Exercise-4 Implementation of File Transfer Protocol using TCP

Develop socket programs in Java to implement FTP Server and Client. The requests and responses given below should be simulated. Also implement **cd, ls, get** and **bye** commands of FTP. Use TCP for communication.

Note: The bolded ones are client requests and other messages are server responses

Specifications:

Develop the application as GUI based application using frames, TextArea, buttons and appropriate EventListeners

The below box is a textarea. Both the textarea and the "Send" button should be on the client side representing the FTP client

FTP Server will be listening on the port (no GUI for server)

When the command is entered and "Send" button is clicked, it should be send to the server and the server should respond with response code and message.

Client: **ftp cs.ssn.edu**
Server: Connected to cs.ssn.edu.
220 FTP server ready
Server: Name (cs.ssn.edu:yourlogin): anonymous
Password:
230 Guest login ok, access restrictions apply.
Client: ftp> cd /pub/serverdir
Server: 250 CWD command successful.
Client: ftp> ls
Server: 200 PORT command successful.
150 ASCII data connection for /bin/ls (128.138.242.10,3133) (0 bytes).
Sample.txt
Test.txt
. . .
Example.txt
226 ASCII Transfer complete.
418 bytes received in 0.043 seconds (9.5 Kbytes/s)
Client: ftp> get sample.txt
Server: 200 PORT command successful.
150 ASCII data connection for sample.txt (128.138.242.10,3134) (2881 bytes).
226 ASCII Transfer complete.
local: yourdir/sample.txt remote: sample.txt
2939 bytes received in 0.066 seconds (43 Kbytes/s)
Client: ftp> bye
Server: 221 Goodbye.

Send

Commands:

cd – should allow you to change the dir

ls – should list the file in the dir

get – should retrieve the file from the server’s directory and store it in your local destination directory

bye – should end the connection with the FTP Server