CMPS211 Advanced Programming Techniques

Collaborative Plain Text Editor

Overview

This project requires you to design and implement a simplified real-time collaborative plain text editor in Java. This application will allow multiple users to simultaneously edit the same document through shareable codes. The aim of this project is to:

- Develop a functional collaborative plain text editor with real-time editing and cursor tracking.
- Practice concepts you learned in the course

Specification

Document & Collaboration Management [20%]

- Import/Export files: Users can import/export txt files from/to their devices. You should preserve line breaks.
- Sharing: Users can request sharable codes from the application server. Each file has 2 codes: one code is dedicated to editors and another is dedicated to read-only collaborators. Collaboration sessions start by sharing the codes with other users (over whatsapp, email, phone, etc…).

Real-time Collaborative Editing [60%]

- Character-Based Editing: Support character-by-character insertion and deletion. Support pasting text as multiple insertions.
- Concurrent Edits: Implement a tree-based Conflict-free Replicated Data Type (CRDT) algorithm to handle concurrency issues and conflicts due to multiple edits happening at the same time by different users.
- Real-time Updates: Send/receive edits to/from other users in real-time. Edits are relayed between users through a central server.
- Cursor Tracking & User Presence : Update the cursor positions of other users in the editor. Update the list of users who are currently present in the collaboration session.
- Undo/Redo Changes: Users can undo/redo their latest actions. The user cannot undo/redo changes done by other users. Your program should be able to undo/redo at least the latest 3 operations.

UI [20%]

Design a user-friendly user interface that includes the following:

- Text Area: Display a text area that shows the document and allows text editing.
- Sharing Option: Display the shareable codes to the user.
- Collaboration: Allow the user to join a collaboration session.
- Import/Export Options: Display a menu to allow the user to import/export files.
- Permission Handling: Viewer users are not allowed to edit in the text area. Viewers cannot also view the sharable codes.
- Cursor Display & User Presence: Visually represent the cursors of other users in the editor. Cursors should be visually distinct (e.g., different colors). Display a list of active users. To restrict the cursor colors, you may assume when designing the UI that the maximum number of concurrent editors per document is 4.

Bonus [6%]

- User comments: Editors can add comments to parts of the text. Comment should be removed if text is removed.
- Reconnection: Allow users a period of 5 minutes to reconnect if they experience a network drop. Remote edits missed should be sent to the user on reconnection. Local changes done should be sent after reconnection.

Implementation and Deliverables

Teams & Deadlines

All teams will deliver the project to the course classroom on **Monday (5th of May) at 2 pm.** Each team will discuss the project on the scheduled day and time.

Provide a zipped folder named **Team_<team number>.zip** containing the following materials:

- Your code files
- A readme.txt, explaining how to run your code
- A members.txt containing the names and IDs of each student in the group

Work in groups of 4 members.

Implementation

- Java should be used for all logic.
- **Concurrency:** Multiple users should be able to edit the same document concurrently, and multiple documents can be active at once. The system must handle these concurrent edits in a non-blocking manner to maintain responsiveness.
- Use any framework for UI as long as the logic and data handling is in Java. Fancy UI is not required and it is not a bonus.
- It is your responsibility to select the best data structures, algorithms, techniques and tools that enhance the project performance
- You should handle network problems and exceptions in your program (connection errors, wrongly formatted messages..etc.)

Libraries and Packages Regulations

You could use a library only if you followed **all** of the points below:

- You can use libraries for UI, networking, JSON handling, html handling, ID generation, and data persistence.
- The core CRDT implementation (data structures and the logic) **must** be your own code.
- You are responsible for the library accuracy. (If it does a bad job, then it is your responsibility).
- You should understand how the library works
- If you are not sure if library usage is allowed or not, contact the instructors.

Evaluation and Grading Criteria

- The project is graded as a whole and the discussion decides what is the grade of each student (there's no piggybacking)

CMPS211 Advanced Programming Techniques

Collaborative Plain Text Editor

- Any delay in delivery will be penalized by losing 10% of the grade for each late day
- Code must be original, and must not be copied or shared from any other source, except as provided by the class instructors.
- Note: A plagiarized project means ZERO in the project and deduct some grades from the other coursework.

~~~~~ Good Luck ~~~~~
~~~~~