SOFE 4790U: Distributed Systems (Fall 2024)
Instructor: Dr. Ahmed Badr

Assignment #2

Honour code: By submitting this assignment, I Mohammed Fawal, affirm this is my own work, and I have not asked any of my fellow students or others for their source code or solutions to complete this assignment, and I have not offered my source code or solutions for this assignment to any of my fellow students.

Name: Mohammed Fawal

Banner ID#: 100704653

1. Application idea

The File Collaborative System is a distributed application designed to manage files collaboratively across clients while maintaining metadata integrity. This system integrates Socket Programming for file transfers and Java RMI (Remote Method Invocation) for managing file metadata. Its modular design includes three components: common, server, and client, enabling flexibility and scalability. The application allows users to upload files, lock/unlock files, retrieve metadata, track version histories, and ensure consistent file state across collaborative users.

2. Describe the core functionalities (5 unique functions)

1. Upload File - Clients can upload a sample text file using the absolute path.

2. Version History - Update metadata multiple times and retrieved the version history for the file.

3. Unlock File - Lock a file using its name to prevent concurrent modification.

4. Update Metadata - Prompts client to enter file name to update metadata

5. Get file Metadata - Query metadata info for an uploaded file using its name.

3. Challenges and solutions

1. **Challenge**: Ensuring File Transfer Reliability
   Solution: Implemented error handling in the socket communication layer, ensuring proper retries for incomplete transfers.

2. **Challenge**: Metadata Consistency Across Multiple Clients
   Solution: Leveraged RMI to maintain a single source of truth for metadata, ensuring updates are synchronized.

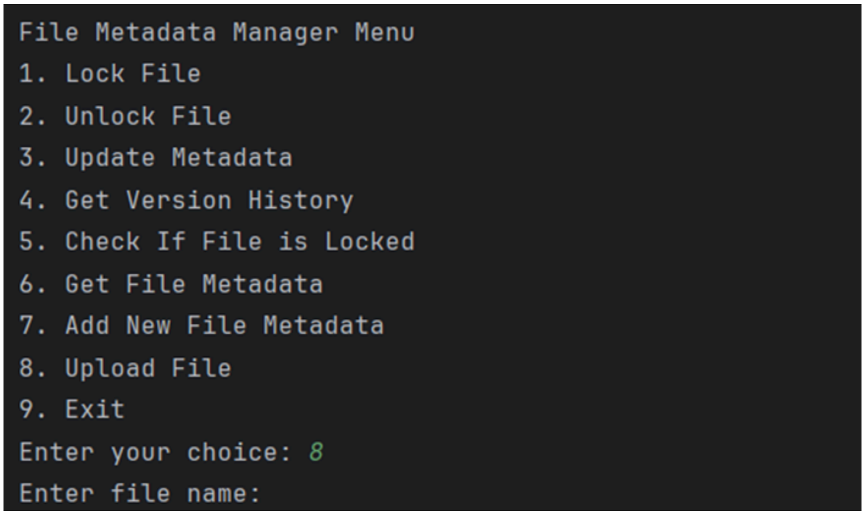
3. **Challenge**: Simulating Concurrent Access
   Solution: Tested with multiple client instances to verify proper locking mechanisms and data integrity.

4. **Challenge**: Usability
   Solution: Designed a user-friendly menu-driven interface for CLI and extended it with a web-based JSP UI for broader accessibility.

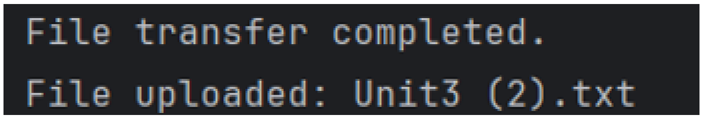
4. Testing

1. File Transfer Test

   ○ Action: Uploaded a sample text file using the absolute path.
   ○ Verification: Verified the file was successfully received on the server and stored in the designated location.



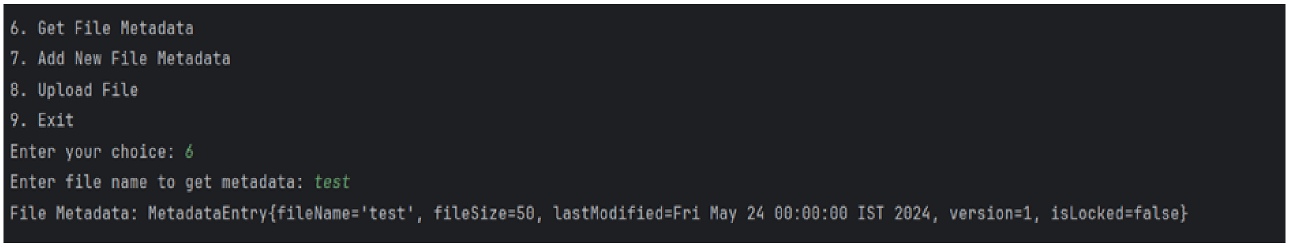
```
File Metadata Manager Menu
1. Lock File
2. Unlock File
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 8
Enter file name:
```



```
File transfer completed.
File uploaded: Unit3 (2).txt
```

2. Metadata Retrieval Test

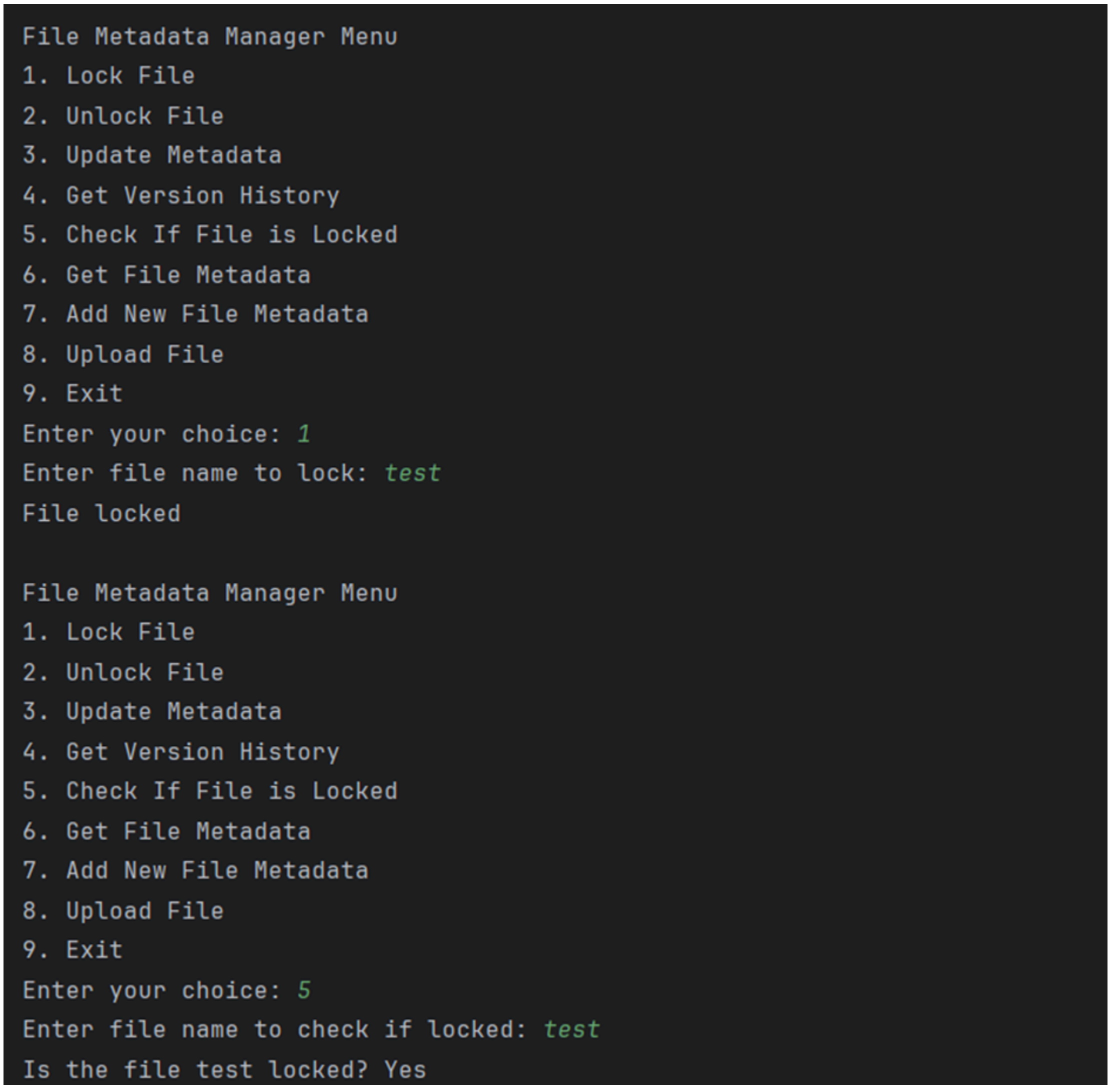
   ○ Action: Queried metadata for an uploaded file using its name.
   ○ Verification: Retrieved file details such as size, version, lock status, and last modified date.



```
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 6
Enter file name to get metadata: test
File Metadata: MetadataEntry{fileName='test', fileSize=50, lastModified=Fri May 24 00:00:00 IST 2024, version=1, isLocked=false}
```

3. File Locking Test

- ○ Action: Locked a file using its name to prevent concurrent modification.
- ○ Verification: Confirmed that the file was locked and attempts to upload or edit were restricted.

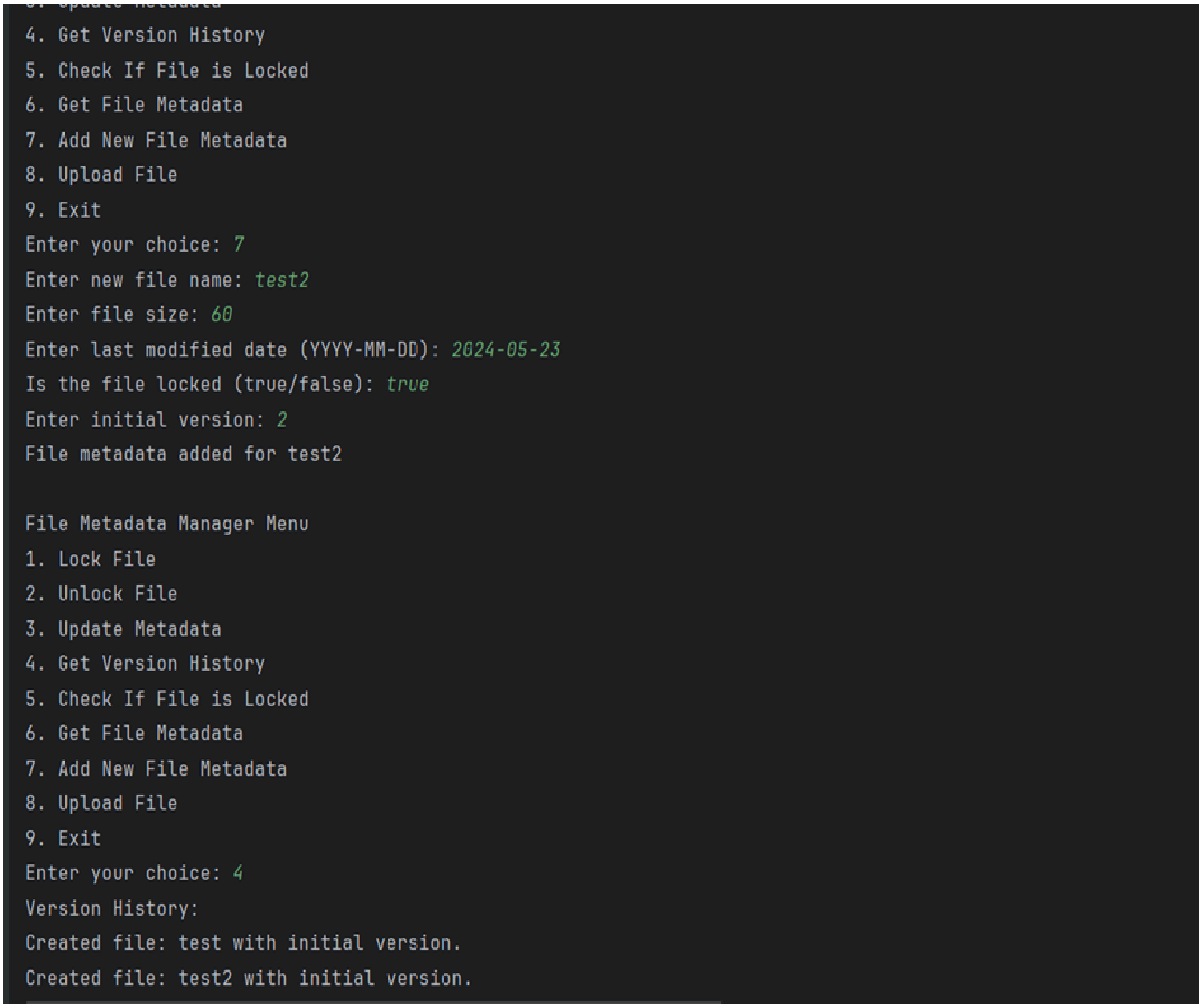


```
File Metadata Manager Menu
1. Lock File
2. Unlock File
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 1
Enter file name to lock: test
File locked

File Metadata Manager Menu
1. Lock File
2. Unlock File
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 5
Enter file name to check if locked: test
Is the file test locked? Yes
```

4. Version History Test

- ○ Action: Updated metadata multiple times and retrieved the version history for the file.
- ○ Verification: Confirmed the correct tracking of all changes in the version history.

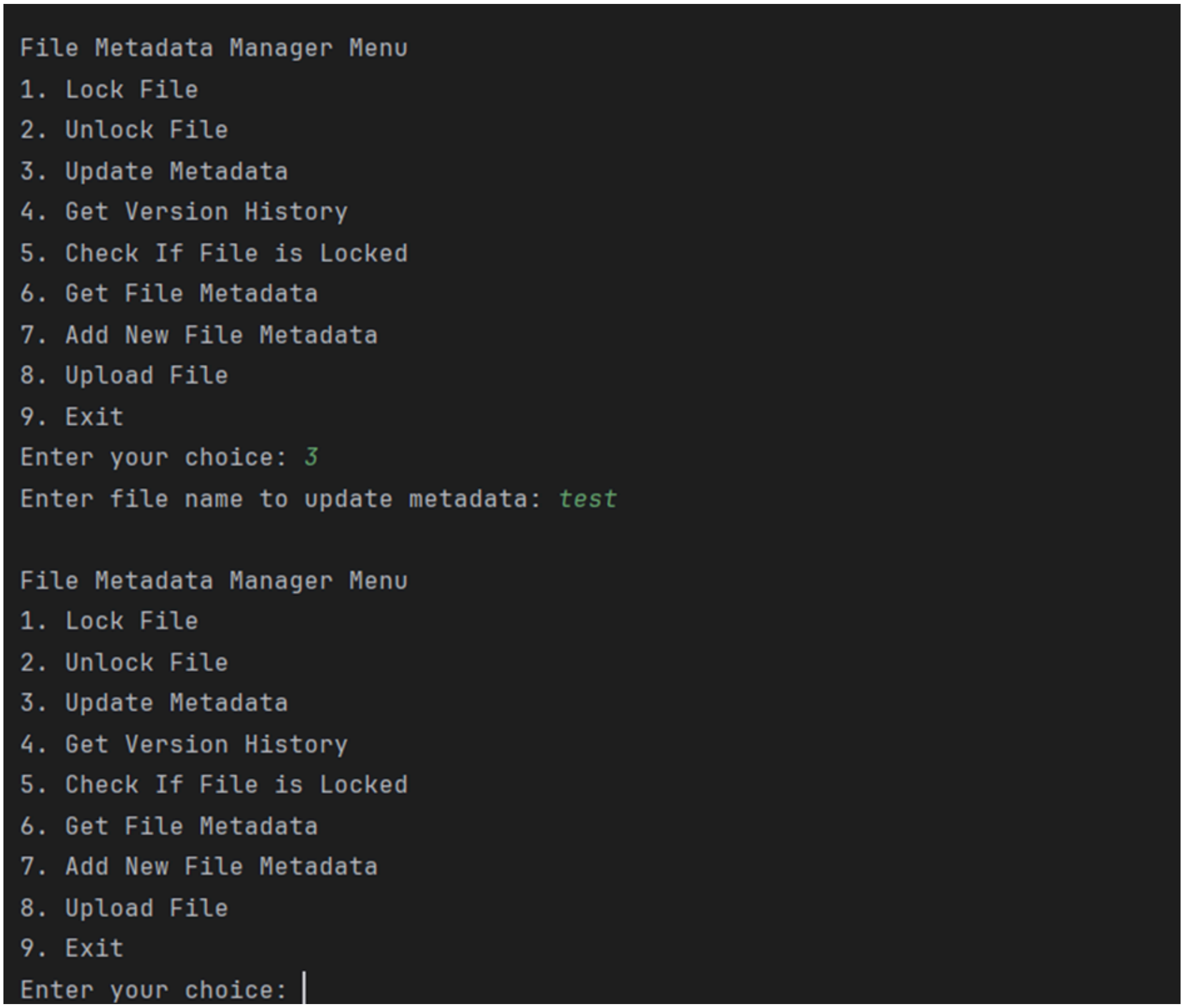


```
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 7
Enter new file name: test2
Enter file size: 60
Enter last modified date (YYYY-MM-DD): 2024-05-23
Is the file locked (true/false): true
Enter initial version: 2
File metadata added for test2

File Metadata Manager Menu
1. Lock File
2. Unlock File
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 4
Version History:
Created file: test with initial version.
Created file: test2 with initial version.
```

5. Concurrent Access Test

- Action: Simulated multiple clients trying to access and edit a file simultaneously.
- Verification: Verified that locked files were inaccessible for editing and changes were properly synchronized for unlocked files.

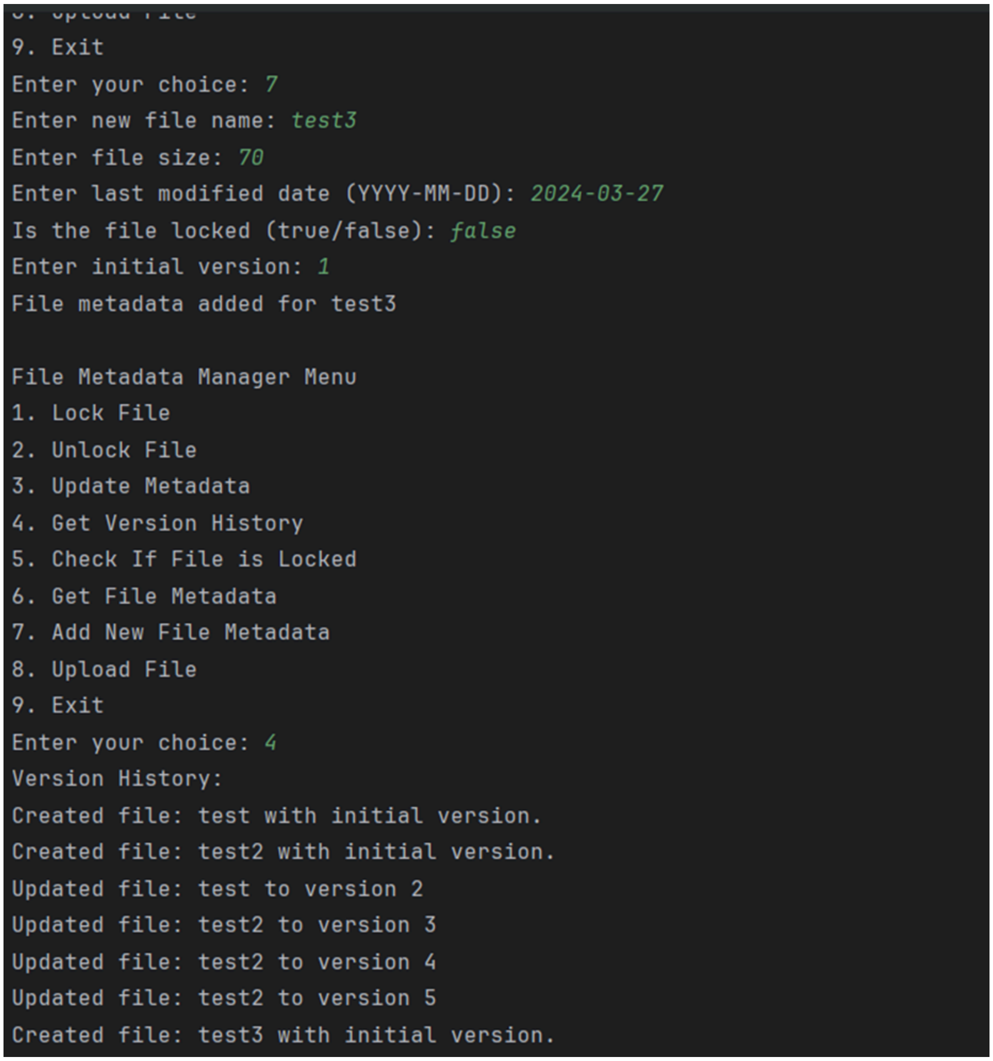


```
File Metadata Manager Menu
1. Lock File
2. Unlock File
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 3
Enter file name to update metadata: test

File Metadata Manager Menu
1. Lock File
2. Unlock File
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 
```

6. File Upload and Metadata Update Test

- Action: Uploaded a new file and added corresponding metadata.
- Verification: Retrieved and verified metadata details after the upload to ensure consistency.



```
8. Upload File
9. Exit
Enter your choice: 7
Enter new file name: test3
Enter file size: 70
Enter last modified date (YYYY-MM-DD): 2024-03-27
Is the file locked (true/false): false
Enter initial version: 1
File metadata added for test3

File Metadata Manager Menu
1. Lock File
2. Unlock File
3. Update Metadata
4. Get Version History
5. Check If File is Locked
6. Get File Metadata
7. Add New File Metadata
8. Upload File
9. Exit
Enter your choice: 4
Version History:
Created file: test with initial version.
Created file: test2 with initial version.
Updated file: test to version 2
Updated file: test2 to version 3
Updated file: test2 to version 4
Updated file: test2 to version 5
Created file: test3 with initial version.
```

Conclusion:

The File Collaborative System successfully integrates file transfer and metadata management in a distributed environment. The tests validate its reliability, scalability, and usability for collaborative file handling. Novel features like metadata management via RMI and collaborative locking ensure consistency and conflict resolution, making this system a robust solution for real-world scenarios

Github link: https://github.com/mohammedfawal/Assigment2_Distributed

Google Drive Link:
https://drive.google.com/file/d/1nLk9MUHSGE6cVLrbgbiGckEhJs3I_i5h/view?usp=drive_link