

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №2

з дисципліни «Технології розроблення програмного забезпечення»

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

Варіант №26

Виконав:
студент групи ІА-23
Мозоль В.О

Перевірив:
Мягкий М. Ю.

Київ 2024

Зміст

Тема.....	3
Мета.....	3
Завдання.....	3
Обрана тема.....	3
Короткі теоретичні відомості.....	4
Діаграма розгортання.....	7
Діаграма компонентів	9
Діаграма взаємодій та послідовностей.....	10
Висновки.	11

Тема.

Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей

Мета.

Розробка та аналіз концептуальної моделі для обраного варіанту з використанням UML-діаграм. Зокрема створення діаграми розгортання, компонентів, взаємодій та послідовностей.

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.

Обрана 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).

Короткі теоретичні відомості

1. Діаграма розгортання (Deployment Diagram)

Діаграми розгортання є важливим інструментом для візуалізації фізичного розміщення програмних компонентів у системі. Вони показують, як програмне забезпечення розгорнуте на апаратному забезпеченні, і які зв'язки існують між різними частинами системи. Основні елементи діаграми включають:

Вузли (Nodes): Це фізичні або віртуальні пристрої, які можуть містити програмне забезпечення. Вузли поділяються на:

Пристрої (Devices): Фізичні елементи, такі як сервери, комп'ютери, маршрутизатори. Вони представляють апаратне забезпечення, на якому запускається програмне забезпечення.

Середовища виконання (Execution Environments): Програмні платформи, такі як операційні системи або сервери додатків, які можуть містити інші програмні компоненти. Вони забезпечують середовище для виконання програмного забезпечення.

Зв'язки (Connections): Визначають, як вузли взаємодіють між собою, зазвичай через мережеві протоколи або інші технології зв'язку. Зв'язки можуть мати атрибути, такі як назва протоколу (наприклад, HTTP, IPC) або технології (наприклад, .NET Remoting, WCF).

Артефакти (Artifacts): Файли або інші фізичні прояви програмного забезпечення, такі як виконувані файли, бібліотеки, конфігураційні файли. Вони представляють програмне забезпечення, яке розгортається на вузлах.

Діаграми розгортання можуть бути описовими, без конкретних деталей про обладнання, або екземплярними, з конкретними деталями про апаратне забезпечення та програмне забезпечення. Описові діаграми корисні на ранніх етапах проектування, тоді як екземплярні діаграми використовуються на завершальних стадіях розробки.

2. Діаграма компонентів (Component Diagram)

Діаграми компонентів описують структуру системи через її модулі або компоненти. Вони допомагають зрозуміти, як різні частини системи взаємодіють одна з одною. Основні види діаграм компонентів:

Логічні: Відображають систему як набір автономних модулів, які взаємодіють між собою. Це допомагає візуалізувати архітектуру системи на концептуальному рівні. Кожен компонент може бути взаємозамінним і не обов'язково знаходиться в межах одного фізичного пристрою.

Фізичні: Показують, як компоненти розподілені між різними фізичними вузлами системи. Цей підхід застарів і зазвичай замінюється діаграмами розгортання.

Виконувані: Кожен компонент представляє собою файл або набір файлів, які можуть бути виконані, такі як .exe, бібліотеки або HTML-сторінки. Це дозволяє візуалізувати систему на рівні виконуваних файлів або процесів.

Діаграми компонентів допомагають візуалізувати загальну структуру коду, специфікувати виконувані варіанти системи та забезпечити повторне використання коду. Вони також можуть включати інтерфейси та схеми баз даних для більш детального уявлення про систему.

3. Діаграма взаємодій та послідовностей (Interaction and Sequence Diagrams)

Діаграми послідовностей використовуються для моделювання динамічної поведінки системи, показуючи, як об'єкти взаємодіють один з одним у певній послідовності. Вони допомагають зрозуміти, як різні частини системи взаємодіють у часі. Основні елементи включають:

Стан дії (Action State): Відображає виконання окремих дій або кроків алгоритму. Кожна дія має вхідний і вихідний перехід. Стан дії зазвичай моделює один крок виконання алгоритму або потоку управління.

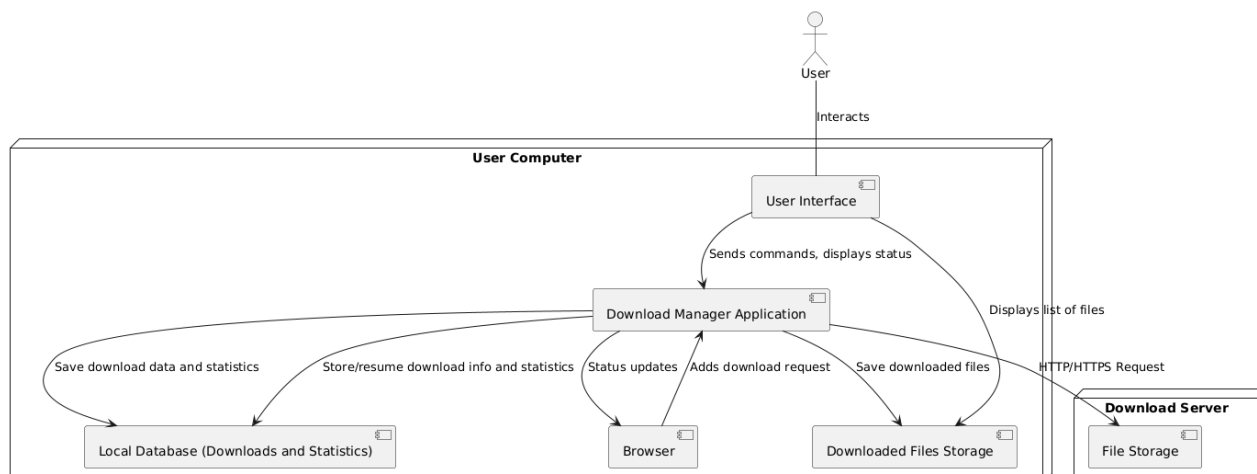
Переходи: Нетриггерні переходи, які виконуються після завершення дії. Вони можуть бути умовними, з використанням сторожових умов. Переходи дозволяють моделювати розгалуження та паралельні процеси.

Дорожки (Swimlanes): Використовуються для розподілу дій між різними підрозділами організації, що дозволяє моделювати бізнес-процеси. Кожна

дорожка представляє окремий підрозділ або роль, відповідальну за виконання певних дій.

Діаграми взаємодій та послідовностей допомагають візуалізувати алгоритми виконання, потоки управління та бізнес-процеси, фокусуючи увагу на результатах і змінах стану системи. Вони є важливим інструментом для розуміння динамічної поведінки системи та взаємодії між її компонентами.

Діаграма розгортання



Компоненти діаграми розгортання

Клієнтські вузли (Client Nodes):

- Користувач – комп'ютер користувача з встановленим додатком Download manager.
- Браузер – інтеграція з браузерами (Firefox, Opera, Internet Explorer, Chrome).
- Серверні вузли (Server Nodes):
- Сервер завантаження (Download Server) – сервер, з якого завантажуються файли через протоколи HTTP/HTTPS.

Компоненти та модулі:

Download Manager Application:

- Download Controller – управляє процесом завантаження (перевірка стану, відновлення завантажень, розподіл швидкості).
- Protocol Handler – обробляє протоколи HTTP/HTTPS для завантаження файлів.
- Resume Manager – забезпечує функцію продовження завантажень із місця зупинки.
- Browser Integration Module – компонент для інтеграції з браузерами для автоматичного додавання завантажень.
- Statistics Module – ведення статистики завантажень (кількість, швидкість, обсяг даних тощо).

- Database (локальна) – для зберігання даних про завантаження (статистика, статуси, посилання).

Мережеві підключення:

- HTTP/HTTPS – з'єднання між Download Manager і Download Server..
- Database Connection – зв'язок із локальною базою даних.

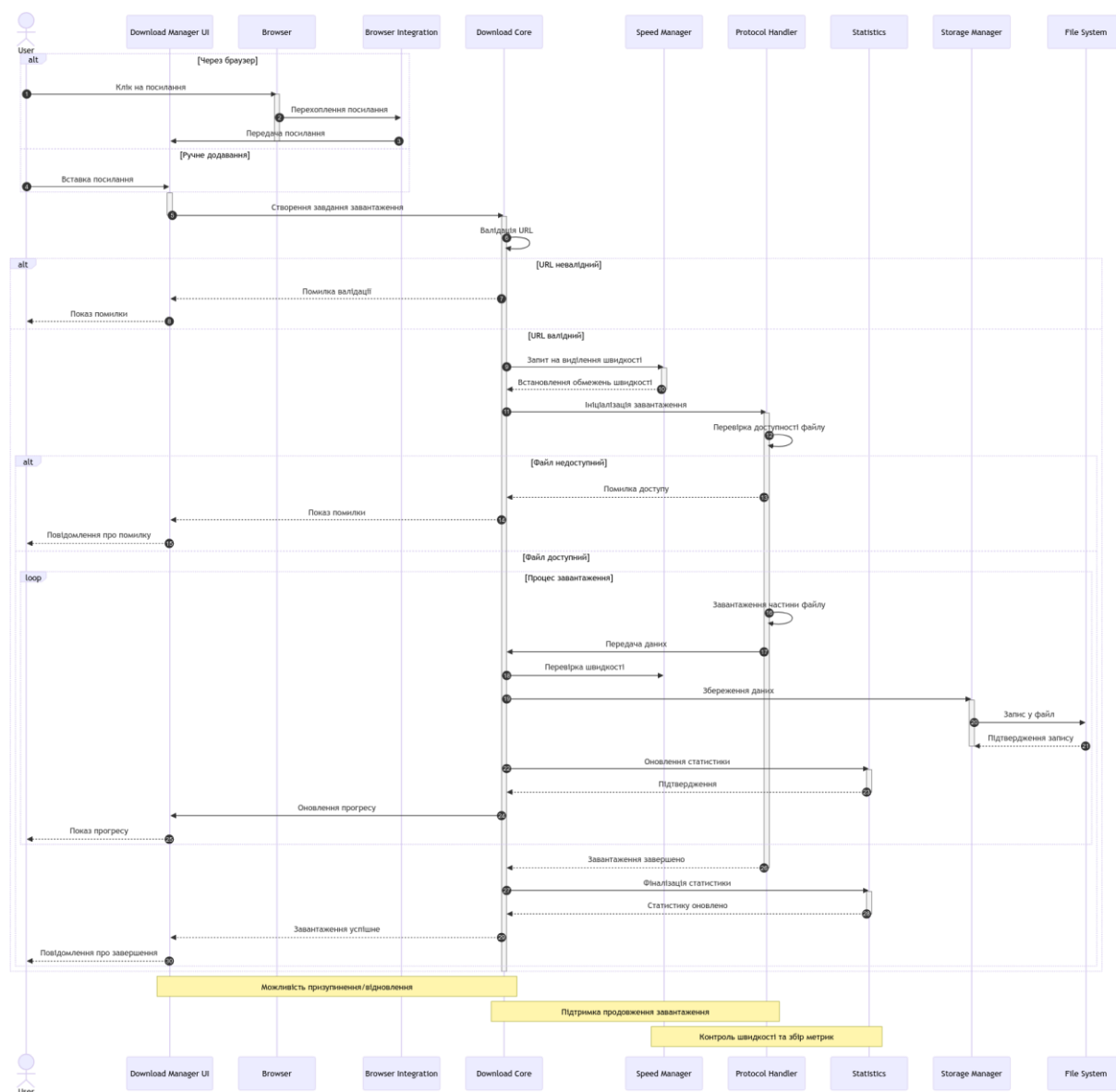
Опис діаграми:

- User Computer: представляє комп'ютер користувача з Download Manager Application, Browser та Local Database.
- Download Server: містить File Storage, де розташовані файли для завантаження.

Основні зв'язки:

- User ініціює завантаження через Download Manager Application.
- Download Manager Application звертається до File Storage на Download Server через HTTP/HTTPS.
- Local Database (Downloads and Statistics): усі дані про завантаження та статистику зберігаються локально на комп'ютері користувача.
- Browser інтегрується з Download Manager Application для додавання запитів на завантаження, а також отримує оновлення статусу.
- Downloaded Files Storage: окремий вузол на комп'ютері користувача для збереження всіх завантажених файлів.
- User Interface (UI): Інтерфейс для взаємодії користувача з Download Manager Application, який дозволяє відображати статус завантажень, список файлів та інші елементи.

Діаграма взаємодій та послідовностей



На діаграмі зображено послідовність дій при завантаженні файлу:

1. Ініціювання завантаження:

Через браузер:

- Користувач клікає на посилання в браузері
- Модуль браузерної інтеграції перехоплює запит
- Download Core створює нове завдання

Через ручне додавання посилання в інтерфейсі програми

2. Додаткові перевірки:

- Валідація URL
- Перевірка доступності файлу
- Помилки валідації URL

- Помилки доступу до файлу
- 3. Підготовка до завантаження:
 - Запит на виділення швидкості
 - Ініціалізація протоколу
 - Підготовка сховища
- 4. Процес завантаження (цикл):
 - Завантаження частин файлу
 - Контроль швидкості
 - Збереження даних
 - Оновлення статистики
 - Показ прогресу завантаження
 - Відображення помилок
 - Повідомлення про статус завантаження
- 5. Завершення:
 - Фіналізація статистики
 - Повідомлення про завершення
 - Оновлення інтерфейсу

Висновки.

Отже на цій лабораторній роботі, було розроблено та проаналізовано концептуальну модель системи менеджера завантажень з використанням UML-діаграм. Зокрема створення діаграми розгортання, компонентів, взаємодій та послідовностей.