**Java logo chatroom project:**

**Class designs:**

Program flow of java sockets is as follows:

1. Open a socket.
2. Open an input stream and output stream to the socket.
3. Read from and write to the stream according to the server's protocol.
4. Close the streams.
5. Close the socket.

Server to client: String objects

Client to server: ChatMessage objects: String message, Type type

We want two controllers:

**Server controller**

FXML variables for each GUI element

Server class: instantiate this object and accept connection from a socket

Socket class: instantiate the socket object and open, attach a stream (input / output) to this

Closing classes: close sockets and streams

Stop class: to stop the chat program when no users and / or an admin decides to terminate the program

Wait class: displays a message saying still waiting for everyone to connect

Broadcast message class: A class to broadcast messages across all clients

Logout class: clients that have logged out, an message that says so

* Event class: class that contains messages other then chat messages i.e. logout

**Client controller**

FXML variables for each GUI element

GetServer class: gets server according to a port number etc.

Start class: starts dialog from server to client

Read messages class: collects messages from server and displays it on client’s screen

We want two views:

**ServerGUI.FXML**

Displays the server GUI view, FXML is coded into this view in order to display it

**ClientGUI.FXML**

Displays the server GUI view, FXML is coded into this view in order to display it

We want one java class for object instantiation

**ChatMessage.java**

String Message;

Type type;

**Screen designs:**

Below is a simple GUI interface I will expand into during the coding part of the project.

Users

Enter chat message here to be sent

Chat messages across all users

Send message

**Description of APIs used:**

Scene builder: JavaFX Scene Builder is a visual layout tool that lets users quickly design JavaFX application user interfaces.

Libraries: I/O, sockets, server, main java library (s)

Java

JavaFX

Git: source control

FXML, Controllers, models (MVC in java)

Netbeans IDE