



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 №2

Технології розроблення програмного забезпечення

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ
ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ

КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ

FTP-server

Виконав

студент

групи ІА–22:

Білокур Євгеній **Перевірив:**

Мягкий Михайло Юрійович

Київ 2024

Зміст

Хід роботи	3
Теоретичні відомості	3
Схема прецедентів	4
Опис прецедентів	4
База даних	6
Діаграма класів	7
Шаблон репозиторію	8
Висновки та код	8

Тема: Діаграма варіантів використання. Сценарії варіантів використання. Діаграми UML. Діаграми класів. Концептуальна модель системи

Мета: Проаналізувати тему, намалювати схему прецеденту, діаграму класів, розробити основні класи і структуру бази

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

Лабораторна робота №2 присвячена темі побудови діаграм варіантів використання, сценаріїв цих варіантів, діаграм UML, діаграм класів, та концептуальної моделі системи. Основна мета роботи полягає у вивченні та практичній реалізації ключових аспектів моделювання програмних систем, зокрема на прикладі системи FTP-сервера.

Основні теоретичні аспекти лабораторної включають:

1. Діаграми варіантів використання - це інструмент, який дозволяє візуалізувати взаємодію між користувачем (актором) та системою. Діаграми варіантів використання є важливою частиною UML, яка допомагає описати функціональні вимоги до системи на початкових етапах проектування. Вони відображають сценарії взаємодії, такі як підключення до FTP-сервера, перегляд файлів або обмеження кількості підключень.
2. Сценарії варіантів використання описують послідовність дій, що виконуються в процесі використання певного функціоналу системи. Наприклад, сценарії можуть охоплювати процеси авторизації

користувача на сервері, перегляду файлів на сервері або встановлення обмежень для кількості підключень.

3. Діаграми класів дозволяють представити структуру системи у вигляді об'єктів (класів) та їх взаємозв'язків. Кожен клас описує певну частину системи, включаючи її атрибути та методи. Ця діаграма є основою для подальшої розробки коду та бази даних.

Схема прецедентів

Побудуємо схему прецедентів. Схема зображена на рисунку 1.

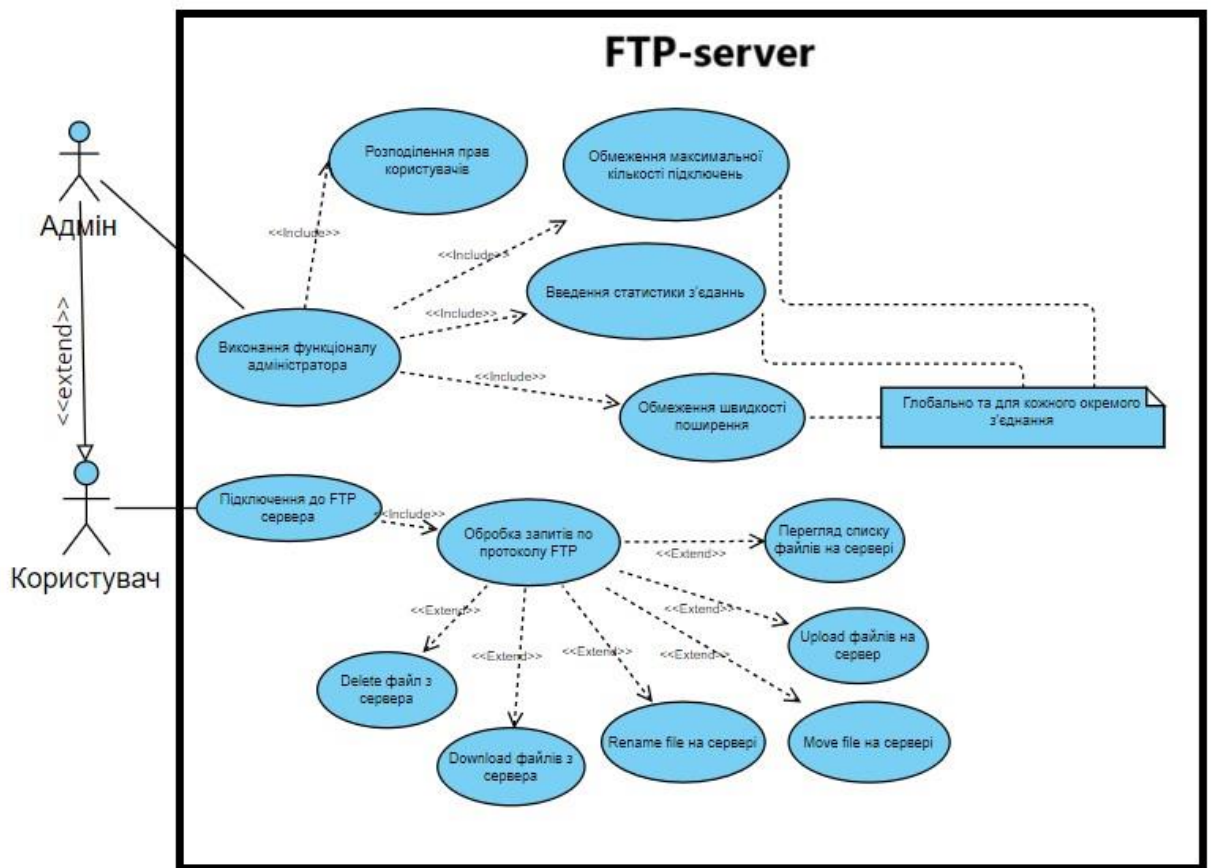


Рисунок 1. – Схема прецеденту

Опис прецедентів

Оберемо 3 прецеденти і напишемо для них сценарії використання

Сценарій використання 1: Підключення до FTP сервера

Передумови: Користувач авторизувався в системі.

Післяумови: Користувач успішно підключився до FTP сервера.

Актори: Користувач.

Опис: Цей сценарій описує процес підключення користувача до FTP сервера.

Основний хід подій:

1. Користувач обирає опцію "Підключитися до FTP сервера".
2. Система відкриває форму для введення даних підключення (IP-адреса, порт, логін, пароль).
3. Користувач вводить дані та натискає кнопку "Підключитися".
4. Система перевіряє введені дані та намагається підключитися до сервера.
5. У разі успіху система повідомляє користувача про успішне підключення.

Винятки: Якщо дані підключення некоректні або перевищено ліміт підключень, система видає повідомлення про помилку. **Примітки:** Відсутні.

Сценарій використання 2: Перегляд списку файлів на сервері

Передумови: Користувач підключений до FTP сервера.

Післяумови: Користувач переглянув список файлів на сервері.

Актори: Користувач.

Опис: Цей сценарій описує процес перегляду файлів у вибраній директорії на FTP сервері.

Основний хід подій:

1. Користувач обирає опцію "Переглянути файли на сервері".
2. Система запитує у сервера список файлів у вибраній директорії. 3.
Система відображає список файлів користувачу.

Винятки: Якщо сервер не може надати список файлів (наприклад, через помилку доступу), система повідомляє про це користувача.

Примітки: Відсутні.

Сценарій використання 3: Обмеження максимальної кількості підключень

Передумови: Адміністратор авторизувався в системі.

Післяумови: Обмеження максимальної кількості підключень встановлено.

Актори: Адміністратор.

Опис: Цей сценарій описує процес встановлення обмеження максимальної кількості підключень до FTP сервера. **Основний хід подій:**

1. Адміністратор обирає опцію " Обмеження максимальної кількості підключень ".
2. Система відкриває форму для введення максимальної кількості підключень.
3. Адміністратор вводить значення та надсилає команду.
4. Система зберігає нові налаштування та застосовує обмеження.

Винятки: немає

Примітки: Обмеження можуть застосовуватися як глобально, так і для кожного користувача окремо.

База даних

Побудуємо структуру бази даних (рисунок 2)

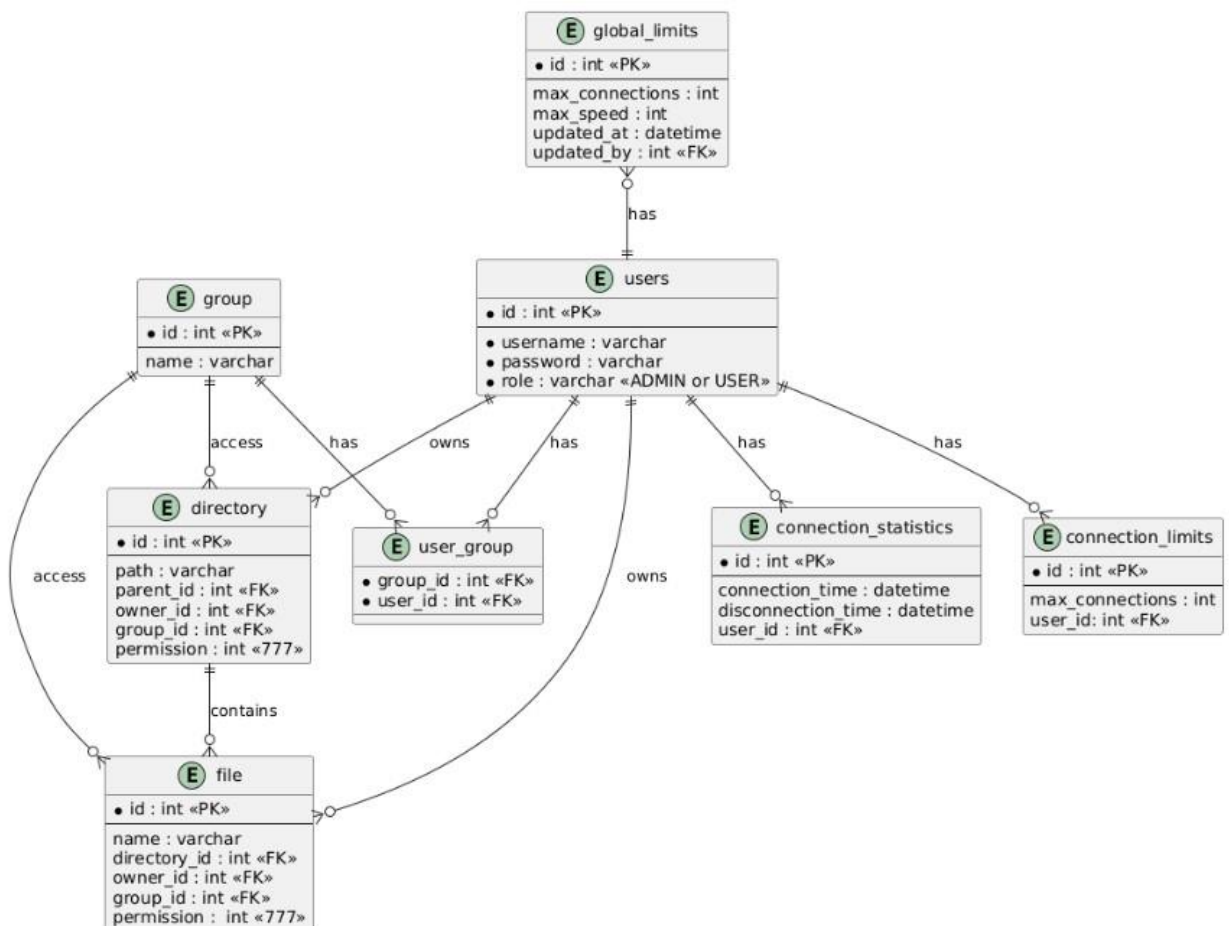


Рисунок 2. Структура бази даних

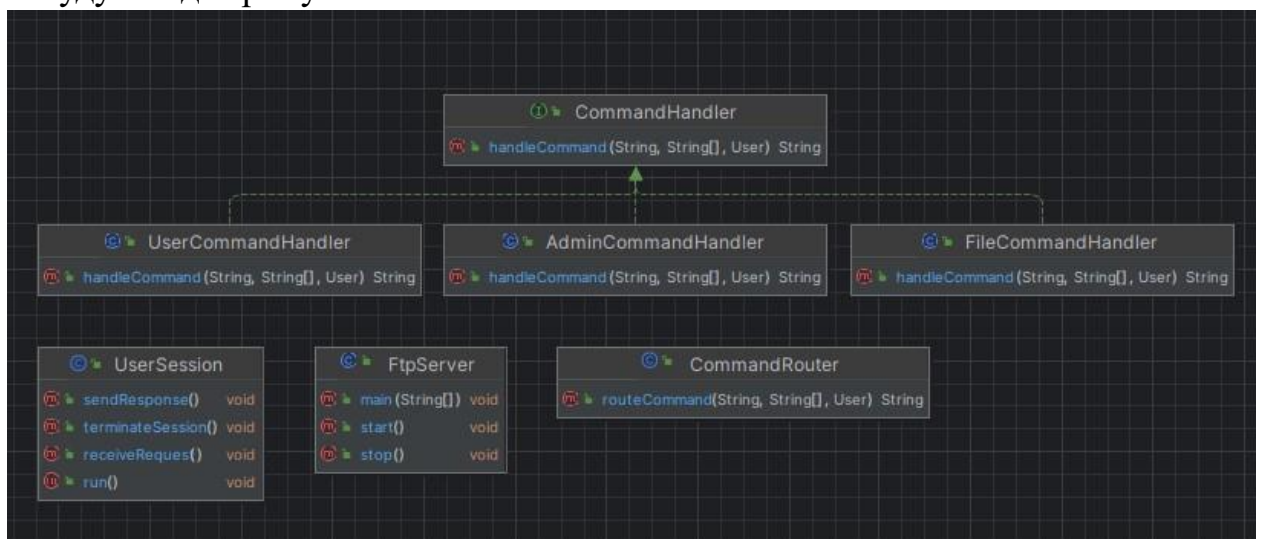
Ця діаграма представляє модель бази даних для системи FTP-сервера. Ось короткий опис основних сутностей та їх зв'язків:

1. **users**: Таблиця користувачів з полями для імені користувача, пароля і ролі (ADMIN або USER).
2. **global_limits**: Таблиця глобальних обмежень, що включає максимальну кількість підключень та швидкість передачі.
3. **connection_statistics**: Статистика підключень з інформацією про час підключення/відключення.
4. **group**: Групи користувачів.
5. **directory**: Директорії файлів з правами доступу та власником.
6. **file**: Файли з правами доступу, власником та відповідною директорією.
7. **connection_limits**: Ліміти підключень для кожного користувача.

Ключі (РК) і зовнішні ключі (ФК) забезпечують зв'язки між таблицями, такими як користувачі, групи та директорії.

Діаграма класів

Побудуємо діаграму класів



Опис діаграми класів:

CommandHandler:

- Базовий клас, що визначає метод `handleCommand()`, реалізований у класах-нащадках:
 - **UserCommandHandler**: обробляє команди звичайних користувачів.
 - **AdminCommandHandler**: обробляє команди адміністраторів.

- **FileCommandHandler**: працює з командами для роботи з файлами.

UserSession:

- Відповідає за сесії користувачів, обробляє запити та відповіді.

FtpServer:

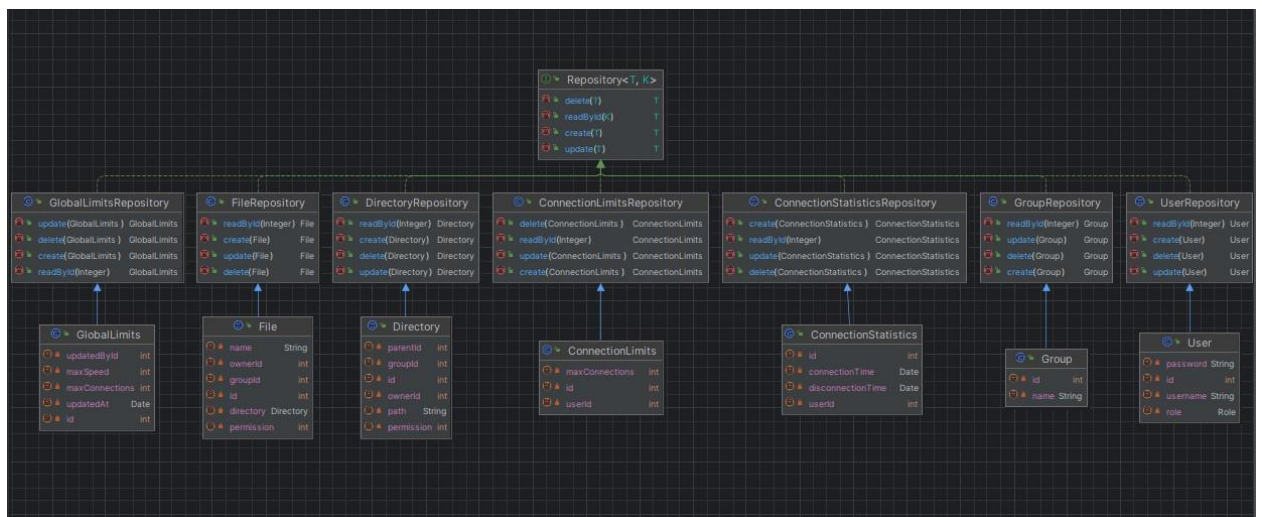
- Головний клас для запуску, зупинки сервера, прийому запитів.

CommandRouter:

- Маршрутизує команди на відповідні обробники

Шаблон репозиторію

Побудуємо шаблон репозиторію



Опис діаграми для патерну репозиторій:

Ця діаграма показує типовий дизайн репозиторію для взаємодії з сутностями бази даних FTP-сервера.

Загальний інтерфейс **Repository** визначає спільні CRUD-операції для всіх сутностей.

Специфічні репозиторії, такі як **FileRepository**, **UserRepository**, **GlobalLimitsRepository** тощо, реалізують операції для конкретних сутностей бази даних.

Кожна сутність (**File**, **Directory**, **ConnectionStats** тощо) має свій відповідний репозиторій для обробки взаємодії з базою даних.

Висновки та код

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

Висновки: під час виконання лабораторної роботи я проаналізував тему, намалював схему прецедентів, діаграму класів, розробив основні класи.