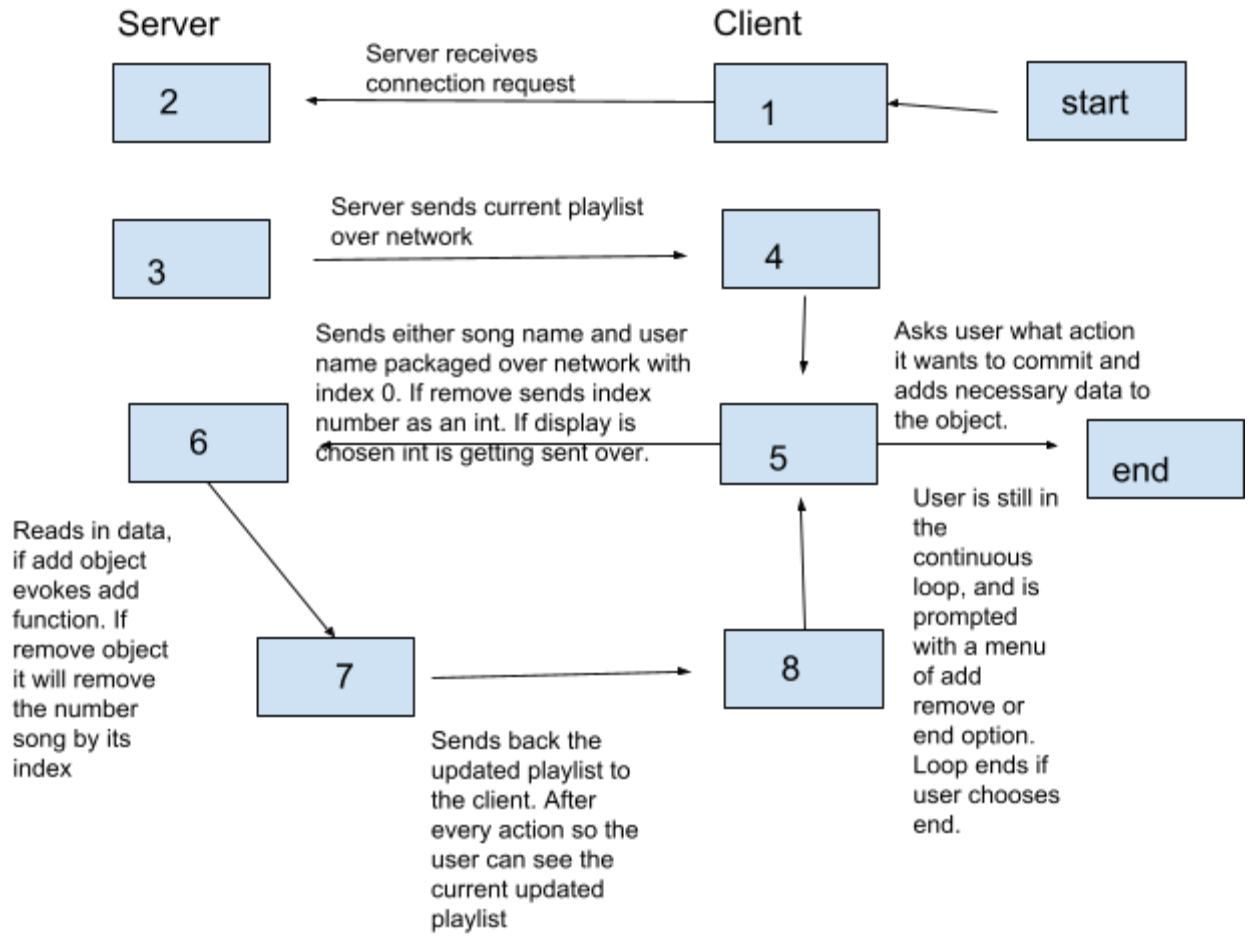

When committing an add song action, the object sent across the network is an object called “song” looking like this:

Index:	song name:	username (given through command arguments)
1	song1	joe
2	song2	jon
3	song3	jack

If you remove a song, the user is prompted to enter an int which is sent across in an object

This object will consist of just the int number of the song it wants to remove. It won't be anything else because it is checked on the client side. The first value will be the action so it will read in either 2 or will send an error. There will be a check on the client side before the information gets packaged into the object so there isn't random data being sent to save resources

- It will go to the next value and find what index it wants to remove. If the index is not there, an index out of bounds exception is thrown. If its there index is removed.

- When sent: index will always be set to zero and will be adjusted when received by server, each client will send this to add songs. Users could add the same song like in a real playlist but the synchronization issue would be both users removing a song at the same time. The playlist is the shared resource between the clients, they need to be managed accordingly to make sure we do not have any synchronization issues.

- We need to address the problem of having users remove the same song at the same time. This is a synchronization issue that needs to be dealt with.

- Information will be stored as objects and the objects will be stored in an array. The array will hold the objects on the server and will have a function to print them in a nice looking list.

- Restrictions of the system are limited to only adding and removing songs. There is no artist name, and we are assuming that the user is actually giving us real songs and not just random messages to keep the integrity of the program. So it remains a "playlist".

- Each thread will be responsible for sending the information over the network so the server can interpret

- Users on the client side will be prompted with a text menu which will be run by a while true loop. It will continuously loop until the user chooses option 4 which is to quit. Which will then return false and end the program.

- If the user chooses to have the playlist sent. We will send the array across the network.

This array will have the objects with the song names and the names of the person who added it. The array can be sent across the network and will be read in and printed by the client.
The client will only be reading in information for the display functionality. It won't need to read in any data because the server sending messages back might be too much sending and might not be necessary.