

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«СИСТЕМА АВТОМАТИЗОВАНОЇ РОЗВІДКИ З ВИКОРИСТАННЯМ
ВІДКРИТИХ ДЖЕРЕЛ ІНФОРМАЦІЇ»

Виконав: студент 2-го курсу, групи 1БС-20м
спеціальності 125 – Кібербезпека

(шифр і назва напрямку підготовки, спеціальності)

_____ Панченко Б. Д.
(прізвище та ініціали)

Керівник к. т. н., доц. каф. ЗІ

_____ Войтович О. П.
(прізвище та ініціали)

Опонент: к. т. н., доц. каф. ОТ

_____ Войцеховська О. В.
(прізвище та ініціали)

« ____ » _____ 2021 р.

Допущено до захисту

Завідувач кафедри ЗІ

д. т. н., проф.

_____ Лужецький В. А.

« ____ » _____ 2021 р.

Вінниця ВНТУ – 2021 року

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Освітній ступінь магістр
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри ЗІ, д. т. н., проф.

_____ **В. А. Лужецький**

_____ **2021 року**

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Панченку Богдану Дмитровичу

1. Тема роботи: «Система автоматизованої розвідки з використанням відкритих джерел інформації» керівник роботи: Войтович Олеся Петрівна, к. т. н., доц. каф. ЗІ,

затверджена наказом ректора ВНТУ 24 вересня 2021 року №277.р.

2. Строк подання студентом роботи 17 грудня 2021 р.

3. Вихідні дані до роботи:

- отримання даних розвідки з веб-сервера;
- можливість розвідки осіб та організацій;
- спосіб реалізації – програмний засіб;
- мова програмування C#.

4. Зміст розрахунково-пояснювальної: Вступ. Аналіз інформаційних джерел. Розробка архітектури системи. Алгоритми роботи системи для автоматизованої розвідки з використанням відкритих джерел інформації. Реалізація і тестування системи для автоматизованої розвідки з використанням відкритих джерел інформації. Економічна частина. Висновки. Перелік інформаційних джерел. Додатки.

5. Перелік ілюстративного матеріалу.

Система автоматизованої розвідки з використанням відкритих джерел інформації. Загальна схема модулів (плакат, А4). Схема алгоритму розвідки

(плакат, А4). Схема алгоритму парсингу (плакат, А4). Схема бази даних (плакат, А4). UML-діаграма (плакат, А4). Результати тестування (плакат, А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Войтович О. П., к. т. н., доц. каф. ЗІ		
2	Войтович О. П., к. т. н., доц. каф. ЗІ		
3	Войтович О. П., к. т. н., доц. каф. ЗІ		
4	Лесько О. Й., к. е. н., проф. каф. ЕПВМ		

7. Дата видачі завдання 24 вересня 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2021 – 04.09.2021	
2	Аналіз інформаційних джерел за напрямком магістерської кваліфікаційної роботи	05.09.2021 – 15.09.2021	
3	Науково-технічне обґрунтування	16.09.2021 – 22.09.2021	
4	Розробка технічного завдання	23.09.2021 – 04.10.2021	
5	Розробка архітектури системи, критеріїв оцінки інформаційних джерел	05.10.2021 – 24.10.2021	
6	Розробка алгоритму, програмна реалізація та тестування системи для автоматизованої розвідки з відкритих джерел інформації	25.10.2021 – 17.11.2021	
7	Аналіз виконання ТЗ, висновки	18.11.2021 – 24.11.2021	
8	Оформлення пояснювальної записки	25.11.2021 – 30.11.2021	
9	Попередній захист та доопрацювання МКР	03.12.2021	
10	Перевірка магістерської роботи на наявність плагіату	03.12.2021 – 04.12.2021	
11	Представлення МКР до захисту, рецензування	05.12.2021 – 20.12.2021	
12	Захист МКР	21.12.2021 – 23.12.2021	

Студент _____ Б. Д. Панченко
 Керівник роботи _____ О. П. Войтович

АНОТАЦІЯ

УДК 004.056

Панченко Б. Д. Система автоматизованої розвідки з використанням відкритих джерел інформації. Магістерська кваліфікаційна робота зі спеціальності 125 Кібербезпека, освітня програма – Безпека інформаційних і комунікаційних систем. Вінниця: ВНТУ, 2021. 101 с.

На укр. мові. Бібліогр.: 41 назв; рис.: 27; табл. 7.

Магістерська кваліфікаційна робота присвячена розробці системи автоматизованої розвідки з використанням відкритих джерел в задачах кібербезпеки інформації. Здійснено аналіз сучасних методів та систем розвідки з використанням відкритих даних. Розроблено алгоритм роботи системи для автоматизованої розвідки в задачах інформаційної безпеки та кібербезпеки. Розроблено програмний засіб, що дозволяє здійснювати автоматизовану розвідку. Виконано тестування системи автоматизованої розвідки з використанням відкритих джерел інформації.

Графічна частина складається з 6 плакатів з демонстрацією результатів проведених досліджень та розробок.

Оцінку витрат на розробку представлено в економічному розділі.

Ключові слова: Osint, розвідка, відкриті джерела

ABSTRACT

Panchenko B. D. System for automated reconnaissance using open sources of information. Master's thesis in the specialty 125 - cybersecurity. Vinnytsia: VNTU, 2021. 101 p.

In Ukrainian language. Bibliographer: 41 titles; fig .: 27; table. 7.

The master's thesis is devoted to the development of an automated intelligence system using open sources in cybersecurity tasks. The analysis of modern methods and systems of intelligence with the use of open data is carried out. An algorithm for the operation of automated intelligence in information security and cybersecurity has been developed. A software tool has been developed that allows automated reconnaissance. The automated intelligence system was tested using open information sources.

The graphic part consists of 6 posters demonstrating the results of modeling and research.

In the Economic part the expenses for development are estimated.

Keywords: Osint, reconnaissance, open source.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	9
1.1 Науково-технічне обґрунтування	9
1.2 Аналіз відкритих джерел інформації	12
1.3 Аналіз стану відкритих інформаційних джерел в Україні.....	18
1.4 Аналіз існуючих програмних засобів.....	24
1.5 Постановка завдання.....	26
2 РОЗРОБКА АЛГОРИТМУ РОЗВІДКИ ТА СТРУКТУРИ СИСТЕМИ	29
2.1 Розробка критеріїв оцінки інформаційних джерел.....	29
2.2 Розробка алгоритму розвідки.....	36
2.3 Вибір програмних технологій засобу.....	41
2.4 Розробка структури програмної системи.....	43
3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОЇ РОЗВІДКИ.....	45
3.1 Розробка алгоритму роботи додатку	45
3.2 Програмна реалізація додатку	49
3.3 Тестування додатку.....	54
4 ЕКОНОМІЧНА ЧАСТИНА.....	60
4.1 Оцінювання комерційного потенціалу розробки (технологічний аудит розробки).....	60
4.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи	65
4.3 Розрахунок мінімальної ціни та чистого прибутку від реалізації розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації.....	71
4.4 Розрахунок терміну окупності коштів вкладених у наукову розробку системи для автоматизованої розвідки з використанням відкритих джерел інформації.....	72
4.5 Висновки до розділу	72
ВИСНОВКИ.....	74
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	76

Додаток А Технічне завдання	80
Додаток Б Код серверної частини	84
Додаток В Код клієнтської частини	92
Додаток Г Критерії оцінювання комерційного потенціалу розробки	98
Додаток Д Результати перевірки роботи на плагіат	100

ВСТУП

Оскільки кожна людина, організація і під'єднаний до інтернету пристрій залишає в мережі свій інформаційний слід, велику частину цих даних можна знайти за допомогою легальних інструментів, що є у відкритому доступі. З появою Інтернету, соціальних медіа та цифрових сервісів з'явилися широкі можливості для збору інформації про ІТ-інфраструктуру організації, а також її співробітників. Здійснення автоматизованої розвідки допомагає знайти загальнодоступну інформацію про працівників, внутрішню діяльність компанії, а також дані за її межами. Іноді конфіденційна інформація міститься в метаданих, які організація випадково опублікувала. Таким чином, аналіз інформації, що знаходиться у відкритому доступі може використовуватись і зловмисниками для проведення атак і спеціалістами з кібербезпеки для запобігання таких атак, тому актуальною задачею є розробка системи автоматизованої розвідки з використанням відкритих джерел.

Люди є основним компонентом організацій. Недостатньо ретельний підбір персоналу може призвести до серйозних проблем з її безпекою. Оскільки час, який люди щороку проводять в інтернеті з кожним роком росте, разом з цим збільшується кількість накопиченої про них на різноманітних ресурсах інформації. У зв'язку з цим у кожної персони сформовано дві особистості, які можна диференціювати, як онлайн та офлайн особистості. Часто інформація з соціальних мереж та схожих онлайн ресурсів може бути недостовірною чи сфабрикованою, тому варто застосувати такий підхід до розробки системи, при якому можливим було б розділення отриманої інформації на достовірну та неперевірену. Також доцільною є розробка критеріїв оцінки якості джерел, що дозволить здійснювати оцінку джерела на доцільність його додавання до бази, та оцінка доданих джерел на те, наскільки негативною можна вважати присутність в ній запису про людину.

Об'єктом дослідження є процес збору та організації інформації з відкритих джерел в задачах кібербезпеки.

Предметом є системи для здійснення автоматизованої розвідки з використанням відкритих джерел інформації в задачах кібербезпеки.

Метою дипломної роботи є покращення пошуку інформації з відкритих джерел для запобігання атак у кіберпросторі з використанням відкритих даних.

Для досягнення мети необхідно:

- проаналізувати відкриті джерела інформації;
- проаналізувати стан відкритих джерел в Україні;
- проаналізувати системи для здійснення автоматизованої розвідки;
- розробити архітектуру системи;
- розробити критерії оцінки інформативності джерел;
- розробити критерії оцінки негативності джерел;
- розробити алгоритм роботи системи;
- розробити програмну систему;
- виконати перевірку коректності роботи системи.

Наукова новизна магістерської роботи полягає в тому, що покращено пошук інформації з відкритих джерел для запобігання атак у кіберпросторі з використанням відкритих даних шляхом створення системи, яка на відміну від інших диференціює отримані результати за негативністю, а джерела підбираються на основі критеріїв достовірності та інформативності, що дозволяє підвищити ефективність роботи спеціалістів з кібербезпеки.

Практична цінність: система автоматизованої розвідки з використанням відкритих джерел інформації.

Результати магістерської роботи доповідалися на таких конференціях:

- XLIX регіональна н. т. к. професорсько-викладацького складу, співробітників та студентів університету з участю працівників науково-дослідних організацій та інженерно-технічних працівників підприємств м. Вінниці та області відбулась 18–29 травня 2020 р. [1];

За результатами магістерської кваліфікаційної роботи опубліковано тези доповідей у збірниках матеріалів конференції [1].

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1.1 Науково-технічне обґрунтування

Люди є найважливішою частиною організації, але вони також можуть бути її слабкою ланкою з позиції кібербезпеки. Згідно з деякими з останніх досліджень станом на 2021 рік до 34% всіх організацій щорічно піддаються впливу інсайдерських атак [2].

Розвідка відкритих джерел інформації (з англійського Open Source Intelligence, OSINT) – це один з важливих та широко застосовуваних державними та приватними організаціями інструментів кібербезпеки.

OSINT має чітко визначену та точну методологію. З науково-технічної точки зору, особливо цікавими є його три основні етапи.

Першим є етап збору, на якому загальнодоступні дані відбираються з відкритих джерел відповідно до цілі або мети. Зокрема, Інтернет є одним з найкращих джерел інформації завдяки обсягу наявного матеріалу та легкодоступності. Процес збору особливо важливий, тому що з цього етапу запускається весь процес формування розвідки [3].

Потім, на етапі аналізу, зібрана інформація обробляється для отримання цінної та зрозумілої інформації. Дані самі по собі не є корисними, тому їх слід інтерпретувати, щоб отримати перші факти, отримані з поглибленого аналізу [3].

Нарешті, у процесі вилучення знань інформація, оброблена раніше, береться як основа для більш складних алгоритмів висновків. Таким чином можна виявляти певні закономірності, поведінку профілю, прогнозувати значення або співвідносити події [3].

Використовуючи отриману з цих джерел інформацію, OSINT постійно розширює обсяг даних про ціль. Таким чином, знайдена інформація знову стимулює процес збору [4].

Найпростішим прикладом OSINT є використання традиційних пошукових систем. Використовуючи пошукові системи, користувач виконує певний запит,

керуючись даними з якого, пошукові системи надають користувачеві інформацію знайдену по запиту.

OSINT використовується для широкого спектру завдань, кожне з яких потребує свого підходу. На рисунку 1.1 зображені основні області застосування розвідки в відкритих джерелах інформації [4].

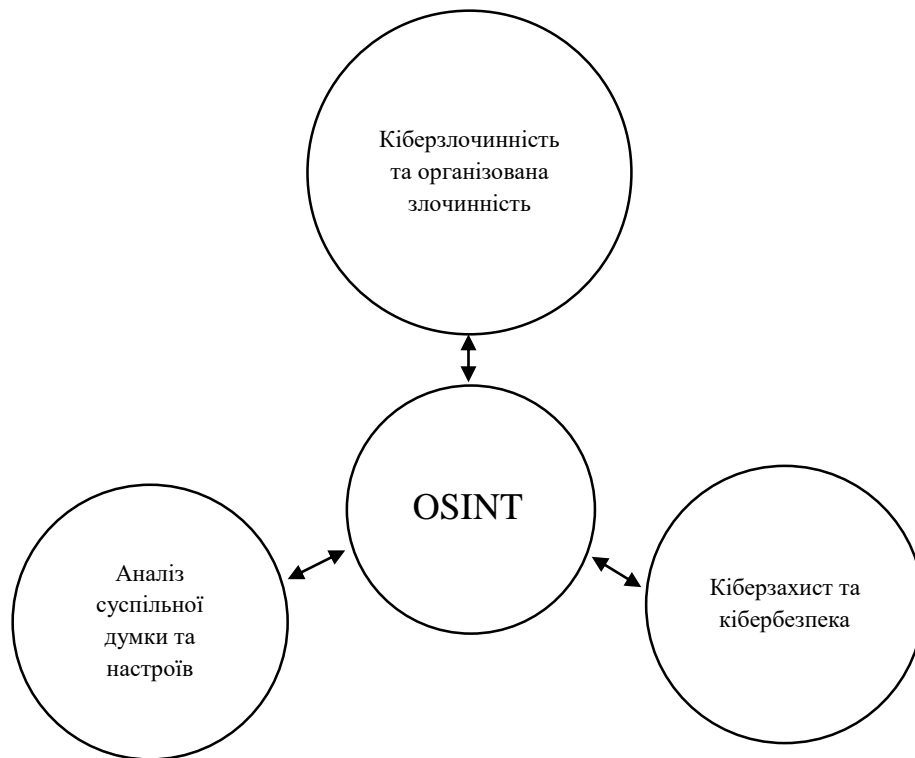


Рисунок 1.1 – Основні області застосування розвідки в відкритих джерелах інформації

Таким чином, розвідка відкритих джерел знайшла своє застосування в трьох основних напрямках, а саме: аналіз суспільної думки та настроїв, кіберзлочинність та організована злочинність, кіберзахист та кібербезпека.

Аналіз суспільної думки та настроїв полягає в тому, що завдяки буму соціальних мереж в Інтернеті можна збирати інформацію про взаємодії користувачів, їхні коментарі, інтереси та уподобання, щоб витягати певну нечітку інформацію. Інформація такого роду може бути застосована, наприклад, в задачах маркетингу, політичних кампаніях або ліквідації наслідків катастроф [5].

Використання в області кіберзлочинності та організованої злочинності полягає в тому, що відкриті джерела інформації постійно аналізуються з метою виявлення злочинних намірів на ранній стадії. Беручи до уваги певні закономірності дій злочинців, OSINT може надати правозахисним структурам можливість оперативно виявляти протиправні дії. Таким чином, використовуючи відкриті дані, можна в певній мірі відстежувати діяльність терористичних організацій, які все активніше діють в Інтернеті [6].

З кожним днем в житті людини все більшу роль відіграють смартфони, та інші портативні засоби, та ними користуються всі прошарки населення, через що відповідно зростає і кількість особистої інформації в інтернеті [1].

Оскільки злочинці постійно атакують інформаційно-комунікаційні системи з метою погіршення доступності послуг, а часом і з метою несанкціонованого доступу до конфіденційної інформації або ж пошкодження цілісності, то своє застосування розвідка відкритих джерел має і у областях кіберзахисту та кібербезпеки [7]. У цій області розвідка може застосовуватись не тільки для аналізу залишених після атаки слідів, а й з метою захисту систем. Завдяки аналізу щоденних атак та моніторингу поточного стану системи, розвідка може використовуватись для своєчасного виявлення атак, та надання можливості оперативної реакції на них [8].

Також, окрім згаданих вище випадків, OSINT можна застосувати і для інших задач. Зокрема, можна отримати відповідну інформацію, здійснюючи атаки соціальної інженерії. У цьому випадку, зловмисники використовують загальнодоступну інформацію, опубліковану в Інтернеті (наприклад, у соціальних мережах), щоб створити гачки для захоплення цілі. Крім того, можна проводити автоматичну оцінку достовірності відкритих даних з метою розкриття, зокрема, фейкових новин [9].

Таким чином, застосування OSINT поширюється на велику кількість сфер діяльності, а кількість програмних продуктів для автоматизованої розвідки невинно росте. Однак, при використанні цієї методології розробникам часто

доводиться йти на певні компроміси, хоча методологія і має ряд переваг, також є і певні обмеження, з якими доводиться мати справу.

До переваг можна віднести:

- великий обсяг доступної інформації;
- високу потужність обчислювальних машин;
- можливість застосування машинного навчання;
- гнучке призначення та широкий спектр застосування.

До недоліків же в свою чергу можна віднести наступне:

- складність управління великим обсягом даних;
- неструктурованість інформації;
- дезінформацію;
- надійність джерел даних;
- етичні та правові обмеження.

Отже, можна дійти до висновку, що OSINT - це перспективний механізм, який лише підсилює сфери кіберрозвідки, кіберзахисту та цифрової криміналістики. Потенціальний вплив, який ця методологія могла б мати на суспільство завдяки сучасним технологіям та великій кількості відкритих джерел досі використаний далеко не повністю, тому аналіз відкритих джерел інформації та існуючих продуктів для автоматизованої розвідки з використанням відкритих джерел надає змогу створення системи для автоматизованої розвідки, що продемонструє кращі результати роботи ніж існуючі аналоги.

1.2 Аналіз відкритих джерел інформації

Поняття open source intelligence (розвідка відкритих джерел, OSINT) сформувалось з поняття open source information (відкрите джерело інформації, OSIF). Розвідувальне співтовариство США визначило відкритою інформацією ту, що кожен може отримати законно через купівлю, спостереження, чи запит. При цьому, збір цієї інформації повинен відповідати вимогам захисту авторських прав [10].

В цілому, інформацію з відкритих джерел можна поділити на чотири категорії [10]:

- загальнодоступні дані та інформація;
- цільові комерційні дані;
- експертні висновки;
- «сіра» література.

До «сірої» літератури можна віднести письмову інформацію, що виробляється приватним сектором, урядом та освітніми установами. В цілому таку інформацію можна охарактеризувати як ту, наявність якої дещо обмежена через малу кількість випущених копій або встановлені обмеження доступу.

Наведені вище чотири категорії можуть включати в себе наступні джерела інформації [10]:

- медіа, наприклад газети, радіо, телебачення, інформація на електронних носіях;
- публічна інформація, така як демографічні дані, бюджети, слухання, прес-конференції, доповіді;
- інформацію, що поступила з професійних та академічних джерел, таких як конференції, симпозіуми, дисертації, тези, експертні думки;
- «сіра» література, до якої можна віднести доповіді з поїздок, неофіційні урядові документи, звіти про дослідження та опитування ринку.

Інформація стає “розвідувальною”, коли її збирають, експлуатують та розповсюджують у відповідний час відповідній аудиторії з метою виконання конкретних вимог.

Як згадувалося вище, одними з найбільш поширених та в той же час наймовірно потужними джерелами відкритої інформації можуть бути пошукові системи. Традиційне їх використання - найпростіший спосіб застосування OSINT.

Пошукові системи Google, Bing або Yahoo, є відомими та широко використовуваними інструментами. Ці пошукові системи дають змогу написати

текстовий запит, в свою чергу, намагаючись надати інформацію, яка відповідає введеним критеріям, працюють дуже ефективно і повертають користувачеві цінну інформацію. Інтерфейс пошукової системи Google зображено на рисунку 1.2.

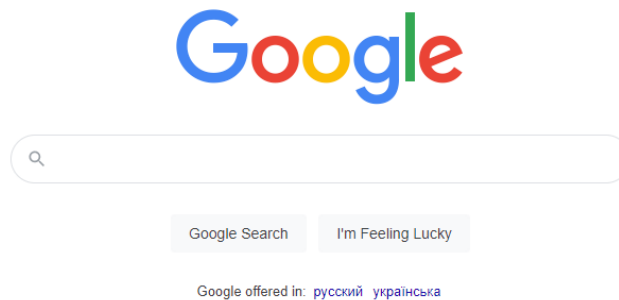


Рисунок 1.2 – Інтерфейс пошукової системи Google

Як видно з наведеного вище рисунку, інтерфейс пошукових систем, як правило, не передбачає багато опцій і весь функціонал зосереджується в написанні користувачем пошукового запиту. Виходячи з цього, кількість результатів може бути настільки великою та розмитою, що навіть може стати контрпродуктивною для користувача. З цієї причини користувачі повинні знати, як сформулювати запити в пошуковій системі, щоб досягти бажаного результату. Пошукові системи підтримують фільтри для уточнення пошуків та отримання саме тієї інформації, яка нас цікавить. Наприклад, використання “” дозволяє шукати точні збіги, AND діє як логічний оператор І, а OR діє як логічний оператор АБО [11]. Приклад використання одного з фільтрів зображено на рисунку 1.3.

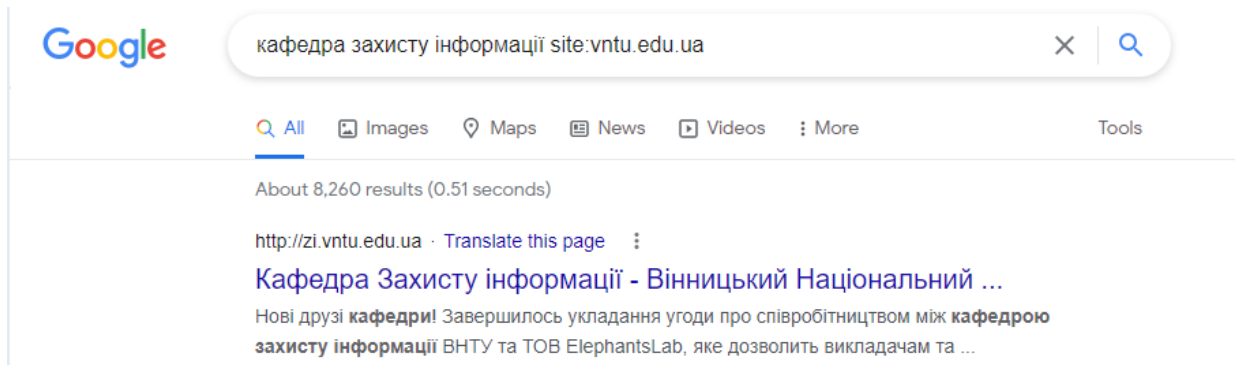


Рисунок 1.3 – Приклад використання пошукового фільтру

Звісно ж, на цьому перелік фільтрів не закінчується і, насправді, він значно більший. Деякі з пошукових фільтрів системи Google та приклади їх використання наведено в таблиці 1.1.

Таблиця 1.1 – Фільтри пошукової системи Google

Опис фільтру	Оператор	Приклад використання
Точний пошук слова чи фрази	«»	«Іванов Іван Іванович»
Виключити з результату слово чи фразу	-	Іванов Іван -Іванович
Шукати X або Y	OR	Вінницький OR Київський університет
Шукати X та Y	AND	Вінницький AND національний університет
Пошук по числу з послідовності	..	Новини 1995..1997
Пошук певного типу файлу	filetype:	ВНТУ filetype:pdf
Шукати в URL	inurl:	ВНТУ inurl:edu.ua

Таким чином, користуючись наведеними фільтрами можна вручну шукати відкриті джерела інформації. Але, можна скористатись і більш вузьконаправленими ресурсами для пошуку бажаної інформації. Наприклад, для пошуку відкритої інформації про певних осіб можна використовувати соціальні мережі, як, наприклад Facebook.

Facebook – це соціальна мережа, що була створена у 2004 році Марком Цукербергом під час його навчання в Гарварді. На сьогоднішній день Facebook –

одна з найбільших соціальних мереж, а кількість користувачів станом на 2021 рік перевищує 2.8 мільярда осіб [12].

Користуючись цим джерелом інформації можна здійснювати пошук людей, публікацій, відео, пошук по геолокації, як зображено на рисунку 1.4

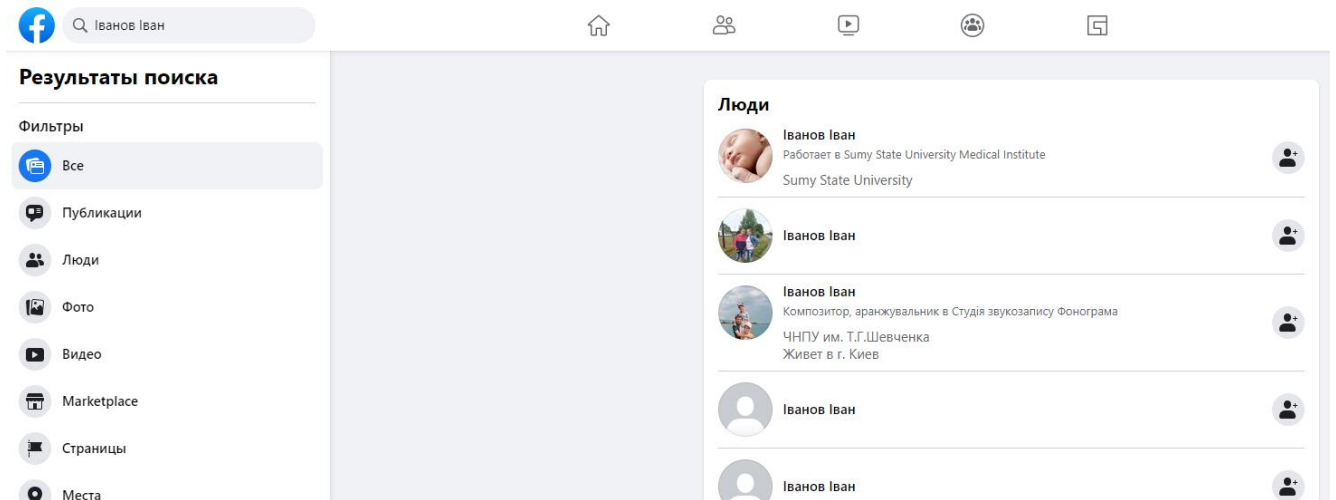


Рисунок 1.4 – Здійснення пошуку на Facebook

Але, одна соціальна мережа часто може покрити тільки певний діапазон користувачів, тому доцільно при розробці системи користуватись декількома відкритими джерелами інформації. Таким чином, було проаналізовано ряд соціальних мереж та ресурсів, а результат дослідження винесено в таблицю 1.2.

Таблиця 1.2 – Характеристика ряду соціальних мереж

Соціальна мережа	Призначення	Масштаб	Потенціальні дані для OSINT
Facebook	Встановлення соціальних контактів	Всесвітній	Персональний профіль, публікації, фото, відео
Flickr	Хостинг фото	Всесвітній	Хобі, місцезнаходження, родинні зв'язки
Instagram	Встановлення соціальних контактів	Всесвітній	Звички, місцезнаходження, родинні зв'язки
Linkedin	Бізнес	Всесвітній	Професійний профіль, освіта, навички
Reddit	Онлайн спільнота	Всесвітній	Публікації, поведінка, вподобання

Продовження таблиці 1.2

Соціальна мережа	Призначення	Масштаб	Потенціальні дані для OSINT
Tumblr	Хостинг фото	Всесвітній	Хобі, місцезнаходження, родинні зв'язки
Twitter	Встановлення соціальних контактів	Всесвітній	Персональний профіль, вподобання, публікації
Vkontakte	Встановлення соціальних контактів	Країни колишнього СНД	Персональний профіль, публікації, фото, відео
Youtube	Хостинг відео	Всесвітній	Відео, коментарі, вподобання

Також варто пам'ятати, що в зв'язку з тим, що інтернет-ресурси не статичні, а з часом як вони самі, так і їх наповнення можуть змінюватись або навіть видаляться, доцільно досліджувати не лише поточний стан даних, а й за можливості здійснювати аналіз історії даних та їх зміну.

Для досягнення цієї цілі можна скористатись інтернет-ресурсом Wayback Machine. Сервіс являється інтернет-архівом, що надає користувачам доступ до колекції збережених у різний час веб-сторінок. Розпочавши свою діяльність у 2001 році, на сьогоднішній день сервіс зберіг більше ніж 347 мільярдів копій сторінок різноманітних сайтів [13]. Як видно з рисунку 1.5, сервіс відображає історію знімків бажаної сторінки і надає можливість переглянути знімок за обрану дату.

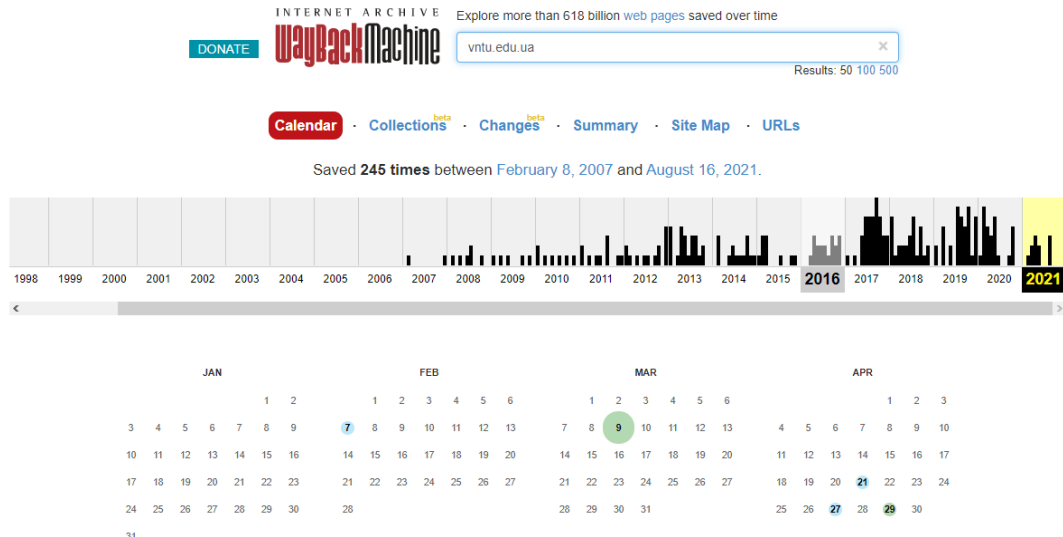


Рисунок 1.5 – Історія знімків веб-сторінки

Заархівовані на сервісі сторінки існують незалежно від їх першоджерела. В процесі архівування цифрові копії зберігаються на серверах самого інтернет архіву. В результаті цього, копії залишаються незмінними навіть у випадку, якщо адміністратори збережених сторінок закривають доступ або докорінно змінюють сайт [13].

Таким чином, було розглянуто у яких випадках інформація може вважатись розвідувальною, а також деякі міжнародні джерела інформації разом з прикладами їх ефективного використання.

1.3 Аналіз стану відкритих інформаційних джерел в Україні

Важливу роль у розвідці з відкритих джерел інформації грають джерела локальні для країни. Незважаючи на те, що свою частку вносять і міжнародні джерела, такі як Facebook, Youtube, та інші, не можна недооцінювати внесок локальних джерел.

Вперше поняття відкритих даних на державному рівні з'явилося у травні 2015 році, разом з ухваленням поправок до Закону України «Про доступ до публічної інформації» [14]. У законі було визначено, що розпорядники інформації зобов'язані надавати публічну інформацію у формі відкритих даних на запит, оприлюднювати і регулярно оновлювати її на єдиному державному веб-порталі відкритих даних та своїх веб-сайтах. Також важливо, що у законі

вказано, що публічна інформація у формі відкритих даних є дозволеною для її подальшого публічного використання [14].

Незабаром, 21 жовтня 2015 року Кабінетом Міністрів України було ухвалено постанову №835 «Про затвердження Положення про набори даних які підлягають оприлюдненню у формі відкритих даних» [15]. У Постанові було визначено вимоги до формату і структури наборів даних, що підлягають оприлюдненню, та перелік таких наборів даних [15]. Також було визначено ряд принципів, за якими здійснюється оприлюднення наборів даних у формі відкритих даних [15]:

- відкритість даних за замовчуванням;
- забезпечення своєчасного оновлення публічної інформації;
- доступність використання;
- інтероперабельність;
- постійна застосовність;
- інклюзивність розвитку.

З метою покращення доступу до відкритих даних та підвищення їх якості, починаючи з 2015 року Європейський портал даних щороку проводить дослідження Open Data Maturity Report, де за допомогою певних індикаторів здійснюється оцінка відкритих даних країни [16].

Україна, в свою чергу, доволі довгий час не брала участь у дослідженні, вперше надіславши звіт європейським колегам лише 10 липня 2020 року. Після оприлюднення результатів дослідження у грудні того ж року, можна зробити висновок, що дебют виявився доволі успішним. Деякі з результатів дослідження зображені на рисунку 1.6 [16].

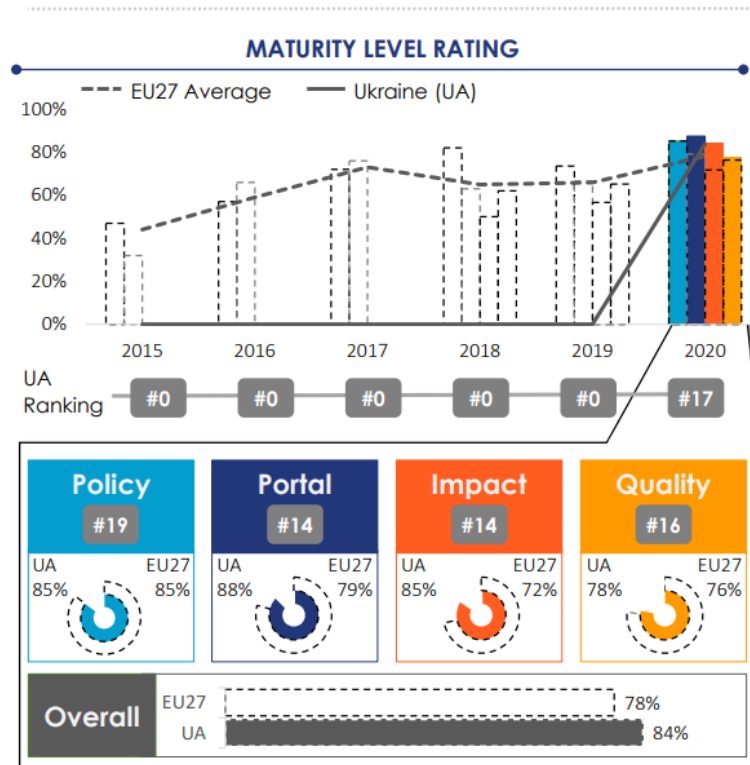


Рисунок 1.6 – Українські результати дослідження Open Data Maturity

Як видно з рисунку, оцінювання проводилось за чотирма основними критеріями:

- державна політика;
- державний портал даних;
- вплив відкритих даних на різні сфери;
- якість відкритих даних.

Результат дослідження державної політики відображає відповідність законодавства міжнародним стандартам, та становить 85%, що відповідає середньому показнику країн Європейського Союзу, та надає 19 місце за цим критерієм.

Наступним було проведено оцінювання державного порталу даних data.gov.ua, де країні було присвоєно результат у 88%, що на 9% випереджує середній результат по країнам Європейського Союзу.

Також був оцінений вплив відкритих даних на різні сфери, такі як соціальна, економічна, екологічна, та інші. Результат виявився на 12% вищим за середній по ЄС, та встановив 85%.

Наостанок, було проведено оцінювання якості відкритих даних, де Україна отримала близький до середнього по ЄС результат – 78%.

Наведені вище результати дозволили Україні зайняти 17 місце в рейтингу, що дало їй змогу увійти до переліку країн, що швидко розвиваються [16].

Одним з найбільших досліджень ринку відкритих даних України є «Зелена книга «Політика відкритих даних»» [17]. Автори дослідження провели надзвичайно масштабну роботу, що стосується багатьох аспектів стану вікритих джерел даних в Україні. Наприклад, було проаналізовано вчасність оновлення відкритих даних та виявлено, рисунок 1.7.

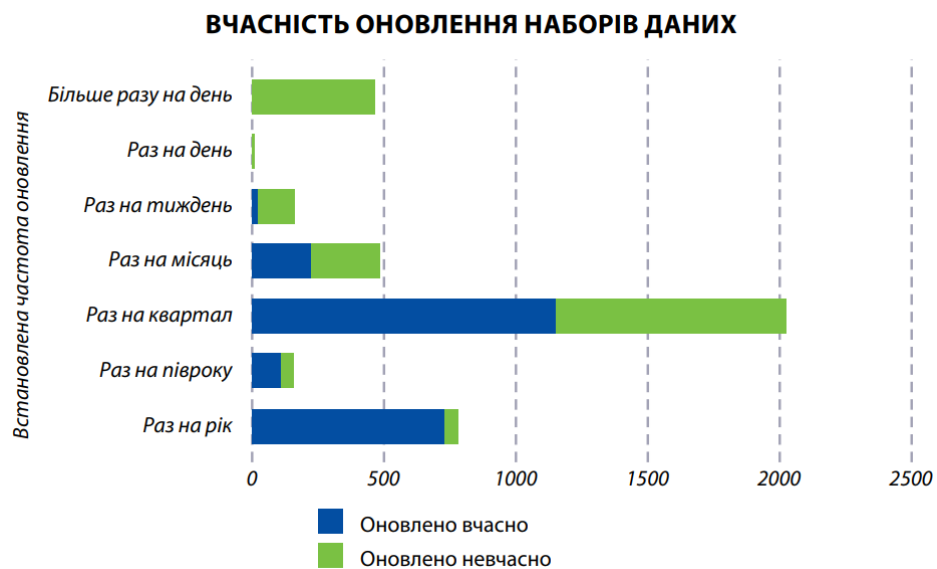


Рисунок 1.7 – Вчасність оновлення наборів даних

Результати дослідження показують, що існують певні проблеми зі своєчасністю оновлення даних. Близько 26% наборів не оновляються вчасно, у зв'язку з чим актуальність деякої інформації може залишатись під питанням. Також була виявлена і дещо протилежна проблема – деякі розпорядники оновлюють інформацію занадто часто, наприклад, при реальній необхідності оновлювати інформацію раз у квартал, було встановлено її оновлення раз у декілька тижнів [17].

Також авторами дослідження було виявлено певну кількість інших вад, наприклад обмеження доступу до інформації без оприлюднення підстав, порушення строків на оприлюднення наборів даних, низьку якість наданих даних

та ряд інших. Важливою рисою дослідження є не тільки виявлення вад, а й пропозиції для їх усунення.

Відповідно до постанови №835 було створено єдиний державний портал відкритих даних, data.gov.ua, про який вже згадувалось раніше. За допомогою ресурсу можна отримати доступ до наборів даних, наприклад, результатів ЗНО, статистичної інформації по різних структурах, інформації по земельним ділянкам, та іншу інформацію [18].

Прикладами такої інформації можуть слугувати:

- єдиний державний реєстр декларацій;
- єдиний державний реєстр юридичних осіб;
- єдиний державний реєстр фізичних осіб-підприємців;
- реєстр атестованих судових експертів.

Одним з найбільш відомих та легкодоступних ресурсів є єдиний державний реєстр декларацій. Це електронна система, що дозволяє вносити, перевіряти та оприлюднювати майнові декларації держслужбовців та інших осіб, що законом на це зобов'язані. Реєстр запрацював у серпні 2016 року і був створений на виконання закону «Про запобігання корупції». Однією з переваг використання такого роду джерел інформації є висока вірогідність того, що внесені дані є достовірними, на відміну від соціальних мереж, оскільки за введення недостовірних даних передбачена відповідальність. Використання даних з такої системи дозволяє значно розширити обсяг даних про особу. Приклад даних з реєстру зображено на рисунку 1.8.

Прізвище:	Порошенко
Ім'я:	Петро
По батькові (за наявності):	Олексійович
Податковий номер:	[Конфіденційна інформація]
Дата народження:	[Конфіденційна інформація]
Серія та номер паспорта громадянина України (ID-картка):	[Конфіденційна інформація]
Унікальний номер запису в Єдиному державному демографічному реєстрі:	[Конфіденційна інформація]
Місце роботи:	
Місце роботи або проходження служби (або місце майбутньої роботи чи проходження служби для кандидатів):	ВЕРХОВНА РАДА УКРАЇНИ
Займана посада (або посада, на яку претендує як кандидат):	НАРОДНИЙ ДЕПУТАТ УКРАЇНИ

Рисунок 1.8 – Приклад даних з реєстру декларацій

Використання цього джерела дає змогу дізнатись як особисту інформацію про суб'єкта декларування, так і деяку інформацію про членів його родини та, наприклад, об'єкти нерухомості та цінне рухоме майно.

Іншими прикладами масштабного локального джерела інформації можуть бути реєстри юридичних осіб та фізичних осіб-підприємців. Приклад даних з такого джерела зображено на рисунку 1.9.

Історія змін компанії

ПОВНА НАЗВА	СКОРОЧЕНА НАЗВА	СТАТУС	ПРОФІЛЬ КОМПАНІЇ	АДРЕСА	
01 жовтня 2021 р.— 20 вересня 2021 р.	ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО "ВІННИЦЬКИЙ МОЛОЧНИЙ ЗАВОД "РОШЕН"	ПРАТ "ВМЗ "РОШЕН"	зарєєстровано	10.51 Перероблення молока, виробництво масла та сиру	Україна, 21022, Вінницька обл., місто Вінниця, ВУЛИЦЯ ЕНЕРГЕТИЧНА, будинок 7
17 вересня 2021 р.	ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО "ВІННИЦЬКИЙ МОЛОЧНИЙ ЗАВОД "РОШЕН"	ПРАТ "ВМЗ "РОШЕН"	зарєєстровано	10.51 Перероблення молока, виробництво масла та сиру	Україна, 21022, Вінницька обл., місто Вінниця, ВУЛИЦЯ ЕНЕРГЕТИЧНА, будинок 7
15 вересня 2021 р.— 08 вересня 2021 р.	ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО "ВІННИЦЬКИЙ МОЛОЧНИЙ ЗАВОД "РОШЕН"	ПРАТ "ВМЗ "РОШЕН"	зарєєстровано	10.51 Перероблення молока, виробництво масла та сиру	Україна, 21022, Вінницька обл., місто Вінниця, ВУЛИЦЯ ЕНЕРГЕТИЧНА, будинок 7

Рисунок 1.9 – Приклад інформації з реєстру юридичних та фізичних осіб-підприємців

На рисунку зображено витяг по конкретному заводу, з використанням ресурсу ring.org.ua, що надає змогу поглибленого пошуку з врахуванням, наприклад, зміни назви вулиці. Ресурс використовує дані з відкритих державних реєстрів фізичних осіб-підприємців та юридичних осіб [19].

Одним з прикладів менш масштабного та більш вузьконаправленого джерела інформації може стати реєстр атестованих судових експертів. За допомогою цього джерела можна, окрім повного ім'я судового експерта дізнатись чи він фахівець державної установи, та орган у якому він працює.

До деяких з реєстрів є змога отримати доступ за допомогою спеціалізованого API, що робить роботу з ними більш комфортною, та дозволяє уникнути написання спеціальних парсерів.

Отже, після проведення аналізу відкритих інформаційних джерел в Україні, можна зробити висновок, що хоча буквально 7 років назад стан справ у цій області був доволі сумним, починаючи з 2015 року Україна стрімко розвивається у цій області та змогла увійти до переліку країн, рівень якості відкритих даних якої швидко розвивається. Незважаючи на певні існуючі недоліки, ведуться масштабні дослідження цієї області та пропонуються варіанти рішень існуючих вад.

1.4 Аналіз існуючих програмних засобів

Незважаючи на те, що пошук відкритих джерел вручну являється достатнім для досягнення базових результатів, використання всього одного чи двох ресурсів може бути неефективним для проведення більш складних розслідувань. Справжня ефективність OSINT розкривається за умови використання якомога більшої кількості джерел інформації, але проводити такі дії вручну, використовуючи методи для розвідки може бути доволі складно та часозатратно.

У зв'язку з цим, спільними зусиллями дослідників та розробників було створено програмні системи для автоматичного застосування технологій розвідки відкритих джерел та автоматизації збору інформації з різних джерел.

Результатом використання таких систем, як правило, є отримання більш цінної та детальної інформації.

Однією з найбільш відомих у своєму роді програм для автоматизованої розвідки є Maltego. Програма дає змогу шукати інформацію з різних джерел, таких як DNS записи, соціальні мережі, метадані файлів, та інші. Даний засіб використовує чотири оригінальні концепти [20]:

- сутності;
- трансформації;
- макроси трансформацій;
- хаб.

Сутність являється вузлом графа, що представляє собою деяку інформацію. Деякими стандартними сутностями є такі, як ім'я, адреса електронної пошти, профіль соціальної мережі, домен, IP-адреса, та інші.

Трансформації – це частинки коду, які можна застосувати до певної сутності. Програма постачається з невеликим набором безкоштовних трансформацій, при цьому залишається можливість придбання додаткових.

Хаб – це група трансформацій та сутностей. За замовчуванням, в Maltego вбудований хаб «PatervaCTAS», що містить сутності, трансформації та макроси трансформацій, що підтримуються офіційними розробниками.

Іншим популярним засобом для кіберрозвідки є Recon-NG. Користувачу він подається у вигляді командного рядка, у якому за допомогою команд можна обрати бажаний модуль. Після вибору модуля необхідно встановити параметри та запустити процес. Результати пошуків постійно зберігаються та використовуються у наступних раундах [21].

Цей засіб містить у собі певну кількість незалежних модулів, які імплементують різну функціональність. Прикладом може послугувати модуль Profiler, що шукає онлайн сервіси, де використовується таке ж ім'я користувача, як те, що було вказано при установці параметрів.

Оскільки під час роботи Recon-NG постійно накопичує у собі всю отриману інформацію та дає змогу її використовувати під час наступних викликів модулів, система гарно себе показує під час проведення складних розвідувальних дій [21].

Дещо більш вузько направленою системою є Shodan. Це спеціальна пошукова система, що надає публічно відому інформацію про підключені до Інтернету пристрої, в тому числі роутери, сервера, камери спостереження, та інше [22].

В основному, цей засіб використовується для забезпечення безпеки мережі шляхом виявлення пристроїв, що видно поза мережею організації, також за допомогою системи можна знаходити вразливості публічно доступних сервісів. Результати роботи можна завантажити на власній пристрій у різноманітних форматах.

Також система має декілька платних сервісів, а саме [22]:

- Shodan Maps;
- Shodan Images.

Shodan Maps надає змогу переглядати результати запитів візуально, з відмітками на карті, на відміну від основної версії, де результати подаються у текстовому вигляді. З технічної точки зору, Shodan Maps – це та сама версія Shodan, що і оригінальна, лише з тією відмінністю, що при створенні запиту автоматично додається фільтр по геолокації і результати відображаються лише по тому регіону, який у вас обрано на карті в поточний момент [22].

З виявлених недоліків можна зазначити практично повну відсутність унікальних для української території джерел інформації.

1.5 Постановка завдання

Отже, в розділі була проаналізована розвідка відкритих джерел інформації та визначено, що OSINT - це перспективний механізм, який лише підсилює сфери кіберрозвідки, кіберзахисту та цифрової криміналістики. Існуючі засоби автоматизованої розвідки на дозволяють забезпечити високий рівень якості

отримуваних даних, та не розділяють результати отримані з достовірних та неперевірених джерел.

Для того, щоб підвищити рівень отримуваних при розвідці результатів, необхідно розробити критерії оцінювання інформативності джерел перед їх додаванням в базу, та користуватись останніми нововведеннями, що стосуються відкритих даних у державі.

Для того, щоб зробити роботу спеціаліста з кіберзахисту більш ефективною, необхідно розробити критерії оцінки джерела за негативністю знаходження в ній.

Одними з найбільш результативних джерел відкритих даних є пошукові системи та соціальні мережі, тому було розглянуто використання пошукових мереж у якості джерела відкритих даних, проаналізовано основні фільтри пошукових мереж та здійснено розподіл соціальних мереж за рядом ознак, таких як призначення, та потенціальні дані для OSINT. Незважаючи на об'єм отримуваних даних, їх не можна з впевненістю вважати достовірними, тому їх буде включено до неперевірених джерел даних.

Після проведення аналізу програмних засобів було виявлено, що незважаючи на широке застосування міжнародних соціальних мереж та ресурсів у якості інформаційних джерел, засоби часто зовсім не використовують відкриті реєстри України, що є суттєвим недоліком при формуванні результату розвідки.

Здійснення розвідки з відкритих джерел – надзвичайно актуальна на сьогоднішній день задача в Україні у зв'язку з широким можливим спектром застосувань системи, починаючи від аналізу суспільної думки і закінчуючи протидією кібератакам.

Зважаючи на згадані вище фактори, було визначено необхідність створення системи для розвідки відкритих джерел у форматі веб-додатку у зв'язку з відсутністю відомих ефективних програмних рішень.

Для того, щоб підвищити ефективність роботи системи, необхідно в першу чергу розробити її архітектуру, опираючись на існуючі сучасні системи для розвідки з відкритих джерел.

Перевага такого роду системи в повній мірі розкриється лише за умови використання доцільних джерел інформації та належного їх аналізу. Тому далі необхідно розробити алгоритм для збору інформації з відкритих джерел, та аналізувати зібрану інформацію.

На основі розробленої архітектури та алгоритмів необхідно реалізувати результат у вигляді програмного рішення, що відповідатиме наступним вимогам:

- поставлено критерії оцінки негативності джерел;
- поставлено критерії оцінки якості та достовірності джерел;
- поставлено критерії оцінки інформативності джерел;
- сформована база на основі відкритих реєстрів даних.

На основі сформованих вимог до системи розвідки з відкритих джерел інформації розроблено технічне завдання.

2 РОЗРОБКА АЛГОРИТМУ РОЗВІДКИ ТА СТРУКТУРИ СИСТЕМИ

2.1 Розробка критеріїв оцінки інформаційних джерел

Оскільки для здійснення розвідки з ціллю запобігання атак у кіберпросторі необхідна інформація з відкритих джерел даних, необхідно створити для них критерії оцінки з метою покращення результатів пошуку інформації, та збільшення її якісних показників, що в результаті збільшить ефективність використання системи спеціалістами з кібербезпеки.

Те, що книга, стаття чи веб-сайт відповідають критеріям пошуку, не означає, що вони обов'язково є надійним та достовірним джерелом інформації.

Загальні критерії попередньої оцінки джерела включають: ціль, точність і надійність, актуальність і своєчасність, а також об'єктивність. Отож, доцільно розібрати кожен з критеріїв та виділити необхідні для формування якісного списку джерел.

Інформація може мати наступні цілі:

- надання інформації (наприклад, наукові статті, реєстри);
- розвага (наприклад, вірусне відео);
- продаж продукту, або послуги (наприклад, рекламні або маркетингові матеріали на веб-сайті компанії).

Для інформації, яку можна було б ефективно використовувати для здійснення розвідки, найбільше підходять джерела, які створені для надання інформації. Це можна обґрунтувати тим, що збір достовірної інформації полягає саме в зборі неупередженої інформації з ресурсів, а інші варіанти не підходять під цей критерій.

Що стосується точності та надійності, можна виділити наступні вимоги до джерела:

- Чи є посилання (наприклад, цитати, виноски чи бібліографія) на джерела, які надають докази зроблених тверджень?

- Якщо джерело містить факти або статистичні дані, чи можна перевірити цю інформацію в іншому джерелі?
- Якщо дані були зібрані за допомогою оригінальних досліджень (наприклад, опитування), яким був метод збору даних? Чи може автор довести достовірність чи надійність даних?

Виконання джерелом вимог до актуальності та своєчасності полягає у наступному:

- Для книг і статей - повинна бути можливість легко перевірити дату публікації.
- Для веб-сайтів потрібно мати змогу визначити дату створення або оновлення веб-сторінки.

У цьому критерії більше стоїть питання у тому, чи потрібна саме актуальна інформація. Якщо такої вимоги немає, інформація може бути застарілою, але при цьому вона повинна мати дані, що стосуються дати її публікації або останнього оновлення.

Об'єктивність:

- Джерело містить думки, чи факти?
- Чи інформація, представлена у джерелі, об'єктивна (неупереджена) чи суб'єктивна (упереджена)?
- Чи впливає інформація на політичний, релігійний чи соціальний стан?
- Чи є чітким виділення рекламного вмісту (зазвичай його можна знайти в ділових журналах чи газетах)?

Таким чином, було проведено аналіз відкритих реєстрів даних за визначеними критеріями. Результати аналізу зображено в таблиці 2.1.

Таблиця 2.1 – Результати аналізу якості реєстрів, як джерела даних

Назва критерію	Висновок
Ціль існування	Державний реєстр ведеться уповноваженим органом держави з метою накопичення, обробки інформації та надання певним відомостям офіційного визнання.
Точність та надійність	Якщо не доведено іншого, інформація в державному реєстрі вважається точною та правдивою, а отже надійною
Актуальність та своєчасність	Забезпечується своєчасність оновлення інформації
Об'єктивність або упередженість	Інформація, розміщена у джерелі, є об'єктивною та містить факти

Таким чином, було визначено, що відкриті державні реєстри даних є надійним та достовірним джерелом інформації.

У якості базового масиву достовірних джерел було виділено ряд реєстрів, зображених на таблиці 2.2.

Таблиця 2.2 – Масив реєстрів

Назва реєстру	Потенціальні дані для OSINT
Єдиний державний реєстр боржників	Категорія боргу, публікатор боргу.
Реєстр осіб, які вчинили корупційні правопорушення	Покарання, місце роботи, займана посада
Особи, що переховуються від СБУ, МВС, МОУ	Категорія (орган, від якого переховується), дата народження, стаття
Реєстр свідоцтв про реєстрацію авторського права на твір	Анотація роботи, співавтори, дата публікації, тип роботи
Єдиний державний реєстр осіб, щодо яких застосовано положення Закону «Про очищення влади»	Місце роботи, стаття, термін ув'язнення
Список справ призначених до розгляду	Дата розгляду, судді, справа (включає імена позивача та доповідача), випадок, описок випадку
Інформація з автоматизованої системи виконавчого провадження	Назва кредитора, дата вступу в дію виконавчого провадження, назва організації

Продовження таблиці 2.2

Назва реєстру	Потенціальні дані для OSINT
Єдиний реєстр арбітражних керуючих України	Номер сертифікату, статус сертифікату, дата отримання сертифікату
Державний реєстр атестованих судових експертів	Назва організації, адреса організації, телефон організації, тип експертизи, спеціалізація експерта
Безвісно зниклі	Дата народження, дата зникнення, місце зникнення, стаття
Єдиний державний реєстр юридичних та фізичних осіб-підприємців	Номер телефону, адреса електронної пошти, місце проживання, види діяльності
Перелік об'єктів права інтелектуальної власності, включених до митного реєстру	Дата та срок реєстрації, назва та адреса суб'єкта права інтелектуальної власності

Далі, кожен з реєстрів було проаналізовано за наступними критеріями, та надано оцінку:

- інформативності персональних даних в джерелі;
- чи є факт наявності персони в реєстрі таким, що негативно її характеризує.

Персональні дані – це будь-яка інформація, яку хтось може використати для ідентифікації людини [23]. Загалом, дані можна поділити на 4 категорії [24]:

- анонімні дані;
- деідентифіковані дані;
- псевдонімні дані;
- дані, що можна ідентифікувати.

Анонімними вважаються такі дані, з яких були вилучені прямі та непрямі ідентифікатори, або були оброблені таким чином, що гарантує неможливість реідентифікації. Прикладом таких даних можуть бути статистичні [24].

Деідентифіковані дані – це такі, з яких ідентифікатори були вилучені, або оброблені, щоб обірвати зв'язок з сутностями. Наприклад, таблиця, в якій вказано середній бал учня та їхню стать. Аналогічно до анонімних даних, як

прямі, так і непрямі ідентифікатори були трансформовані або повністю вилучені [24].

Псевдонімні дані – це такі, з яких було вилучено або трансформовано прямі ідентифікатори, але непрямі ідентифікатори залишились. Така інформація може міститись в датасетах, в яких ім'я того, з ким пов'язана інформація, певним чином зашифроване, або доступ до ідентифікатора має тільки куратор [24].

Ідентифіковані дані – такі, що містять прямі та непрямі ідентифікатори [24]. Оскільки всі дані, що містяться в опрацьованих реєстрах містять ім'я та прізвище людини, то їх можна вважати ідентифікованими даними.

Чутливі дані – це певний набір даних про людину, які необхідно обробляти з додатковим захистом. Це включає наступну інформацію [25]:

- політичні вподобання;
- релігійні або філософські вірування;
- расове або етнічне коріння;
- генетичні дані;
- біометричні дані;
- дані про стан здоров'я.

Після аналізу даних, що зберігаються в оброблених реєстрах було визначено, що жоден з них не містить інформації, яку можна було б вважати чутливою, проте майже кожен містить деяку персональну інформацію.

Таким чином, реєстри будуть оцінюватись за фактом наявності в них прямих та непрямих ідентифікаторів (не включаючи імені), де кожен прямий ідентифікатор буде оцінено в три бали, а кожен непрямий – в один бал. Результати оцінки занесено в таблицю 2.3.

Таблиця 2.3 – Оцінка виду ідентифікатора персональних даних в реєстрах

Персональні дані	Вид (прямий/непрямий ідентифікатор)
Місце роботи	Непрямий
Займана посада	Непрямий
Дата народження	Непрямий
Точне місце проживання (адреса прописки)	Прямий

Продовження таблиці 2.3

Персональні дані	Вид (прямий/непрямий ідентифікатор)
Регіон проживання	Непрямий
Пов'язані особи	Непрямий
Дані документів, пов'язаних з особою	Непрямий
Організації, пов'язані з особою	Непрямий
Електронна пошта	Прямий
Номер телефону	Прямий

Після цього, для кожного з реєстрів було визначено максимально можливий бал персональних даних, який він може отримати за умови, що всі його поля заповнені. Результат оцінки зображено на таблиці 2.4.

Таблиця 2.4 – Оцінка інформативності персональних даних в реєстрах

Назва реєстру	Дані, що ідентифікують людину	Оцінка
Єдиний державний реєстр боржників	публікатор боргу	1
Реєстр осіб, які вчинили корупційні правопорушення	місце роботи, займана посада	4
Особи, що переховуються від сбу, мвс, мо	орган, від якого переховується, дата народження	2
Реєстр свідоцтв про реєстрацію авторського права на твір	співавтори	1
Єдиний державний реєстр осіб, щодо яких застосовано положення Закону «Про очищення влади»	Місце роботи	3
Список справ призначених до розгляду	судді, справа (включає імена позивача та відповідача)	2
Інформація з автоматизованої системи виконавчого провадження	Назва кредитора, назва організації	2
Єдиний реєстр арбітражних керуючих України	Номер сертифікату	1
Державний реєстр атестованих судових експертів	Назва організації, адреса організації, телефон організації	3
Безвісно зниклі	Дата народження, місце зникнення	2
Єдиний державний реєстр юридичних та фізичних осіб-підприємців	Номер телефону, адреса електронної пошти, місце проживання	9
Перелік об'єктів права інтелектуальної власності, включених до митного реєстру	назва та адреса суб'єкта права інтелектуальної власності	4

Отже, найбільшу можливу оцінку має реєстр юридичних та фізичних осіб-підприємців, оскільки містить одразу три прямих ідентифікатори.

Далі було проаналізовано, наявність в яких реєстрах може негативно характеризувати людину. Було виділено три рівні, для кожного з яких було призначено окремий колір, за допомогою якого можна його ідентифікувати, а саме, червоний, жовтий та сірий.

Факторами, через які інформацію в джерелі можна віднести до червоного рівня було обрано наступні:

- наявність історії правопорушень;
- наявність боргів.

Отже, до червоного рівня можна віднести реєстри, наявність людини в яких безумовно негативно характеризує людину. Такими є:

- реєстр осіб, що переходять від СБУ, МС та МО;
- реєстр осіб, щодо яких застосовано положення Закону «Про очищення влади»;
- реєстр осіб, які вчинили корупційні правопорушення;
- єдиний державний реєстр боржників;
- інформація з автоматизованої системи виконавчого провадження.

До реєстрів жовтого рівня можна віднести такі, що потребують самостійного аналізу від користувача системи, оскільки наявність запису в них можна віднести як до червоного рівня, так і до сірого. Прикладами таких реєстрів є:

- реєстр безвісно зниклих;
- реєстр списку справ призначених до розгляду.

До реєстрів сірого рівня можна віднести решту реєстрів, оскільки наявність запису в них, як правило, не характеризує особу, що в ній знаходиться, з негативної сторони.

Таким чином, було створено критерії оцінювання джерела за якістю та достовірністю, інформативністю персональних даних та негативністю наявності запису в ньому.

Отже, розроблено критерії оцінювання інформаційних джерел за надійністю та достовірністю, інформативністю, та негативністю знаходження в джерелі, що в результаті збільшує ефективність роботи спеціалістів з кіберзахисту. Після вибору початкового масиву джерел, доцільно перейти до розробки алгоритму розвідки.

2.2 Розробка алгоритму розвідки

Оскільки поняття розвідки є широким і використовується у багатьох сферах діяльності, доцільно спочатку визначити, якою є ціль розвідки у контексті дипломної роботи.

Організація може займатись розвідкою людей з різних причин, включаючи виявлення злочинних об'єднань або злочинного походження майбутнього працівника. Кіберзлочинці можуть бути чудовими соціальними інженерами, завдяки чому їм легко проникнути в організації, щоб вкрати дані або виконувати інші дії злочинного характеру. Використання людської розвідки може допомогти зменшити цей ризик, оскільки під час цих розслідувань часто з'являються злочинні зв'язки, або підозріле минуле.

Крім того, людина може просто не мати належних навичок кібергігієни. Розвідка, як приклад, може виявити надмірне поширення конфіденційної інформації. Наївний співробітник стає легкою мішенню зловмисного програмного забезпечення та фішингових кампаній, які можуть завдати величезної шкоди організації. Згідно з даними RedefinePrivacy, лише у 2020 році 66% компаній зазнали фішингових атак [26].

Тому впровадження людської розвідки в організації може допомогти оцінити ризики та збитки. Крім того, це може допомогти організації уникнути шкоди репутації, яка супроводжується кіберінцидентом через зловмисних або не зловмисних інсайдерів.

У контексті магістерської роботи, під розвідкою розуміється збір даних про певну людину з метою отримати якомога більше інформації про неї. Інформація такого типу буде добуватися з декількох джерел, основними з яких будуть відкриті державні реєстри, обрані та опрацьовані у попередньому підрозділі.

Основними складовими, що характеризують людину, можна назвати наступні:

- ПІБ;
- попередні місця роботи;
- наявність боргів;
- наявність правопорушень;
- наявність наукових робіт;
- номер телефону;
- адреса електронної пошти.

Розвідка відбуватиметься за алгоритмом, зображеним на рисунку 2.1.



Рисунок 2.1 – Алгоритм розвідки

Таким чином, після введення початкових даних буде здійснюватися розвідка в обраних достовірних базах (наприклад, ЄДР фізичних та юридичних осіб підприємців [27], реєстр боржників [28], та інші), після чого сформований результат обробляється клієнтським додатком.

Оскільки немає якоїсь однієї норми щодо того, в якому форматі зберігаються реєстри, для багатьох з них виникає необхідність писати окремі парсери в базу даних, оскільки читання з XML, XLS та CSV файлів є не продуктивним порівняно з читанням з бази даних. Приклад алгоритму парсера XML файлу зображено на рисунку 2.2.

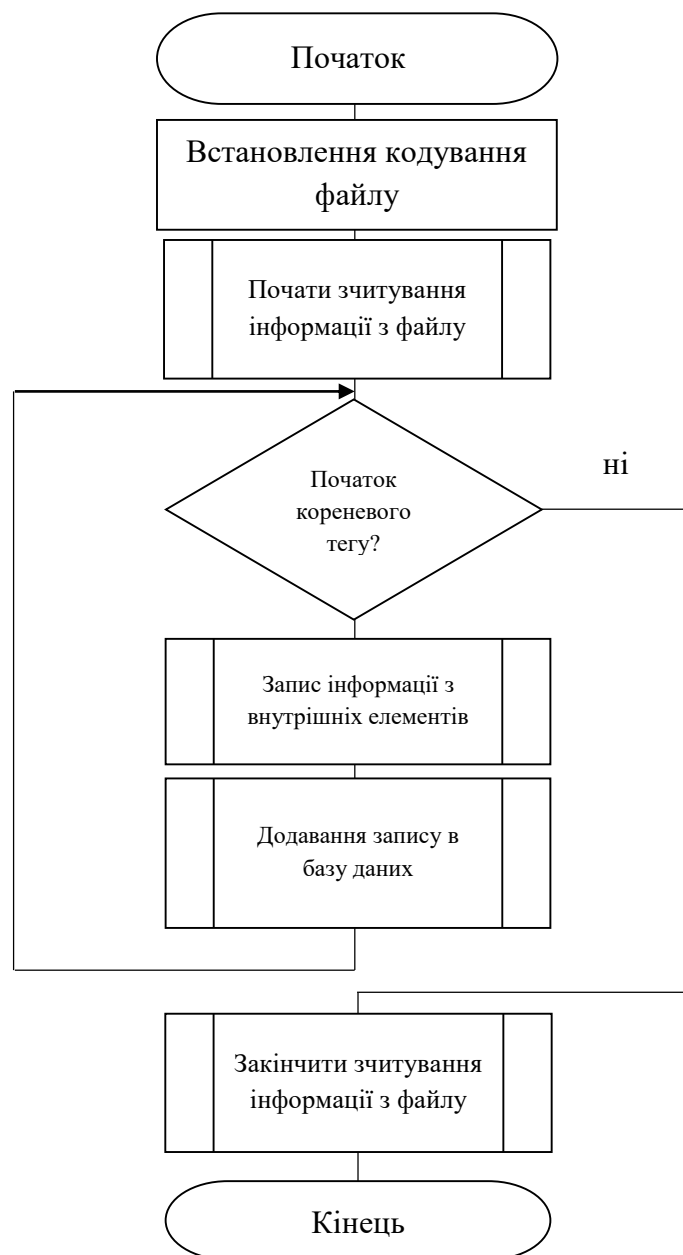


Рисунок 2.2 – Алгоритм парсингу

Таким чином, початковим етапом алгоритму парсингу є визначення кодування файлу реєстру та встановлення цього значення. Наприклад, кодування файлу може windows-1254, windows-1253, utf-8 або інше, за бажанням розробників реєстру. Після цього відбувається відкриття файлу та його парсинг його значень в базу даних, доки в файлі містяться кореневі елементи. Після закінчення робота з файлом зупиняється, та алгоритм на цьому завершується.

Приклад запису в реєстрі фізичних та юридичних осіб підприємців зображено на рисунку 2.3.

```
<DATA FORMAT_VERSION="1.0">
<SUBJECT><RECORD>11538737</RECORD><NAME>МАКІВСЬКА СІЛЬСЬКА РАДА МАКІВСЬКОЇ СІЛЬСЬКОЇ ОБ'ЄДИНАНОЇ ТЕРИТОРІАЛЬНОЇ ГРОМАДИ</NAME><SHORT_NAME>МАКІВСЬКА СІЛЬСЬКА РАДА</SHORT_NAME><OFF>ОРГАН МІСЦЕВОГО САМОВРЯДУВАННЯ</OFF><EDRPOU>40134461</EDRPOU><ADDRESS>Україна, 32445, Хмельницька обл., Дунаєвецький р-н, село Маків, ВУЛИЦЯ ВОЗДВИЖЕНСЬКА, будинок 22</ADDRESS><STAN>зарєєстровано</STAN><FOUNDING_DOCUMENT_NUM/><EXECUTIVE_POWER/><FOUNDERS/><BENEFICIARIES/><ACTIVITY_KINDS><ACTIVITY_KIND><CODE>84.11</CODE><NAME>Державне управління загального характеру</NAME><PRIMARY>так</PRIMARY></ACTIVITY_KIND></ACTIVITY_KINDS><SUPERIOR_MANAGEMENT/><SIGNERS><SIGNER>Галачинський Владислав Станіславович; сільський голова; - керівник</SIGNER></SIGNERS><AUTHORIZE_D_CAPITAL/><STATUTE>Діє на підставі установчих документів, затверджених засновниками (учасниками)</STATUTE><REGISTRATION>23.11.2015; 23.11.2015; 16571020000001292</REGISTRATION><MANAGING_PAPER/><BRANCHES/><TERMINATION_STARTED_INFO/><BANKRUPTCY_READJUSTMENT_INFO/><PREDECESSORS><PREDECESSOR><NAME>МАКІВСЬКА СІЛЬСЬКА РАДА</NAME><CODE>04406466</CODE></PREDECESSOR><PREDECESSOR><NAME>МИХАЙЛІВСЬКА СІЛЬСЬКА РАДА</NAME><CODE>04406495</CODE></PREDECESSOR><PREDECESSOR><NAME>ЧЕЧЕЛЬНИЦЬКА СІЛЬСЬКА РАДА</NAME><CODE>21313312</CODE></PREDECESSOR></PREDECESSORS><ASSIGNEES/><TERMINATED_INFO/><TERMINATION_CANCEL_INFO/><CONTACTS>+380673800928</CONTACTS><EXCHANGE_DATA><EXCHANGE_ANSWER><AUTHORITY_NAME>ДЕРЖАВНА СЛУЖБА СТАТИСТИКИ УКРАЇНИ</AUTHORITY_NAME><AUTHORITY_CODE>37507880</AUTHORITY_CODE><START_DATE>24.11.2015</START_DATE></EXCHANGE_ANSWER><EXCHANGE_ANSWER><AUTHORITY_NAME>ГОЛОВНЕ УПРАВЛІННЯ ДПС У ХМЕЛЬНИЦЬКІЙ ОБЛАСТІ, ДУНАЄВЕЦЬКА ДПІ (ДУНАЄВЕЦЬКИЙ РАЙОН)</AUTHORITY_NAME><AUTHORITY_CODE>44070171</AUTHORITY_CODE><TAX_PAYER_TYPE>Реєстр платників податків</TAX_PAYER_TYPE><START_DATE>23.11.2015</START_DATE><START_NUM>1687</START_NUM></EXCHANGE_ANSWER><EXCHANGE_ANSWER><AUTHORITY_NAME>ГОЛОВНЕ УПРАВЛІННЯ ДПС У ХМЕЛЬНИЦЬКІЙ ОБЛАСТІ, ДУНАЄВЕЦЬКА ДПІ (ДУНАЄВЕЦЬКИЙ РАЙОН)</AUTHORITY_NAME><AUTHORITY_CODE>44070171</AUTHORITY_CODE><TAX_PAYER_TYPE>Реєстр платників єдиного внеску</TAX_PAYER_TYPE><START_DATE>23.11.2015</START_DATE><START_NUM>10000000516550</START_NUM></EXCHANGE_ANSWER></EXCHANGE_DATA><VP_DATES/><CURRENT_AUTHORITY>Дунаєвецька районна державна адміністрація Хмельницької області</CURRENT_AUTHORITY></SUBJECT>
```

Рисунок 2.3 – Приклад запису в реєстрі фізичних та юридичних осіб

Як видно з рисунку, одними з основних полів реєстру фізичних та юридичних осіб підприємців є наступні:

- номер запису;
- повна та коротка назва суб'єкту;
- код ЄДРПОУ;
- місце знаходження;
- адреса;
- дані про керівника;

Наступним кроком відбуватиметься пошук в більш вузьконаправлених, але все ще достовірних базах даних, таких як реєстр осіб, які переходять від органів влади [29], єдиний держаний реєстр осіб, які вчинили корупційні правопорушення [30], та інші.

Приклад запису в реєстрі осіб, які вчинили корупційні правопорушення зображено на рисунку 2.4.

```

<RECORD>
  <REG_NUM>46605</REG_NUM>
  <REG_DATE>2012-08-17</REG_DATE>
  <PUNISHMENT>На підставі ст.70 КК України за сукупністю злочинів шляхом поглинення менш суворого покарання більш суворим остаточно визначити Бондаренко Тетяні Федорівні покарання у виді 3 років позбавлення волі з позбавленням права обіймати посади, пов'язані із виконанням організаційно-розпорядчих та адміністративно-господарських функцій на строк 3 роки. На підставі ст.75 КК України звільнити засуджену Бондаренко Т.Ф. від відбування призначеного основного покарання у виді 3 років позбавлення волі з випробуванням терміном на 2 роки.</PUNISHMENT>
  <FIO>Бондаренко Тетяна Федорівна</FIO>
  <JOBPLACE>магазин №0302 «Маркет»</JOBPLACE>
  <JOBPOST>завідуюч</JOBPOST>
  - <CODEX_ST>
    <CODEXARTICLE>ККУ, Стаття 366. Службове підроблення</CODEXARTICLE>
    <CODEXARTICLE>ККУ, Частина друга статті 191 Привласнення, розтрата майна або заволодіння ним шляхом зловживання службовим становищем</CODEXARTICLE>
  </CODEX_ST>
  <JUDGMENTDATE>2012-03-20</JUDGMENTDATE>
  <JUDGMENTNUMBER/>
</RECORD>
-----

```

Рисунок 2.4 – Приклад інформації з реєстру осіб, які вчинили корупційні правопорушення

Як видно з рисунку, у випадку реєстру осіб, що вчинили корупційні правопорушення у якості бази даних виступає файл XML формату, основними полями якого є:

- реєстраційний номер;
- дата реєстрації;
- покарання;
- ПІБ;
- місце роботи;
- займана посада;
- масив статей, які інкримінуються особі;
- дата суду.

Таким чином, отримавши при первинному пошуку інформації, наприклад, членів сім'ї особи над якою проводиться розвідка, можна отримати інформацію про причетність його рідних до корупційних правопорушень, та іншу інформацію.

Після цього, на основі здобутих раніше відомостей про індивіда, відбудеться побудова спеціальних пошукових запитів, на основі отриманих на попередніх етапах інформації. Деякі з таких запитів зображено в таблиці 2.4.

Таблиця 2.5 – Приклади пошукових запитів

Формат пошукового запиту	Потенціальні дані для OSINT
«(ПІБ)» + «(Населений пункт)», або «(П)» + «(Населений пункт)»	Сторінки особи в соціальних мережах, місцеві новини з участю особи та інше.
П + «(Номер телефону)»	Соціальні мережі де вказано прізвище та номер телефону особи, згадування на інших сайтах
«(Номер телефону)»	Усі сайти, де вказаний номер телефону особи
«(Адреса електронної пошти)»	Усі сайти, де вказана електронна пошта особи
«П» + (місце роботи)	Пошук інформації
«П» + site:facebook.com	Пошук облікового запису у соц. мережі facebook
«П» + site:vk.com	Пошук облікового запису у соц. мережі vk
«П» + site:youtube.com	Пошук відеозаписів у YouTube, де згадується ім'я особи

Після проходження всіх згаданих вище етапів буде здійснено формування результату та його виведення.

Таким чином, було розроблено алгоритм розвідки на основі відкритих державою наборів даних та сформульованих за їх результатом пошукових запитів.

2.3 Вибір програмних технологій засобу

Для створення додатку було обрано мову програмування C#, та її фреймворк Asp.NET.

Мова програмування C# відноситься до сім'ї мов з синтаксисом, схожим на синтаксис мови C, тобто є близькою до мов програмування C++ та Java. Формально, .NET це відкрита платформа для розробки, створена Microsoft для побудових різних типів додатків [31].

Це високорівнева мова програмування, що має велику підтримку як від спільноти її розробників, так і користувачів, тому щорічно виходять оновлення, у яких мові додаються нові можливості та усуваються попередні недоліки.

C# — це компільована мова, що означає, що код, що зберігається на сервері, має двійкову форму [32]. Якщо ваш сервер зламано, хакер не отримає автоматично доступу до вашого вихідного коду. З іншими поширеними мовами, такими як PHP, зловмисник отримує доступ до вихідного коду, що може погано обернутись з точки зору безпеки. З C# хакер повинен декомпілювати або «зламати» ваше програмне забезпечення, перш ніж він зможе побачити критичні компоненти.

До основних переваг мови можна віднести:

- висока швидкість;
- об'єктно-орієнтованість;
- безпечність типів;
- масштабованість і оновлюваність;
- багата бібліотека;
- компонентно-орієнтованість.

В якості інтегрованого середовища розробки було вирішено використовувати JetBrains Rider. Однією з переваг цього середовища є наявність за замовчуванням засобу ReSharper, що забезпечує безперервний аналіз якості коду C# за допомогою більш ніж 2200 інспекцій, серед яких можна зазначити виявлення, як базових, так і більш важких проблем:

- відсутність значення, що повертається, або відсутність такого значення у деяких з умовних гілок;
- некоректного форматування рядків;
- невикористовувані локальні змінні;
- незакриті поля, до яких не здійснюється доступ;
- членів і типів, які можна зробити внутрішніми, а не загальнодоступними.

2.4 Розробка структури програмної системи

При розробці системи, вона має задовольняти наступним вимогам:

- розширюваність;
- використання реляційної бази даних;
- можливість створення фронтенду

Оскільки для створення додатку було обрано мову програмування C#, можливий вибір з таких платформ, як вебсайт, телеграм бот, та десктопний додаток. Незважаючи на те, що кожна з платформ задовільняє поставленим до системи вимогам, для реалізації додатку було обрано саме формат вебсайту, оскільки десктопні додатки не є мультиплатформенними, а створення телеграм бота, при ідентичному у даному випадку функціоналі, є незручним зі сторони користувацького інтерфейсу.

Отже, система складається з трьох елементів – бази даних, сервера web api, та клієнта Angular app, який відповідає за фронтенд. Буде бекенд ASP.NET Core, за допомогою якого буде створено API. Цей API дозволить користувачам виконувати операції CRUD. Отримані дані зберігатимуться в базі даних на сервері. API може дозволити нам обслуговувати всі дані, які були попередньо додані в базу даних. У якості фронтенду ми будемо використовувати фреймворк Angular. За допомогою Angular ми можемо створювати служби та компоненти, щоб робити запити до бекенда, щоб опублікувати нові дані та отримати всі наявні дані. Архітектура системи зображена на рисунку 2.5.

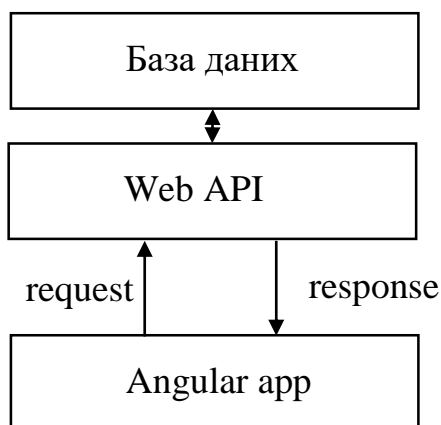


Рисунок 2.5 – Архітектура системи

Модульна структура програми зображена на рисунку 2.6.

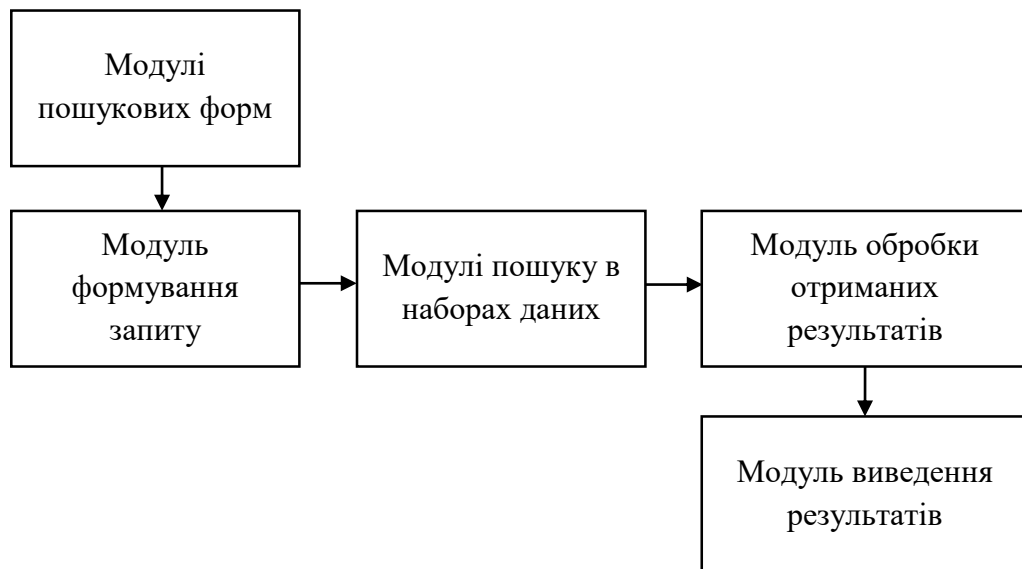


Рисунок 2.6 – Модульна структура програми

Модулі пошукових форм відповідають за форми введення пошукових даних користувацького інтерфейсу та здійснюють валідацію введених даних

Модуль формування запиту приймає вхідні дані та формує API запит до серверу.

Модулі пошуку в наборах даних відповідають за пошук інформації по особі на основі отриманих в запиті вхідних даних у конкретній таблиці.

Модуль обробки отриманих результатів відповідає за формування вихідного масиву отриманих в результаті пошуку даних.

Модуль виведення результатів відповідає за виведення кінцевого результату розвідки.

Таким чином, у розділі було розглянуто, описано та обґрунтовано розробку алгоритму розвідки на основі відкритих державою джерел, що було відібрано на основі створених критеріїв надійності та достовірності, інформативності та негативності джерела, що в результаті збільшить ефективність роботи спеціалістів з кібербезпеки. Описано вибір використовуваних програмних технологій. Розроблена структура та основні можливості програми.

3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОЇ РОЗВІДКИ

3.1 Розробка алгоритму роботи додатку

Основною частиною додатку є API, тому основними для створення є такі елементи, як:

- контроллери;
- сутності.

Для керування web API необхідно створити відповідні контроллери, основна задача яких визначати та групувати дії [33]. Під діями розуміються методи контроллера, що обробляють запити. Тобто, основна задача контроллерів - групувати разом схожі дії. При створенні контроллера необхідно наслідувати від класу ControllerBase, що містить в собі багато властивостей та методів, що корисні для керування HTTP запитами, наприклад методи:

- BadRequest;
- NotFound;
- PhysicalFile;
- інші.

Клас BadRequest, наприклад, повертає клієнту статус код 400 (сервер не зміг зрозуміти запит через недійсний синтакс) [34]. Клас NotFound повертає клієнту статус код 404 (не знайдено) [34], PhysicalFile повертає клієнту файл і так далі.

Для кожного з реєстрів необхідно створити власний контроллер, у яких імплементувати інтерфейс парсингу даних з джерела (оскільки кожен набір має свою структуру, для якої необхідно написати окрему реалізацію парсера) та необхідні функції для реалізації CRUD (створення, читання, оновлення та видалення інформації з бази даних) [35]. Проте, оскільки бази даних з формату CSV будуть заноситись та оновлюватись через стороннє програмне забезпечення, контролерів для них створено не буде. Таким чином, слід створити наступні контролери:

- CopyrightController;
- CorruptionController;
- DebtorController;
- HidingController;
- PurificationController;
- ArbitralController;
- ForensicController;
- LostController;

Також слід написати функції для пошуку записів у відповідній до контролеру таблиці баз даних. Далі необхідно створити сутності, які би відносились до відповідних таблиць в базах даних. Таким чином, необхідно створити наступний список сутностей:

- Copyright;
- Corruption;
- Customs;
- Debtor;
- Hiding;
- Enforcement;
- Purification;
- Arbitral;
- IndivAndLegal;
- Forensic;
- Lost;

Після цього необхідно створити нову міграцію, в якій буде згенеровано код для переносу сутностей в базу даних. Міграції виступають чимось в роді системи контролю версій, тільки не для коду, а для бази даних. Також до кожної з сутностей буде створено відповідний клас об'єкту передачі даних (з однаковою до сутності назвою, та прибавкою Dto до назви класа).

Таким чином в результаті розробки алгоритму було складено UML-діаграму, що зображено на рисунку 3.1.

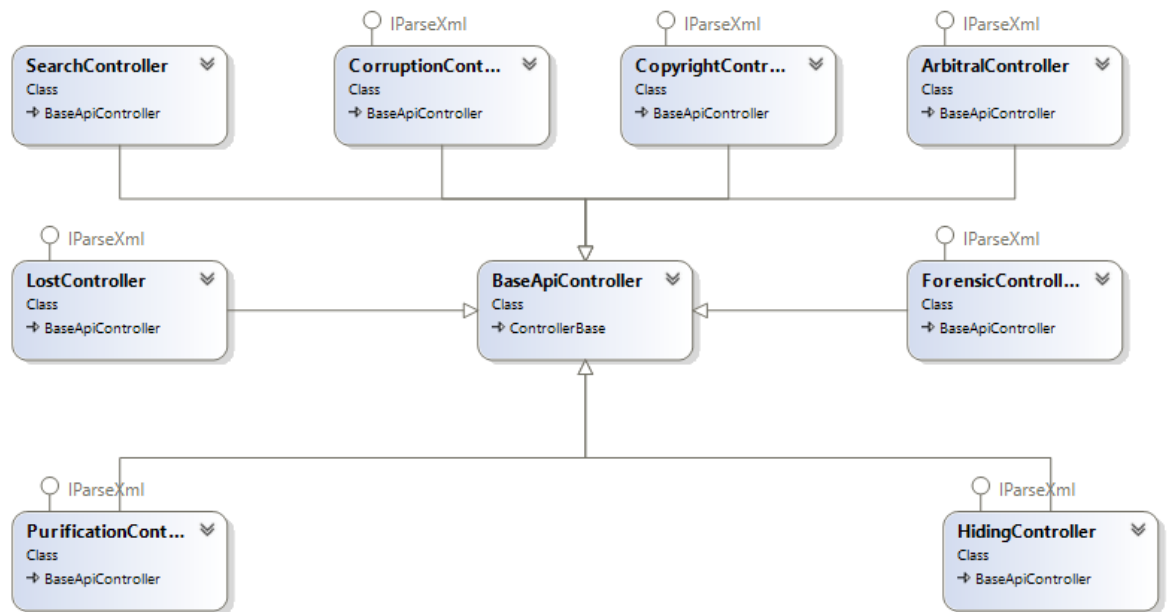


Рисунок 3.1 – UML-діаграма додатку

Оскільки кожна сутність відповідає за роботу з певною таблицею даних, після генерування таблиць з міграцій було створено базу даних, модель якої зображено на рисунку 3.2.

Так як деякі з таблиць виявились дуже великими (більше декількох гігабайт), необхідно їх оптимізувати, для того, щоб пошук здійснювався швидше. Для цього було вирішено використати розбиття таблиці даних [36].

Розбиття бази даних підвищує продуктивність і спрощує її обслуговування. Розбиваючи велику таблицю на менші, окремі таблиці, запити, які отримують доступ лише до частини даних, можуть виконуватися швидше, оскільки потрібно сканувати менше даних.

Розбиття на розділи можна досягти без розбиття таблиць шляхом фізичного розміщення таблиць на окремих дисках. Розміщення таблиці на одному фізичному диску та пов'язаних таблиць на окремому диску може підвищити продуктивність запитів, оскільки під час виконання запитів, які

передбачають об'єднання між таблицями, дані зчитуються одночасно кількома головками дисків.

В загальному, можна виділити три види розбиття:

- фізичне;
- горизонтальне;
- вертикальне.

Фізичне розбиття вже було згадано вище, горизонтальне полягає в розбитті однієї таблиці на декілька, за допомогою збереження кількості стовпчиків, та зменшення кількості рядків, шляхом їх логічного розбиття за якимось параметром. Наприклад, записи діляться за місяцем року, таким чином, таблицю можна розбити на 12 частин.

Вертикальне розбиття полягає у розбитті таблиці на декілька за допомогою розділення стовпців. Саме таке розбиття і буде використовуватись у цьому випадку. Було виявлено, що кожна з таблиць має різну кількість стовпців, але завжди має один спільний – ПІБ. Тому, даний стовпець було виділено у окремі таблиці.

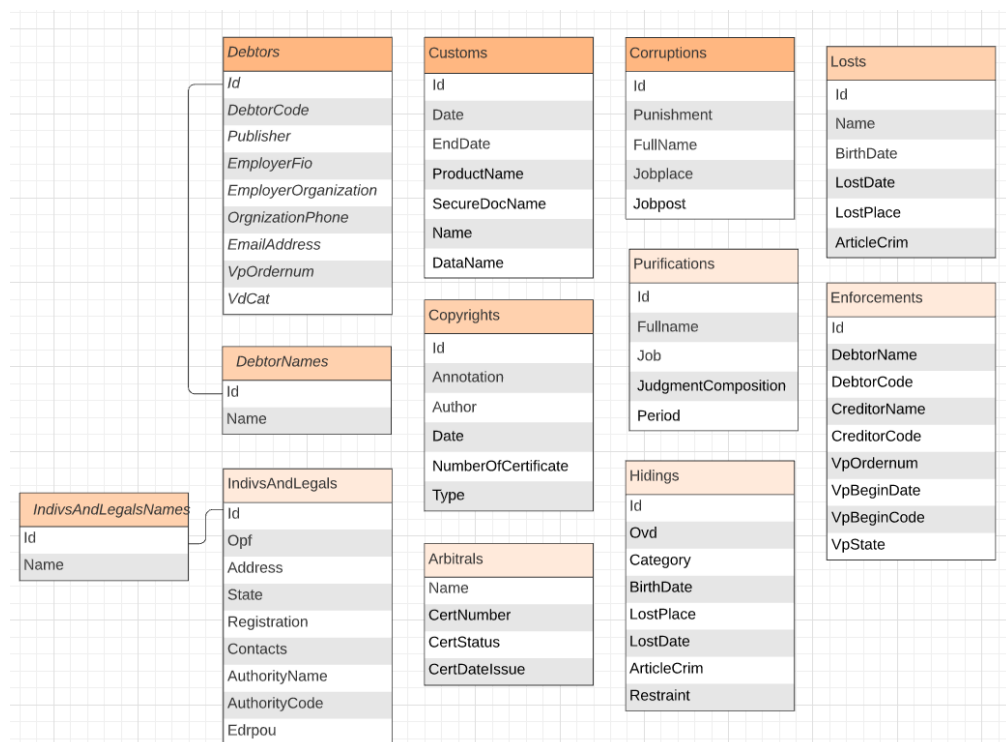


Рисунок 3.2 – Модель бази даних

Оскільки користувацький інтерфейс буде розроблено за допомогою веб-застосунку, було розроблено структуру його сторінок, результат зображено на рисунку 3.3.

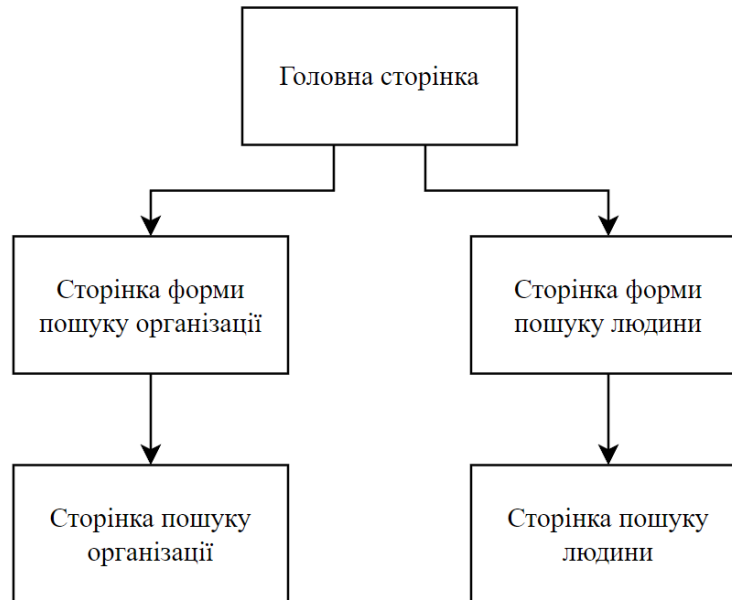


Рисунок 3.3 – Структура сторінок веб-додатку

Таким чином, розробивши алгоритм роботи веб-додатку доцільно перейти до їх реалізації.

3.2 Програмна реалізація додатку

Розробку веб-додатку доцільно почати з розробки бекенду. Для цього створимо новий проект Asp.net core web application, та почнемо розробку.

Важливим для розробки є механізм, за допомогою якого розробники можуть створювати, поширювати та використовувати корисний код. Часто такий код поміщається в пакети, що містять скомпільований код (у виді DLL) та весь інший вміст, необхідний для початку його використання в проекті.

Для .NET та .NET core таким засобом є підтримуваний Microsoft механізм поширювання коду NuGet. У своїй основі NuGet являється ZIP файлом, що містить скомпільований код, інші файли що відносяться до коду, та описовий маніфест, що містить таку інформацію, як версія пакета [37].

Для початку завантажимо в проект необхідні nuget пакети:

- Swashbuckle.AspNetCore;

- Microsoft.EntityFrameworkCore.Sqlite;
- Microsoft.EntityFrameworkCore.

Пакет Swashbuckle.AspNetCore являється особливо корисним при розробці API. Він створює документацію та інтерактивний інтерфейс на основі специфікацій OpenAPI і цим значно спрощує розробку та тестування додатку.

Пакет Microsoft.EntityFrameworkCore необхідний для функціонування фреймворку Entity Framework, що є сучасним модулем співставлення об'єкт-база даних для .NET. Він підтримує запити LINQ, відслідковування змін, оновлення та схеми міграцій, при цьому підтримуючи більшість популярних баз даних [38]. Пакет Microsoft.EntityFrameworkCore.Sqlite дозволяє фреймворку працювати з базами даних SQLite, що будуть використовуватись у додатку.

Оскільки доведеться працювати з великими базами даних, що в деяких випадках займають більше 5 гігабайт, більшість даних в базах буде зберігатись у форматі string, з метою найбільш ефективного парсингу без перетворення типів під час роботи.

Для початку розробки необхідно створити клас, що слугуватиме мостом між кодом, та базою даних. Створимо DataContext.cs, що є наслідником класу DbContext помістивши його в папку Data, в якій міститимуться всі класи, пов'язані з даними. В подальшому, в ньому будуть прописуватись набори сутностей DbSet<TEntity> для всіх створюваних сутностей.

Додатки конфігуруються за допомогою за допомогою ін'єкції залежності (dependency injection). Тому, додаємо сервіс створеного класу-моста в метод ConfigureServices автоматично створеного класу Startup.cs.

Сутності в Entity Framework являють собою класи, що пізніше відобразатимуться в базі даних. Кожна сутність прив'язується до окремої таблиці, а кожна властивість сутності – до окремої колонки в базі даних [38].

Створимо окрему теку для сутностей, та назовемо її Entities. Спочатку створимо сутність Corruption, в якій зазначимо елементи, які можна виділити

після аналізу відповідного XML файлу з даними реєстру, та унікальний ідентифікатор:

- int Id;
- string Punishment;
- string Name;
- string Jobplace;
- string Jobpost.

Таким чином, було створено своєрідний макет для майбутньої таблиці. Після додавання відповідного набору сутності в клас DataContext.cs, створюємо початкову міграцію за допомогою команди `dotnet ef migrations add InitialCreate` – о Data/Migrations. За допомогою цієї команди було створено міграцію InitialCreate. Кожна міграція містить дві функції, Up() та Down(). Функція Up() додає зміни в базу даних, а функція Down() забирає їх звідти. У даному випадку, виклик першої функції створить таблицю Corruptions, а другої – видалить її.

Для того, щоб додати міграцію в базу даних, пропишемо в термінал команду `dotnet ef database update`. З її допомогою базу буде оновлено до останньої міграції. Відповідним чином було створено та додано в базу даних і решту сутностей, згаданих у попередньому підрозділі.

Після цього, можна перейти до створення контролерів. Контролери призначені для обробки HTTP запитів та відповіді на них. До контролерів є декілька вимог, а саме, його назва має закінчуватись на Controller, а сам він має наслідуватись від класу ControllerBase [39].

Для початку, створимо базовий контролер, від якого будуть наслідуватись інші наші контролери. Назвемо його BaseApiController. Він буде наслідуватись від класу ControllerBase, та матиме атрибути [ApiController] та [Route(“api/[controller]”)]. Атрибути в мові програмування C# представляють з себе засіб для зв’язування метаданих або декларативної інформації з кодом (збірки, типи, методи, властивості і т.д.) [40].

Застосування атрибута [ApiController] додає класу контроллера наступну поведінку [33]:

- прокладання шляху (routing) тепер є обов'язковим для кожного контроллера;
- автоматичні відповіді HTTP 400 (валідація моделі);
- вивід параметрів джерел прив'язки;
- вивід багатокomпонентних запитів і запитів даних форм;
- відомості про проблеми для кодів стану помилки.

Атрибут [Route("api/[controller]")] створюється, відповідно, для забезпечення виконання першої умови, обов'язкового прокладання шляху для кожного контроллера.

Наприклад, створимо контроллер CorruptionController для опрацювання в базу реєстру осіб, які вчинили корупційне правопорушення, що буде наслідуватись від класу BaseApiController.

Для того, щоб контроллер мав доступ до бази даних, створюється приватне readonly поле DataContext _context, що ініціалізуватиметься в конструкторі класу. Як правило, в контроллерах реалізуються CRUD функції, тому контроллер буде використано для додавання в таблицю Corruptions відповідних записів з файлу формату XML.

Щоб виконати дану задачу, створено функцію ReadXmlToDatabase з атрибутом [HttpGet], що позначає, що це буде функція для GET-запиту. Оскільки за замовчуванням затримка після запису в базу даних становить 30 мілісекунд, зменшимо її до нуля за допомогою команди _context.Database.SetCommandTimeout(0). Також встановимо кодування "Windows-1254", оскільки саме таке кодування XML файлу, що розглядається. Створюємо об'єкт класу XmlReader, в конструктор якого подаємо шлях до файлу, який будемо парсити.

Після цього, орієнтуючись по тому, що тег <Record> позначає початок нового запиту, було написано умовний перемикач (switch), завдяки якому,

знаючи схему файлу, було записано значення кожного з полів у об'єкт `CorruptionDto`, та на його основі створено новий запис у базі даних, допоки файл не закінчився.

Після цього були написані відповідні контролери з аналогічним призначенням для інших реєстрів, на основі XML файлів.

Для перенесення в базу CSV файлів спочатку було створено тимчасові таблиці, в які за допомогою ПЗ `SQLiteStudio` було перенесено всі їхні значення [41]. Після цього, їх було скопійовано в програмно створені таблиці, аналогічні використаним для XML баз.

Наостанок, після переносу баз, було створено контроллер для реалізації API пошуку по базам даних за ПІБ, в якому створено функцію `public object FindPerson`, з параметрами ім'я, прізвище, та по-батькові. У відповідну до кожної таблиці колекцію було додано записи зі знайденими співпадіннями. Наприклад, за пошук та заповнення колекції боржників відповідає наступний код:

```
List<Debtor> resultDebtorsList = new List<Debtor>();
resultDebtorsList.AddRange( _context.Debtors.Where(i => i.DebtorName.Contains(surname +
" " + name + " " + patronym)));
```

Після пошуку по всім таблицям, було складено об'єкт, що містить всі колекції, та надіслано у якості відповіді на запит.

Для реалізації фронтенду було використано фреймворк `Angular`. Після ініціалізації та підключення проекту до API, було створено основні компоненти веб-додатку, а саме:

- компонент сторінок пошуку `person-detail` та `organization-detail`;
- компонент форми пошуку людини `search-person`;
- форма пошуку організації `search-organization`.

На формі пошуку було додано три поля для вводу, для імені, прізвища та по-батькові. Також було додано дві кнопки, «Пошук», та «Повернутись». Після натиснення кнопки «Пошук», буде здійснено переадресацію на сторінку пошуку разом з даними, введеними у форму, за допомогою наступної функції:

```
this.router.navigate(['/search/persons'], { state: this.model });
```

Тобто, при переході на нову сторінку, передається стан моделі, що сформована з полів форми. Форма пошуку організації відрізняється лише тим, що замість трьох полів ПІБ, доступно лише одне поле для вводу, назва організації.

На програмній стороні коду компонентів сторінок пошуку дані, що були передані разом зі станом сторінки були перехоплені за допомогою наступної функції, та записані у створені в компонентах поля:

```
this.credentials = history.state;
```

Далі, на основі цих даних було сформовано GET-запит та отримано відповідь, яку пізніше проаналізовано, та проведено оцінювання результатів на наявність персональних даних. Також, реєстри, які негативно характеризують особу при виведенні відмічаються відповідними кольорами.

Оскільки для кожного з джерел було попередньо визначено негативність джерел – кожен з елементів <details>, у яких можна розгорнути більш детальну інформацію про записи з реєстру, було позначено попередньо визначеним кольором, як показано у фрагменті коду нижче:

```
<div *ngIf="items.resultCopyrightsList.length != 0" class="container">
  <details>
    <summary style="color: grey">Записи з реєстру авторських свідоцтв</summary>
```

На фрагменті видно, що спочатку йде перевірка, чи було в результаті API-запиту заповнено масив даних, що відповідає реєстру авторських свідоцтв, після чого, за умови наявності даних в реєстрі, відбувається виведення отриманих результатів. Оскільки реєстр авторських свідоцтв було віднесено до реєстрів «сірого» рівня, йому було встановлено сірий колір.

3.3 Тестування додатку

Для початку, проведемо тестування коректності роботи API. Складемо правильний запит за допомогою Postman, результат зображено на рисунку 3.4.

Overview GET https://localhost:5001/api/Search/find?surname=Іванов&name=Іван&patronym=Іванович

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> surname	Іванов	
<input checked="" type="checkbox"/> name	Іван	
<input checked="" type="checkbox"/> patronym	Іванович	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 8.65 s Size: 243.75 KB Save Response

Pretty Raw Preview Visualize JSON

```

1
2
3
4
5
6
7
8
9
10
{
  "resultDebtorsList": [
    {
      "id": 618843,
      "debtorName": "Іванов Іван Іванович",
      "debtorCode": "",
      "publisher": "Управління патрульної поліції у м.Києві",
      "employerFio": "Стрепетова Олександра Сергіївна",
      "employerOrganization": "Деснянський районний відділ державної виконавчої служби міста Київ Головного територіального управління",
      "organizationPhone": "(044) 515-80-22",
    }
  ]
}

```

Рисунок 3.4 – Тестування на коректність API відповіді

Таким чином, при складанні коректного запиту та за умови, що всі необхідні поля заповнені, API надсилає статус код 200 (OK), та повертає результат пошуку.

Спробуємо скласти некоректний запит, де не вкажемо ім'я розшукуваної особи. Результат зображено на рисунку 3.5.

Overview GET https://localhost:5001/api/Search/find?surname=Іванов&name=&patronym=Іванович

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> surname	Іванов	
<input checked="" type="checkbox"/> name		
<input checked="" type="checkbox"/> patronym	Іванович	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 8 ms Size: 181 B Save

Pretty Raw Preview Visualize Text

```

1 Some strings are null or empty

```

Рисунок 3.5 – Тестування на коректність API відповіді

В результаті некоректного запиту, де немає одного або кількох необхідних змінних, API повертає статус код 400 (Bad Request).

Оскільки взаємодія користувача з системою відбувається через користувацький інтерфейс, варто перевірити і його. Для початку, спробуємо перейти на неіснуючу сторінку сайту (/24124125), результат зображено на рисунку 3.6.

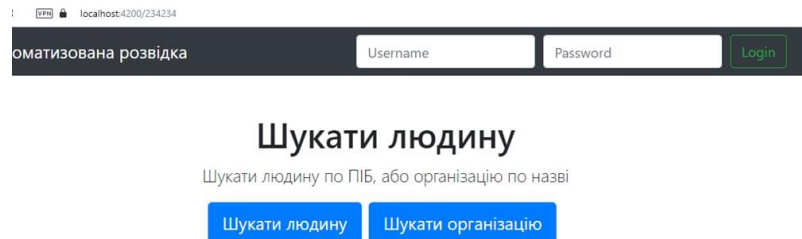


Рисунок 3.6 – Перехід на неіснуючу сторінку сайту

У результаті переходу на неіснуючу сторінку, здійснюється переадресація користувача на головну. Як видно з рисунку, у користувача є вибір з двох варіантів – шукати людину, або організацію.

При натисненні кнопки «Шукати людину» користувачу видається наступна форма, зображена на рисунку 3.7.

Рисунок 3.7 – Форма пошуку людини

Оскільки пошук відбувається в україномовних базах, користувачу не повинно давати змогу вводити дані латиницею, залишати поля пустими, або коротшими за 2 символи. Результати тестування зображено на рисунку 3.7.

The image shows three examples of a search form titled "Пошук людини в базах" (Search for a person in the database). Each example has three input fields: "Ім'я" (Name), "По-батьк" (Patronymic), and "Прізвище" (Surname). Below the fields are "Пошук" (Search) and "Повернутись" (Back) buttons.

- Left example:** All fields are empty. A tooltip above the "По-батьк" field says "Заповніть це поле." (Fill in this field).
- Middle example:** The "Ім'я" field contains the letter "i". A tooltip above it says "Виберіть потрібний формат. Дозволена лише кирилиця. Повинно бути більше 2 символів" (Select the correct format. Only Cyrillic is allowed. It must be more than 2 symbols).
- Right example:** The "Ім'я" field contains "Іван", "По-батьк" contains "Іванович", and "Прізвище" contains "Johnson". A tooltip above the "Прізвище" field says "Виберіть потрібний формат. Дозволена лише кирилиця. Повинно бути більше 2 символів" (Select the correct format. Only Cyrillic is allowed. It must be more than 2 symbols).

Рисунок 3.8 – Тестування форми пошуку людини

У прикладі зліва не було заповнено жодного поля, у прикладі посередині було введено лише один символ (при необхідних двох, або більше), а у прикладі справа було введено латиницю (дозволена лише кирилиця).

Наступною було перевірено форму пошуку організації.

Оскільки назви організацій часто непередбачувані, та можуть приймати різний вигляд в залежності від креативу творців, то єдиним обмеженням для форми було встановлено неможливість введення пустого поля.

Результати тестування форми зображено на рисунку 3.9.

The image shows two examples of a search form titled "Пошук організації в базах" (Search for an organization in the database). Each example has one input field: "Назва організації" (Organization name). Below the field are "Пошук" (Search) and "Повернутись" (Back) buttons.

- Left example:** The "Назва організації" field is empty.
- Right example:** The "Назва організації" field is empty. A tooltip above it says "Заполните это поле." (Fill in this field).

Рисунок 3.9 – Результати тестування форми пошуку організації.

Далі, необхідно провести тестування пошуку. Спробуємо знайти інформацію по комусь, хто точно є в реєстрі боржників, наприклад, Шурпа

Оксана Валентинівна (дані отримано після перевірки реєстру). Результат зображено на рисунку 3.10.

Автоматизована розвідка Username Password Login

Знайдено записів: 8
Серед них: 1, 0, 0

▼ Записи з реєстру боржників

▼ Боржник

Ім'я боржника: Шурпа Оксана Валентинівна
Публікатор: Головне управління ДФС у Чернігівській області
Ім'я співробітника: Зуєнко Наталія Іванівна
Організація співробітника: Новозаводський відділ державної виконавчої служби у місті Чернігові Північно-Східного міжрегіонального управління Міністерства юстиції (м. Суми)
Телефонні номери організації: (0462) 674-290
Електронна пошта організації: info_nz@cnm.cn.dvs.gov.ua
Номер виконавчого провадження: 58943774
Опис: стягнення коштів на користь держави

▼ Боржник

Ім'я боржника: Шурпа Оксана Валентинівна
Публікатор: Приватний нотаріус Київського міського нотаріального округу Хара Н.С.
Ім'я співробітника: Приходько Юрій Михайлович
Організація співробітника: Приватний виконавець Приходько Ю.М.
Телефонні номери організації: (0462) 674-757
Електронна пошта організації: privat_dvs2017@ukr.net

Рисунок 3.10 – Результат пошуку існуючої людини.

В результаті, було знайдено всі записи по вказаній людині. Оскільки реєстр боржників відноситься до реєстрів, що відносяться до червоного рівня за негативністю, записи з цього реєстра виділено яскраво червоним кольором. Зверху сторінки висвітлюється інформація про загальну кількість записів в реєстрах, знайдених про цю людину, а нижче кількість реєстрів, у яких її було знайдено. Таким чином, всього було знайдено 8 записів про людину з таким іменем і всі з них знаходяться у одному реєстрі, реєстрі боржників, що відноситься до червоного рівня за негативністю.

Приклад іншого пошукового запиту зображено на рисунку 3.11. У якості іншого прикладу була здійснена спроба пошуку інформації за іменем Войтович Олеся Петрівна. Як видно з результатів, всього було знайдено 11 записів про людину з таким іменем, та всі з них відносяться до реєстру сірого рівня негативності – реєстру авторських свідоцтв.

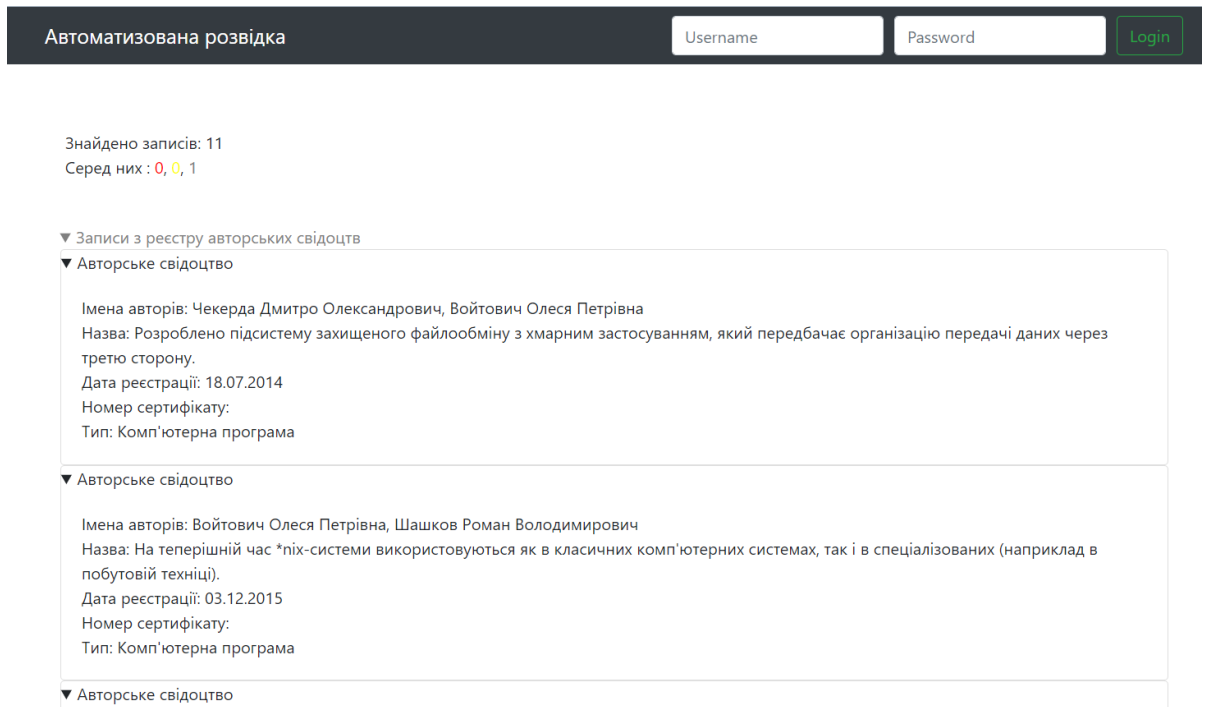


Рисунок 3.11 – Приклад пошукового запиту

Спробуємо знайти неіснуючу людину, введемо нісенітницю «Аааа Аааа Аааа». Результат зображено на рисунку 3.12.

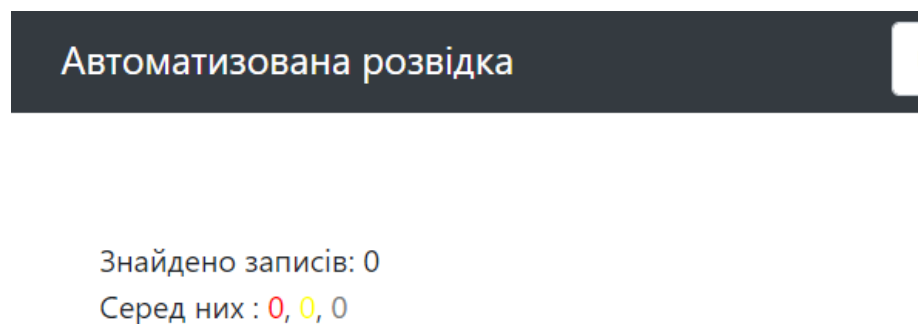


Рисунок 3.12 – Результат пошуку неіснуючої людини

В результаті пошуку неіснуючої людини результатів знайдено не було.

Таким чином, в розділі розроблено алгоритм та складено модель бази даних системи, структуру екранів веб-додатку, після чого розроблено програмний засіб, та протестовано на наявність помилок.

4 ЕКОНОМІЧНА ЧАСТИНА

Метою економічної частини магістерської кваліфікаційної роботи є обґрунтування економічної доцільності системи автоматизованої розвідки. Для виконання поставленої мети необхідно:

- оцінити комерційний потенціал розробки;
- оцінити витрати на виконання та впровадження результатів наукової роботи;
- розрахувати ціну та чистий прибуток реалізації результатів розробки;
- розрахувати період окупності наукової роботи та результатів розробки.

4.1 Оцінювання комерційного потенціалу розробки (технологічний аудит розробки)

Для проведення технологічного аудиту було залучено трьох незалежних експертів: Войтович О. П., Куперштейн Л. М., Лукічов В. В.. Кожен з експертів повинен ознайомитися з запропонованою розробкою та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка. Здійснюємо оцінювання комерційного потенціалу розробки за 12-ю критеріями, наведеними в додатку. Результати оцінювання комерційного потенціалу розробки наведено в таблиці 5.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали експерта		
	Войтович О. П.	Куперштейн Л. М.	Лукічов В. В.
Бали, виставлені експертами:			
1	3	2	2
2	2	3	3
3	3	3	3
4	2	2	1
5	2	2	2

Продовження таблиці 4.1

Критерії	Прізвище, ініціали експерта		
	Войтович О. П.	Куперштейн Л. М.	Лукічов В. В.
	Бали, виставлені експертами:		
6	2	2	2
7	3	2	3
8	4	3	4
9	3	3	3
10	4	4	4
11	3	3	3
12	4	4	4
Сума балів	$CB_1=35$	$CB_2=33$	$CB_3=34$
Середньоарифметична сума балів \overline{CB}	$\overline{CB} = \frac{\sum_1^i CB_i}{i} = \frac{35 + 33 + 34}{3} = 34$		

За даними таблиці 4.1 робимо висновок щодо рівня комерційного потенціалу розробки системи автоматизованої розвідки з використанням відкритих джерел інформації. При цьому використано рекомендації, що наведено у таблиці 4.2.

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів	Рівень комерційного потенціалу
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-50	Високий

Отже, відповідно до результатів оцінювання експертами рівень комерційного потенціалу розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації вище середнього.

Оцінювання рівня якості розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації здійснюється з метою порівняльного аналізу та визначення найбільш ефективного, з технічної точки зору, варіанта інженерного рішення.

Рівень якості – кількісна характеристика міри придатності певного виду продукції для задоволення конкретного попиту на неї при порівнянні з відповідними базовими показниками за фіксованих умов споживання.

Абсолютний рівень якості розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації знаходимо обчисленням обраних для вимірювання показників, не порівнюючи їх із відповідними показниками аналогічних засобів.

Для визначення рівня якості розробки обрано систему параметрів: достовірність отримуваних даних, швидкість пошуку, об'єм даних, кількість взаємодій між сторонами.

Визначаємо величину параметрів якості в балах та встановлюємо граничні значення (кращі, гірші, середні). Дані для кожного параметра представлено у таблиці 4.3.

Таблиця 4.3 – Основні параметри системи для автоматизованої розвідки з використанням відкритих джерел інформації

Параметри	Абсолютне значення параметра			Коефіцієнт вагомості параметра
	Краще +5...+4	Середнє +3	Гірше +1..+2	
достовірність отримуваних даних	4			0.6
швидкість пошуку			2	0.2
об'єм даних			2	0.1
кількість взаємодій		3		0.1

Із врахуванням коефіцієнтів вагомості відповідних параметрів можна визначити абсолютний рівень якості інноваційного рішення за формулою

$$K_{я.а.} = \sum_{i=1}^n P_{H_i} * a_i, \quad (4.1)$$

де P_{n_i} – числове значення i -го параметра інноваційного рішення, n – кількість параметрів інноваційного рішення, що прийняті для оцінювання, a_i – коефіцієнт вагомості відповідного параметра (сума коефіцієнтів вагомості всіх параметрів повинна дорівнювати 1).

Отже, абсолютний рівень якості системи для автоматизованої розвідки з використанням відкритих джерел інформації складає 3,3 бали.

Одночасно визначаємо відносний рівень якості розробленої системи для автоматизованої розвідки з використанням відкритих джерел інформації, шляхом порівнюючи показники з абсолютними показниками якості найліпших аналогів, що представлено у таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів системи для автоматизованої розвідки з використанням відкритих джерел інформації та товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
достовірність отримуваних даних	0,6	1	1,66	0,6
швидкість пошуку	5	8	0,625	0,2
об'єм даних	3КВ	2.5КВ	0,83	0,1
кількість взаємодій	8	6	1,33	0,1

Відносний рівень якості системи для автоматизованої розвідки з використанням відкритих джерел інформації визначаємо за формулою 4.2:

$$K_{я.в.} = \sum_{i=1}^n q_i * a_i \quad (4.2)$$

За розрахунками відносний рівень якості системи для автоматизованої розвідки з використанням відкритих джерел інформації складає 1,34. Це означає, що нова розробка якісніша на 34% відносно товару-аналога.

У найширшому розумінні конкурентоспроможність товару – це можливість його успішного продажу на певному ринку і в певний проміжок часу. Водночас конкурентоспроможною можна вважати лише однорідну продукцію з технічними параметрами і техніко-економічними показниками, що ідентичні аналогічним показникам уже проданого товару. Для того, щоб високоякісний товар був одночасно і конкурентоспроможним, він має відповідати критеріям оцінювання споживачів конкретного ринку в конкретний період часу.

Дані для розрахунку загального показника конкурентоспроможності розробки необхідно занести до таблиці 4.5.

Таблиця 4.5 – Нормативні, технічні та економічні параметри системи для автоматизованої розвідки з використанням відкритих джерел інформації та товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
достовірність отримуваних даних	0,6	1	1,66	0,6
швидкість пошуку	5	8	0,625	0,2
об'єм даних	3КВ	2.5КВ	0,83	0,1
кількість взаємодій	8	6	1,33	0,1
Ціна за продукт, грн	19000	15000	0,79	-

Загальний показник конкурентоспроможності розробки (K) з урахуванням вищезазначених груп показників визначаємо за формулою 4.3:

$$K = \frac{I_{Т.П.}}{I_{Е.П.}} = \frac{1,34}{0,79} = 1,69, \quad (4.3)$$

де $I_{Т.П.}$ – індекс технічних параметрів (відносний рівень якості інноваційного рішення); $I_{Е.П.}$ – індекс економічних параметрів розрахований нижче за формулою 4.4:

$$I_{E.II.} = \frac{PH_{EI}}{PB_{EI}} = \frac{15000}{19000} = 0,79, \quad (4.4)$$

де PH_{EI} , PB_{EI} – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

Згідно розрахунків загальний показник конкурентоспроможності 1,69, що свідчить про більшу конкурентну спроможність системи для автоматизованої розвідки з використанням відкритих джерел інформації у порівнянні з товаром-аналогом на 69%.

4.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи

4.2.1 Розрахунок витрат, що стосуються виконавців розробки системи автоматизованої розвідки з використанням відкритих джерел інформації

Команда розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації складається з керівника, розробника та тестувальника.

Основна заробітна плата для розробників (дослідників) Z_o , якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$Z_o = \frac{M}{T_p} t \quad (4.5)$$

де M – місячний посадовий оклад розробника;

T_p – кількість робочих днів у місяці, $T_p = 22$ дні; t – число днів роботи.

Розрахунки заробітної плати для розробників наведені в таблиці 4.6

Таблиця 4.6 – Розрахунки основної заробітної плати розробників

Працівник	Оклад М, грн.	Оплата робочий грн.	за день, Число роботи, t	днів	Витрати на оплату праці, грн.
Керівник	25000	1136,36	7		7945,52
Розробник	23500	1068.18	15		16022,72
Всього:					23968,24

Основна заробітна плата робітників Z_p , якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Z_p = \sum_{i=1}^n t_i \cdot C_i, \quad (4.6)$$

де t_i – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; C_i – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. C_i визначається за формулою:

$$C_i = \frac{M_M * K_i}{T_p * T_{зм}}, \quad (4.7)$$

де M_M – розмір мінімальної заробітної плати за місяць, грн.; в 2021 році мінімальна заробітна плата становить – 6500 грн., K_i – тарифний коефіцієнт робітника відповідного розряду, $T_p = 22$ дні; $T_{зм}$ – тривалість зміни, $T_{зм} = 8$ годин.

Таблиця 4.7 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Розробка	8	5	56,88	1,54	455
Тестування	8	4	54,52	1,45	436,16
Впровадження	2	2	40,25	1,09	80,51
Всього					971,67

Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0.1 * (Z_o + Z_p) = 0,1 * (23968,24 + 971,67) = 2493,991 \text{ (грн.)} \quad (4.8)$$

Нарахування на заробітну плату $H_{ЗП}$ розраховується як 22% від суми основної та додаткової заробітної плати:

$$H_{ЗП} = (Z_O + Z_P + Z_D) \times \frac{\beta}{100} = (23967,24 + 971,67 + 2493,991) * 0,22 = 6035,46 \text{ (грн.)} \quad (4.9)$$

де Z_O – основна заробітна плата розробників, грн.;

Z_P – основна заробітна плата робітників, грн.;

Z_D – додаткова заробітна плата розробників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне страхування.

Розрахунок амортизаційних відрахувань виконується за такою формулою:

$$A = \frac{Ц}{t_B} \times \frac{T}{12} \quad (4.10)$$

де $Ц$ – балансова вартість обладнання, грн.;

T – термін використання ($T=22$ дні= 0,73 місяців);

t_B – корисний час використання (t_B для комп'ютера становить 4 роки).

Під час виконання розробки використовувався ноутбук вартістю 35000 грн. Амортизаційні відрахування для ноутбуку представлені у таблиці 4.8.

Таблиця 4.8 - Амортизаційні відрахування

Найменування	Ціна, грн.	Корисний час використання, роки	Термін використання, міс.	Сума амортизації, грн.
Ноутбук	35000	4	0,73	532,3
Всього			532,3	

Витрати на силову електроенергію розраховуються за формулою:

$$B_E = B \times P \times \Phi \times K_{II} \quad (4.11)$$

де B – вартість 1кВт-години електроенергії ($B=4,62$ грн/кВт);

P – установлена потужність комп'ютеру ($P=0,74$ кВт);

Φ – фактична кількість годин роботи комп'ютеру ($\Phi=22*8=176$ год);

K_{II} – коефіцієнт використання потужності ($K_{II} < 1$, $K_{II} = 0,8$).

Відповідно до формули 5.11 витрати на силову електроенергію:

$$B_E = 4,62 \times 0,74 \times 176 \times 0,8 = 481,36 \text{ (грн.)}$$

Інші витрати B_{in} можна прийняти як (100-300)% від суми основної заробітної плати розробників, які виконували роботу, тобто:

$$B_{in} = 1 \cdot (23967,24 + 971,67) = 24938,91 \text{ (грн.)} \quad (4.11)$$

Сума усіх попередніх витрат дає загальні витрати на виконання роботи. Усі витрати складають:

$$B = 23967,24 + 971,67 + 2493,991 + 6035,46 + 481,36 + 24938,91 + 523,3 = 59411,93$$

(грн.)

Розрахунок загальної вартості наукової розробки B_{zag} за формулою:

$$B_{zag} = \frac{B}{\alpha}, \quad (4.12)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відносних одиницях.

$$B_{zag} = \frac{59411,93}{1} = 59411,93 \text{ (грн.)}$$

Прогнозування загальних витрат $3B$ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$3B = \frac{B_{zag}}{\beta} \quad (4.13)$$

Розрахунок прогнозованих загальних витрат:

$$3B = \frac{59411,93}{0,7} = 84874,19 \text{ (грн.)}$$

4.2.2 Розрахунок собівартості розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації

Витрати на силову електроенергію розраховуються за формулою:

$$B_E = B \times \Pi \times \Phi \times K_{\Pi} \quad (4.14)$$

де B – вартість 1кВт-години електроенергії ($B=4,62$ грн/кВт);

Π – установлена потужність комп'ютеру ($\Pi=0,74$ кВт);

Φ – фактична кількість годин роботи комп'ютеру ($\Phi=22*8=176$ год);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,8$).

Відповідно до формули 5.14 витрати на силову електроенергію:

$$B_E = 4,62 \times 0,74 \times 176 \times 0,8 = 481,36 \text{ (грн.)}$$

Основна заробітна плата робітників Z_p , якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Z_p = \sum_{i=1}^n t_i \cdot C_i, \quad (4.15)$$

де t_i – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; C_i – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. C_i визначається за формулою:

$$C_i = \frac{M_M * K_i}{T_p * T_{zm}}, \quad (4.16)$$

де M_M – розмір мінімальної заробітної плати за місяць, грн.; в 2021 році мінімальна заробітна плата становить – 6500 грн., K_i – тарифний коефіцієнт робітника відповідного розряду, $T_p = 22$ дні; T_{zm} – тривалість зміни, $T_{zm} = 8$ годин.

Таблиця 4.9 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Розробка	8	5	56,88	1,54	455
Тестування	8	4	54,52	1,45	436,16
Впровадження	2	2	40,25	1,09	80,51
Всього					971,67

Додаткова заробітна плата Z_d всіх робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0.1 * (Z_p) = 0,1 * (971,67) = 97,17 \text{ (грн.)} \quad (4.17)$$

Нарахування на заробітну плату $H_{зп}$ розраховується як 22% від суми основної та додаткової заробітної плати:

$$H_{зп} = (Z_p + Z_d) \times \frac{\beta}{100} = (971,67 + 97,17) * 0,22 = 235,1448 \text{ (грн.)} \quad (4.18)$$

де Z_p – основна заробітна плата робітників, грн.;

Z_d – додаткова заробітна плата робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне страхування.

Загальновиробничі витрати з рахунку на одиницю продукції можна розрахувати за нормативами відносно до основної заробітної плати основних робітників, які виготовляють продукцію :

$$ЗВВ = H_B * Z_o, \quad (4.19)$$

Норматив загальновиробничих витрат для програмних продуктів становить 230-270%.

$$ЗВВ = 2,7 * 971,67 = 2623,51 \text{ (грн.)}$$

Сума попередніх витрат утворює виробничу собівартість розробки.

$$S_B = 481,36 + 971,67 + 97,17 + 235,14 + 2623,51 = 4408,85 \text{ (грн.)} \quad (4.20)$$

4.3 Розрахунок мінімальної ціни та чистого прибутку від реалізації розробки системи автоматизованої розвідки з використанням відкритих джерел інформації

Ціна – це грошовий вираз вартості товару (продукції, послуги). Вона завжди коливається навколо ціни виробництва (перетвореної форми вартості одиниці товару, що дорівнює сумі витрат виробництва й середнього прибутку) та відображає рівень суспільне необхідних витрат праці.

Виходячи з того, що розробки, як правило, приймаються та впроваджуються за завданням замовника, або коли результатом розробки є продукція, що підлягає державному регулюванню, то нижню межу ціни реалізації розробки можна розрахувати за формулою 5.21:

$$C = S_B \cdot \left(1 + \frac{P}{100}\right) \cdot \left(1 + \frac{\omega}{100}\right), \quad (4.21)$$

де S_B – виробнича собівартість інноваційного рішення, грн.;

P – норматив рентабельності узгоджений із замовником або встановлений державою, ($P=30\dots60\%$);

ω – ставка податку на додану вартість, % (з осені 2021 року $\omega = 20\%$).

$$C = 4408,85 \cdot \left(1 + \frac{60}{100}\right) \cdot \left(1 + \frac{20}{100}\right) = 8464,99 \text{ (грн.)} \quad (4.21)$$

Із врахуванням коефіцієнта якості ціна розробки становить 27934,47 грн.

Чистий прибуток від реалізації розробки можна розрахувати за формулою:

$$\Pi = \left(C - \frac{(C - MP) \cdot f}{100} - S_B - \frac{q \cdot S_B}{100}\right) \cdot \left(1 - \frac{h}{100}\right) \cdot P\Pi, \quad (4.22)$$

де C – ціна розробки, грн.; MP – вартість матеріальних та інших ресурсів, що були придбані виробником для виготовлення розробки ($MP=(0,1\dots0,2) C$), грн.; f – зустрічна ставка податку на додану вартість, %; S_B – виробнича собівартість розробки, грн.; q – норматив, який визначає величину адміністративних витрат,

витрат на збут та інші операційні витрати, % (рекомендовано $q=5\dots 10\%$); h – ставка податку на прибуток, %, PP – прогнозований попит продажів.

$$P = (27934,47 - \frac{(27934,47 - 27934,47 \cdot 0,2) \cdot 14}{100} - 4408,85 - \frac{5 \cdot 4408,85}{100}) \cdot (1 - \frac{18}{100}) \cdot 2 = 33811,8$$

(4.23)

Прогнозований чистий прибуток від реалізації розробки складає 33811,8 грн.

4.4 Розрахунок терміну окупності коштів вкладених у наукову розробку системи автоматизованої розвідки з використанням відкритих джерел інформації

Термін окупності вкладених у реалізацію наукового проекту інвестицій розраховано за формулою 4.24:

$$T_{ок} = \frac{3B}{P} = \frac{84874,19}{33811,8} = 2,51 \text{ (роки)} \quad (4.24)$$

Оскільки $T_{ок} < 3$ років, то фінансування наукової розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації є доцільним.

4.5 Висновки до розділу

Отже, у цьому розділі виконано обґрунтування економічної доцільності проведення наукового дослідження та розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації.

Рівень комерційного потенціалу розробки системи для автоматизованої розвідки з використанням відкритих джерел інформації вище середнього.

На основі параметрів системи автоматизованої розвідки визначено абсолютний рівень якості системи, який складає 3,3 бали.

Відносний рівень якості розробки, що складає 1,34. Це означає, що нова розробка якісніша на 34% відносно товару-аналога. Загальний показник конкурентоспроможності становить 1,69, що свідчить про більшу конкурентну спроможність системи для автоматизованої розвідки з використанням відкритих джерел інформації у порівнянні з товаром-аналогом на 69%.

Загальні витрати, що стосуються виконавців розробки склали 84874,19 грн, а собівартість розробки – 4408,85 грн.

Розраховано мінімальну ціну та прогнозований чистий річний прибуток від реалізації розробки, які склали 27934,47 грн. та 33811,8 грн. відповідно. Термін окупності продукції вкладених інвестицій складає 2,51 роки, що свідчить про доцільність фінансування розробки.

ВИСНОВКИ

Розглянуто поняття розвідки з відкритих джерел, сфери використання її результатів, поняття відкритого джерела та стан таких джерел в Україні. Проведено аналіз аналогічних програм, у ході якого виділено їхні переваги та недоліки та з'ясовано, що існуючі програми часто не включають в себе інформацію з відкритих реєстрів України, не проводять аналізу отриманих результатів та поділу джерел на достовірні та неперевірені. Поставлено завдання на розробку системи автоматизованої розвідки на основі відкритих джерел.

Проаналізувавши ряд відкритих джерел, було створено критерії для виділення серед них таких, які можна вважати достовірними. Відкриті державні реєстри задовольнили поставленим до них вимогам. Серед них було виділено найбільш інформативні, для проведення їх парсингу в базу даних системи. Розглянуто категорії персональних даних та визначено, що в реєстрах містяться ідентифіковані дані, після чого проведено подальший їх аналіз та виділено прямі та непрямі ідентифікатори особистості, на основі яких здійснено оцінку максимально можливого рівня інформативності для кожного реєстру, що надає можливість в подальшому оцінювати доцільність введення в базу нових реєстрів даних. Проаналізовано, факт наявності в яких реєстрах може негативно характеризувати персону, розвідка якої проводиться, та розподілено реєстри за декількома рівнями. В якості додаткового джерела неперевірених даних запропоновано використання згенерованих на основі початкових даних пошукових посилань. Під час опису програмних технологій було доведено доцільність використання обраних програмних засобів та технологій, після чого розроблено структуру та архітектуру веб-додатку.

Розроблено алгоритм роботи програми, після чого відбулася програмна реалізація програми та створено інтерфейс, що є інтуїтивно зрозумілим для користувача. Проведено тестування, в результаті чого визначено коректність роботи програмного засобу.

Оцінювання економічної доцільності системи автоматизованої розвідки з використанням відкритих джерел інформації показало, що дана розробка є доцільною, оскільки період окупності складає менше трьох років, що демонструє економічну перспективність отриманих результатів. Розроблену систему автоматизованої розвідки можна використовувати в організаціях, де є потреба до забезпечення підвищеного рівня безпеки.

Реалізований засіб можна використовувати як для особистого, так і для службового використання з метою проведення розвідки окремих персон та організацій. В подальшому планується введення нових джерел даних з переліку відкритих реєстрів, та додавання неперевіраних джерел для підвищення ефективності запобігання атак у кіберпросторі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Панченко Б. Д., Войтович О. П. Кіберраудит в ОС Android. *XLIX Науково-технічна конференція підрозділів Вінницького національного технічного університету*: тези наук.-практ. конф. ВНТУ, Вінниця, 2020 р.
2. 21+ Insider Threat Statistics to Look Out For in 2021. URL: <https://techjury.net/blog/insider-threat-statistics/#gref> (дата звернення: 22.09.2021)
3. Gibson H. Acquisition and preparation of data for OSINT investigations //Open Source Intelligence Investigation. – Springer, Cham, 2016. – С. 69-93.
4. H. J. Williams and I. Blum, "Defining second generation open source intelligence (OSINT) for the defense enterprise", 2018.
5. G. Bello-Orgaz, J. J. Jung and D. Camacho, "Social big data: Recent achievements and new challenges", *Inf. Fusion*, vol. 28, pp. 45-59, Mar. 2016.
6. M. Dawson, M. Lieble and A. Adeboje, "Open source intelligence: Performing data mining and link analysis to track terrorist activities" in *Information Technology—New Generations*, Cham, Switzerland:Springer, vol. 558, pp. 1-11, Jul. 2018.
7. J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity", *J. Comput. Syst. Sci.*, vol. 80, pp. 973-993, Aug. 2014.
8. P. Nespoli, D. Papamartzivanos, F. Gomez Marmol and G. Kambourakis, "Optimal countermeasures selection against cyber attacks: A comprehensive survey on reaction frameworks", *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1361-1396, 2nd Quart. 2018.
9. Z. Jin, J. Cao, Y. Zhang and J. Luo, "News verification by exploiting conflicting social viewpoints in microblogs", *Proc. 13th AAAI Conf. Artif. Intell. (AAAI)*, pp. 2972-2978, 2016.
10. Open Source Intelligence (OSINT): Issues for Congress. — 2007. URL: <https://sgp.fas.org/crs/intel/RL34270.pdf> (дата звернення: 22.09.2021)
11. Google Search Operators: The Complete List URL: <https://ahrefs.com/blog/google-advanced-search-operators/> (дата звернення: 25.09.2021)
12. 10 Facebook Statistics URL: <https://www.oberlo.com/blog/facebook-statistics#:~:text=Facebook%20has%202.80%20billion%20monthly,site%20on%20a%20daily%20basis.> (дата звернення: 25.09.2021)
13. What is wayback machine and what can you use it for URL: <https://henkinschultz.com/2019/02/what-is-wayback-machine-what-can-you-use-it-for/> (дата звернення: 26.09.2021)
14. Про доступ до публічної інформації URL: <https://zakon.rada.gov.ua/laws/show/2939-17#Text> (дата звернення: 28.09.2021)

15. Про затвердження Положення про набори даних, які підлягають оприлюдненню у формі відкритих даних URL: <https://zakon.rada.gov.ua/laws/show/835-2015-п#n12> (дата звернення: 28.09.2021)
16. Open Data in Europe 2020 URL: <https://data.europa.eu/en/dashboard/2020> (дата звернення: 29.09.2021)
17. Зелена книга «Політика відкритих даних» URL: https://regulation.gov.ua/book/145-zelena-kniga-politika-vidkritih-danih?fbclid=IwAR3guC8I_UYr6HZ5DCICVaIQthEwR183VNthCDrjUoPcw6ojKKdzyz8yxfk (дата звернення: 29.09.2021)
18. Набори даних URL: <https://data.gov.ua/dataset> (дата звернення: 29.09.2021)
19. Відкритий реєстр юридичних осіб URL: <https://ring.org.ua/edr/uk/> (дата звернення: 30.09.2021)
20. Maltego URL: <https://www.maltego.com> (дата звернення: 26.09.2021)
21. Recon-NG: Tutorial, Usage, and Tool installation URL: <https://securitytrails.com/blog/recon-ng> (дата звернення: 27.09.2021)
22. Shodan Search Engine URL: <https://www.shodan.io> (дата звернення: 27.09.2021)
23. Personal data vs. sensitive data: what's the difference? URL: <https://www.itgovernance.co.uk/blog/the-gdpr-do-you-know-the-difference-between-personal-data-and-sensitive-data> (дата звернення: 27.10.2021)
24. A VISUAL GUIDE TO PRACTICAL DATA DE-IDENTIFICATION URL: https://fpf.org/wp-content/uploads/2016/04/FPF_Visual-Guide-to-Practical-Data-DeI-D.pdf (дата звернення: 27.10.2021)
25. Definition of Sensitive Data URL: <https://research.csc.fi/definition-of-sensitive-data> (дата звернення: 01.11.2021)
26. Cybersecurity Statistics & Data Trends In 2021 URL: <https://redefineprivacy.com/cybersec/cybersecurity-statistics/> (дата звернення: 01.11.2021)
27. Єдиний державний реєстр юридичних осіб, фізичних осіб-підприємців та громадських формувань URL: <https://usr.minjust.gov.ua/content/free-search> (дата звернення: 04.11.2021)
28. Єдиний реєстр боржників URL: <https://erb.minjust.gov.ua/#/search-debtors> (дата звернення: 04.11.2021)
29. Розшукові обліки МВС URL: <https://wanted.mvs.gov.ua/searchperson/> (дата звернення: 04.11.2021)
30. ЄДР осіб, які вчинили корупційні, або пов'язані з корупцією правопорушення URL: <https://corruptinfo.nazk.gov.ua/> (дата звернення: 07.11.2021)

31. What is .NET URL: <https://chudovo.com/net-developer/> (дата звернення: 07.11.2021)
32. How is C# Compiled? URL: <https://freecontent.manning.com/how-is-c-compiled/> (дата звернення: 08.11.2021)
33. Create web APIs with ASP.NET Core URL: <https://docs.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-6.0> (дата звернення: 08.11.2021)
34. HTTP response status codes URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status> (дата звернення: 11.11.2021)
35. CRUD URL: <https://www.sumologic.com/glossary/crud/> (дата звернення: 11.11.2021)
36. Partitioning URL: [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms178148\(v=sql.105\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms178148(v=sql.105)) (дата звернення: 15.11.2021)
37. NuGet documentation URL: <https://docs.microsoft.com/en-us/nuget/> (дата звернення: 15.11.2021)
38. Entity Framework documentation URL: <https://docs.microsoft.com/en-us/ef/> (дата звернення: 17.11.2021)
39. ControllerBase Class URL: <https://docs.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.mvc.controllerbase?view=aspnetcore-6.0> (дата звернення: 19.11.2021)
40. Attributes (C#) URL: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/attributes/> (дата звернення: 21.11.2021)
41. SQLite studio URL: <https://sqlitestudio.pl> (дата звернення: 22.11.2021)

ДОДАТКИ

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

ЗАТВЕРДЖУЮ

завідувач кафедри ЗІ, д. т. н, проф.

_____ В. А. Лужецький

_____ 2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему

«Система автоматизованої розвідки з використанням відкритих джерел
інформації»

08-20.МКР.009.00.000 ТЗ

Керівник магістерської кваліфікаційної роботи

к. т. н., доц. каф. ЗІ

О. П. Войтович

Вінниця 2021

1 Підстави для проведення робіт

Робота проводиться на підставі наказу ректора ВНТУ від 1 вересня 2021 року № 277.

Дата початку роботи 01.09.2021 р.

Дата закінчення роботи 23.12.2021 р.

2 Мета та призначення НДР

Метою магістерської кваліфікаційної роботи є покращення пошуку інформації з відкритих джерел для запобігання атак у кіберпросторі з використанням відкритих даних.

Об'єктом дослідження є процес збору та організації інформації з відкритих джерел в задачах кібербезпеки.

Предметом є системи для здійснення автоматизованої розвідки з використанням відкритих джерел інформації в задачах кібербезпеки.

Актуальність теми. Оскільки кожна людина, організація і під'єднаний до інтернету пристрій залишає в мережі свій інформаційний слід, велику частину цих даних можна знайти за допомогою легальних інструментів, що є у відкритому доступі. З появою Інтернету, соціальних медіа та цифрових сервісів з'явилися широкі можливості для збору інформації про ІТ-інфраструктуру організації, а також її співробітників. Здійснення автоматизованої розвідки допомагає знайти загальнодоступну інформацію про працівників, внутрішню діяльність компанії, а також дані за її межами. Іноді конфіденційна інформація міститься в метаданих, які організація випадково опублікувала.

Таким чином необхідність підвищення стійкості при автентифікації користувачів є дуже актуальною, оскільки відомі засоби не завжди надають достатній рівень безпеки для сучасних інформаційних систем. Для покращення захищеності автентифікації користувачів варто розглянути засоби автентифікації з найкращими характеристиками з точки зору безпеки та здійснити їх модифікацію.

3 Вихідні дані для проведення НДР

3.1 Tayebi M. A., Glässer U., Skillicorn D. B. Open Source Intelligence and Cyber Crime. – Springer International Publishing, 2020.

3.2 Akhgar B., Bayerl P. S., Sampson F. (ed.). Open source intelligence investigation: from strategy to implementation. – Springer, 2017.

3.3 Троелсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core. : пер. с англ./ Э. Троелсен. – М.: Вильямс, 2020. – 1312 с.

3.4 Uzun E., Yerlikaya T., Kurt M. A lightweight parser for extracting useful contents from Web pages //2nd International Symposium on Computing in Science & Engineering-ISCSE 2011, Kusadasi, Aydin, Turkey. – 2011. – С. 67-73.

4 Виконавці НДР

Студент групи ІБС-20м Панченко Богдан Дмитрович

5 Вимоги до виконання НДР

Для покращення окращення пошуку інформації з відкритих джерел для запобігання атак у кіберпросторі з використанням відкритих даних необхідно розв'язати такі задачі:

- науково-досліджене обґрунтування необхідності дослідження та розробки системи автоматизованої розвідки;
- аналіз інформаційних ресурсів щодо існуючих систем автоматизованої розвідки;
- удосконалення критеріїв для підбору інформаційних джерел;
- розробка алгоритмів функціонування програмного засобу;
- розробка програмного засобу згідно розроблених алгоритмів;
- експериментальне дослідження удосконалених методів автентифікації користувачів.

6 Вимоги до супровідної документації

Графічна і текстова документація повинна відповідати діючим стандартам України.

7 Стадії та етапи розробки

№	Назва етапів роботи	Строк виконання	Результат
Аналіз завдання. Вступ	01.09.2021	04.09.2021	Вступ
Розробка технічного завдання	05.09.2021	10.09.2021	Технічне завдання
Обґрунтування доцільності досліджень, аналіз інформаційних джерел	11.09.2021	04.10.2021	Аналіз існуючих аналогів. Вибір напрямку дослідження Аналіз відомих систем. Постановка завдання
Розробка алгоритму засобу	24.10.2021	17.11.2021	Алгоритм розвідки, критерії оцінки інформаційних джерел

№	Назва етапів роботи	Строк виконання	Результат
Експериментальні дослідження	18.11.2021	17.11.2021	Програмний засіб, який реалізує розроблений алгоритм
Розробка економічного розділу	18.11.2021	24.11.2021	Економічні показники дослідження
Оформлення пояснювальної записки	25.11.2021	30.11.2021	Пояснювальна записка

8 Очікувані результати та порядок реалізації НДР

Передбачається розробка нових (удосконалення існуючих) систем автоматизованої розвідки з відкритих джерел інформації. Заплановане створення програмного засобу, який може бути використаний у навчальному процесі.

9 Матеріали які подаються після закінчення НДР

По завершенню роботи подається пояснювальна записка та ілюстративна частина.

10 Порядок приймання НДР та її етапів

Апробація на науково-технічних конференціях та семінарах. Результати роботи будуть розглядатися на засіданні ДЕК із захисту магістерських кваліфікаційних робіт.

Попередній захист та доопрацювання МКР грудень 2021 р.

Представлення МКР до захисту 5 грудня 2021 р.

Захист МКР 21-23 грудня 2021 р.

11 Вимоги до розроблення документації

Документація буде виконуватись за допомогою комп'ютерного набору у відповідності вимог ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання».

12 Вимоги щодо технічного захисту інформації з обмеженим доступом

У зв'язку з тим, що дана робота не містить інформації, що потребує захисту у відповідності до законів України, заходи з її технічного захисту не передбачаються.

Розробив студент групи ІБС-20м

_____ Панченко Б. Д.

Додаток Б Код серверної частини

Клас BaseApiController

```
using Microsoft.AspNetCore.Mvc;

namespace API.Controllers;

[ApiController]
[Route("api/[controller]")]
public class BaseApiController : ControllerBase
{
}
```

Клас SearchController

```
using System.Net;
using System.Text.Json.Serialization;
using API.Data;
using API.DTOs;
using API.Entities;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Newtonsoft.Json;

namespace API.Controllers;

public class SearchController : BaseApiController
{
    private readonly DataContext _context;

    public SearchController(DataContext context)
    {
        _context = context;
    }

    [AllowAnonymous]
    [HttpGet("find")]
    public object FindPerson(string name, string surname, string patronym)
    {
        if (string.IsNullOrEmpty(name) || string.IsNullOrEmpty(surname) ||
            string.IsNullOrEmpty(patronym))
        {
            return this.BadRequest("Some strings are null or empty");
        }

        List<AppDebtor> resultDebtorsList = GetDebtors(surname, name, patronym);

        /*
        List<AppCorruption> resultCorruptionsList = new List<AppCorruption>();
        resultCorruptionsList.AddRange(_context.Corruptions.Where(i =>
            i.FullName.Contains(surname + " " + name + " " + patronym)));
        */

        List<AppCorruption> resultCorruptionsList = GetCorruptioners(surname, name,
```

```

patronym);
    List<AppCopyright> resultCopyrightsList = GetCopyrights(surname, name,
patronym);

    return new { resultDebtorsList, resultCorruptionsList, resultCopyrightsList };
}

private List<AppCorruption> GetCorruptioners(string surname, string name, string
patronym)
{
    List<AppCorruption> resultCorruptionsList = new List<AppCorruption>();
    resultCorruptionsList.AddRange( _context.Corruptions.Where(i =>
i.FullName.Contains(surname + " " + name + " " + patronym)));

    return resultCorruptionsList;
}
private List<AppDebtor> GetDebtors(string surname, string name, string patronym)
{
    List<AppDebtorNames> resultDebtorNamesList = new List<AppDebtorNames>();
    resultDebtorNamesList.AddRange( _context.DebtorNames.Where(i =>
i.DebtorName.Contains(surname + " " + name + " " + patronym)));

    List<AppDebtor> resultDebtorsList = new List<AppDebtor>();
    foreach (var res in resultDebtorNamesList)
    {
        resultDebtorsList.Add(_context.Debtors.Find(res.Id));
    }

    return resultDebtorsList;
}

private List<AppCopyright> GetCopyrights(string surname, string name, string
patronym)
{
    List<AppCopyright> resultCopyrightsList = new List<AppCopyright>();
    resultCopyrightsList.AddRange( _context.Copyrights.Where(i =>
i.Author.Contains(surname + " " + name + " " + patronym)));

    return resultCopyrightsList;
}
}

```

```

}

```

Клас CopyrightController

```

using System.Text;
using System.Xml;
using API.Data;
using API.Entities;
using API.Interfaces;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.IdentityModel.Tokens;

namespace API.Controllers;

public class CopyrightController : BaseApiController, IParseXml
{
    private readonly DataContext _context;
}

```

```

public CopyrightController(DataContext context)
{
    _context = context;
}

[AllowAnonymous]
[HttpGet]
public void ParseXml()
{
    _context.Database.SetCommandTimeout(0);
    Encoding.RegisterProvider(CodePagesEncodingProvider.Instance);
    Encoding.GetEncoding("windows-1254");

    string annotation = "",
        author = "",
        employer = "",
        date = "",
        numberOfCertificate = "",
        type = "";
    using (XmlReader reader =
        XmlReader.Create("E:\\dotnetcore\\DatingApp\\API\\XmlFiles\\derzhavnii-reiestr-
        svidotstv-pro-reiestratsiiu-avtorskogo-prava-na-tvir_22-04-2021.xml"))
    {
        while (reader.Read())
        {
            if (reader.IsStartElement())
            {
                switch (reader.Name.ToString())
                {
                    case "ROW":
                        Console.WriteLine("RECORD START");
                        author = reader.GetAttribute("author");
                        annotation = reader.GetAttribute("annotation");
                        employer = reader.GetAttribute("employer");
                        date = reader.GetAttribute("date");
                        numberOfCertificate = reader.GetAttribute("number");
                        type = reader.GetAttribute("object");
                        if (annotation.IsNullOrEmpty() && author.IsNullOrEmpty() &&
                            employer.IsNullOrEmpty() && date.IsNullOrEmpty() &&
                            numberOfCertificate.IsNullOrEmpty())
                        {
                            break;
                        }

                        var copyright = new AppCopyright
                        {
                            Annotation = annotation,
                            Author = author,
                            Empoloyer = employer,
                            Date = date,
                            NumberOfCertificate = numberOfCertificate,
                            Type = type
                        };
                        _context.Copyrights.Add(copyright);
                        _context.SaveChanges();

                        author = "";
                        employer = "";
                        date = "";
                        numberOfCertificate = "";
                        type = "";
                    }
                }
            }
        }
    }
}

```



```

        judgementdate.IsNullOrEmpty())
    {
        Console.WriteLine("EMPTY RECORD");
        break;
    }

    Console.WriteLine("Name: " + fullname + " " + "Punishment: " +
punishment + " " + "Jobplace: " +
        + jobplace + " " + "Jobpost" + jobpost + " " +
"Judgementdate: " + judgementdate);

    var corruption = new AppCorruption
    {
        FullName = fullname,
        Punishment = punishment,
        Jobplace = jobplace,
        Jobpost = jobpost,
        JudgmentDate = judgementdate
    };
    _context.Corruptions.Add(corruption);
    _context.SaveChanges();

    fullname = "";
    punishment = "";
    jobplace = "";
    jobpost = "";
    judgementdate = "";
    break;
case "FIO":
    fullname = reader.ReadString();
    break;
case "PUNISHMENT":
    punishment = reader.ReadString();
    break;
case "JOBPLACE":
    jobplace = reader.ReadString();
    break;
case "JOBPOST":
    jobpost = reader.ReadString();
    break;
case "JUDGMENTDATE":
    judgementdate = reader.ReadString();
    break;
    }
    }
    }
    }
}

```

Клас Program

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;

```

```
namespace API
```

```

{
    public class Program
    {
        public static void Main(string[] args)
        {
            CreateHostBuilder(args).Build().Run();
        }

        public static IHostBuilder CreateHostBuilder(string[] args) =>
            Host.CreateDefaultBuilder(args)
                .ConfigureWebHostDefaults(webBuilder =>
                {
                    webBuilder.UseStartup<Startup>();
                });
    }
}

```

Клас Startup

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using API.Data;
using API.Extensions;
using API.Interfaces;
using API.Services;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Options;
using Microsoft.IdentityModel.Tokens;
using Microsoft.OpenApi.Models;

namespace API
{
    public class Startup
    {
        private readonly IConfiguration _config;
        public Startup(IConfiguration config)
        {
            _config = config;
        }

        // This method gets called by the runtime. Use this method to add services to
        // the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddApplicationServices(_config);
            services.AddControllers();
            services.AddSwaggerGen(c =>
            {

```

```

        c.SwaggerDoc("v1", new OpenApiInfo { Title = "WebAPIv5", Version = "v1"
    });
    });
    services.AddCors();
    services.AddIdentityServices(_config);

    }

    // This method gets called by the runtime. Use this method to configure the
    HTTP request pipeline.
    public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        if (env.IsDevelopment())
        {
            app.UseDeveloperExceptionPage();
            app.UseSwagger();
            app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json",
"WebAPIv5 v1"));
        }

        app.UseHttpsRedirection();

        app.UseRouting();

        app.UseCors(x =>
x.AllowAnyHeader().AllowAnyMethod().WithOrigins("https://localhost:4200"));

        app.UseAuthentication();
        app.UseAuthorization();

        app.UseEndpoints(endpoints =>
        {
            endpoints.MapControllers();
        });
    }
}
}
}

```

Класи Entity

```

namespace API.Entities;

public class AppCorruption : BaseEntity
{
    public string Punishment { get; set; }
    public string FullName { get; set; }
    public string Jobplace { get; set; }
    public string Jobpost { get; set; }
    public string JudgmentDate { get; set; }
}

namespace API.Entities;

public class AppCustoms
{
    public string Id { get; set; }
    public string Date { get; set; }
    public string EndDate { get; set; }
    public string ProductName { get; set; }
    public string SecureDocName { get; set; }
    public string Name { get; set; }
}

```

```
    public string DataName { get; set; }
}
namespace API.Entities;

public class AppDebtor : BaseEntity
{
    public string DebtorName { get; set; }
    public string DebtorCode { get; set; }
    public string Publisher { get; set; }
    public string EmployerFio { get; set; }
    public string EmployerOrganization { get; set; }
    public string OrganizationPhone { get; set; }
    public string EmailAddress { get; set; }
    public string VpOrdernum { get; set; }
    public string VdCat { get; set; }
}
namespace API.Entities;

public class AppCopyright : BaseEntity
{
    public string Annotation { get; set; }
    public string Author { get; set; }
    public string Empoloyer { get; set; }
    public string Date { get; set; }
    public string NumberOfCertificate { get; set; }
    public string Type { get; set; }
}
namespace API.Entities;

public class BaseEntity
{
    public long Id { get; set; }
}
namespace API.Entities;

public class AppPurification : BaseEntity
{
    public string Fullname { get; set; }
    public string Job { get; set; }
    public string JudgmentComposition { get; set; }
    public string Period { get; set; }
}
```

Додаток В

Код клієнтської частини

person-detail.component.html

```

<div class="card-body">
  Знайдено записів: {{ countRegisters()}}<br>
  Серед них : <font style="color: red">{{ countRedRegisters()}}</font>, <font
style="color: yellow">{{ countYellowRegisters()}}</font>, <font style="color: grey">{{
countGreyRegisters()}}</font> <br>
</div>

<div *ngIf="items.resultCorruptionsList.length != 0" class="container">
  <details>
    <summary style="color: red">Записи з реєстру корупціонерів</summary>
    <div *ngFor="let result of items.resultCorruptionsList">
      <details class="card" open>
        <summary >Корупціонер</summary>
        <div class="card-body">
          Ім'я: {{ result.fullName }}<br>
          Покарання: {{ result.punishment }}<br>
          Місце роботи: {{ result.jobplace }}<br>
          Займана посада: {{ result.jobpost }}<br>
        </div>
      </details>
    </div>
  </details>
</div>

<br>
<div *ngIf="items.resultDebtorsList.length != 0" class="container">
  <details>
    <summary style="color: red">Записи з реєстру боржників</summary>
    <div *ngFor="let result of items.resultDebtorsList">
      <details class="card" open>
        <summary>Боржник</summary>
        <div class="card-body">
          Ім'я боржника: {{ result.debtorName }}<br>
          Публікатор: {{ result.publisher}}<br>
          Ім'я співробітника: {{ result.employerFio }}<br>
          Організація співробітника: {{ result.employerOrganization}}<br>
          Телефонні номери організації: {{ result.organizationPhone }}<br>
          Електронна пошта організації: {{ result.emailAddress}}<br>
          Номер виконавчого провадження: {{ result.vpOrdernum }}<br>
          Опис: {{ result.vdCat }}
        </div>
      </details>
    </div>
  </details>
</div>

<div *ngIf="items.resultCopyrightsList.length != 0" class="container">
  <details>
    <summary style="color: grey">Записи з реєстру авторських свідоцтв</summary>
    <div *ngFor="let result of items.resultCopyrightsList">
      <details class="card" open>
        <summary >Авторське свідоцтво</summary>
        <div class="card-body">
          Імена авторів: {{ result.author }}<br>

```

```

        Назва: {{ result.annotation }}<br>
        Дата реєстрації: {{ result.date }}<br>
        Номер сертифікату: {{ result.numberofCertificate }}<br>
        Тип: {{ result.type }}<br>
    </div>
</details>
</div>
</details>
</div>
...

```

person-detail.component.ts

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { PersonService } from 'src/app/_services/person.service';

@Component({
  selector: 'app-person-detail',
  templateUrl: './person-detail.component.html',
  styleUrls: ['./person-detail.component.css']
})

export class PersonDetailComponent implements OnInit {

  items = {resultDebtorsList: [], resultCorruptionsList: [], resultCopyrightsList: []};
  credentials: any = {};

  constructor(private http: HttpClient, private router: Router) {}

  getMember(){
    var url = 'https://localhost:5001/api/Search/find?name='
    var firstname = this.credentials.firstname + '';
    if(firstname == 'undefined'){
      this.router.navigate(['/search-person']);
    }
    this.http.get(url+this.credentials.firstname+'&surname='+this.credentials.lastname+
    '&patronym='+this.credentials.patronym).subscribe((data: any) =>{
      const { resultDebtorsList, resultCorruptionsList, resultCopyrightsList } = data;
      this.items.resultDebtorsList = resultDebtorsList
      this.items.resultCorruptionsList = resultCorruptionsList
      this.items.resultCopyrightsList = resultCopyrightsList
    })
  }

  countRegisters(){
    var registerCounter = 0;
    this.items.resultCorruptionsList.forEach(value => {
      registerCounter++;
    })

    this.items.resultDebtorsList.forEach(value => {
      registerCounter++;
    });

    this.items.resultCopyrightsList.forEach(value => {
      registerCounter++;
    });
  }
}

```

```

    return registerCounter;
}

evaluateRegisters(){
    var value = 0;

    if(this.items.resultCorruptionsList.length != 0)
        value += 4;
    if(this.items.resultDebtorsList.length != 0)
        value += 2;
}

countRedRegisters(){
    var counter = 0;
    if(this.items.resultCorruptionsList.length != 0)
        counter++;
    if(this.items.resultDebtorsList.length != 0)
        counter++;

    return counter;
}
countYellowRegisters(){
    var counter = 0;

    return counter;
}
countGreyRegisters(){
    var counter = 0;
    if(this.items.resultCopyrightsList.length != 0)
        counter++;

    return counter;
}

ngOnInit(): void {
    this.credentials = history.state;
    console.log(this.credentials);
    this.getMember();
}

}

search-person

<form #searchPersonForm="ngForm" (ngSubmit)="searchPerson()" autocomplete="off"
ngNativeValidate>
    <h2 class="text-center text-primary">Пошук людини в базах</h2>
    <hr>
    <div class="form-group">
        <input type="text" class="form-control" name="firstname"
[(ngModel)]="model.firstname" placeholder="Ім'я" required minlength="2"
pattern="[\u0400-\u04ff]{3,30}" title="Дозволена лише кирилиця. Повинно бути більше 2
символів">
    </div>
    <div class="form-group">
        <input type="text" class="form-control" name="patronym"
[(ngModel)]="model.patronym" placeholder="По-батькові" required minlength="2"
pattern="[\u0400-\u04ff]{3,30}" title="Дозволена лише кирилиця. Повинно бути більше 2
символів">

```

```

    </div>
    <div class="form-group">
      <input type="text" class="form-control" name="lastname"
[(ngModel)]="model.lastname" placeholder="Прізвище" required minlength="2"
pattern="[\u0400-\u04ff]{3,30}" title="Дозволена лише кирилиця. Повинно бути більше 2
символів">
    </div>

    <div class="form-group text-center">
      <button class="btn btn-success mr-2" type="submit">Пошук</button>
      <button class="btn btn-secondary mr-2" (click)="cancel()"
type="button">Повернутись</button>
    </div>

```

```
</form>
```

```

import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { PersonService } from '../_services/person.service';

```

```

@Component({
  selector: 'app-search-person',
  templateUrl: './search-person.component.html',
  styleUrls: ['./search-person.component.css']
})
export class SearchPersonComponent implements OnInit {
  @Output() cancelSearch = new EventEmitter();
  @Input() model: any = {};

  constructor(private router: Router) { }

  ngOnInit(): void { }

  searchPerson(){
    this.router.navigate(['/search/persons'], { state: this.model });
    console.log(this.model);
  }

  cancel(){
    this.cancelSearch.emit(false);
  }
}

```

app.module.ts

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/common/http';
import { FormsModule } from '@angular/forms'

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { NavComponent } from './nav/nav.component';
import { HomeComponent } from './home/home.component';
import { SearchPersonComponent } from './search-person/search-person.component';
import { SearchOrganizationComponent } from './search-organization/search-
organization.component';
import { PersonDetailComponent } from './search/person-detail/person-detail.component';

```



```

import { OrganizationDetailComponent } from './search/organization-detail/organization-
detail.component';

@NgModule({
  declarations: [
    AppComponent,
    NavComponent,
    HomeComponent,
    SearchPersonComponent,
    SearchOrganizationComponent,
    PersonDetailComponent,
    OrganizationDetailComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    BrowserAnimationsModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

  home

<div class="container mt-5">
  <div *ngIf="!searchOrganizationMode && !searchPersonMode" style="text-align: center">
    <h1>Шукати людину</h1>
    <p class="lead">Шукати людину по ПІБ, або організацію по назві</p>
    <div class="text-center">
      <button (click)="searchPersonToggle()" class="btn btn-primary btn-lg mr-2">Шукати
людину</button>
      <button (click)="searchOrganizationToggle()" class="btn btn-primary btn-lg mr-
2">Шукати організацію</button>
    </div>
  </div>

  <div *ngIf="searchPersonMode" class="container">
    <div class="row justify-content-center">
      <div class="col-4">
        <app-search-person (cancelSearch) ="cancelSearchMode($event)"> </app-search-
person>
      </div>
    </div>
  </div>

  <div *ngIf="searchOrganizationMode" class="container">
    <div class="row justify-content-center">
      <div class="col-4">
        <app-search-organization (cancelSearch) ="cancelSearchMode($event)"> </app-
search-organization>
      </div>
    </div>
  </div>
</div>
import { Component, OnInit } from '@angular/core';

@Component({

```

```
    selector: 'app-home',
    templateUrl: './home.component.html',
    styleUrls: ['./home.component.css']
  })
  export class HomeComponent implements OnInit {
    searchPersonMode = false;
    searchOrganizationMode = false;

    constructor() { }

    ngOnInit(): void {
    }

    searchPersonToggle(){
      this.searchPersonMode = !this.searchPersonMode;
    }
    searchOrganizationToggle(){
      this.searchOrganizationMode = !this.searchOrganizationMode;
    }

    cancelSearchMode(event: boolean){
      this.searchPersonMode = event;
      this.searchOrganizationMode = event;
    }
  }
}
```

Додаток Г
Критерії оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює аналогам	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси. Джерела фінансування відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності більше 5-ти р	Термін реалізації ідеї менше 3-х років. Термін окупності від 3-х до 5-ти р.	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Додаток Д

Результати перевірки роботи на плагіат



Ім'я користувача:
Каплун В.А. ЗІ

ID перевірки:
1009709683

Дата перевірки:
17.12.2021 21:23:52 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
17.12.2021 21:24:19 EET

ID користувача:
61408

Назва документа: МКР_Панченко_На_плагіат

Кількість сторінок: 53 Кількість слів: 9218 Кількість символів: 69616 Розмір файлу: 1.68 MB ID файлу: 1009708141

2.46%

Схожість

Найбільша схожість: 0.82% з джерелом з Бібліотеки (ID файлу: 1009662821)

1.37% Джерела з Інтернету 11 Сторінка 55

1.35% Джерела з Бібліотеки 3 Сторінка 55

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0.23%

Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 15 слів та 0%)

0.1% Вилучення з Інтернету 73 Сторінка 56

0.13% Вилученого тексту з Бібліотеки 11 Сторінка 56

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 5

Схожість

Джерела з Інтернету

11

2	https://data.gov.ua/uploads/files/2018-08-11-104353.234698Recomendation.pdf	0.63%
3	https://nubip.edu.ua/sites/default/files/u145/dis_gorbata.pdf	2 джерела 0.59%
4	http://ippi.org.ua/sites/default/files/zbornik_ot_10.05._3.pdf	0.47%
5	https://nazk.gov.ua/wp-content/uploads/2020/05/Proyekt-Natsdopovidi-2019.pdf	0.46%
6	http://nauka.nlu.edu.ua/download/diss/Mukomela/d_Mukomela.pdf	0.4%
7	https://www.kmu.gov.ua/npas/248573101	0.38%
8	https://thedigital.gov.ua/storage/uploads/files/%D0%9F%D0%A0%D0%9E%D0%95%D0%9A%D0%A2_%D0%9F%D0%9E%D0%...	0.38%
10	http://conf.vntu.edu.ua	0.25%
12	https://www.pravo-izdat.com.ua/image/data/Files/486/3_Publiczna_sluzba_NPd_ispitu_vnutri.pdf	0.2%
13	https://ukranews.com/publication/1230-vyvod-venecyanskoy-komyssyy-po-zakonu-o-lyustracyi-polnyy-tekst-na-ukraynskom-ya..	0.2%

Джерела з Бібліотеки

3

1	Селезньов ID файлу: 1009662821 Навчальний заклад: Vinnytsia National Technical University	0.82%
9	122БДР-КлімінМО2021 ID файлу: 1008368841 Навчальний заклад: Vinnytsia National Technical University	0.3%
11	БДР_Панченко_Р ID файлу: 1004030242 Навчальний заклад: Vinnytsia National Technical University	0.22%

Вилучення

Вилучення

73

http://frgn.mk.ua/wp-content/uploads/2018/09/Itog%20analit.pdf	16 джерел	0.15%
http://diser.ntu.edu.ua/Tsymbol_dis.pdf	7 джерел	0.14%
http://ena.lp.edu.ua:8080/bitstream/ntb/26349/1/22-193-203.pdf		0.12%
https://socialdata.org.ua/manual/manual9	19 джерел	0.1%
https://smirnovska-gromada.gov.ua/news/1588855058	13 джерел	0.1%
http://komsamovr.rada.gov.ua/uploads/documents/42345.pdf		0.09%
http://dspace.onua.edu.ua/bitstream/handle/11300/2378/Diss_Bespalova.pdf?sequence=3&isAllowed=y		0.09%
https://city.cv.ua/storage/files/604f32cd992df.docx		0.09%
https://www.forester.org.ua/pro-zatverdzhennya-zagalnyh-metodychnyh-rekomendatsij-shhodo-zmistu-ta-poryadku-skla	10 джерел	0.09%
http://vtn.ztu.edu.ua/article/download/148328/147951		0.09%
https://niss.gov.ua/sites/default/files/2019-12/dopov_gromad-suspilstvo_druk_04_12_2019.pdf	3 джерела	0.09%

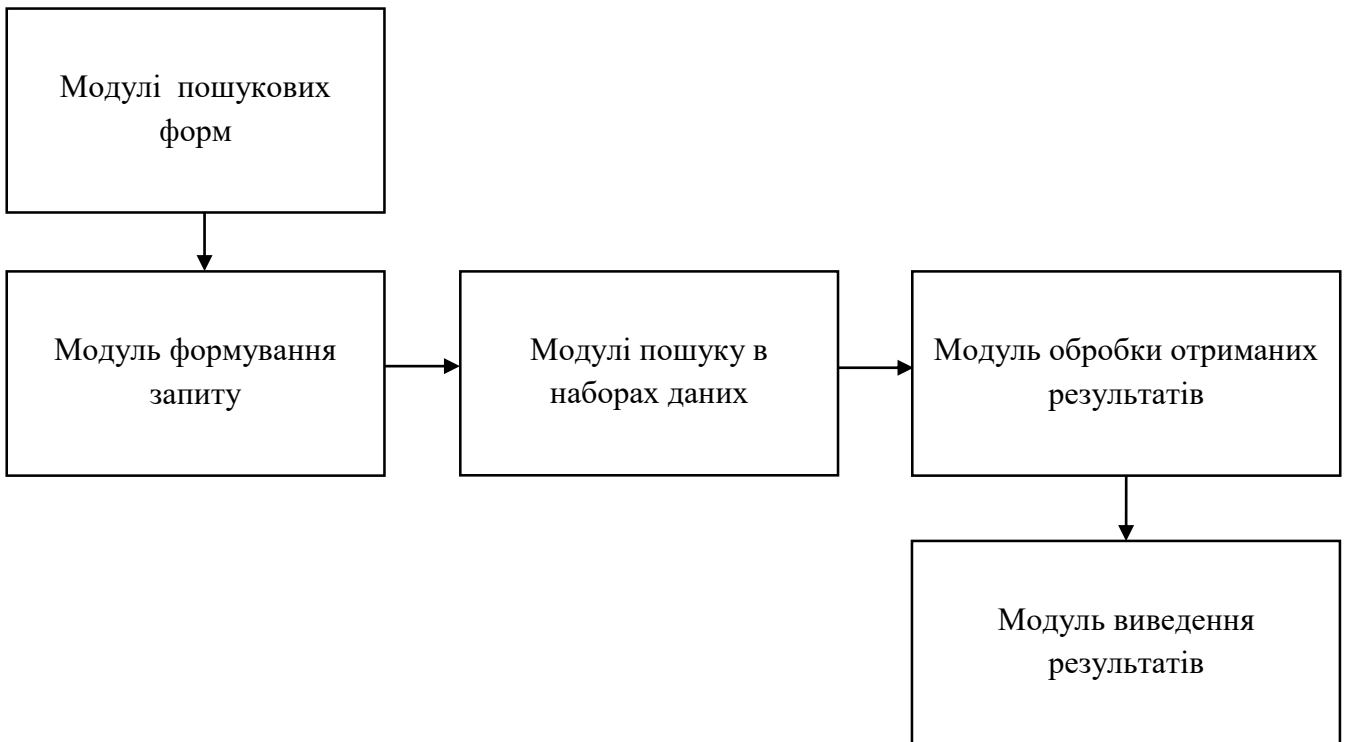
Вилучення по Бібліотеці акаунту

11

122БДР-ШушаковМЮ2021	ID файлу: 1008420152	Навчальний заклад: Vinnytsia National Technical University	0.13%
ЗІ_Головенько	ID файлу: 1000731162	Навчальний заклад: Vinnytsia National Technical University	0.09%
Лиськов	ID файлу: 1009703289	Навчальний заклад: Vinnytsia National Technical University	0.09%
122МКР-КозиреваАА2021	ID файлу: 1009679146	Навчальний заклад: Vinnytsia National Technical University	7 джерел 0.09%
124МКР ЛяховичЛМ2021	ID файлу: 1008162579	Навчальний заклад: Vinnytsia National Technical University	0.09%

ІЛЮСТРАТИВНА ЧАСТИНА

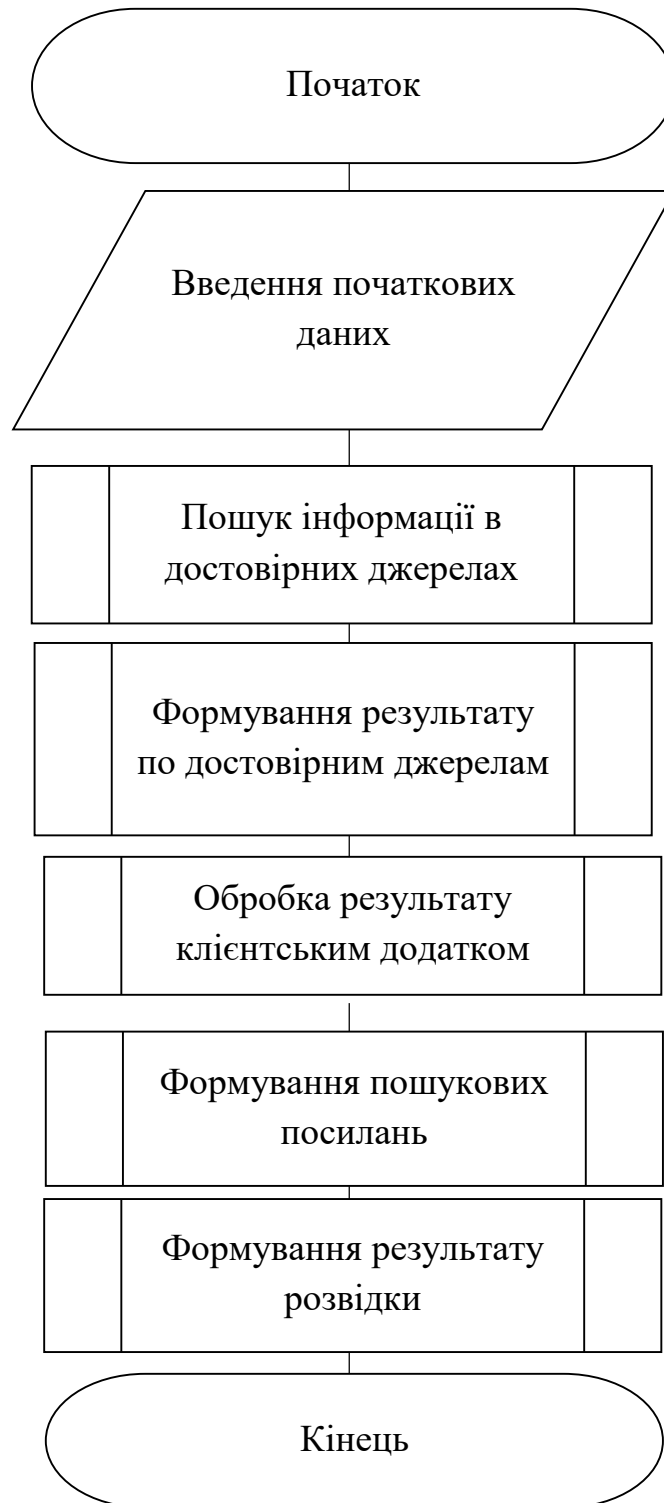
ЗАГАЛЬНА СХЕМА МОДУЛІВ



08-20.МКР.009.00.000 ІЧ1

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Панченко Б. Д.			Система автоматизованої розвідки з використанням відкритих джерел інформації. Загальна схема модулів	Літ.	Арк.	Аркушів
Перевір.		Войтович О. П.					1	1
Реценз.		Войцеховська О. В.				ВНТУ, 1БС-20м		
Н. Контр.		Войтович О. П.						
Затверд.		Лужецький В.А.						

СХЕМА АЛГОРИТМУ РОЗВІДКИ

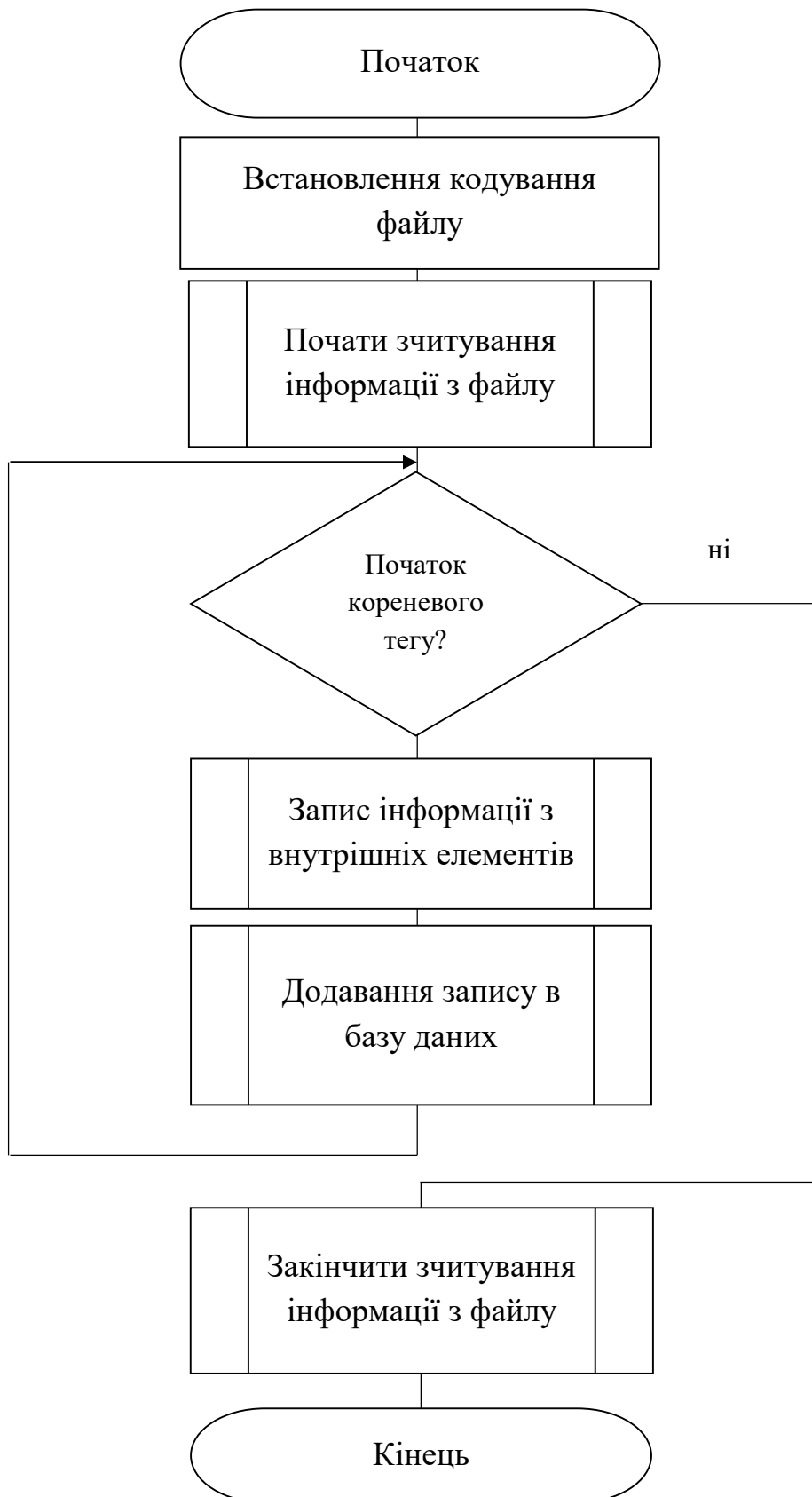


08–20.МКР.009.00.000 І42

Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Панченко Б. Д.			Літ.	Арк.	Аркушів
Перевір.		Войтович О. П.				1	1
Реценз.		Войцеховська О. В.			ВНТУ, 1БС-20м		
Н. Контр.		Войтович О. П.					
Затверд.		Лужецький В.А.					

Система автоматизованої розвідки
з використанням відкритих джерел
інформації. Схема алгоритму
розвідки

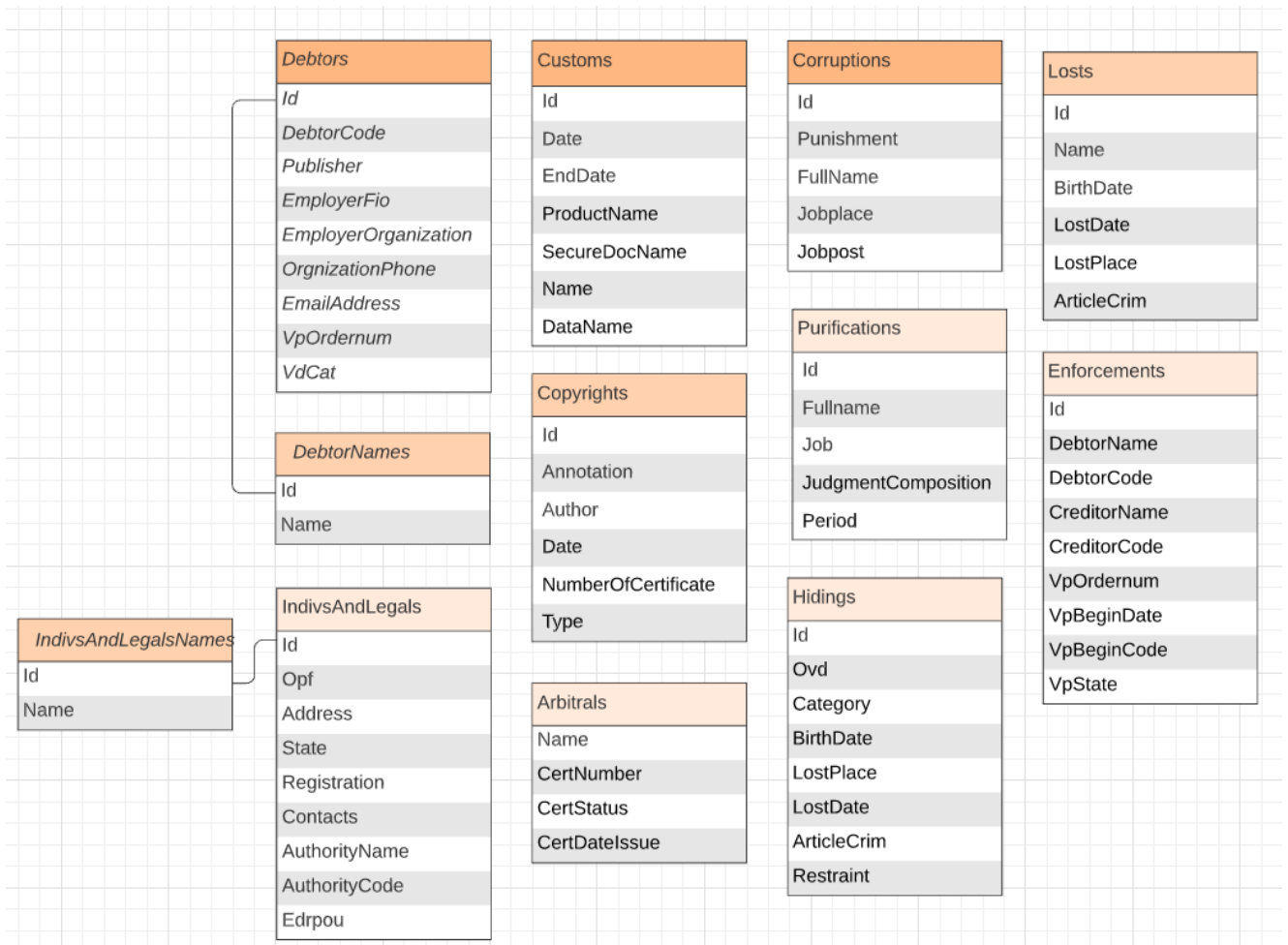
СХЕМА АЛГОРИТМУ ПАРСИНГУ



08–20.МКР.009.00.000 І43

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Панченко Б. Д.			Система автоматизованої розвідки з використанням відкритих джерел інформації. Схема алгоритму парсингу	Літ.	Арк.	Аркушів
Перевір.		Войтович О. П.					1	1
Реценз.		Войцеховська О. В.				ВНТУ, 1БС-20м		
Н. Контр.		Войтович О. П.						
Затверд.		Лужецький В.А.						

СХЕМА БАЗИ ДАНИХ

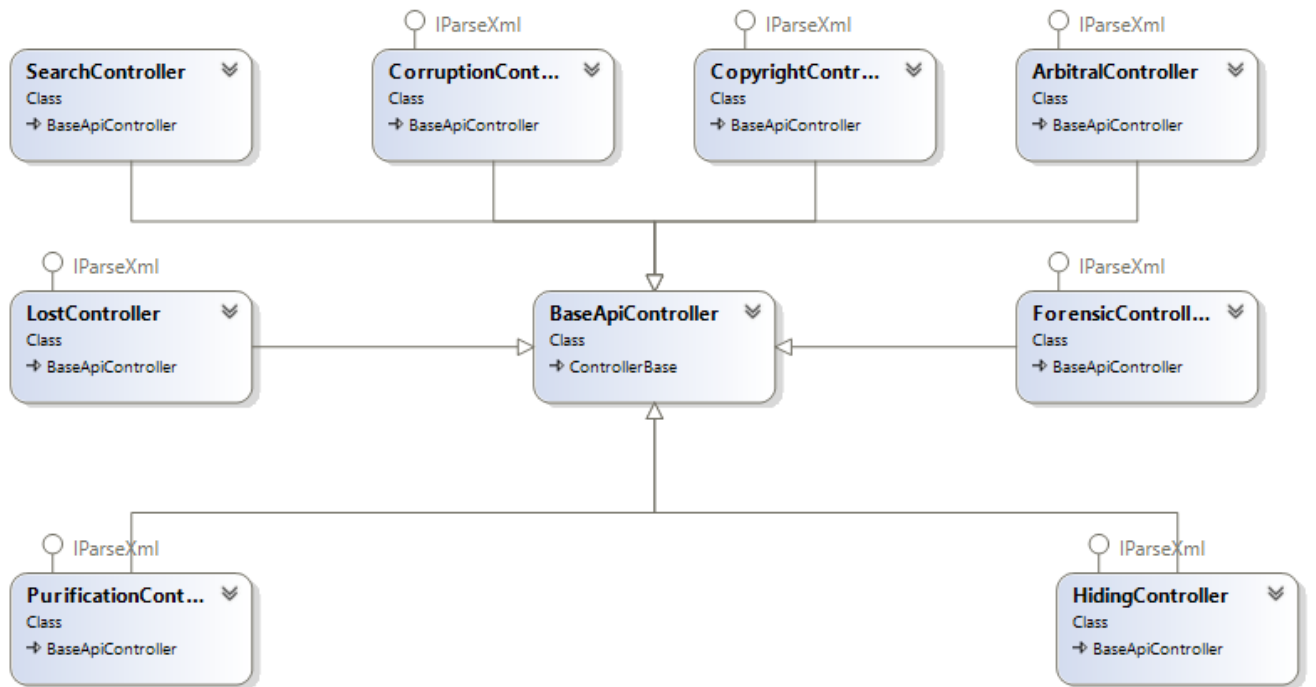


08–20.МКР.009.00.000 ІЧ4

Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Панченко Б. Д.			Літ.	Арк.	Аркушів
Перевір.		Войтович О. П.				1	1
Реценз.		Войцеховська О. В.			ВНТУ, 1БС-20м		
Н. Контр.		Войтович О. П.					
Затверд.		Лужецький В.А.					

Система автоматизованої розвідки з використанням відкритих джерел інформації. Схема бази даних

UML-ДІАГРАМА



08–20.МКР.009.00.000 ІЧ5

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Панченко Б. Д.			Система автоматизованої розвідки з використанням відкритих джерел інформації. UML-діаграма	Літ.	Арк.	Аркушів
Перевір.		Войтович О. П.					1	1
Реценз.		Войцеховська О. В.				ВНТУ, 1БС-20м		
Н. Контр.		Войтович О. П.						
Затверд.		Лужецький В.А.						

РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Пошук людини в базах

Виберіть потрібний формат.
Дозволено лише кирилиця. Повинно бути більше 2 символів

Пошук людини в базах

Виберіть потрібний формат.
Дозволено лише кирилиця. Повинно бути більше 2 символів

Пошук організації в базах

Заповніть це поле.

Overview GET https://localhost:5001/api/Search/find?surname=Іванов&name=Іван&patronym=Іванович No Environment

https://localhost:5001/api/Search/find?surname=Іванов&name=Іван&patronym=Іванович Save

GET https://localhost:5001/api/Search/find?surname=Іванов&name=Іван&patronym=Іванович Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> surname	Іванов	
<input checked="" type="checkbox"/> name	Іван	
<input checked="" type="checkbox"/> patronym	Іванович	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 8.65 s Size: 243.75 KB Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "resultDebtorsList": [
3      {
4        "id": 610843,
5        "debtorName": "Іванов Іван Іванович",
6        "debtorCode": "",
7        "publisher": "Управління патрульної поліції у м.Києві",
8        "employerFio": "Стрепетова Олександра Сергіївна",
9        "employerOrganization": "Деснянський районний відділ державної виконавчої служби міста Київ Головного територіального управління",
10       "organizationPhone": "(844) 515-80-22",
11     }
12   ]
13 }
```

Автоматизована розвідка

Знайдено записів: 11

Серед них: 0, 0, 1

▼ Записи з реєстру авторських свідоцтв

▼ Авторське свідоцтво

Імена авторів: Чекерда Дмитро Олександрович, Войтович Олеся Петрівна

Назва: Розроблено підсистему захищеного файлообміну з хмарним застосуванням, який передбачає організацію передачі даних через третю сторону.

Дата реєстрації: 18.07.2014

Номер сертифікату:

Тип: Комп'ютерна програма

▼ Авторське свідоцтво

Імена авторів: Войтович Олеся Петрівна, Шашков Роман Володимирович

Назва: На теперішній час *nix-системи використовуються як в класичних комп'ютерних системах, так і в спеціалізованих (наприклад в побутовій техніці).

Дата реєстрації: 03.12.2015

Номер сертифікату:

Тип: Комп'ютерна програма

▼ Авторське свідоцтво

08–20.МКР.009.00.000 ІЧ6

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Панченко Б. Д.			Система автоматизованої розвідки з використанням відкритих джерел інформації. Результати тестування	Літ.	Арк.	Аркушів
Перевір.		Войтович О. П.					1	1
Реценз.		Войцеховська О. В.				ВНТУ, 1БС-20м		
Н. Контр.		Войтович О. П.						
Затверд.		Лужецький В.А.						