



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

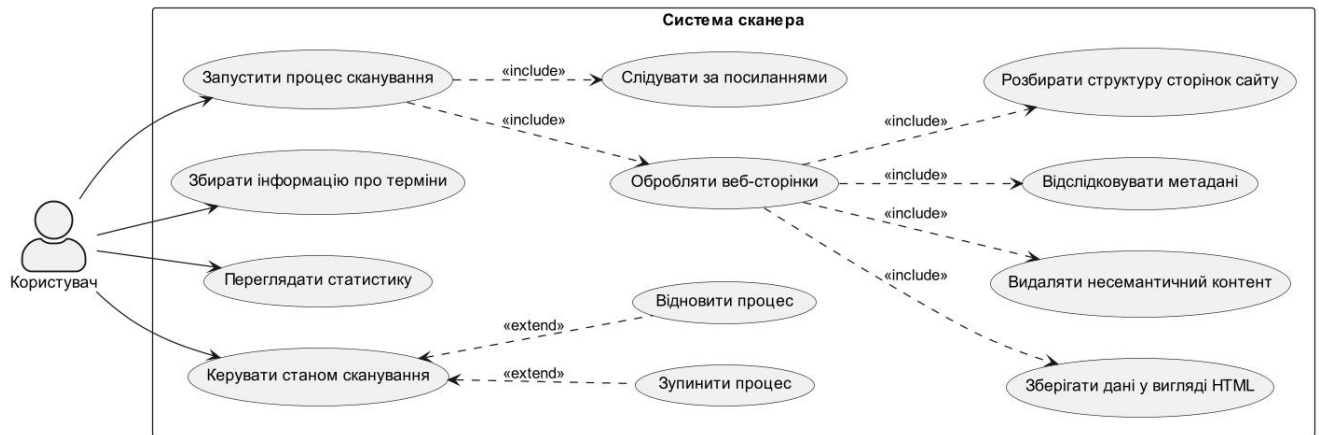


Рис.1. Діаграма прецедентів

Ця діаграма описує функціональні можливості системи сканування веб-сторінок (web crawler або "сканер"). Діаграма визначає, як користувач взаємодіє зі системою, а також різні функціональні блоки, які реалізує система. Давайте розберемо ключові компоненти діаграми.

1. Запускає процес сканування через функцію **"Запустити сканування"**, яка включає :
 - 1) Розбирати структуру сторінок сайту.
 - 2) Відслідковувати метадані.
 - 3) Видаляти несемантичний контент.
 - 4) Зберігати дані у вигляді HTML
2. Користувач може зупинити або відновлювати стан сканера.
3. Після завершення роботи користувач може переглядати статистику.
4. Користувач може переглянути інформацію про зпзначений термін.

Прецеденти використання веб-сканера

Прецедент 1: Запуск сканування сайту

Назва:

Запуск сканування вказаного сайту.

Опис:

Користувач вводить URL-адресу сайту, після чого веб-сканер починає аналізувати сторінки сайту та збирати необхідну інформацію.

Актор:

Користувач.

Передумови:

1. Користувач має доступ до веб-застосунку.
2. Веб-застосунок запущено.
3. База даних готова до збереження даних.

Основний потік:

1. Користувач переходить на головну сторінку веб-застосунку.
2. Користувач вводить URL-адресу сайту у поле для введення.
3. Користувач натискає кнопку "Почати сканування".
4. Система створює нову сесію сканування.
5. Система починає сканування з вказаного URL:
 - 5.1. Завантажує сторінку за URL-адресою.
 - 5.2. Аналізує контент сторінки.
 - 5.3. Зберігає знайдені дані (терміни, текст, метадані) у базі даних.
 - 5.4. Переходить за посиланнями на сторінці та повторює кроки 5.1–5.3.
6. Користувач отримує повідомлення про початок сканування.

Альтернативні потоки:

A1. Некоректна URL-адреса:

1. Система перевіряє правильність URL-адреси.
2. Якщо адреса некоректна, система показує повідомлення про помилку.
3. Користувач вводить правильну URL-адресу.

A2. Сканування зупинено користувачем:

4. Користувач натискає кнопку "Зупинити сканування".

5. Система зупиняє поточну сесію сканування.
6. Система оновлює статус сесії до "Зупинено".

Постумови:

1. Дані зі сканування (сторінки, терміни, метадані) збережено у базі даних.
2. Статус сесії оновлено до "Завершено" або "Зупинено".

Прецедент 2: Пошук терміна на сканованих сторінках

Назва:

Пошук терміна серед зібраних даних.

Опис:

Користувач вводить термін у поле пошуку, а система повертає результати зі сторінок, на яких цей термін був знайдений.

Актор:

Користувач.

Передумови:

1. Веб-застосунок запущено.
2. У базі даних вже є збережені сторінки зі знайденими термінами.

Основний потік:

1. Користувач переходить на сторінку пошуку.
2. Користувач вводить термін у поле пошуку.
3. Користувач натискає кнопку "Знайти".
4. Система виконує пошук терміна у базі даних:
 - 4.1. Шукає термін у таблиці термінів.
 - 4.2. Повертає список сторінок, де цей термін був знайдений.
5. Система показує результати пошуку користувачу, включаючи URL сторінок і короткий опис.

Альтернативні потоки:

- **A1. Термін не знайдено:**

1. Якщо термін відсутній у базі даних, система показує повідомлення: "Термін не знайдено".

Постумови:

1. Користувач отримує список сторінок із вказаним терміном (або повідомлення про відсутність результатів).

Прецедент 3: Перегляд статистики сканування

Назва:

Перегляд статистики про виконані сканування.

Опис:

Користувач переглядає загальну статистику по виконаних сесіях сканування, включаючи кількість відвіданих сторінок, семантичних сторінок та загальну кількість сканувань.

Актор:

Користувач.

Передумови:

1. Веб-застосунок запущено.
2. Сканування вже було виконано, і в базі даних є статистика.

Основний потік:

1. Користувач переходить на сторінку статистики.
2. Система запитує дані статистики з бази даних.
3. Система показує користувачу таку інформацію:
 1. Загальна кількість сканувань.
 2. Загальна кількість відвіданих сторінок.
 3. Кількість семантичних сторінок.
 4. Час останнього сканування.

Альтернативні потоки:

- **A1. Статистика відсутня:**

1. Якщо в базі даних немає жодного запису статистики, система показує повідомлення: "Статистика ще не сформована".

Постумови:

1. Користувач отримує актуальну статистику по сесіях сканування.

2. Діаграма класів системи

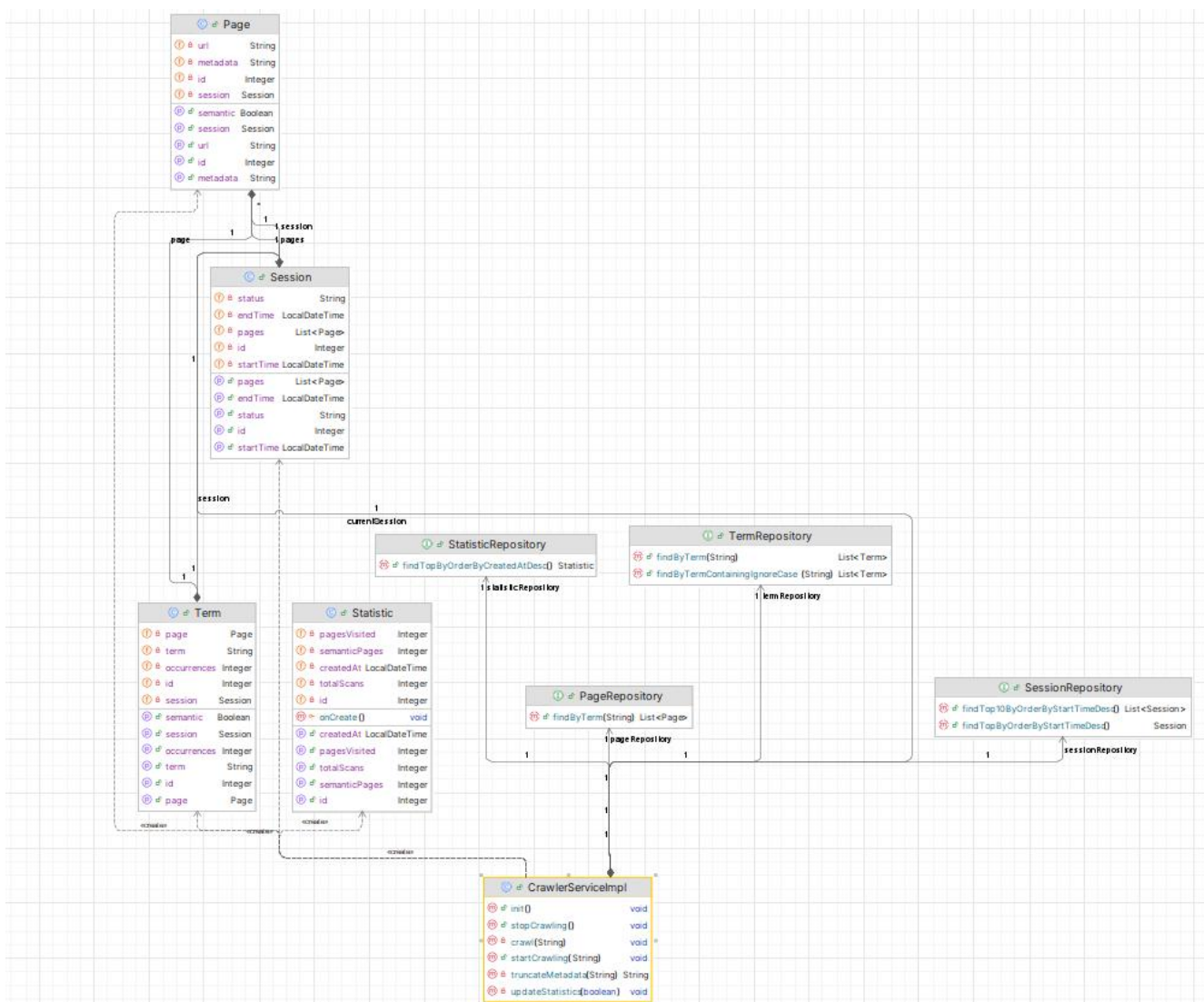


Рис.2. Діаграма класів

Ця діаграма класів відображає архітектуру системи веб-сканера, її ключові компоненти, взаємозв'язки між ними та механізми обробки даних.

3. Структура бази даних

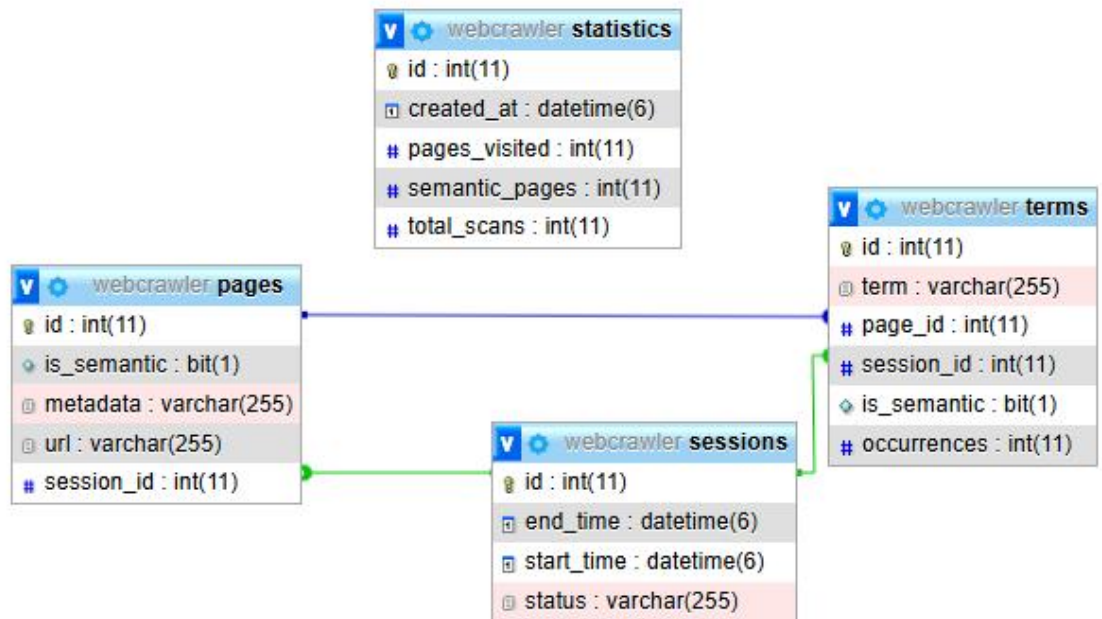


Рис.3 Структура бази даних

sessions: Відстежує кожний сеанс сканування. Це дає можливість запускати нові сеанси, зупиняти їх або відновлювати.

pages: Містить оброблену інформацію про сторінки, щоб можна було визначити, які сторінки були успішно оброблені.

terms: Зберігає терміни, які були знайдені під час сканування.

statistics: Центральне місце, щоб бачити загальну статистику сканувань, кількість успішно відвіданих сторінок та інші ключові дані.

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.

У процесі розробки системи управління проєктами використано різноманітні засоби моделювання для забезпечення чіткого розуміння структури та функціональності системи:

Діаграми варіантів використання допомагають ідентифікувати ключові ролі користувачів (наприклад, менеджери, розробники) і їх взаємодію із системою.

Вони визначають основні функції, такі як створення проєктів, планування задач, управління командами та робота з файлами.

Сценарії варіантів використання деталізують поведінку системи в рамках кожного випадку використання. Це дозволяє краще зрозуміти послідовність дій користувача та відповідь системи, що сприяє виявленню можливих проблем на етапі проектування.

Діаграми UML надають графічне представлення структури та динаміки системи, зокрема її класів, об'єктів та їх взаємодії. Вони є важливими інструментами для комунікації між членами команди.

Діаграми класів формують основу концептуальної моделі системи, описуючи її основні елементи (класи, атрибути, методи) та зв'язки між ними. Вони забезпечують цілісне уявлення про структуру даних і функціональні можливості системи.

