

2232-CSE-6324-002

ADVANCED TOPICS IN SOFTWARE ENGINEERING
PROJECT REPORT

Instructor: Jeff Lei

TEAM 02

Team Members:

Chandu Pavan Budda	1001977117
Kaviya Karthikeyan	1002069064
Shivani Reddy Aleti	1002034487
Varsha Bhavani Musham	1001945743

Table Of Contents

S.No	Title	Page No
1	Project Overview	3
2	Core Functionalities	3
3	File Sync	3
4	Sync Status	4
5	Delta Sync	4
6	Error Handling	5
7	Test Approach	5
8	Screen Shots	6

PROJECT OVERVIEW:

- A simplified cloud-based storage application using Java similar to google drive.
- The main focus of the project is oHere we support one user who can access the storage through multiple clients.
- One user who can access the storage through multiple clients is supported in this project.
- Created as many threads as possible to achieve maximum concurrency.

CORE FUNCTIONALITIES:

- **File Sync:** Files should be synced continuously as changes are detected.
- **Delta Sync:** Only the changed portion of the file needs to be synced, and not the entire file.
- **Sync Status:** Monitor and report the sync status.
- **Error Handling:** Data consistency must be preserved in the event of network failures and other.

FILE SYNC:

Files must be synced continuously as changes are detected.

Multiple files are synced for multiple clients simultaneously.

- **File Upload:**

- File upload is done through UDP as datagram packets of block size 50 KB.
- The client calculates and splits the files into datagram packets based on the file size.
- The server receives the packets and then saves them to a directory on the server's machine.
- The server also generates a hash value for the received file using the SHA-256 algorithm, which is stored in a separate metadata file along with the name of the received file.
- Each folder object contains a list of files stored in the folder, along with their corresponding hash values.
- The server uses the original file name to determine the name of the folder object in the metadata file, and the name of the file object within the folder object.

- **File Download:**

- File Download is done automatically onto the clients as and when changes are made to the document.
- Just as the client maintains a metadata file for the uploaded files the client also does the same and routinely checks if the metadata files match and if the file exists on all the clients.
- If the server detects that there are changes in the metadata file or if the file does not exist on the client it downloads the file or changes on to the client.

- **File Delete:**

- Deletion of a file is done on the client side.
- Deletion should also be done manually on the server side.

SYNC STATUS:

The sync status is monitored and the sync status of each file is reported.

- **Sync start:** The system reports a sync started message when the file is uploaded.
- **Sync Complete:** The system reports a sync complete message when the file is uploaded to the server.
- **Sync Terminated:** The system displays a sync terminated message when the user stops the file upload or in case an error occurs.

DELTA SYNC:

- Delta sync is used to minimise the amount of data that needs to be synced, instead of syncing the entire file, every time.
- The clients and server generate metadata files using SHA 256 algorithm and keep track of them. If there are any changes made to the file by the client the metadata file on the client side changes.
- The server compares the metadata files on the client and server side and when the files don't match, it understands that the client has made changes to the file and syncs only the changed parts of the file.

- The server then checks if the other clients have the updated version of the file by comparing the metadata files and if they don't, it downloads the changed parts onto the client.
- Two threads are used to carry out these tasks, upload and download threads.
- It enables fast synchronization, and is an efficient and secure way, minimizing network bandwidth and resources.

ERROR HANDLING:

- handled unknownHostException and SocketException while accesing local host using InetAddress.
- Handling IOException while sending datagram packets.
- Handling InterruptedException while running thread handling FileNotFoundException exception.
- Handling NoSuchAlgorithmException while generating hashCode, throwing IllegalArgumentException from that method.

TEST APPROACH:

The tests were done manually by devising different scenarios such as:

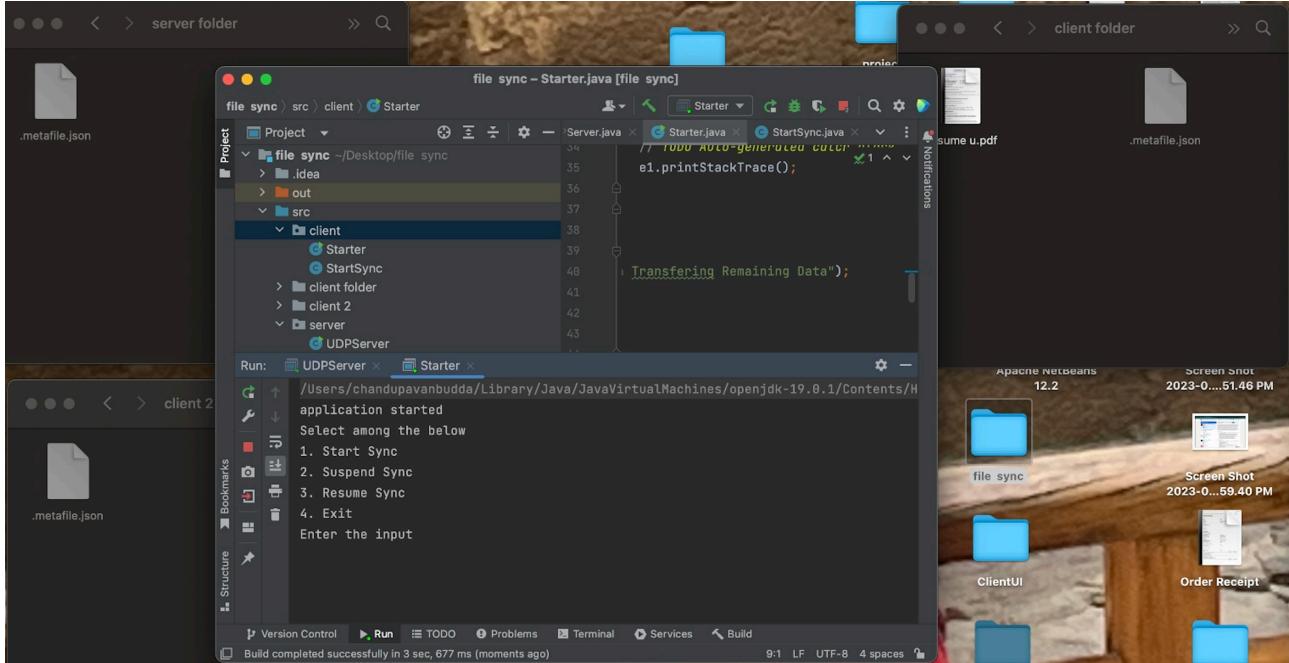
- Files of various sizes were uploaded.
- Files were uploaded from different clients at the same time.
- Multiple files were uploaded from the same client.
- If other clients can see and access the files uploaded by a client.
- Only the modified parts are uploaded.
- If the changes are updated in the server as well as other clients.

CHALLENGES:

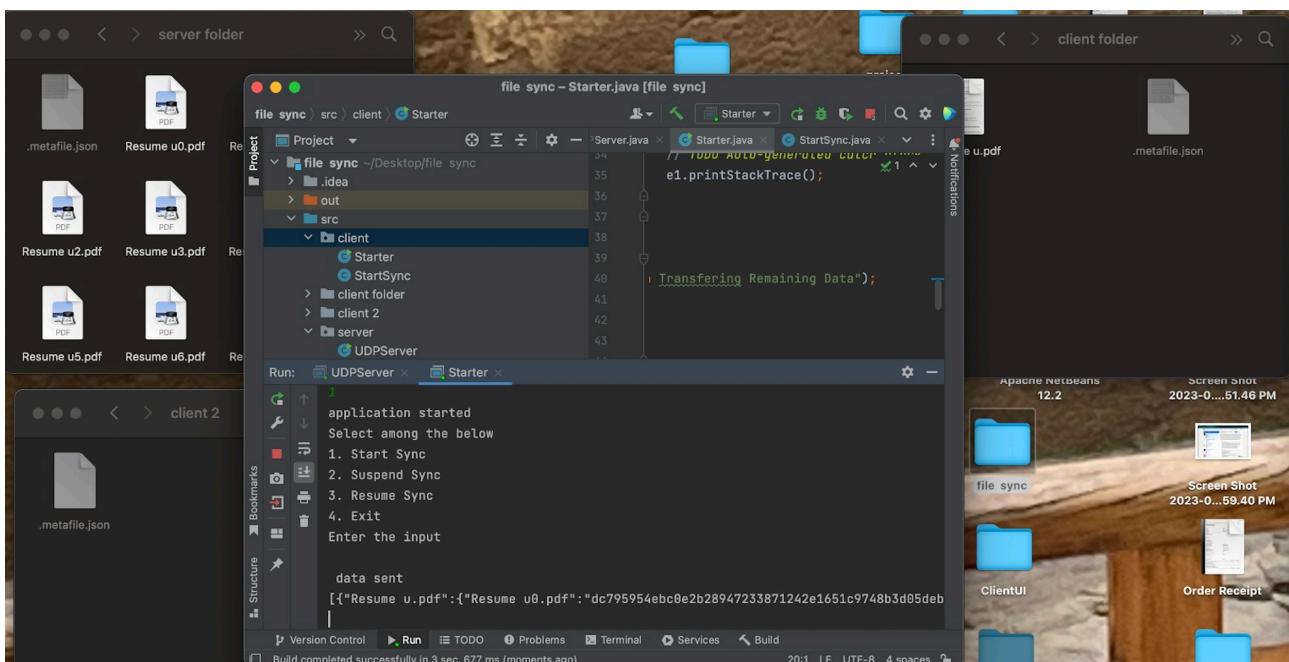
- **Java:** None of the team members were experienced with java which was made it difficult to work on the project with just basic knowledge of java.
- **Delta Sync:** We faced a lot of challenges while trying to figure out how to detect the changes in files and to just upload the changed parts of the file.
- **Data Transmission:** We initially had a 50 byte block size for file transmission but it made the process really slow for large file sizes.

SCREEN SHOTS:

- The image below shows the folder only on the client side before upload.



- The image below shows the syncing of the file with server.



- The image below shows the syncing of the file with the client 2.

