

CIS 457 – Winter 2022

CIS457: Project 2 – Building GV-NAP File Sharing System

Date Posted: 3/14/22

Date Assigned: 3/15/22

Due by: 3/29/22 by 1:00PM EST; late submission policy (20%/day for up-to 5 days including weekend/holidays)

Point Value: 100 pts

Project Demo and Work Protocol:

- For this project, 4 - 5 students will team up to work on this project as a group. The groups are already pre-formed as per the info posted in this spreadsheet
https://docs.google.com/spreadsheets/d/1hvASVAsRFT22SzAhDwJJ8pEWgs7JMc_mc1QDJbnzRz4/edit#gid=0
- Students need to submit the following items for grading the project; otherwise no grade will be given to the submitted work:
 1. A 5-minutes video to demo the project work. Students can take parts of the demo or only one student can demo the entire project work. **In the video, please state (i) the completion percentage of the project and (ii) what functions worked and what functions do not work.** Group will demo the project live during our lab meeting on **Tue 3/29**
 2. A copy of your project code with a written report showing screen capture demonstrating how **each service** provided by the project is working. The report should include the basic logic of the program, how you implemented each service, what problems you have encountered and how you solved them. The programs should be submitted in an electronic version that contains all of your source codes.
All materials listed in items (1) and (2) must be uploaded to BB using the link "SubmitYourProject02Here"
 3. **The team members are responsible for:**
 - a. Writing, debugging and making the project work as required using any programming language that the team deem appropriate.
 - b. Sample Code is provided as part of the project materials to just show you the code's overall structure. You can use it as a guide to build your own code or simply ignore it and design your own code.
 - c. Dividing the project work equally among the team members as well as contributing fairly to the overall project deliverables.
 - d. If this is not the case, you may consider leaving the group and form another group as well **inform** the instructor about making this change. This action has to be taken place within the first few days after assigning the project. Please, Do NOT wait until the last minute to make this decision.
 4. Detailed self and peer evaluation may be conducted randomly to get more info about how the project work was going on during the course of the project.

Introduction and Project Requirements

In the first project you implemented a file transfer protocol that supports multiple client connections. In this project, your file transfer protocol will form the basis of a file sharing system that allows users to access a distributed data storage system based on simple keyword search. The specification of this project relies on the principles of using the *centralized directory indexing service* to support a P2P file transfer. The system consists of two parts:

- The first part of the system is the **centralized server**, which provides a **search facility** that can be used to perform simple keyword searches.
- The result of the search is the **location of the remote resource**.
- The centralized server also **tracks current users and the files shared** from their site.
- The second part is the host system, which can query the centralized server for files using keywords.
- The host system also has an FTP client and server. The ftp client allows a user to access files stored at remote user locations. The ftp server is responsible for providing file transfer services requested by a remote client.

The project assignment suggests a method of implementation such as described below. However, you are **free** to implement it in any way you like provided that you deliver the same required functionality as outlined above and in the following sections.

Implementation Approach

The project consists of two components such as given in Figure 1.

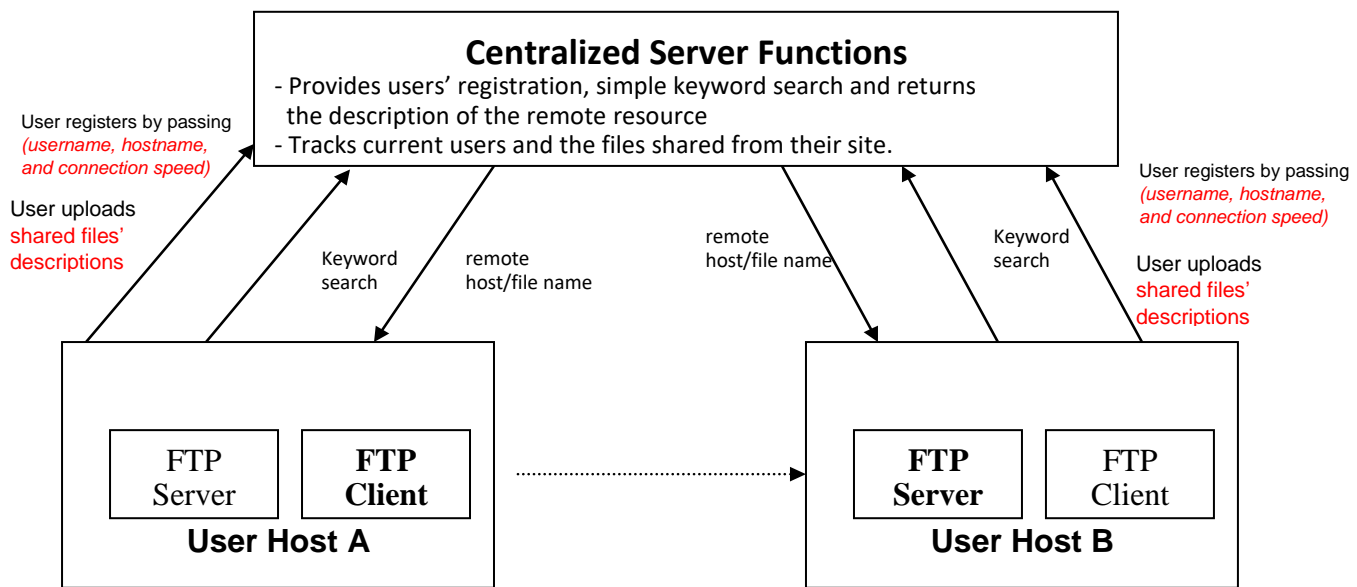


Figure 1: Architecture of the GV-NAP file sharing system.

1. Host Implementation

The host is a GUI application. The user must specify the **hostname and port** of the server in order to connect to it. Once a connection is initiated with the server, the host sends the **three** pieces of information (**username, hostname, and connection speed**). After that, the host uploads the **shared files' descriptions** (for example a file called "*filelist.txt*" in the current working directory). In the "*filelist.txt*", the descriptions include the **file name** and the **description of each shared file such as** (*file-1 → slow-music, file-2 → jazz-music, ..etc*).

The shared files are also located in the current working directory. The descriptions for these files are used for the keyword searches. After user and file information are successfully registered, the client is now able to do keyword

searches. Once the search is executed, the remote host and file name of all matching files of the keyword are given back by the server.

After getting the results, the client can contact the remote ftp server to retrieve the file. The ftp client can only be used to “get” a file and **not** to “put” a file. When the client wants to disconnect from the NAP server, client can click “Disconnect”, which sends out a **QUIT** command to the server.

2. Centralized Server Implementation

The server tracks current users and the files shared from their site. Once a connection is initiated from a host, it stores *the username, hostname and connection speed*, in a “**users**” table/list. After this, it acknowledges to the host that it has received this information and then waits for the *shared files’ descriptions* to be uploaded. Once it receives the *shared files’ descriptions (file name and the description of each shared file)*, it’ll be stored in a “**files**” table/list.

It also provides simple keyword search and returns the description of the remote resource. On receipt of a keyword search, the server performs the search in the “**files**” table/list and returns the **resource location** of the remote files. The resource location includes the **remote hostname, port number, remote file name, and the connection speed** (the host name and connection speed are retrieved from the corresponding entry in the “users” table/list).

The server allows multiple clients to register and upload their descriptions at the same time. It also allows multiple hosts to query for files at the same time. It tracks the availability of the remote resources too. If a host “unregistered” from the system, the associated file descriptions and user information are deleted from the system. A screen capture of the required server side of the application is given in Figure 2.

Project grading (100Pts):

- 40Pts: Multi users’ registrations
- 30Pts: Keyword search and obtaining a match
- 30Pts: P2P file download
- Please, note that 10Pts will be deducted if a single thread application is delivered.

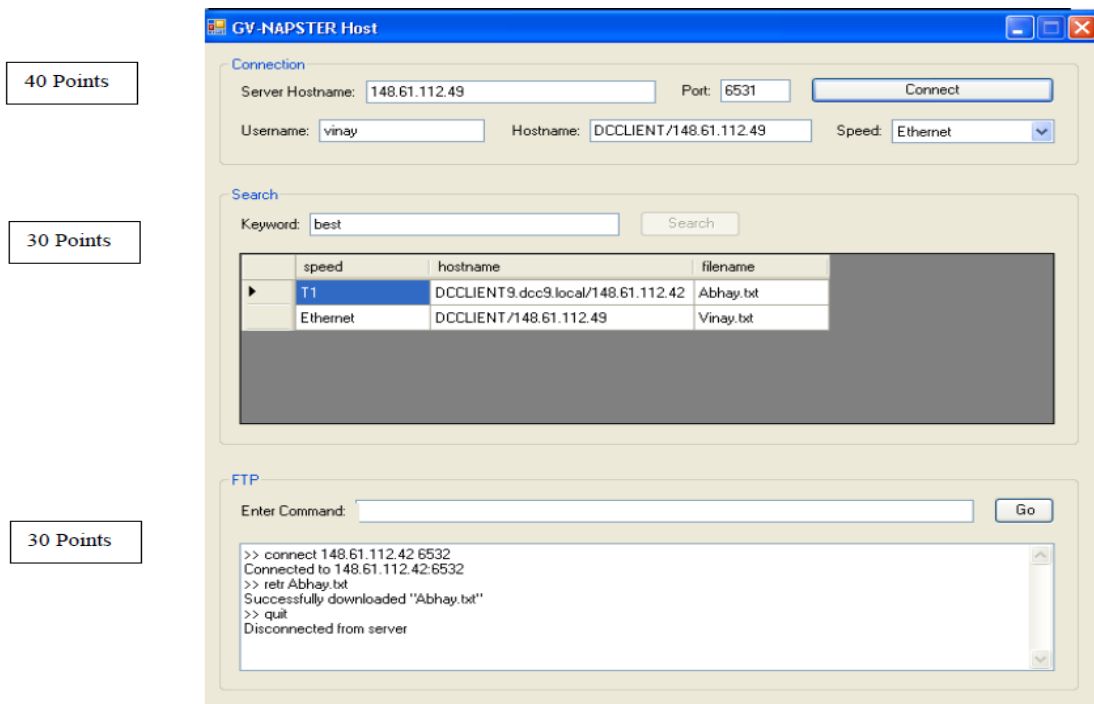


Figure 2: Suggested Sample GUI for the Application

Design Requirements Checklist and Expectations:

1. Your FTP server at the remote sites should support concurrency using multi-threading to allow multiple clients to connect at the same time.
2. Your centralized server should allow multiple users to upload their descriptions at the same time, i.e. you need a concurrent centralized server.
3. When a user registers with the centralized server for the first time, the user may be prompted for the capacity of the link to the Internet (e.g. modem, Ethernet, T1, T3 etc).
4. The centralized server should be able to:
 - ✦ Perform keyword searches on the uploaded files' descriptions based on a user supplied keyword.
 - ✦ Track the availability of the remote resources. For instance, if a user "unregistered" from the system, the associated file descriptions should be deleted at the centralized server.
5. Your code can be implemented in any language. You may use any file transfer protocol such as ftp or http if you wish, ...etc.
6. Your project must be demonstrated in order for me to grade it. Until this will happen, your project is on a late submission status for up to 5 days (max) with a late submission policy of (20%/day for up-to 5 days including weekend/holidays); (even if you emailed me or uploaded to BB your project materials)