Development Track

Team Member and Co-ordinator: Mukta Jaiswal (muktaj2@illinois.edu)

Title: Email Inbox Organizer

Functions and Users

I am planning to develop an extension for an existing email application that can take the user's inbox data in CSV format, sort the emails based on the subject line and sender's email address, and help users prioritize their email responses. The target users are individuals who receive a large number of emails per day (in the thousands) and struggle to organize their inboxes, leading to missed or delayed responses.

Significance

This tool aims to address the pain point of email overload and the challenge of effectively managing and prioritizing a high volume of incoming emails. In today's digital world, where email remains a primary mode of communication, the ability to efficiently organize and prioritize emails can significantly improve productivity and responsiveness. My tool can potentially help users stay on top of their inbox, respond to important emails in a timely manner, and avoid overlooking crucial messages, thereby enhancing their professional and personal communication.

Approach

Since I am developing an extension for an existing email application, I will need to familiarize myself with the procedures and guidelines for contributing to that application. This may involve understanding its codebase, APIs, and extension development processes. Regarding technologies, I might consider using programming languages and libraries suitable for the target email application's ecosystem. Additionally, I may need to research CSV parsing and data manipulation libraries to handle the inbox data. Potential risks or barriers could include compatibility issues with the existing application, performance concerns for large datasets, or user interface integration challenges. Mitigating these risks may involve thorough testing, and optimizing algorithms.

Evaluation

To demonstrate the usefulness and correctness of my tool, I can conduct user studies or gather feedback from a sample of target users. Metrics such as time saved, improved email response rates, and user satisfaction can be measured. Additionally, I can create test cases with varying email datasets to validate the sorting and prioritization algorithms' accuracy.

Timeline

I only have two weeks for the development:

Week 1: Familiarize myself with the existing email application's codebase and extension development process. Set up the development environment and any required dependencies.
Week 1: Implement the CSV parsing and data manipulation components to handle the inbox data.
Week 2: Develop the sorting and prioritization algorithms based on subject lines and sender email addresses.
Week 2: Integrate the developed components with the existing email application's user interface and test this extension thoroughly.

Task division

I will be responsible for all aspects of the development, including research, implementation, testing, and integration.