**2225-CSE-6324-001-ADV TOPS SOFTWARE ENGINEERING**

**Team-7**

1. Bhatt Jatan Rajeshbhai - 1001927386
2. Patel Jay Ashok - 1001870971
3. Patel Heer Chirag - 1001965102

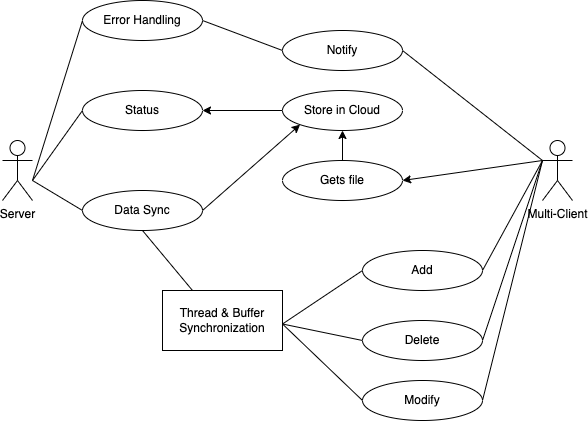
**Definition: Java to implement a simplified cloud-based storage application**

**Use Case Diagram**

The User can Transfer multiple files (i.e., add a file) to the Server using UDP Protocol which will in turn create multiple threads and create multiple Buffers having a size of around 4MB.

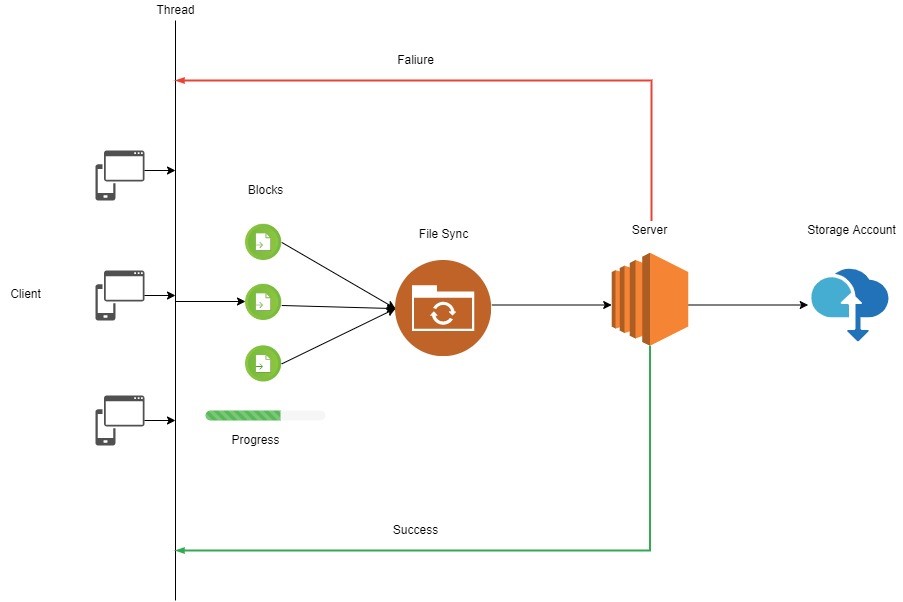
The Data Synchronization takes place, and the data is stored in the cloud-based server and the synchronization status of each file is sent to the server.

During this process if there are any errors the Server will notify the User.



**High Level Design**

One can upload/delete/download operations. We have used cloud storage to store files, one server with whom the client would interreact. There can be multiple users who can connect to one server.



**Overview**

We have implemented a cloud-based application similar to dropbox where users can download, upload, and delete files.

Users can upload multiple files to the server, creating multiple threads and buffers.

Data is Synchronized and stored on a cloud storage and its status is sent by the server to the client.

File is downloaded using blocks because if error occurs during transmission, then only those blocks which are not transmitted will be sent again instead of sending whole file again.

Multiple users can work together on a single server.

**File Handling**

Concurrency is maintained during uploading, downloading, or deleting the file using multi-threading.

Status of synchronization is continuously maintained when a client sends the file to the server.

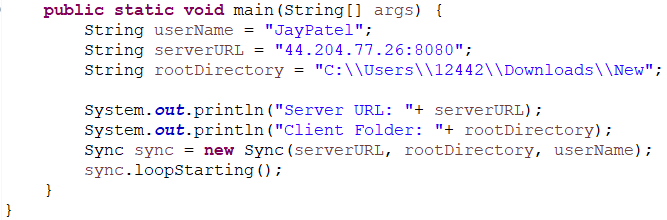
Error handling is done if there is any error, if the connection is lost then client will not be able to send until the connection is reestablished.

When a user deletes a file on the client side, the thread detects the change and deletes the file on the server side.

.

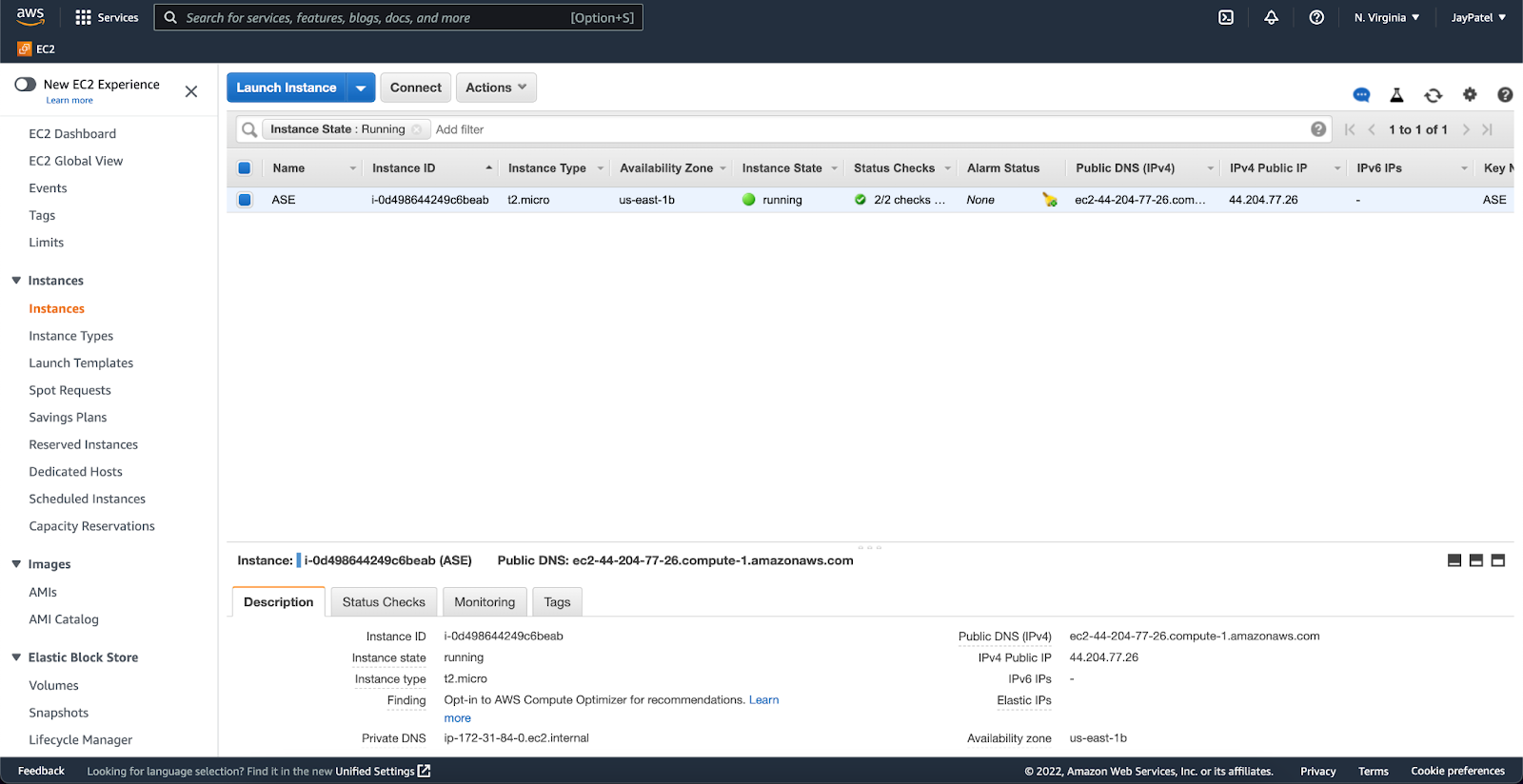
**Synchronization server**

We have a Synchronization server which ensures that any updates from one client as sync with other devices. We also have a monitor to check the sync status.



**Cloud Storage**

For cloud storage we are using AWS (Amazon Web Services) EC2 to store data on the cloud.



**Thread Management**

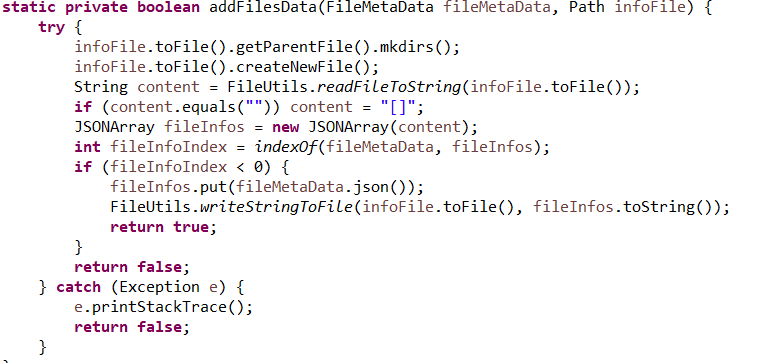
In implementation all this information will be abstracted in command line as an argument, and we implement one Synchronization class and, in that class, we will start clientLoop thread.

The thread will execute in synchronized manner, and it will work for multiple clients. All the users are synchronized with one server at a time.

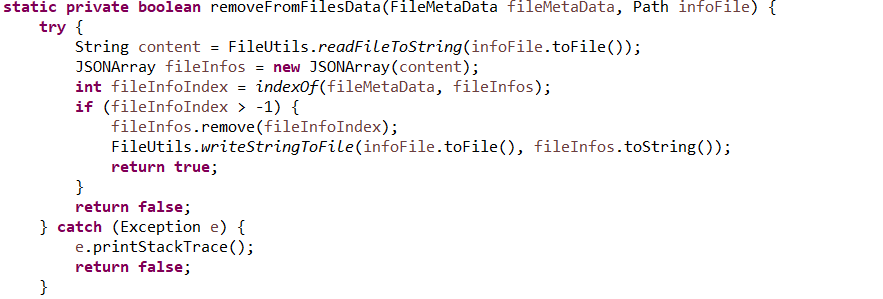
Also, data is synchronized for example when client makes any changes, server will detect that and make changes on server side.

**API**

This API is used to add files to the server.

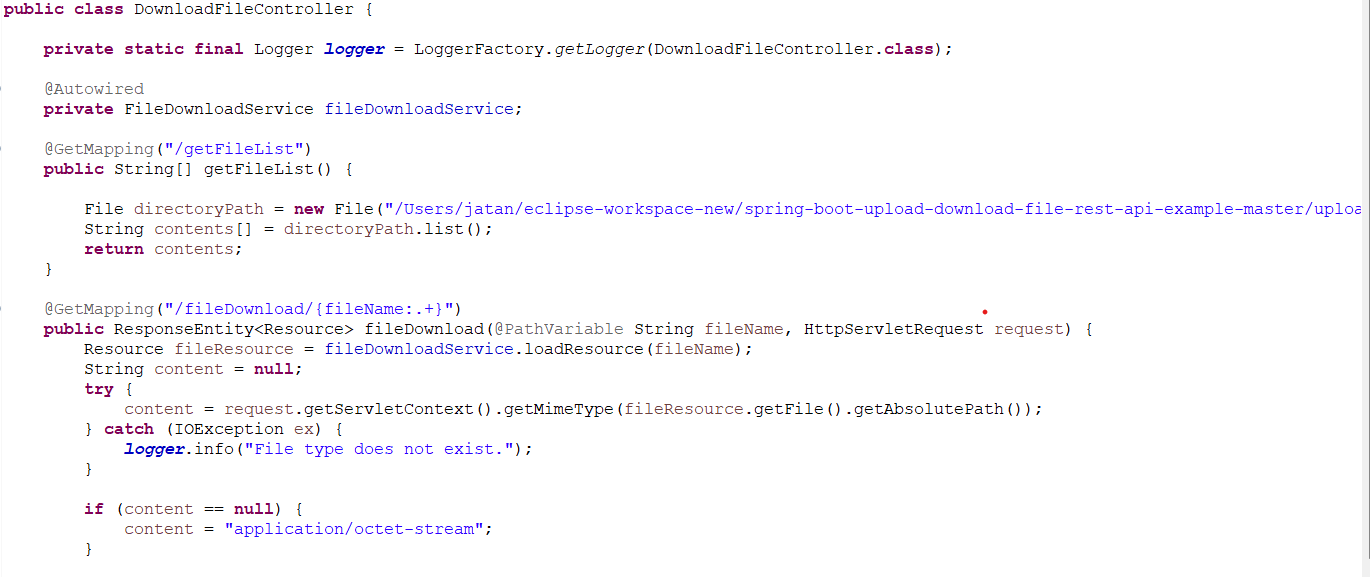


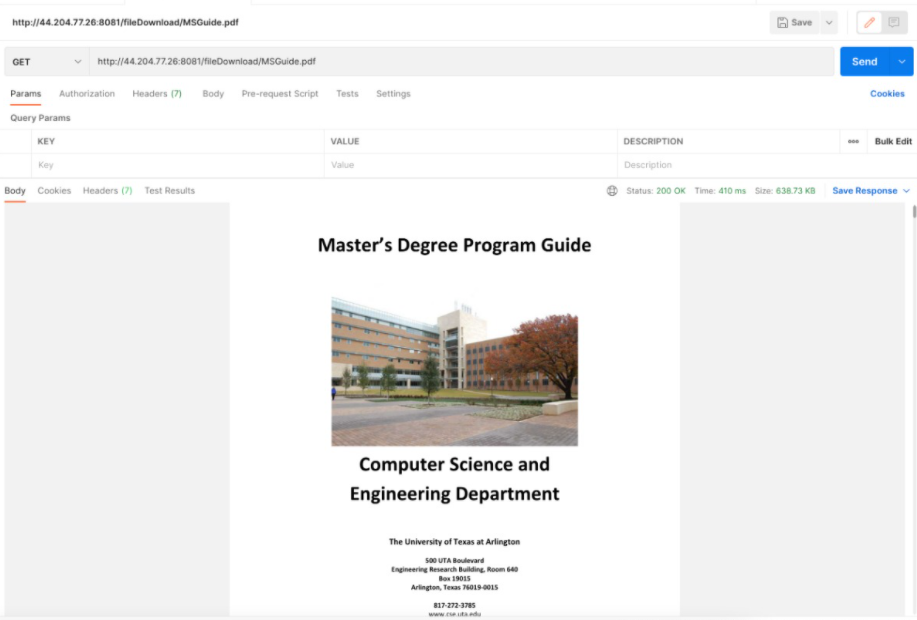
This is an API to delete files from server.





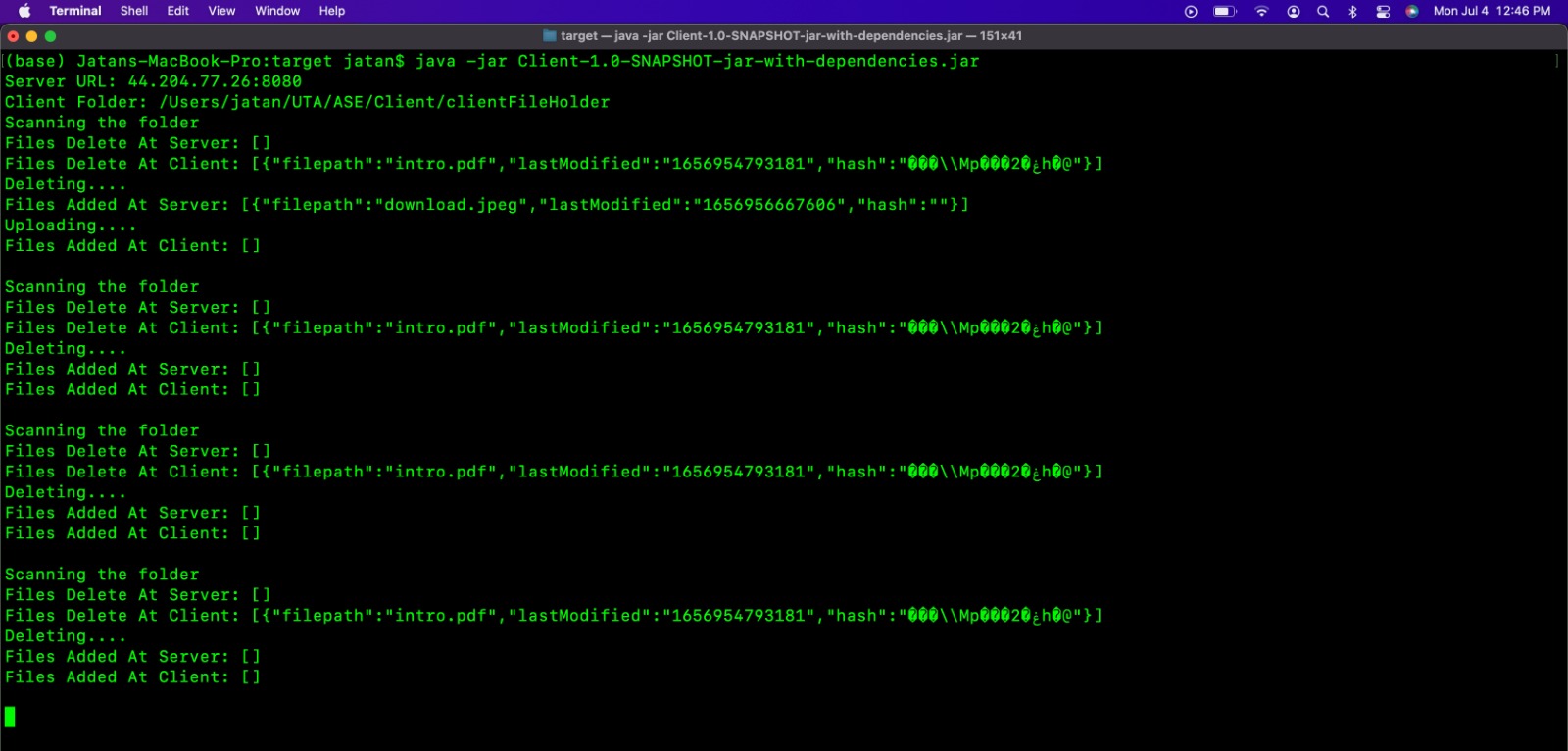
This is an API to download files from server.

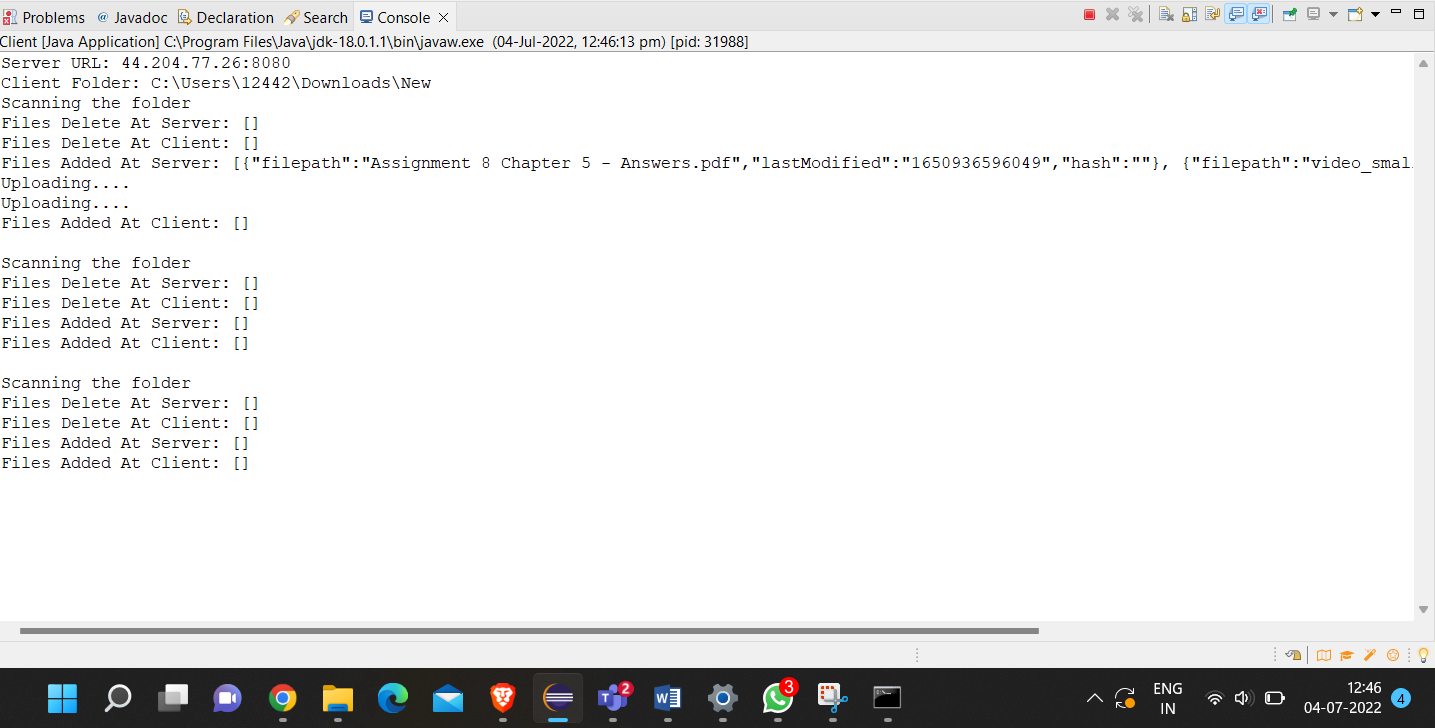




**Test Plans**

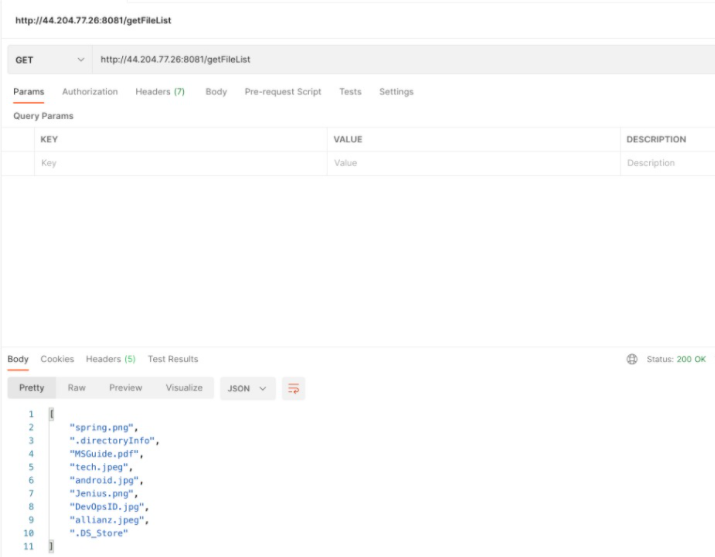
Multiple users can be connected to the server.



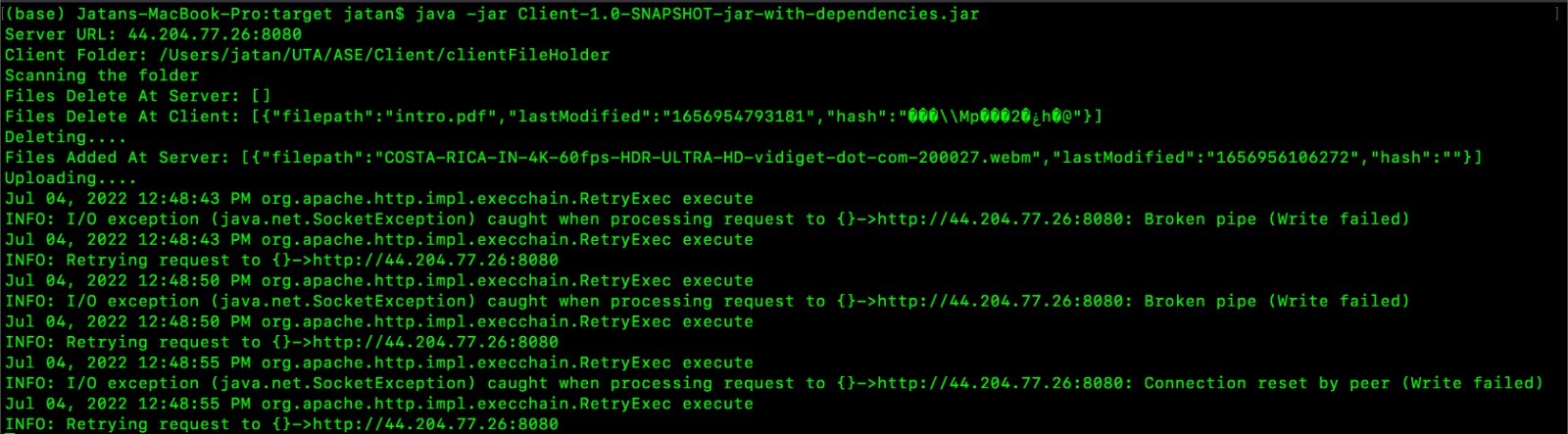


The file which client uploads should get uploaded successfully on cloud. If there are multiple files than in cloud, there should be multiple files. Correctness should be archived.





Error Handling



**Lesson Learned**

* Time management.
* Distribution of tasks among teammates.
* Dealing with sockets.
* Multithreading.
* Synchronization.

**Challenges faced**

* How to deal with multiple clients.
* Cloud connection.
* Hosting server on cloud.
* Sending file in blocks.
* Time management.

**Collaboration Plan**

# *week: 06.25.22 – 07.01.22*

## TASKS TO BE PERFORMED THIS WEEK:

Task Name(s):

* Requirement gathering - Heer
* Make an Architecture for the project - Jay
* Get high level Understanding of UDP Protocol to transfer file  -Jatan
* Work on Initial code to Perform File storage using Threading  -Heer
* Cloud storage management  -Jay
* Test cases -Jatan

Number of hours allocated:

Actual hours put in: < member >

1. 25: Jatan
2. 25: Jay
3. 25: Heer

# *week: 07.01.22 – 07.08.22*

## TASKS TO BE PERFORMED THIS WEEK:

Task Name(s):

1. Test & debugging the code and solve the errors: Heer, Jay
2. Updated Proposal and High-Level Design: Jatan, Jay, Heer
3. Make final code with Java Code standardization implementation: Jatan
4. Make ReadMe of the Project & add comment in code: Heer, Jay
5. Making a Final Test Report: Jatan, Jay, Heer

Number of hours allocated:

Actual hours put in: < member >

1. 18: Jatan
2. 18: Jay
3. 18: Heer