FEATURES CHECKLIST

Student Number: 100610788

Please complete this checklist and include it in the zip file with your submission. If you have completely implemented a feature, then you can just state 'Complete'. If there are known issues with the implementation of a feature, please provide details.

Basic Requirements 40% or higher – 50%

Feature	Level of Implementation - Grade (max 50%)
The client and server components must be capable of running on separate machines.	[3] Complete
The client must be able to play .WAV files	[3] Complete
All audio tracks must only exist on the server	[3] Complete
The client and server must communicate and transfer data using sockets. The client must NOT read audio data from the server using any file-handling functions	[3] Complete
The server must be able to accept connections from and process requests from multiple clients concurrently.	[3-4] Complete
The user of the client must be able to request a list of audio tracks from the server and the user must be able to select one for playback	[3-4] Complete
The user of the client must be able to start and stop playback of the selected track.	[3-4] Complete (with bugs)
The server should be a console application that does not require any user input during execution. The server may access a configuration file or the registry for sufficient startup information required to establish communication with clients (such as IP port number to use, etc.). All other configuration of the server should be done via the client.	[3-4] Complete
The server may output logging or diagnostic messages to the console.	[3-4] Complete
The server must be written in unmanaged (native code) C++. You may not use Microsoft's managed or universal application extensions to C++.	[3-4] Complete
The part of the client application that communicates with the server and plays the audio must be written in unmanaged (native code) C++	[3-4] Complete

Systems Programming (6CC514) – Features Checklist 2022-23

The audio content must be played by the client using the DirectSound API	[4-5] Complete
The size of any individual buffer used in the client to receive data from the server can be no larger than 128KB.	[3-4] Complete
The size of the secondary buffer used in DirectSound to hold the section of audio to be played can be no larger than that required to hold approximately four seconds of music.	[3-4] Complete
TOTAL	[40-50]

Additional Features 50% or higher - 100%

Feature	**Level of Implementation - Grade (max 30%)**
Use of modern C++ features, including smart pointers.	[0-5] Complete
Client provides a graphical user interface.	[0-15] Complete
Other additional features. For example: - Creation of a playlist of tracks that automatically play one after another with the ability to fade out one track and fade in another. - The user interface shows the progress of playback of the audio file. - The user interface allows the control of the volume of the playback.	[0-10]
TOTAL	[50-80]
List any other features implemented	**Details - Grade (max 20%)**
Implementation of the server component as a Windows service or a Linux/Unix daemon.	[0-10]
Ability to play other types of audio file as well as .WAV.	[0-10]
TOTAL	[80-100]