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
З дисципліни «Технології розроблення програмного забезпечення»
Тема: **«Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей.»**
HTTP-сервер

Виконав:
Студент групи ІА-22
Білецький С. В.

Перевірив:
Мягкий М. Ю.

Київ-2024

Зміст

Тема:
Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей.

Мета:
Навчитися будувати діаграми розгортання, компонентів, взаємодій та послідовностей для моделювання архітектури та логіки роботи програмних систем, а також поглибити розуміння структурної та поведінкової взаємодії між компонентами системи.

Завдання:
HTTP-сервер. Сервер повинен мати можливість розпізнавати вхідні запити і формувати коректні відповіді (згідно протоколу HTTP), надавати сторінки chtml (html сторінки з додаванням найпростіших C# конструкцій на розсуд студента), вести статистику вхідних запитів, обробку запитів у багатопотоковому/подієвому режимах.

Хід роботи

1. Діаграма розгортання

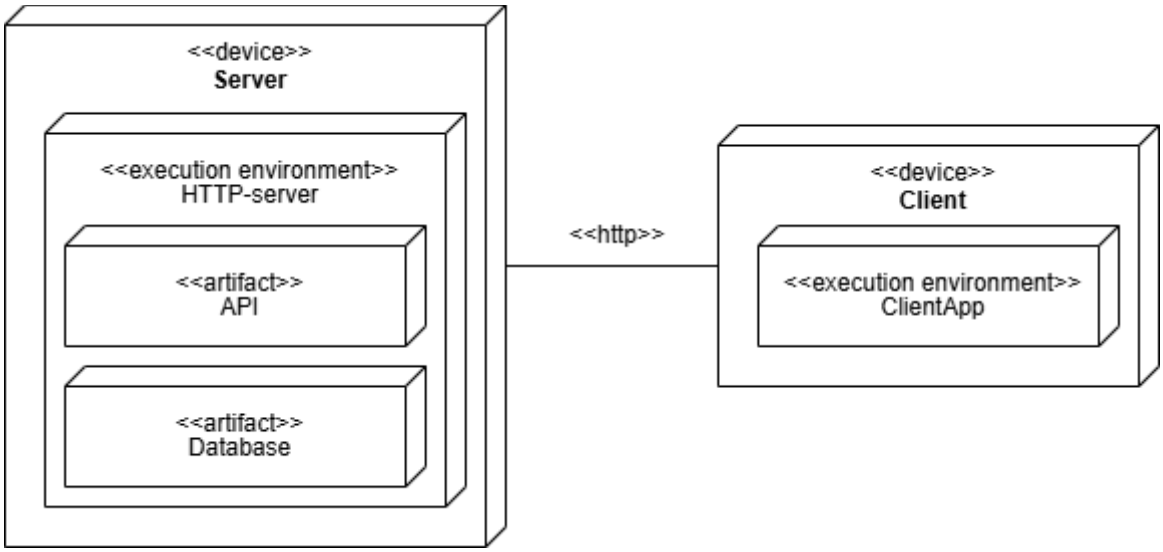

Рис. 1 — Діаграма розгортання

Ця діаграма розгортання ілюструє архітектуру клієнт-серверної системи з HTTP-сервером, який надає API та базу даних. На діаграмі зображено два пристрої: сервер та клієнт.

1. Сервер:

- HTTP-сервер працює як виконуюче середовище, що забезпечує обробку HTTP-запитів і формування відповідей відповідно до протоколу HTTP.
- API є артефактом, що надає функціональність для взаємодії з клієнтом, використовуючи абстрактний інкапсульований API. Сервер може надсилати сторінки CHTML (HTML сторінки з елементами C#) на запит клієнта.
- База даних використовується для збереження статистики про вхідні запити та інших даних, необхідних для обслуговування клієнтів.
- Сервер підтримує багатопоточну або подієву обробку запитів, що дозволяє обробляти кілька запитів одночасно, підвищуючи ефективність системи.

2. Клієнт:
   - Клієнт пристрій містить виконуюче середовище ClientApp, яке взаємодіє із сервером через API, надсилаючи запити і отримуючи відповіді.

Ця система побудована для обробки HTTP-запитів від клієнтів, забезпечуючи багатопоточну обробку та можливість надання динамічного контенту через CHTML.

2. Діаграма компонентів

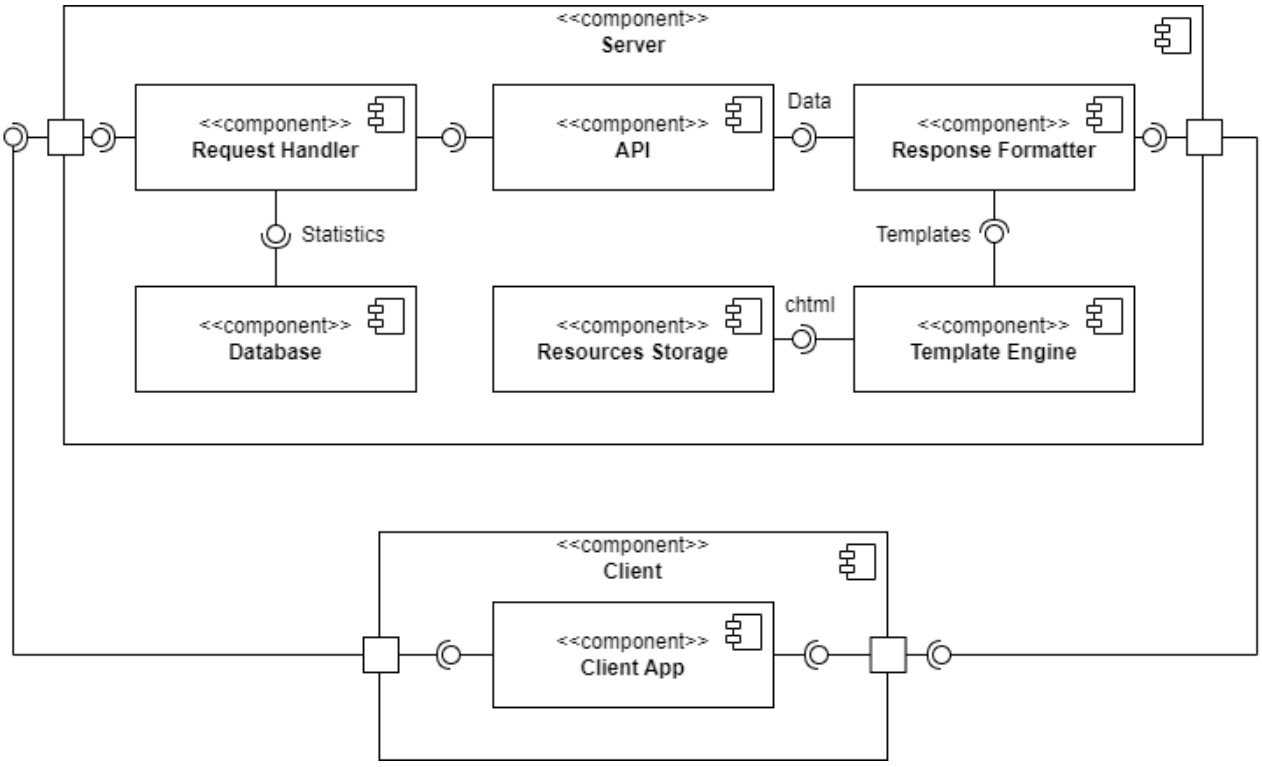

Рис. 2 — Діаграма компонентів

Ця діаграма представляє структуру HTTP-сервера з можливістю обробки запитів та формування відповідей відповідно до протоколу HTTP. Сервер складається з кількох компонентів, що працюють разом для забезпечення функціональності.

1. Request Handler – обробляє вхідні HTTP-запити, спрямовуючи їх до відповідного компонента API для подальшої обробки. Він також відповідає за збір статистики запитів і підтримку багатопотокового або подієвого режимів для оптимізації роботи сервера.
2. API – це основний компонент, який взаємодіє з іншими частинами системи для обробки даних і створення відповідей. Він приймає дані від Request Handler і передає їх далі.
3. Response Formatter – формує коректні HTTP-відповіді на основі даних, отриманих від API. Цей компонент відповідає за структурування даних відповідно до протоколу HTTP і надсилання готових відповідей клієнту.
4. Database – використовується для зберігання статистики про вхідні запити, що дає змогу аналізувати роботу сервера та оптимізувати його продуктивність.
5. Resources Storage – зберігає статичні ресурси та файли chtml, які використовуються для відображення сторінок з елементами C#. Запити до ресурсів обробляються цим компонентом, щоб забезпечити клієнтам доступ до потрібних даних.
6. Template Engine – генератор шаблонів, який створює HTML-сторінки (chtml) з використанням базових C# конструкцій. Він динамічно формує шаблони для відповіді на запити, що надходять від клієнта, використовуючи дані API.
7. Client App – клієнтська частина, яка ініціює запити до сервера, взаємодіючи з API і отримуючи HTML-відповіді, підготовлені сервером.

Ця система забезпечує обробку HTTP-запитів, генерацію HTML-відповідей та ведення статистики запитів.

3. Діаграма послідовностей

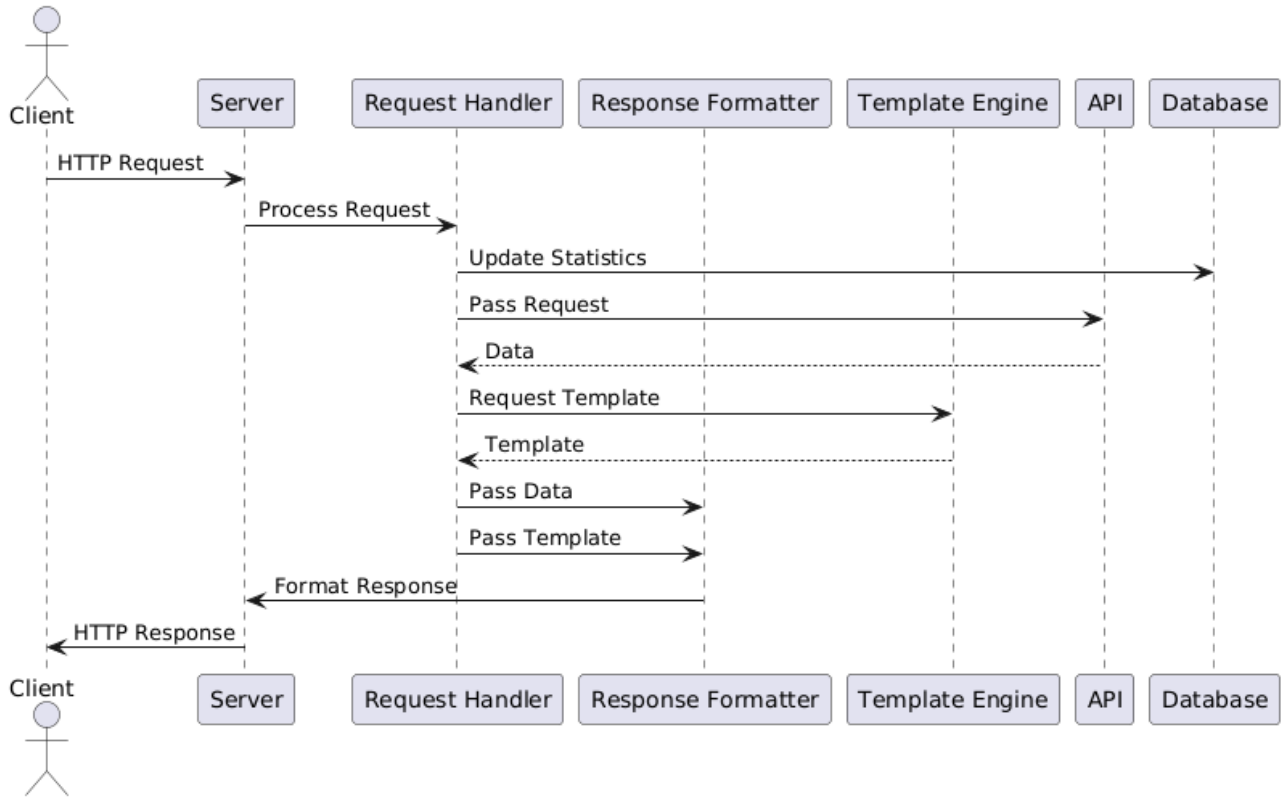

Рис. 3 — Діаграма послідовностей

Ця діаграма послідовностей описує роботу HTTP-сервера, який обробляє вхідні HTTP-запити, формує відповіді та взаємодіє з API і базою даних. Процес обробки запиту відбувається наступним чином:

1. Клієнт надсилає HTTP-запит на сервер.
2. Сервер отримує запит та передає його в Request Handler для подальшої обробки.
3. Request Handler обробляє запит і оновлює статистику запитів (ведеться статистика про вхідні запити).
4. Після цього Request Handler передає запит до API для отримання необхідних даних.
5. API запитує інформацію з бази даних і повертає потрібні дані до Request Handler.
6. Request Handler запитує необхідний шаблон для відповідної chtml сторінки від Template Engine.
7. Template Engine формує шаблон з chtml сторінки і повертає його до Request Handler.
8. Request Handler передає отримані дані та шаблон до Response Formatter, який формує остаточну відповідь.

9. Response Formatter передає сформовану відповідь серверу, а той — клієнту у вигляді HTTP-відповіді.

Сервер побудований для роботи в багатопотоковому або подієвому режимі, що забезпечує одночасну обробку декількох запитів.

Висновки:

Навчились будувати діаграми розгортання, компонентів, взаємодій та послідовностей для моделювання архітектури та логіки роботи програмних систем, а також поглибили розуміння структурної та поведінкової взаємодії між компонентами системи.