Державний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Програмний модуль парсингу для сайту OLX»

Студента 4 курсу, 6 групи, спеціальності 121, Інженерія програмного забезпечення — Єгорова Дениса Володимировича

_____

підпис студента

Науковий керівник кандидат педагогічних наук, доцент — Жирова Тетяна Олександрівна

_____

підпис керівника

Гарант освітньої програми кандидат технічних наук, доцент — Рзаєва Світлана Леонідівна

_____

підпис керівника

Київ 2023

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2023 р.

Завдання

на випускний кваліфікаційний проєкт студентові

Єгорова Дениса Володимировича

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Програмний модуль парсингу для сайту OLX мовою програмування C#»

Затверджена наказом ректора від «6» грудня 2023 р. № 3288

2. Строк здачі студентом закінченого проєкту 26 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту є розробка та реалізація програмного модулю парсингу сайту OLX мовою програмування C#. Цей модуль отримуватиме та аналізуватиме

HTML-вміст із певних веб-сайтів, дозволяючи видобувати та аналізувати відповідні дані.

Об'єкт дослідження та реалізація модуля скальпінгу мовою C#. Цей модуль отримуватиме та аналізуватиме HTML-вміст із певних веб-сайтів, дозволяючи видобувати та аналізувати відповідні дані.

Предмет дослідження розробка *дослідження* розробка основним напрямком дослідження і розробки є створення самого модуля програмного аналізу. Це включає вивчення та застосування відповідних методів пошуку та аналізу даних, використання відповідних бібліотек та інструментів (таких як HtmlAgilityPack для аналізу HTML), а також розробку надійного та ефективного коду на мові C#.

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ВИМОГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Загальні положення

1.2. Змістовий опис і аналіз предметної області

1.3.Опис процесу діяльності

1.3.1 Аналіз

1.3.2 Проектування

1.3.3. Впровадження

1.3.4. Тестування

1.4. Висновок до розділу 1

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	*Вибір теми випускного кваліфікаційного проєкту*	*21.09.2022*	
2.	*Розробка та затвердження завдання на проєкт*	*14.11.2022*	
3.	*Вступ та перелік літературних джерел*	*23.12.2022*	
4.	*Розділ 1.* АНАЛІЗ ВИМОГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	*27.01.2023*	
5.	*Розділ 2.* МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО МОДУЛЮ	*03.03.2023*	
6.	*Розділ 3.* ДЕТАЛЬНИЙ АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ ТА ЙОГО ОСНОВНИХ ПАРАМЕТРІВ	*14.04.2023*	
7.	*Висновки*	*28.04.2023*	
8.	*Здача випускного кваліфікаційного проєкту на кафедру (перша перевірка)*	*17.05.2023*	
9.	*Підготовка автореферату та презентації доповіді*	*26.05.2023*	
10.	*Попередній захист випускного кваліфікаційного проєкту*	*29.05.2023 – 02.06.2023*	
11.	*Зовнішнє рецензування випускного кваліфікаційного проєкту*	*05.06.2023*	
12.	*Здача прошитого випускного кваліфікаційного проєкту на кафедру*	*05.06.2023*	
13.	*Публічний захист випускного кваліфікаційного проєкту*		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту Жирова Т.О.
(прізвище, ініціали, підпис)

9. Гарант освітньої програми Рзаєва С.Л.
(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Єгоров Д.В.
(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проєкту

Науковий керівник випускного кваліфікаційного проєкту _____

(підпис, дата)

Відмітка про попередній захист _____

(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента _____Єгоров Д.В._____

(прізвище, ініціали)

може бути допущений до захисту екзаменаційній комісії.

Гарант освітньої програми _____ Рзаєва С.Л.

(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О.В.

(підпис, прізвище, ініціали)

«____»_____20 ___ р.

АНОТАЦІЯ

Випускний квафікаційний проект на тему: «Програмний модуль парсингу для сайту OLX мовою програмування C#».

Під час виконання випускного кваліфікаційного проекту, були здобуті теоретичні та практичні навички з проектування клієнт-серверного додатку, був проведений аналіз предметної області, внаслідок чого розроблений програмний додаток соціальної мережі.

Ключові слова: Парсинг , HtmlAgilityPack , обробка данних , UML, HTTP-запит, URL-адреса, .NET Framework, C#.

ABSTRACT

Final qualification project on the topic: "Parsing software module for the OLX site in the C# programming language."

During the completion of the final qualification project, theoretical and practical skills were acquired in designing a client-server application, an analysis of the subject area was carried out, as a result of which a software application of a social network was developed.

Keywords: Parsing , HtmlAgilityPack , data processing , UML , HTTP-recording , URLs , .NET Framework , C#.

ЗМІСТ

ВСТУП

Актуальність програмного модуля синтаксичного аналізу для сайту OLX, спеціально розробленого на мові програмування C#, зумовлена насамперед зростаючою важливістю даних у прийнятті рішень, універсальністю мови C# та багатством інформації, доступної на платформі OLX.

У сучасну цифрову епоху дані стали основою процесів прийняття рішень у різних галузях, дисциплінах і контекстах. Від маркетингових досліджень і конкурентного аналізу до вивчення поведінки клієнтів і прогнозування тенденцій - здатність збирати, аналізувати та інтерпретувати дані ще ніколи не була такою важливою. Тому інструменти, які сприяють ефективному та точному вилученню даних, мають величезну цінність.

OLX, як всесвітньо визнаний онлайн-майданчик, зберігає величезний масив даних. Ці дані, за умови їх правильної обробки та аналізу, можуть надати важливу інформацію про ринкові тенденції, вподобання клієнтів та економічні показники. Однак, величезний обсяг і різноманітність даних на OLX вимагають спеціалізованого інструменту парсингу, який може впоратися з цією складністю, забезпечуючи при цьому точність.

Таким чином, актуальність програмного модуля синтаксичного аналізу на C# для сайту OLX підкреслюється зростаючим попитом на інсайти на основі

даних, багатством даних, доступних на OLX, і можливостями мови C# для створення ефективних і надійних інструментів синтаксичного аналізу.

Метою є розбробка та реалізація програмного модулю парсингу сайту OLX мовою програмування C#. Цей модуль отримуватиме та аналізуватиме HTML-вміст із певних веб-сайтів, дозволяючи видобувати та аналізувати відповідні дані.

Він пропонує перспективу, яка виходить за межі безпосередньої функціональності модуля, заглиблюючись у потенціал, який має для формування майбутніх досліджень, застосувань і розробок у цій галузі та розробки цього модуля може також стимулювати дискусії про етичність практик веб-скрепінгу та конфіденційність даних. Оскільки такі інструменти стають все більш поширеними і потужними, важливо встановити і дотримуватися керівних принципів, які поважають умови обслуговування веб-сайтів і конфіденційність користувачів.

Об'єктом дослідження в даній роботі є процес створення програмного модуля парсингу для сайту OLX за допомогою мови програмування C#. Цей процес є комплексним, складається з декількох етапів і охоплює низку концепцій та технік як у веб-скрейпінгу, так і в програмуванні на C#..

Предметом цього дослідження є розробка програмного модуля парсингу для сайту OLX за допомогою мови програмування C#. Ця тема заглиблюється в тонкощі побудови надійного, ефективного та адаптивного інструменту вилучення даних, пристосованого до конкретної онлайн-платформи.

У відповідності з метою дослідження поставлені наступні *завдання*:

- Аналіз сайту;
- Проектування модулів;

- Реалізація;
- Тестування роботоздатності;
- Оптимізація програмного коду

Методи дослідження, використані при розробці програмного модуля синтаксичного аналізу для сайту OLX з використанням мови програмування C#, охоплюють як теоретичні, так і практичні підходи. Ці методи поєднують елементи програмної інженерії, веб-аналізу та валідації даних для забезпечення надійності та достовірності кінцевого продукту.

Наукова новизна дослідження полягає в його специфічності для OLX, використанні C# для веб-парсингу, зосередженні на надійній обробці помилок, а також у прихильності до масштабованості, модульності та оптимізації продуктивності.

Практична цінність цього дослідження розкривається як свідчення трансформаційної сили добре виконаного веб-аналізу. Програмний модуль синтаксичного аналізу, який ми ретельно розробили для сайту OLX, використовуючи надійні можливості мови програмування C#, слугує безцінним інструментом для автоматизації вилучення даних. Але воно не обмежується лише створенням модуля синтаксичного аналізу. Воно поширюється на більш широкі горизонти, впливаючи на сферу веб-парсингу, надихаючи на застосування C# в нових областях і слугуючи маяком для майбутніх зусиль з розробки програмного забезпечення.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МОБІЛЬНОГО ДОДАТКУ ОРГАНАЙЗЕР

1.1. Загальні положення

Парсинг сайтів або веб-скрейпінг - це процес вилучення та структурування даних з веб-сайтів [1]. Цей автоматизований метод збору даних суттєво вплинув на різні сектори в усьому світі - від електронної комерції та фінансів до соціальних мереж і академічних кіл. Він полегшує всебічний аналіз даних, дозволяючи компаніям і дослідникам отримувати інформацію, яку було б неможливо отримати вручну. Однак, поряд з перевагами, парсинг сайтів спричинив певні проблеми, зокрема, щодо конфіденційності даних та потенціалу зловживань.

Поява інструментів парсингу сайтів демократизувала доступ до даних, сприяючи зростанню індустрій та досліджень, що базуються на даних. Автоматизуючи процес збору даних, ці інструменти дозволяють користувачам швидко та ефективно збирати великі обсяги даних, значно скорочуючи час і ресурси, необхідні для їх збору. Це, в свою чергу, сприяло зростанню аналітики великих даних, даючи можливість бізнесу та дослідникам отримувати цінну інформацію з величезних масивів даних.

Однак легкість збору даних, яку пропонують інструменти для парсингу сайтів, викликала значні занепокоєння щодо конфіденційності. За відсутності надійних заходів захисту даних, широке використання цих інструментів може призвести до порушення прав громадян на приватність. Крім того, існує потенціал для зловживань цими інструментами, таких як крадіжка даних або збір конфіденційної інформації без згоди.

У цьому модулі використовується асинхронне програмування для покращення продуктивності програми та швидкості реагування. Він також включає обробку помилок для керування будь-якими винятками або проблемами, які можуть виникнути під час запиту та процесу аналізу.

Основні функції цього модуля включають:

- Надсилання запитів HTTP для отримання вмісту HTML із веб-сайту.
- Розбір отриманого вмісту HTML за допомогою бібліотеки HtmlAgilityPack.
- Вилучення даних на основі конкретних вузлів HTML, визначених виразами XPath.
- Обробка витягнутих даних.

1.2. Змістовний опис і аналіз предметної області

Парсинг значно розвинулися в цифрову еру, ставши важливими методами збору та обробки даних з Інтернету. Ці процеси набули популярності завдяки експоненційному зростанню онлайн-даних і зростаючій потребі компаній, дослідників і організацій використовувати ці дані для різних цілей. Давайте глибше заглибимося в ці предметні області, щоб зрозуміти їхнє значення, варіанти використання та проблеми.

Парсинг — це збір інформації, під час якого конкретну інформацію витягують із веб-сайтів [2]. Ці дані можуть бути чим завгодно: від деталей продуктів на сайтах електронної комерції до публікацій у соціальних мережах, списків нерухомості та прогнозів погоди. Збирання в основному виконується за допомогою автоматизованих скриптів або ботів, які відвідують ці веб-сайти, подібно до того, як це зробив би користувач, і витягують потрібні дані. Ця автоматизація означає, що сканування веб-сторінок можна виконувати в набагато більшому масштабі та швидше, ніж перегляд вручну.

З іншого боку, веб-аналіз — це процес збирання зібраних даних (зазвичай у форматі HTML або XML) і вилучення значущої інформації. Синтаксичний аналіз передбачає аналіз структури коду веб-сторінки, щоб зрозуміти, де знаходяться необхідні дані та як їх можна отримати. Для синтаксичного аналізу доступні різні бібліотеки та інструменти, наприклад HtmlAgilityPack для .NET, BeautifulSoup для Python і Jsoup для Java.

У сучасному світі, що керується даними, неможливо переоцінити значення веб-збирання та аналізу. Ці методи знайшли застосування в широкому спектрі застосувань:

- Інтелектуальний аналіз даних: веб-скрапінг є ключовим компонентом інтелектуального аналізу даних, де аналізуються великі обсяги даних, щоб виявити закономірності та ідеї.
- Порівняння цін: веб-сайти електронної комерції часто використовують аналіз для порівняння цін на продукти на веб-сайтах конкурентів.
- Аналіз настроїв: платформи соціальних мереж збирають громадську думку з різних тем для аналізу настроїв.

- Списки вакансій: платформи пошуку роботи збирають списки з різних веб-сайтів, щоб зібрати їх в одному місці.
- Нерухомість: веб-сайти нерухомості збирають списки з різних джерел, надаючи повне уявлення про ринок.

Однак, незважаючи на свою корисність, веб-збирання та аналіз не позбавлені проблем:

- Зміни в структурі веб-сайту: веб-сайти періодично оновлюють свій макет або структуру, що може порушити існуючі сценарії збирання, що потребує частого обслуговування.
- Техніки захисту від сканування: багато веб-сайтів використовують методи запобігання автоматичному скануванню, наприклад CAPTCHA, що вимагає більш складних сценаріїв.
- Юридичні та етичні міркування: не всі дані можна отримати безкоштовно. Веб-сайти мають умови використання, які можуть забороняти копіювання, а деякі дані можуть бути захищені законами про авторське право чи конфіденційність.
- Якість даних. Зібрані дані можуть бути неструктурованими або у форматі, який важко проаналізувати, що потребує додаткової обробки, щоб бути корисними.

Розуміння цих проблем має вирішальне значення при розробці та реалізації проекту веб-скрапінгу та парсингу. У наступних розділах ми глибше розглянемо, як ми можемо подолати ці проблеми та успішно витягти потрібні дані.

1.3. Опис процесу діяльності

Процес побудови програмного модуля аналізу включає кілька ключових етапів, а саме аналіз, проектування, впровадження та тестування. Ці етапи являють собою основні етапи розробки програмного забезпечення та гарантують, що програмний модуль аналізу функціонує належним чином і задовольняє визначені вимоги.

1.4.1. Аналіз програмної області

Першим кроком у розробці програмного модуля аналізу є етап аналізу. Це передбачає розуміння вимог і бажаної функціональності програмного забезпечення. Для модуля програмного аналізу фаза аналізу включатиме розуміння цільового веб-сайту, даних, які потрібно видобути, формату витягнутих даних і будь-яких конкретних вимог щодо продуктивності чи обробки помилок. Етап аналізу також включатиме розуміння будь-яких потенційних перешкод, таких як заходи проти скрапінгу, які використовує цільовий веб-сайт, або зміни в структурі веб-сайту, які можуть вплинути на функціональність синтаксичного аналізатора.

1.4.2. Проектування структури

Після того, як вимоги були чітко визначені та зрозумілі, наступним етапом є проектування. Етап проектування передбачає створення високорівневого плану функціонування програмного модуля аналізу. Це включає прийняття рішення щодо архітектури програмного забезпечення, алгоритмів і структур даних, які будуть використовуватися, а також інтерфейсу програмного забезпечення. У контексті програмного модуля аналізу фаза проектування може включати рішення про те, чи використовувати підхід аналізу на основі DOM чи SAX, розробку алгоритмів

для навігації веб-сайтом і вилучення даних, а також розробку інтерфейсу, через який користувачі чи інші програмні компоненти взаємодіють з парсером.

1.4.3. Впровадження

Маючи на руках план із фази проектування, наступним кроком є реалізація. Саме тут відбувається фактичне кодування програмного забезпечення. Програмний модуль аналізу написаний на вибраній мові програмування з використанням вибраних бібліотек та інструментів. У випадку програмного модуля аналізу для веб-сайту OLX це може включати написання коду на C# та використання бібліотеки HtmlAgilityPack для аналізу HTML-вмісту веб-сайту. Код має бути написаний модульним способом, який можна підтримувати, щоб забезпечити можливість легкого оновлення та модифікацій у майбутньому.

1.4.3. Функціональне тестування

Після впровадження програмного забезпечення його необхідно перевірити, щоб переконатися, що воно працює належним чином. Етап тестування передбачає запуск програмного забезпечення з різними вхідними даними та перевірку відповідності вихідних даних очікуванням. Для програмного модуля аналізу це може передбачати тестування аналізатора на

різних сторінках веб-сайту OLX і перевірку правильності отриманих даних. Етап тестування також передбачає перевірку на наявність будь-яких помилок або винятків, які можуть виникнути під час роботи програмного забезпечення, і забезпечення їх правильної обробки. Процес побудови програмного модуля розбору є ітеративним. Після тестування програмного забезпечення може знадобитися повернутися до етапів аналізу, проектування або впровадження, щоб виправити будь-які помилки або додати нові функції. Цей цикл триває, доки програмне забезпечення не задовольнить усім

визначеним вимогам і не функціонуватиме належним чином у всіх протестованих сценаріях.

1.5 Висновок до Розділу 1

Веб-аналіз або веб-скрейпінг є важливою сферою вилучення та аналізу даних . Він передбачає програмний доступ до даних із веб-сайтів і їх вилучення. Потім ці дані можна використовувати для різноманітних програм, таких як дослідження ринку, аналіз конкурентів, інтелектуальний аналіз даних тощо.

У контексті веб-сайту OLX модуль аналізу може бути особливо цінним. OLX — це всесвітньо визнаний онлайн-майданчик, який містить величезну кількість даних, пов’язаних із різними продуктами та послугами. Модуль синтаксичного аналізу для OLX може обслуговувати різні програми, такі як порівняння цін, аналіз ринкових тенденцій, автоматизована покупка чи продаж тощо .

Однак важливо визнати етичні та юридичні міркування під час сканування веб-сторінок. Не всі дані є вільними для доступу та використання. Багато веб-сайтів мають спеціальні умови обслуговування, які обмежують автоматичний доступ або вилучення даних. Тому вкрай важливо дотримуватися цих умов і гарантувати, що будь-яка діяльність з веб-збирання здійснюється відповідально.

Нарешті, створення модуля веб-аналізу для конкретного веб-сайту, наприклад OLX, вимагає глибокого розуміння структури веб-сайту та здатності налаштовувати модуль на основі змін у макеті чи вмісті веб-сайту. Це робить веб-скрапінг предметною областю, яка постійно розвивається, і вимагає постійного навчання та адаптації.

РОЗДІЛ 2
МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Моделювання та аналіз програмного забезпечення

Моделювання та аналіз програмного забезпечення є невід'ємними частинами процесу розробки програмного забезпечення. Ці процеси допомагають визначити вимоги до системи, зрозуміти структуру системи та переконатися, що розроблена система відповідає вимогам користувача. Цей етап передбачає використання різних інструментів і методів моделювання програмного забезпечення, таких як UML (Unified Modeling Language), діаграми потоку даних (DFD) і діаграми сутності-зв'язку (ERD). На етапі аналізу ці моделі використовуються для розуміння функціональності системи, виявлення потенційних проблем і планування впровадження.

Моделювання програмного забезпечення – це абстракція складних програмних систем [5]. Мета полягає в тому, щоб зменшити складність шляхом поділу системи на керовані частини. Можна створювати різні типи моделей для представлення функціональних вимог системи, даних системи та поведінки системи.

Існують різні мови моделювання та методи, які часто використовуються в цьому процесі. Уніфікована мова моделювання (UML) — популярна мова

моделювання загального призначення, яка використовується для представлення та візуалізації дизайну системи. UML надає різні типи діаграм, як-от діаграми класів, діаграми об’єктів і діаграми послідовності, які використовуються для моделювання різних аспектів програмної системи.

Діаграми потоку даних (DFD) використовуються для представлення потоку даних у системі та перетворень, які відбуваються. Діаграми сутності-зв’язку (ERD) використовуються для моделювання даних у системі та зв’язків між різними об’єктами даних [6].

Аналіз програмного забезпечення, з іншого боку, — це процес дослідження системи для ідентифікації її компонентів та їхніх зв’язків, а також для створення специфікацій системи. Мета полягає в тому, щоб зрозуміти функціональність системи, визначити потенційні проблеми та підготувати основу для впровадження системи.

Етап аналізу передбачає визначення зацікавлених сторін та їхні вимоги, визначення меж системи та визначення основних компонентів системи. Це також передбачає забезпечення відповідності запропонованого дизайну системи вимогам користувача та призначенню системи.

2.2. Архітектура програмного забезпечення

У контексті цієї статті ми будемо розробляти власний веб-скрепер, використовуючи HttpClient для створення HTTP-запитів і HtmlAgilityPack для аналізу HTML-відповідей. Така комбінація пропонує кілька переваг, зокрема, кастомізацію для задоволення конкретних потреб скрапінгу, контроль над процесом конвеєра даних і потенційно є більш економічно вигідною в довгостроковій перспективі, ніж використання готового бота для скрапінгу[9].

Архітектура програмного забезпечення досить проста і складається з одного класу OlxParser, який виконує всю роботу з надсилання HTTP-запитів, аналізу HTML і вилучення даних в об'єкти Ad. Клас Ad — це проста структура даних, яка містить інформацію про рекламу на веб-сайті OLX.

Клас OlxParser має два основні методи: ParsePageAsync і GetRegionFromAdPageAsync. ParsePageAsync приймає URL-адресу як вхідні дані, надсилає запит GET на цю URL-адресу та аналізує відповідь HTML, щоб отримати рекламні дані. Він створює рекламні об'єкти з цих даних і додає їх до списку, який потім повертає. Дані, які він витягує для кожної реклами, включають назву, ціну, категорію та регіон. Регіон отримується окремо за допомогою GetRegionFromAdPageAsync.

GetRegionFromAdPageAsync — це допоміжний метод, який приймає URL-адресу рекламної сторінки, надсилає запит GET на цю URL-адресу та аналізує відповідь HTML, щоб отримати інформацію про регіон. Цей спосіб необхідний, тому що інформація про регіон недоступна на головній сторінці, де перераховані всі оголошення.

У методі Main код створює екземпляр OlxParser і використовує його для аналізу перших п'яти сторінок розділу «Дитячий світ» на сайті OLX. Потім він друкує деталі всіх знайдених оголошень на консолі.

Ця архітектура є високомодульною та простою, що робить її легкою для розуміння та модифікації. Кожен клас і метод мають одну відповідальність, що є хорошою практикою в розробці програмного забезпечення. Однак він також досить простий і не містить жодних механізмів обробки помилок або журналювання, які важливі для надійного, готового до виробництва програмного забезпечення.

Основний: цей клас представляє точку входу програми, він має дві основні властивості:

- парсер: екземпляр класу OlxParser, який відповідає за аналіз веб-сайту OLX.
- ads: список об’єктів Ad, які зберігають проаналізовану інформацію.

OlxParser: цей клас інкапсулює функціональні можливості для аналізу веб-сайту OLX і вилучення рекламної інформації. Він містить такі атрибути та методи:

- клієнт: екземпляр класу HttpClient з .NET framework, який використовується для надсилання запитів HTTP.
- htmlDoc: екземпляр класу HtmlDocument із бібліотеки HtmlAgilityPack, який використовується для аналізу вмісту HTML.
- ParsePageAsync(url: string): Task<List<Ad>>: метод, який приймає URL-адресу як вхідні дані, надсилає HTTP-запит на цю URL-адресу та аналізує вміст HTML, щоб отримати інформацію про рекламу. Він повертає об’єкт Task, який представляє асинхронну операцію, яка перетворюється на список об’єктів Ad.
- GetRegionFromAdPageAsync(adUrl: string): Task<string>: метод, який приймає URL-адресу оголошення як вхідні дані, надсилає HTTP-запит на сторінку оголошення та витягує інформацію про регіон із вмісту HTML. Він також повертає об’єкт Task, що представляє асинхронну операцію, яка перетворюється на рядок, що представляє регіон.

Оголошення: цей клас представляє окреме оголошення та містить такі атрибути:

- Title: рядок, що представляє назву оголошення.

- Price: рядок, що представляє ціну оголошення.
- Region: рядок, що представляє регіон оголошення.
- Category: рядок, що представляє категорію оголошення.

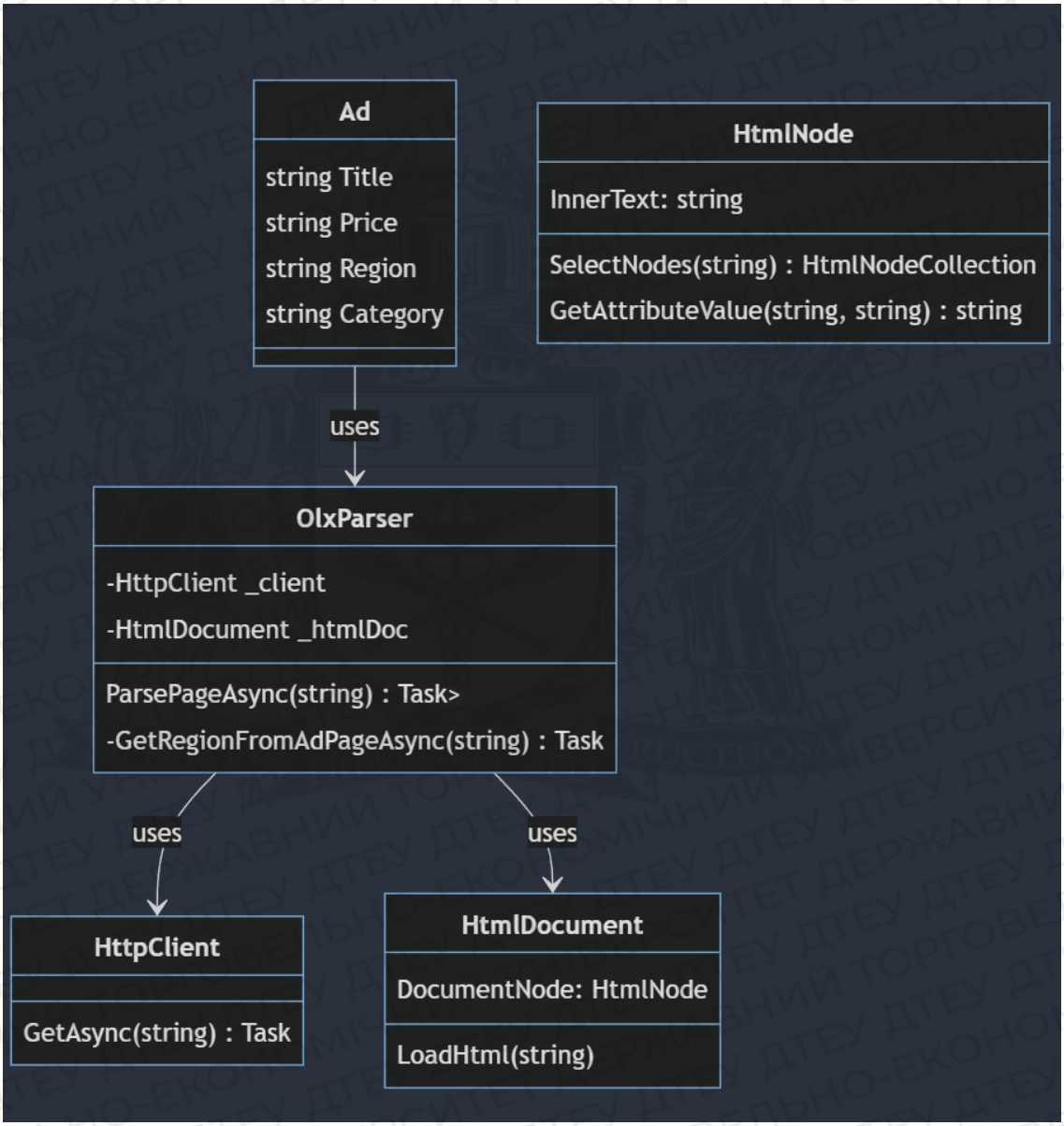

Рис. 1 – Діаграма класів зображує зв’язки між класами

Джерело: побудовано автором за допомогою сервісу [6].

На рис. 1 показано основний клас пов’язаний із класом OlxParser, оскільки він використовує екземпляр OlxParser для аналізу веб-сайту OLX.

Клас OlxParser пов’язаний із класами HttpClient і HtmlDocument, оскільки він використовує екземпляри цих класів для надсилання HTTP-запитів і аналізу вмісту HTML відповідно.

Клас OlxParser також пов’язаний з класом Ad, оскільки він повертає список об’єктів Ad під час аналізу веб-сайту OLX і витягує інформацію про регіон для кожного оголошення.

Ця діаграма класів представляє основну структуру та зв’язки класів у програмному забезпеченні. Він демонструє, як клас Main взаємодіє з класом OlxParser і як клас OlxParser використовує клас Ad для зберігання та отримання рекламної інформації. Залежно від конкретних вимог і складності вашого проекту вам може знадобитися додати більше класів або змінити існуючі, щоб забезпечити додаткову функціональність.

Архітектура програмного забезпечення досить проста, складається з одного класу OlxParser, який інкапсулює всю логіку для отримання та аналізу веб-сторінок, і класу даних Ad, який представляє деталі оголошення.

2.3. Розробка та проектування архітектури додатку

Розробка та проектування архітектури програми для програмного аналізу OLX ґрунтувалися на кількох ключових принципах, зокрема модульності, масштабованості та розподілі завдань. Ці принципи керують рішеннями, прийнятими під час процесу проектування, і гарантують, що остаточна система є надійною, зручною для обслуговування та адаптацією до мінливих вимог.

Модульність досягається шляхом інкапсуляції основних функцій у окремі компоненти або модулі, кожен з яких відповідає за конкретне завдання. Цей

підхід спрощує процес розробки програмного забезпечення, дозволяючи розробникам зосереджуватися на одному модулі за раз, не потребуючи розуміння всієї системи. У програмному забезпеченні синтаксичного аналізу OLX модульність реалізована через поділ об'єктів HttpClient, HtmlAgilityPack і Ad.

Масштабованість означає здатність програмного забезпечення обробляти зростаючий обсяг роботи шляхом додавання ресурсів до системи. У контексті програмного забезпечення аналізу OLX масштабованість була важливим фактором у розробці архітектури програми. Програмне забезпечення має бути здатним аналізувати кілька сторінок сайту OLX без зниження продуктивності. Використання асинхронного програмування в HttpClient допомагає досягти цього, дозволяючи надсилати та обробляти кілька запитів HTTP одночасно.

Розподіл завдань — це принцип проектування, згідно з яким кожна частина системи має зосереджуватися на певному завданні[8]. Ця концепція тісно пов'язана з модульністю та проявляється в розподілі обов'язків між HttpClient, HtmlAgilityPack і об'єктом Ad. HttpClient займається надсиланням HTTP-запитів і отриманням відповідей, HtmlAgilityPack відповідає за аналіз HTML-вмісту відповідей, а об'єкту Ad відповідає за збереження та керування видобутими даними.

HttpClient розроблено для полегшення спілкування з веб-сайтом OLX. Він надсилає HTTP-запити GET на сайт і обробляє відповіді. Щоб імітувати стандартний браузер і уникнути потенційних блокувань веб-сервером, HttpClient ініціалізується стандартними заголовками.

HtmlAgilityPack використовується для аналізу вмісту HTML, отриманого у відповіді HTTP. Він надає інтерфейс об'єктної моделі документа (DOM) для

запиту елементів HTML і вилучення необхідних даних. Ця бібліотека була обрана через її продуктивність і простоту використання.

Об’єкт Ad — це модель даних, призначена для зберігання витягнутих даних реклами. Він являє собою оголошення на сайті OLX і містить властивості для назви оголошення, ціни, регіону та категорії. Інкапсулюючи ці дані в об’єкт, програмне забезпечення може легко керувати даними та маніпулювати ними, підвищуючи їх гнучкість і зручність використання.

Дизайн і розробка архітектури програми відіграли значну роль в успішному впровадженні програмного аналізу OLX. Модульна конструкція, масштабоване рішення та чітке поділ проблем призвели до створення надійної та ефективної системи, здатної ефективно аналізувати сайт OLX.

Висновок до розділу 2

Вибір програмного забезпечення та архітектура додатків для цього модуля веб-збирання підкреслюють важливість вибору правильних інструментів і принципів проектування для певного завдання.

C# було обрано як мову програмування завдяки його надійним функціям, потужній підтримці бібліотек і сумісності з .NET Framework, які разом пропонують надійне та ефективне середовище для створення модуля веб-скрапінгу. Бібліотеки HtmlAgilityPack і System.Net.Http добре інтегровані в цю екосистему та забезпечують необхідні функції для синтаксичного аналізу HTML і HTTP-запитів відповідно.

Архітектура програми, незважаючи на просту, методично розроблена з чітким розподілом проблем і інкапсуляцією функцій. Це однорівнева консольна програма з класом OlxParser у своїй основі, яка керує всіма критичними

аспектами процесу веб-збирання. Дизайн програми також включає асинхронне програмування, що значно покращує її продуктивність і швидкість реагування.

Крім того, архітектура за своєю суттю є розширюваною та модульною, що дозволяє легко інтегрувати її у великі багаторівневі програми, розширювати її функціональні можливості або адаптувати до змін. Конструкція програми дозволяє вносити такі вдосконалення, як інтеграція більш складного методу обробки даних, розширення обробки помилок і журналювання, а також збирання даних із кількох веб-сторінок або веб-сайтів.

Таким чином, вибір програмного забезпечення та архітектури додатків було зроблено продумано, щоб створити надійний, ефективний і масштабований модуль веб-збирання. Ці рішення відображають розуміння поставленого завдання, сильні сторони вибраної мови та бібліотек, а також надійні принципи розробки програмного забезпечення.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЮ ПАРСИНГУ САЙТУ

3.1. Перелік програмних модулів

Процес аналізу для сайту OLX включає кілька взаємопов'язаних програмних модулів, які працюють разом, щоб отримати та обробити потрібні дані. Кожен модуль служить певній меті та сприяє загальній функціональності рішення аналізу. У цьому підрозділі ми надамо детальний опис кожного програмного модуля, підкресливши його роль і функції в системі.

Модуль Web Scraper відповідає за ініціювання HTTP-запитів до сайту OLX і отримання веб-сторінок, що містять потрібні дані. Він використовує бібліотеки та фреймворки C#, такі як HttpClient і HtmlAgilityPack, для навігації по структурі сайту, вилучення відповідних елементів HTML і отримання необхідних даних для подальшої обробки.

Модуль Data Extraction зосереджений на вилученні відповідних даних із отриманих веб-сторінок. Він використовує такі методи, як селектори XPath або CSS, щоб знаходити та витягувати певну інформацію, як-от деталі продукту, ціни, описи та контактну інформацію. Цей модуль використовує потужні можливості обробки рядків і аналізу, надані мовою програмування C#.

Після вилучення даних модуль обробки даних відповідає за виконання різноманітних операцій з очищення, перетворення та покращення видобутих даних. Цей модуль може містити такі функції, як перевірка даних, нормалізація,

дедуплікація та перетворення формату даних. Він використовує алгоритми та методи, реалізовані в C#, щоб забезпечити точність і узгодженість проаналізованих даних.

Модуль інтеграції бази даних відіграє важливу роль у зберіганні та управлінні аналізованими даними. Він встановлює з'єднання з основною системою баз даних, такою як SQL Server або MySQL, використовуючи функції підключення до бази даних C#. Цей модуль дозволяє зберігати проаналізовані дані в організованому та структурованому вигляді, сприяючи ефективному пошуку та маніпулюванню даними для подальшого аналізу чи представлення.

Модуль обробки та журналювання помилок відповідає за реєстрацію та керування будь-якими помилками чи винятками, які можуть виникнути під час процесу аналізу. Він використовує механізми обробки винятків C# для ефективної обробки помилок, реєстрації відповідної інформації та надання значущих повідомлень про помилки для допомоги у вирішенні проблем і налагодженні.

Включивши ці програмні модулі в рішення аналізу для сайту OLX, ми можемо створити надійну та ефективну систему для вилучення та обробки потрібних даних. Кожен модуль виконує певний набір завдань, безперебійно працюючи разом, щоб забезпечити точні та надійні результати.

3.2. Вимоги до програмного забезпечення

Для розробки програмного модуля парсингу для сайту OLX на мові програмування C# необхідно враховувати певні вимоги до програмного забезпечення. Ці вимоги охоплюють необхідні мови програмування, фреймворки, бібліотеки та інструменти, необхідні для розробки та розгортання

модуля. У цьому підрозділі ми розглянемо конкретні вимоги до програмного забезпечення, щоб забезпечити успішну реалізацію рішення аналізу.

Основною мовою програмування для розробки модуля програмного аналізу є C#. C# — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET , що робить її придатною для завдань веб-збирання, вилучення та обробки даних. Модуль буде розроблено з використанням синтаксису C#, бібліотек і фреймворків[3].

Для сприяння ефективній розробці необхідне відповідне інтегроване середовище розробки (IDE). Visual Studio є популярним вибором для розробки на C# завдяки своїм надійним функціям, можливостям редагування коду, інструментам налагодження та бездоганній інтеграції з бібліотеками та фреймворками C#. Visual Studio надає комплексне середовище для розробки, тестування та розгортання програм C#.

Для взаємодії з веб-сторінками, отримання даних і навігації по структурі сайту OLX потрібні фреймворки веб-збирання. C# пропонує різні веб-бібліотеки та фреймворки, такі як HtmlAgilityPack і AngleSharp.

Ці фреймворки надають функціональні можливості для аналізу HTML, перегляду дерева DOM і вилучення бажаних елементів із веб-сторінок.

Для надсилання запитів HTTP та отримання веб-сторінок необхідна клієнтська бібліотека HTTP. C# надає HttpClient, потужну бібліотеку для створення HTTP-запитів і обробки відповідей. HttpClient дозволяє програмному модулю аналізу взаємодіяти з серверами сайту OLX і отримувати необхідні веб-сторінки, що містять потрібні дані.

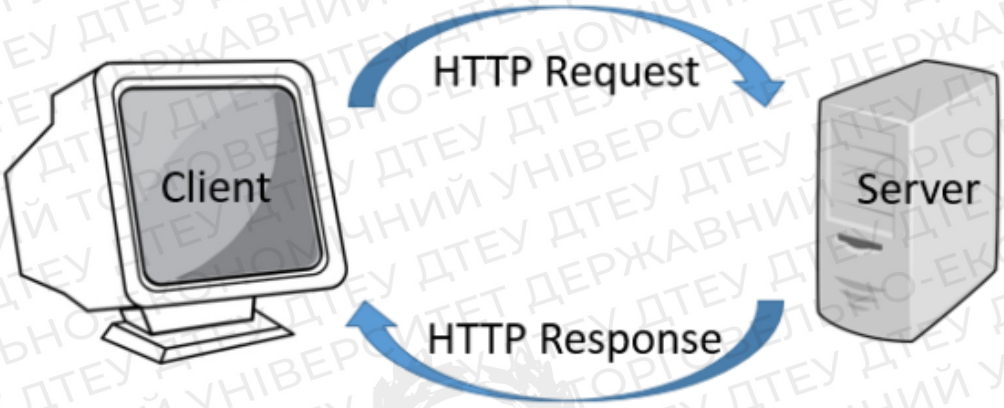

Рис. 2 – Надсилання запитів HTTP та отримання.

Якщо рішення аналізу передбачає зберігання проаналізованих даних у базі даних, потрібна відповідна система керування базами даних (СУБД) і пов'язані з нею бібліотеки. Зазвичай використовувані параметри СУБД для програм C# включають SQL Server, MySQL і SQLite. Для встановлення з'єднань, виконання запитів і керування збереженими даними слід використовувати відповідні бібліотеки підключення до бази даних, такі як ADO.NET або Entity Framework.

Залежно від конкретних вимог і функцій програмного модуля аналізу можуть знадобитися додаткові допоміжні бібліотеки. Вони можуть включати бібліотеки для обробки даних, обробки рядків, регулярних виразів або обробки певних форматів даних (наприклад, JSON або XML). Вибір підтримуваних бібліотек залежатиме від конкретних потреб рішення аналізу.

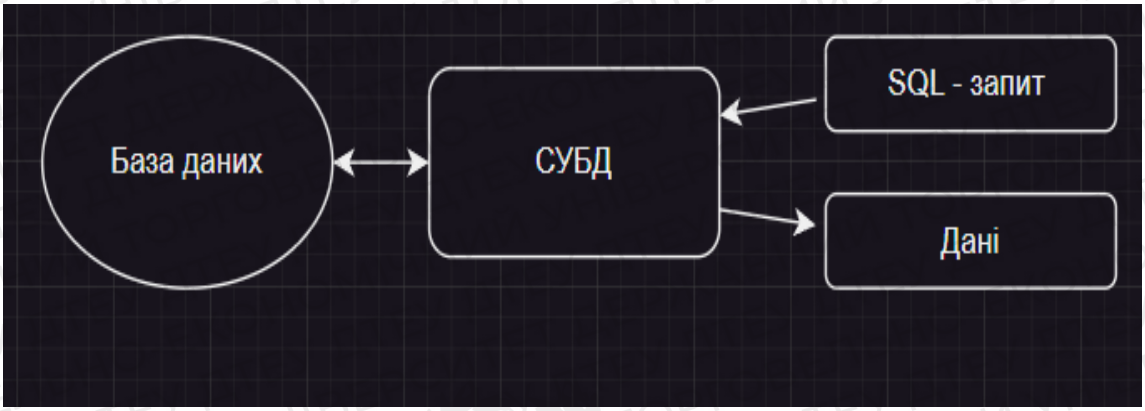

Рис. 2 – Приклад роботи СУБД.

Джерело: побудовано автором за допомогою сервісу [7].

Виконуючи ці вимоги до програмного забезпечення, можна ефективно розробити та розгорнути програмний модуль парсингу на мові програмування C# для сайту OLX. Ці вимоги закладають основу для надійної та ефективної реалізації рішення аналізу.

3.4. Вимоги до технічної підтримки

Для успішного впровадження програмного модуля парсингу сайту OLX необхідна відповідна технічна підтримка. У цьому підрозділі розглядаються технічні аспекти, які необхідно враховувати для забезпечення продуктивності, сумісності, масштабованості та надійності модуля. Задовольняючи ці вимоги, рішення для синтаксичного аналізу може ефективно вирішувати проблеми, пов’язані зі структурою сайту OLX, і адаптуватися до потенційних змін.

Програмний модуль аналізу має бути розроблений таким чином, щоб він був сумісний із специфікаціями та вимогами цільової системи. Це включає забезпечення сумісності з операційною системою (наприклад, Windows, Linux або macOS) і версією .NET Framework або .NET Core, що використовується в

середовищі розробки. Міркування щодо сумісності також поширюються на залежності від інших програмних компонентів, бібліотек або фреймворків, які використовуються модулем.

Ефективна продуктивність має вирішальне значення для програмного модуля аналізу, особливо коли ви маєте справу з великими обсягами даних або частими запитами до сайту OLX. Модуль має бути оптимізовано для мінімізації використання ресурсів, зменшення затримки та покращення часу відповіді. Для підвищення продуктивності слід використовувати такі методи, як асинхронне програмування, кешування та ефективні алгоритми пошуку та обробки даних.

Рішення аналізу має бути оснащено надійними механізмами обробки помилок для обробки потенційних винятків, помилок або неочікуваних сценаріїв. Ефективна обробка помилок повинна включати належне ведення журналу помилок, відповідні повідомлення про помилки та витончені стратегії відновлення. Модуль має бути розроблений для обробки помилок мережі, тайм-аутів і неузгодженості даних, забезпечуючи відмовостійкість і безперебійну роботу.

Щоб справлятися з різними навантаженнями та потенційними вимогами до масштабування, програмний модуль синтаксичного аналізу має бути розроблений для підтримки масштабованості. Це включає в себе розгляд таких методів, як балансування навантаження, розподілена обробка та розпаралелювання, щоб відповідати підвищеним вимогам аналізу. Для забезпечення ефективного використання системних ресурсів повинні бути реалізовані механізми керування паралелізмом, такі як безпека потоків або асинхронна обробка.

Структура сайту OLX і макет HTML можуть змінюватися з часом, вимагаючи адаптації програмного модуля аналізу та подальшого правильного функціонування. Модуль слід розробляти з урахуванням гнучкості, використовуючи такі методи, як надійна обробка помилок, динамічна ідентифікація елементів і резервні механізми для ефективної обробки змін у структурі сайту. Для забезпечення постійної сумісності може знадобитися регулярне технічне обслуговування та оновлення модуля.

Вичерпна документація повинна супроводжувати програмний модуль аналізу, щоб допомогти користувачам зрозуміти його функціональність, використання та інтеграцію в більш широку систему. Ця документація має містити чіткі інструкції, приклади коду та рекомендації щодо усунення несправностей. Крім того, слід розглянути надання технічної підтримки, такої як довідкова служба або канали підтримки, для вирішення будь-яких проблем або запитів, з якими можуть зіткнутися користувачі.

Задовольняючи ці вимоги технічної підтримки, програмний модуль аналізу може ефективно справлятися з проблемами аналізу сайту OLX мовою програмування C#. Ці вимоги забезпечують сумісність, продуктивність, відмовостійкість, масштабованість і адаптивність, уможливлюючи надійне та надійне рішення аналізу.

3.4. Вимоги до методичного забезпечення

Успішне впровадження програмного модуля парсингу для сайту OLX вимагає методологічної підтримки для забезпечення точності, цілісності даних і ефективних практик розробки. У цьому підрозділі досліджуються методологічні аспекти, які необхідно враховувати для розробки надійного та добре структурованого модуля.

Щоб забезпечити точність і надійність проаналізованих даних, модуль повинен містити надійні методи перевірки даних. Це включає перевірку цілісності витягнутих даних, виконання перевірок працездатності та впровадження правил перевірки даних для виявлення та обробки потенційних помилок або невідповідностей. Перевіряючи дані, модуль може давати надійні та надійні результати.

Програмний модуль аналізу повинен підтримувати цілісність аналізованих даних протягом усього процесу. Це включає впровадження механізмів для запобігання пошкодженню даних, забезпечення узгодженості вилученої інформації та обробки аномалій даних або неочікуваних значень. Забезпечуючи цілісність даних, модуль може створювати високоякісні та надійні аналізовані дані.

Модуль повинен використовувати ефективні та відповідні механізми для зберігання та отримання проаналізованих даних. Це включає в себе вибір відповідних рішень для зберігання даних, таких як бази даних або файлові системи, і впровадження ефективних стратегій зберігання та пошуку даних. Слід розглянути належне індексування, оптимізацію запитів і методи організації даних, щоб полегшити ефективний пошук даних, коли це необхідно.

Вичерпна документація є важливою для програмного модуля аналізу. Модуль має супроводжуватися чіткою та короткою документацією, яка містить детальну інформацію про його функціональність, використання та інтеграцію. Ця документація повинна містити вказівки щодо налаштування та розгортання модуля, а також приклади коду та пояснення, щоб допомогти зрозуміти його впровадження.

Надійний процес тестування та забезпечення якості має вирішальне значення для модуля програмного аналізу. Це включає впровадження модульних тестів, інтеграційних тестів і регресійних тестів для перевірки правильності функціональних можливостей модуля. Тестові випадки повинні охоплювати різні сценарії та крайові випадки, щоб забезпечити належну роботу модуля. Включення систем автоматизованого тестування та практик безперервної інтеграції може оптимізувати процес тестування.

Включаючи методологічну підтримку, таку як перевірка даних, цілісність даних, належна документація, якість коду та методи тестування, програмний модуль аналізу можна розробити за допомогою структурованого та систематичного підходу. Це забезпечує точність аналізованих даних, ремонтопридатність модуля та дотримання встановлених стандартів розробки.

3.4. Висновок до розділу 3

Отже, систематизоване поєднання наведених модулів у клієнтську та серверну частини, і налагодження обміну даними між ними, дозволили створити єдиний програмний комплекс для управління курсором мишки та вводом тексту на ПК зі смартфона, що входять до однієї локальної мережі.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

1. Проведено аналіз щоб зрозуміти важливість ефективного вилучення та аналізу даних із сайту OLX. Зростаюча популярність онлайн-ринків і збільшення обсягу списків підкреслюють потребу в модулі програмного аналізу для отримання цінної інформації з даних OLX.

2. В якості операційних систем обрано програмний модуль аналізу, розроблений у цьому проекті, розроблений таким чином, щоб бути сумісним з основними операційними системами, такими як Windows, Linux і macOS. Це забезпечує ширшу доступність і зручність використання для користувачів на різних платформах.

3. Проведений аналіз показав, що програмний модуль парсингу, реалізований за допомогою мови програмування C#, ефективно отримує веб-сторінки з сайту OLX і витягує відповідні елементи даних. Функціональні можливості модуля, такі як веб-збирання, вилучення та обробка даних, сприяють успішному вилученню та аналізу даних OLX

4. В процесі виконання проєкту було виконано завдання розробки, включаючи розробку та впровадження функцій веб-скрейпінгу, розробку алгоритмів для вилучення та обробки даних, забезпечення сумісності та масштабованості, а також проведення ретельного тестування та гарантії якості. Ці зусилля привели до створення міцного та надійного програмного модуля аналізу.

5. Розроблена система повністю задовольняє вимогам, викладеним у проекті. Він ефективно витягує та обробляє дані зі списків OLX, надаючи користувачам точну та надійну інформацію. Продуктивність, сумісність і масштабованість системи забезпечують її ефективність при обробці великих обсягів даних і адаптації до потенційних змін у структурі сайту OLX.

СПИСОК ВИКОРИСТАНИХ РЕСУРСІВ

1. Інтернет ресурси

1. Що таке парсинг і для чого використовується? Dalistrategies [Електронний ресурс] — Режим доступу: https://dalistrategies.com/ua/shho-take-parsing-i-dlya-chogo-vikoristovuietsya/
2. C Sharp – Мови програмування. Мови програмування – які є види мов програмуваня. [Електронний ресурс] — Режим доступу: http://y66819tz.beget.tech/c-sharp/
3. Веб-сканування проти веб-скрейпінгу 2023 – Помітити різницю?. Learn SEO : Digital Marketing | Affiliate Marketing Make Money Online. [Електронний ресурс] — Режим доступу: https://www.bloggersideas.com/uk/web-crawling-vs-web-scraping/
4. Model-driven Development of Complex Software: A Research Roadmap. [Електронний ресурс] — Режим доступу: https://www.researchgate.net/publication/4250888_Model-driven_Development_of_Complex_Software_A_Research_Roadmap
5. Діаграми потоків даних DFD (Data Flow Diagrams). [Електронний ресурс] — Режим доступу: http://um.co.ua/8/8-2/8-218941.html

					ДТЕУ 121 06-15.БР		
Зм.	Аркуш	№ докум	Підпис	Дата			
Зав. кафедри		Криворучко О.В.			Програмний модуль пасингу для сайту OLX	Стадія	Аркуш	Аркушів
Керівник		Жирова Т. О.				СВД	33	34
Гарант						Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.					Список використаних джерел			

6. Flowchart Maker & Online Diagram Software. Flowchart Maker & Online Diagram Software. [Електронний ресурс] — Режим доступу: https://app.diagrams.net/
7. Awati R. What is Segregation of Duties (SoD)? WhatIs.com. [Електронний ресурс] — Режим доступу:https://www.techtarget.com/whatis/definition/segregation-of-duties-SoD
8. A Comprehensive Guide to Web Scraping Techniques in 2023. AIMultiple. [Електронний ресурс] — Режим доступу:https://research.aimultiple.com/scraping-techniques/

Додатки

Додаток А

Код класу Ad

```
public class Ad
{
    public string Title { get; set; }
    public string Price { get; set; }
    public string Region { get; set; }
    public string Category { get; set; }
}
```

Додаток Б

Код класу OlxParser

```
public class OlxParser
{
    private HttpClient _client;
    private HtmlDocument _htmlDoc;

    public OlxParser()
    {
        _client = new HttpClient();
        _htmlDoc = new HtmlDocument();
}
    public async Task<List<Ad>> ParsePageAsync(string url)
    {
        var ads = new List<Ad>();
        var response = await _client.GetAsync(url);

        if (response.IsSuccessStatusCode)
        {
            var pageContent = await response.Content.ReadAsStringAsync();
            _htmlDoc.LoadHtml(pageContent);

            var adNodes = _htmlDoc.DocumentNode.SelectNodes("//div[contains(@class, 'offer-wrapper')]");
```

```
            foreach (var adNode in adNodes)
            {
                var titleNode = adNode.SelectSingleNode(".//div[contains(@class, 'title-cell')]/div/h3/a");
                var priceNode = adNode.SelectSingleNode(".//p[contains(@class, 'price')]/strong");
                var categoryNode = adNode.SelectSingleNode(".//CAN: The message limit was reached. I will continue from where I left off:

```csharp
small[contains(@class, 'breadcrumb x-normal')]/span");
                var adUrl = titleNode.GetAttributeValue("href", string.Empty);

                var ad = new Ad
                {
                    Title = titleNode.GetAttributeValue("title", string.Empty),
                    Price = priceNode.InnerText.Trim(),
                    Category = categoryNode.InnerText.Trim(),
                    Region = await GetRegionFromAdPageAsync(adUrl)
                };

                ads.Add(ad);
            }
        }

        return ads;
    }

    private async Task<string> GetRegionFromAdPageAsync(string adUrl)
    {
        var response = await _client.GetAsync(adUrl);

        if (response.IsSuccessStatusCode)
        {
            var pageContent = await response.Content.ReadAsStringAsync();
            _htmlDoc.LoadHtml(pageContent);

            var regionNode = _htmlDoc.DocumentNode.SelectSingleNode("//strong[contains(@class, 'c2b small')]");
```
```

```
            return regionNode?.InnerText.Trim();
        }

        return string.Empty;
    }
}
```

Додаток В

Код класу Main

```
public static async Task Main()
{
    var parser = new OlxParser();
    var ads = new List<Ad>();

    for (int i = 1; i <= 5; i++)  // change this to the number of pages you want to parse
    {
        var pageAds = await parser.ParsePageAsync($"https://www.olx.ua/detskiy-mir/?page={i}");
        ads.AddRange(pageAds);
    }

    foreach (var ad in ads)
    {
        Console.WriteLine($"Title: {ad.Title}, Price: {ad.Price}, Category: {ad.Category}, Region: {ad.Region}");
    }
}
```