Use Case: Email Priority Processing System

Actor(s):

- **User**: The user interacts with the system to process emails by reading, counting, and retrieving the next email.
- **System**: The system manages the email queue, assigns priorities, and processes the email actions.

Use Case Description:

This use case describes a system that reads and processes emails from a file, categorizes them by sender priority, and allows users to interact with the email queue by viewing the next email, reading it, or counting how many emails are left to process.

Preconditions:

- The system must be initialized and the test file (Assignment5_Test_File.txt) must be accessible.
- The email data in the file should be formatted as EMAIL commands followed by sender, subject, and date.
- The priority map for sender categories should be defined and available.

Main Flow (Success Scenario):

1. **Initialization**:
    - The system reads the email file (Assignment5_Test_File.txt).
    - A priority queue is initialized to store emails sorted by priority and timestamp.

2. **Adding Emails**:
    - When an EMAIL command is read, the system extracts the sender, subject, and date from the file.
    - The system assigns a priority based on the sender's category (e.g., Boss, Subordinate).
    - The email is added to the priority queue with its assigned priority and timestamp.

3. **Counting Emails**:
    - When the COUNT command is issued, the system displays the number of emails currently in the queue.
4. **Viewing the Next Email**:

    - When the NEXT command is issued, the system displays the highest-priority email (the one at the top of the priority queue) and includes the sender, subject, and date.
5. **Reading Emails**:
    - When the READ command is issued, the system removes the highest-priority email from the queue and moves it to the "read" state.

Alternative Flows:

1. **No Emails to Read**:
    - If there are no emails in the queue, and the NEXT or READ command is issued, the system responds with "No emails available."
2. **File Not Found**:
    - If the file Assignment5_Test_File.txt cannot be opened, the system outputs an error message: "Error: Could not open Assignment5_Test_File.txt" and exits.

Postconditions:

- The system displays the count of emails remaining in the queue after each COUNT command.
- The next highest-priority email is displayed after the NEXT command, and it is removed from the queue after the READ command.
- The email queue is updated as emails are read or processed.

Exceptions:

1. **File Handling Error**:
    - If the file cannot be opened, the system exits with an error code (1).
2. **Empty Queue**:
    - If the NEXT or READ command is issued and the queue is empty, the system responds with a message indicating there are no emails to process.

Assumptions:

- The email file is properly formatted with each EMAIL command having a sender, subject, and date.
- The system uses the priority map to categorize senders (e.g., Boss, Subordinate, Peer).
- The system is designed to prioritize emails by sender category and timestamp (with higher priority emails processed first).

Email Processing System Created using Ereader