1. To compile Server and Client
   1. Through separate project folder – may be simplest way to do this.
      1. Create new classes in the new project.
         1. Ctrl + A, and then copy and paste all code from following:
            1. Server.java
            2. ClientProgram.java
            3. Client.java
            4. AES.java
            5. History.java
         2. These may be in a package. Create appropriate directory “\ServerClient” and place these class files in them or remove the “package ServerClient” line at the top.
         3. Compile and run Server first, and then run ClientProgram twice. History and AES are auxiliary classes and will be compiled with all other classes.
            1. You may have multiple instances of a program. Pin the console on one server and separate consoles for the clients.
         4. In the zip file, there will be a file called ‘subscribers.txt.’ This is the list of subscribers that are able to pass the authentication. You may add your name and a number if you desire, with a space in between.
            1. For example: “Raymond 50”

Everything in the quotes may be added.

Note the number part, as you will need it for authentication.

* + - * 1. If you did not add an entry, look upon the list for any user and use these credentials.

I.E. “A 4”

* + - 1. Connection may be initiated by user.
  1. Compile and run through console
     1. May ignore this section if running it through a project works.
     2. Navigate to directory where java files are kept.
     3. Hold shift and right click, and open a command prompt/ PowerShell in the folder
     4. Type “javac [Filename.java] to compile.
        1. Complications may occur.
           1. Java mismatch for javac and java

Download and install matching compiler and java runtime environment/ or latest Java version for compilation and runtime environment. We used Java 11 for compilation and running. The latest version should work.

* + 1. Go up one directory and type “java [folder that class files are in].[actual java class name]”
       1. Don’t type .class at the end.
       2. May open up multiple instances of power shell for each program
    2. This may work if creating a new project does not work. Run program as intended.

1. To run, simply press the build and run button at the top of the button ribbon in Eclipse.
   1. For the server, this is all you need to do. Interaction for the server is finished at this point.
   2. Do the same for the clientProgram class, and use one console for each instance of a client.
   3. When you run the program, authentication is handled by clientProgram. It will prompt you to enter your userID and secretKey. Enter one from the subscriber list, or enter something else entirely for error checking.
      1. “A 4” works, and so does “B 7”
         1. These may be used for the demoing.
   4. Afterwards, the authentication process will begin. Once it finishes, if the credentials you provided are ‘correct,’ then “Authorization Successful” will be printed. Otherwise, “Authorization Failed.”
   5. When you are authorized, send a request to server to become connected by typing “Log on”
      1. Once you do so, you may begin a chat with another client by typing “Chat [ClientID]”
      2. If the client is online & not in chat, then a sessionID will be created for the both of you.
         1. sessionID is a counter for the instance of your chat session, e.g. “0, 1, 2,”
            1. These are the 0th, 1st, 2nd chats
      3. Once the chat has been set up, both clients will be notified that they are chatting with the other client.
      4. Afterwards, you may begin to type. Anything you type, besides the keywords, will be sent to the other user.
      5. Keywords to be used after logging on:
         1. End chat
            1. Ends chat with other user
         2. History [clientID]
            1. Displays history of chat with [clientID]
         3. Log off
            1. Closes TCP connection
            2. To be used after ending chat.