WIPRO NGA Program – 25SUB4508_LSP

Capstone Project Presentation – 13th Feb & 14th Feb 2026

Project Title Here - Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Presented by -

www.rpsconsulting.in

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Introduction :

Managing large volumes of files and directories manually across systems is time-consuming and error-prone. Identifying specific content within distributed storage environments can be difficult and inefficient without automation.

The **Distributed Content Discovery & Inspection Utility (DCDIU)** allows users to recursively traverse directories, search for specific keywords within files, and inspect file contents efficiently from a centralized interface.

It uses a C++ POSIX-based backend with a modular client-server architecture to provide reliable directory scanning, deep content analysis, structured logging, and robust error handling across Unix-like systems.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Problem Statement :

In modern computing environments, organizations and system administrators face several challenges, such as:

-- Difficulty in locating specific files within deeply nested directory structures.
-- Time-consuming manual inspection of large volumes of files to find relevant content.
-- Inefficient keyword searching across distributed or decentralized storage systems.
-- Lack of centralized tools to perform recursive directory traversal and deep content analysis.
-- Limited visibility into file contents without opening each file individually.
-- A robust, automated utility is essential to ensure efficient, accurate, and scalable content discovery across systems.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Objective :

➔ Primary Objectives

-- Develop a secure command-line utility for recursive directory traversal and content discovery.

-- Enable searching and inspection of files using keyword-based deep scanning.

-- Provide a centralized system to manage file discovery and inspection efficiently.

➔ Secondary Objectives

-- Ensure portability across Unix-like and POSIX-compliant systems.

-- Maintain structured logging for monitoring and debugging.

-- Support multi-client handling using a modular client-server architecture.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Key Features:

- Client–Server Architecture

Clear separation between request initiation (client) and processing (server).

- Remote Directory Traversal

Recursive traversal of directories from a configurable base path.

- Deep Content Discovery

Scans file contents for specific strings or patterns across all traversed files.

- File Inspection Mode

Allows targeted inspection and retrieval of file content using absolute paths.

- Structured Response Handling

Server sends processed results back to the client in a readable format.

- Robust Error Handling & Logging

Centralized exception handling and logging for reliability.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Requirement Tag	Requirement Description
Functional Requirements	
DCDIU_FR_01	The utility shall be capable of performing recursive directory traversals on a remote target server, with an optional configurable base path.
DCDIU_FR_02	DCDIU must support deep-content discovery by scanning for specific strings, phrases, or patterns across all files in the target directory and returning a comprehensive match list.
DCDIU_FR_03	The system shall allow for targeted retrieval of specific files when an absolute path is provided, enabling the user to view the raw data stream.
DCDIU_FR_04	The utility must provide a 'Content Inspection' mode, allowing users to select and display the full data contents of any file identified during the discovery phase.
DCDIU_FR_05	The framework shall implement an exception handling module to display descriptive system messages when traversal or discovery operations fail.
DCDIU_FR_06	A command-line driven interface (CLI) shall be provided, featuring a structured menu for all supported discovery and inspection operations.
DCDIU_FR_07	The system must integrate a diagnostic logging engine with four standardized severity levels: FATAL, INFO, WARNING, and DEBUG.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Non-Functional Requirements	Requirement Description
DCDIU_NF_01	Portability: The utility shall be compatible with all standard Unix-like and POSIX-compliant Operating Systems.
DCDIU_NF_02	Modularity: The system architecture shall utilize well-defined interfaces to ensure scalability and the addition of future data-parsing modules.

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Functional Requirements Mapping:

Requirement ID	Description	Implementation Module
DCDIU_FR_01	Client–Server Connection	TCP Socket
DCDIU_FR_02	Recursive Directory Traversal	DirectoryTraverser
DCDIU_FR_03	Deep Content Scanning	ContentScanner
DCDIU_FR_04	File Inspection by Absolute Path	FileInspector
DCDIU_FR_05	Request Parsing & Delegation	ClientHandler / Server
DCDIU_FR_06	Structured Response Delivery	Server → Client Communication
DCDIU_FR_07	Error Handling	ExceptionHandler
DCDIU_FR_08	Logging & Observability	Logger

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Function Module :

1. Directory Traversal Module:
-- Recursive scanning of directories and subdirectories
-- Collect and manage file paths dynamically

2. Content Scanning Module:
-- Search for specific keywords within files
-- Perform deep content discovery across all scanned files

3. File Inspection Module:
-- Display full file content on request
-- Enable detailed inspection of selected files

4. Logging & Exception Module:
-- Maintain structured logs (INFO, WARNING, DEBUG, FATAL)
-- Handle system errors gracefully using ExceptionHandler

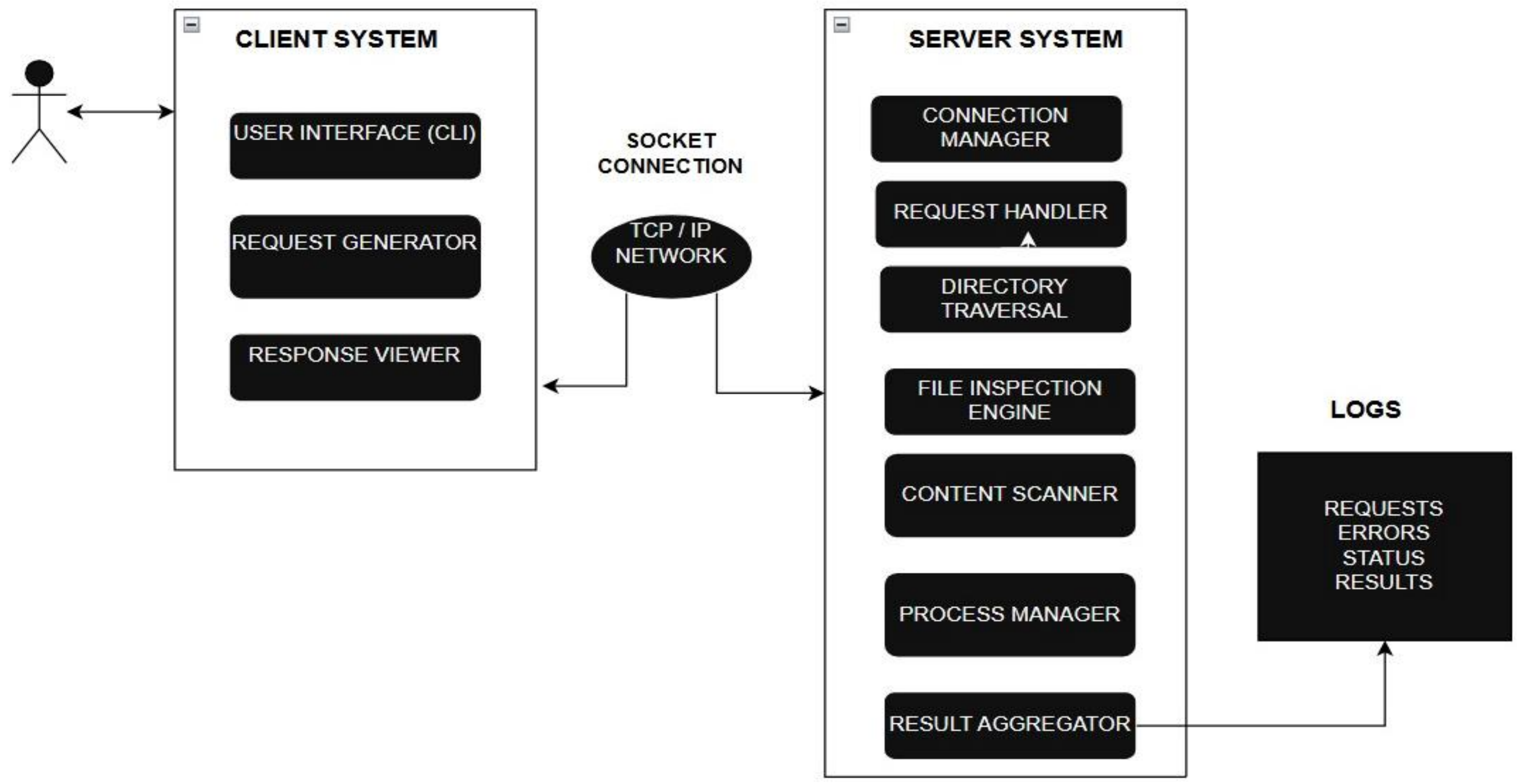
5. Multi-Client Handling Module:
-- Handle multiple client connections concurrently
-- Manage independent client requests using threading

6. Communication Module:
-- Establish client-server socket connection
-- Exchange directory path and keyword data securely

7. System Monitoring Module:
-- Maintain detailed activity logs for auditing
-- Monitor execution status and error reporting

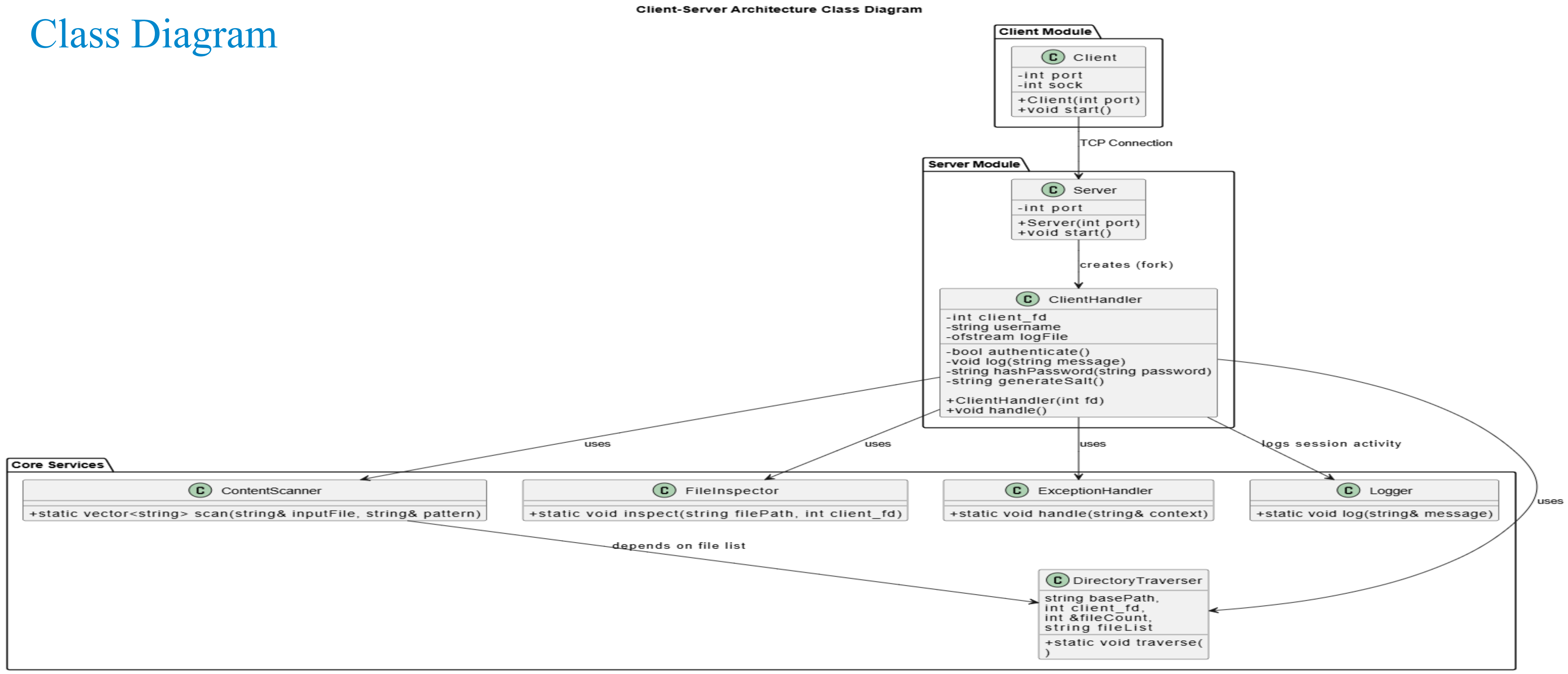
Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

System Architecture:



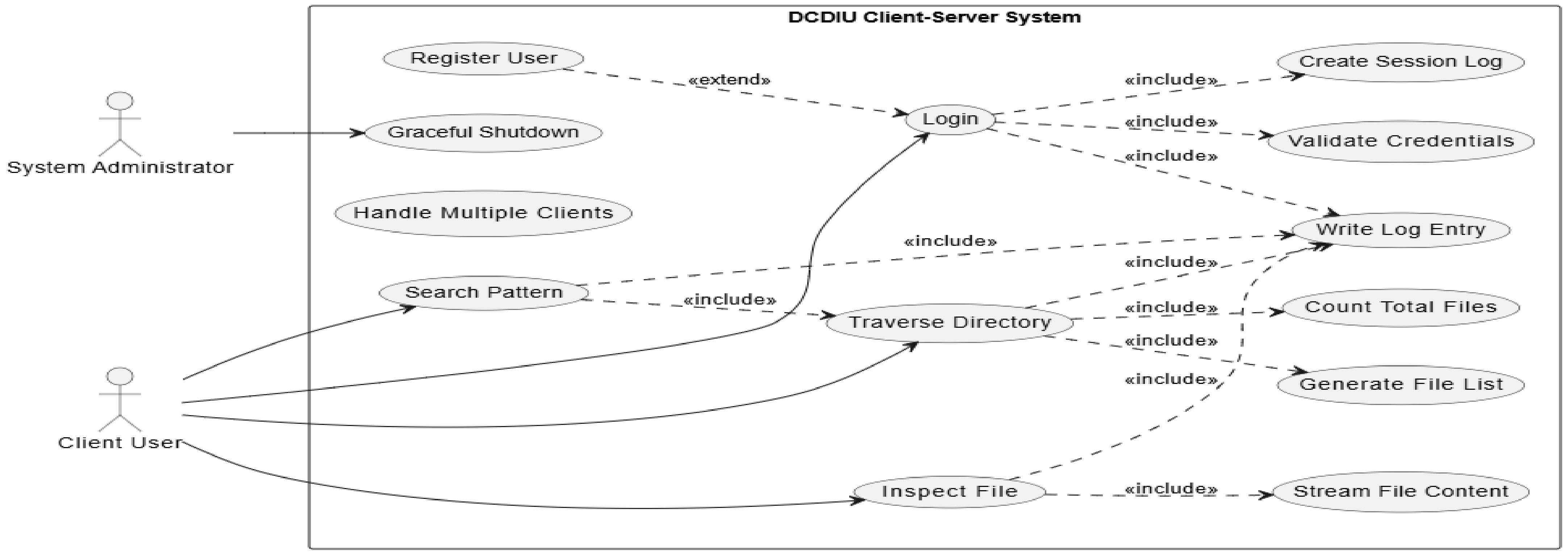
Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Class Diagram



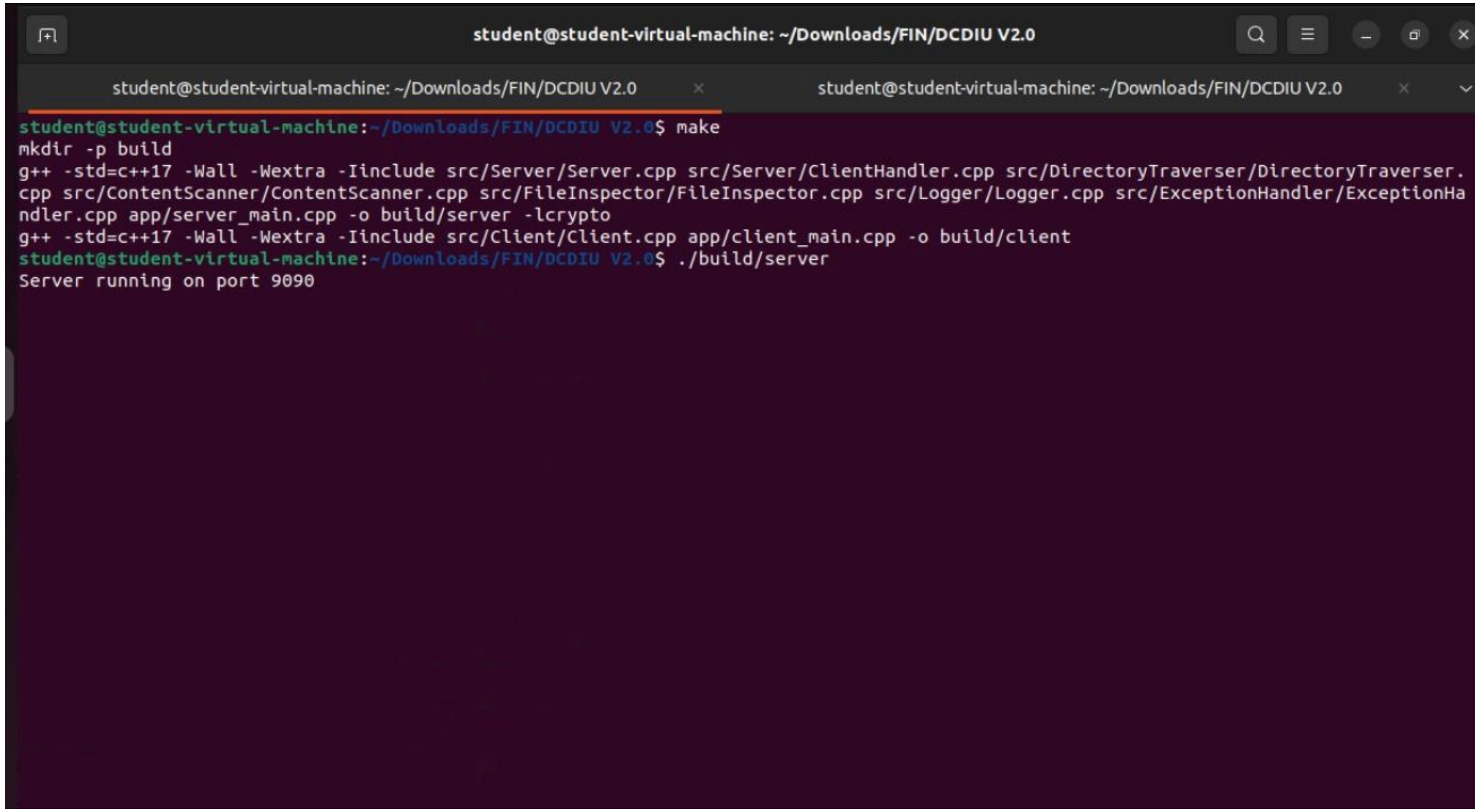
Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Use-Case Diagram:



Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

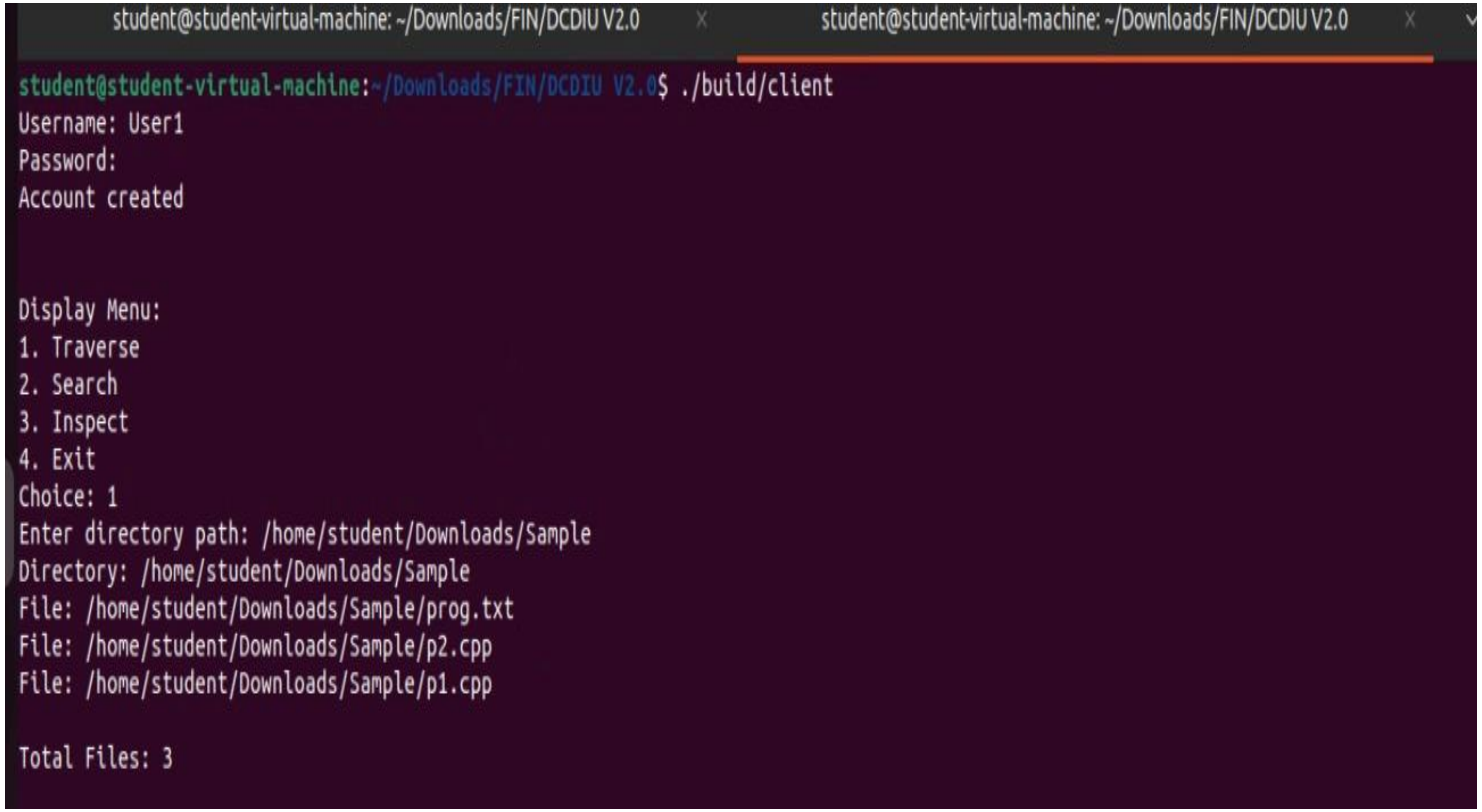
Output: Connecting to the server


student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0
student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0
student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0

```
student@student-virtual-machine:~/Downloads/FIN/DCDIU V2.0$ make
mkdir -p build
g++ -std=c++17 -Wall -Wextra -Iinclude src/Server/Server.cpp src/Server/ClientHandler.cpp src/DirectoryTraverser/DirectoryTraverser.
cpp src/ContentScanner/ContentScanner.cpp src/FileInspector/FileInspector.cpp src/Logger/Logger.cpp src/ExceptionHandler/ExceptionHa
ndler.cpp app/server_main.cpp -o build/server -lcrypto
g++ -std=c++17 -Wall -Wextra -Iinclude src/Client/Client.cpp app/client_main.cpp -o build/client
student@student-virtual-machine:~/Downloads/FIN/DCDIU V2.0$ ./build/server
Server running on port 9090
```

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Output: Choice 1)Traverse



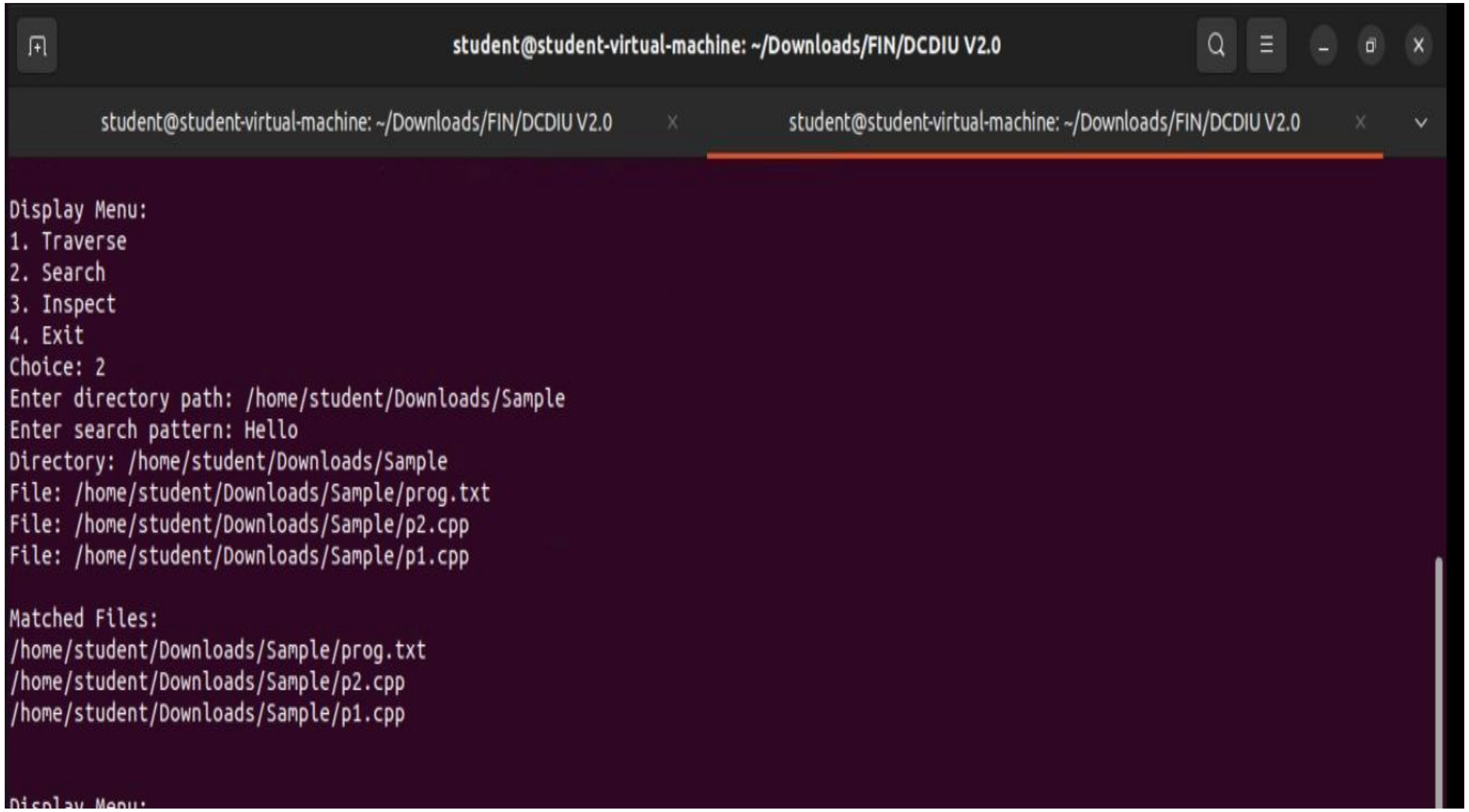
```
student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0        student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0
student@student-virtual-machine:~/Downloads/FIN/DCDIU V2.0$ ./build/client
Username: User1
Password:
Account created


Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 1
Enter directory path: /home/student/Downloads/Sample
Directory: /home/student/Downloads/Sample
File: /home/student/Downloads/Sample/prog.txt
File: /home/student/Downloads/Sample/p2.cpp
File: /home/student/Downloads/Sample/p1.cpp

Total Files: 3
```

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Output: Choice 2) Search


student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0
student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0
student@student-virtual-machine: ~/Downloads/FIN/DCDIU V2.0

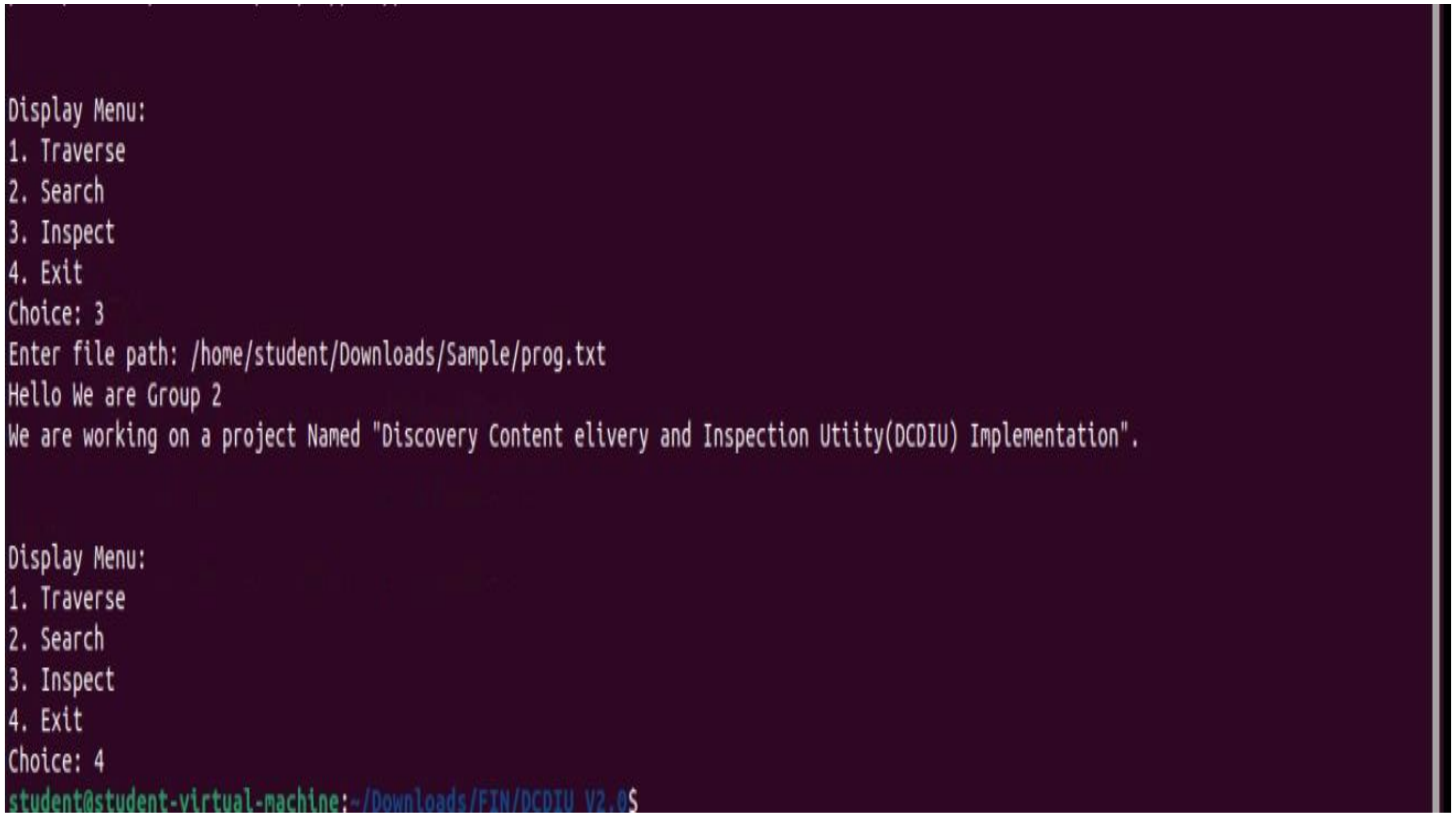
```
Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 2
Enter directory path: /home/student/Downloads/Sample
Enter search pattern: Hello
Directory: /home/student/Downloads/Sample
File: /home/student/Downloads/Sample/prog.txt
File: /home/student/Downloads/Sample/p2.cpp
File: /home/student/Downloads/Sample/p1.cpp

Matched Files:
/home/student/Downloads/Sample/prog.txt
/home/student/Downloads/Sample/p2.cpp
/home/student/Downloads/Sample/p1.cpp


Display Menu:
```

Distributed Content Discovery & Inspection Utility (DCDIU) Implementation

Output: Choice 3) Inspect



```
Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 3
Enter file path: /home/student/Downloads/Sample/prog.txt
Hello We are Group 2
We are working on a project Named "Discovery Content elivery and Inspection Utiity(DCDIU) Implementation".


Display Menu:
1. Traverse
2. Search
3. Inspect
4. Exit
Choice: 4
student@student-virtual-machine:~/Downloads/FIN/DCDIU_V2.0$
```