Alexandria Univeristy
Faculty of Engineering
Computer and Systems Engineering Department

CSE 223: Programming 2
Assignment 4
Due: December 20th, 2023

Assignment #4: A simple web-based email program

1 Objectives

- Dealing with JSON files.
- Dealing with Unit testing and Mocking tutorial.
- Implement a Gmail-like web application.
- Applying learned design patterns.
- Increase knowledge of REST APIs.

2 Requirements

The aim of the project is to implement the basic functionalities of a mail server, including the manipulation of mails, attachments, and contacts.

You should provide the following requirements. However, you are not limited to them, you can add more functionalities to your application

2.1 Mail Manipulation

This section will describe all the work needed (but not limited to) regarding the mailing system.

- Inbox Folder (default, priority)
- Trash Folder: auto delete emails after 30 days.
- Composing and Drafts.
- Sent Mails Folder.
- User Folders (Adding, Renaming, Deleting).
- Filters: to filter mails according to subject or sender and direct them to a specific mail folder.
- Searching and Sorting based on different attributes (Date, Sender, Receivers, Importance, Subject, Body, Attachments, . . .).

2.2 Attachments

This section will describe all the work needed (but not limited to) regarding the attachments.

- Adding, Deleting attachment(s) of an email.
- Viewing an attachment.

2.3 Contact Manipulations

This section will describe all the work needed (but not limited to) regarding the contacts.

- Adding, Editing, and Deleting.
- Searching and Sorting.
- Contact information: Name, One or multiple email addresses.

3 Implementation Details

You can either reuse your own data structures implemented last year, or use built-in ones.

- Use the appropriate HTTP request types corresponding to the different CRUD operations for sending, retrieving and manipulating the emails for example:
  - On creation use post request and send details in the request body.
  - On delete use delete request.
- Design an efficient JSON schema to store the email content and metadata required for retrieving emails and their folder organizations.
  - Your JSON schema will be evaluated for its efficiency and optimality.
- Apply at least 5 design patterns.
- When sending an email to multiple receivers, a queue data structure is used to handle the sending to multiple receivers operation.
- You should support any file type as an attachment.
- Each email can hold none, one, or many attachments.
- Emails are retrieved using pagination.
  - Moving forward and backward loads the appropriate page.
- Inbox should support 2 modes of operations:
  - Default: Which will show the emails newest to oldest (be default), user can choose a different sorting later.
  - Priority: You should use priority queue data structure to view important emails first, you should support at least 4 priorities. Priority will be assigned when composing a new email manually.
- You should enable selecting multiple emails to be able to bulk move, delete, ...etc.

4 Application

You should develop a web user interface that will ease the use of your Mailing Server App. You can test your application by opening two instances from your application at the same time on the same machine and try sending emails between them. You should implement a refresh button to check for any new email. Your Mailing Server will be operating on the same machine, there should be a parent folder that all the accounts' folders are inside it. **You are not required to develop a networked app.**

5 Notes

- Follow the Client server approach all logic should be on the server side and UI in the client side.
- By now, you should organize your code well in classes and directories. Use functions and make each function related to exactly one and only one responsibility.
- Try to follow **clean code principles** [**Top Priority**] Your program MUST NOT crash under any circumstances, even against malicious users!
- **Hint: you may reuse parts of the online available email web tutorials in Vue (for the front-end only), but it is a must to cite the references used.**

6 Deliverables

- You should work in groups of four.
- Try to use the previous mail-server application you implement last year.
- Develop this assignment in Java Spring Boot and Vue.
- You should provide an implementation for the given requirements.
- You should deliver a report that:
  - Describes thoroughly a full list of the steps required to run your code.
  - Includes a UML diagram describing your code design thoroughly.
  - Describes thoroughly how you have applied the required design pattern in your code.
  - Includes any design decisions that you have made should be listed clearly.
  - Includes snapshots of your UI and a user guide that explains how to use your application.
  - Upload your report, and source code zipped to Microsoft teams.
  - Be creative! The required features are only the beginning of what you can do, add more features or spice up the required ones, bonus marks will be given to those with eye-catching extra features and user-friendly interfaces.
  - Delivering a copy will be severely penalized for both parties, so delivering nothing is so much better than delivering a copy.