

# 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

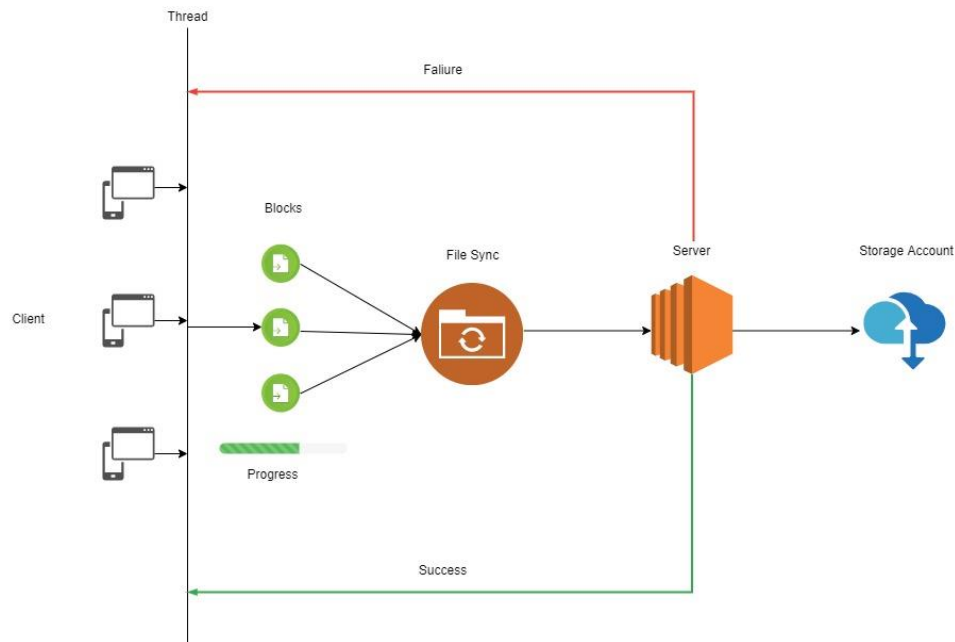
## Index

• Project Definition	-----	(2)
• Architecture	-----	(2)
a. Client	-----	(2)
b. File Handling	-----	(3)
c. Synchronization server	-----	(3)
d. Cloud Storage	-----	(3)
• Thread Management	-----	(3)
• Collaboration Plan	-----	(4)
• Project Schedule	-----	(4)
• Test Plan	-----	(6)

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

### High Level Design

One can upload/edit/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.



### Client

A client has jar file which to start the application and can perform upload, download, and modify files to backend cloud storage. Below are some major operations done by the client -

#### Upload/download file

Client can upload or download any text or binary which is lesser than 200 MB.

We may divide the file into smaller blocks to make it easier to upload.

#### Delete file

Client can delete files which are stored in server.

## **File Handling**

We will use smaller blocks to upload the file because if there is any failure at that time we can only resend the remaining blocks, we do not need to send whole file again.

As we are sending modified blocks only to the server in case of update instead of the entire file again, it will decrease the cloud storage consumption.

Transmitting the entire file at once consumes a lot more time as compared to small blocks. With multiple smaller data blocks, we can make use of concurrency also to upload/download the file using multi-threading or multi processes.

As we are only transmitting the modified block in case of updates, it helps us in proving a history of versions of the file. We can directly lookup at the modified block to see the modifications.

## **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 Azure to store data on the cloud.

## **Thread Management**

First user will access client jar file in which user will provide server link and path from which he wants to transfer the file.

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.

**addfiles (fileInfo, metadata, serverUrl)**

**editfiles (fileInfo, metadata, serverUrl)**

**deletefiles (fileInfo, metadata, serverUrl)**

**downloadfiles (fileInfo, metadata, serverUrl)**

This method will hit different server APIs according to usecase. Once the client submits the file it gets divided into blocks and sends it to server. Server will get file buffer blocks in a thread synchronization and pass the status to the client.

## **Collaboration Plan**

WEEK: <. /06.25.22 – 07.01.22> | MEETING ORGANIZER:<. /JAY>

MEETING ATTENDEES: <. /JATAN> <. /HEER> |

MEETING MODALITY: VIRTUAL/IN PERSON

### **TASKS TO BE PERFORMED THIS WEEK:**

**project phase: 2**

#### **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> | MEETING ORGANIZER:<./HEER>

MEETING ATTENDEES: <./JAY><./JATAN> |

MEETING MODALITY: VIRTUAL/IN PERSON

TASKS TO BE PERFORMED THIS WEEK:

project phase: 3

Task Name(s):

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

Number of hours allocated:

Actual hours put in: < member >

9. 18: Jatan
10. 18: Jay
11. 18: Heer

### **Test Plan**

We need to test if multiple users can connect to server at same time.

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.

If the user deletes any file, then that operation should be successfully completed.

If multiple clients are using the same server, then the file should not get overwritten.

Client should not upload files which are larger than 200MB.