

Appendix

Verbal Transcripts

1.1: Initial Meeting, December 4, 2019

Me: Hello Mr. Devry, thank you for meeting with me. I heard you have a problem.

Client: Yes, I have a problem. My problem is that I want to be able to monitor what my students are doing on their computer lab PC's, and I'm not able to do that right now.

Me: Okay, have you tried anything to solve this?

Client: Yes, we bought a commercial package, but it proved difficult to set up and not intuitive to change or use.

Me: Why is that?

Client: It did not work happily in the network we had in this lab. It had a special configuration file which you manually had to build, it offered all kinds of stuff that we would never use so it was overly complex for what we needed to be able to do. Primarily, it did not offer us what we specifically wanted.

Me: So, would an ideal solution streamline the application to its most basic necessary features?

Client: Yes, absolutely. Once the PC's in the lab are configured, the software should be able to automatically pick up the PC's that are running and should be able to locate a PC in the room, click on it, and see what is being run on it. That's the basic bottomline for what we need it to be able to do.

Me: Are there any additional features you would like?

Client: Yes, I would like to be able to send a message to a particular PC. So, I want to be able to select a PC, type in a message, and have it show up on their screen. That would be nice.

Me: And then, would you like any configurable options or is it alright to have everything preconfigure?

Client: Just have them preconfigured, I think that would be fine.

Me: Alright, well thank you for meeting with me, Mr. Devry. I will come later when we have the time to discuss the problem in more detail.

1.2: Discussion of Problem, December 12, 2019

Me: Hello Mr. Devry.

Client: Welcome back, Jonathan.

Me: Thank you for meeting with me previously. I would like to talk in-depth about the issue and how I might go about implementing a solution.

Client: Okay, well we would like to be able to just have an installable file on a jump drive or something where we can simply go around to each PC and install it using our admin password, and have it boot up and run whenever the PC starts up. Those are for the clients. The host, the teacher's computer, should be able to detect all running PC's in the lab when the server is started and perhaps show a list of who is connected. The ease of installation is a big factor since the PC's are often updated by the district, so we need to be able to go back and just quickly reinstall this piece of software.

Me: Well, I've been thinking about using Java 8, in specific, for the language of the product. Do you think there would be any issues with updates or whatnot?

Client: We are standardized on Java 8 for this computer lab for the foreseeable future, so I don't think that would be an issue.

Me: And do you think there would be an issue with the network, any problems with connectivity?

Client: Possibly. I'm not 100% sure if all these computers are connected to the same hub or switch, so there may be some issues there that will require further investigation.

Me: Alright, well thank you for meeting me. I will come back later to present a basic design.

Client: Okay, excellent. See you then.

1.3: Presentation of Design, December 30, 2019

Me: Hi, Mr. Devry.

Client: Jonathan.

Me: So, I've come up with a basic design for the application and it includes a rather basic UI for the user, the host. It consists of just a column that will display the user list with a button at the top where you can start or stop streaming the student's screen, a large screen that will display the student's screen, and finally a textfield at the bottom where you can send messages.

Client: Yeah, that works fine. I like that. Will I be able to just click on a PC in the list?

Me: Yes, and it will only show currently connected PC's, so when you click on one it should begin to stream their screen.

Client: Okay, sounds great.

Me: Anyway, that's the UI design. In terms of the processes, I've come up with a few key classes I may need to make the program. So, some of these include the Client class, which will be the program distributed to the students to send their screens to the central server, a DiscoveryHandler class, which is essentially a datagram socket in Java which uses UDP

broadcast to allow the students' computers to discover and connect to the central server, a ClientHandler class which will handle a specific client's connection validation, the processing of the data they receive from the students, getting their usernames so you can actually identify them, since the IP would not be sufficient, and also sending them specific commands to start or stop streaming. Finally, I will need the Server class for the teacher which will handle the acceptance of clients, their selection of a specific client, streaming that client's screen, sending a message to that client, and finally updating the user list when a student's PC connects.

Client: That sounds excellent. Anything else?

Me: In terms of my specific success criteria, I was thinking about seven specific ones.

Client: Alright, and what would those be?

Me: Well, the first would be the simplicity of the GUI, the second the identifiability of the students, the third the ability to stream of the students' screens, the fourth the teacher's ability to switch between clients, the fifth the ability to send messages to the students, the sixth the automatic connection of the students to the central server, and the seventh the automatic disconnection of the client after their connection is broken.

Client: Okay. Those all seem aligned with what I wanted. Hadn't thought about that last one.

How do you think you'll go about accomplishing things like identifying the client or automatically disconnecting them?

Me: I'm planning to identify them by using the logged in user's username, if that's fine, and the automatic disconnection by validating the students' connection by sending a packet to them and the server kicking them out if it doesn't receive a response within two seconds.

Client: Oh, that's nice. It all sounds pretty good, I think you can go ahead and start.

Me: I'll get to coding then. I'll meet with you again after I finish the program.

1.4: Presentation of Product, January 7, 2020

Me: Hello again, Mr. Devry.

Client: Hello, Jonathan.

Me: I've finally finished the product and would like to present it to you.

Client: Well I'm looking forwards to seeing it.

Me: So, here's how it works. You just start up the client program by double clicking the Client.jar on the students' computer, which should run in the background. Actually, could you run it as an administrator? It seems to need that.

Client: Sure, let me just quickly log in here.

Me: Alright. Over on this side, whenever you start up the server program by double clicking on Server.jar on your computer, it'll open up a window where you can interact with the clients. Could you get this one too?

Client: Yes, just step aside for a moment there.

Me: There we go. Anyways, for example, you can click on the student to get a stream of their screen, send a message, and once the client program is closed, it will automatically be cleared from the client list on the server program.

Client: That seems all well and good. I think it'll work great.

Me: Is it okay if we talk about the success criteria for the IA? It's just the last bit for today.

Client: That's fine. We can just sit over here and have a quick chat about it.

Me: Alright. So, if you recall, these were the seven success criteria I defined around the start of this process.

Client: Yes, I remember.

Me: I need you to talk through each of them, just to make sure I pinned down what you wanted. I'll just go ahead and talk through each of them. Do you believe that I fulfilled criterion A, that the GUI should be as simple as possible?

Client: I believe you fully fulfilled that requirement. It's intuitive to use, no need for any instruction or anything, and all the controls are right there. There were also no unneeded features, so the application is simple enough to not be overly complex but still accomplishes what I wanted it to be able to do.

Me: Thanks. How about criterion B, which is being able to identify the connected clients?

Client: Well, clearly the clients showed up with their username, so I will be able to rather easily distinguish between the owners of the screens I'm streaming.

Me: Alright, well how about criterion C, which is just being able to stream?

Client: It definitely works to stream their screen, just clicking on a client from the list demonstrates that rather clearly.

Me: And criterion D, switching between clients?

Client: Yeah, with multiple clients connected I can click one and see only their screen. It also doesn't seem to break when I'm swapping.

Me: How about sending messages to individual students, criterion E?

Client: It seems that typing into the text field at the bottom and pressing enter sends the message over to the client, so yes, I believe you've fulfilled that requirement.

Me: And then F, which allows the clients to connect automatically to the server?

Client: Yup, running the client program and then later launching the server seems to allow the client to connect.

Me: Finally, how about G, which is just the client automatically being cleaned up by the server when they disconnect?

Client: If we shutdown this client program, it seems that their username does disappear from the server's user list after a bit, so yes, I believe you've finished that criterion as well.

Me: Okay. So overall, do you feel satisfied with this product? Is it up to your standards?

Client: Yeah, I'm quite pleased with this product. It's easy to use and it fulfills all the requirements we discussed. Most importantly, it works, so that's good.

Me: Thanks, Mr. Devry. Just to finish up, do you think there are any features you would like to have implemented in the future?

Client: Well, I would prefer an install script of some kind which would let me just double click on that, then automatically have the client program launch on startup for any computer. It takes quite a bit of time right now to add it to the startup list of each PC.

Me: Alright. Also, I saw that the server had trouble connecting to the clients at one time. Would you want some way to configure the timeout on the connections between the client and the server? It's two seconds right now, but it seems that it's way too short.

Client: Yes, that would be excellent too. Maybe you could just up it to ten seconds, as that would be plenty of time, even with the lab's slow internet.

Me: Yeah, I can get that done for you right now. Any more recommendations?

Client: As we saw, we had a bit of trouble figuring out this issue with the disconnection time, so it would be wonderful if you could add in some kind of diagnostic tool which maybe just displays logs any issues into a text file of some sort. It would make debugging a lot easier.

Me: Alright. Well, I'm happy to have worked with you, Mr. Devry. Thanks for meeting with me and being my client. I'll see you later.