦІЇ.....	4
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	4
4.1 Вимоги до функціональних характеристик.....	4
4.2 Вимоги до надійності.....	4
4.3 Умови експлуатації.....	5
4.4 Вимоги до складу і параметрів технічних засобів.....	5
5 СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	6
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ.....	7
6.1 Види випробувань.....	7

					ДП 6206.01.000 ТЗ						
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>							
Розроб.		Патлай М.О.			Система для аналізу, фільтрації та структурування контенту						
Керівн		Коган А.В									
Н. кон.		Телишеа Т.О.									
Затв.		Павлов О.А.									
					<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><i>Лім.</i></td> <td style="text-align: center;"><i>Лист</i></td> <td style="text-align: center;"><i>Листів</i></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">7</td> </tr> </table>	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>	1	2	7
<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>									
1	2	7									
					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361						

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Система призначена для аналізу, структурування та фільтрації контенту у інформаційних мережах.

2.2 Цілі створення системи

Цілями створення є:

- аналіз тексту за емоційним забарвленням;
- аналіз графічного контенту;
- підтримка існуючих програмних реалізацій;
- структурування контенту по відповідним тек;
- власне користування.

3 ХАРАКТЕРИСТИКА ОБ'ЄКТУ АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є процес аналізу та фільтрації контенту.

Для того, щоб вирішити задачу аналізу та фільтрації контенту необхідно:

- підключити програмний продукт до свого сервісу або використовувати офіційне графічне оточення;
- вибрати необхідний метод або опцію;
- надати необхідні дані у вигляді речення, фотографії або дата сету;
- зберегти результат у текстовому файлі, файлі csv або у базі даних.

По завершенню роботи ми отримаємо систему, яка дозволяє аналізувати, фільтрувати та структурувати контент по різним критеріям і параметрам.

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Задача підготовки побудови система для аналізу, фільтрації та структурування контенту вимагає наявного графічного редактору діаграм.

4.2 Вимоги до надійності

Система повинна адекватно реагувати на помилки застосування та видавати відповідні повідомленні користувачеві, а також записувати їх у log-файл.

											ДП 6206.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата								4

4.3 Умови експлуатації

Для адекватної роботи системи необхідний пристрій з платформою, яка відповідає вимогам зазначеним в розділі 4.4 .

Усі користувачі системи повинні дотримуватися правил експлуатації електронної обчислювальної техніки.

4.4 Вимоги до складу і параметрів технічних засобів

Даний програмний продукт представлений у вигляді графічної середи або у вигляді АРІ, якщо спосіб використання - це підключення до готового програмного продукту нашу систему.

Для коректної роботи алгоритму необхідний Python 3, dlib. Для графічного застосунку необхідний PyQt5. АРІ використовує Flask для серверної логіки.

					ДП 6206.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

У таблиці 5.1 наведено календарний план робіт та терміни їх виконання.

Таблиця 5.1 – Календарний план виконання робіт

№ з/п	Назва етапів створення продукту	Строк виконання
1.	Вивчення рекомендованої літератури	15.02.2020 р.
2.	Аналіз існуючих методів розв'язання задачі	22.02.2020 р.
3.	Постановка та формалізація задачі	01.03.2020 р.
4.	Розробка інформаційного забезпечення	15.03.2020 р.
5.	Алгоритмізація задачі	22.03.2020 р.
6.	Обґрунтування використовуваних технічних засобів	30.03.2020 р.
7.	Розробка програмного забезпечення	20.04.2020 р.
8.	Налагодження програми	27.04.2020 р.
9.	Виконання графічних документів	04.05.2020 р.
10.	Оформлення пояснювальної записки	18.05.2020 р.
11.	Подання ДП на попередній захист	22.05.2020 р.
12.	Подання ДП на основний захист	01.05.2020 р.
13.	Подання ДП рецензенту	05.06.2020 р.

Змн.	Арк.	№ докум.	Підпис	Дата

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

6.1 Види випробувань

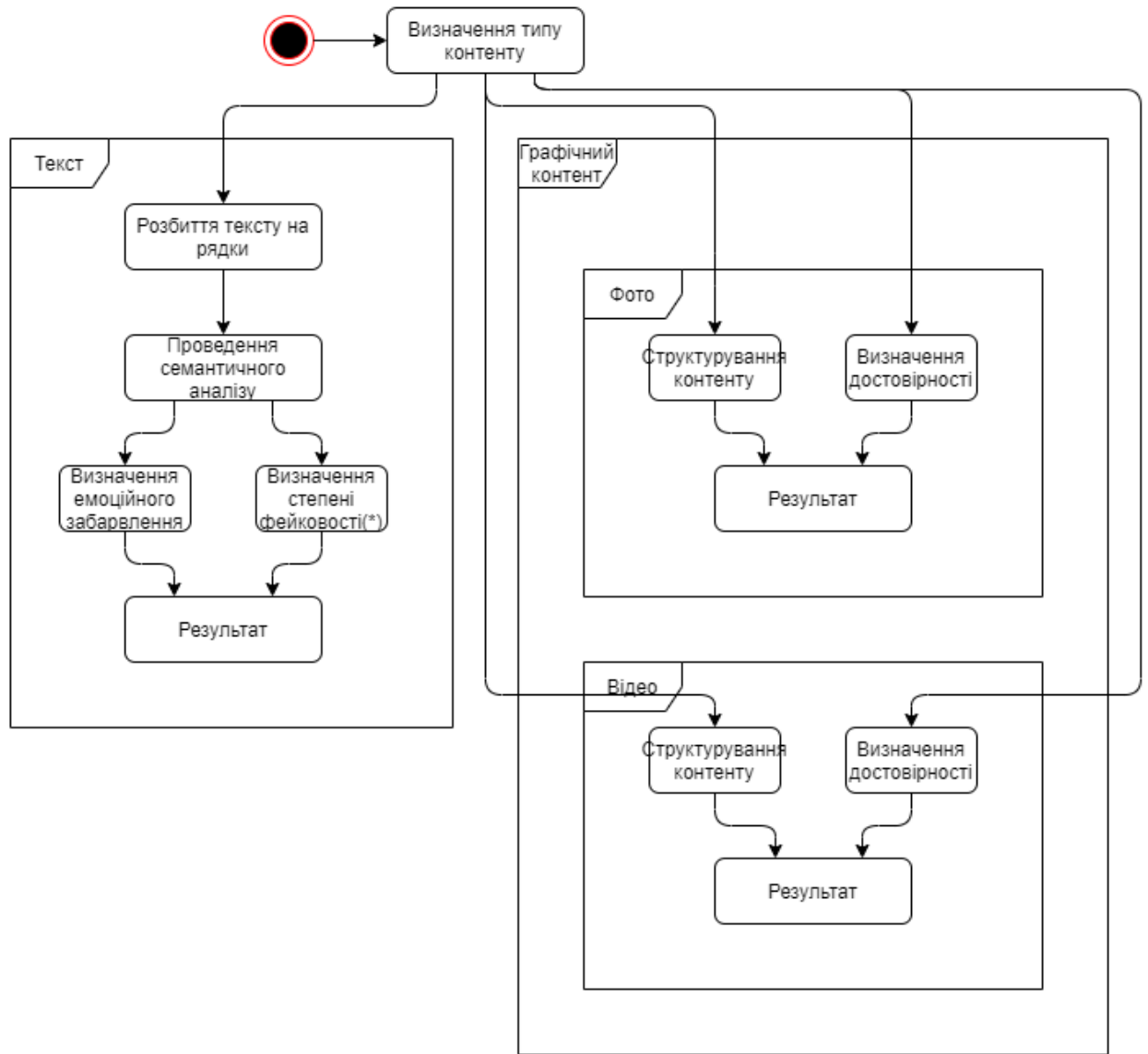
Види випробувань узгоджуються із замовником до проведення випробувань. Здача - прийом робіт виконується поетапно на комп'ютерах замовника в аудиторіях кафедри АСОІУ у відповідності з робочою програмою та календарним планом.

Усі програмні продукти, що створюються в рамках даної системи передаються замовнику як у вигляді готових модулів, так і у вигляді вихідних кодів, представлених в електронній формі.

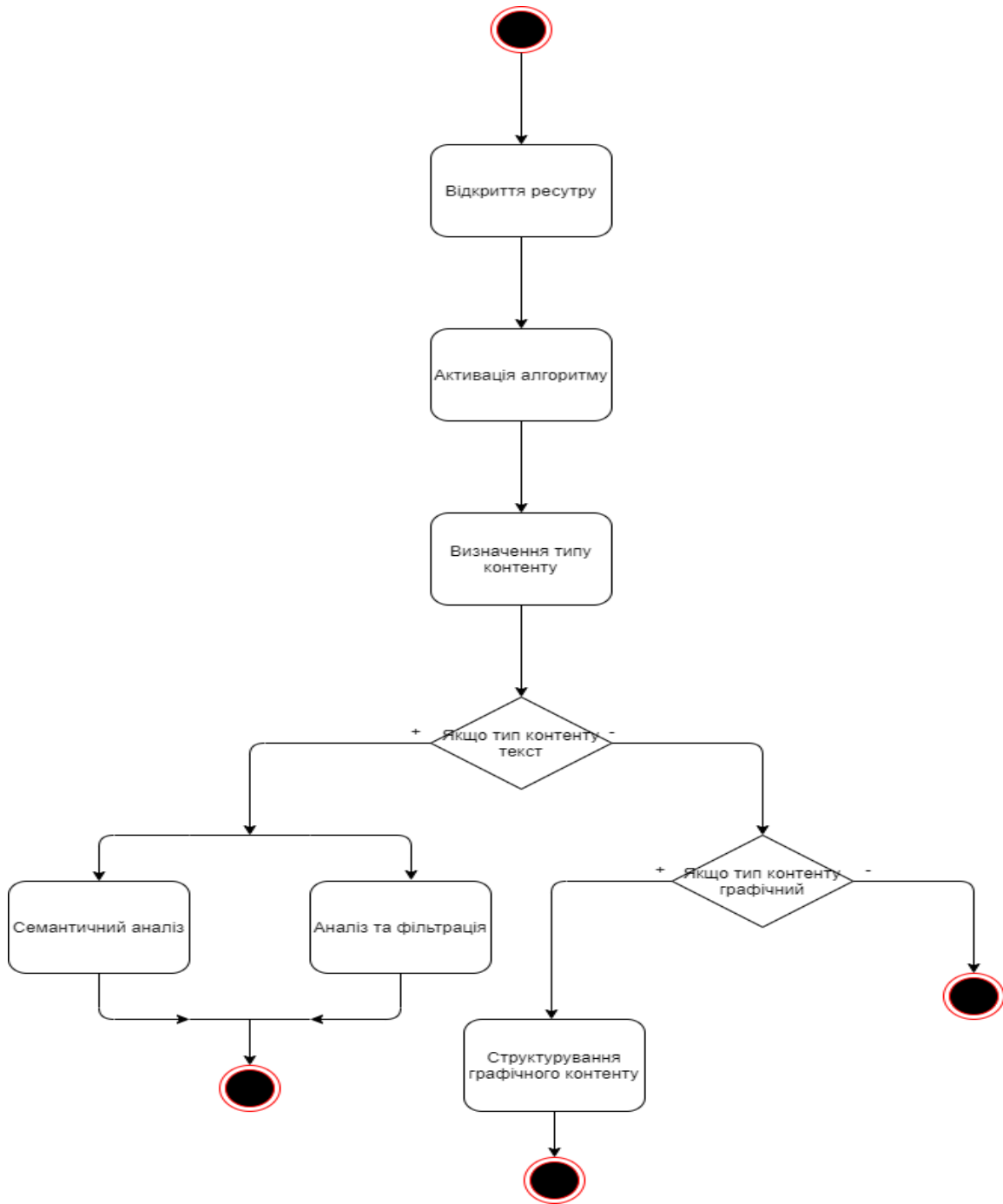
					ДП 6206.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

**Графічний матеріал
до дипломного проекту**

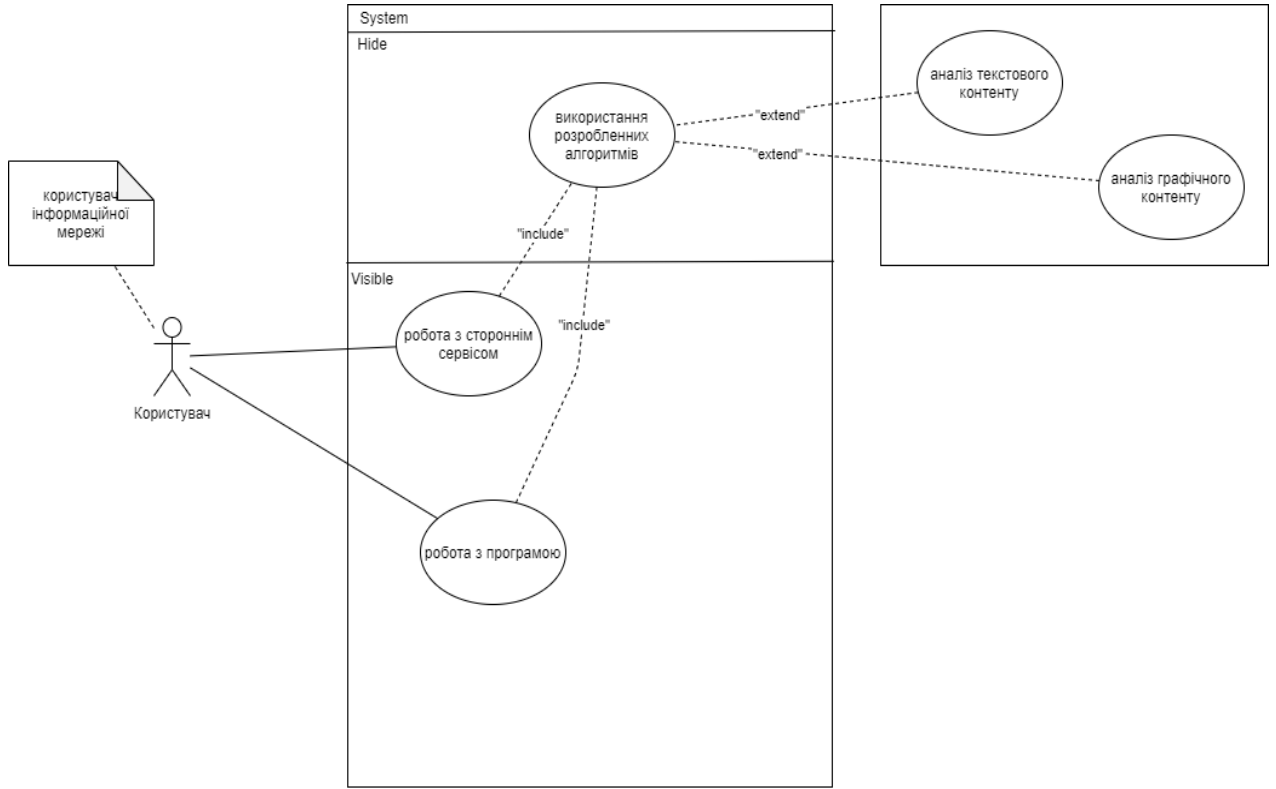
на тему: Система для аналізу, фільтрації та структурування контенту.



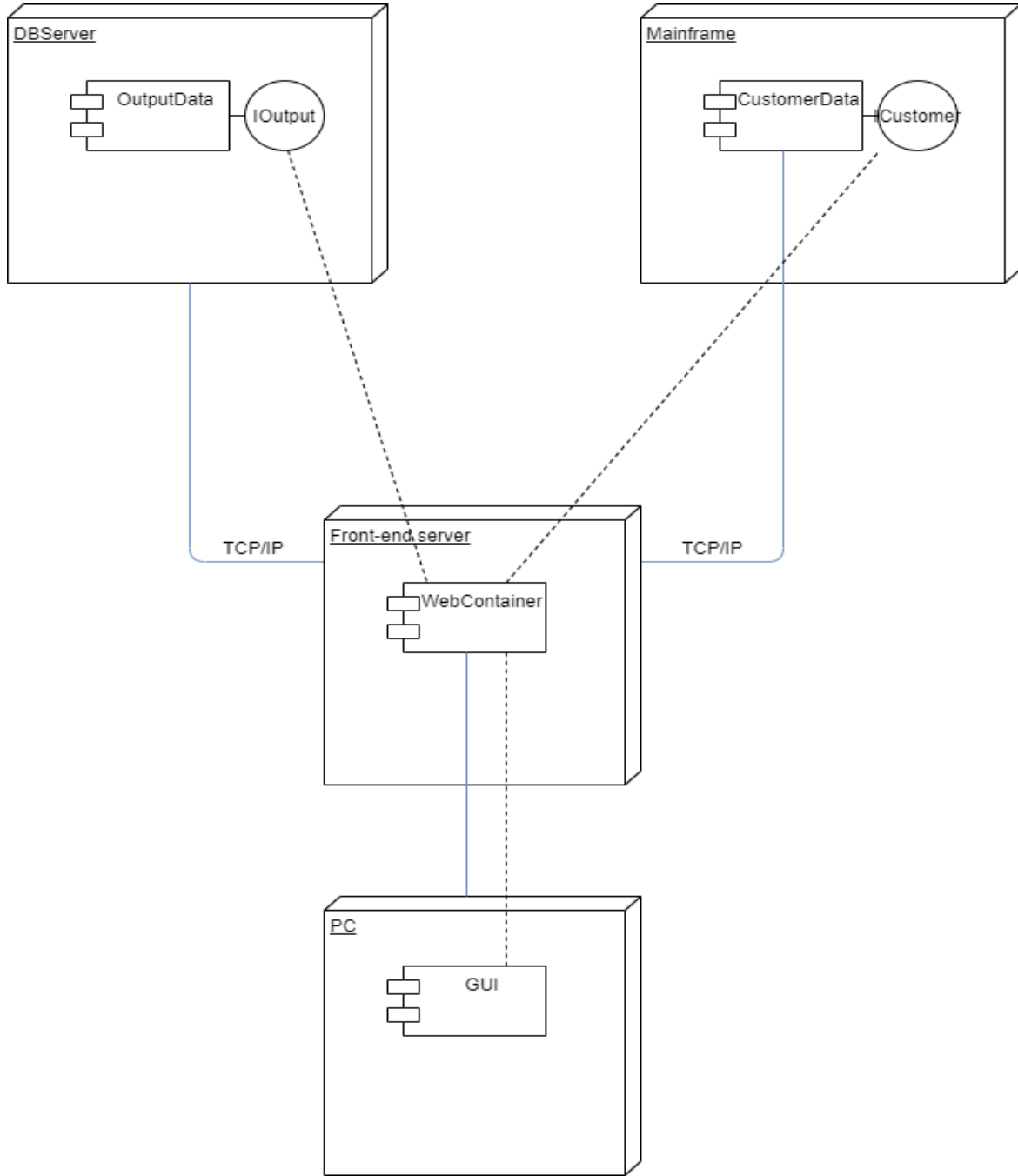
					ДП 6206.02.000.ССС		
					Літера		
					Маса		Масштаб
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна станів системи		
Розроб.	Патлай М.О.						
					Аркуш 1		Аркушів 7
Перев.	Коган А.В.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Н. кон.	Телишева Т.О.						
Затв.	Коган А.В.						
Система для аналізу, фільтрації та структурування контенту							



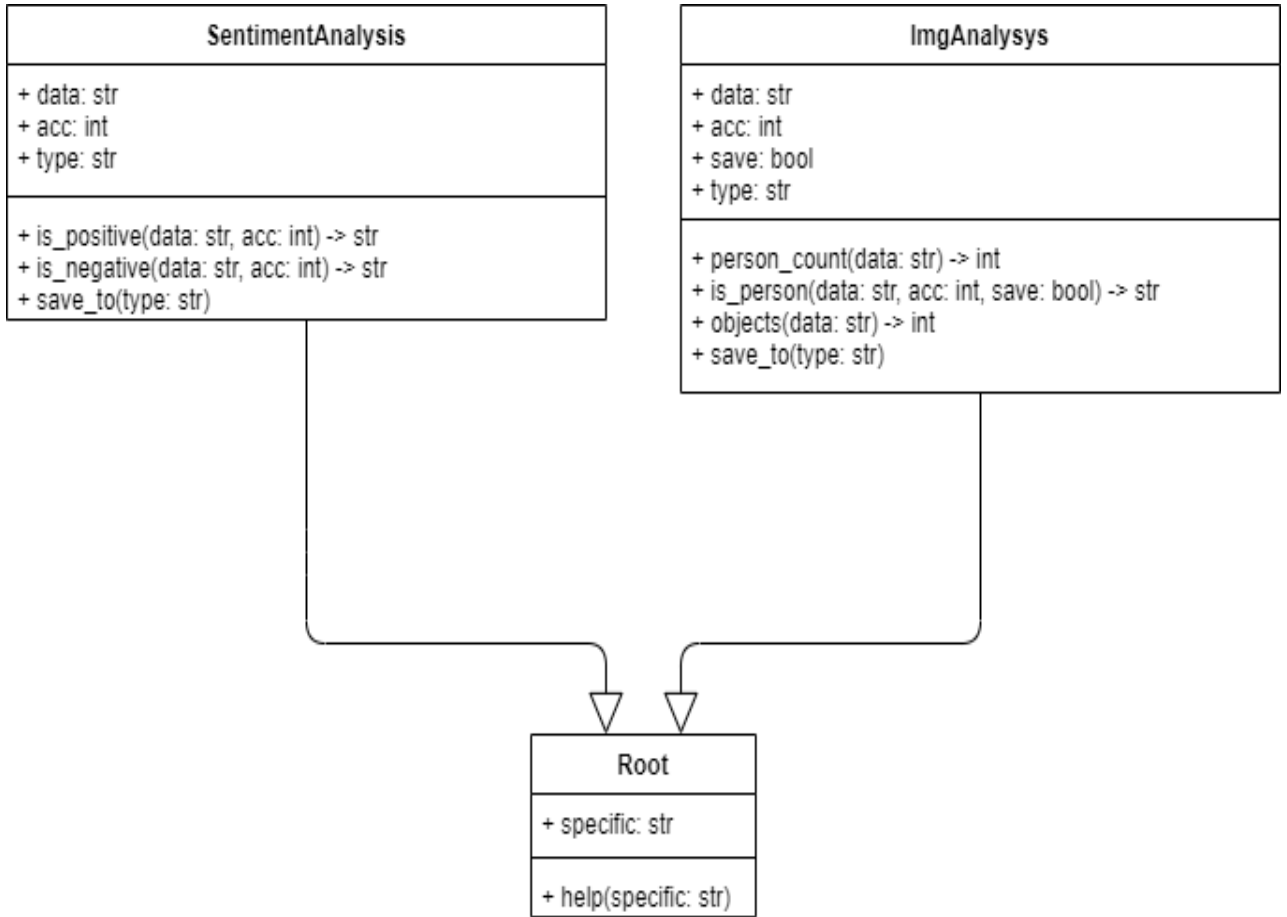
					ДП 6206.03.000.ССД			
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна діяльності	Літера	Маса	Масштаб
Розроб.		Патлай М.О.						
Перев.		Коган А.В.				Аркуш 2	Аркушів 7	
Н. кон.		Телишева Т.О.				Система для аналізу, фільтрації та структурування контенту		
Затв.		Коган А.В.			КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361			



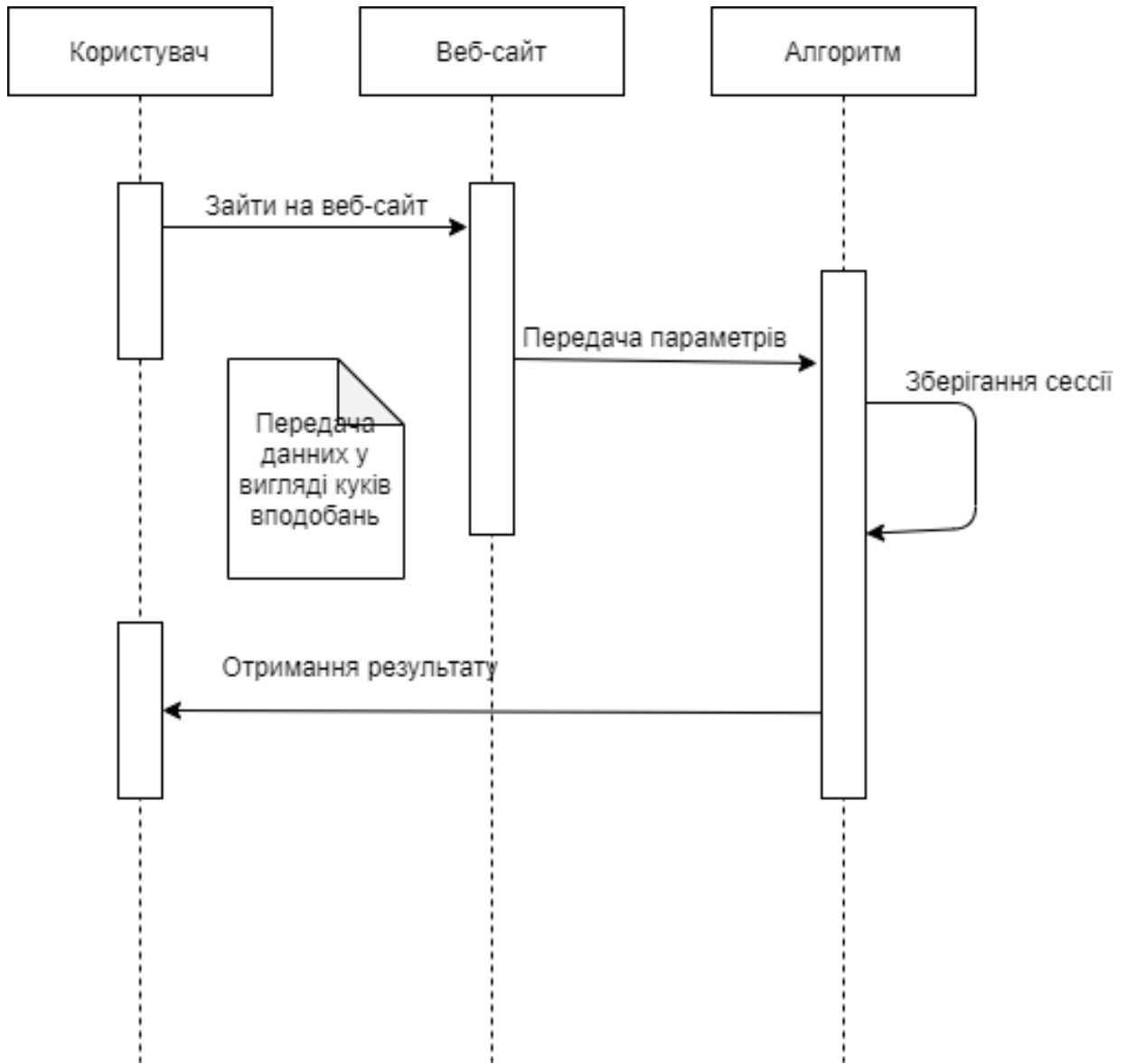
					ДП 6206.04.000.ССВ			
						Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна варіантів використань			
Розроб.		Патлай М.О.						
Перев.		Коган А.В.			Аркуш 3		Аркушів 7	
Н. кон.		Телишева Т.О.			Система для аналізу, фільтрації та структурування контенту			
Затв.		Коган А.В.						
					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361			



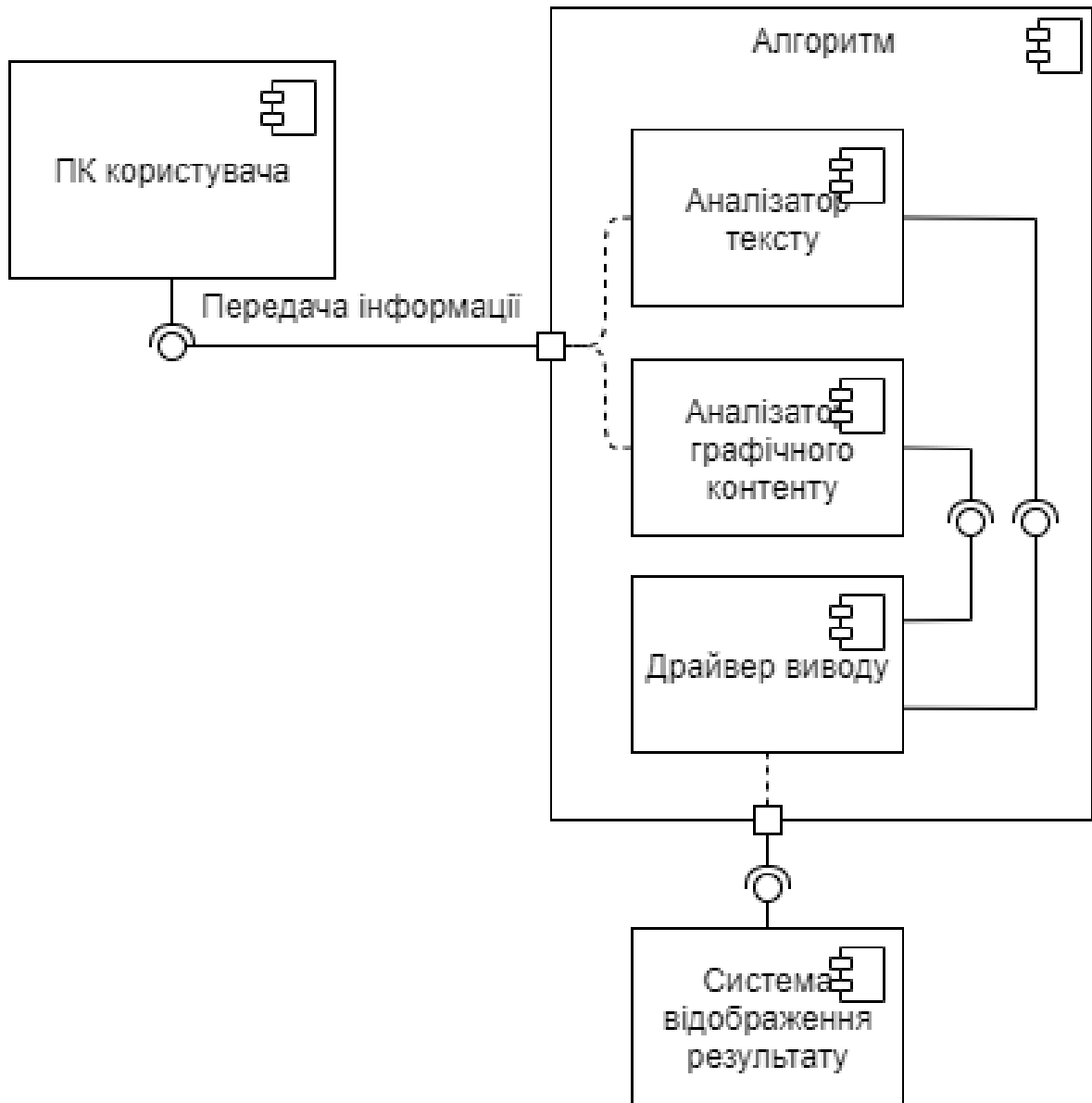
					ДП 6206.05.000.ССК			
					Схема структурна компонентів програмного забезпечення	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розроб.		Патлай М.О.						
						Аркуш 4	Аркушів 7	
Перев.		Коган А.В.			Система для аналізу, фільтрації та структурування контенту	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Н. кон.		Телишева Т.О.						
Затв.		Коган А.В.						



					ДП 6206.06.000.ССК							
					Схема структурна класів програмного забезпечення			Літера	Маса	Масштаб		
Зм.	Арк.	№ документа	Підпис	Дата								
Розроб.		Патлай М.О.			Система для аналізу, фільтрації та структурування контенту			Аркуш 5		Аркушів 7		
Перев.		Коган А.В.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361				
Н. кон.		Телишева Т.О.										
Затв.		Коган А.В.										



					ДП 6206.07.000.ССП			
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна послідовності	Літера	Маса	Масштаб
Розроб.		Патлай М.О.						
Перев.		Коган А.В.				Аркуш 6	Аркушів 7	
Н. кон.		Телишева Т.О.			Система для аналізу, фільтрації та структурування контенту	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Затв.		Коган А.В.						



					ДП 6206.08.000.ССК			
					Схема структурна компонентів програмного забезпечення	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розроб.		Патлай М.О.			Система для аналізу, фільтрації та структурування контенту	Аркуш 7		Аркушів 7
Перев.		Коган А.В.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Н. кон.		Телишева Т.О.						
Затв.		Коган А.В.						