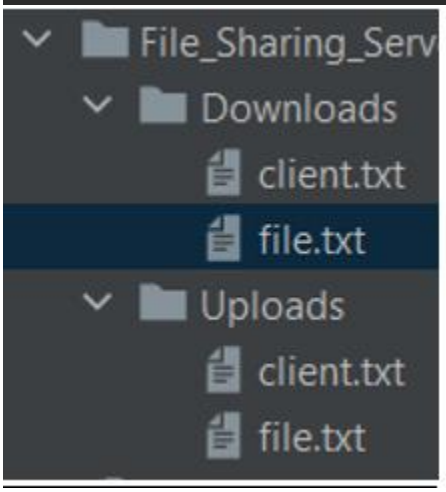
```
Welcome To GV-NAP File Sharing System
Menu
(1) Register
(2) Search
(3) List all files
(4) Download
(5) Exit server
1
Enter a keyword: clie
Enter file name: client.txt
File Registered Successfully.
Client is ready to share the files
Menu
(1) Register
(2) Search
(3) List all files
(4) Download
(5) Exit server
4
Enter the peer id #: 516
Enter file name: file.txt

File is downloaded successfully.
```

```
(1) Register
(2) Search
(3) List all files
(4) Download
(5) Exit server
1
Enter a keyword: chase
Enter file name: file.txt
File Registered Successfully.
Client is ready to share the files
Menu
(1) Register
(2) Search
(3) List all files
(4) Download
(5) Exit server
3
Peer ID    File Name    Date Added
516    file.txt      2022-04-15 12:49:40.636461
413    client.txt      2022-04-15 12:49:32.229538
---------------------------------
Menu
(1) Register
(2) Search
(3) List all files
(4) Download
(5) Exit server
Got connection from 127.0.0.1  port at 64835
File Sent Successfully
```



Report

Server.py is the Centralized server which has functions register, search, list, and quit. Ftp_Server.py is the ftp server which provides file transfer services by a remote client. Client.py is the host system which has function register, search, list, download, and quit.

The program first starts with a client connecting to server.py, the centralized server, at localhost on port 18964. Clients then registers files with the centralized server. The clients can search for files registered by themselves and other clients on the centralized server. The centralized server will respond which client owns the file requested and send a port number to transfer the file over. Clients then downloads files through ftp_Server.py the ftp server that provides file transfer services.

Things learned while creating this program where the functions semaphore and pickling. Semaphore is for thread synchronization and pickling is for serializing and deserializing messages sent between client and server. One thing that can be expanded upon is in client.py the variables nameList and fileList can be expanded to larger list sizes, but if I added 0s the list function would be large and filled with blanks.

To understand the function register in client.py, the client connects to server.py over localhost on port 18964. Then the client sends the serialized data over port 18964 and sends the command 1 which tells the server to register the file. Then the client receives the status and decodes the messages send from the server, it prints the status, then closes the connection with the server.

The function search in client.py starts by connecting to server.py. Then the client enters the keyword at which files in the centralized server, server.py, are registered. The function search then returns “file found”, or “file not found” if the file by that keyword is registered on the centralized server.

The function download in client.py starts by connecting to the peer_id port which the client searched for on the centralized server. The client then connects to ftp_Server.py which is the ftp server that provides file transfer services. Ftp_Server.py also receives the filename sent by the client and reads the data in its uploads folder and sends the read file. The client then writes the data to its downloads folder.

One thing I’ve realized is in client.py, the search functions for loop can be removed. That for loop was for debugging and making sure the right filenames were being sent to the server. If the for loop is kept in, then the Client will get “file not found” when searching for files registered by other clients/peers.