

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: «Основи проектування FTP-серверу»

**Виконав:**

студент групи ІА-34  
Сухоручкін Гліб

**Перевірив:**

Асистент кафедри ІСТ  
Мягкий М.Ю.

## **Зміст:**

1. Вступ
2. Теоретичні відомості
3. Хід роботи:
4. Діаграма варіантів використання
5. Діаграма класів системи
6. Сценарії використання FTP-серверу
7. Вихідний код класів системи
8. Структура БД
9. Висновки

## Вступ

Метою цієї роботи обрати зручну систему побудови UML-діаграм та навчитися будувати діаграми варіантів використання для системи що проєктується, розробляти сценарії варіантів використання та будувати діаграми класів FTP-серверу.

Проекторваною системою цієї лабораторної роботи є FTP-server (використовуючи 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

Мова UML є загальноцільовою мовою візуального моделювання, яка розроблена для специфікації, візуалізації, проєктування та документування компонентів програмного забезпечення, бізнес-процесів та інших систем. Мова UML є досить строгим та потужним засобом моделювання, який може бути ефективно використаний для побудови концептуальних, логічних та графічних моделей складних систем різного цільового призначення. Ця мова увібрала в себе найкращі якості та досвід методів програмної інженерії, які з успіхом використовувалися протягом останніх років при моделюванні великих та складних систем.

З погляду методології ООАП (об'єктно-орієнтованого аналізу та проєктування) досить повна модель складної системи є певною кількістю взаємопов'язаних уявлень (views), кожне з яких відображає аспект поведінки або структури системи. У цьому найзагальнішими уявленнями складної системи прийнято вважати статичне і динамічне, які у своє чергу можуть поділятися інші більш приватні.

Принцип ієрархічної побудови моделей складних систем передбачає розгляд процес побудови моделей на різних рівнях абстрагування або деталізації в рамках фіксованих уявлень.

**Рівень представлення (layer)** – спосіб організації та розгляду моделі на одному рівні абстракції, що представляє горизонтальний зріз архітектури моделі, тоді як розбиття представляє її вертикальний зріз.

При цьому вихідна або початкова модель складної системи має найбільш загальне уявлення та відноситься до концептуального рівня. Така модель, що отримала назву концептуальної, будується на початковому етапі проєктування і може не містити багатьох деталей та аспектів системи, що моделюється. Наступні моделі конкретизують концептуальну модель, доповнюючи її уявленнями логічного та фізичного рівня.

Загалом процес ООАП можна розглядати як послідовний перехід від розробки найбільш загальних моделей та уявлень концептуального рівня до більш приватних і детальних уявлень логічного та фізичного рівня. У цьому кожному етапі ООАП дані моделі послідовно доповнюються дедалі більше деталей, що дозволяє їм адекватно відбивати різні аспекти конкретної реалізації складної системи.

В рамках мови UML уявлення про модель складної системи фіксуються у вигляді спеціальних графічних конструкцій, що отримали назву діаграм.

**Діаграма (diagram)** – графічне уявлення сукупності елементів моделі у формі зв'язкового графа, вершинам і ребрам (дугам) якого приписується певна семантика. Нотація канонічних діаграм є основним засобом розробки моделей мовою UML.

У нотації мови UML визначено такі види діаграм:

- варіантів використання (use case diagram);
- класів (class diagram);
- кооперації (collaboration diagram);
- послідовності (sequence diagram);
- станів (statechart diagram);

- діяльності (activity diagram);
- компонентів (component diagram);
- розгортання (deployment diagram).

Перелічені діаграми є невід'ємною частиною графічної нотації мови UML. Понад те, процес ООАП нерозривно пов'язаний з процесом побудови цих діаграм. При цьому сукупність побудованих таким чином діаграм є самодостатньою в тому сенсі, що в них міститься вся інформація, яка потрібна для реалізації проєкту складної системи.

Кожна з цих діаграм деталізує та конкретизує різні уявлення про модель складної системи у термінах мови UML. При цьому діаграма варіантів використання являє собою найбільш загальну концептуальну модель складної системи, яка є вихідною для побудови інших діаграм. Діаграма класів, за своєю суттю, логічна модель, що відбиває статичні аспекти структурної побудови складної системи.

Діаграми кооперації та послідовностей є різновидами логічної моделі, які відображають динамічні аспекти функціонування складної системи. Діаграми станів та діяльності призначені для моделювання поведінки системи. І, нарешті, діаграми компонентів і розгортання служать уявлення фізичних компонентів складної системи і тому представляють її фізичну модель.

## Хід роботи

### Діаграма варіантів використання

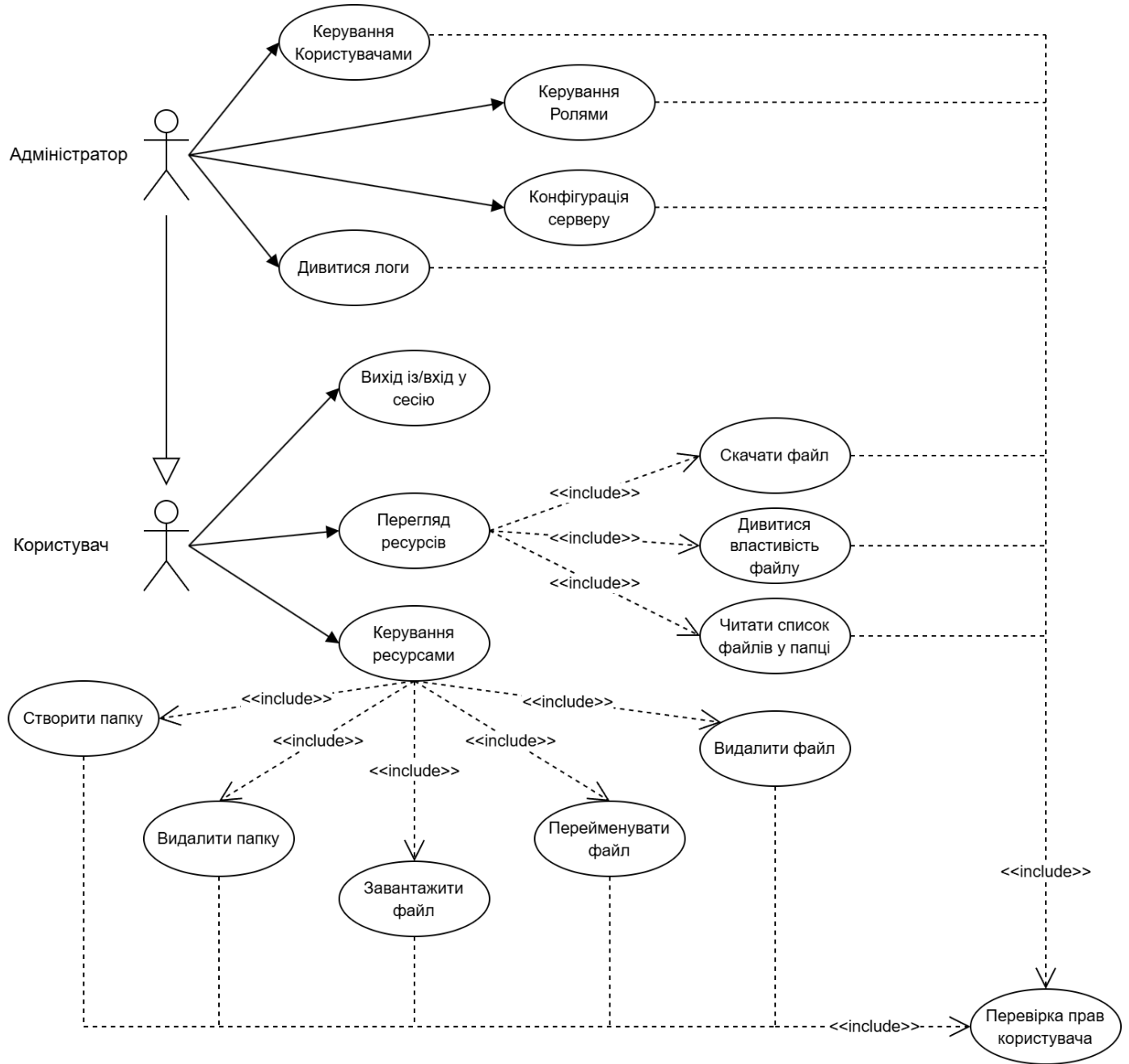


Рис. 1 – Діаграма варіантів використання FTP-серверу

## Діаграма класів системи

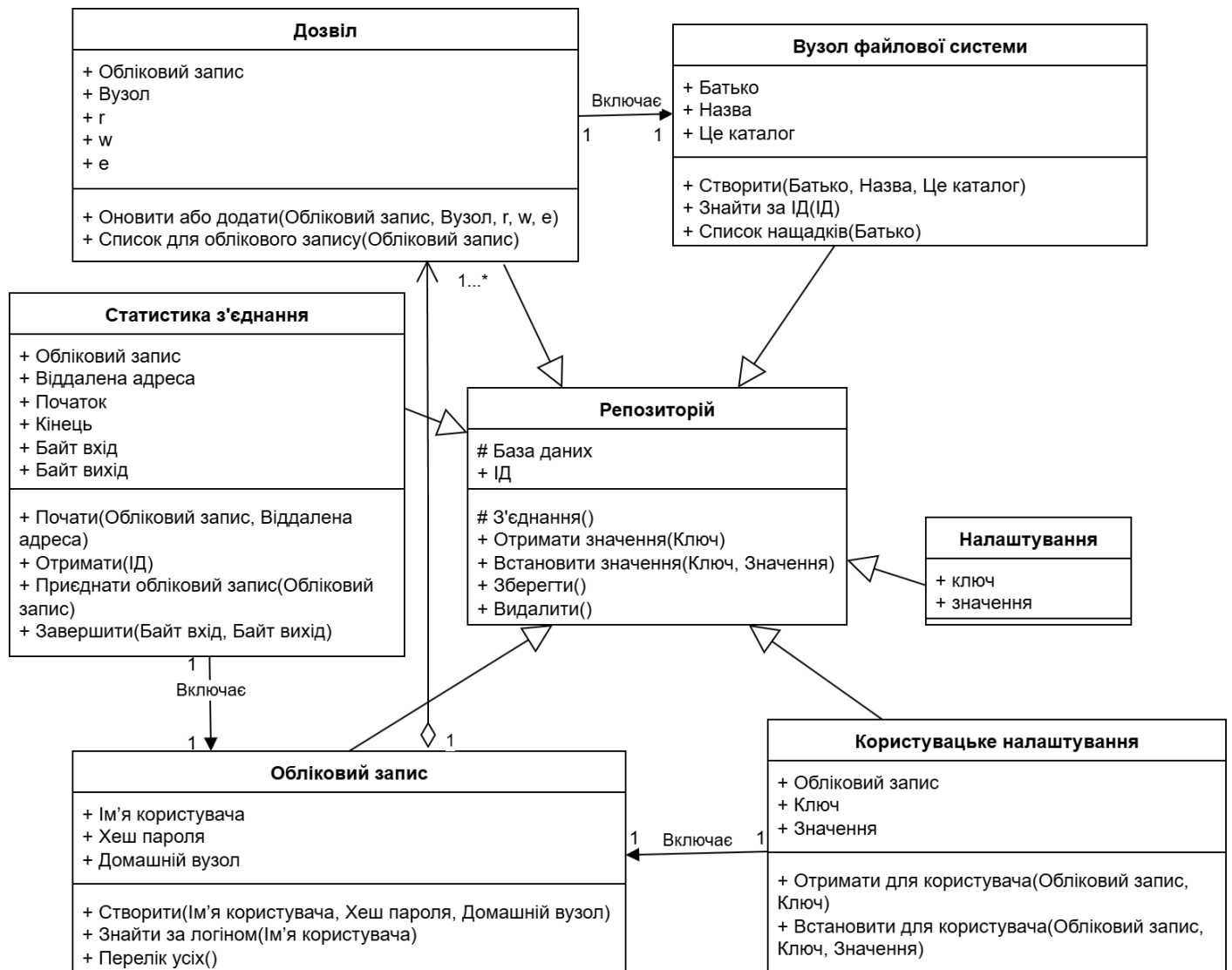


Рис. 2 - Діаграма класів системи

## Сценарій використання FTP-серверу

### 1. Авторизація користувача (USER/PASS)

#### Передумови

- Встановлено керуюче (CONTROL) з'єднання з сервером (порт 21), користувач не автентифікований.
- Сервер надіслав привітання (код відповіді FTP серверу є 220).
- Обліковий запис існує у системі керування користувачами; сервер доступний.

#### Постумови

- У разі успіху сесію автентифіковано, встановлено робочий каталог користувача
- У разі помилки стан сесії лишається «неавтентифіковано», доступ до команди DATA/FILE заборонено.

#### Взаємодіючі сторони

- Основні: Користувач (FTP-клієнт), FTP-сервер.
- Вторинні: Сховище облікових даних / СУБД користувачів.

#### Короткий опис

Користувач вводить логін і пароль; сервер перевіряє облікові дані та переводить сесію в автентифікований стан або відхиляє запит.

#### Основний перебіг подій

1. Клієнт ініціює з'єднання, FTP сервер повертає код 220.
2. Клієнт надсилає USER <login>.
3. Сервер відповідає 331 (необхідний пароль) або 530 (доступ заборонено).
4. Клієнт надсилає PASS <secret>.
5. Сервер перевіряє облікові дані та повертає 230 (успіх) і встановлює робочий каталог користувача.

#### Винятки

**Виняток №3:** Внутрішня помилка сервера чи СУБД.

Якщо під час перевірки облікових даних або обробки запиту стається локальна помилка (збій підсистеми, БД), система відповідає кодом 451 («Requested action aborted: local error in processing»). Користувач може повторити спробу пізніше або повідомити адміністратора.

#### Примітки

- За потреби може застосовуватись TLS (FTPS) – поза межами цього сценарію.
- Після автентифікації застосовуються політики доступу (ACL) у наступних сценаріях.



## 2. Перегляд вмісту каталогу (LIST/NLST)

### Передумови

- Користувач автентифікований.
- Робочий каталог існує, вказаний шлях (якщо задано) коректний.
- Вибрано режим даних: активний (PORT/EPRT) або пасивний (PASV/EPSV); порт/канал доступний.

### Постумови

- Користувач отримує перелік файлів/папок (формат LIST або імена NLST).
- У разі помилки стан файлової системи не змінюється.

### Взаємодіючі сторони

- Основні: Користувач, FTP-сервер.
- Вторинні: Файлова система (локальна/віддалена).

### Короткий опис

Користувач відкриває DATA-канал і запитує список вмісту каталогу.

### Основний перебіг подій

1. Клієнт уже під'єднаний до CONTROL-каналу (порт 21) і хоче переглянути вміст каталогу.
2. Клієнт надсилає EPSV (або PASV).
3. Сервер відкриває слухаючий порт для DATA-каналу й відповідає 229 (або 227) із параметрами.
4. Клієнт встановлює TCP-підключення до вказаного серверами порту та надсилає по CONTROL-каналу команду LIST [path] (або NLST [path]).
5. Сервер перевіряє шлях і права доступу та відповідає 150 («Opening data connection...»).
6. Сервер відсилає вміст каталогу через DATA-канал (для LIST — детальні рядки; для NLST — лише імена).
7. Після завершення сервер закриває DATA-канал і повертає по CONTROL-каналу 226 («Transfer complete»).

### Винятки

**Виняток №1:** Шлях не існує або недостатні права.

Якщо на етапі перевірки шляху/доступу виявиться, що каталогу чи файла не існує або бракує прав, система відповідає кодом 550 («No such file or directory» / «Permission denied»). Користувач може скоригувати шлях або права доступу і повторити запит з відповідного кроку

### Примітки

- LIST повертає деталі (права, розмір, дату), NLST — лише імена.
- Формат рядків LIST залежить від сервера; клієнт повинен бути готовим до варіацій.

### **3. Завантаження файла на сервер (STOR)**

#### **Передумови**

- Користувач автентифікований, має право запису в цільовий каталог.
- Достатньо квоти/вільного місця.
- Встановлено DATA-канал (PASV/EP SV або PORT/EP RT).

#### **Постумови**

- Файл збережено у вказаному розташуванні; метадані (розмір/час) оновлено.
- У разі помилки файл не створено або частковий результат видалено/не публікується.

#### **Взаємодіючі сторони**

- Основні: Користувач, FTP-сервер.
- Вторинні: Файлова система, модуль обліку квоти (якщо є).

#### **Короткий опис**

Користувач передає файл на сервер через DATA-канал командою STOR.

#### **Основний перебіг подій**

1. Клієнт встановлює DATA-канал (PASV/EP SV або PORT/EP RT).
2. Клієнт надсилає STOR <remote-path>.
3. Сервер відповідає 150 – «Opening data connection for STOR».
4. Клієнт передає вміст файла; сервер приймає та записує.
5. Після завершення сервер закриває DATA-канал і повертає 226 – «Transfer complete».

#### **Винятки**

**Виняток №1:** Недостатні права або некоректна назва файла.

Якщо під час виконання Основного перебігу подій виявиться, що шлях недоступний або назва файла заборонена політиками/ФС, система відповідає кодом 550 («Permission denied») або 553 («File name not allowed»). Користувач може змінити шлях/назву та повернутися до відповідного кроку або завершити варіант використання.

**Виняток №2:** Перевищено квоту або бракує місця.

Якщо під час запису файла виявиться, що перевищено виділену квоту або на носії немає вільного простору, система відповідає кодом 552 («Requested file action aborted; exceeded storage allocation»). Користувач може звільнити місце/збільшити квоту та повторити спробу.

#### **Примітки**

- Режим передачі (ASCII/бінарний) задається окремо (наприклад, TYPE I для бінарних даних).

**Вихідний код класів системи**

<https://github.com/GlebTiKsTRPZ/lab2>

# Структура БД

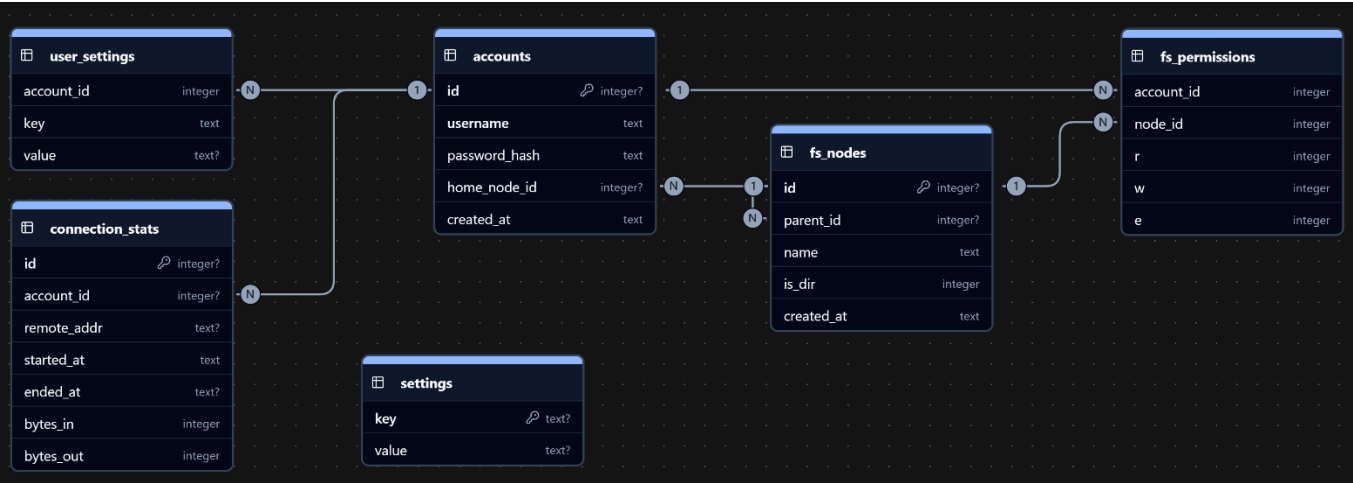


Рис. 3 – Структура базы данных.

## **Висновок**

У роботі спроектовано та описано UML діаграми для FTP-сервера: діаграму варіантів використання (use-case diagram), діаграму класів і сценарії (USER/PASS, LIST/NLIST, STOR). Побудовано структуру бази даних і застосовано шаблон Repository для класів даних. Підготовлено базову версію на Java/SQLite (FtpServer, Database, Repository, моделі), що узгоджується з проєктною моделлю.