Faculty of Science

Course:	CSCI 2020U: System Development and Integration
Component:	Assignment #2
Topic:	File Sharing System

Collaboration Policy

You are permitted to work on this assignment in a team, and submit the results as a team. For this sort of assignment, with an open-ended component, the collaboration between multiple team members can be beneficial. Between groups, however, please limit the discussion to the level of general strategy (not code). Groups of size 2 are recommended. Larger groups will be considered with the proviso that the marker will mark your assignment with higher expectations. In any case, be sure that all members of the team fully understand all code, otherwise they will miss intended learning objectives, which may be a considerable disadvantage at exam time.

Overview

You and your roommates have decided that you want your own file sharing system, so you decide to write it yourselves. The file sharing system will expose a directory of files on the server to the clients directly.

Primary Instructions

The file sharing clients will connect to a central server, which will respond to a single client command, and then disconnect. Each time the client needs to issue another command, it will re-connect before sending the command. The server will respond to the following commands:

- `DIR`
  - Returns a listing of the contents of the shared folder, on the server's machine
  - The server will disconnect immediately after sending the list of files to the client
- `UPLOAD` *`filename`*
  - Immediately after the newline after this command will be the contents of a text file
  - The server will connect the text from this text file, and save it as a new file *filename*
  - The server will disconnect immediately after saving the text file's contents
- `DOWNLOAD` *`filename`*
  - The server will load the text from the text file filename, and will immediately send that text to the client and disconnect

Server

The server does not need to have any user interface, but it must be multi-threaded. Each incoming client connection should be handled with a separate thread (`ClientConnectionHandler`). This thread, and its corresponding socket, will remain open only until the command has been handled.

Client

The client will have a simple user interface. When the client is started, the computer name and shared folder path are passed as command-line arguments. The client will then show a split screen showing two

lists. Both lists will consist of filenames. On the left will be the list of all files in the shared folder of the local client. On the right will be the list of files in the shared folder of the server.

***Note**: No mechanism for navigating to sub-folders is required for this assignment. It can be presumed that only text files, not directories or binary files, can be in the shared folder.*

At the top of the window will be two buttons. A `DOWNLOAD` button, when clicked, will cause the file selected in the right list to transfer from the remote server's shared folder to the local client's shared folder. This transfer will simply be a copy of every character in the file across the network.

An `UPLOAD` button will facilitate a transfer in the other direction (copying a file from the client's local shared folder to the server's remote shared folder). As with downloads, the upload will consist of a transfer of every character in the file.

When an `UPLOAD` or `DOWNLOAD` occur, the user interface will need to refresh both lists of files to show the newly uploaded or downloaded file.

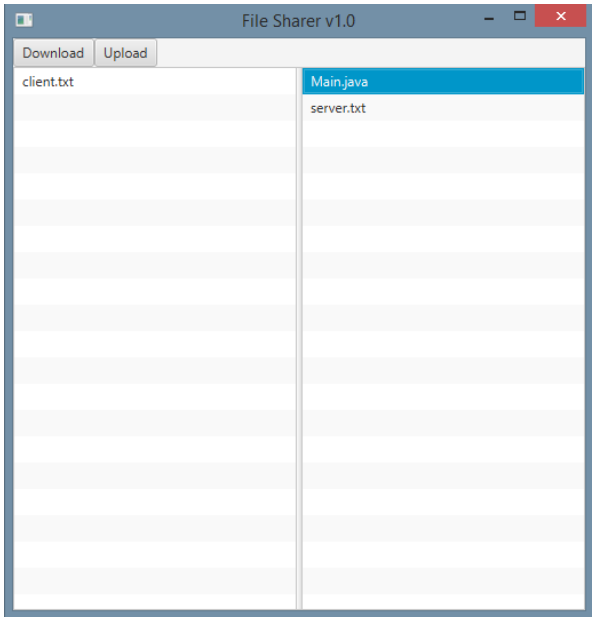

Figure 1: The user interface for the client application

Secondary Instructions (Optional)

Can you think of any way to improve the file sharing system? What if you could navigate to sub-directories in a shared folder? What if you could configure your shared folder graphically, instead of merely setting a variable? Could the user interface be made more user-friendly?

How to Submit

You will maintain a git repository for this assignment, shared among all group members, and the instructor will be added as an author to the repository (or you can simply use a public repository). To submit the assignment, create a single file 'README.txt' that contains instructions on how to download, compile, and run your application. A .zip, .7z, or .rar file will not be acceptable.

***Note**: Comments are mandatory. Failure to properly document your program will result in a deduction on the marks you receive for this (and any other) assignment.*