



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №2
Технологія розроблення програмного забезпечення
«ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ
ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ.
КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»
Варіант 11

Виконала
студентка групи ІА-24
Тильна Марія Сергіївна

Перевірив:
Мягкий М.Ю.

Київ 2024р.

Зміст

1. Теоретичні відомості	2
2. Завдання	3
3. Діаграма прецедентів	4
4. Прецедент 1	5
5. Прецедент 2	6
6. Прецедент 3	7
7. Діаграма класів	9
8. Структура бази даних	10

Тема: Діаграма варіантів використання. Сценарії варіантів використання. Діаграми uml. Діаграми класів. Концептуальна модель системи

Теоретичні відомості

Діаграма прецедентів (Use Case Diagram)

Діаграма прецедентів використовується для візуалізації функціональних вимог до системи. Вона ілюструє, як користувачі (актори) взаємодіють із системою через конкретні сценарії використання (прецеденти). Основними елементами цієї діаграми є актори, прецеденти та зв'язки між ними.

Прецеденти дозволяють виявити основні функції системи та забезпечити їхнє розуміння на високому рівні.

Діаграма класів (Class Diagram)

Діаграма класів моделює статичну структуру системи. Вона демонструє класи, їх атрибути, методи (операції) та зв'язки між класами, такі як асоціації, агрегації, композиції та успадкування. Класи представляють основні компоненти системи, їхні характеристики (атрибути) та поведінку (методи), а зв'язки між ними показують, як ці класи взаємодіють.

База даних та її структура

База даних — це організований набір інформації, що зберігається у структурованому вигляді, зазвичай у формі таблиць. Таблиці складаються з рядків (записів) і стовпців (полів), де поля містять атрибути даних.

Структура бази даних визначає, як дані пов'язані між собою. Основні елементи бази даних включають таблиці, ключі (первинні та зовнішні), а також зв'язки між таблицями, які можуть бути одного з типів: один-до-одного, один-до-багатьох або багато-до-багатьох.

Шаблон Репозиторію (Repository Pattern)

Шаблон Репозиторію використовується для абстрагування доступу до даних. Він дозволяє працювати з базою даних через клас-репозиторій, що інкапсулює операції зі збереження, отримання, оновлення та видалення даних. Це знижує залежність між бізнес-логікою та логікою доступу до даних, роблячи систему більш гнучкою до змін або оновлень.

Завдання.

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.
3. Намалюйте діаграму класів для реалізованої частини системи.
4. Виберіть 3 прецеденти і напишіть на їх основі прецеденти.
5. Розробити основні класи і структуру системи баз даних.
6. Класи даних повинні реалізувати шаблон Репозиторію для взаємодії з базою даних.
7. Підготувати звіт про хід виконання лабораторних робіт. Звіт, що подається повинен містити: діаграму прецедентів, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних.

..11 Web crawler (proxy, chain of responsibility, memento, template method, composite, p2p)

Веб-сканер повинен вміти розпізнавати структуру сторінок сайту, переходити за посиланнями, збирати необхідну інформацію про зазначений термін, видаляти не семантичні одиниці (рекламу, об'єкти javascript і т.д.), зберігати знайдені дані у вигляді структурованого набору html файлів вести статистику відвіданих сайтів і метадані.

Хід роботи

1. Діаграма прецедентів

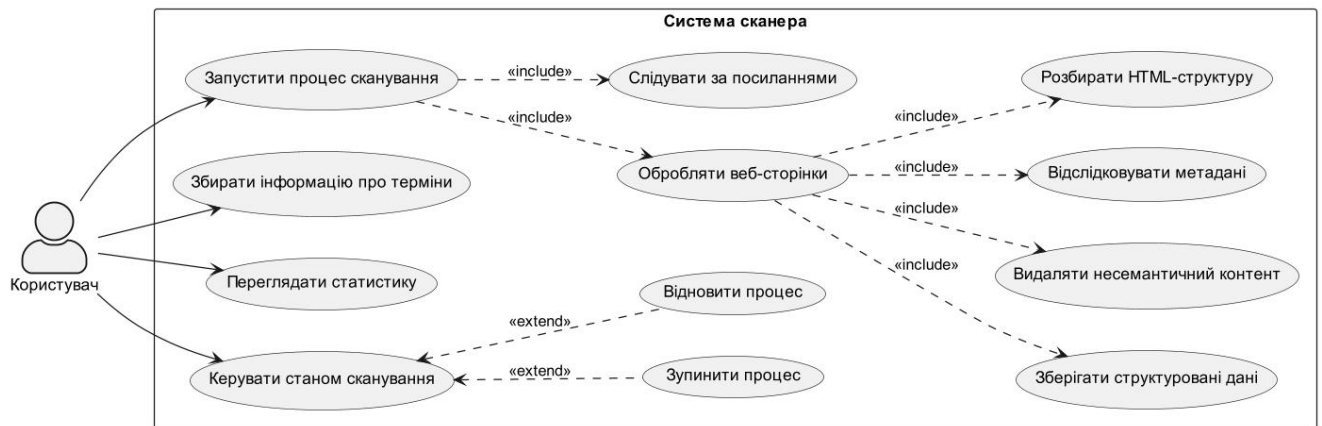


Рис.1. Діаграма прецедентів

1. Запускає процес сканування через функцію **"Запустити сканування"**, яка:
 - 1) Надсилає запити до веб-серверів.
 - 2) Обробляє отримані веб-сторінки.
 - 3) Використовує базу даних для збереження інформації.
2. Користувач може зберігати або відновлювати стан сканера.
3. Після завершення роботи користувач може переглядати статистику або експортувати дані.

Ця система підтримує модульний підхід, що полегшує її масштабування та розширення. Вона добре організована, і завдяки використанню шаблонів проєктування (як зазначалося раніше), її легко підтримувати й оновлювати.

Прецеденти використання веб-сканера

Прецедент 1: Запуск веб-сканування

Назва: Запуск веб-сканування

Актор: Користувач

Опис: Користувач ініціює процес сканування конкретного веб-сайту.

Передумови:

1. Система налаштована належним чином.
2. База даних ініціалізована.
3. Цільовий URL доступний.

Основний потік:

1. Користувач вказує початковий URL.
2. Користувач задає параметри сканування (глибина, максимальна кількість сторінок, кількість потоків).
3. Система перевіряє коректність URL.
4. Система ініціалізує компоненти сканера.
5. Система починає сканування веб-сайту.
6. Система обробляє кожну сторінку через ланцюжок процесорів:
 1. Витягує заголовок.
 2. Очищає та витягує основний вміст.
 3. Ідентифікує та перевіряє посилання.
 4. Збирає метадані.
7. Система зберігає оброблені сторінки у базу даних.
8. Сканування продовжується до досягнення встановлених меж або завершення сайту.
9. Система коректно завершує роботу.

Альтернативні потоки:

- **A1: Надано некоректний URL**

1. Система відображає повідомлення про помилку.
2. Повертається до кроку 1.

- **A2: Помилка мережі під час сканування**

1. Система реєструє помилку.
2. Продовжує обробку наступного URL у черзі.

- **A3: Помилка бази даних**

1. Система реєструє помилку.
2. Проводить спробу зберегти стан.
3. Повідомляє користувача про помилку.

Постумови:

1. Оброблені сторінки збережені у базу даних.
2. Статистика сканування оновлена.
3. Ресурси належним чином звільнені.

Прецедент 2: Збереження та відновлення стану сканера

Назва: Збереження та відновлення стану сканера

Актор: Користувач

Опис: Користувач зберігає поточний стан сканера і відновлює його пізніше.

Попередні умови:

1. Сканер ініціалізовано.
2. База даних доступна.

Основний потік:

1. Користувач запитує збереження поточного стану сканера.
2. Система фіксує поточний стан:
 1. Набір відвіданих URL.
 2. Черга очікуваних URL.
 3. Позначка часу.
3. Система серіалізує стан.
4. Система зберігає стан.
5. Користувач може пізніше запитати відновлення стану.
6. Система завантажує збережений стан.
7. Система повторно ініціалізує сканер із завантаженим станом.
8. Сканування продовжується з відновленої точки.

Альтернативні потоки:

- **A1: Не вдалося зберегти стан**
 1. Система реєструє помилку.
 2. Повідомляє користувача.
 3. Продовжує сканування.
- **A2: Не вдалося відновити стан**
 1. Система реєструє помилку.
 2. Пропонує розпочати сканування з нуля.

Постумови:

1. Стан сканера збережений або відновлений.
2. Сканування може продовжуватися з точки збереження.

Прецедент 3: Обробка та очищення веб-контенту

Назва: Обробка та очищення веб-контенту

Актор: Система (ланцюжок процесорів)

Опис: Система обробляє веб-сторінку через ланцюжок відповідальності.

Попередні умови:

1. Отримано сирий HTML-контент.
2. Ініціалізовано ланцюжок обробки.

Основний потік:

1. Система створює контекст обробки для сторінки.
2. **Процесор заголовків:**
 - Витягує заголовок сторінки.
 - Передає наступному процесору.
3. **Процесор контенту:**
 - Видаляє несемантичні елементи (рекламу, скрипти).
 - Витягує основний контент.
 - Передає наступному процесору.
4. **Процесор посилань:**

- Виявляє всі посилання.
- Перевіряє їхню валідність.
- Фільтрує посилання, що ведуть на інші домени.
- Передає наступному процесору.

5. Процесор метаданих:

- Витягує мета-теги.
 - Збирає дані OpenGraph.
6. Система створює фінальний об'єкт Page.
 7. Система зберігає оброблену сторінку в базі даних.

Альтернативні потоки:

• A1: Помилка обробки в процесорі

1. Реєструє помилку.
2. Пропускає до наступного процесора.
3. Позначає сторінку як частково оброблену.

• A2: Неприпустимий контент

1. Позначає сторінку як непридатну.
2. Реєструє проблему.
3. Пропускає подальшу обробку.

Постумови:

1. Сторінка оброблена та очищена.
2. Структуровані дані готові для збереження.
3. Посилання витягнуто для подальшого сканування.

2. Діаграма класів системи

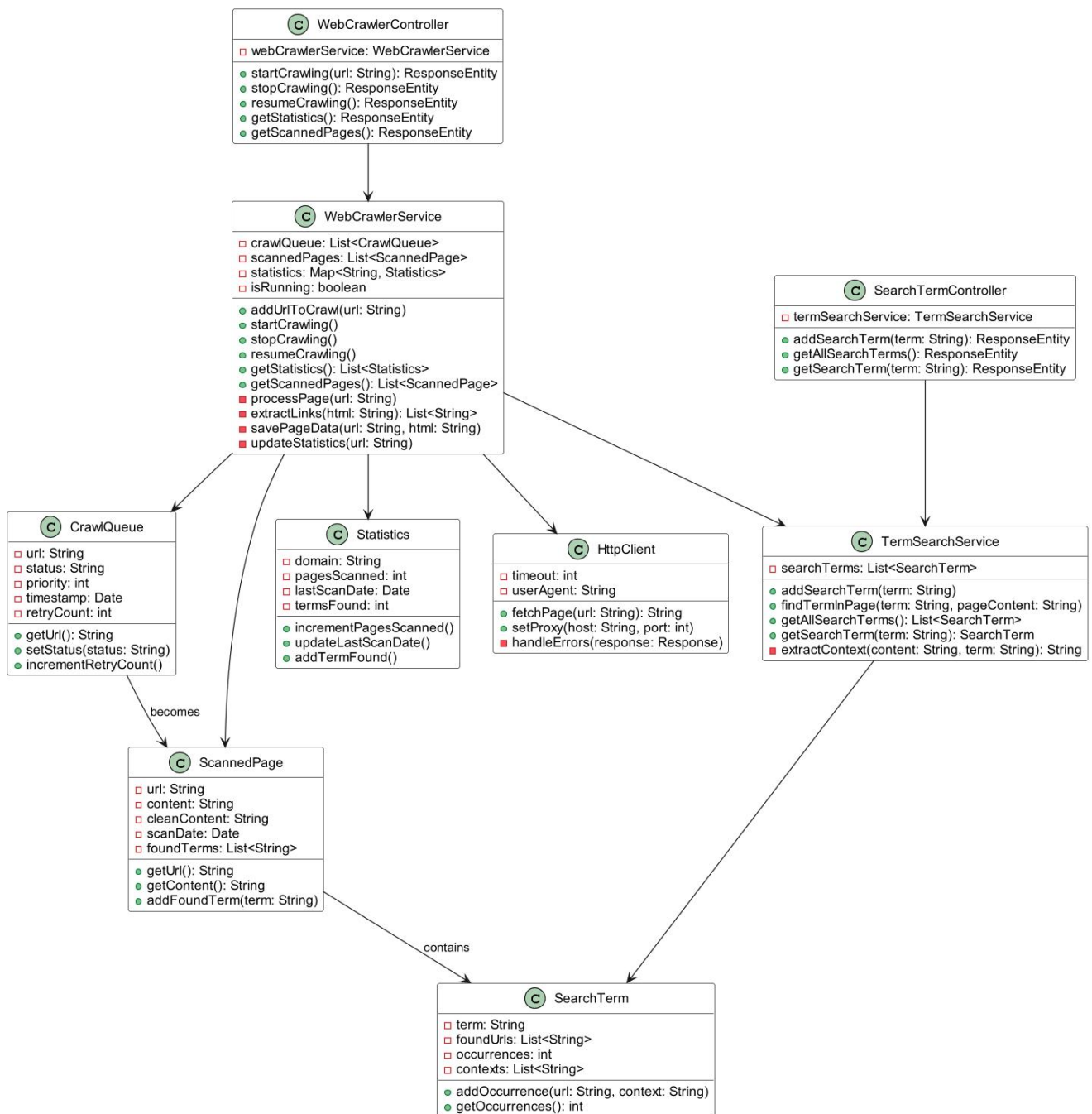


Рис.2. Діаграма класів

Ця діаграма класів відображає архітектуру системи веб-сканера, її ключові компоненти, взаємозв'язки між ними та механізми обробки даних. Вона побудована на основі принципів модульності та узагальнення, що сприяє гнучкості та масштабованості системи.

3. Структура бази даних

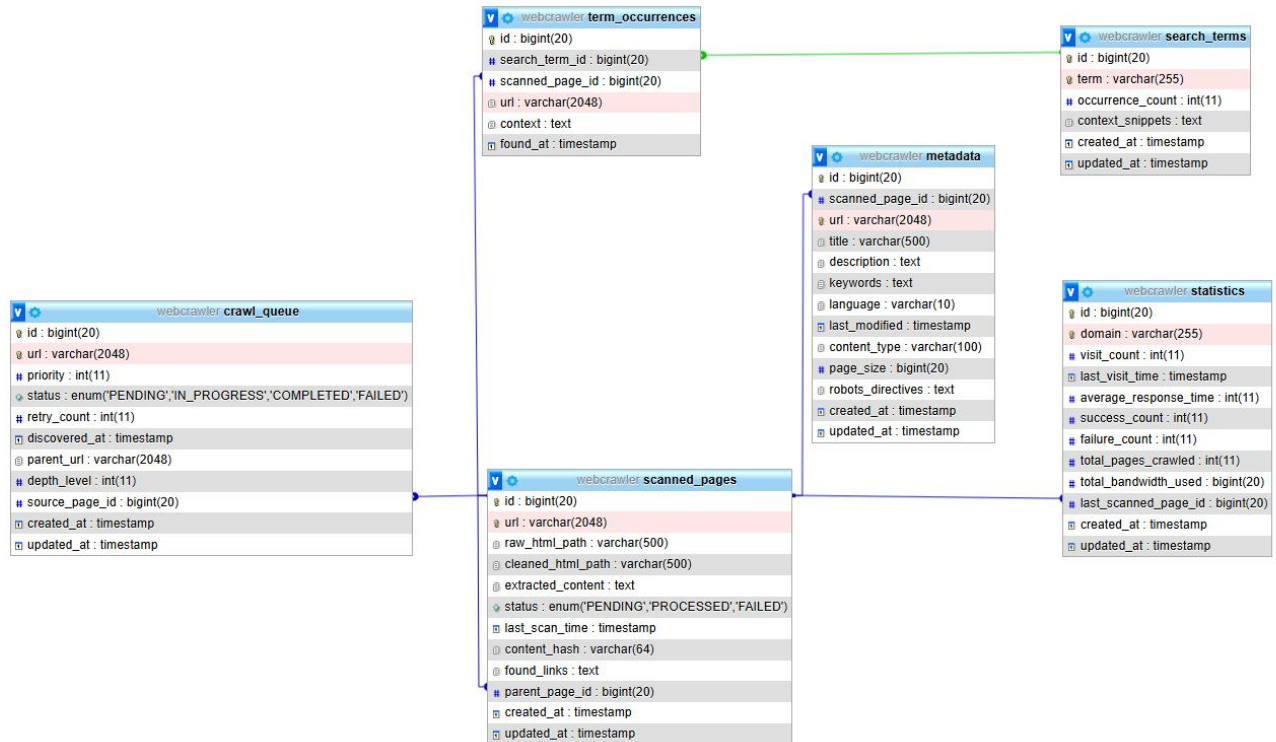


Рис.3 Структура бази даних

Висновок : У ході виконання лабораторної роботи було проведено ознайомлення з теоретичними відомостями та розроблено прецеденти та діаграми класів для системи керування завданнями. Розроблено структуру бази даних. Окрім того, підготовлений звіт включає всі необхідні компоненти, що відображають структуру розробленої системи.

