



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

Технології розроблення програмного забезпечення

Лабораторна робота №3

“ДІАГРАМА РОЗГОРТАННЯ. ДІАГРАМА КОМПОНЕНТІВ.
ДІАГРАМА ВЗАЄМОДІЙ ТА ПОСЛІДОВНОСТЕЙ.”

Виконала
студентка групи ІА–24:
Кармазіна А. В.

Перевірив:
Мягкий М. Ю.

Зміст

Завдання	3
Тема: 26 Download manager	3
Теоретичні відомості	3
Діаграма розгортання	4
Діаграма компонентів	5
Діаграма послідовностей	7
Висновок	9

Завдання

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

Тема: 26 Download manager (iterator, command, observer, template method, composite, p2p)

Інструмент для скачування файлів з інтернету по протоколах http або https з можливістю продовження завантаження в зупиненому місці, розподілу швидкостей активним завантаженням, ведення статистики завантажень, інтеграції в основні браузері (firefox, opera, internet explorer, chrome)

Теоретичні відомості

Діаграма розгортання (Deployment Diagram)

Діаграма розгортання використовується для відображення фізичної архітектури системи. Вона показує, як програмні компоненти розміщуються на апаратних пристроях (вузлах) і як між ними встановлюється зв'язок.

Діаграма компонентів (Component Diagram)

Використовується для моделювання програмних компонентів системи та їх взаємодії. Вона показує, з яких модулів складається система і як вони взаємодіють.

Діаграма послідовностей (Sequence Diagram)

Діаграма послідовностей описує порядок виконання операцій або обмін повідомленнями між об'єктами у часі. Вона демонструє, як різні частини системи взаємодіють для виконання певного сценарію.

Хід роботи

Діаграма розгортання

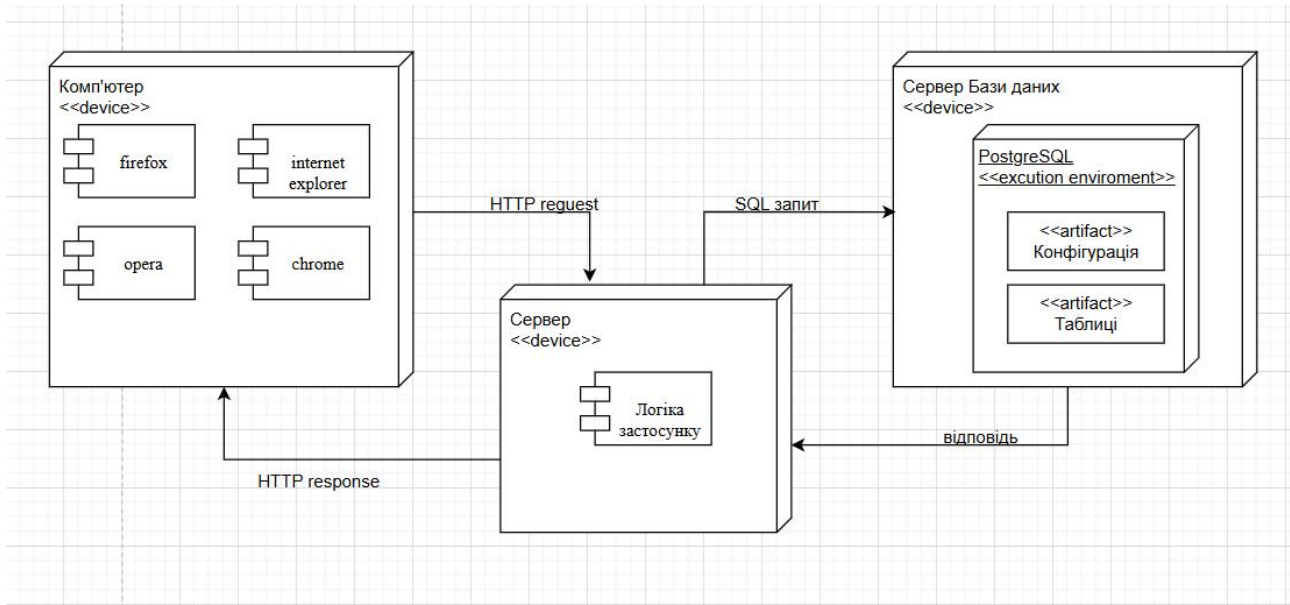


Рис. 1 - Діаграма розгортання

Ця діаграма демонструє архітектуру взаємодії між компонентами системи

Клієнтська частина (Комп'ютер):

Представлена браузерами (Firefox, Internet Explorer, Opera, Chrome), які надсилають HTTP-запити до сервера.

Сервер:

1. Містить логіку застосунку, яка обробляє HTTP-запити від браузерів і формує відповіді.
2. Взаємодіє з базою даних через SQL-запити.

Сервер бази даних:

1. Використовує PostgreSQL як середовище виконання.
2. Містить два артефакти: Конфігурація (параметри роботи БД) та Таблиці (дані).

Процес взаємодії:

- Браузер відправляє HTTP-запит до сервера.
- Сервер, обробивши запит, звертається до бази даних через SQL-запит.
- База даних надсилає відповідь серверу.
- Сервер формує HTTP-відповідь і повертає її браузеру.

Діаграма компонентів

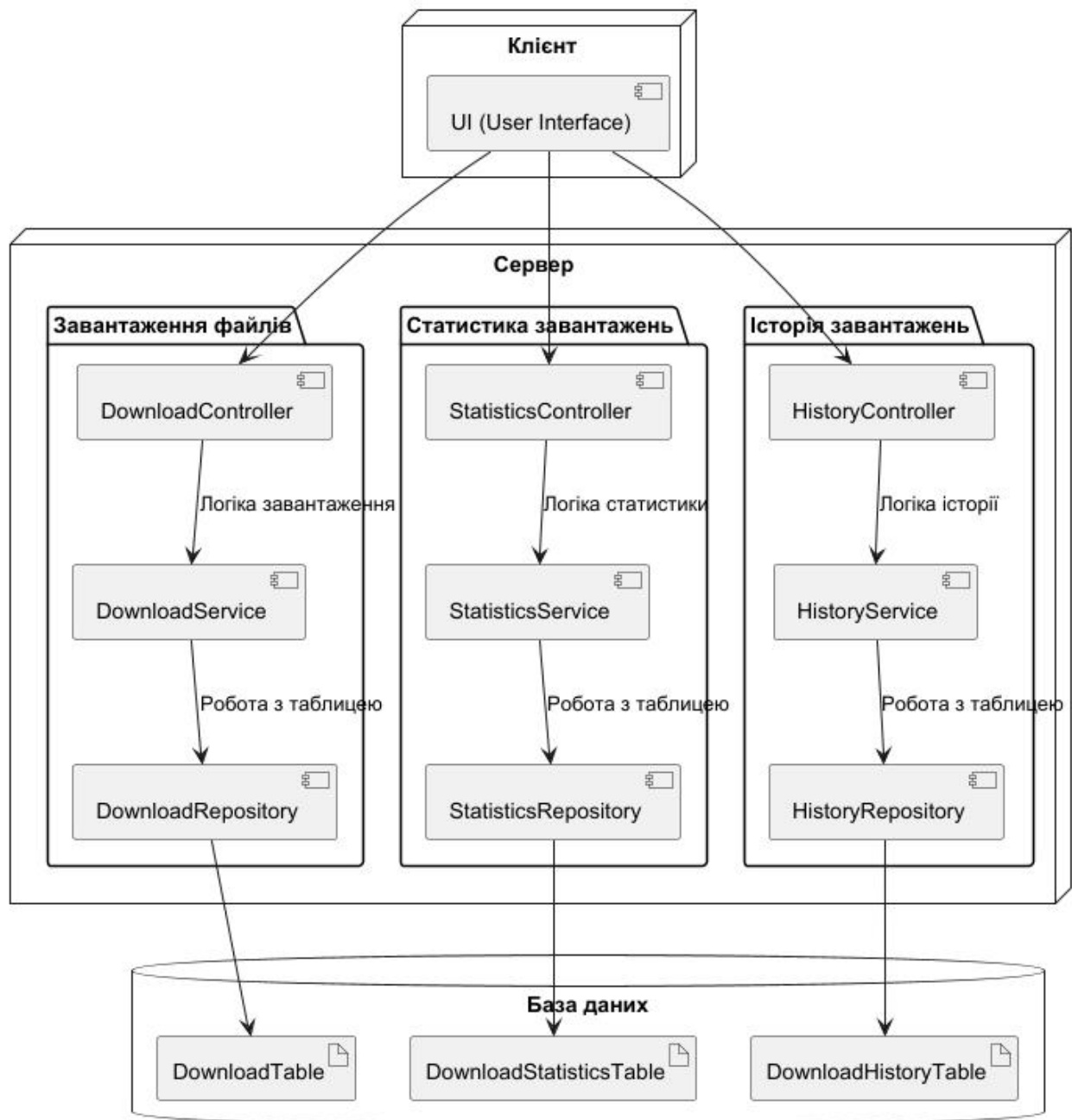


Рис. 2 -Діаграма компонентів

Ця діаграма демонструє архітектуру компонентів системи для управління процесами завантаження файлів, статистики завантажень та історії завантажень.

Клієнт:

Представлений компонентом UI (User Interface), який відповідає за взаємодію з користувачем. Клієнт надсилає запити до серверних компонентів.

Сервер:

Завантаження файлів:

DownloadController — отримує запити від клієнта та керує операціями.

DownloadService — обробляє бізнес-логіку, пов'язану із завантаженням файлів.

DownloadRepository — здійснює доступ до таблиці даних завантажень у базі.

Статистика завантажень:

StatisticsController — керує запитам, пов'язаними зі статистикою.

StatisticsService — обробляє логіку обчислення статистики.

StatisticsRepository — працює з таблицею статистики у базі даних.

Історія завантажень:

HistoryController — керує запитам, пов'язаними з історією завантажень.

HistoryService — обробляє логіку збереження та отримання історії.

HistoryRepository — працює з таблицею історії у базі даних.

База даних:

DownloadTable — зберігає дані про завантаження файлів.

DownloadStatisticsTable — містить статистичні дані.

DownloadHistoryTable — зберігає інформацію про історію завантажень.

Діаграма послідовностей

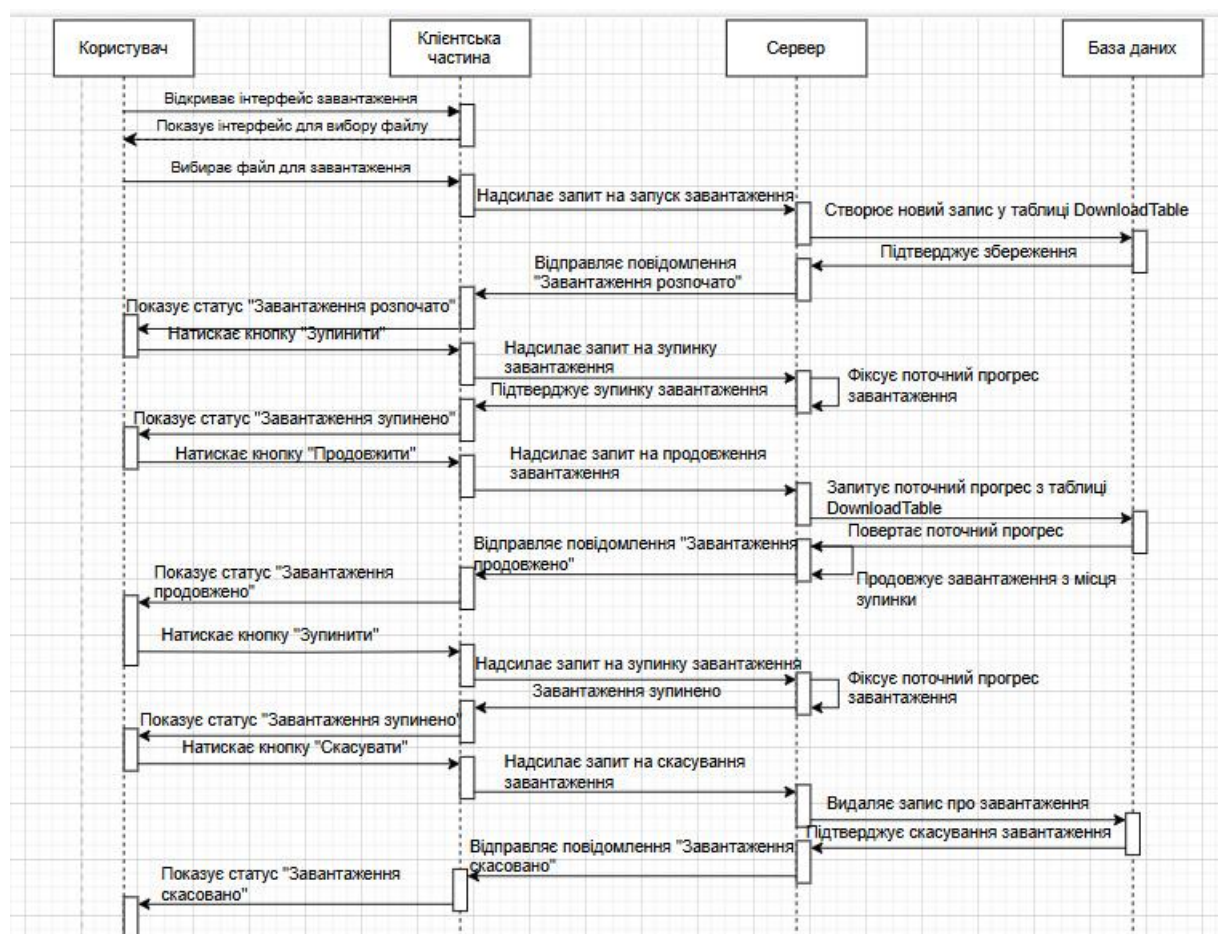


Рис. 3 - Діаграма послідовностей

Запуск завантаження файлу:

1. Користувач відкриває інтерфейс завантаження через Клієнтську частину (UI) і вибирає файл для завантаження.
2. Клієнтська частина передає запит до Контролера на запуск завантаження, передаючи шлях до файлу.
3. Контролер викликає метод `StartDownload(filePath)` в Сервісі для ініціації завантаження.
4. Сервіс створює запис про завантаження у Репозиторії, який зберігає ці дані у базі даних.
5. Після збереження даних у базі даних, Репозиторій підтверджує збереження, і Сервіс передає підтвердження назад до Контролера.

6. Контролер надсилає повідомлення про успішне запуск завантаження, а Клієнтська частина відображає користувачу статус "Завантаження розпочато".

Зупинка завантаження:

1. Користувач натискає кнопку "Зупинити" у Клієнтській частині для зупинки поточного завантаження.
2. Клієнтська частина передає запит до Контролера на зупинку завантаження.
3. Контролер викликає метод StopDownload() в Сервісі.
4. Сервіс фіксує поточний прогрес завантаження та зупиняє процес.
5. Сервіс передає підтвердження зупинки до Контролера, який надсилає повідомлення до Клієнтської частини, відображаючи користувачу статус "Завантаження зупинено".

Продовження завантаження:

1. Якщо завантаження було зупинено, Користувач може натиснути кнопку "Продовжити".
2. Клієнтська частина передає запит до Контролера на продовження завантаження.
3. Контролер викликає метод ResumeDownload() в Сервісі.
4. Сервіс звертається до Репозиторію для отримання останнього збереженого прогресу завантаження.
5. Репозиторій запитує базу даних для отримання поточного стану завантаження.
6. Після отримання прогресу з бази даних, Сервіс продовжує завантаження з місця зупинки.
7. Сервіс підтверджує продовження завантаження і передає це підтвердження Контролеру, який відправляє повідомлення користувачу про продовження процесу завантаження.

Скасування завантаження:

1. Користувач може скасувати завантаження, натиснувши кнопку "Скасувати".

2. Клієнтська частина передає запит до Контролера на скасування завантаження.
3. Контролер викликає метод `CancelDownload()` в Сервісі.
4. Сервіс видаляє запис про завантаження з Репозиторію, і Репозиторій здійснює видалення запису з бази даних.
5. Після видалення запису, Сервіс повідомляє Контролер про скасування завантаження, а Контролер передає відповідне повідомлення Клієнтській частині.
6. Клієнтська частина відображає користувачу статус "Завантаження скасовано".

Висновок

Під час розробки проекрованої системи були виконані ключові етапи: ознайомлення з теоретичними відомостями, розробка діаграм розгортання, компонентів і послідовностей. Це допомогло чітко визначити структуру системи, її компоненти та взаємодію між ними. Розроблені діаграми сприяють кращому розумінню архітектури, зменшенню ймовірності помилок і забезпечують ефективну розробку та тестування системи.