



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”

Виконав
студент групи ІА–22:
Гордун Д.В.

Перевірів
Мягкий М.Ю.

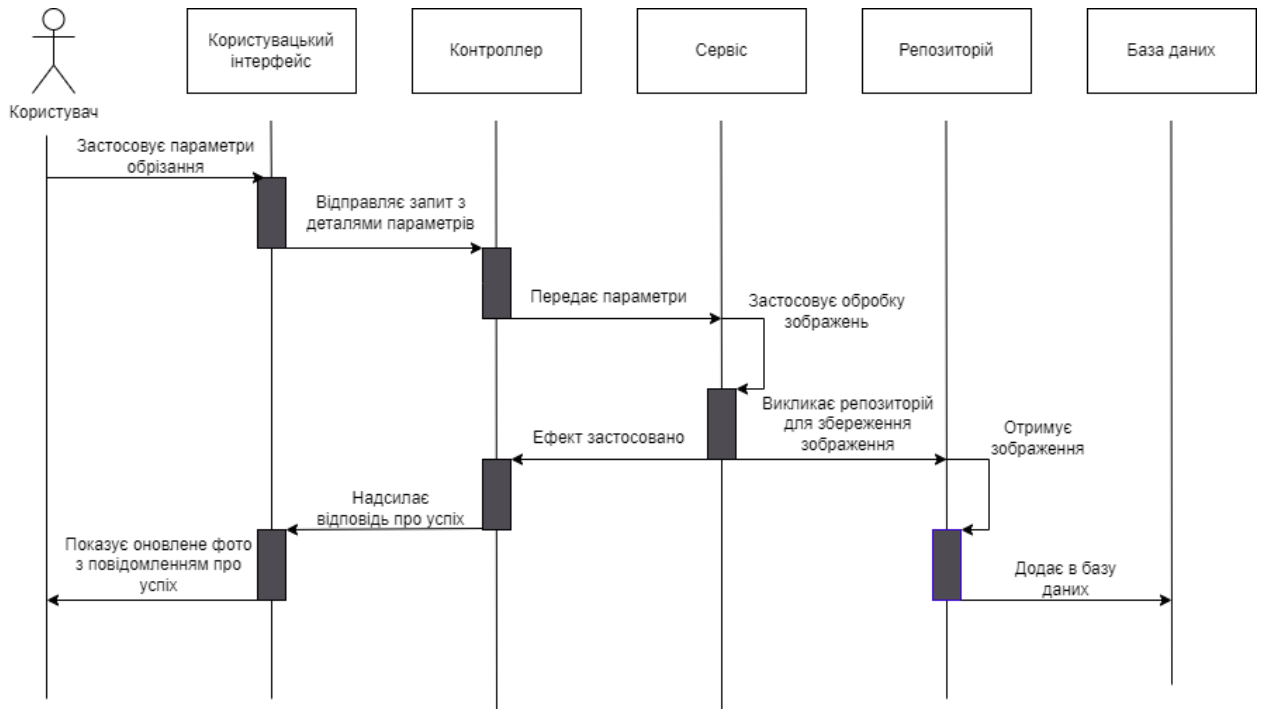
Київ 2024

Хід роботи

1. Реалізувати частину функціональності системи, описану діаграмами з попередньої лабораторної роботи згідно обраної теми

Код можна побачити за [посиланням](#)

2. Діаграма послідовності для функції обрізання



Діаграма послідовностей призначена для моделювання послідовності взаємодій між об'єктами або компонентами системи в часі. Вона показує, як об'єкти обмінюються повідомленнями (викликами методів) для виконання певної функції або сценарію. Ця діаграма дозволяє візуалізувати порядок передачі повідомлень між компонентами системи, що допомагає зрозуміти їхню динамічну поведінку та взаємодію під час виконання певної операції.

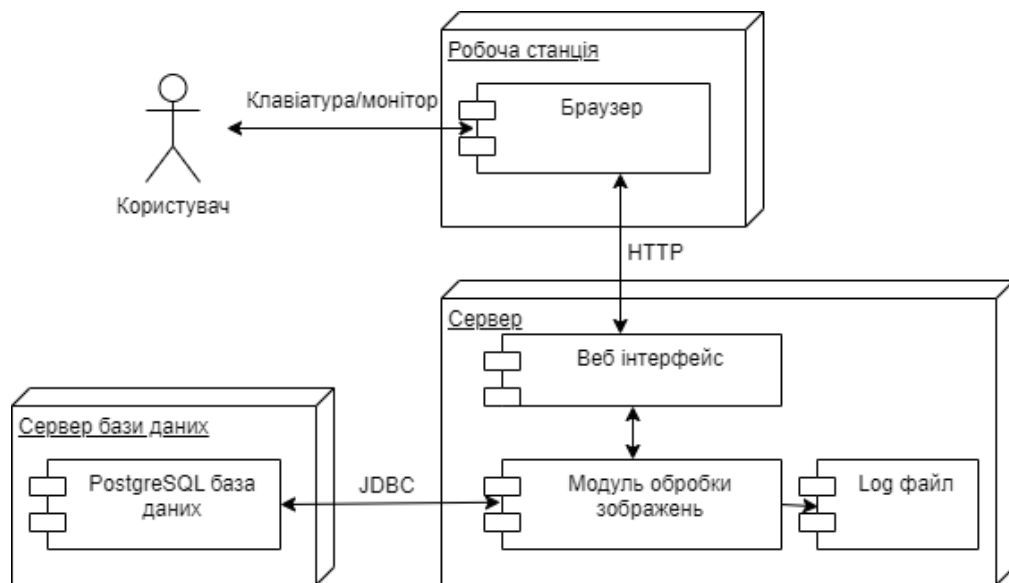
Основне призначення діаграми:

- Відобразити взаємодії між об'єктами у визначеній послідовності.
- Візуалізувати процес обробки запитів у системі.
- Допомогти проектувати функціональність системи, розуміючи, які об'єкти беруть участь у взаємодії.
- Пояснити бізнес-процеси і програмну логіку на рівні послідовності дій.

Основні елементи діаграми послідовностей:

- **Актори** : Зовнішні користувачі або системи, які взаємодіють з об'єктами всередині системи (наприклад, користувач або інший програмний модуль).
- **Об'єкти** : Системні елементи, які обмінюються повідомленнями. Вони можуть представляти класи або компоненти системи (наприклад, інтерфейс користувача, контролери, сервіси).
- **Лінії життя**: Вертикальні лінії, що йдуть від об'єктів, які відображають період існування або активності об'єкта під час взаємодії. Лінія життя починається з об'єкта і триває доти, доки об'єкт активний або взаємодіє з іншими.
- **Повідомлення**: Стрілки між об'єктами, які показують передачу інформації або виклик методів. Повідомлення можуть бути синхронними (виконуються і чекають відповіді) або асинхронними (виконуються без очікування відповіді).
- **Активації**: Прямокутні блоки на лінії життя, що показують період, коли об'єкт виконує певну дію або знаходиться в активному стані.

3. Діаграма розгортання



Діаграма розгортання відображає фізичне розташування компонентів системи на конкретному обладнанні. Вона показує, на якому фізичному пристрої або програмному середовищі запускається кожен компонент системи, а також як ці вузли пов'язані між собою. Вузли можуть включати артефакти (файли, скрипти, бази даних), які є фізичними об'єктами програмного забезпечення.

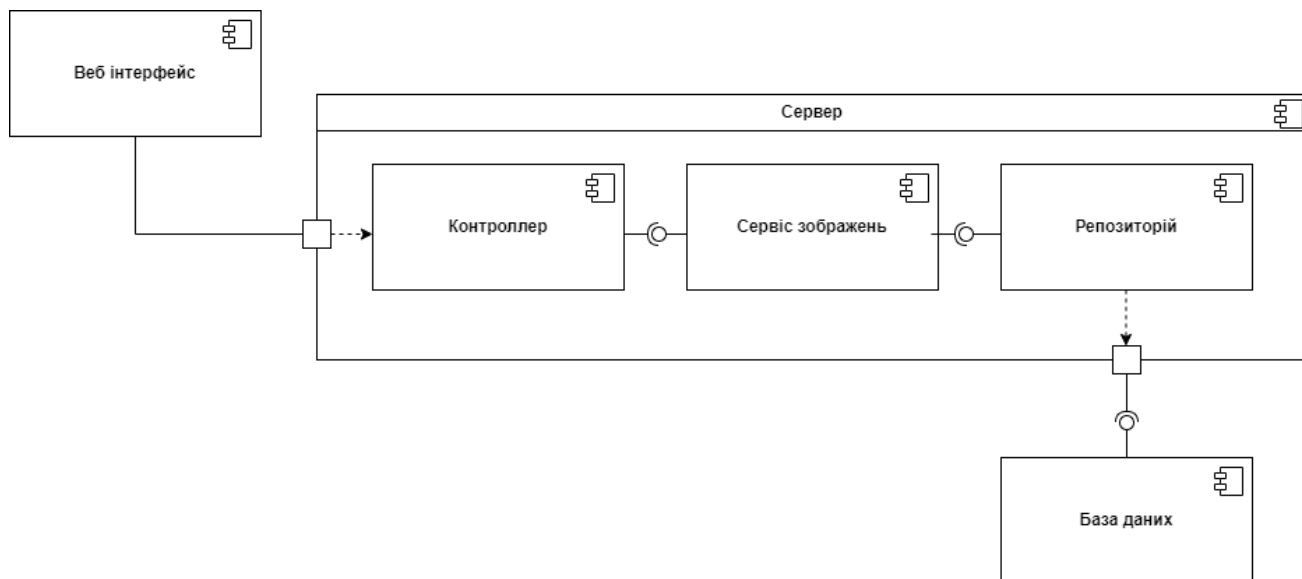
Існує два типи вузлів на діаграмі розгортання:

- **Пристрій (Device)** — фізичне обладнання, таке як комп'ютери, сервери або інші пристрої, що можуть містити програмне забезпечення.
- **Середовище виконання (Execution Environment)** — це програмне забезпечення, яке може запускати інші програми. Наприклад, операційна система, віртуальні машини, веб-сервери.

Є три зв'язки:

- **Протокол:** зв'язок може бути позначений назвою протоколу, за допомогою якого відбувається обмін інформацією між вузлами, наприклад, HTTP, IPS.
- **Технології:** у назві зв'язку можуть бути вказані технології, що використовуються для забезпечення взаємодії, наприклад, .NET Remoting, WCF.
- **Атрибути зв'язків:** можуть включати кількість підключених клієнтів до одного сервера або інші характеристики.

4. Діаграма компонентів



Діаграма компонентів — це тип діаграми UML, який використовується для візуалізації архітектури програмного забезпечення на рівні окремих модулів або компонентів. Вона показує, як різні компоненти системи взаємодіють між собою і як ці компоненти об'єднуються для виконання функцій системи. Кожен компонент може бути окремою частиною системи, яку можна незалежно розробляти, тестувати та використовувати повторно.

На діаграмі компонентів присутні такі основні елементи:

- **Компоненти:** Це окремі модулі системи, які представляють певні функціональні частини програмного забезпечення. Компоненти можуть бути логічними (автономні модулі) або фізичними (виконувані файли, бази даних, таблиці тощо).
- **Інтерфейси:** Компоненти взаємодіють через інтерфейси, які представляють собою точки доступу до їхньої функціональності.
- **Артефакти:** Файли або фізичні одиниці, які можуть бути виконуваними файлами, сторінками HTML, базами даних, тощо. Вони використовуються для показу фізичних компонентів на діаграмі.
- **Вузли (nodes):** Місця розташування компонентів (фізичні пристрої або сервери).

Зв'язки на діаграмі компонентів представляють взаємодію між компонентами. Це можуть бути:

- **Інтерфейсні зв'язки:** Показують, як один компонент використовує інтерфейс іншого компонента для доступу до його функцій.
- **Залежності:** Вказують на те, що один компонент залежить від іншого для виконання певної функціональності.
- **Фізичні зв'язки:** Вказують на фізичне розміщення компонентів на різних вузлах системи.

З видів діаграм взаємодії можна виділити діаграму послідовностей, про яку вже написано у звіті, і діаграму комунікації. Вона фокусується на структурованій взаємодії між об'єктами системи та відображає зв'язки і обмін повідомленнями між цими об'єктами для виконання певної функціональності.

Висновок: на цій лабораторній роботі я ознайомився з діаграмами послідовностей, компонентів і розгортання.