

ABSTRACT

Explanatory note the project consists of four sections, contains 12 figures, 13 tables, 4 appendices, 28 sources.

Diploma project is dedicated to the development of software for data collection of Internet resources. The purpose of creating software for the aggregation of materials the Internet-editions is providing opportunities to create an updateable archive of materials suitable for further analysis by other software. The thesis project reviewed current approaches to the problem of aggregation of web pages, technical solutions which can solve such a problem.

In the analyze section of the software requirements were defined inputs and outputs to the complex of the tasks, describes the features download papers from online sources. This section describes the use cases, formulated the main functional and non-functional requirements.

In the section simulation and design software was developed by data storage structure, defined the software architecture modeled diagrams of business processes. Identified components and their interaction, constructed class diagram of the software.

In the third section the analysis of the quality and testing of the product, the results obtained are presented

In the fourth section describes the process of software deployment.

					IA/ЛЛ.045430-02-81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

**ВМЕСТО ЭТОГО ЛИСТА ВСТАВИТЬ ЛИСТ ТИТУЛА
ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

					ІАЛЦ.045430-02-81	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 9

ВСТУП 10

1	АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	11
1.1	ЗАГАЛЬНІ ПОЛОЖЕННЯ	11
1.2	ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.3	АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ	14
1.3.1	Аналіз відомих технічних рішень.....	14
1.3.2	Аналіз відомих програмних продуктів	16
1.4	АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	18
1.4.1	Розроблення функціональних вимог	19
1.4.2	Розроблення нефункціональних вимог.....	26
1.4.3	Постановка комплексу задач модулю.....	26
	Висновки по розділу	26
2	МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	28
2.1	МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
2.2	АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	30
2.3	КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	36
	Висновки по розділу	39
3	АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	40
3.1	МЕТА ВИПРОБУВАНЬ ТА ТЕСТУВАННЯ	40
3.2	ОПИС МЕТОДІВ ВИПРОБУВАНЬ	40
3.3	ПЛАН ВИПРОБУВАНЬ	40
	Висновки по розділу	42
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	43
4.1	РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	43
4.1.1	Мінімальна конфігурація технічних засобів	43
4.2	РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ	44
	Висновки по розділу	44

ВИСНОВКИ	45
ПЕРЕЛІК ПОСИЛАНЬ	46
ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ	49
ДОДАТОК Б ОПИС ПРОГРАМИ	50
ДОДАТОК В КЕРІВНИЦТВО КОРИСТУВАЧА	51
ДОДАТОК Г ГРАФІЧНИЙ МАТЕРІАЛ	52
ЛИСТ 1. СХЕМА СТРУКТУРНА ВАРІАНТІВ ВИКОРИСТАНЬ	53
ЛИСТ 2. СХЕМА СТРУКТУРНА КОМПОНЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	54
ЛИСТ 3. СХЕМА СТРУКТУРНА ПОСЛІДОВНОСТЕЙ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	55
ЛИСТ 4. СХЕМА СТРУКТУРНА КЛАСІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	56
ЛИСТ 5. СХЕМА СТРУКТУРНА ДІЯЛЬНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GUI – Graphical User Interface, графічний інтерфейс користувача.

CLI – Command Line Interface, інтерфейс командного рядка.

NoSQL – Not Only SQL, підхід до проектування не реляційних БД.

SQL – Structured Query Language, структурована мова запитів (до БД).

БД – база даних.

СКБД – система керування базами даних.

HTML – HyperText Markup Language, мова розмітки гіпертексту.

XML – eXtensible Markup Language, розширювана мова розмітки.

XSD – XML Schema Definition, мова опису структури XML.

					ІАЛЦ.045430-02-81	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Інформація завжди була основою для людської роботи. Первісні племена зберігали дані про місця полювання, збору ягід і пасовиська. Торговці давніх цивілізацій вели записники із обліком своїх клієнтів та торгових операцій. Полководці – зберігали знання про свої походи, літописці – зберігали знання про історію, прості люди – знання про свята, традиції й свою працю. Спочатку усе зберігали в усній формі та передавали з покоління в покоління, але такий спосіб збереження інформації не був надійним. Із розвитком цивілізації почала розвиватись писемність, що спростило процеси збереження та передачі інформації.

Логічним розвитком прагнення людей спростити процес збереження та передачі інформації стала поява книгодрукування. Дешеві та доступні для мас книги, а отже і освіта, стали основою бурхливого розвитку Європи протягом XVI-XX сторіч.

Із розвитком поліграфії почали з'являтися газети – регулярні видання, де публікували найсвіжішу, найактуальнішу інформацію для читачів. Серед читачів та дописувачів перших газет були королі та кардинали, графи та лорди. Новини, опубліковані у виданнях впливали на світову політику. У наш час цільова аудиторія новин є набагато ширшою – від президентів та міністрів, до простих робочих. На сьогодні через розвиток мережі Інтернет друковані видання заміщуються своїми онлайн-версіями, де публікують тексти, що впливають на сприйняття світу мільйонів людей.

Через великий об'єм публікацій, їх структурну неоднорідність важко автоматизувати їх обробку комп'ютером. Машинну обробку текстів необхідно проводити для спрощення дослідження суспільних трендів, світової думки про ту чи іншу тему ^[1]. Для спрощення машинної обробки необхідно створити інструмент, який дозволить узагальнити, зберегти та гомогенізувати дані з багатьох неоднорідних джерел, таких як сайти новин.

					ІАЛЦ.045430-02-81	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Із самого початку свого існування всесвітня мережа Інтернет росла із приголомшливою швидкістю: щодня з'являлися сотні й тисячі нових веб-сторінок, наповнених вмістом. У результаті бурхливого росту пошук інформації став надзвичайно складним. Для вирішення цієї проблеми з'явилися перші пошукові сервіси – WebCrawler та Lycos (що використовували робота для збору інформації з усієї доступної мережі та її подальшої індексації), і служби каталогів, що заповнювались спеціальними редакторами вручну, – такі як Yahoo. Ці сервіси стали дуже популярними серед користувачів, оскільки зробили веб більш дружнім та легким у використанні. Людям подобається мати доступ до усієї необхідної інформації з одного місця. Пізніше також з'явилися сервіси AltaVista та HotBot, що намагались сформувати найбільший, найповніший індекс мережі Інтернет. [2]

Проте ці сервіси мали численні недоліки. Вони були повільними, – повільно відповідали на запити користувачів, повільно будували та оновлювали свої індекси. Дуже часто сторінки, що досі були у каталогах, насправді змінювали свою адресу або й зовсім зникали, що приводило до поламаних посилань і невдоволення користувачів. Починають з'являтися технології пошукової оптимізації сайтів, через що у перші місця пошукової видачі потрапляли сторінки, що формально підходили під запит, але насправді не містили бажаного користувачем вмісту. З подальшим експоненційним ростом числа веб-користувачів навіть ці пошукові системи, що робили повну індексацію Інтернет, вже не могли задовольнити потреби своїх відвідувачів. [3]

Так само, як наприкінці 20-го сторіччя з'явився попит на пошукові системи, у першому десятилітті 21-го сторіччя з'являється попит на аналітичні системи для роботи з Інтернет. Величезні маси інформації, що

					ІАЛЦ.045430-02-81	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

накопилися та кожного дня з'являються у Всесвітній павутині, являють чималий інтерес для великого числа дослідників – лінгвістів, маркетологів, політологів, соціологів, економістів.

Для вирішення проблеми агрегації даних у мережі Інтернет необхідно створити сервіс із відповідними функціями, який на відміну від наявних пошукових систем міг би гарантувати повноту та цілісність отриманих текстів.

Для найкращої роботи такий сервіс має працювати із розподіленою СКБД. Розподілена база даних – це сукупність логічно зв'язаних баз даних або частин однієї бази, які розпаралелені між декількома територіально-розподіленими ПЕОМ і забезпечені відповідними можливостями для управління цими базами або їх частинами. Тобто, розподілена база даних реалізується на різних просторово розосереджених обчислювальних засобах, разом з організаційними, технічними і програмними засобами її створення і ведення. В дійсності розподілена база даних є віртуальною базою даних, компоненти якої фізично зберігаються на декількох різних реальних базах даних на декількох різних вузлах.

Розподілена система керування базою даних (СКБД) – це програмний комплекс, призначений для управління розподіленими базами даних і забезпечує прозорий доступ користувачів до розподіленої інформації. Прозорий доступ означає непомітний для користувача, тобто у користувача має складатися враження ніби він працює з єдиною базою даних, яка розміщена на його власному комп'ютері. [4]

1.2 Змістовний опис і аналіз предметної області

Предметною областю проекту є створення рішення для агрегації текстів Інтернет-видань, що дозволить з мінімальними зусиллями додавати нові джерела інформації, завантажувати з них дані та надасть єдиний інтерфейс для отримання агрегованих даних сторонніми сервісами.

					ІАЛЦ.045430-02-81	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

У сфері штучного інтелекту та машинної обробки великих масивів даних однією з важливих задач є збір даних для обробки. Задача збору текстових даних, що розміщені у мережі Інтернет (таких як тексти новин, дописи у онлайн-журналах або щоденниках), не є тривіальною – кожен веб-сайт має свою розмітку, структуру та обмеження на кількість запитів. В свою чергу, аналітичні системи, що обробляють такі тексти, для своєї коректної роботи повинні мати доступ до свіжих та актуальних даних.

На Інтернет-ресурсах, що спеціалізуються на новинах, матеріали зазвичай відсортовано за датою, і за датою ж до них можна отримати доступ. Різна структура кожного Інтернет-видання обумовлює необхідність створення окремого обробника для кожного з них. Зібрані дані повинні також бути збережені у структурованому вигляді для подальшого легкого до них доступу.

На діаграмі діяльності на рисунку 1.1 наведено ті види діяльності, що можуть бути здійснені у рамках предметної області.

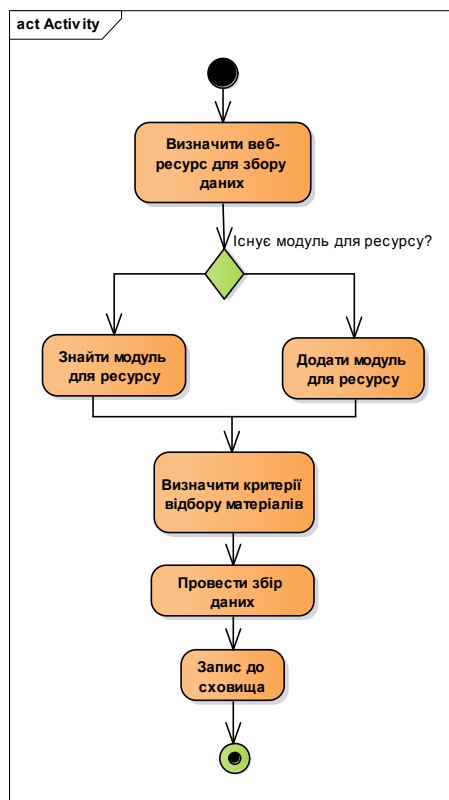


Рисунок 1.1 – Схема структурна діяльності процесу «Агрегація матеріалів»

Змн.	Арк.	№ докум.	Підпис	Дата

1.3 Аналіз успішних ІТ-проектів

Пошук рішень для даної предметної області не показав результатів, які б повністю задовольняли поставленим вимогам, проте є декілька рішень, що виконують схожі функції.

1.3.1 Аналіз відомих технічних рішень

Для агрегації даних з Інтернет-видань та стрічок новин був розроблений формат RSS (Rich Site Summary) – універсальний формат XML файлів, у яких сайти передають всім зацікавленим клієнтам дані про новини, – у тому числі, їх заголовки, тексти, дати публікації та посилання на повний текст новини. Практично усі сайти новин надають RSS-стрічку для своїх користувачів. Спеціальні додатки – читачі RSS-стрічок та агрегатори новин дають можливість читати ці стрічки з багатьох веб-сайтів одночасно.

RSS – спеціальний формат, призначений для опису стрічок новин, анонсів статей, змін у блогах тощо. Інформація з різних джерел, подана у форматі RSS, може бути зібрана, опрацьована і подана користувачеві в зручному для нього вигляді спеціальними програмами.

Зазвичай за допомогою RSS подається короткий опис нової інформації, що з'явилася на сайті, і посилання на її повну версію. Інтернет-ресурс у форматі RSS називається RSS-каналом, RSS-стрічкою або RSS-фідом. ^[5]

Більшість сучасних браузерів та поштових клієнтів працюють з RSS-стрічками. Крім того, існують спеціалізовані програми (RSS-агрегатори), які збирають і опрацьовують інформацію RSS-каналів. Також дуже популярними є веб-агрегатори, які спеціалізуються на збиранні та відображенні RSS-каналів, такі як Яндекс.Лента^[6], Feedly Reader^[7], Новотека^[8]. ^[9]

Недоліком RSS є те, що як правило у стрічках розміщують не повні, а лише часткові тексти новин та їх заголовки, тому для досягнення поставленої мети застосування стрічок RSS не є прийнятним варіантом.

					ІАЛЦ.045430-02-81	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Чимало веб-сайтів також надають API – програмні інтерфейси для взаємодії із собою. Через такі інтерфейси зазвичай можна напряму отримувати дані. На жаль, не всі веб-сайти новин надають публічний API. Деякі сайти мають API, але доступ до нього платний. У будь-якому разі, кожен сайт має свій програмний інтерфейс, тому не вийде зробити універсальний засіб взаємодії з API.

Іншим варіантом технічного рішення задачі агрегації даних є розбір (скрейпінг) Інтернет-ресурсу з метою пошуку текстів новин. Необхідно повністю переглянути html-тексти сторінок Інтернет-ресурсу та виявити, де саме розміщуються тексти новин, вилучити їх з коду сторінки та зберегти на зовнішній носій (у файл чи у базу даних). Такий підхід є більш трудомістким порівняно з розбором RSS-стрічок, оскільки необхідно досліджувати дані з нерегулярною структурою (структура різних веб-сайтів може кардинально відрізнятись). Іншим недоліком скрейпінгу є велике навантаження на сервер стороннього веб-ресурсу. Якщо надсилати запитів для завантаження текстів надто часто, сервер може заблокувати клієнта, що його перевантажує, і таким чином можна втратити можливість завантажувати дані. Перевагою ж скрейпінгу є набагато більша гнучкість при виборі даних та незалежність від поставника даних. У кінцеву вибірку завжди потраплять повні тексти новин, якщо не перевантажувати сервер веб-ресурсу. ^[10]

Існує також технологія розбору веб-сторінок за допомогою «комп'ютерного зору». За допомогою такої технології програма сприймає сайт візуально, як його сприймав би користувач, і зчитує певні елементи спершу у вигляді зображень, а потім оброблює їх та перетворює у текстові або інші дані. ^[11]

Зважаючи на переваги та недоліки кожної технології, а також на розмаїття даних, що необхідно обробити, тяжко обрати якусь одну підходящу для роботи технологію. Найбільш потужною була б можливість скомбінувати

усі доступні технології та дати можливість користувачу самому обрати, яку використовувати, у залежності від його потреб.

1.3.2 Аналіз відомих програмних продуктів

Відомими програмними продуктами, які реалізують функціонал агрегації новин з RSS-стрічок, є Inoreader.Com^[12], Digg.Com^[13], Theoldreader.Com^[14], численні розширення для браузерів, а також додатки для Windows – QuiteRSS^[15], RSS Bandit^[16], FeedReader^[17], FeedDemon^[18]. Поштові клієнти типу Microsoft Outlook^[19], Mozilla Thunderbird^[20] мають функціонал для завантаження RSS-підписок.^[9]

Усі ці продукти дають можливість завантажувати і переглядати RSS-стрічки. На жаль, для економії трафіку в RSS-стрічках немає повних текстів новин, лише короткий опис та посилання.

Також описані продукти не дають можливості завантажувати інформацію у придатному для подальшої автоматичної обробки вигляді. По суті, вони просто є інтерфейсом для читання матеріалів.

Для вирішення задачі скрейпінгу створено чимало різних додатків, сервісів та фреймворків. Усі вони дають можливість у тому чи іншому вигляді запрограмувати спосіб завантаження даних з певної веб-сторінки. Одним з найвідоміших сервісів такого роду є ParseHub^[21], а також його найбільші конкуренти – Scrapy, import.io, Portia. Можна використати утиліти, призначені для емуляції роботи браузера, на кшталт Selenium Web Driver. Також є чимало додатків до веб-браузерів, що виконують схожі функції, але не є достатньо гнучкими, щоб задовольнити задані вимоги.

ParseHub – це онлайн сервіс, що дає можливість у прямо у веб-браузері задати формат даних, що необхідно зібрати, налаштувати агент для збору даних і запустити його. Результати зберігаються на серверах ParseHub, а отримати доступ до них можна, експортувавши дані в Excel, JSON або з власного додатку за допомогою наявного у сервісі API.^[21]

Нижче наведено переваги ParseHub.

- а) Можливості налаштування роблять інструмент гнучким та дають повний контроль над тим, які дані та як будуть витягнуті.
- б) Багатий вбудований інструментарій дасть раду навіть складним сучасним веб-сторінкам з динамічним контентом.
- в) Результати роботи можна переглядати наживо та відповідно змінювати модель даних.
- г) Можна використовувати API проекту та завантажувати результати;
- д) Хороші посібники користувача, оформлені у вигляді додатків до браузера.
- е) Відмінна технічна підтримка – команда розробників відкрита до діалогу та співпраці.

Нижче наведено недоліки ParseHub.

- а) Крута крива навчання – необхідно затратити деякий час, щоб опанувати інструмент.
- б) Динамічні результати сильно сповільнюють роботу на великих обсягах даних.
- в) При запуску свого проекту на сервері результати не відображаються у браузері, що сповільнює розробку.
- г) Немає API для отримання усіх результатів одним викликом.
- д) Порівняно великий час виконання запитів.
- е) Немає кешування результатів, а отже і доступу до них, якщо вихідний сайт недоступний. ^[10]
- ж) Безкоштовна версія є обмеженою та повільною. Платна ж версія стартує від \$150 в місяць, що є достатньо дорого. ^[21]

Scrapy – це безкоштовний проект з відкритим вихідним вихідним кодом, розроблений на мові Python. Scrapy – це фреймворк, що дозволяє створювати свої. Scrapy використовує HTTP-запити до веб сторінок, щоб отримати їх HTML-код, після чого є можливість розбирати HTML за

допомогою XPath-запитів, зберігати результат у файлову систему або в базу даних у вигляді регулярної структури. Перевагою Scrapy є вже готова інфраструктура для забору даних з веб-сторінок, а також реалізація на мові Python, що автоматично означає доступ до широкої та розвиненої інфраструктури бібліотек Python. Для Scrapy реалізований сервер Scrapyd, також із відкритим вихідним кодом. Перевагою, як і недоліком Scrapy є використання ним протоколу HTTP напряду, що робить роробку більш гнучкою, але й трудомісткою. Це також зумовлює ще один великий недолік – Scrapy не виконує клієнтський javascript-код, за допомогою якого багато сайтів генерують свій вміст. [22]

Selenium Web Driver – це безкоштовний фреймворк, призначений для написання автоматизованих тестів для веб-додатків. Через своє основне призначення цей фреймворк має багато корисних функцій для навігації веб-сайтами, у тому числі перехід по сторінках за допомогою емуляції натискань на елементи веб-інтерфейсу, вибір елементів інтерфейсу за її CSS-класом, XPath, повну навігацію DOM-деревом сторінки. Selenium повністю емулює роботу веб-переглядача [23] Така емуляція означає, що коректно виконується весь javascript-код, а веб-сайти в принципі не можуть відрізнити робота, що збирає вміст сайтів, від справжнього користувача.

Отже, усі згадані вище програмні продукти мають свої переваги та недоліки, але загалом, із деякими доопрацюваннями можуть задовольнити заданим вимогам.

1.4 Аналіз вимог до програмного забезпечення

Після аналізу технічного завдання було розроблено модель варіантів використання системи, зображену на рисунку 1.2. Опираючись на варіанти використання було розроблено вимоги до системи.

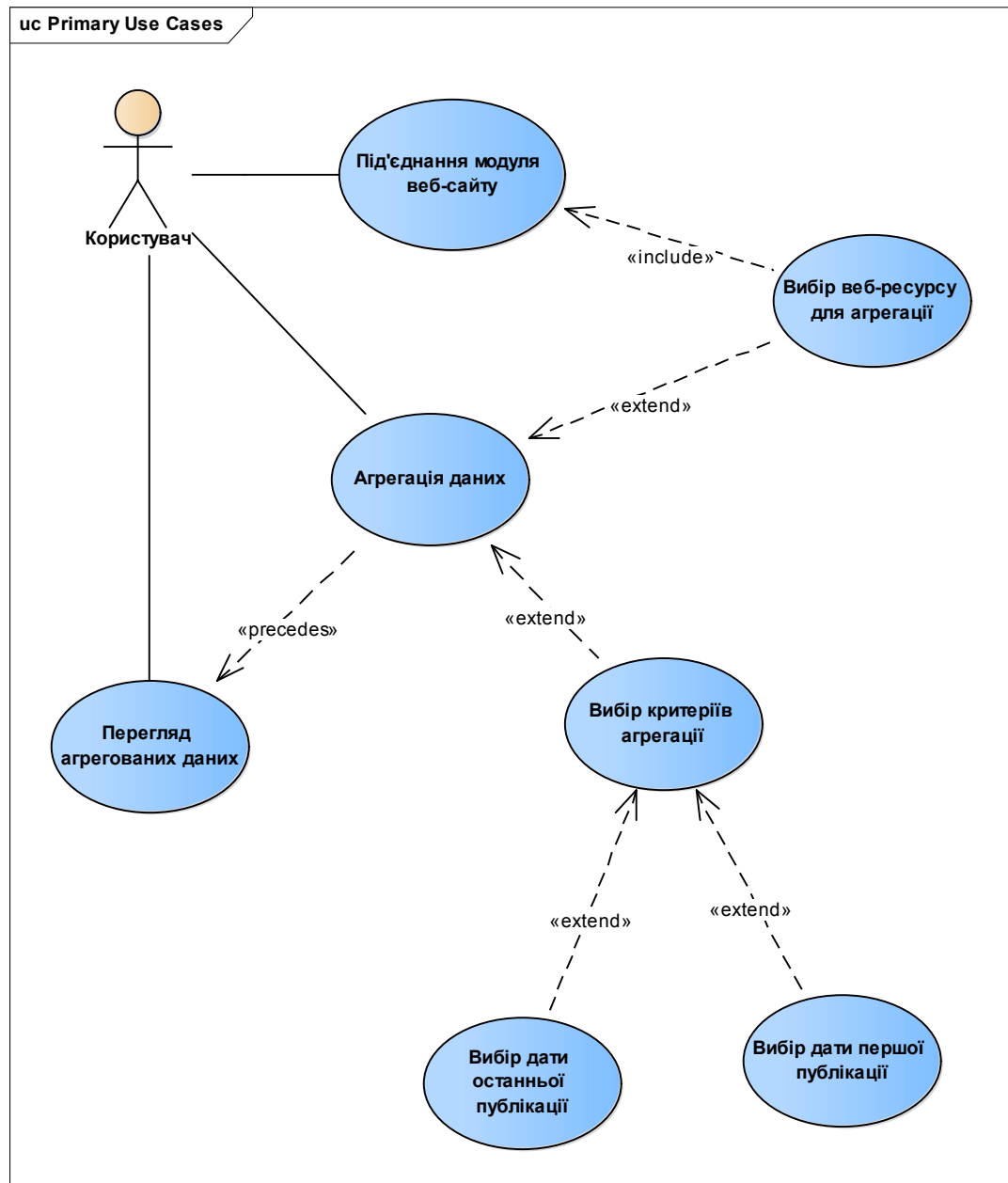


Рисунок 1.2 – Схема структурна варіантів використання

1.4.1 Розроблення функціональних вимог

Програмне забезпечення для агрегації матеріалів інтернет-видань повинна забезпечувати виконання наступних функцій:

- обрання періоду для збереження матеріалів;
- обрання джерела або джерел матеріалів;
- перегляд збережених матеріалів;
- експорт збережених матеріалів до зовнішнього сховища даних;

– створення та підключення додатків (модулів) для завантаження матеріалів з інших джерел.

Функціональні вимоги (див. рисунок 1.3) напряду зв'язані із прецедентами варіантів використання. Їх описано у схемі трасування вимог, яку зображено на рисунку 1.4.

На рисунку 1.5 зображена матриця трасування вимог. Вона допомагає визначити джерело будь-яких вимог та їх залежність від найвищого рівня до найнижчого.

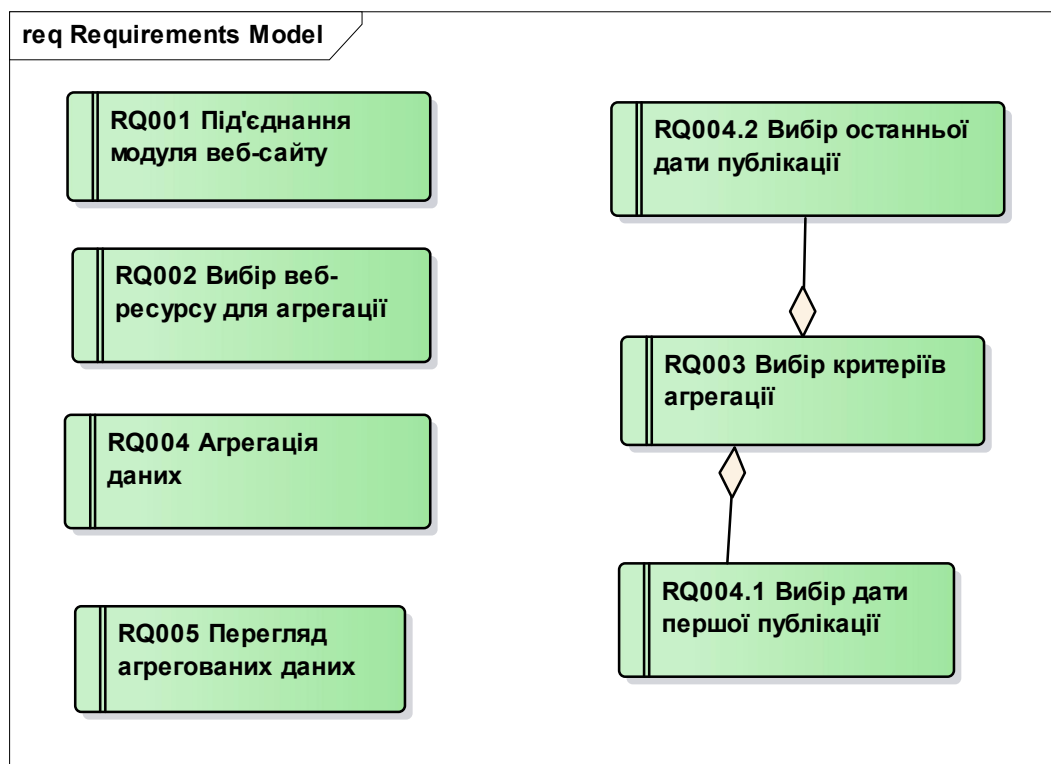


Рисунок 1.3 – Схема структурна вимог

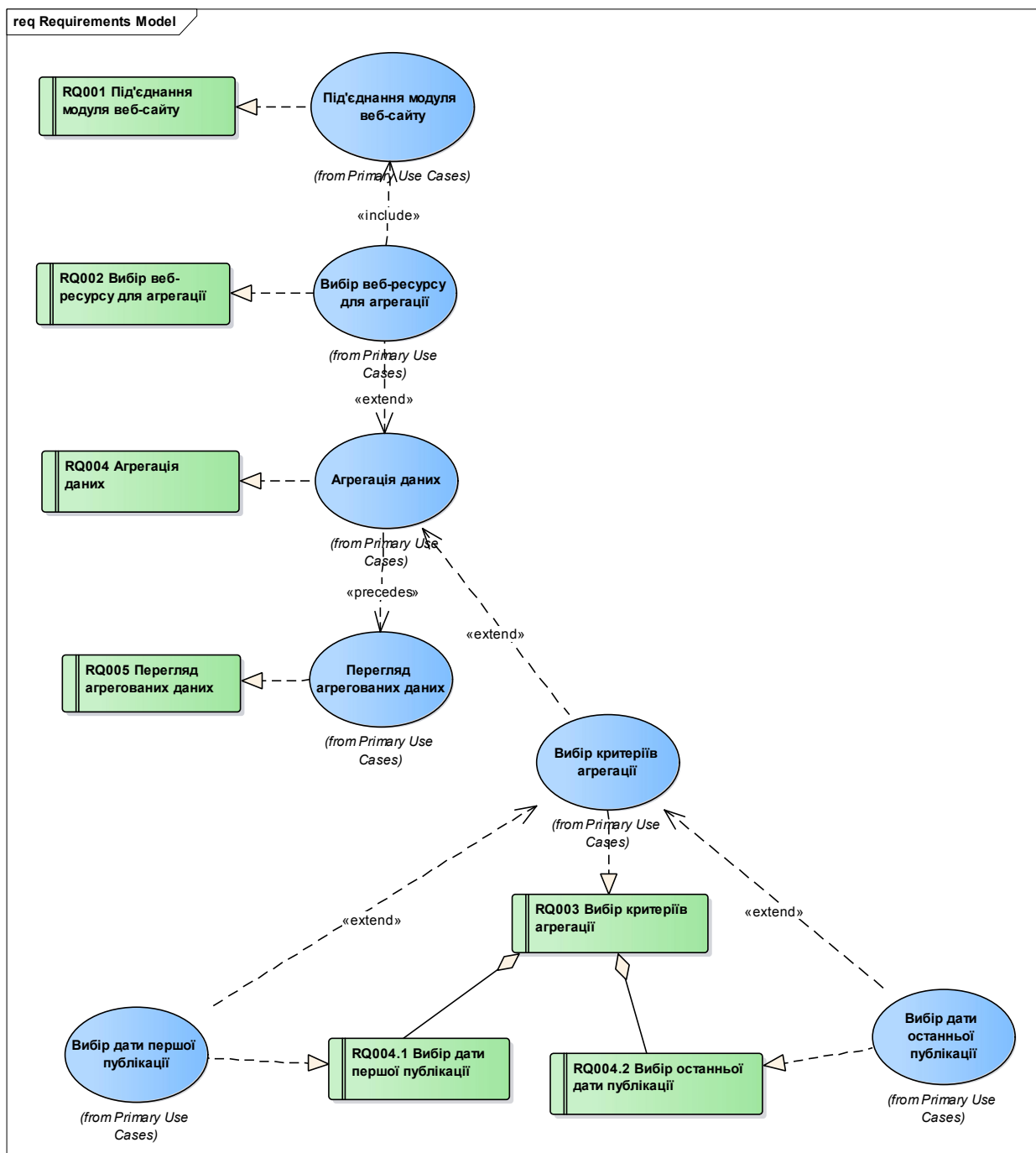


Рисунок 1.4 – Схема структурна трасування вимог

Специфікацію вимог та детальний опис кожного прецеденту наведено нижче у формі таблиць опису прецедентів.

У таблиці 1.1 наведено опис прецеденту «Під'єднання модуля веб-сайту».

Таблиця 1.1 – UC001 «Під'єднання модуля веб-сайту»

Дійові особи	Користувач
Передумова	Модуль веб-сайту не під'єднано
Сценарій	1. Користувач копіює модуль веб-сайту у папку модулів. 2. Система завантажує модуль.
Альтернативний сценарій	3.1. При невірному типі модулю система виводить на екран повідомлення про помилку 3.2. Система продовжує роботу, але модуль не завантажується.
Постумова	Успішне завершення: у списку сайтів з'являється новий сайт. Невдале завершення: модуль не завантажено.

У таблиці 1.2 наведено опис прецеденту «Вибір веб-ресурсу для агрегації». Сценарій цього прецеденту містить лише одну дію – обрання користувачем веб-ресурсу для агрегації.

Таблиця 1.2 – UC002 «Вибір веб-ресурсу для агрегації»

Дійові особи	Користувач
Передумова	Систему запущено. Модулі веб-ресурсів завантажено.
Сценарій	1. Користувач обирає веб-ресурс для агрегації зі списку.
Альтернативний сценарій	3.1. Якщо модуль веб-ресурсу завантажено неправильно, система видає помилку та аварійно завершує роботу.
Постумова	Успішне завершення: є можливість запустити агрегацію даних.

У таблиці 1.3 наведено опис прецедента «Вибір критеріїв агрегації».

Таблиця 1.3 – UC003 «Вибір критеріїв агрегації»

Дійові особи	Користувач
Передумова	Систему запущено.
Сценарій	1. Користувач обирає критерій агрегації. 2. Користувач задає значення критерія агрегації.
Альтернативний сценарій	3.1. Якщо користувач увів неправильне значення критерія, система видає повідомлення про помилку. 3.2 Користувач натискає кнопку «ОК», і продовжує роботу.
Постумова	Успішне завершення: критерії агрегації встановлено.

У таблиці 1.4 наведено опис прецедента «Вибір дати першої публікації».

Таблиця 1.4 – UC004 «Вибір дати першої публікації»

Дійові особи	Користувач
Передумова	Систему запущено. Користувач обрав критерій «Дата першої публікації»
Сценарій	1. Користувач задає значення дати першої публікації.
Альтернативний сценарій	3.1. Якщо користувач увів неправильне значення дати, система видає повідомлення про помилку. 3.2 Користувач натискає кнопку «ОК», і продовжує роботу.
Постумова	Успішне завершення: дату першої публікації встановлено.

У таблиці 1.5 наведено опис прецедента «Вибір дати останньої публікації».

Таблиця 1.5 – UC005 «Вибір дати першої публікації»

Дійові особи	Користувач
Передумова	Систему запущено. Користувач обрав критерій «Дата останньої публікації»
Сценарій	1. Користувач задає значення дати останньої публікації.
Альтернативний сценарій	3.1. Якщо користувач увів неправильне значення дати, система видає повідомлення про помилку. 3.2 Користувач натискає кнопку «ОК», і продовжує роботу.
Постумова	Успішне завершення: дату останньої публікації встановлено.

У таблиці 1.6 наведено опис прецедента «Агрегація даних».

Таблиця 1.6 – UC006 «Агрегація даних»

Дійові особи	Користувач, система
Передумова	Систему запущено.
Сценарій	1. Користувач обирає пункт «Агрегація даних». 2. Система завантажує встановлені значення критеріїв агрегації. Якщо критерії не встановлені, вони ігноруються.
Альтернативний сценарій	3.1. У випадку будь-якої помилки система видає повідомлення та аварійно завершує роботу.
Постумова	Успішне завершення: виведено повідомлення про успішну агрегацію даних веб-ресурсів.

У таблиці 1.7 наведено опис прецедента «Перегляд агрегованих даних»

Таблиця 1.7 – UC007 «Перегляд агрегованих даних»

Дійові особи	Користувач
Передумова	Систему запущено. Було проведено агрегацію даних.
Сценарій	1. Користувач обирає пункт «Перегляд агрегованих даних». 2. Система відображає деталі агрегованих даних.
Альтернативний сценарій	3.1. У випадку будь-якої помилки система видає повідомлення та аварійно завершує роботу.
Постумова	Успішне завершення: показано тексти агрегованих статей.

Source	Requirements Model::RQ001 Під'єднання модуля веб-сайту	Requirements Model::RQ002 Вибір веб-ресурсу для агрегації	Requirements Model::RQ003 Вибір критеріїв агрегації	Requirements Model::RQ004 Агрегація даних	Requirements Model::RQ004.1 Вибір дати першої публікації	Requirements Model::RQ004.2 Вибір останньої дати публікації	Requirements Model::RQ005 Перегляд агрегованих даних
Requirements Model::RQ001 Під'єднання модуля веб-сайту							
Requirements Model::RQ002 Вибір веб-ресурсу для агрегації							
Requirements Model::RQ003 Вибір критеріїв агрегації							
Requirements Model::RQ004 Агрегація даних							
Requirements Model::RQ004.1 Вибір дати першої публікації			↑				
Requirements Model::RQ004.2 Вибір останньої дати публікації			↑				
Requirements Model::RQ005 Перегляд агрегованих даних							

Рисунок 1.5 – Матриця трасування вимог

1.4.2 Розроблення нефункціональних вимог

Для зручного використання, програмне забезпечення для агрегації матеріалів Інтернет-видань повинне задовольняти наступним нефункціональним вимогам:

- система повинна забезпечувати повноту отриманих даних;
- система повинна бути відмовостійкою;
- сховище даних системи повинне надавати можливість одночасного доступу до нього багатьох користувачів;
- сховище даних системи повинне бути доступним незалежно від самої системи;
- наявність графічного інтерфейсу;
- наявність інтерфейсу командного рядка;
- наявність документованого інтерфейсу програмного забезпечення для розробки модулів агрегації.

1.4.3 Постановка комплексу задач модулю

На основі наведених вимог, опишемо комплекс задач, що повинен виконувати програмний продукт для агрегації матеріалів Інтернет-видань.

- а) Приймати модулі для агрегації різних веб-ресурсів.
- б) Завантажувати тексти статей заданих Інтернет-видань.
- в) Давати можливість перегляду завантажених даних.

Висновки по розділу

У першому розділі обґрунтовано потребу у розробці програмного забезпечення для агрегації матеріалів Інтернет-видань, розглянуто технічні та програмні рішення, що існують на сьогодні. Проведено детальний опис та аналіз предметної області. Наведено діаграму варіантів використання із їх описом. Наведено опис прецедентів, їх зв'язок із функціональними

вимогами. Наведено структурну схему вимог до програмного забезпечення.
Показано матрицю трасування вимог та таблицю трасування вимог.

					ІАЛЦ.045430-02-81	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		