# Architecture Plan

## Introduction

UFmyMusic will be a networked application that synchronizes files from multiple machines. It will follow the client-server architecture, where the server is a centralized machine that houses the current state of files.

## Client-Server Architecture

The server should be capable of handling multiple client connections simultaneously. It should also be able to store files and respond to requests while giving the appropriate response.

The client should communicate with the server via a command line interface. They solely have the ability of initiating and breaking down a session with the server.

## Message Structs

The messages the server will respond to are LIST, DIFF, PULL, and LEAVE.

All messages will have a common header that carries the information of the message type (1-4 inorder respecting above list) and the type (request or response)

**LIST request message will just have the header**

**DIFF request message will have the header, filenames, and hashes of the contents of those files**

**PULL request message will have the header, filenames, and hashes of the contents of those files**

**LEAVE request message will just have the header**

**If the client sends a LIST request, the server will send a LIST response that will contain the filenames of all the files it currently has**

**If the client sends a DIFF request, the server will send a DIFF response that will contain the filenames of the files that aren't in sync**

**If the client sends PULL request, the server will send a PULL response that will contain the filenames and contents of the files that aren't in sync**

**If the client sends a LEAVE request, it will close the client's connection to the server**

Anything else will respond with an error message detailing the only messages the server can respond to.

Hashes will be computed using the SHA-256 algorithm.