



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 №3
Технології розроблення програмного забезпечення
«ДІАГРАМА РОЗГОРТАННЯ. ДІАГРАМА КОМПОНЕНТІВ. ДІАГРАМА ВЗАЄМОДІЙ ТА
ПОСЛІДОВНОСТЕЙ »

Виконала

студентка групи ІА–22:

Фоменко Альона

Перевірив:

Мягкий Михайло Юрійович

Київ 2024

Тема: ДІАГРАМА РОЗГОРТАННЯ. ДІАГРАМА КОМПОНЕНТІВ. ДІАГРАМА ВЗАЄМОДІЙ ТА ПОСЛІДОВНОСТЕЙ

Мета: Розробити діаграму розгортання для проектованої системи. Розробити діаграму компонентів для проектованої системи. Розробити діаграму послідовностей для проектованої системи.

Теоретичні відомості

Діаграма розгортання (Deployment Diagram)

Діаграми розгортання представляють фізичне розташування системи, показуючи, на якому фізичному устаткуванні запускається та чи інша складова програмне забезпечення.

Головними елементами діаграми є вузли, пов'язані інформаційними шляхами. Вузол (node) – це те, що може містити програмне забезпечення. Вузли бувають двох типів. Пристрій (device) – це фізичне обладнання: комп'ютер або пристрій, пов'язаний із системою. Середовище виконання (execution environment) – це програмне забезпечення, яке може включати інше програмне забезпечення, наприклад, операційну систему або процес-контейнер (наприклад, веб-сервер).

Між вузлами можуть стояти зв'язки, які зазвичай зображуються у вигляді прямої лінії.

Як і на інших діаграмах, зв'язки можуть мати атрибути множинності (для показання, наприклад, підключення 2х і більше клієнтів до одного сервера) та назва.

У назві зазвичай міститься спосіб зв'язку між двома вузлами — це може бути назва протоколу (http, IPC) чи використовувана технологія забезпечення взаємодії вузлів (. Remoting, WCF).

Вузли можуть містити артефакти (artifacts), які є фізичними. уособленням програмного забезпечення; зазвичай це файли. Такими файлами можуть бути виконувані файли (такі як .exe файли, двійкові файли, файли DLL, файли JAR, складання або сценарії) або файли даних, конфігураційні файли, HTML документи і т. д. Перелік артефактів усередині вузла вказує на те, що на цьому вузлі артефакт розгортається в систему, що запускається.

Артефакти можна зображати у вигляді прямокутників класів або перераховувати їхні імена усередині вузла. Якщо ви показуєте ці елементи у вигляді прямокутників класів, можна додати значок документа або ключове слово «artifact». Можна супроводжувати вузли або артефакти значеннями у вигляді міток, щоб вказати різну цікаву інформацію про сайт, наприклад постачальника, операційну систему, місце - загалом, все, що прийде вам на думку.

Часто у вас буде безліч фізичних вузлів для вирішення однієї і тієї ж логічного завдання. Можна відобразити цей факт, намалювавши безліч прямокутників вузлів або поставивши число у вигляді значення-мітки.

Артефакти часто є реалізацією компонентів. Це можна показати, задавши значення-мітки всередині прямокутників артефактів.

Основні види артефактів:

вихідні файли;

- виконувані файли;
- сценарії;
- таблиці баз даних;
- документи;
- результати розробки, UML-моделі.

Можна також деталізувати артефакти, що входять у вузол; наприклад, додатково всередині файлу, що розгортається вказати, які туди входять компоненти чи класи. Така деталізація, як правило, не має сенсу на діаграмах розгортання, оскільки може зміщувати фокус уваги від моделі розгортання програмного забезпечення для його внутрішнього пристрою, однак іноді може бути корисною. При цьому, можливо встановлювати зв'язки між компонентами/класами межах різних вузлів.

Діаграми розгортань розрізняють двох видів: описові та екземплярні. На діаграмах описової форми вказуються вузли, артефакти та зв'язку між вузлами без зазначення конкретного обладнання чи програмного забезпечення, необхідного для розгортання. Такий вид діаграм корисний на ранніх етапах розробки для розуміння, які впринципі фізичні пристрої необхідні для функціонування системи або для опису процесу розгортання у загальному ключі.

Діаграми екземплярної форми несуть у собі екземпляри обладнання, артефактів та зв'язків між ними. Під екземплярами розуміються конкретні елементи - ПК з відповідним набором характеристик та встановленим ПЗ; цілком можливо, в межах однієї організації це може бути якийсь конкретний вузол (наприклад, ПК тестувальника Василя). Діаграми екземплярної форми розробляються на завершальних стадіях розробки ПЗ - коли вже відомі і сформульовано вимоги до програмного комплексу, обладнання закуплено та все готове до розгортання. Діаграми такої форми є скоріше планом розгортання у графічному вигляді, ніж модель розгортання.

Хід роботи

Тема 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.

Діаграма розгортання

Користувач:

- Представлений піктограмою людини.
- Користувач взаємодіє з додатком через веб-браузер за допомогою клавіатури та монітора.

2. Сервер додатків (Application Server):

- Включає дві основні частини:
 - Веб-браузер — використовується для взаємодії з користувачем. Відображає інтерфейс додатку і взаємодіє з локальним сервером.
 - CrawlerApiApplication — основний серверний компонент, що містить кілька рівнів:
 - Presentation layer (Шар представлення) — відповідає за статичні та шаблонні ресурси, які використовуються для відображення інтерфейсу користувача.
 - Контролери, сервіси та репозиторії — основна логіка додатку, яка забезпечує функціональність, обробку запитів і доступ до даних.
 - Table to entities і Query selections — забезпечують перетворення даних з бази даних у відповідні сутності для роботи у додатку.

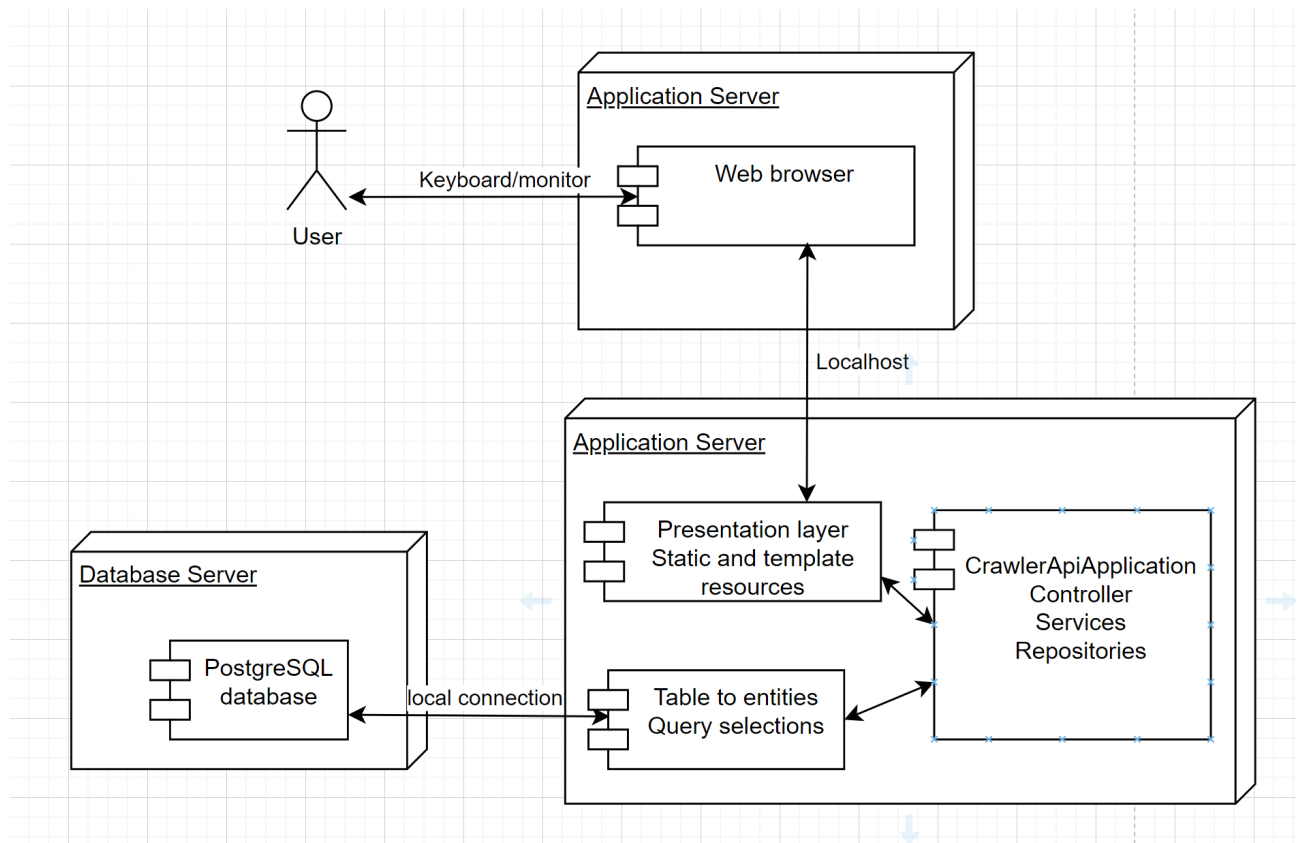
3. Сервер бази даних (Database Server):

- Містить базу даних PostgreSQL, яка зберігає основні дані додатку.

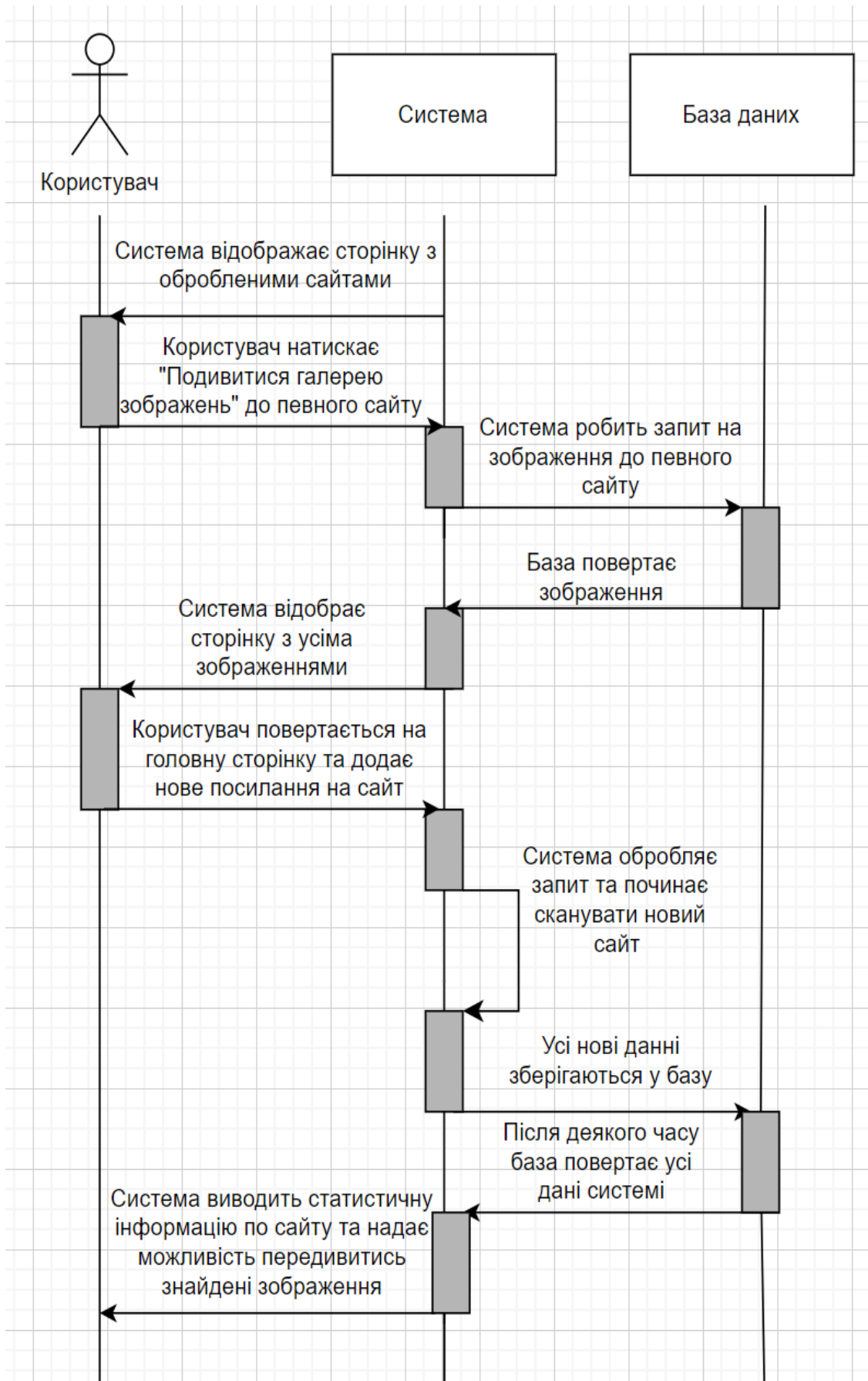
- Local connection — зв'язок між сервером додатків і сервером бази даних. Сервер додатків надсилає запити до бази даних і отримує дані через цей канал.

Взаємодія між компонентами:

- Користувач за допомогою браузера на сервері додатків надсилає запити та отримує відповіді від додатку.
- Сервер додатків обробляє запити, звертаючись до бази даних для отримання або збереження інформації, після чого повертає результати користувачу.



Діаграма послідовностей



Відображення сторінки з обробленими сайтами:

- Система показує користувачу сторінку, де перераховані сайти, з яких вже зібрано зображення.

Запит на перегляд галереї зображень:

- Користувач натискає на кнопку "Подивитися галерею зображень" для вибраного сайту.
- Система робить запит до бази даних, щоб отримати зображення з обраного сайту.

Отримання зображень із бази даних:

- База даних повертає зображення, і система відображає їх користувачу у вигляді галереї.

Додавання нового посилання на сайт:

- Користувач повертається на головну сторінку та додає нове посилання на сайт, який потрібно сканувати для зображень.

Сканування нового сайту:

- Система приймає запит, починає обробку та сканує новий сайт на наявність зображень.
- Усі знайдені зображення зберігаються в базі даних.

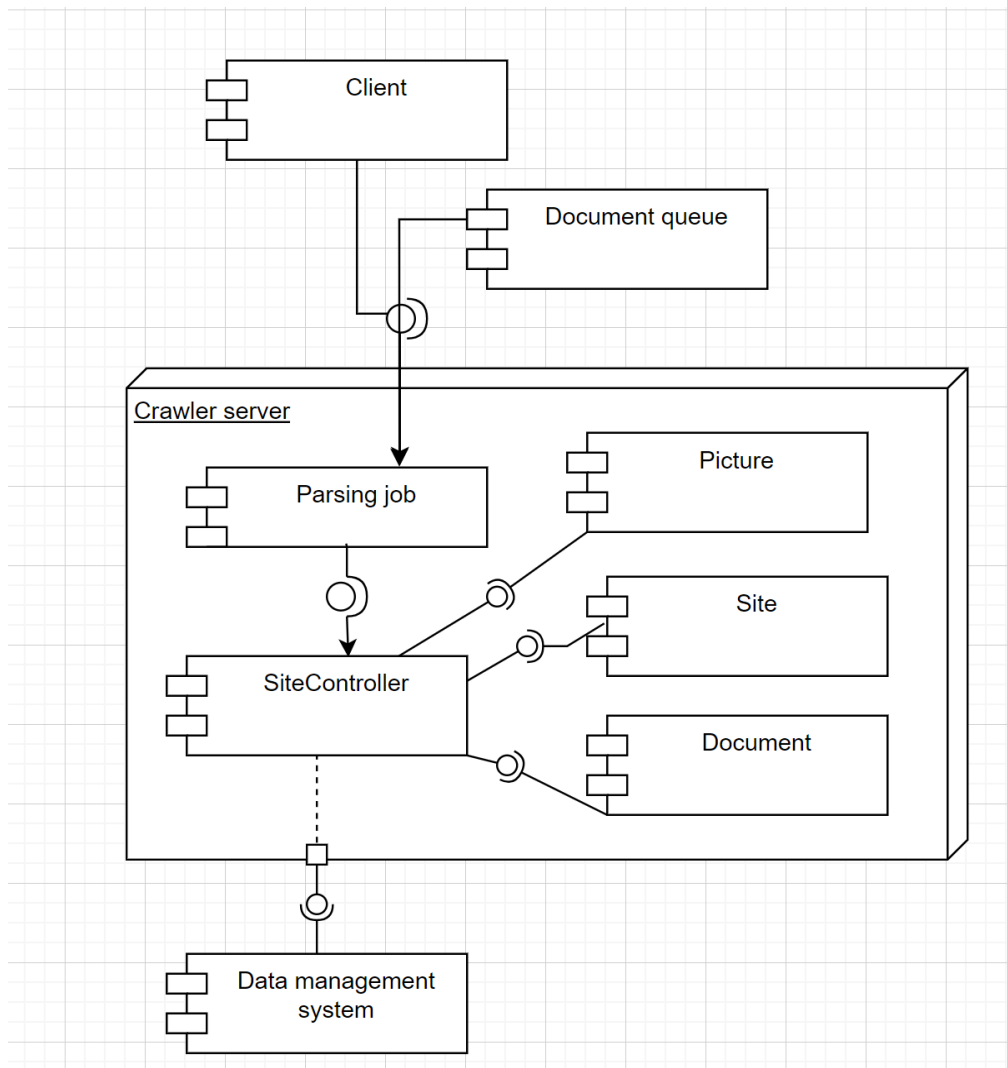
Оновлення даних у базі:

- Після завершення процесу сканування база даних повертає оновлені дані системі.

Виведення статистики та зображень:

- Система виводить статистичну інформацію по обробленому сайту та надає користувачу можливість переглянути знайдені зображення.

Діаграма компонентів



Ця діаграма зображує компонентну архітектуру системи збирання та обробки документів.

1. **Client (Клієнт)** – взаємодіє із системою, надсилаючи запити на обробку документів.
2. **Document queue (Черга документів)** – черга, в якій зберігаються документи, що очікують на обробку. Клієнт відправляє документи в цю чергу для подальшої обробки.
3. **Crawler server (Сервер краулера)** – основний компонент, який виконує завдання обробки документів:
 - **Parsing job (Завдання парсингу)** – відповідає за аналіз документів, передаючи їх далі до інших компонентів.

- Picture (Зображення) Site (сайт) та Document (Документ) – компоненти, що обробляють та зберігають інформацію відповідно до типу даних (зображення або текст).

- SiteController – основний модуль, який інтегрує дані та координує обробку документів між різними компонентами.

4. Data management system (Система управління даними) – зберігає та керує обробленими даними, отриманими з краулера.

Висновок: завдяки цій лабораторній роботі ми змогли створити діаграму послідовностей, діаграму компонентів, а також діаграму розгортання.