**FILE TRANSFER PROJECT DOCUMENTATION**

By

BAO NGUYEN – BCN140030

CS 6349.001

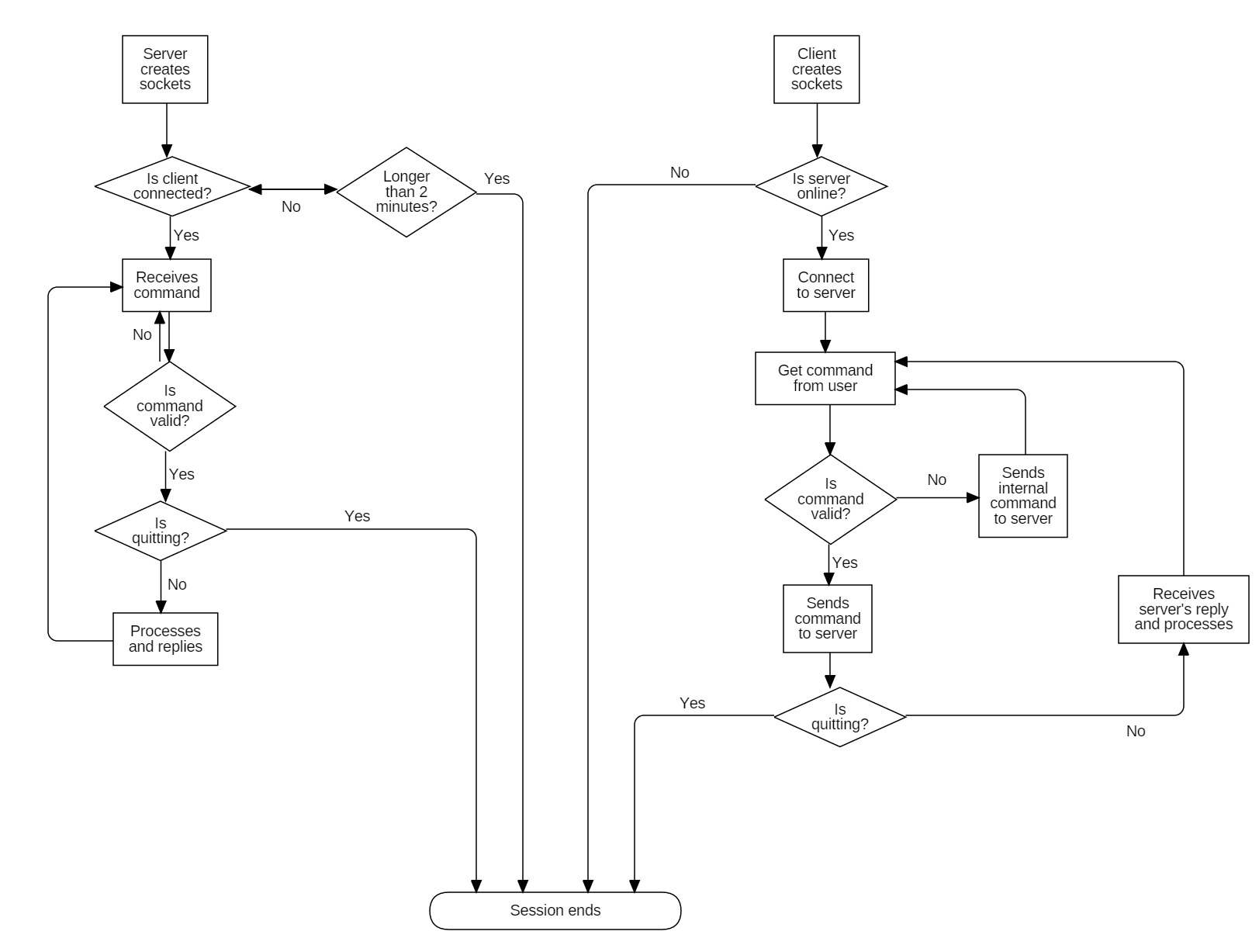
1. **Project description and solution**

The problem is to create a file transfer system that utilizes Socket Programming, Connection Management, Reliable Communication, and security protocol that utilizes SHA-1. It allows 2 individuals (1 server and 1 client) on different computers to communicate via network. *This project is developed toward Windows environment.*

I use Java to build this program. I create 2 different projects, a *Client.java* and a *Server.java*. The server sets up the sockets and wait for connection. The client sets up its socket and connect to the server. This program also records the start and end time of the session.

There are several commands that allows the client to communicate to the server. The server does not initiate any message. It waits for the commands from the client and sends responses. The client can request server to view all the files the server contains, to download files from the server, or to upload files from local host to store in the server. All messages transported over the network shall be secured with respect to Authentication, Confidentiality, and Integrity.

There are several occurrences when the program ends (the flow of the program is provided in the next section of this report):

1. If the server waits too long without any connection from the client, the program ends at that point.
2. If the client tries to establish a connection without an active server (on the other word, the client starts before the server does), the program ends immediately.
3. After the connection is established, they can continue communicating (the client starts first) until the client says “quit”. When this event occurs, all sockets are closed and the program ends.
4. During the session, if either the server or the client goes offline for any reason (indicating by a null message received by the other party), the one who remains online displays an error message to the screen to indicate the event and the session ends safely (without crash).
5. **Implementation**
   1. *Abstract Flow chart*

*Basic flow of program. Authentication mechanism will be added as separated diagram.*

1. **Code executions**

*a. Compile and run*

There are two ways to run the system: via IDE or via Command Line environment.

To run program via IDE (this was developed via IntelliJ IDEA from Jetbrains), first open Server and Client projects on separated windows. Then individually click “run” to run each code in parallel.

To run program in the command line environment, first navigate to where the Server folder and the Client folder located on your local machine. In each module, navigate to the src folder. Run command “javac Server.java” for Server and “javac Client.java” for Client to compile the code. Then run “java Server” to start the Server and “java Client” to run the Client. After this, run as program instructs.

To run program as jar files, to be added later.