Collaborative Document Editing System

Submitted by: Zubair Khan(BSCS-36) and Alina Khan(BSCS-42)

Date: January 24, 2024

Course: Computer Network

Submitted to: Dr Arshad Farhad

A report submitted in partial fulfillment of the requirements for the course

Department of Computer Science

Namal University Mianwali

Contents

List of Figures

1 Abstract

This report outlines the development of a Collaborative Document Editing System tailored to enable real-time concurrent editing among multiple users utilizing distinct PCs (server and client). The project addresses the fundamental challenge of fostering seamless collaboration across a LAN network by allowing users to collectively edit documents concurrently, swiftly disseminating modifications to all collaborating members. The system's core functionalities include the integration of robust version control mechanisms, ensuring meticulous tracking of document modifications and enabling users to revisit prior editing sessions. Moreover, the platform is equipped with conflict resolution capabilities to adeptly manage simultaneous edits by multiple users. Additionally, an embedded synchronized chat system serves as a communication conduit during collaborative editing sessions, prioritizing user-friendliness to cater especially to beginners venturing into collaborative systems. Ultimately, this initiative aims to establish a secure, intuitive, and efficient environment, fostering effective real-time collaborative document editing while emphasizing collaborative integrity and enhancing overall user engagement.

2 Introduction

Collaborative document editing refers to a concept where a group of users can simultaneously work on the same set of documents. In the past, when users wanted to share a file, they used to send an entire copy via email or some internal communication channels. When another user wants to edit the document, the user downloads and edits the copy locally and multicasts the edited copy to the interested group via some communication channel. During this process, there might be another user who is working on the previous copy and now received a new copy, the user must manually figure out the changes performed by the previous user and resolve collisions, if any and send back the edited copy. This process is slow and time-consuming. Collaborative document editing tools are a solution to this problem by allowing multiple users to access the document and make edits to the document, so the other users can see the changes instantly. We built a platform, the collaborative document editing system, which is a real-time document editor. It is a Tk inter based project which allows multiple users to simultaneously view and edit document from concurrently. The Collaborative Document Editing System is a network-based application designed to facilitate real-time collaboration among users for editing documents and exchanging messages. The system allows multiple users to connect to a central server, log in, edit a shared document concurrently, and exchange messages in a collaborative environment. A client after login can make edit and can see the updated what other clients made an updated and also can see the entire history who has updated document. A chat option to send messages and also clients can see the updates of all the messages.

3 Problem statement

We intend to develop and design a system in which two or more PCs (client and server) on LAN network can communicate on real time covering Collaborative Document editing, uses can edit document concurrently and can see the history of all the updates and system also facilitates users to enjoy chat during collaboration.

4 Related work

Different working on it has been happened in past. Collaborative systems like Google docs[5] and Notion[6] are well known in its domain. But this project itself is a collaborating document editing system which will made on the intention of keeping it simple and user friendly so that it will be more interactive for beginners who want to explore and work on collaborative system.

5 Background literature

Collaborative editing research and tools developed in the past have followed client-server architecture. In the context of collaborative document editing, client-server architecture can be viewed as multiple clients and a server holding a copy of the document. Whenever any client modifies its local copy, it updates the server's copy, and the server then sends the updates to the remaining clients accessing the shared document.

6 Objectives

The primary objectives of the Collaborative Document Editing System are:

- To understand the role of socket in networking.
- Enable multiple users to log in securely.
- Facilitate real-time collaborative editing of a shared document.
- Maintain an edit history for the shared document.
- Allow users to exchange messages in a shared conversation.

7 Components

7.1 Server

- The server is responsible for handling connections from multiple clients.
- It manages user authentication, ensuring secure access to the system.

- The server maintains the shared document content and edit history.
- Real-time updates are sent to clients when edits occur.

7.2 Client

- Clients can log in with a username and password.
- The main interface provides options to edit the document, view edit history, and send messages.
- Edits made by one client are immediately reflected in the shared document for all clients.
- Users can view the edit history to see changes made by others.
- A messaging feature allows users to communicate within the system.

8 Tools and technologies

8.1 python[4]

For implementation

8.2 Customtkinter[2]

For GUI

8.3 Socket[1]

For networking

8.4 Threading[3]

To Achieve concurrency

9 Architecture and dataflow

The system follows a client-server architecture. The server is responsible for handling connections, user authentication, and distributing document updates and messages to connected clients. Clients, on the other hand, interact with the server, authenticate users, and manage the collaborative editing and messaging functionalities

10 Data Flow Diagram

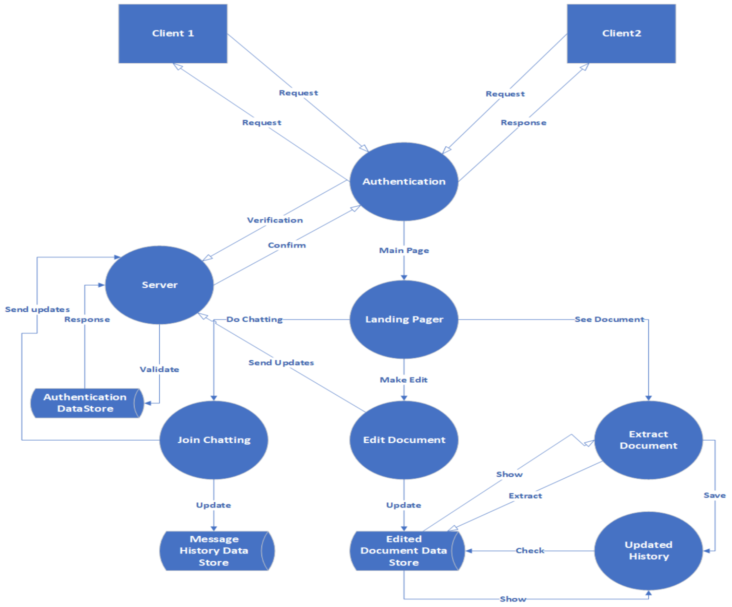


Figure 1: DFD of Collaborative document editing system

11 System Sequence Diagram of Login

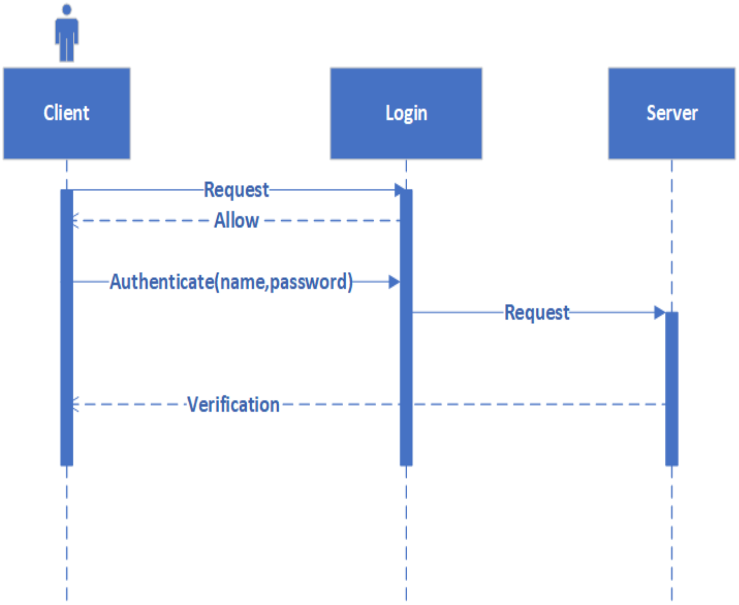


Figure 2: Login SSD of Collaborative document editing system

12 System Sequence Diagram of Document Editing

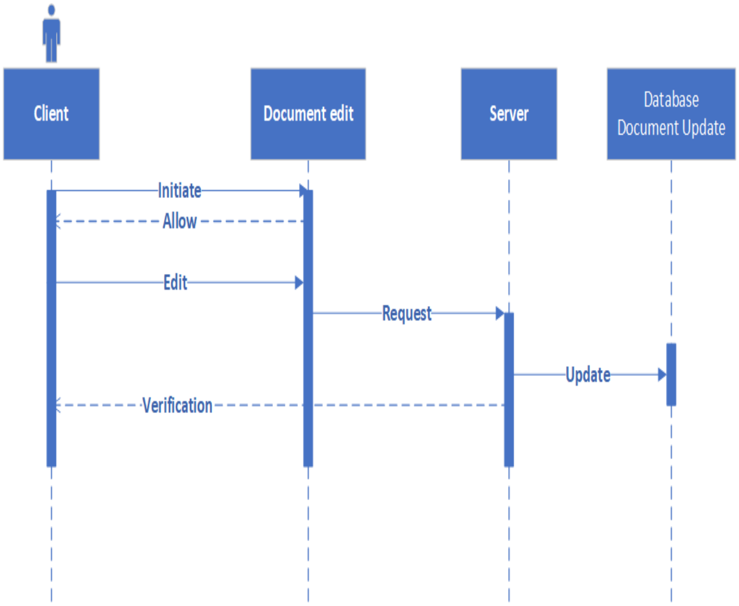


Figure 3: Document editing SSD of Collaborative document editing system

13 System Sequence Diagram of Message

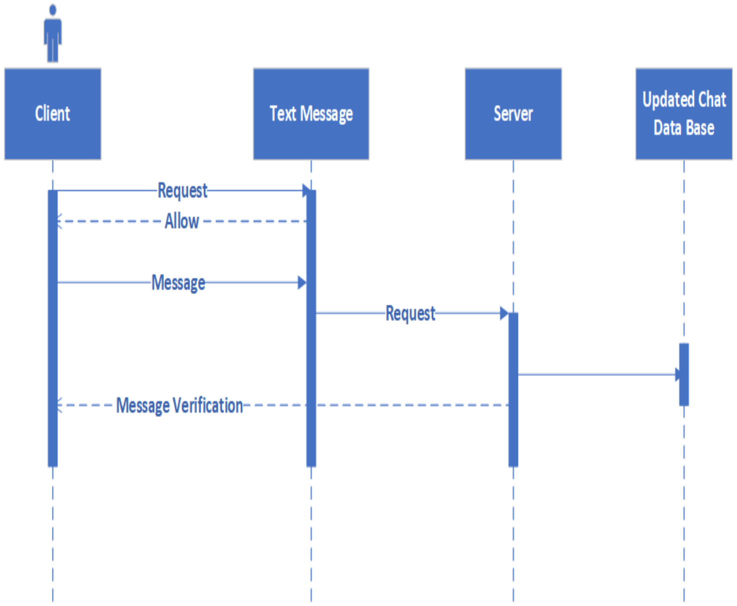


Figure 4: Message SSD of Collaborative document editing system

14 Flowchart

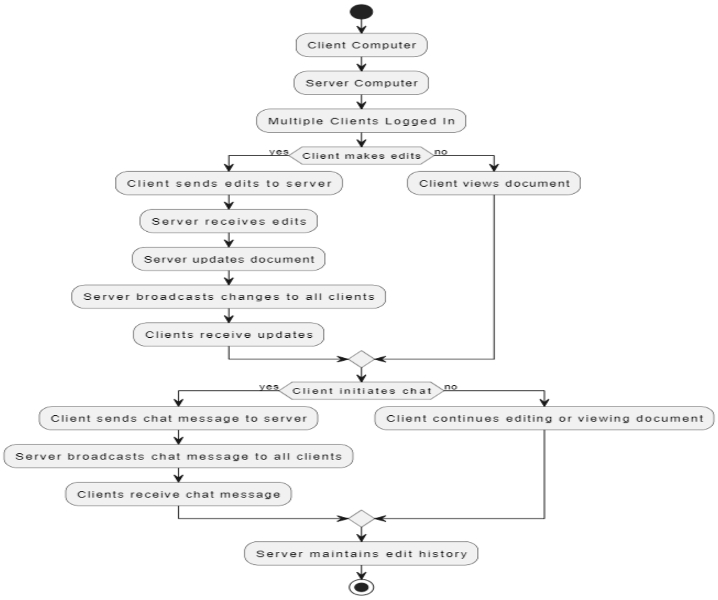

Client Computer
Server Computer
Multiple Clients Logged In
yes
Client makes edits
no
Client sends edits to server
Client views document
Server receives edits
Server updates document
Server broadcasts changes to all clients
Clients receive updates
yes
Client initiates chat
no
Client sends chat message to server
Client continues editing or viewing document
Server broadcasts chat message to all clients
Clients receive chat message
Server maintains edit history

Figure 5: Flowchart of Collaborative document editing system

15 Summary of methodology

- First, we define and analyze the objectives of a collaborating document editing system and research and gather all the requirements from the potential users according to their needs.
- Study existing collaborative systems and necessary networking protocols for real-time collaboration.
- We will use python3[4] in which we will use OOP (classes objects) and socket[1] for client-server architecture concepts.
- A version control to keep track of changes made in a document over a specific time
- A server that will listen for connections and manage the collaborative document editing.
- Used customtkinter[2] for the front end for a user-friendly interface.
- A login page which allows users to interact with the main functionalities after entering their credentials.

- A first window after login which allows users to choose their option.
- A document editing window for user editing covering functionalities of showing history, making edits, seeing updated documents as well as chatting options.
- Develop a window that will show document updates.
- Another window to see the history of updates.
- A message window that allows the user to text a message using TCP/IP or IGMP[7] protocol.
- This window facilitates users to see updated messages on a new window.

15.1 Login Interface

A login interface will be displayed to all the clients for their interaction with the system. It allows clients to enter their name and password to successfully log in. The server authenticates clients, and it authenticates them according to the authentication database.

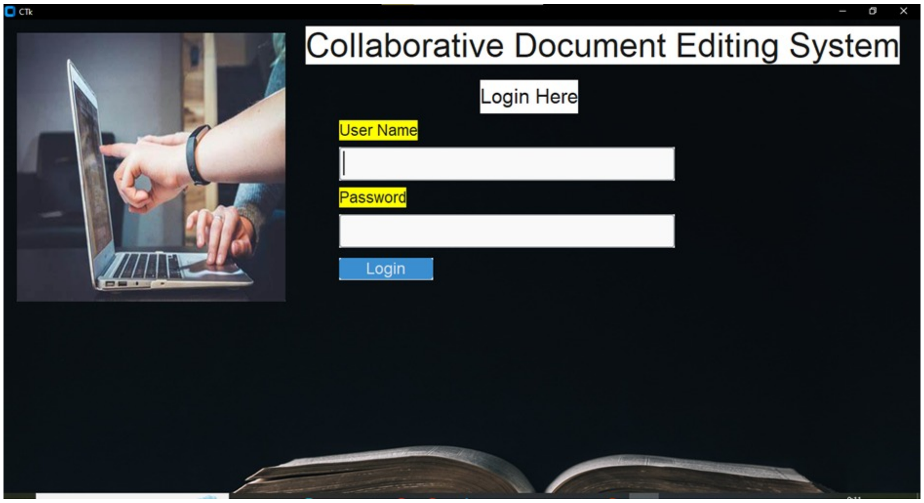


Figure 6: Login Window

15.2 user option selection window

After successfully logging in clients have options to choose according to his interest edit document or go with the conversation.

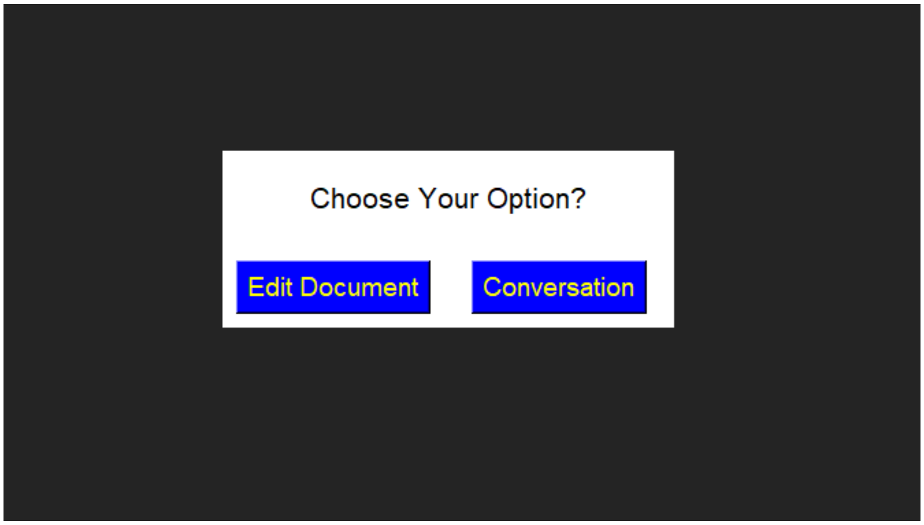


Figure 7: Options provided to user

15.3 Document editing window

In the document editing window, clients have options to edit the document, see all the updates in the document, and also can see the history of editing (who has edited) and go in the message window too.

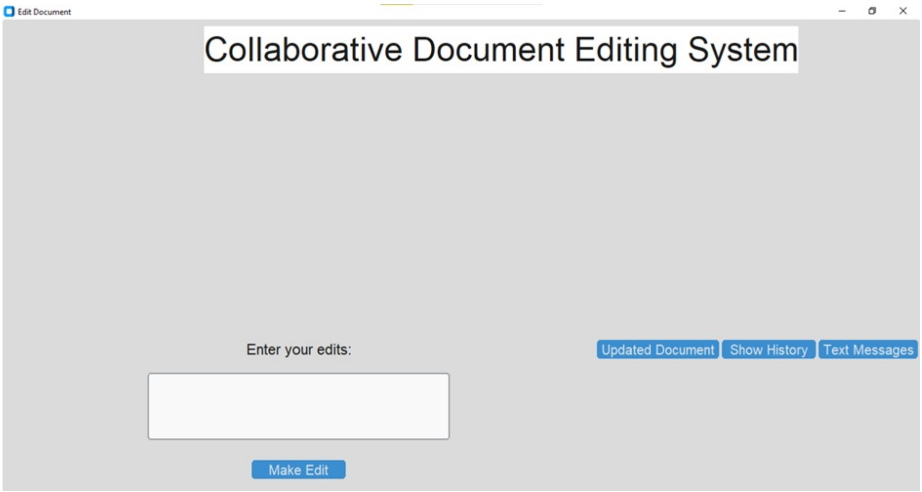


Figure 8: Document editing

15.4 Chat window

A chat window in which clients can send messages and also can see that entire chatting who text messages etc.

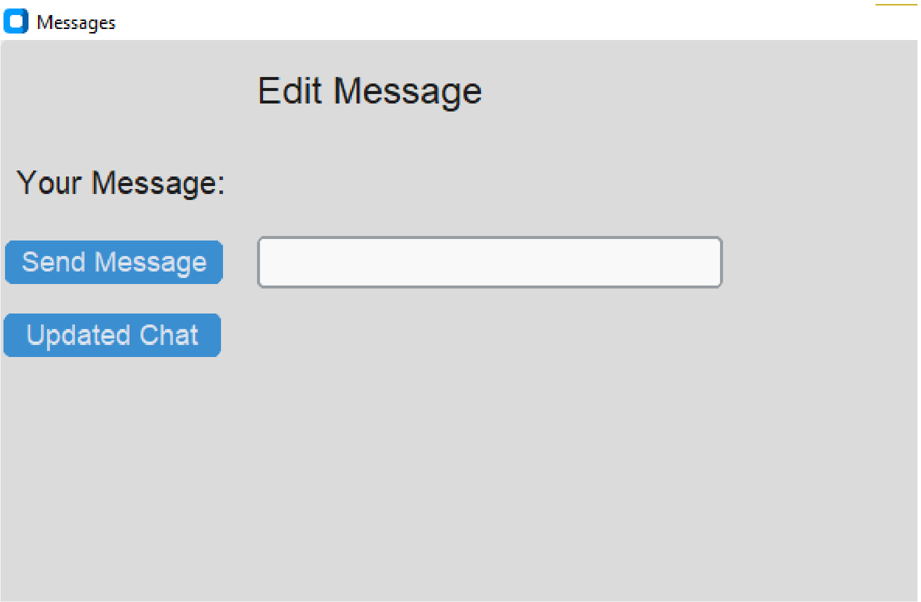


Figure 9: Chating of client

16 System working

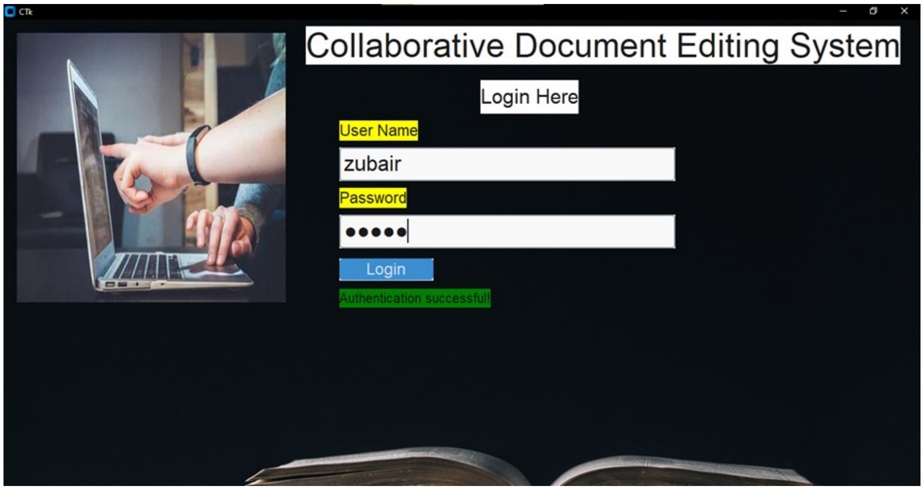
Some snapshots are mentioned to see how the system works.

16.1 Client login

A basic working has been shown how the system works in the case of two clients.

16.1.1 First client login

1st client enters its name and password. These details are sent to the server which confirms the authentication status and acts accordingly.



User Name
zubair

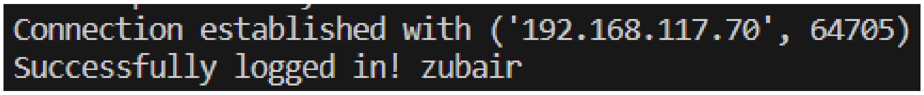
Password
•••••

Authentication successful!

Figure 10: 1st client login

16.1.2 Server Response for Client 01

Here is server response for client one.

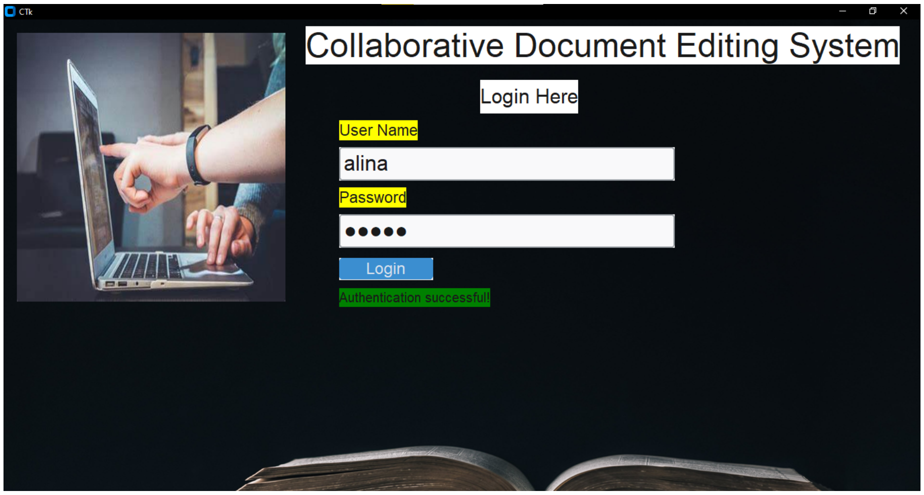


```
Connection established with ('192.168.117.70', 64705)
Successfully logged in! zubair
```

Figure 11: Server response for 1st client

16.1.3 Second client login

The 2nd client enters its name and password. These details are sent to a server which confirms authentication status and acts accordingly.



Collaborative Document Editing System

Login Here

User Name

alina

Password

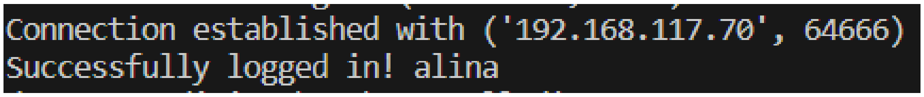
●●●●●

Authentication successful!

Figure 12: 2nd client login

16.1.4 Server Response for Client 02

Here is the server response for the second client .



```
Connection established with ('192.168.117.70', 64666)
Successfully logged in! alina
```

Figure 13: Server response for 2nd client

16.2 Document editing

The client has the following options.

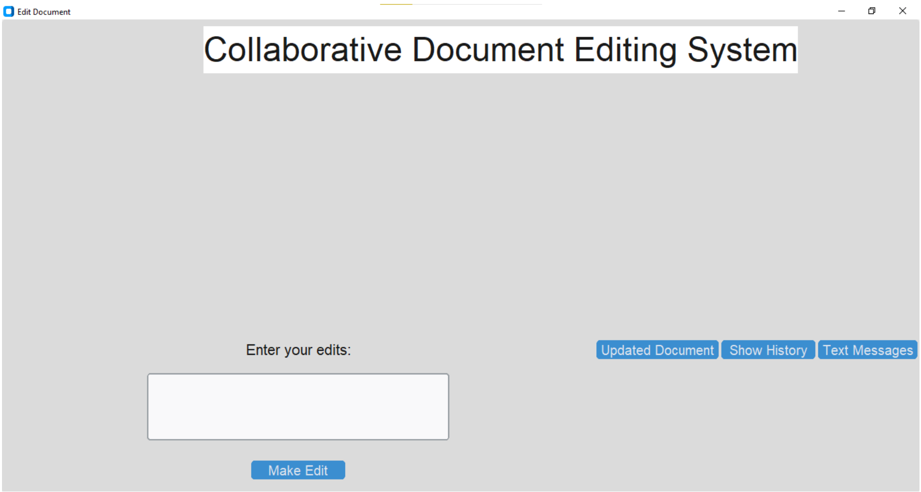
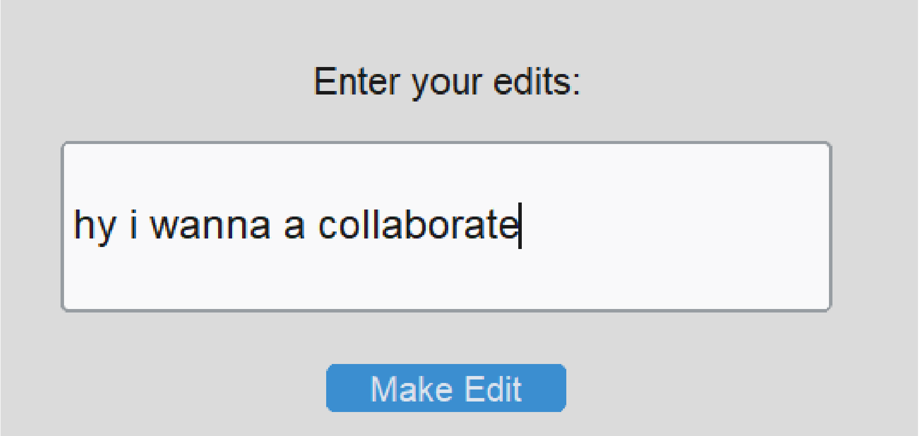


Figure 14: Document editing option

16.2.1 First client editing

1st client has made some edits in the document.

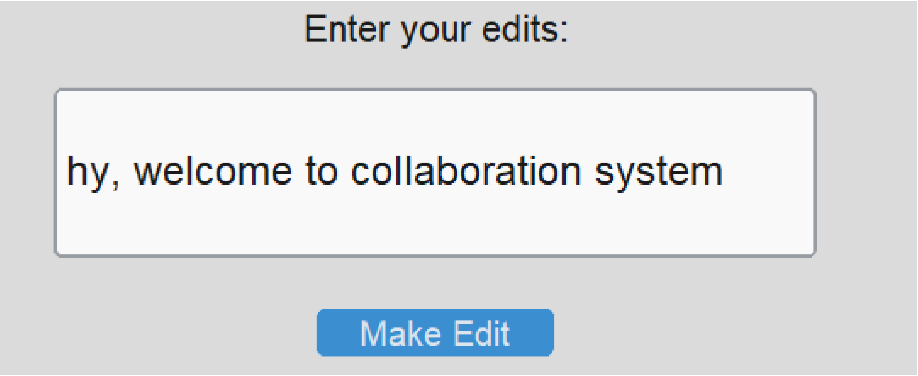


hy i wanna a collaborate

Figure 15: 1st Client Editing

16.2.2 Second client editing

2nd client has also made some edits in the document. Here is the snapshot of the second client editing in the document.

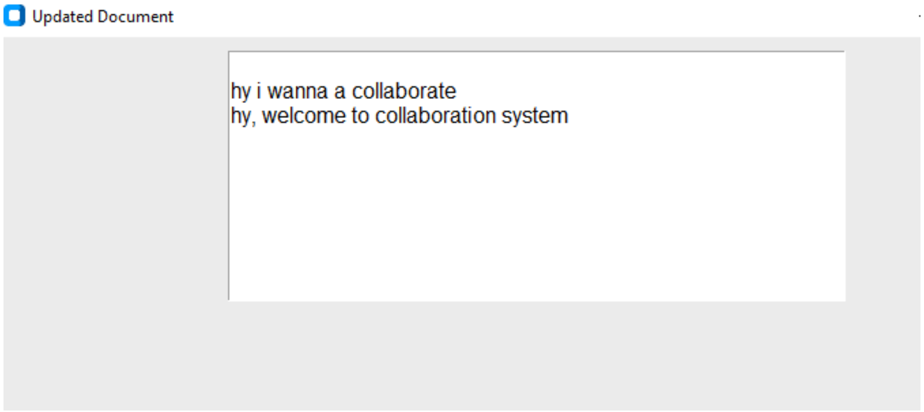


hy, welcome to collaboration system

Figure 16: 2nd Client Editing

16.3 Updated document

Now if any client wants to see the updated document, it will click on the updated document and then it can see all the editing like that.

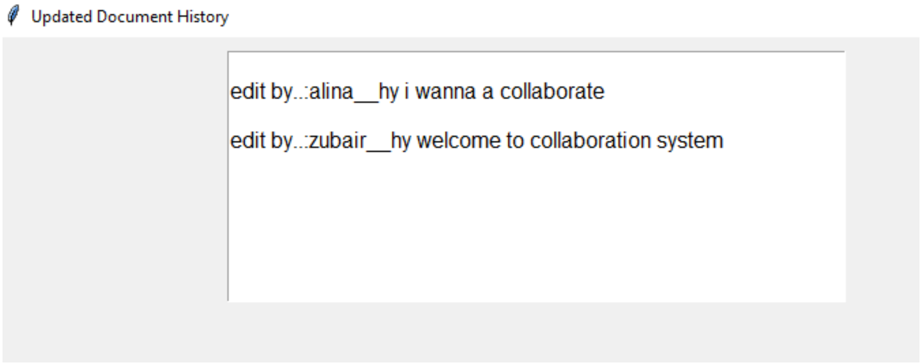


hy i wanna a collaborate
hy, welcome to collaboration system

Figure 17: Updated Document

16.4 History

To see the complete history of who has updated the document and what he has updated the history button will display all the details like that.



edit by..:alina__hy i wanna a collaborate

edit by..:zubair__hy welcome to collaboration system

Figure 18: History

16.5 Chatting

The following figure shows how two clients can text messages to each other. Here are possible options for a client.

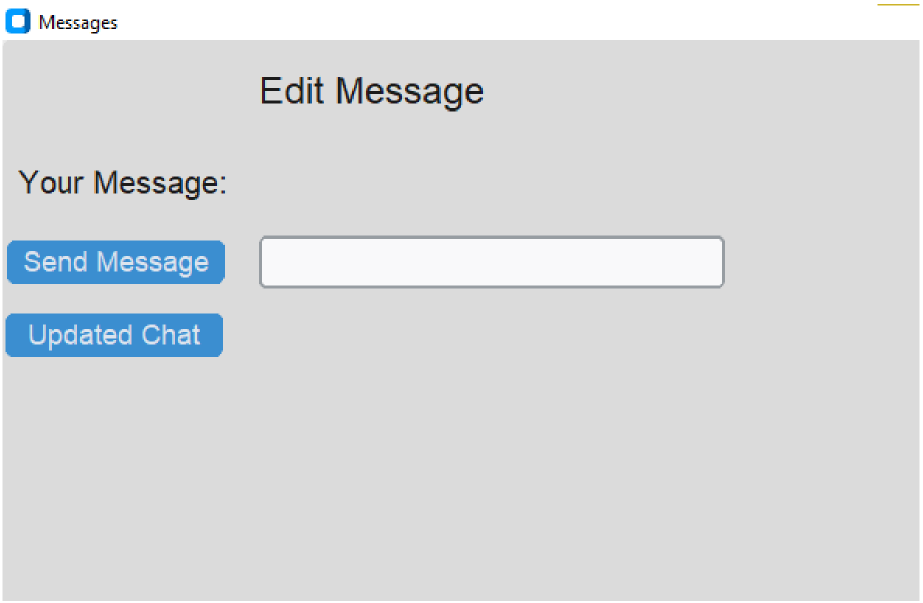
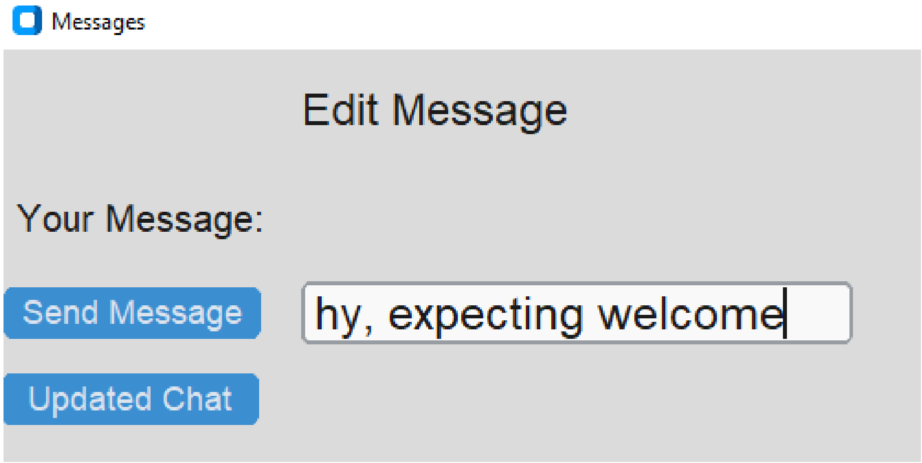


Figure 19: Message Editing Window

16.5.1 First client message

A client after typing a message, clicks on send message, it will be sent to all clients which are collaborating.



Edit Message

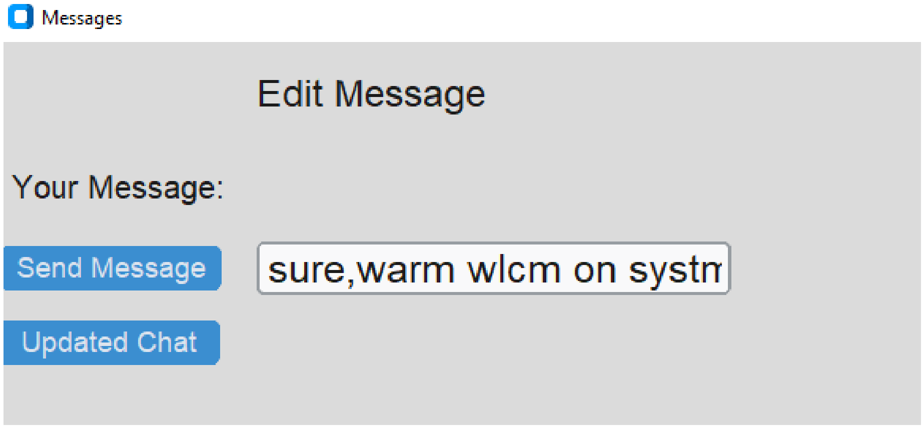
Your Message:

hy, expecting welcome

Figure 20: 1st Client message

16.5.2 Second client message

2nd client has also texted a message.



Edit Message

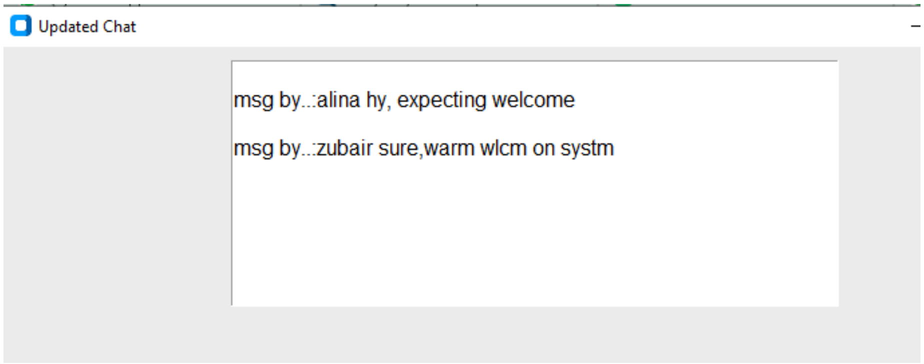
Your Message:

sure,warm wlcm on systm

Figure 21: 2nd Client message

16.6 Message updated

Now any users can see the updated chat history, who texted, and what texted.



msg by..:alina hy, expecting welcome

msg by..:zubair sure,warm wlcm on systm

Figure 22: Updated Chat

17 Implementation

17.1 Server side implementation

The server implementation includes:

17.1.1 Socket[1] initialization

Creates and binds a socket for listening to incoming connections.

17.1.2 user authentication

Authenticates users based on predefined credentials.

17.1.3 Document editing

Manages shared document editing, history, and updates.

17.1.4 Message exchange

Handles the exchange of messages between connected clients.

17.2 Client side implementation

The client implementation includes:

17.2.1 Graphical user interface

Utilizes ‘customTkinter’[2] to create an interactive GUI for user interaction.

17.2.2 user authentication

Sends user credentials to the server for authentication.

17.2.3 Document editing

Allows users to make edits to the shared document.

17.2.4 Message exchange

Enables users to send and receive messages.

18 User interface

18.1 Login screen

- A secure login screen where users enter their credentials.
- User authentication against the server's records.

18.2 Document editing interface

- A collaborative editing interface where users can make real-time edits to the shared document.
- Edit history display to visualize the sequence of changes made by each user.
- Options to update the shared document and view detailed edit history.

18.3 Messaging interface

- A dedicated space for users to exchange messages.
- Real-time message updates within the system

19 Testing

The system has undergone extensive testing to ensure reliability and functionality. Testing includes:

19.1 Unit testing

Individual components, such as user authentication and document editing, were tested independently.

19.2 Integration testing

Ensured that all components work seamlessly together.

19.3 User acceptance testing

Real users were involved in testing to ensure the system meets user expectations.

20 Performance evaluation

20.1 Scalability

- Evaluate the system's ability to handle an increasing number of users concurrently.
- Assess the server's performance under varying loads.

20.2 Responce time

- Measure the response time for document updates and messaging.

20.3 Error handling

- Implement robust error handling mechanisms to gracefully handle unexpected scenarios.
- Monitor error logs and address issues promptly.

20.4 Documentation and support

- Maintain comprehensive documentation for users and developers.
- Provide responsive support to address user queries and issues.

21 Future predictions

21.1 User$_{b}aseexpansion$

- Anticipate an increase in the user base as more users recognize the benefits of collaborative document editing.
- Plan for scalability to accommodate a larger number of simultaneous users.

21.2 Feature enhancement

- Integrate additional features based on user feedback and evolving needs
- Consider implementing version control, advanced document formatting, and collaborative editing tools.

21.3 Security measures

- Enhance security features to protect user data and documents.
- Implement encryption for data transmission and storage to ensure confidentiality.
- Integration with Cloud Services:

21.4 Integration with cloud service

- Explore integration with cloud services for document storage and synchronization.
- Enable users to access and edit documents from different devices seamlessly.

21.5 User role and permissions

- Implement user roles and permissions to control access to specific document features.
- Allow customization based on user responsibilities within collaborative projects. Real-time Collaboration Tools:
- Integrate real-time collaboration tools such as comments, and presence indicators.
- Enhance the user experience for synchronous collaboration.

22 Conclusion

The Collaborative Document Editing System provides a robust platform for users to collaboratively edit documents and exchange messages in real time. The system's server-client architecture allows for scalability, and its intuitive GUI ensures user-friendly interaction.

23 References

socket [Online]. Available: `https://socket.io/docs/v4/`. [Accessed on January 21, 2024].

customtkinter [Online]. Available: `https://pypi.org/project/customtkinter/0.3/`. [Accessed on January 21, 2024].

threading [Online]. Available: `https://realpython.com/intro-to-python-threading/`. [Accessed on January 21, 2024].

python [Online]. Available: `https://www.python.org/`. [Accessed on January 21, 2024]. https://docs.google.com/document/u/0/

googledocs [Online]. Available: https://docs.google.com/document/u/0/. [Accessed on January 21, 2024].

notion [Online]. Available: https://www.notion.so/. [Accessed on January 21, 2024].

igmp [Online]. Available: https://www.ibm.com/docs/en/powerha-aix/7.2?topic=multicasting-internet-group-management-protocol. [Accessed on January 21, 2024].