

Alan Decowski

CPE-490

Professor Iyengar

Final Project

Stevens Institute of Technology

2020F

Hello! What I have completed over the past two weeks honestly came out as a bit of a disappointment to me. Meanwhile the basics are *there* the final prototype that I have finished and submitted was quite sub-par for my original goal. It is riddled with bugs and if you go to the next line- or quite frankly blow on it too hard it will crash. With more time a much more stable prototype could have been built, but it’s here.

To run: Sr.jar is the Server, Cl is the client. Server must be run from command line using   
“java -jar Sr.jar” and client can be run from command line or by executing the jar from your desktop

The point of this project was to create a network shared file that can be edited at multiple end points by different users. Think Google Docs. This solution already exists- and from multiple companies. The difference that I was going for was a utility that would allow you to do this when you are not connected to the internet, allowing a LAN service that you can connect to in order to have multiple editors on the document.

To build this I started with a simple network protocol- the point of this class. I built off a Java Websocket/ServerSocket pair for this communication. Java implements the use of TCP to get data back and forth. This is good for me, since the application I am creating is data-based this ensures no loss of packets, and thus no data loss. The communication to/from the clients is really simple. When a client connects to the server(which can be run off of anything from an editors computer to a Raspberry Pi thanks to Java’s portability), the server sends the entire file to the client in a really poor and obscure way that I thought was best for storing this file.

After this initial communication, the client will create a GUI that can be used to edit the contents. Each keystroke will compare the current text on the screen to a cache on the Client side. The difference between these two “DataModels” (as I so gracefully named them) is then put onto the line back to the server. The server then relays that difference information back out to the other clients. In a perfect world, all clients would remain on pace with eachother.

Shown in the demo video ( [here](https://www.youtube.com/watch?v=tjP0uma_mGQ) ) is how the program was supposed to look. Attempting to run the files yourself will definitively yield mixed results.  
This is not the first time that I have attempted this project. My Senior year in High School I attempted this. Unfortunately, that code is long gone, and if my memory serves me right, it was even buggier than the one I present you with. I had learned a bit from that one, but ultimately ran into the same types of bugs. (I.E. cursor not placing correctly on updates, new lines giving me really big problems for some reason)

I may continue this project, because this is an idea that I could have used on multiple occasions. If I were to continue, I may keep the network code- which was all compiled separately.