



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
Технології розроблення програмного забезпечення
ДІАГРАМА РОЗГОРТАННЯ. ДІАГРАМА КОМПОНЕНТІВ.
ДІАГРАМА ВЗАЄМОДІЙ ТА ПОСЛІДОВНОСТЕЙ.
FTP-server

Виконав

студент групи ІА–22:

Білокур Євгеній

Перевірив:

Мягкий Михайло Юрійович

Київ 2024

Зміст

Хід роботи.....	3
Теоретичні відомості.....	3
Реалізація попередньої лабораторної.....	4
Діаграма послідовностей.....	6
Опис діаграми послідовностей.....	6
Діаграма розгортання.....	7
Опис діаграми розгортання.....	7
Діаграма компонентів.....	8
Опис діаграми компонентів.....	8
Висновки та код.....	9

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

Мета: Проаналізувати тему, розробити діаграму розгортання, діаграму компонентів, діаграму взаємодій та послідовностей.

Хід роботи

..22 FTP-server (state, builder, memento, template method, visitor, client-server)

FTP-сервер повинен вміти коректно обробляти і відправляти відповіді по протоколу FTP, з можливістю створення користувачів (з паролями) і доступних їм папок, розподілу прав за стандартною схемою (rwe), ведення статистики з'єднань, обмеження максимальної кількості підключень і максимальної швидкості поширення глобально і окремо для кожного облікового запису.

Теоретичні відомості

У цій лабораторній роботі розглядаються кілька ключових діаграм, які є частиною UML (Unified Modeling Language) та необхідні для моделювання складних програмних систем. Зокрема, діаграма розгортання, діаграма взаємодій та послідовностей.

1. Діаграма розгортання (Deployment Diagram)

Діаграма розгортання показує фізичне розташування компонентів системи на апаратних ресурсах. Ця діаграма допомагає зрозуміти, як програма функціонує на різних рівнях системи та які фізичні або віртуальні машини беруть участь у її роботі.

Основними елементами діаграми розгортання є:

- **Ноди** – фізичні або віртуальні машини, на яких розгорнуто програму (сервери, клієнтські машини тощо).
- **Компоненти** – окремі частини системи (програми або модулі), які розгорнуті на певних нодах.
- **З'єднання** – зв'язки між нодами, що відображають взаємодію між ними, наприклад через мережу або інтерфейси.

2. Діаграма компонентів (Component Diagram)

Діаграма компонентів відображає логічну структуру системи, зокрема, які компоненти входять до складу програмного забезпечення і як вони

взаємодіють між собою. Компонент може бути представлений у вигляді бібліотеки, модуля або іншого незалежного елемента програмного коду.

Основні елементи діаграми компонентів:

- **Компоненти** – модулі або частини системи, які виконують певну функцію (наприклад, обробка даних, управління з'єднаннями).
- **Інтерфейси** – методи взаємодії між компонентами.
- **Зв'язки** – асоціації між компонентами, що показують, які саме компоненти використовують інші.

3. Діаграма послідовностей (Sequence Diagram)

Діаграма послідовностей є одним із видів діаграм взаємодій і використовується для моделювання процесів у системі з погляду часової послідовності виконання дій. Вона відображає, як різні елементи системи обмінюються повідомленнями для виконання певної функції.

Основні елементи діаграми послідовностей:

- **Актори** – користувачі або зовнішні системи, що ініціюють певні дії.
- **Об'єкти** – компоненти або класи системи, що беруть участь у процесі.
- **Повідомлення** – передача даних між об'єктами.
- **Часова шкала** – вертикальна лінія, що показує часову послідовність подій.

Реалізація попередньої лабораторної

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

Під час реалізації частини функціональності системи, описаної діаграмами з попередньої лабораторної роботи, було створено програмні класи сутностей User, Role, Group, Directory, File, ConnectionLimits, ConnectionStatistics, GlobalLimits, а також класи для обробки команд: CommandHandler, UserCommandHandler, FileCommandHandler, AdminCommandHandler. Окрім цього, були реалізовані класи для роботи з базою даних у вигляді репозиторіїв: UserRepository, GroupRepository, FileRepository, DirectoryRepository, ConnectionLimitsRepository, ConnectionStatisticsRepository, GlobalLimitsRepository. Нижче можна побачити структуру проєкту після реалізації частини функціональності системи (рис. 1).

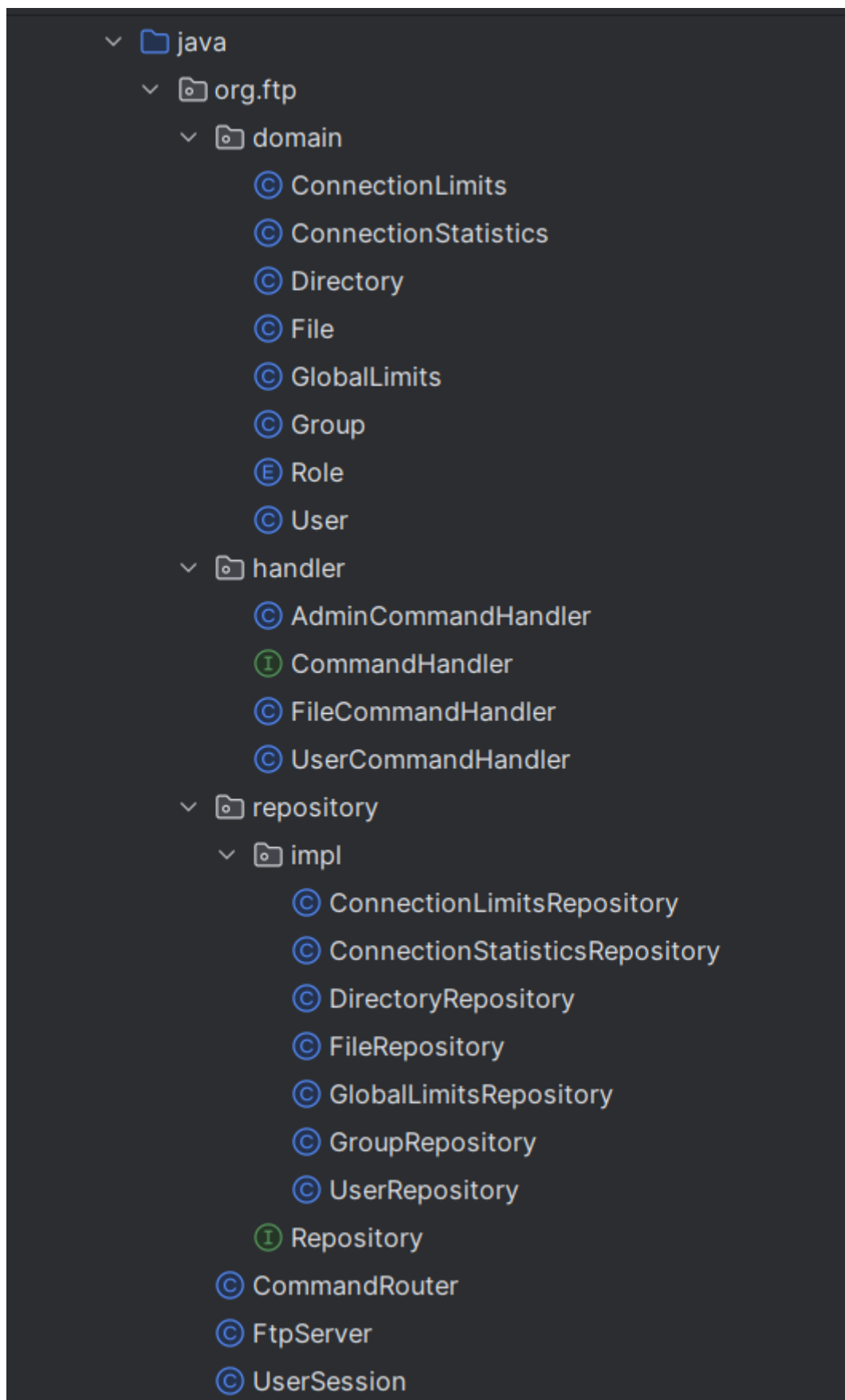


Рисунок 1 – Структура проекта

Діаграма послідовностей

2. Спроекувати діаграму послідовностей для одного із процесів розроблюваної системи

Діаграма послідовностей для процесу завантаження файлу на сервер зображена на рисунку 2.

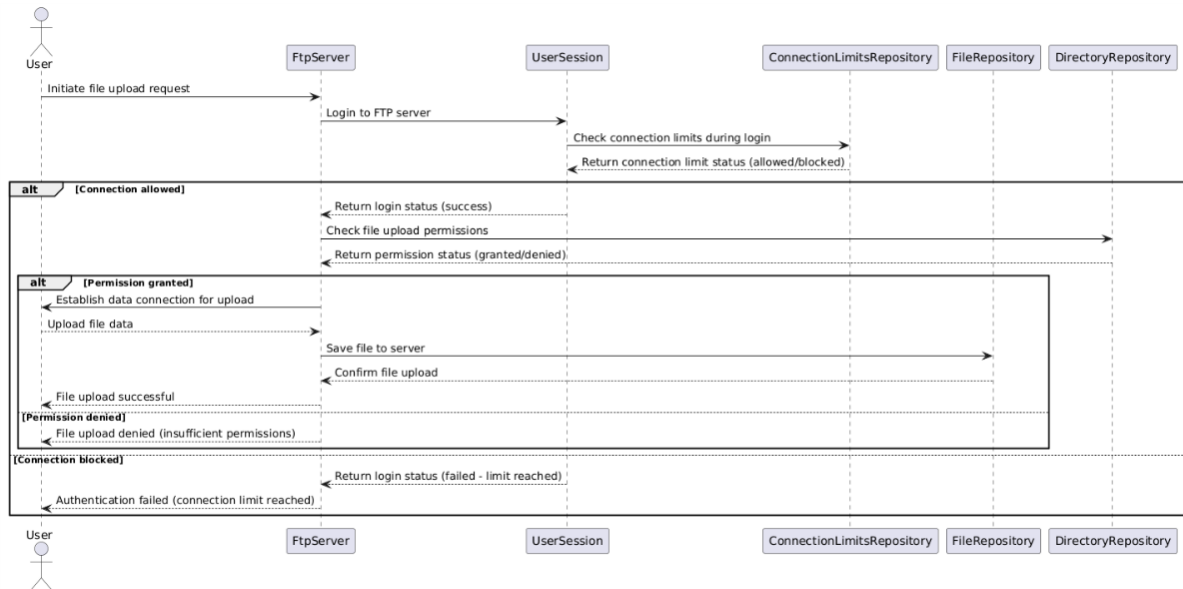


Рисунок 2 – Діаграма послідовностей

Опис діаграми послідовностей

Ця діаграма послідовностей моделює процес завантаження файлу на FTP-сервер.

- Ініціація запиту:** Користувач ініціює запит на завантаження файлу.
- Авторизація:** FTP-сервер передає запит на авторизацію до об'єкта `UserSession`, який перевіряє обмеження з'єднань у `ConnectionLimitsRepository`.
 - Якщо обмеження не перевищені, з'єднання дозволено, і користувач може продовжувати завантаження.
 - Якщо ліміт досягнутий, авторизація не пройде, і завантаження блокується.
- Перевірка прав:** Після успішного логіну перевіряються права на завантаження файлу.
 - Якщо права надані, користувач передає файл на сервер.
 - Якщо права відсутні, запит на завантаження буде відхилений.

4. **Передача файлу:** Якщо дозволено, дані передаються, файл зберігається у FileRepository та підтверджується завершення завантаження.

Альтернативні шляхи показують, коли з'єднання або доступ блокуються через досягнення лімітів або відсутність прав.

Діаграма розгортання

3. Спроекувати діаграму розгортання для розроблюваної системи.

Діаграма розгортання системи зображена на рисунку 3.

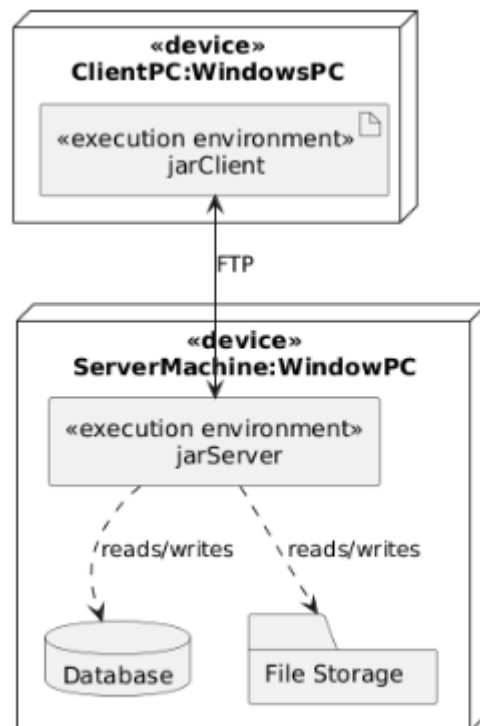


Рисунок 3 – Діаграма розгортання

Опис діаграми розгортання

Ця діаграма розгортання відображає архітектуру клієнт-серверної системи:

1. Клієнтська частина (ClientPC:WindowsPC):
 - Містить середовище виконання jarClient
2. Серверна частина (ServerMachine:WindowPC):
 - Включає середовище виконання jarServer
 - Взаємодіє з базою даних та файловим сховищем

- Зв'язок між клієнтом і сервером здійснюється через FTP протокол
- Сервер має доступ для читання/запису до бази даних та файлового сховища

Діаграма ілюструє розподіл компонентів системи між клієнтським пристроєм та серверною машиною, а також їхню взаємодію та зв'язок із зовнішніми ресурсами даних.

Діаграма компонентів

- Спроектувати діаграму компонентів для розроблюваної системи.
Діаграма компонентів розроблюваної системи зображена на рисунку 4.

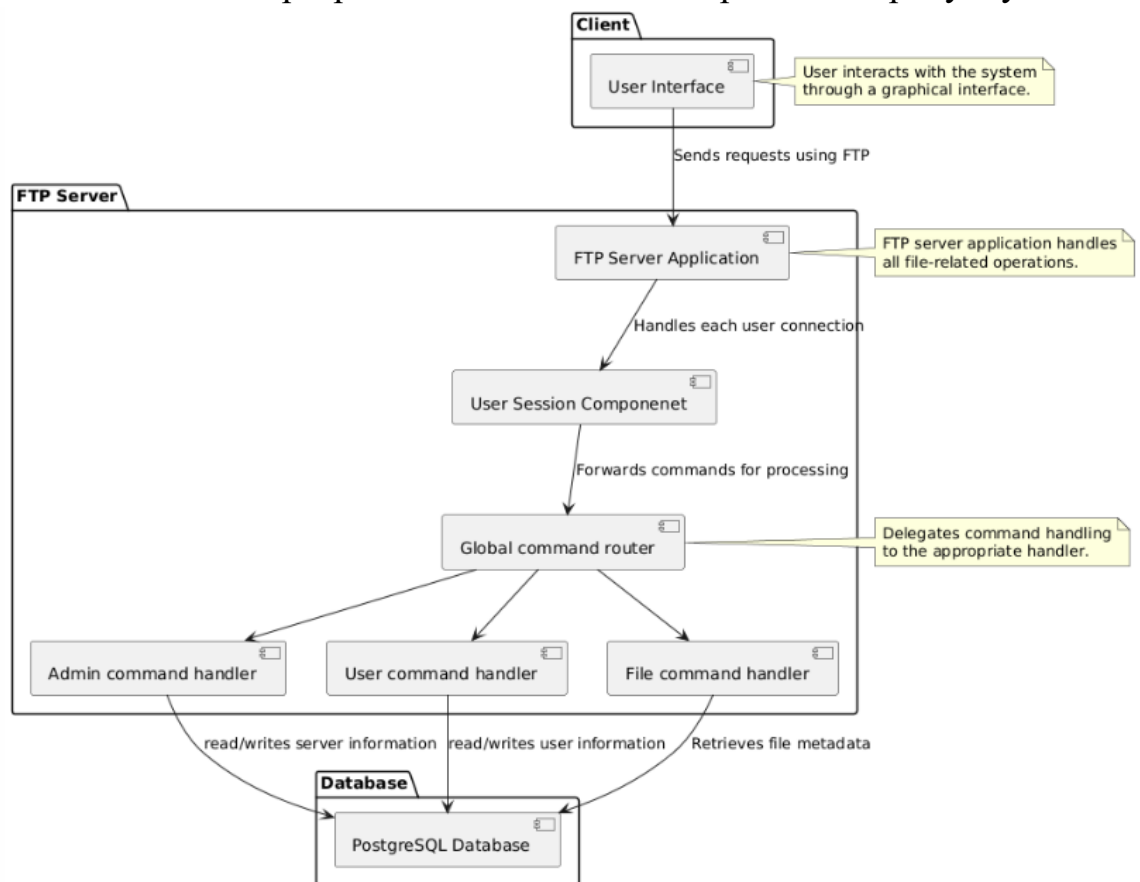


Рисунок 4 – Діаграма компонентів

Опис діаграми компонентів

Ця діаграма компонентів відображає архітектуру FTP-системи:

- Клієнтська частина:
 - Містить користувацький інтерфейс для взаємодії з системою
 - Надсилає запити до сервера через FTP

2. Серверна частина (FTP Server):

- FTP Server Application обробляє запити клієнтів
- User Session Component керує сесіями користувачів
- Global command router розподіляє команди між обробниками:
 - Admin command handler
 - User command handler
 - File command handler

3. База даних:

- PostgreSQL Database зберігає інформацію про сервер, користувачів та метадані файлів

Діаграма показує взаємодію між компонентами, потік даних та обробку команд у системі FTP-сервера, демонструючи її модульну структуру та розподіл відповідальності між компонентами.

Висновки та код

Код можна знайти за посиланням - <https://github.com/B1lok/trpz>

Висновок: під час виконання лабораторної роботи я проаналізував тему, розробив діаграму розгортання, діаграму компонентів, діаграму взаємодій та послідовностей.