



GDrive Java Implementation

Design Document

CSC 3374

Dr. Omar Houssaini Iraqi

Submitted in

17th January 2022

by

Youssef Gaimes

1. Purpose of the Application:

This client-server application allows taking control of a server directory in order to back-up files and synchronize that directory across many clients. The system allows:

- Uploading files to the server
- Receiving files from the server

2. End-User Manual:

2.1. Running the server:

The server is a NetBeans project that follows its default structure. Its directory is **Server/src/main/java/com/mycompany/server**. It contains the server runnable java file as well as supporting files to ensure the good functioning of the application. In order to start the server, run the `Server.java` file.

2.2. Running the client:

The client is also a NetBeans project with the default structure. The directory of the application is called **Client/src/main/java/com/mycompany/client**. It contains the client runnable java file as well as its supporting files. The client runnable file can be run multiple times to simulate multiple clients connected.

3. Protocol:

1. The client opens a connection with the server under port 1000.
2. The client starts by synchronizing its directory with the server by sending the header: **SYNC**
 - a. Upon receipt of the verb **SYNC**, the server checks whether the directory is empty or not. If it is, the server sends the following message to the client: **NO**
 - b. If the server directory is populated with files, the server returns the following: **OK[number of files in directory][File1 name][File1 Content length][line feed]...**
 - c. The client reads the given response and updates its own directory.
 - d. The client sends the **SYNC** verb periodically in order to ensure synchronization.
3. Upon pressing the button “Add File”, the client prompts the user to select a file to be backed-up. The client sends the request with the **ADD** verb using the following convention: **ADD[file name length][file name bytes][file content length][line feed]**

4. GDrive Structure:

The application's client software initiates the connection by creating an instance of the socket class and supplying it the IP address of the machine hosting the server and the port to which it is listening.

Depending on what the client would like to do, it sends one of two verbs which are:

a. **ADD verb:**

The verb is created in string format and is sent to the server. Then, the client sends over the file name and file contents through the same socket connection. The server then stores the given file to the shared repository.

b. **SYNC verb:**

The verb is sent periodically to the server in order to return the entirety of the repository. This allows synchronization between the server and the connected clients.