Ana Barcenas
Mitchell Maliglig
Manuel Miranda
Eric Valdez

Final Project Report

PROJECT DESCRIPTION

The goal of the project is to build a simple and functional mail client server that can send an email to any recipient by establishing a TCP connection with the mail server(we did gmail) and dialog with the server using SMTP. Other requirements are: include a login page, a displayed inbox, and a send message or compose email page. The purpose of the project is to implement the knowledge learned in class and focus on the functionality of the application.

PROGRAMMING EXPERIENCE

BEGINNING/PLANNING

The beginning stages of the project were to research what was the best approach for the group to work together, i.e. programming language, API, interface. We decided to use Java as our programming language because it is what everyone was most familiar with. The JavaMail library was used to make the connection between the user and the mail server, whether it was to send an e-mail or fetch the user's inbox. We decided to use JavaFX to implement all the user interfaces.

BACKEND

The backend was created by implementing the JavaMail java package. This allowed us to make both SMTP connections for sending messages and IMAP connections for retrieving the inbox. We created some of our own objects for things like email drafts, and the email messages that are returned from the IMAP connection. The class that handles sending emails uses our EmailDraft object to create a standardized MIME message which SMTP can send. The class that retrieves the inbox has a method that returns an ArrayList of our EmailMessage object type. This ArrayList can be iterated through to display the emails in an inbox.

FRONTEND

The frontend of the application consists of three separate scenes, the login screen, the inbox page and the compose an email page. The login screen just asks the user to input their valid email and password. The user has to enter their credentials every time they open the mail client. Once the user inputs their correct credentials, they click the login button and are taken to the inbox page. The inbox is displayed in a table with four columns (Date, From, Subject and Message) and the messages are displayed from oldest to newest. There is a slight delay when changing to the inbox because the program has to iterate through all the received emails and add them to the table. The inbox screen has a button that leads to the compose an email screen. The compose an email screen asks the user for the recipient's email address, the subject of the message, and the body of the e-mail. When the user clicks send, there is a label that notifies the user if the email was sent successfully. They are also given the option to return to the inbox screen. When the user is finished using the program they can exit by closing the window.

CHALLENGES AND RESOLUTIONS

WORKING WITH DIFFERENT ENVIRONMENTS

One of the first issues we ran into was that some group members were using different environments to run the code and it would not compile correctly or run into an error on one person's computer and run fine on others. This would probably have been remedied by implementing a dependency manager like Maven from the start.

IMPLEMENTING THE GUI

One of the biggest issues we had to resolve was working with Scenebuilder to implement the inbox. Scenebuilder was really easy to use to make the inbox scene, however, its application lifecycle is incompatible with that of native JavaFX. As a result merging these GUIs was impossible and very late into the development stage, so the inbox was built from the ground up using native JavaFX.

MULTIPART MIME MESSAGES

Once we had finally gotten the socket programming aspect of the project completed we realized that displaying the emails in an inbox was not as easy as we had expected. Usually, emails are multipart MIME messages that contain many parts for things like text formatting and attachments. We had to create a parser that would go through this MIME message and extract the body of the message to display in our GUI.

CONCLUSION AND REFLECTION

Overall, working on this project was a very rewarding experience for all of us. We learned the internal workings of email, a tool we use constantly in our daily lives, and learned how network protocols govern the communications on the internet. This was also yet another experience in practicing software engineering principles from designing a full application, to implementing goals and objectives within set timeframes, to merging a cohesive vision for what the project should look like out of all of our independent ideas.

PROJECT SPECIFICATIONS

Included in our submission is a "src" folder that contains all of our source code for this project. Our main method is located in "MailClient.java". This application requires Java 1.8

(Java 8) to run, as that is the version of java we used which includes JavaFX. Another dependency is JavaMail, whose .jar is included in the lib folder. This program was only tested on Visual Studio Code.