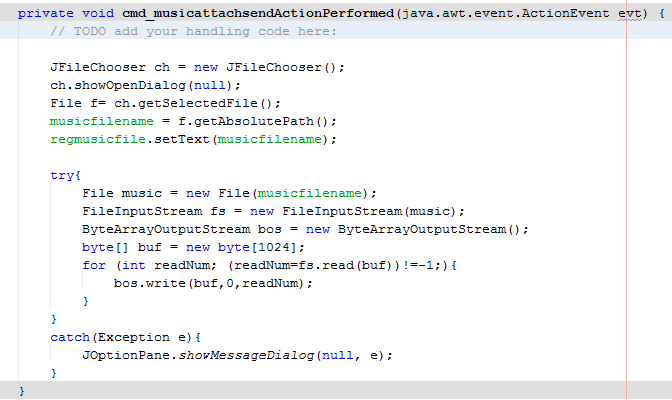
Furthermore, we also start the server that will be always active for the social networking. This server allows for adding friends, listing people online, showing who your friends are online and posting to friends. For the posting it is basically the same as chat but posts to all your friends. It also lists people online in pretty much the same manner as the chat, in that users names are displayed in the chat. Users can add each other as friends by typing their name from the people connected and pressing the friend request button. The user being accepted can then accept their request and add them as their friend and then send a request back to the other user so they can have that person their friends list.

One of the other main features from our social network form is that it can allow sharing of music between friends. It works by a user selecting music using the attach button and then clicking the share button. This then starts a server with the users own personal port and posts a message in the post saying there sharing their song. Friends of this user can then select that users name and press the download button and they will receive the song.

Now if we dive into the code of this a bit more, we can look at the first button of attach music. This uses JFileChooser to open up a dialog so that user can select what file they want to upload and it sets the text field to that file path. Then using a try-catch we create a new file called music, which uses the music file name and then this gets all the file information.

Here is code for this button:



The next button share is basically what uploads the file to the server so it can be downloaded. Firstly it starts it as a thread, then it sends a message to the news feed that says they are sharing a song and the writer gets flushed. Then using a try-catch the share creates a socket and then confirms a connection. It then creates a new file using the file path from the text field. It then creates an array of bytes form the file length. Then it create a new file input stream (fis) and buffered input stream (bis). Then it gets the output stream (os) from the socket. It then writes to the byte array to the output stream and then that is flushed. Finally it closes the bis, os and sock. It also has created another stream which runs the main program.

Here is main code for this button:



The final button is download. This firstly uses and sql statement. The statement checks the username selected in friends list is in the database and then finds the portid number, which is the port of the person sharing. It then uses this port number to create a socket to access the file on the server. It creates a new file output stream (fos) and new buffered output stream (bos). It reads the my byte array and bytesread equals that and then current equals that. Using bos it writes my byte array and current. Then bos is flushed. Finally everything is closed and a pop up message says the file is downloaded.

Here is the code for this button:





In conclusion, overall we believe we have demonstrated go system software skills. We have done plenty of out of hour’s research, learning lots about how servers work, which was something neither of us had ever done before and our first exposure to Java, but we worked very hard and manged to produce a good quality piece of coursework. This has been demonstrated, we also have a very good knowledge of SQL and use it to good effect to create a good database to hold our clients. The layout of the jframe is easy to use and has some complicated features.

References:

* [varietytubechan](https://www.youtube.com/channel/UCVs76UZVOiH43nh6YzSsQFQ)’s chat client and server side tutorials: <https://www.youtube.com/playlist?list=PLCBF515A460A08185&nohtml5=False>
* Music file transfer: http://www.rgagnon.com/javadetails/java-0542.html