



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
**Технологія розроблення програмного
забезпечення**

«Менеджер скачувань. Діаграма розгортання. Діаграма компонентів.
Діаграма взаємодій та послідовностей.»

Варіант 26

Виконала
студентка групи ІА-13
Танасієнко Анастасія Вячеславівна

Перевірив:
Мягкий Михайло
Юрійович

Київ 2023р.

Мета: Навчитися розробляти діаграму розгортання. діаграма компонентів. діаграма взаємодій та послідовностей.

Хід роботи

..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).

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.

Посилання на діаграми :

<https://drive.google.com/file/d/1RjYv5jiscdK4jNcZRTToAeIPAiTSwsnUI/view?usp=sharing>

1. Діаграма розгортання

Діаграма розгортання (Deployment Diagram) є одним із видів діаграм UML (Unified Modeling Language), що використовується для моделювання фізичної архітектури програмних систем.

Ця діаграма дозволяє візуалізувати, як програмні та апаратні компоненти системи розташовані в просторі та як вони взаємодіють між собою та з навколишнім середовищем. Вона допомагає розібратися у фізичній конфігурації системи, її складових та їх розміщенні на апаратних пристроях, серверах, мережах тощо.

Основні елементи діаграми розгортання:

- **Вузли (Nodes):** Представляють апаратні або програмні ресурси, на яких працюють компоненти системи. Це можуть бути, наприклад, фізичні сервери, віртуальні машини, контейнери, додатки тощо.
- **Артефакти (Artifacts):** Представляють файли або набори файлів, що містять додаткові дані для компонентів.
- **Зв'язки (Associations):** Вказують на взаємозв'язки між вузлами та артефактами, що вказують на те, як компоненти використовують або зберігають дані на різних ресурсах.
- **Джерела та приймачі (Source and Destination):** Показують, як компоненти взаємодіють один з одним, вказуючи на потоки даних чи комунікаційні канали.
- Діаграми розгортання важливі для розробки систем, які мають фізичний аспект, такі як розподілені програми, мережеві системи, хмарні додатки та інші. Вони допомагають команді розробників та архітекторам зрозуміти та спілкуватися щодо фізичної конфігурації системи.

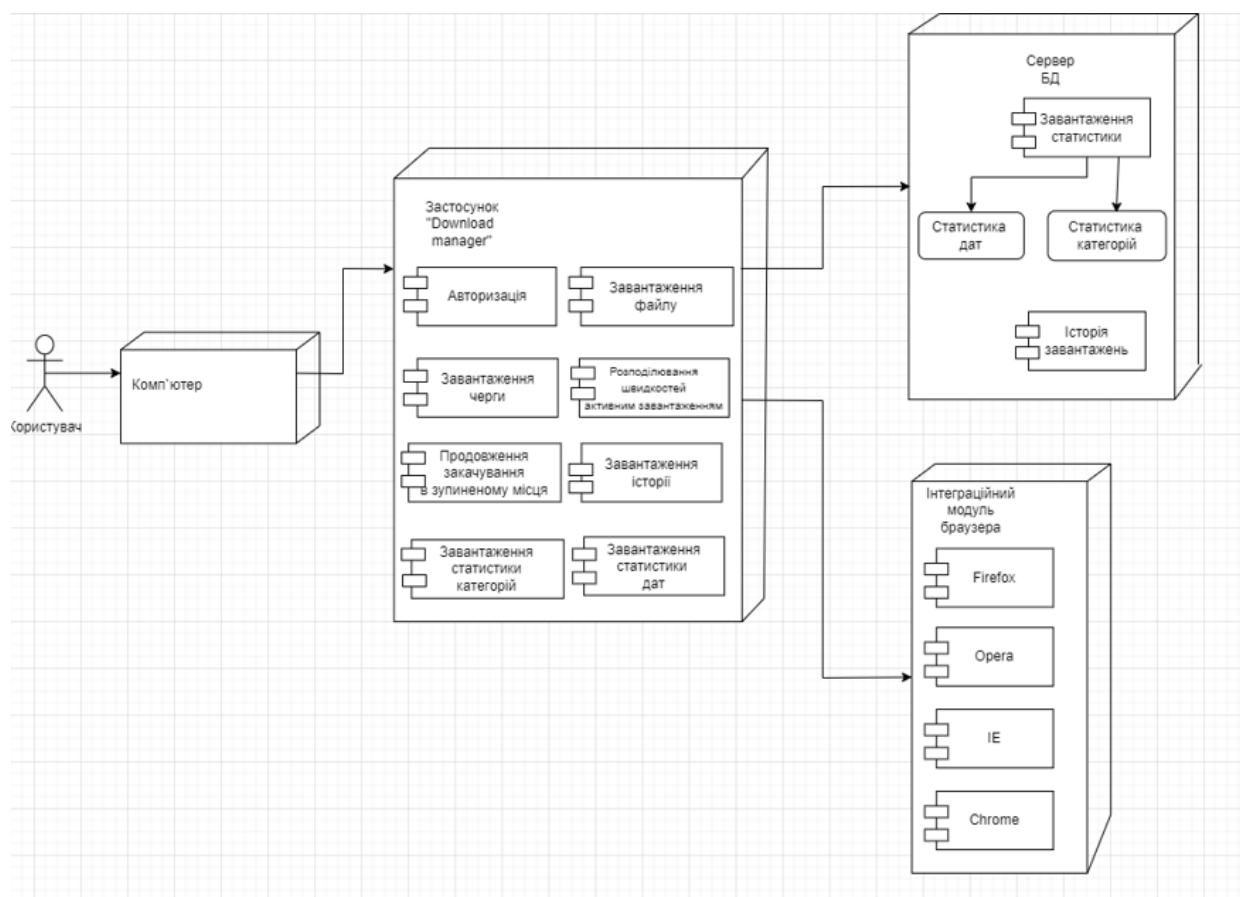


Рис. 1 Діаграма розгортання

2. Діаграма компонентів

Діаграма компонентів (Component Diagram) є одним із видів діаграм UML (Unified Modeling Language) і використовується для моделювання внутрішньої структури та організації компонентів програмної системи.

Основні елементи діаграми компонентів включають:

- Компоненти (Components): Вони представляють собою найменші виконувані частини програми, які можуть бути відокремлені та замінені. Компоненти можуть бути, наприклад, класами, пакетами, бібліотеками, сервісами тощо.
- Інтерфейси (Interfaces): Вказують на те, які послуги або методи надаються компонентами для взаємодії з іншими компонентами.
- Залежності (Dependencies): Показують, як компоненти можуть використовувати послуги або функціонал інших компонентів.
- Залежності від артефактів (Artifact Dependencies): Показують, як компоненти можуть бути пов'язані з артефактами, такими як файли, бібліотеки тощо.
- Артефакти (Artifacts): Вони представляють файли, бібліотеки, модулі або інші об'єкти, що використовуються компонентами для виконання своїх функцій.

Діаграми компонентів використовуються для показу внутрішньої архітектури програмної системи, вказуючи, як компоненти взаємодіють між собою та з іншими частинами системи. Вони допомагають розробникам та архітекторам у розумінні та аналізі складної системи, а також у прийнятті рішень щодо її дизайну та розробки.

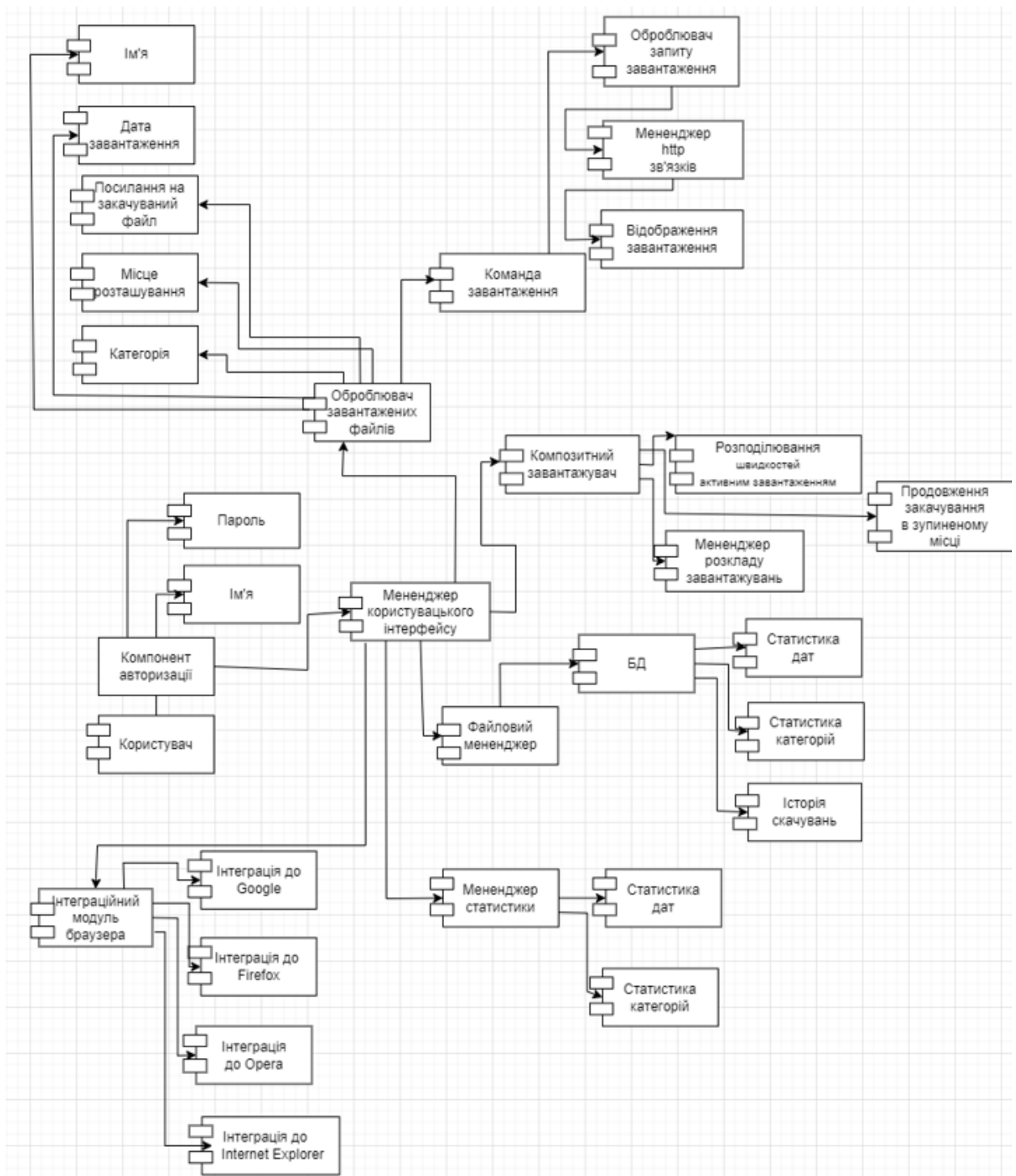


Рис. 2 Діаграма компонентів

3. Діаграма взаємодій та послідовностей

- Діаграма послідовностей (Sequence Diagram):

Діаграма послідовностей також відображає взаємодії між об'єктами, але робить акцент на послідовності та порядку цих взаємодій. Вона показує, як об'єкти взаємодіють один з одним впродовж певного періоду часу. Ця діаграма допомагає визначити, які об'єкти виконують які дії та в якому порядку.

Основні компоненти діаграми послідовностей включають об'єкти (які позначаються вертикальними лініями, що показують часовий інтервал), повідомлення (які позначаються стрілками, що з'єднують об'єкти) та умови.

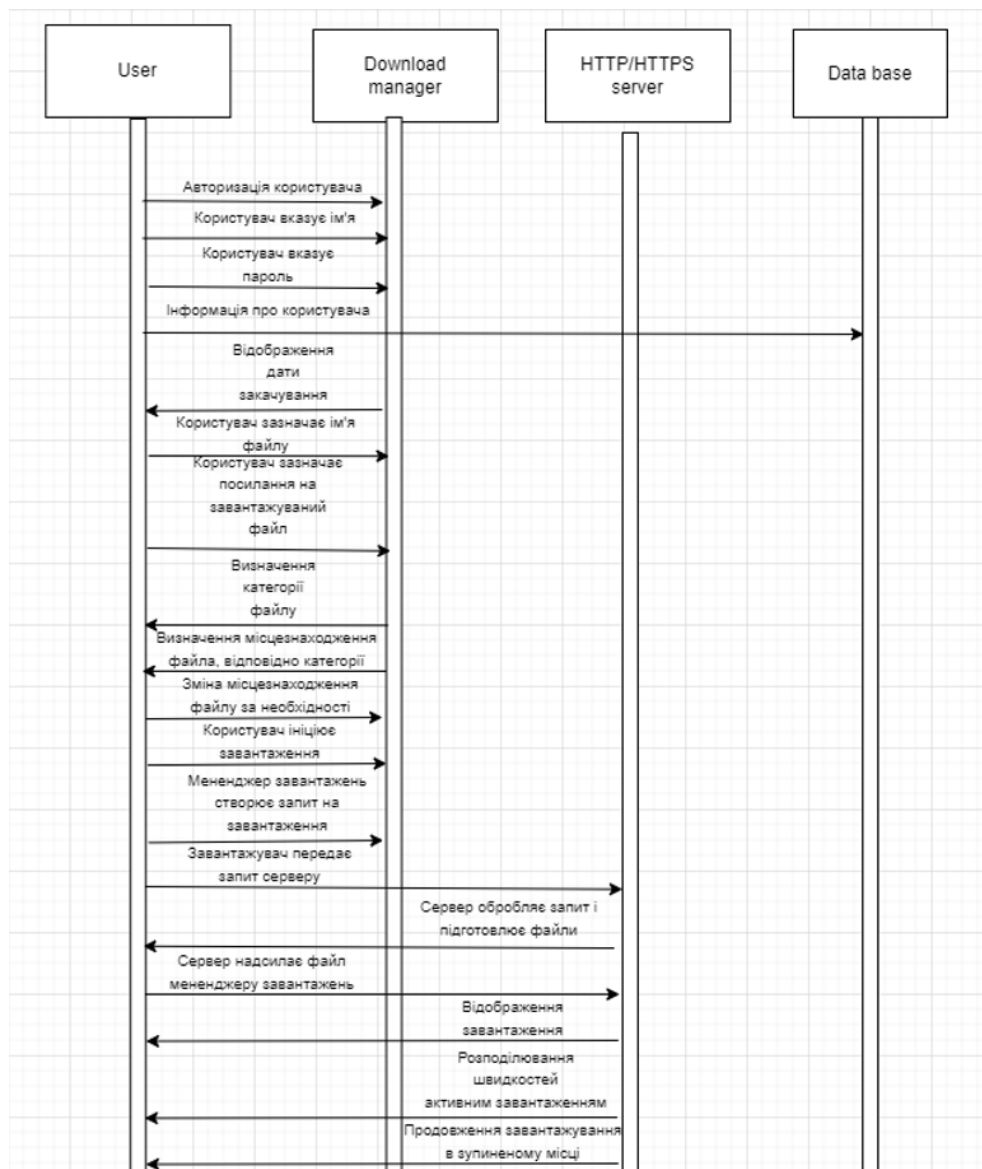


Рис. 3.1 Діаграма взаємодій та послідовностей

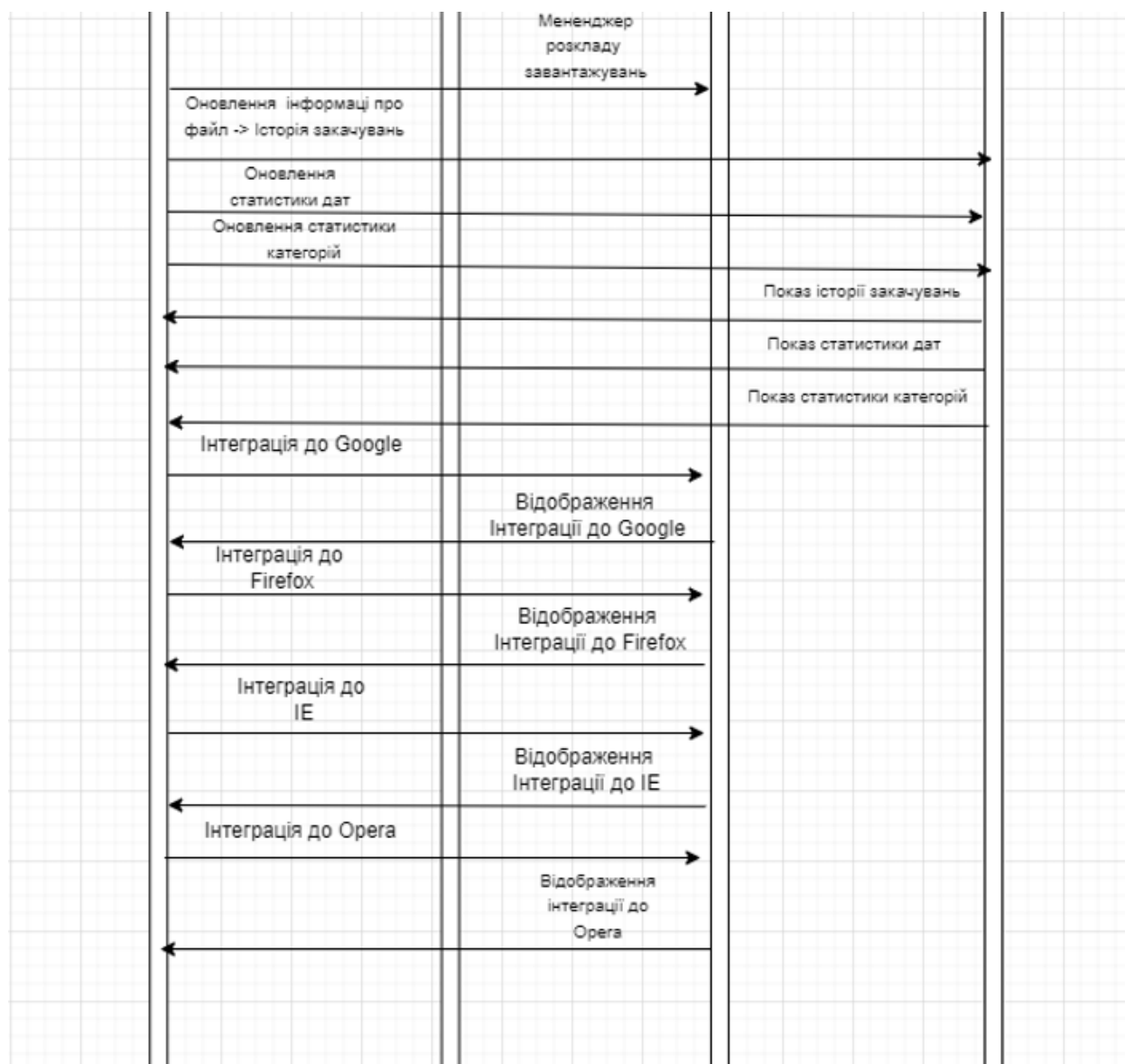


Рис. 3.2 Діаграма взаємодій та послідовностей

Висновок:

Лабораторна робота була спрямована на навчання розробки різних типів діаграм у контексті моделювання програмних систем. У ході виконання лабораторної роботи були вивчені та застосовані такі види діаграм:

- Діаграма розгортання: Була використана для візуалізації фізичної архітектури системи, її компонентів та взаємозв'язків. Ця діаграма дозволяє легко сприймати, як різні компоненти взаємодіють у фізичному середовищі.
- Діаграма компонентів: Була використана для моделювання внутрішньої структури кожного компонента системи та їх залежностей один від одного. Ця діаграма надає глибокий уявлення про компоненти системи та їх функціональність.
- Діаграма взаємодій та послідовностей: Була використана для представлення взаємодій між об'єктами або компонентами системи на високому рівні абстракції. Ця діаграма допомагає у розумінні процесів та потоків даних в системі. У процесі роботи було розглянуто принципи та правила побудови кожного типу діаграми, а також використання їх в реальних сценаріях розробки програмних систем. Ця лабораторна робота надала необхідні навички для ефективного моделювання та аналізу програмних систем за допомогою різних видів діаграм.