README

In our program we have set up a system that uses the client server connection based on the user has put into the command line as an argument. On the client side we have catches for each of the twelve commands. The three that we don't access the server is configure, add, and remove. These functions are all local on the client’s side.

When we were to connect to the server for the first time we would have a check that would see if the directory that we want to put the project existed or not. If the directory did not exist then we would create one so that we know the path would be the same every time to get to the project.

We also incorporated threading for the connection to the server so that it would be able to accept multiple connections from multiple clients. The only time that this would run into an issue is when there is multiple pushes to the file at the same time, since it would result in a merge conflict. The fix that we used for this is that we put in a mutex that would lock until the first push is done and then it would move onto the second instruction.

The design we have for the server is that it is constantly listening for a connection. Since we are able to have multiple connections to the same socket, even though it might already have a connection it will still continue to listen.

The client side is where most of the work is being done. Since most of the project is done through the client’s side this is where much of the code is. We have that based on the user input it will go to the proper method and then connect to the server. Once we are done with the function and the action is done we then break the connection. We do not keep the connection throughout the entire program since it can lead to errors if there is contradicting arguments that are given, therefore we have to establish a new connection every time we call a new command. This does not affect the client since it will always need to listen.