File Sharing Torrent like application

This project is a storage system and a file sharing application. There will be a server for the management and storage of files and multiple clients that can connect to that server.

When the server is opened it receives 3 parameters:
./server <port_server> <users_config_file> <static_shares_config_file>

And the client the following parameters:
./client <IP_server> <port_server>

There will be no threads used in the application and will be implemented a Round Robin algorithm for sharing simultaneous.

When the server is started there will be created a socket and it will wait for connections. Because there will be a login session before a user can do something, each client will create automatically a log file of his actions with the name: client-<ID>.log.

In the users_config file there will be a number N which represents the number of existing users in the system and afterwards there will be their credentials.

In the shared_files file there will be a number M which represents the number of shared files from users and afterwards there will be M lines with the name of user and his shared file.

In the system there won't be 2 users with the same account, but the same user can be logged in with multiple instances from different clients.

Commands available in the system:

- LOGIN

login <username> <password>

The client logs in with his credentials. If the user is already logged in with that client there will be a warning message. After 3 attempts with wrong login the server will finish the connection with that client.

- LOGOUT

logout

The client logs out and will be logged in with the guest account (the default one).

- GETUSERLIST

getuserlist

The client can see the active users in the system from that moment.

- GETFILELIST

getfilelist <username>

The client can see the files of that user. There will be a list with the name of file, dimension and status (SHARED/PRIVATE).

- UPLOAD

upload <file_name>

The client will upload a file up to server. If the file is already uploaded or the file is inexistent there will be a warning. The file will be uploaded in the user folder from the server.

- DOWNLOAD

download <username> <file_name>

The client will download that specific file if is shared by that user. It will be downloaded in the current user's folder.

- SHARE

share <file_name>

The client will share a file from his folder. This means it will make accessible for other users to download it.

- UNSHARE

unshare <file_name>

The client will make a shared file private.

- DELETE

delete <file_name>

The client will delete a file from his folder.

- QUIT

quit

The client finished the connection with the server. If there are files in the queue for download/upload, it will be disconnected after the files are downloaded/uploaded.

For each command there will be a verification of the usernames, password, filenames, paths. If there will be differences between the credentials or the files

are inexistent, or the paths are incorrect there will be warning messages with specific codes.

- -1 : Client isn't authenticated
- -2 : Session already opened
- -3 : Wrong username/password
- -4 : Inexistent file
- -5 : Forbidden download. User doesn't have authorization.
- -6 : File already shared
- -7 : File already private
- -8 : Brute-force detected (if 3 wrong logins)
- -9 : File already on server
- -10 : File in transfer
- -11 : Inexistent user

The usernames and password with be alphanumeric with maximum 24 characters.