Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №3
Технології розроблення програмного забезпечення
«*Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей.*»
Варіант 2

Виконала
студентка групи ІА–24:
Кійко А. О.

Перевірив
Мягкий М. Ю.

Київ 2024

ЗМІСТ

Тема: HTTP-сервер (state, builder, factory method, mediator, composite, p2p). 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.

Короткі теоритичні відомості

Діаграма розгортання показує фізичне розташування програмних компонентів на апаратному забезпеченні системи. Вона відображає вузли (пристрої або сервери), де запускаються програмні модулі, та зв'язки між ними. Основною метою діаграми розгортання є візуалізація того, як система буде працювати в реальному середовищі. Вона демонструє, на яких пристроях або серверах розміщуються різні частини системи (клієнтська частина, сервер додатків, база даних) та як вони взаємодіють між собою. Це корисно для розуміння фізичної архітектури системи, її масштабованості та ефективності.

Діаграма компонентів використовується для відображення логічної структури програмного забезпечення та показує, як різні модулі або компоненти системи взаємодіють між собою. Компоненти — це окремі частини програмного забезпечення, які виконують конкретні функції та можуть взаємодіяти через інтерфейси. Ця діаграма допомагає зрозуміти структуру системи, які модулі вона містить, та як ці модулі обмінюються інформацією. Використовується для документування системи на етапі проектування та дає уявлення про логічну організацію коду.

Діаграма послідовностей показує, як об'єкти системи взаємодіють між собою в хронологічному порядку для виконання певної операції. Вона фокусується на тому, які повідомлення передаються між компонентами та в якому порядку. Це корисний інструмент для моделювання динамічних аспектів системи, таких як процес реєстрації користувача або збереження даних. Діаграма допомагає

зрозуміти, як відбувається взаємодія між клієнтом, сервером та базою даних у реальному часі, забезпечуючи чітке уявлення про порядок виконання операцій.

Хід роботи

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Розробити діаграму розгортання для проектованої системи.
3. Розробити діаграму компонентів для проектованої системи.
4. Розробити діаграму послідовностей для проектованої системи.
5. Скласти звіт про виконану роботу

Діаграма розгортання:

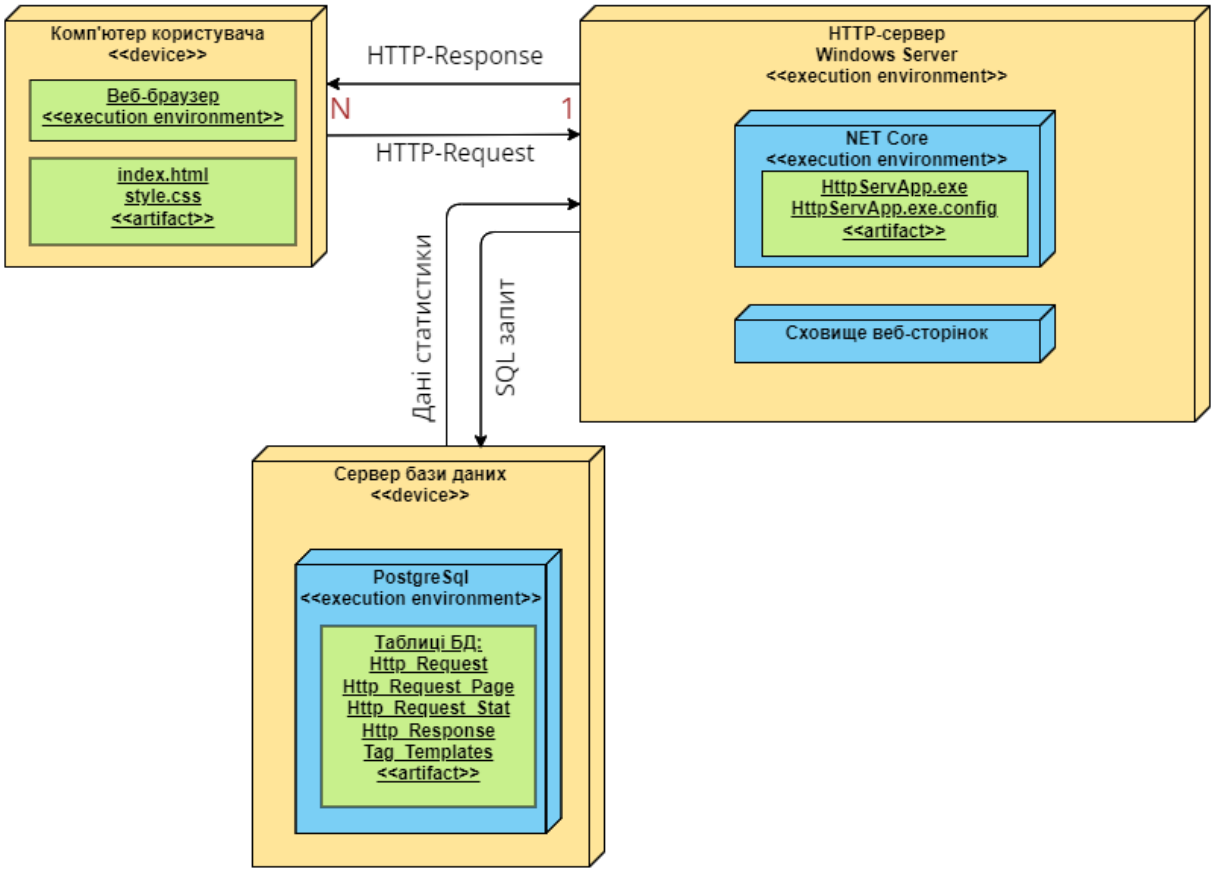

Рис. 1 - Діаграма розгортання

Діаграма розгортання містить 3 основні складові: пристрій користувача, сервер застосунку (Http-сервер) та сервер бази даних. Розглянемо детально кожну з них.

1. **Користувацький пристрій** (Комп'ютер користувача) містить наступні складові:
   - середовище виконання <<execution environment>> – *клієнтська операційна система* зі встановленим *Web-браузером*;
   - <<artifact>> *index.html* (зі вставками JavaScript), *style.css*: містять структуру та стиль основної сторінки Http-серверу, до функції якої, зокрема, належить можливість введення параметрів для отримання адміністратором статистики запитів.

Взаємодія: клієнтська частина надсилає Http-запит до сервера на отримання Web-сторінки (на запит користувача) або статистичної інформації (на запит адміністратора) та відображає результати виконання.

2. **Серверний пристрій (Http-сервер)** містить наступні складові:
   - середовище виконання <<execution environment>> – *серверна операційна система* зі встановленим кросплатформеним середовищем *.NET Core* для запуску додатку Http-сервера;
   - <<artifact>>:
     - *HttpServApp.exe* – серверний додаток, що виконує обробку Http-запитів клієнтів та формування відповідей. Мова реалізації – C#;
     - *HttpServApp.exe.config* – файл конфігурації додатку;
     - *Сховище Web-сторінок* - файлове сховище сторінок, що можуть бути надані у відповідь на запит клієнта.

   Взаємодія: сервер приймає Http-запит від клієнта, перевіряє валідність, виконує обробку, передає інформацію про запит до бази даних на збереження та надає клієнту результат виконання.
3. **Сервер бази даних** (БД) містить наступні складові:
   - середовище виконання <<execution environment>> – *операційна система* зі встановленою СУБД *PostgreSQL;*
   - <<artifact>> – *таблиці БД*, що забезпечують збереження інформації про оброблені запити та результати їх виконання.

   Взаємодія: сервер БД отримує SQL-запити від Http-серверу для додавання та збереження інформації у таблицях про оброблений запит користувача, а також надає статистичну інформацію про запити користувачів.

Діаграма компонентів:

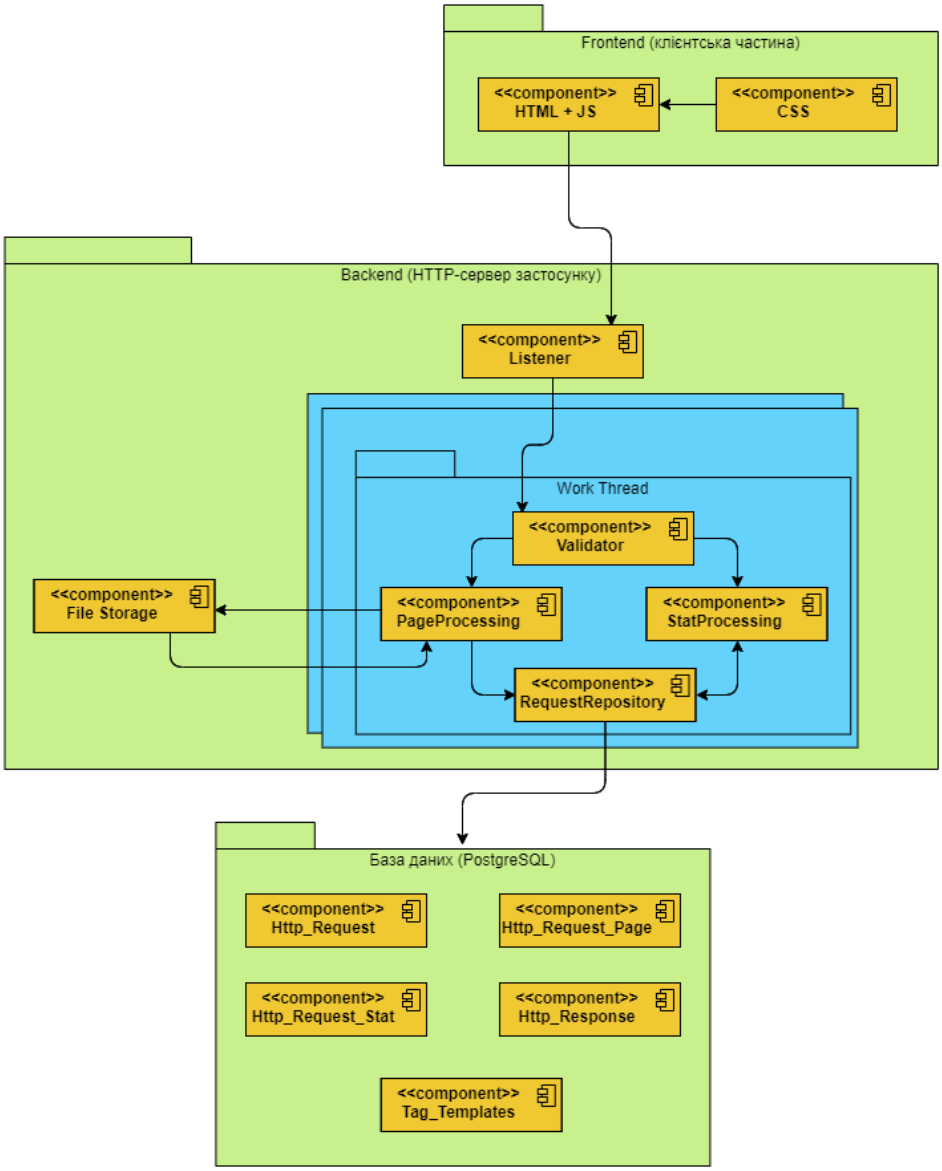

Рис. 2 - Діаграма компонентів

1. **Frontend** (клієнтська частина):
   - *HTML + JS* – компонент, що відповідає за структуру та функціонал стартової сторінки, що відображається користувачу.
   - *CSS* – компонент, що забезпечує стилістичне оформлення графічного інтерфейсу.
2. **Backend** (HTTP-сервер застосунку):

- *Listener* – компонент, що здійснює керування процесом обробки: забезпечує прослуховування відповідного сокету, створення та запуск потоку обробки запиту.
- *Validator*: відповідає за парсинг вхідного потоку інформації, валідацію та визначення типу запиту: запит Web-сторінки чи запит статистики.
- *PageProcessing*: відповідає за процес обробки запиту користувача на отримання Web-сторінки, додатково здійснює конвертацію C# конструкцій у фрагменти HTML-кода. Взаємодіє з компонентом файлового сховища Web-сторінок щодо отримання вмісту сторінки та компонентом репозиторію щодо передачі інформації про результати обробки запиту.
- *StatProcessing*: відповідає за процес обробки запиту статистичних даних. Взаємодіє з компонентом репозиторію щодо отримання даних статистики та передачі інформації про результати обробки статистичного запиту.
- *RequestRepository* – інтерфейс, що забезпечує читання, додавання, модифікацію, видалення та збереження даних таблиць бази даних.
- File Storage - компонент файлового сховища Web-сторінок.

3. **База даних** (СУБД PostgreSQL):
   - *Http_Request* – таблиця для зберігання загального переліку запитів до системи;
   - *Http_Request_Page* – таблиця для зберігання додаткових даних про запити Web-сторінок;
   - *Http_Request_Stat* – таблиця для зберігання параметрів запитів статистики;
   - *Http_Response* – таблиця для зберігання даних результатів обробки запитів;
   - *Tag_Templates* – таблиця для зберігання конвертації вихідних C# конструкцій у фрагменти HTML-кода Web-сторінок.

Діаграма послідовностей:

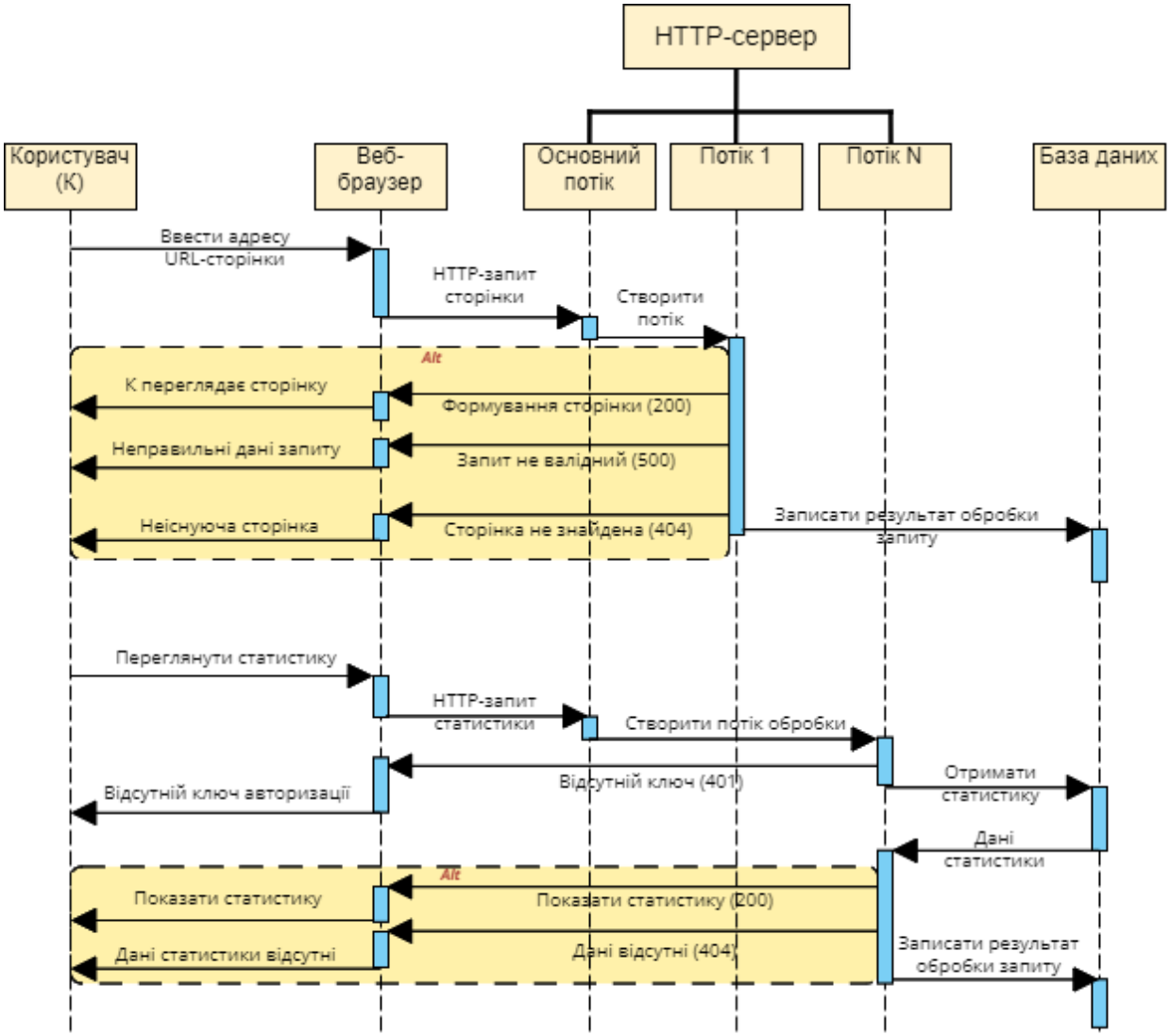

Рис. 3 - Діаграма послідовностей

Опис діаграми послідовностей:

1. **Запит Web-ресурсу (html-сторінки):**
   - після вводу користувачем URL-адреси сторінки Web-браузер надсилає Http-запит до серверу;
   - сервер застосунку створює потік для обробки запиту, в якому:
     - перевіряється валідність запиту: якщо запит не валідний, сервер формує відповідь клієнту про некоректний запит зі статусом 500;
     - перевіряється наявність ресурсу (сторінки), що запитується: якщо ресурс не знайдено, то сервер формує відповідь клієнту зі статусом 404;

- за необхідності виконується обробку серверних C# вставок;
- формується відповідь та надсилається сторінка клієнту зі статусом обробки 200;
- надсилається SQL-запит до бази даних на додавання інформації про користувацький запит та результат його обробки.

- сервер бази даних після отримання SQL-запиту виконує запис інформації у відповідні таблиці бази даних.

2. **Запит користувачем з правами адміністратора даних статистики:**
   - користувач обирає параметри для формування статистики та надсилає Http-запит на сервер з ключом адміністратора;
   - сервер застосунку створює потік для обробки запиту, в якому:
     - перевіряється наявність та валідність ключа доступу адміністратора: якщо ключ відсутній або не валідний, то сервер формує відповідь клієнту про помилку авторизації зі статусом 401;
     - надсилається SQL-запит до бази даних;
     - сервер бази даних виконує запит та повертає результати на сервер застосунку;
     - аналізується відповідь від бази даних: якщо дані, згідно параметрів запиту, не знайдені, то сервер формує відповідь клієнту про відсутність статистичних даних зі статусом 404;
     - формується відповідь та надсилається сторінка відображення статистичних даних зі статусом обробки запиту 200;
     - надсилається SQL-запит до бази даних на додавання інформації про запит статистичних даних та результат його обробки.
   - сервер бази даних після отримання SQL-запиту виконує запис інформації у відповідні таблиці бази даних про запит статистичних даних.

Висновок: у ході виконання даної лабораторної роботи було створено UML-діаграми (діаграми розгортання, компонентів та послідовностей). Діаграма

розгортання допомагає візуалізувати фізичне розташування компонентів, зрозуміти, як клієнт, сервер та база даних співпрацюють під час обробки запитів. Діаграма компонентів допомагає структурувати логічні частини системи, а діаграма послідовностей відображає динамічну взаємодію між ними.

Посилання на репозиторій: AnnaKiikoIA24/TRPZ_labs_Kiiko_AO_IA24 (github.com)