

Alexandria University
Faculty of Engineering
CSED 2025



M.(Mkario) Y.(Yasser s) Mail

OOP Lab 4

Report

Submitted to:

Eng. Ismail El-Yamany

Faculty of Engineering Alex. University

Submitted by:

Islam Yasser Mahmoud 20010312

Ehab Yasser Mahmoud 20010382

Marwan Yasser Sabry 20011870

Mkario Michel Azer 20011982

Faculty of Engineering Alex. University

Table of Contents

Problem Statement:.....	4
Features in the program:	4
Required steps to run the program:	4
User guide:	5
Snapshots of the UI:.....	12
Class diagrams:.....	13
How the design patterns are applied:.....	15
Design decisions and assumptions:	16

Table of Figures

Figure 0-1: Login screen	5
Figure 0-2: Signup screen	5
Figure 0-3: Inbox(Default) screen	6
Figure 0-4: Email-form screen.....	6
Figure 0-5: compose screen	7
Figure 0-6: Trash screen	8
Figure 0-7: Draft screen.....	8
Figure 0-8: Sent screen	9
Figure 0-9: Add contact screen	9
Figure 0-10: Contacts screen	10
Figure 0-11: Contact show screen.....	10
Figure 0-12: Add folder screen	11
Figure 0-13: Folder screen.....	11

Problem Statement:

A simple web-based email program similar to that of Google whose operations are done on the server.

Features in the program:

- 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)
- Adding, Deleting attachment(s) of an email.
- Each email can hold none, one, or many attachments.
- Any file type is supported as an attachment.
- Adding, Editing, and Deleting contacts.
- Searching and Sorting contacts.
- Contact information: Name, One or multiple email addresses.
- Pagination for messages sections(Inbox, Trash, Draft, Sent) and contacts.
 - Moving forward and backward loads the appropriate page. Or you can press on the page's number.
- There are 5 priority levels.
- Email-selection is supported including:
 - Single email selection to be able to move, delete, ...etc.
 - Multiple emails selection to be able to bulk move, delete, ...etc.
- Logging out

Required steps to run the program:

1. Extract the compressed program folder.
2. Run the back end:
 - Open the Email_Back folder using IntelliJ IDE (recommended) or any other java IDE , or open it by simply running the pom.xml file
 - Run the EmailBackApplication on: src/main/java/com.example Email_Back/Model.
3. Run the front end:
 - You should have NodeJS and Angular-CLI if you don't download them.
 - Open the Email_Front folder using VS code IDE (recommended) then copy this to your terminal **npm install** and press Enter.
 - Copy this to your terminal **ng serve** and press Enter. Then the paint program is going to run on **localhost:4200** copy it to your MS edge end press Enter the interface of the program is going to show up and enjoy 😊. (You can replace MS edge by Chrome but first you have to open launch.json in the Email_Front from inside the VS code IDE and replace the **2 msedge** written in the page by **pwa-chrome** and **chrome** respectively.)

Note that:

- To run the program, you have to stick to the previous steps with their order and if you restarted anyone of them for any reason you have to restart the other and put the order into consideration.
- If there is a problem with the libraries copy these to the terminal of the Front end:
 - `npm i ngx-pagination --save`
 - `npm i @fortawesome/fontawesome-free --save`
 - `npm i file save --save`

User guide:

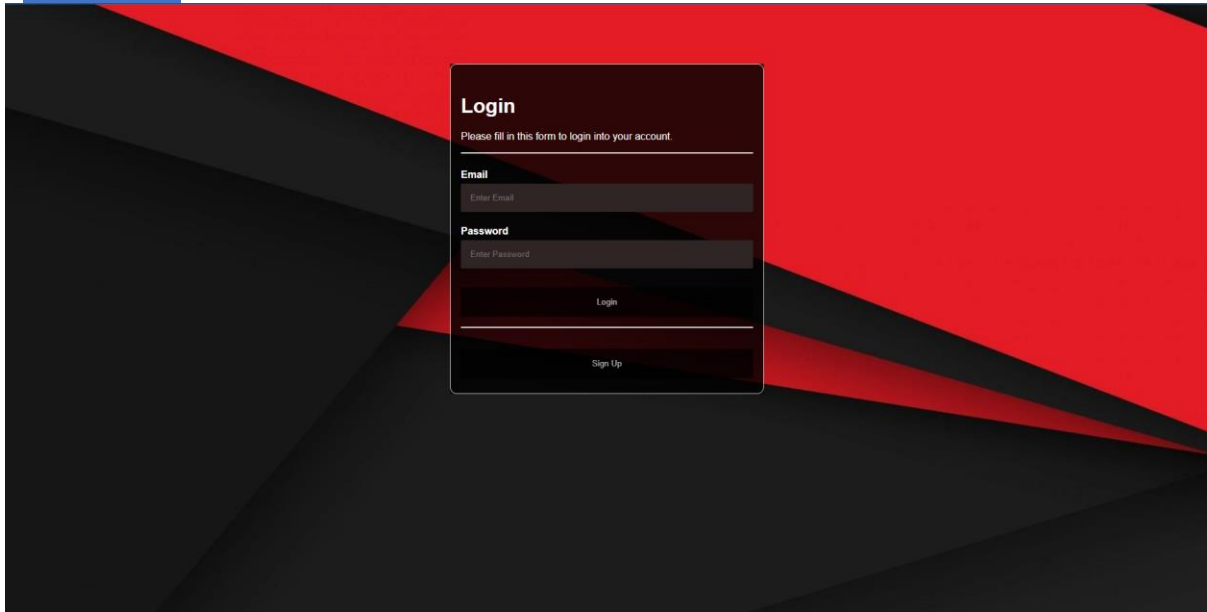


Figure 0-1: Login screen

In Fig 0-1: we have the login-screen asking for:

- An email
- Its password

Or if you don't have one you can press Sign-up button to go and make one.

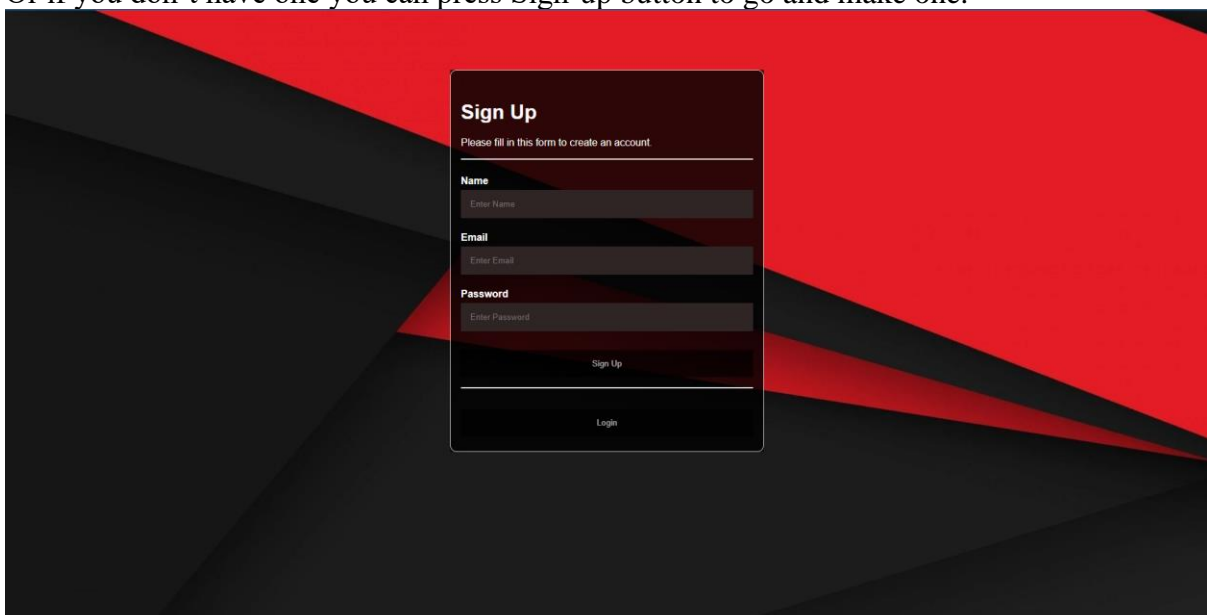


Figure 0-2: Signup screen

In Fig 0-2: we have the signup-screen asking for:

- A name
- An email
- A password

Or if you have one you can press login button to go and log into it.

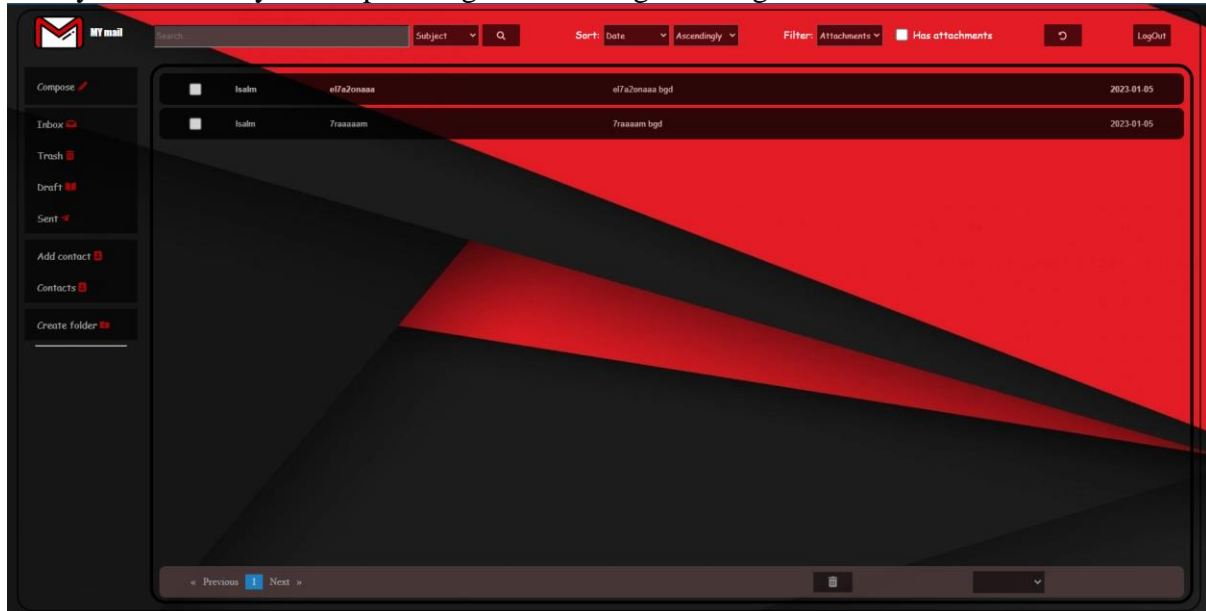


Figure 0-3: Inbox(Default) screen

In Fig 0-3: we have the Inbox(Default) screen containing:

- Emails (from left to right):
 - You have a select button for performing deletion or moving to any folder – if exist – These options are in the taskbar as shown.
 - The name of the sender
 - The subject
 - The header
 - The date

To open any email you have to simply press on it and this will show up:

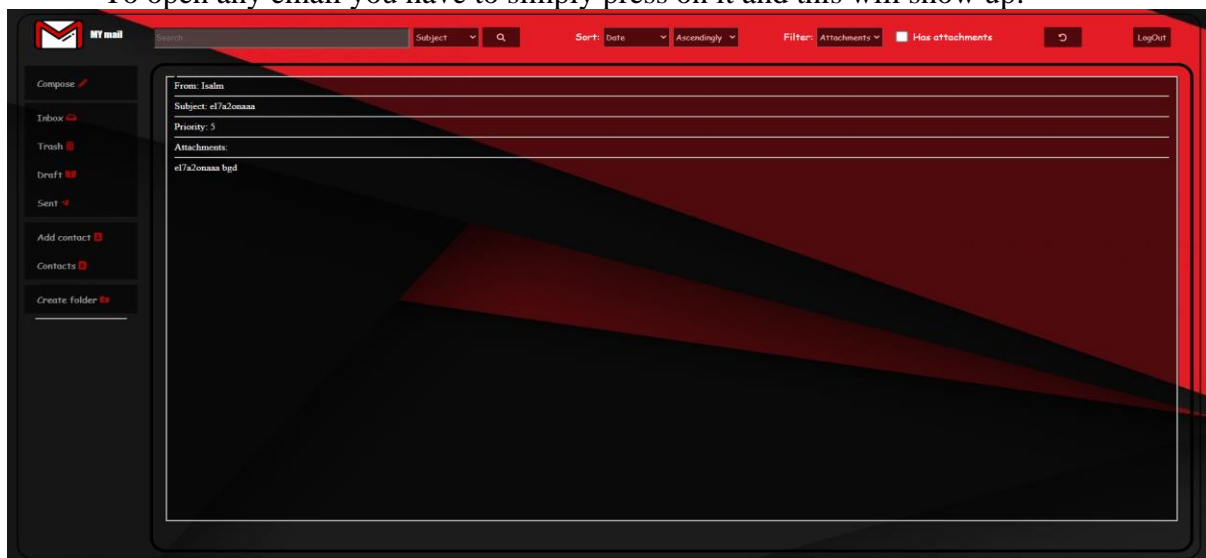


Figure 0-4: Email-form screen

- The email is containing:
 - The name of the sender
 - The subject
 - The priority of the message
 - The attachments – if exist –
 - The body of the message
- On the top in the header there are:
 - Search section containing:
 - Search bar
 - Search menu to choose what do you want to search about
 - Search button to approve the search
 - Sort section containing:
 - Sort menu to choose the thing that you want the sort to be performed on.
 - Sort type menu:
 - Ascendingly – default –
 - Descendingly
 - Filter section containing:
 - Filter menu to choose what do you want to filter about
 - Checkbox that has to options:
 - If you choose Attachments from the menu: it will be for having attachments(marked) or not(unmarked)
 - If you choose either Date or length: it will be asking below (unmarked) or above (marked)
 - In addition to an input box will show up to take the input that you want to search about
 - Refresh button
 - Logout button

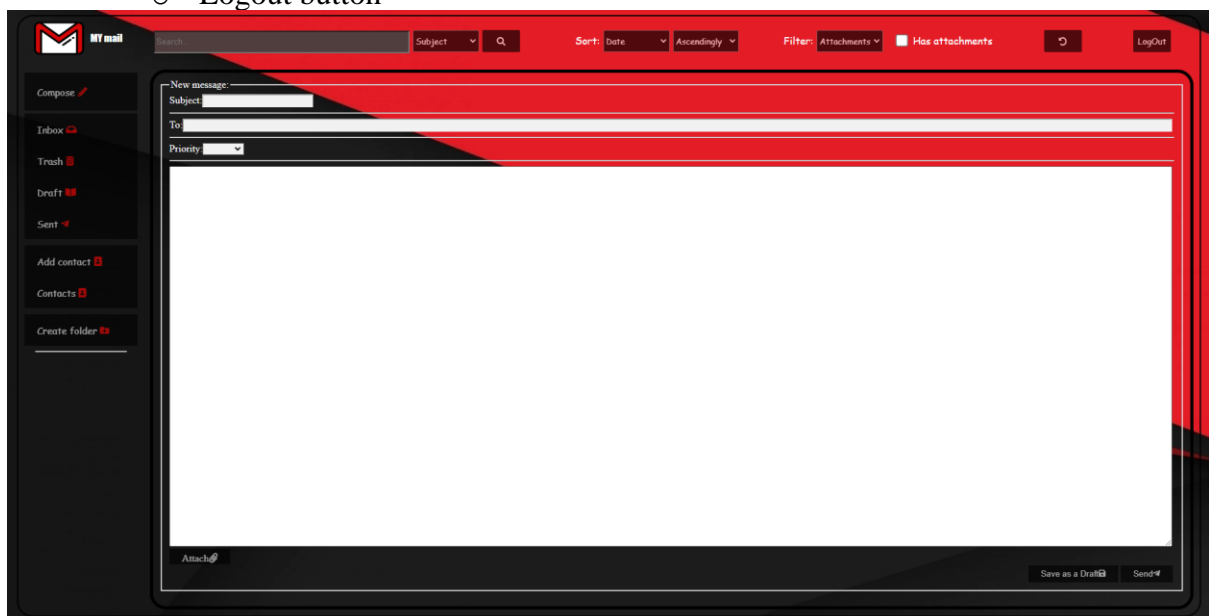


Figure 0-5: compose screen

In Fig 0-5: we have the compose screen containing:

- Subject field
- To field

- Priority field
- Body field
- Attachment button
- Save to draft button
- Send button

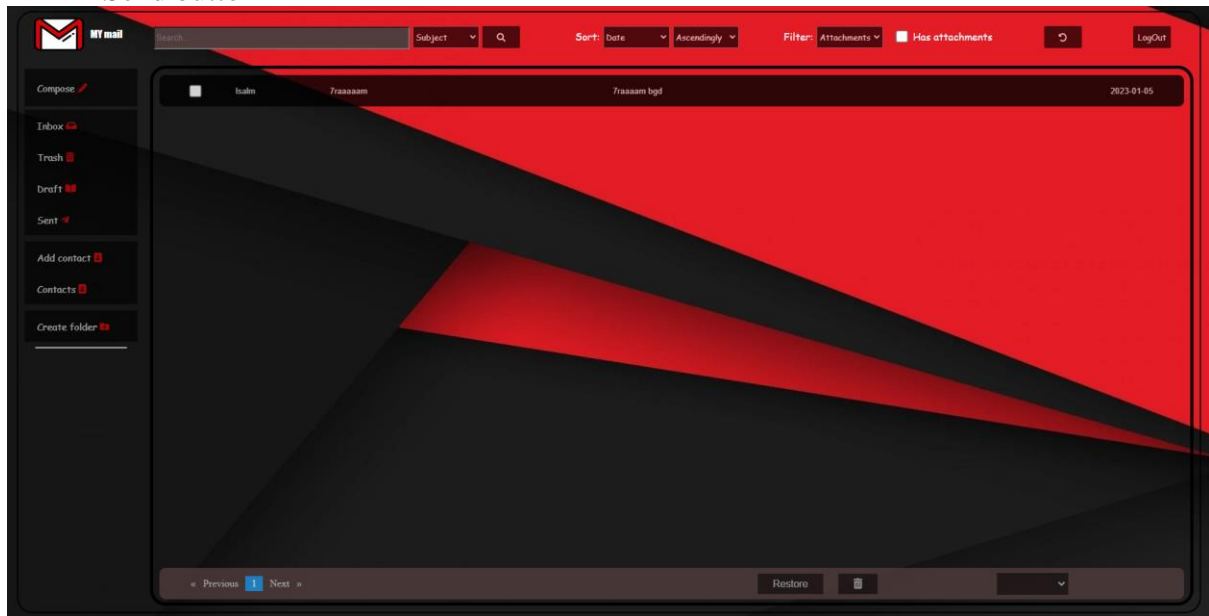


Figure 0-6: Trash screen

In Fig 0-6: we have the trash screen containing:

- The emails
- The taskbar containing:
 - Pagination
 - Restore button
 - Permanently delete button
 - Folders menu to move to

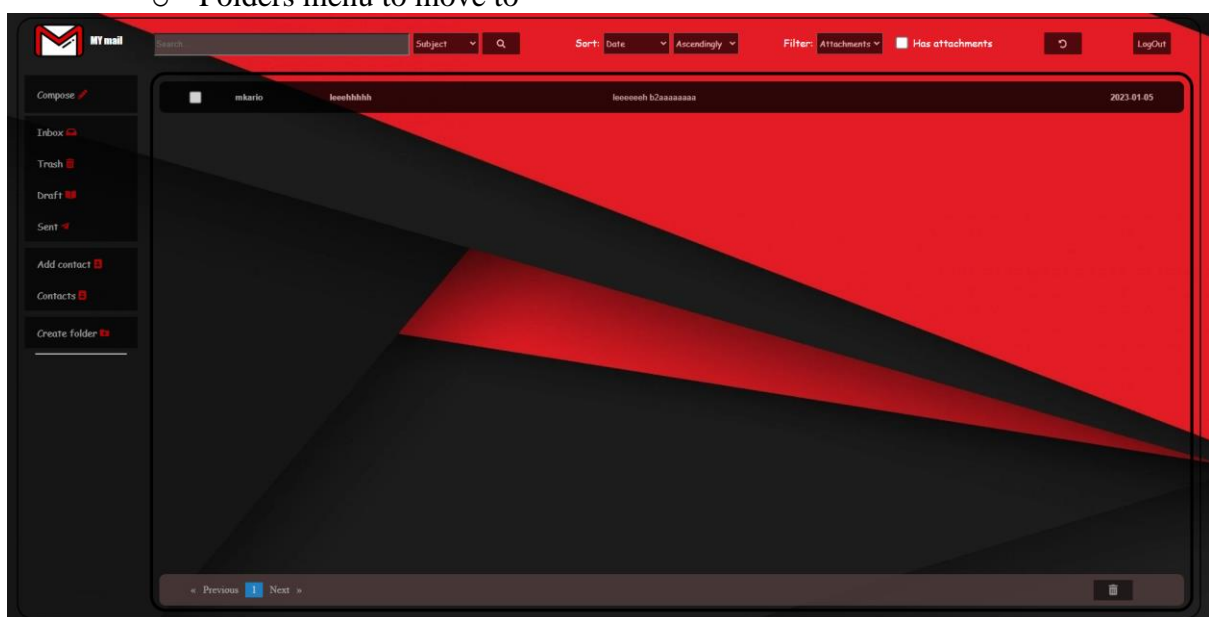


Figure 0-7: Draft screen

In Fig 0-7: we have the draft screen containing:

- The emails
 - If you press on it the compose screen will open up again to continue typing
- The taskbar containing:
 - Pagination
 - delete button

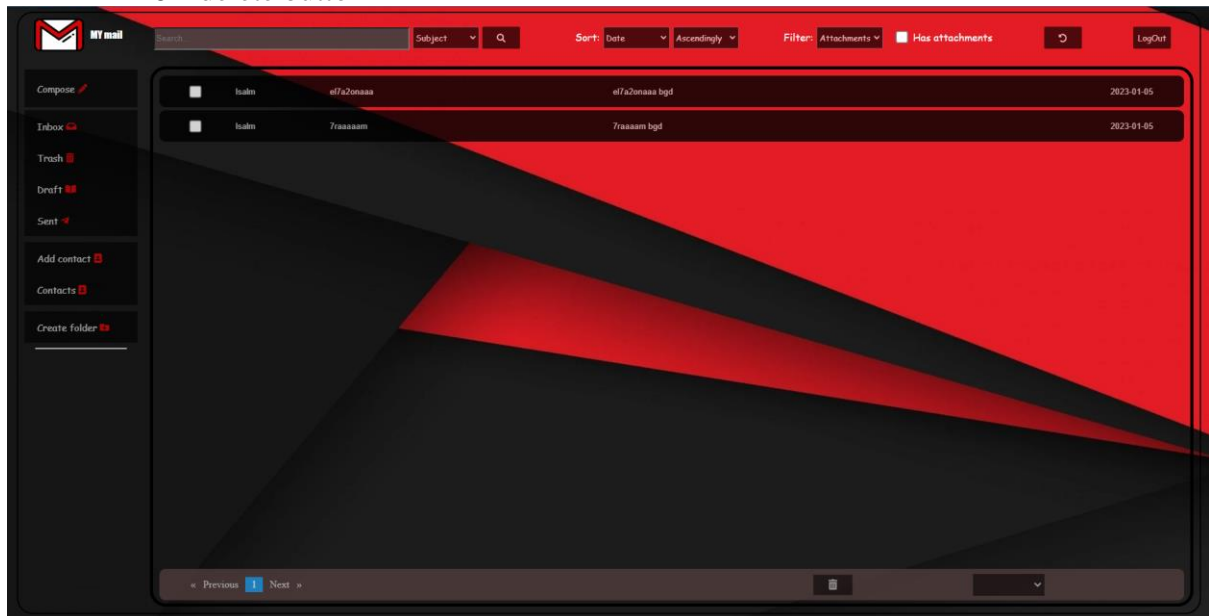


Figure 0-8: Sent screen

In Fig 0-8: we have the draft screen containing:

- The emails
- The taskbar containing:
 - Pagination
 - delete button
 - Folders menu to move to

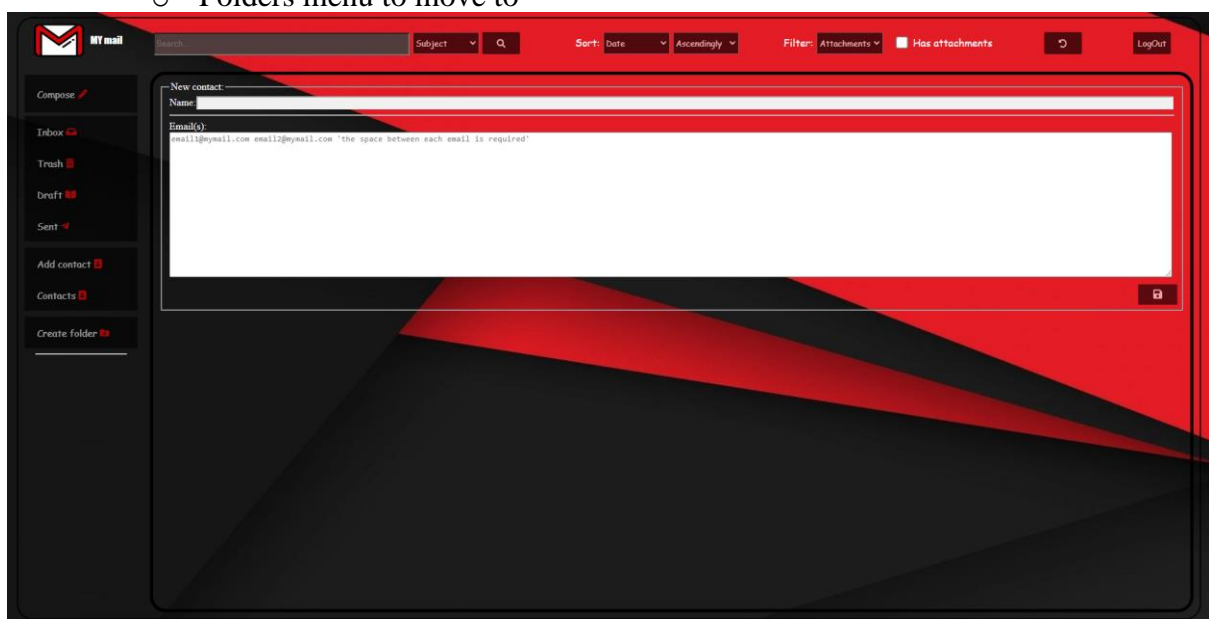


Figure 0-9: Add contact screen

In Fig 0-9: we have the Add contact screen containing:

- Name field
- Emails field
- Save button

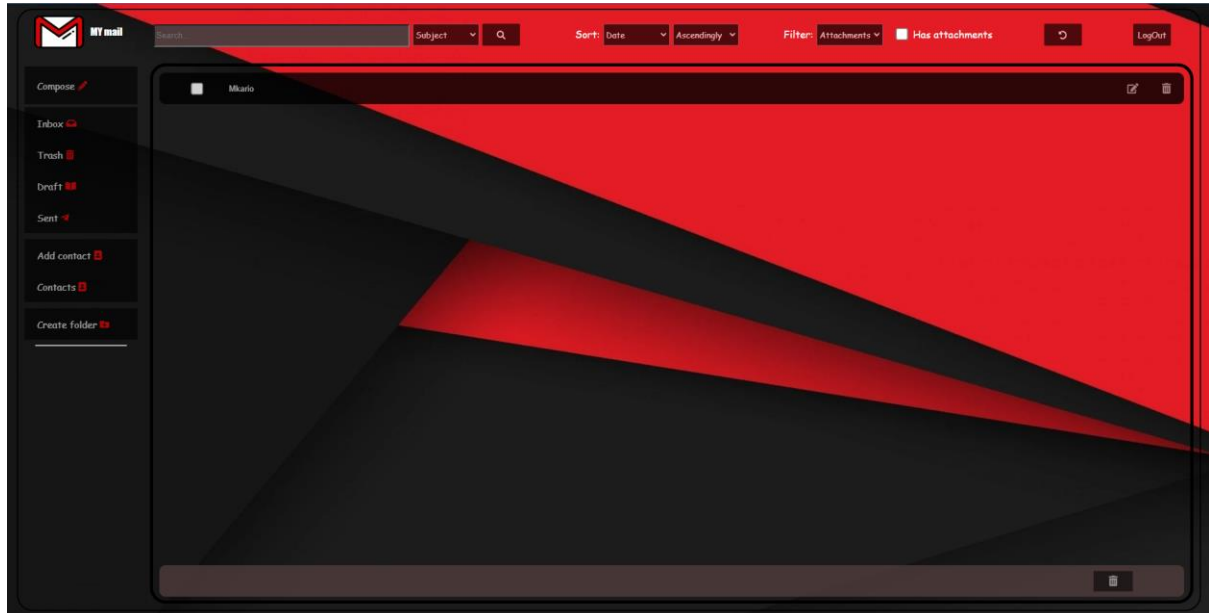


Figure 0-10: Contacts screen

In Fig 0-10: we have the contacts screen containing:

- You have a select button for performing bulk deletion in the taskbar.
- Name of the contact
- Edit button leading you to the add contact screen
- Delete button

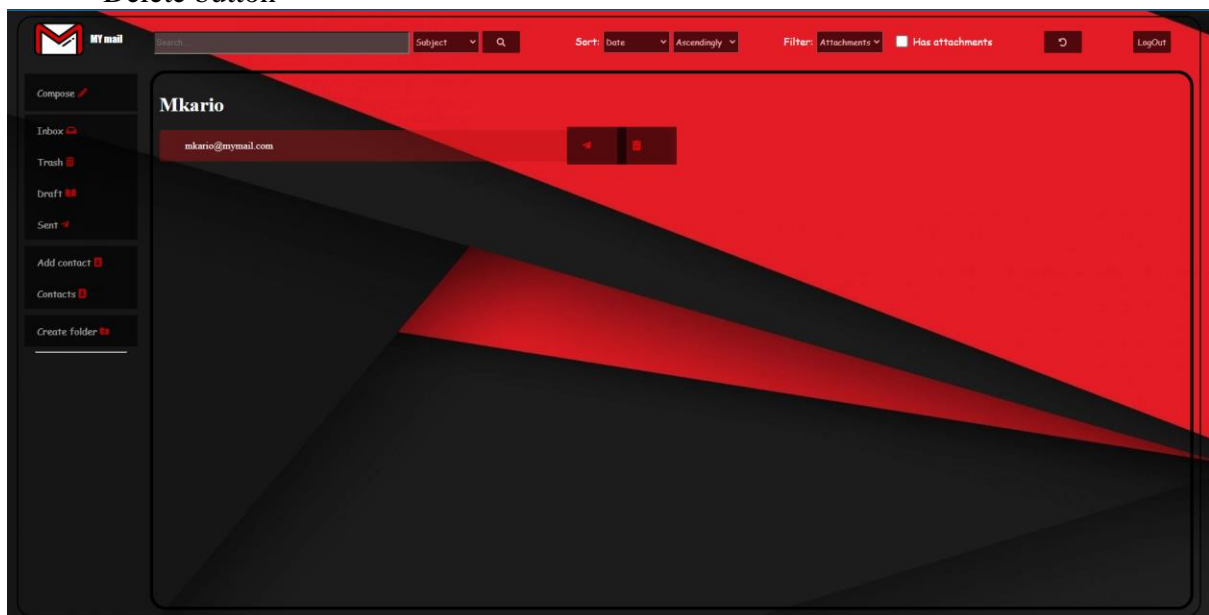


Figure 0-11: Contact show screen

In Fig 0-11: we have the contact show screen (after pressing on a contact)containing:

- The emails
- Send button leading you to the compose screen
- Delete button

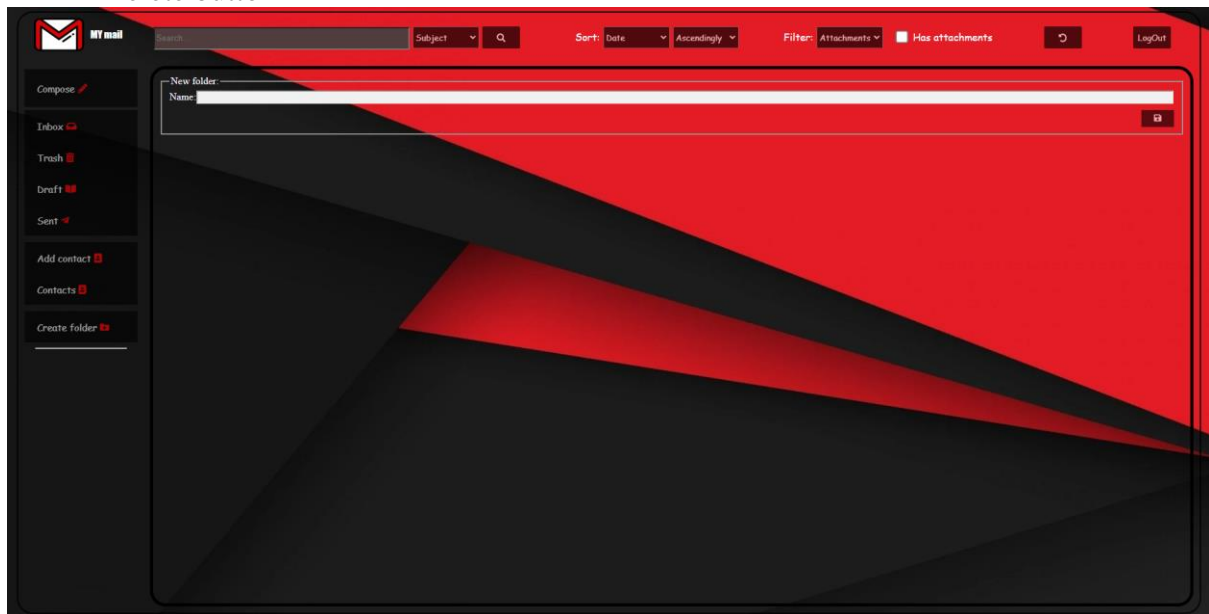


Figure 0-12: Add folder screen

In Fig 0-12: we have the Add folder screen containing:

- Name field
- Save button

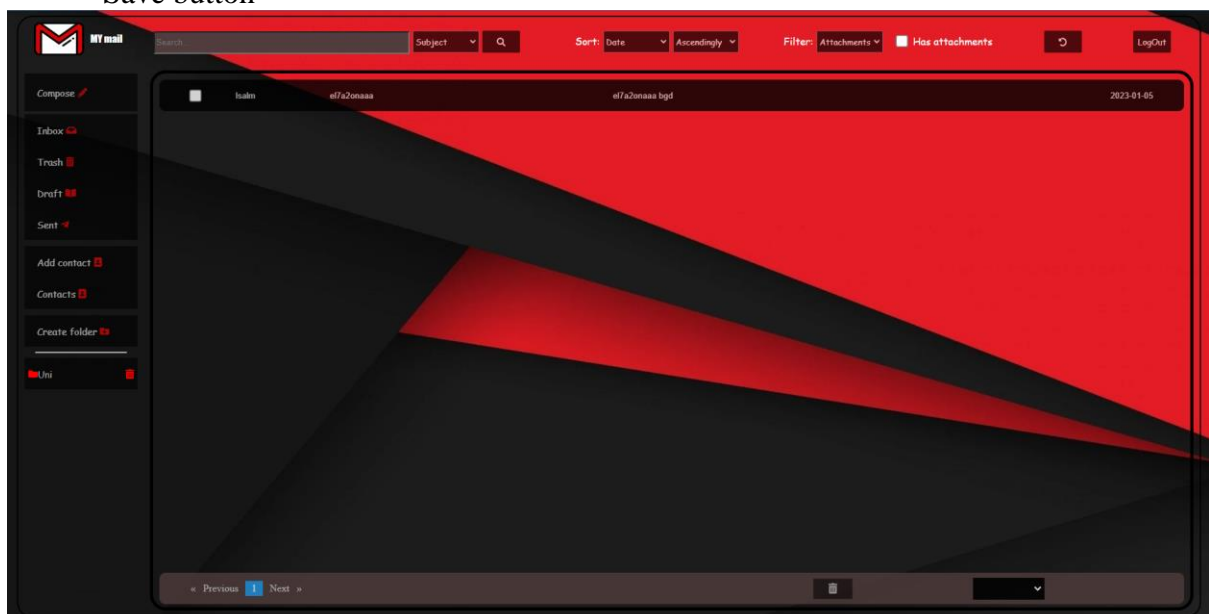
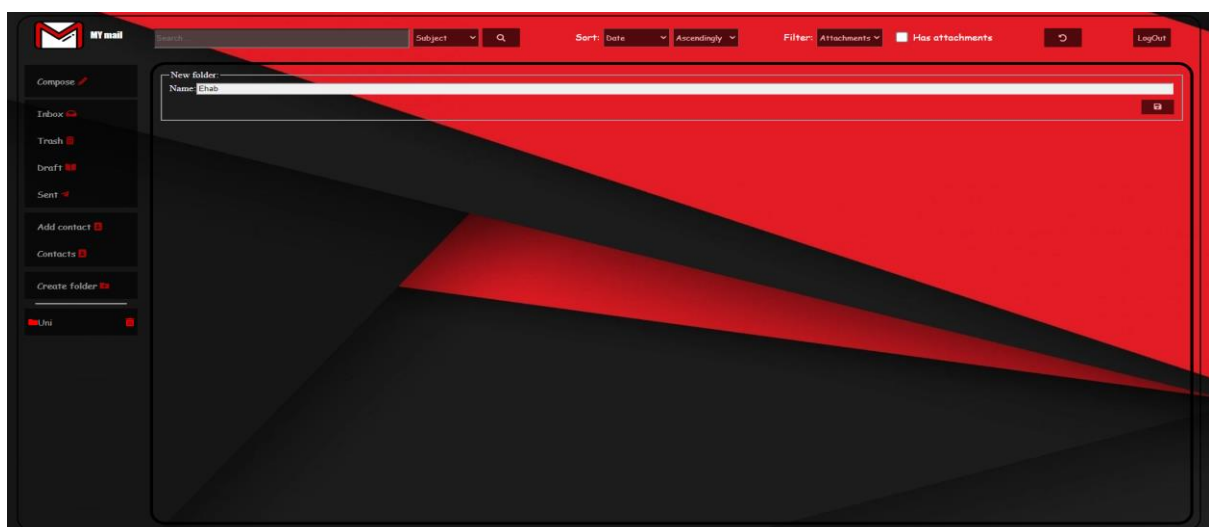
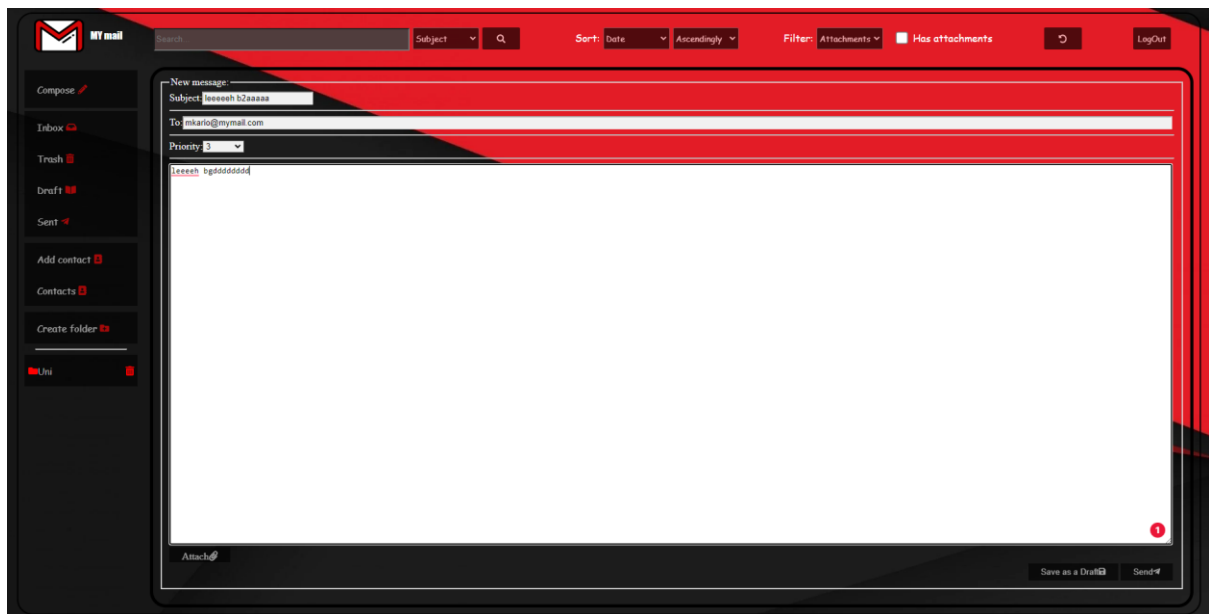
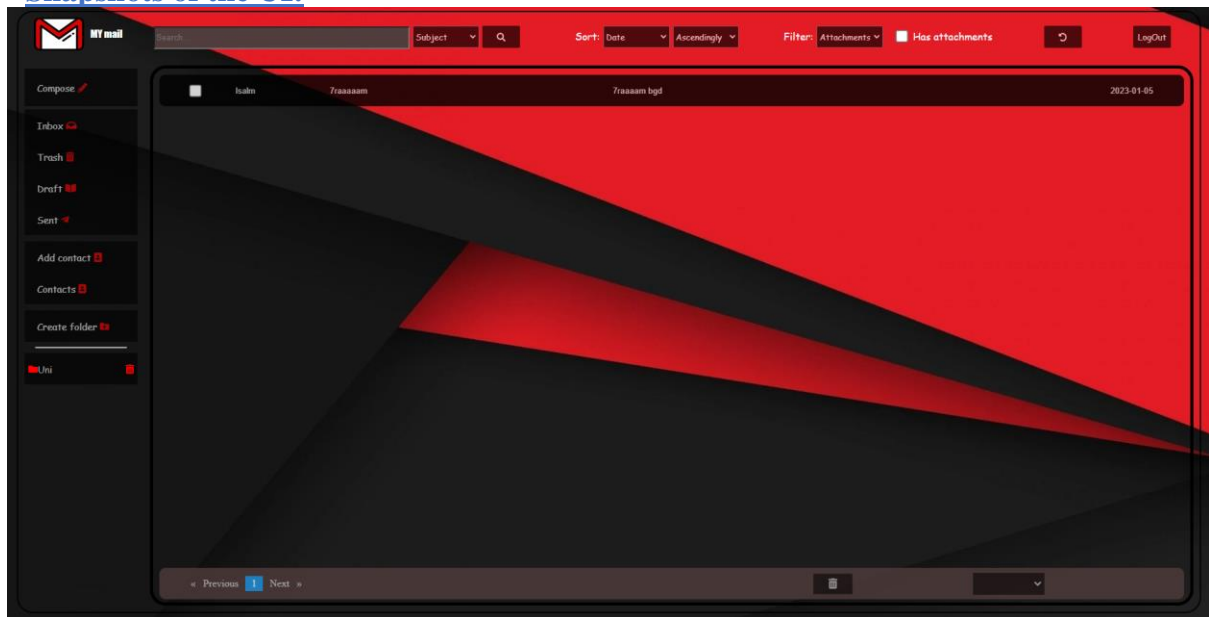


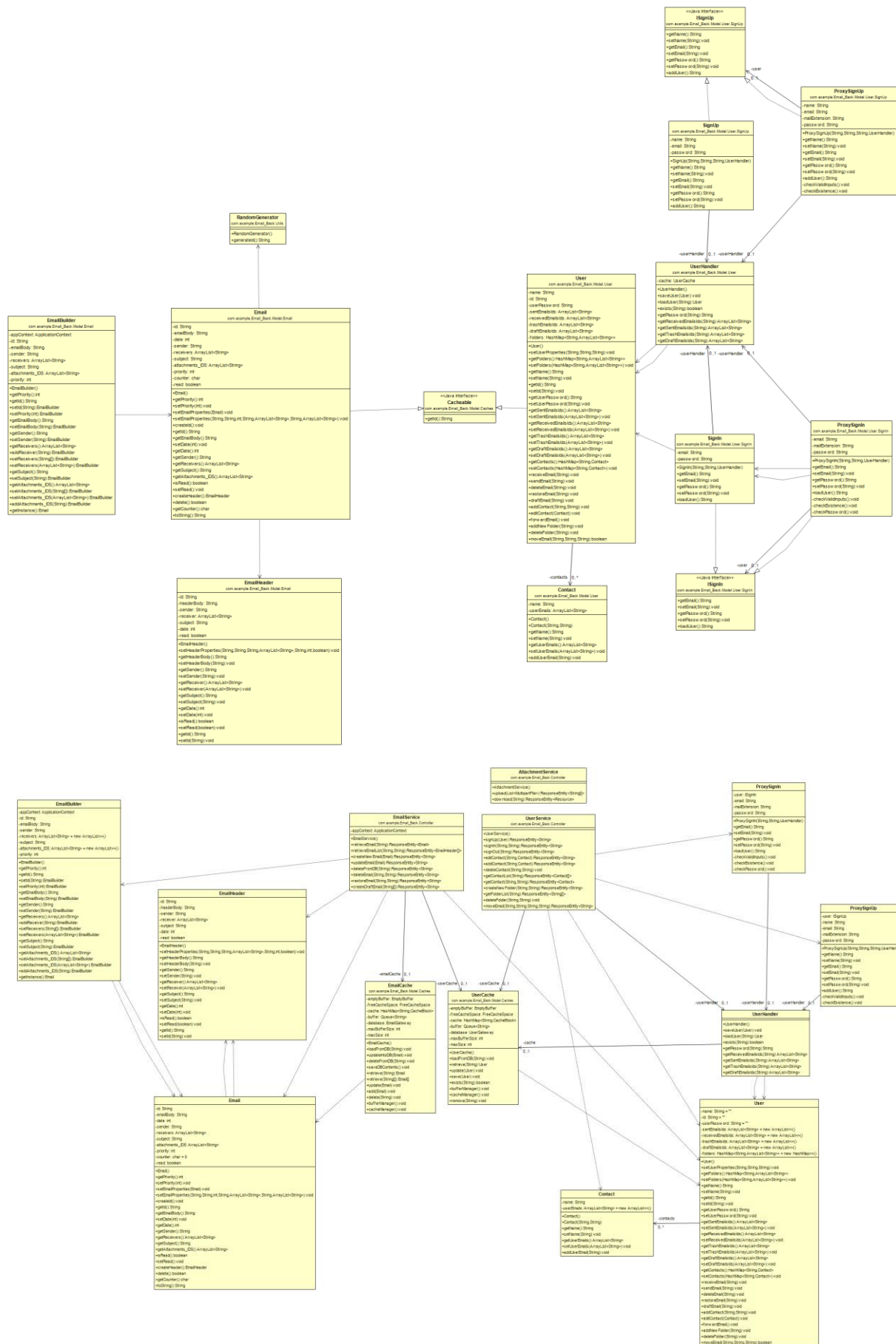
Figure 0-13: Folder screen

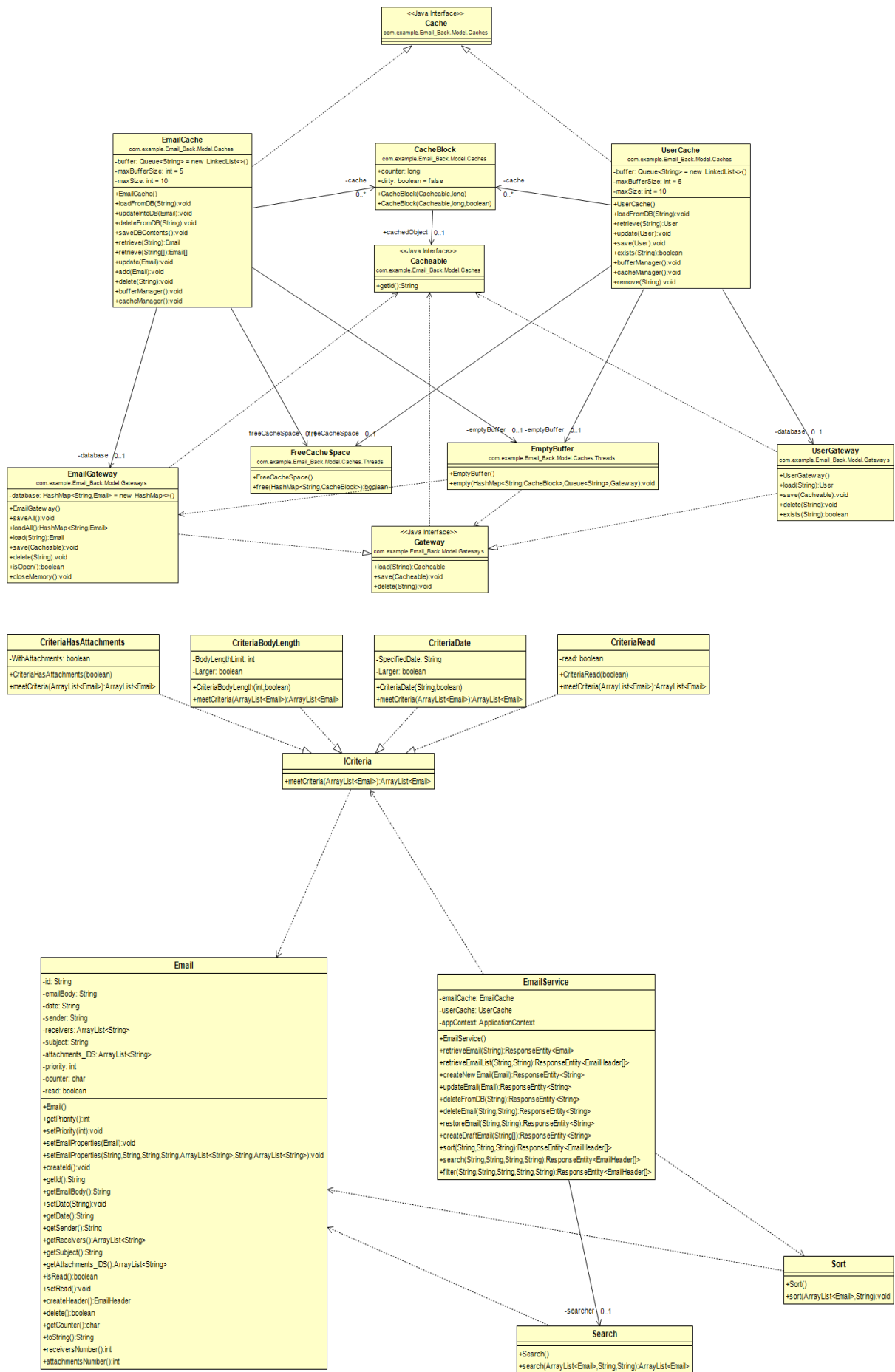
In Fig 0-13: we have the folder screen same as inbox

Snapshots of the UI:



Class diagrams:





How the design patterns are applied:

1. Proxy:

- Front:

We use Proxy Design pattern in the FrontEnd where it is responsible for sending and receiving Requests to/from HttpService which in order sends post and get requests to the server side and returns responses to the Proxy to handle them.

- Back:

We used access control proxy design pattern while sign in and sign up users to validate the user inputs and make sure that the password entered is correct while signing in the user and also to prevent wrong input data while signing up the user

2. Builder:

The builder design pattern was used to create drafts for when there is not enough information for a full email.

3. Cache:

It is used to decrease the number of database accesses and file reads from the server in order to increase performance. By saving all the needed data in cache, all relevant information could be easily retrieved.

4. Singleton:

The Singleton design pattern was used with the two cache classes in order to make sure there is one instance of the cache at all times.

5. Iterator

It is used in deallocating from the cache whenever it reaches a certain size threshold. The iterator is used for a priority queue that arranges all elements in the cache by the time since they were last retrieved or used.

This is the Least Recently Used replacement policy in cache design.

6. Filter:

we used filter design pattern to filter emails according to some criteria chosen by the user e.g., we filter emails according to whether they have attachments or no whether the body length is larger or smaller than some value entered by user whether the date is earlier or later than the date entered by user

7. Prototype:

Prototype design pattern is used for instantiating new emails, email builders and user classes.

Design decisions and assumptions:

- We used Angular for Front-End coding.
- We used Spring boot for the Back-end coding.
- In front End we have used:
 - npm ngx pagination library
 - npm file saver library
 - npm fortawesome library
- The organization of components and the routing required making lots of components and modules.
- "Service" class contain many functions to connect frontend to backend.