Electronics and Computer Science
Faculty of Engineering and Physical Sciences
University of Southampton

Charlotte Graham – cb6g19

May 2022

A Better Email Client

Project supervisor: Leslie Carr
Second examiner: Gary Willis

A project report submitted for the award of
MEng Computer Science

Abstract

Email has existed for the last 30 years with little change. It has fallen behind other communication methods such as team communication software like Microsoft Teams or Slack. Users often experience “email overload”, where they feel overwhelmed at the quantity of emails they receive. This research project investigates why “email overload” occurs, how it affects users and potential solutions.

The research identified a number of issues, including the need for more automation in the organisation of inboxes, the desire to place the email client into a ‘mode’ to suit the users’ needs at that time and the need for fine-tuned control over their notifications to prevent distraction. It was discovered that many large email client providers such as Microsoft have been working to develop better automatic organisation of email inboxes; for example Outlook has introduced “Focused Inbox”. It was therefore decided that since this concept is already under development, it would not be included in this project.

It was found that the concept of ‘modes’ and finely tuned notification control were the least explored and therefore an email client with ‘modes’ to allow the user more control over their notifications was developed as a proof of concept.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

I have acknowledged all sources, and identified any content taken from elsewhere.

I have not used any resources produced by anyone else.

I did all the work myself, or with my allocated group, and have not helped anyone else.

The material in the report is genuine, and I have included all my data/code/designs.

I have not submitted any part of this work for another assessment.

My work did involve human participants, their cells or data, or animals - ERGO/FEPS/ 41626.A2

ECS Statement of Originality Template, updated August 2018, Alex Weddell
aiofficer@ecs.soton.ac.uk

Acknowledgements

I would like to thank my supervisor Leslie Carr for his support and guidance throughout this project.

I would also like to thank Gary Wills for his valuable advice as second examiner.

Finally I would like to thank my friends and family for their unwavering support during this project.

Contents

Chapter 1 - Introduction

This chapter introduces the problem identified, the goal and scope of the project.

1.1 Problem

Users of email clients often feel overwhelmed at the vast quantity of emails they receive, this is termed “email overload” [2]. Research into why this happens and how to prevent it has shown that users expressed a need for better automatic organisation of their inboxes [3], a desire for ‘modes’ in their email client to suit their needs at a given moment [3] and a desire for more control over their notifications [4]. Since automatic organisation of inboxes is an area which companies like Microsoft and Google have sort to improve upon with their “Focused Inbox” and “Tabbed Inboxes” respectively, this concept has been proven and is already implemented. Therefore I am focusing my project on the development of modes in email clients that will give users better control over their email notifications and through this aim to reduce the effect of “email overload” and “getting lost” in emails after receiving a notification [5].

1.2 Goal

The goal of this research project is defined below:

- The primary aim of this project is to research into the current problems that exist within mainstream email clients today and develop a proof of concept that would demonstrate how the problems identified in the research could be reduced
- To demonstrate the proof of concept, modes within an email client will be designed and implemented, these will aim to reduce the problems identified in the research
- The modes will demonstrate how this concept could be useful to users and integrated into main stream email clients

1.3 Scope

The scope of this research project is defined below:

- To develop an email application using Java, JavaFX and Microsoft’s Outlook Mail API to implement a proof of concept of modes
- The email client will only be a proof of concept therefore will not include all production-level features

Chapter 2 - Background Research

This chapter describes the background research carried out for this project. The background research includes a literature review and a review of a similar product. This research was using to identify the issues within current email clients.

2.1 Literature Review

2.1.1 Information Overload

The number of emails sent and received per day will exceed 319 billion in 2021 [6]. "Email overload" is the term used for when users feel overwhelmed at the vast number of emails they receive [2]. Szóstek claims this feeling is caused by current email clients not being able to support users to "effectively and efficiently manage their inboxes". The large quantity of emails received will usually lead to the user "triaging" their emails - the process in which the user will handle the incoming messages [7], [8]. However due to lack of time, resources or information, users will defer handling emails until a later time or date [7], [8]. Sarrafzadeh et al.[8] found that 48% of user triage their emails sequentially and 41% triage them by priority. They also observed that slightly over half of their survey respondents look through their inbox multiple times in order to triage their emails and 46% look through their inbox only once.

2.1.2 Revisiting Emails

Reasons for revisiting emails include retrieval of information from the message body, accessing attachments and taking action against the message [9]. Awadallah et al. [9] found that 73.4% of emails were revisited to find some information, 20.6% to respond to the email and 0.6% to delete or organise it. 71.6% of their survey respondents said they used search to revisit an email, 11% browsed their inbox and 9.9% browsed a folder/category. Furthermore, the survey showed that 62.3% of users anticipated they would revisit certain emails [9]. Sarrafzadeh et al. [7] found that in a sample of logs from a popular email client, 10% of all emails received a Reply, ReplyAll or Forward and 26% of these actions were performed only when the email was revisited.

2.1.3 Organisation

Online mailbox storage is ever increasing and therefore deleting messages is becoming nonessential [2], [10]. When monitoring 2 months of users' activities on Yahoo! Web mail service, it was found that 82% of users had never deleted a single message [10]. However, research has shown that when re-finding emails, 88% of finding operations were successful and the average number of finding operations per sequence was 3.85 [11]. The same study proposed that there were two different access behaviours when it came to re-finding emails: preparatory activity and opportunistic accesses [11]. Preparatory activity occurs when users create folders and tags to prepare for the arrival and retrieval of emails. Opportunistic accesses rely on sorting, scrolling and searching [11]. Whittaker at al. found that opportunistic behaviours dominate – accounting for 87% of accesses in their sample. This was mainly made up of scrolling which accounts for 62% of all accesses. Their results also

suggested that users tend to rely exclusively on either preparatory or opportunistic behaviours, this may indicate that not all email clients are optimal for all users [11]. These findings were further supported when it was found that 70% of users of the Yahoo! Mail service had never defined a single folder [12].

Research into how users can best organise their inboxes shows that 83% of participants would like the ability to "annotate" their emails – which can be defined as "the possibility to indicate a relative importance of an email and also its relationship to other emails" [2]. Participants of the same study wanted to assign priority and links between related messages to their emails and have them arrange in this while whilst maintaining the automatic chronological structure of the inbox [2].

Respondents of a survey were also interested in alternative views of their inbox, especially alternative views of email threading – some suggesting a tree-like presentation of email threads and others suggesting a presentation that more closely mimics instant messages services like Facebook or Slack [3]. When analysing scripts found on GitHub that were written in order to customise inboxes, it was found that scripts that provided statistics and insights to the users in order to measure productivity were popular [3].

2.1.4 Automation of Inboxes

Existing email clients and other add-in software has tried to reduce the feeling of email overload by offering some automation features, however this can leave users having to juggle 3rd party plugins to manage all of their needs, have a blended approach of some automated features with some human intervention or simply choose to do everything manually [3]. The research done by Park et al. [3] showed several areas where automation in email clients could be introduced or improved. For example, a "richer data model" for emails so that users could understand information such as priority, topic, deadline and need for a reply [3]. Users expressed the desire for automations to leverage the context of an email, for example the user would be able to find out characteristics of an email thread such as number of responses previously and the rate of responses by others as well as the state of recipients, such as whether they are in the office and available to reply, busy or on holiday [3]. Another example of automation users brought up was automated content processing, for example to automatically accumulate replies to an invitation or save photos that have been attached to an email into a predefined location [3].

Users expressed interest in better support for prioritising of emails, for example important contacts, emails that contained large numbers of follow-up tasks and a sender they reply to frequently would all have higher priority than email blasts, FYIs or senders they rarely or do not reply to [3]. Prioritisation of emails could better capture the attention of users, this is relevant since it was found that "curiosity drives attention" [13]. "Information gap" has influence on attention, independent of message importance [13]. When conducting an experiment to find why some emails are opened over others, participants read 56% of messages with an information gap but only 38% of message with no gap [13]. Wainer et al. [13] then suggested that email clients could automatically and intuitively adjust the way they are presented based on the user's inferred demands.

2.1.5 The Introduction of 'Modes' in Email Clients

When asked in a survey, respondents said they would like the opportunity to configure their email clients according to their own set of defined rules, essentially creating 'modes' for the email client depending on some external context like time of day or some input from the user [3]. Some examples of modes that respondents offered were: work, vacation, home/family, busy, study etc [3]. The ability to more finely tune push notifications also arose in the answers to the survey carried out by Park et al [3], for example users didn't want to be notified about every email they received but they wanted to be notified immediately if they received a high priority email.

2.1.6 Notifications

Email 'modes' could also help users regulate their notifications. It has been found that sadness can be detected on the faces of those who are frequently interrupted during an essay writing task [4]. Giving users more control over their notifications could also allow them more privacy. When asked in a survey, 17% of respondents said they would be uncomfortable receiving any work email in the presence of others and 40% said they would feel uncomfortable receiving personal emails [12].

It has also been shown that users are "getting lost" in emails, which refers to users spending more time in email than originally intended, especially when an email notification interrupts a working session [5]. When completing diary entries, 64% of participants reported they "got lost in email", both during short and long sessions. It was found that if participants started their email session because they were interrupted, then they were 3.025 times more likely to "get lost" [5].Hanrahan and Pérez-Quiñones [5] found that a high number of emails awaiting replies was the primary reason for users "getting lost", as replying to messages was a strong correlate to becoming lost in their emails. Moreover, for each message that a participant of the diary study composed, they were 0.5 times less likely to "get lost" and therefore Hanrahan and Pérez-Quiñones [5] hypothesised that the "purposeful act" of composing a message keeps the user more focused in their interaction with the email client [5].

2.1.7 Inbox as a "To-Do" List

Many researchers have found that users keep emails in their inboxes in order to remember to take action on them at a later time – similar to deferral [9], [15]. However, a phenomenon less focused on in research was that of users sending themselves emails [15]. Bota et al. [15] found that 81-92% of users engage in this behaviour and that 32.4-40% of users send two or more emails to themselves per week. The intention of these emails is usually to keep them as reminders or to-do items, however users also stated that they used emails addressed to themselves as a way to: transfer information or documents across devices or accounts, copy and paste information, record or transfer event notes and more [15].

2.2 Review of a Similar Product – Outlook Application

The Outlook Desktop Application was reviewed to learn which of the problems identified in the literature review are addressed in popular email clients.

2.2.1 Organisation

Outlook's default layout consists of the folder pane to the left, the list of messages in the middle and the reading pane to the right. You are able to sort mailboxes and folders by date, sender, recipient, categories, flag status, size etc. You can filter mailboxes and folders by unread mail, flagged mail and mentioned mail. Outlook also allows you to "flag" emails and assign deadlines to them. Similarly you can create categories of different names and colours, and assign emails to these categories. Threads are displayed as a series of connected emails in the email list and replies are listed one after another in the reading pane.

2.2.2 Automation of the Inbox

Outlook offers users a "focused inbox", which is defined as "a feature that uses machine learning to control the overflow of emails by analysing incoming messages and placing the most important emails in the 'Focused' tab, while the rest go to the 'Other' tab" [1]. You are able to set up automatic replies (out of office) in Outlook and create specific rules of when and what to reply. You can also set up more general rules for your whole inbox, for example: move messages from a specific sender to a folder, display mail from someone in the New Item Alert Window or create a new rule from scratch. These rules can be manually enabled and disabled at any time.

2.2.3 Modes

Other than automatic "out of office" replies there is no other option to set the client into a 'mode'. However users could edit the rules in order to achieve the specific behaviours that they would like to achieve at any one time. Although these rules can be manually enabled/disabled at any time, it does not appear that it is possible to automatically enable/disable them at certain times only.

2.2.4 Notifications

Outlook has the option to turn desktop notifications on and off, however there are not many options for the customisation of notifications, which as was mentioned in the literature review, may cause some privacy issues for some users. There is not a way to turn off desktop notifications according to some external factors (e.g. time of day) through the "rules" feature.

2.2.5 To-Do List Features

Outlook has an integrated to-do list feature to which you can assign emails as tasks and handwritten tasks. You have the ability to add high, normal or low priority to the tasks and set a due date. However, there is not a feature that allows you to be reminded about a task at a given date or time. Outlook also has an integrated notes feature, however these are just text based and therefore cannot fully replace the functionality of emails addressed to oneself, for example you cannot add an attachment to a note in order to transfer it across devices.

Chapter 3 - Analysis

This chapter entails how the design requirements were derived. A survey was taken and analysed, personas and user stories were created and used to create the design requirements.

3.1 Survey

The survey received a total of 32 anonymous responses from students of the University of Southampton. As per the ERGO reference ERGO/FEPS/41626.A2, all participants were recruited through social networks and were provided with the 'participants information sheet' and 'consent form' included in the appendices. The purpose of the survey was to investigate which of the main issues identified in the literature review affects users the most on a daily basis.

3.2 Survey Questions

The survey asked the following questions:

1. How would you describe your work or education related email usage?

The participants could choose from three options: 'Light', 'Moderate' and 'Heavy'.

This was asked to gauge how relevant the survey responses would be. The most common response was 'Moderate' with 59.38% of the answers followed by 'Light' and then 'Heavy'.

2. Which tool do you find the most useful for communicating in your work environment?

The options available were 'Email', 'Team Collaboration Software (e.g. Teams or Slack)' and 'Other' where the participants could provide their own suggestions.

This question was asked so that the participants could compare the features of the options available in later questions. The most common response was 'Email' which accounted for 51.61% of the answers, with 'Team Collaboration Software (e.g. Teams or Slack) coming second with 45.16% and one participant suggesting 'in person communication'.

3. Which features of [ANSWER TO QUESTION 2] make it a more useful method of communication?

The respondents could choose from: 'More Convenient', 'Faster', 'Simpler', 'Better for group communication', 'Better for file sharing and management', 'It's what I'm used to' and 'Other' where they could provide their own suggestions.

This question was asked in order to evaluate which features users find the most useful when trying to effectively communicate within their working environment. The most popular answer was 'It's what I'm used to' with 31.11% of the responses. The most popular answers for those respondents that has previously chosen 'Team Collaboration Software' were: 'Better for group communication' and 'Better for file sharing and management'.

4. What feature of [ANSWER TO QUESTION 2] do you find most useful during your working day?

The participants were provided a text box in order to answer this question.

Similarly to the previous question, this was asked to determine which features users find most useful. Common themes within the answers were identified and some of the most common themes found were: Team Collaboration software allows you to share resources easily, such as screen and file sharing; Team Collaboration software was faster and required less formality in the messages; Email is simple and widely used and therefore most people can use it with ease.

5. What feature of [ANSWER TO QUESTION 2] hinders your productivity during your working day?

The participants were provided a text box in order to answer this question.

This was asked to identify features that irritate and hinder users' productivity. Interestingly, notifications were brought up often by respondents who preferred email as well as those who preferred Team Collaboration software. Those who preferred email tended to say that they received too many notifications, especially from unhelpful emails such as newsletters and promotions. Too many notifications were also brought up by those participants who preferred Team Collaboration software, however "buggy" notifications were also cited as an issue.

6. How often do you find yourself overwhelmed with the volume of emails you receive?

The respondents could choose from: 'Never', 'Sometimes', 'About half the time', 'Most of the time' and 'Always'.

This question was asked to evaluate how prevalent the issue of 'Email Overload' is in users' day to day lives. Exactly half of the participants answered with 'Sometimes' and 23.08% responded 'Most of the time'.

7. How easy it is for you to find old emails in your email client?

The respondents could choose from: 'Extremely difficult', 'Somewhat difficult', 'Neither easy nor difficult', 'Somewhat easy' and 'Extremely easy'.

This was asked to better understand the need for better organisation features in email clients. Surprisingly, 40% of respondents answered that it was 'Somewhat easy' to retrieve old emails. Both answers 'Neither easy nor difficult' and 'Somewhat difficult' received 24% of answers. From the research done in the literature review this was expected to be a bigger issue. However since this is an issue affecting a minority of users it will not be prioritised as a feature to be improved.

8. How easy it is to organise your emails using built in annotation methods (e.g. flagging, starring, categorising)?

The respondent could choose from: 'Extremely difficult', 'Somewhat difficult', 'Neither easy nor difficult', 'Somewhat easy' and 'Extremely easy'.

This was asked to determine how valuable these features are to users. 44.00% of respondents answered 'Neither easy nor difficult' and 32% responded 'Somewhat easy'.

9. How often does your work get interrupted by email notifications?

The respondents could choose from: 'Never', 'Sometimes', 'About half the time', 'Most of the time' and 'Always'.

This was asked to understand how useful more fine-tuned notification control would be to users. 44% of respondents answered that email notifications 'Sometimes' interrupts their work and 36% answered that they 'Never' interrupted their work. This was surprising since some of the literature mentioned that users often "get lost" in their emails after being interrupted by an email notification.

3.3 Survey Analysis

The survey was used to prioritise the issues identified in the background research and decide which to improve upon.

Information Overload:

The survey shows that information overload is an issue that affects the majority of respondents and therefore is an issue to improve upon.

Revisiting Emails:

As previously mentioned, it was surprising that many of the respondents had no trouble re-finding emails from their inbox/folders since this was such a prevalent theme in the literature. This could be a reflection on the younger average age of the respondents, since they were all University students. Another potential reason for this could be the improvement of searching algorithms in email clients over the last few years.

Organisation:

Organisation of the inbox and folders was addressed in question 8 of the survey. The majority of participants answered neutrally. This may imply that these features aren't used to their fullest potential.

Automation of Inboxes:

This was a topic that wasn't directly addressed in the survey since this is an area on which many large email providers are focused and a better algorithm or system wouldn't be able to be produced within the time or resource constraints of this project.

‘Modes’ in Email Clients:

This area is the least explored in the most popular email clients today and therefore it was difficult to ask about in the survey. However, respondents’ feedback about notifications can be used and incorporated into the design of the ‘modes’ of the email client.

Notifications:

Fewer people cited notification interruptions as an issue than expected. Therefore notifications alone won’t be the main focus of the project, but since some participants were bothered by interruptions, notification control will be included in the email client.

Inbox as a “To-do” List:

This is another topic that wasn’t directly asked about in the survey since it was felt to be too complex to explain. However, as discovered in the literature review, users will often keep emails in their inboxes or send emails to themselves to remind them to reply or complete a task. Both Outlook and Gmail have support for in-built to-do lists, therefore this isn’t a feature that needs to be improved upon, but better utilised by users.

3.4 Personas

In order to better understand the potential end users of the email client, personas were created. This helped to recognise that different people will have different needs and expectations.

Lorea Oakley – Accountant
Lorea is a 45-year-old accountant, whose work has been made permanently remote since the pandemic. She manages the accounts of multiple clients on a daily basis. Lorea finds that email notifications from her other clients often distracts her from the work she is trying to concentrate on. She doesn't want to turn off her notifications completely because she still needs to see emails that are flagged as important and she would still like to be notified of emails from the client that she is currently working with. Lorea would like a way to filter her notifications so that she is only notified of emails flagged as important and/or emails from specific clients.

Table 1 - Persona 1

Roman Alesio – Part-Time Marketing Consultant
Roman is a 57-year-old marketing consultant that has recently semi-retired. Despite only being paid to work part-time, he often finds himself receiving and reading work emails during his time off, which hinders his ability to relax. Roman struggles with technology and often finds himself confused by the settings and options of most email clients. He would like a simple way to stop notifications outside of his designated working time.

Table 2 - Persona 2

Hira Popov – Architect
Hira is an architect that is often involved in collaborative work. She frequently has to present her designs to others in her office but is bothered by notifications from her personal email accounts coming through when other people are looking at her screen, as this occasionally means that they are able to see sensitive information. Hira would like a quick and simple way to turn off her email notifications when she knows she is going to be presenting to other people.

Table 3 - Persona 3

Tasia Abhishek – Marine Biologist
Tasia is a marine biologist. She has subscribed to many scientific journals using her work email address. She is sent many email notifications from journals with editions and articles that do not relate to her field of study. She doesn't want to unsubscribe to the journals completely because she doesn't want to miss those articles that are related to her work. Tasia would like to receive email notifications from the journals where the subject or body of the email contains at least one specified keyword.

Table 4 - Persona 4

Dan McCrory – Self Employed Fashion Designer
Dan is a self-employed fashion designer who works from home. Due to his ADD, he finds that email notifications often distract him from his work and cause him to "get lost" in his emails. However, Dan also feels overwhelmed when he has a lot of new emails in his inbox. Dan would like to only be notified of his emails when there is a specified number of new emails have arrived in his inbox. This would allow him to dedicate enough time to his current projects while still answering and filing emails before he finds his inbox overwhelming.

Table 5 - Persona 5

3.5 User Stories

User stories were used to capture the main points raised in the personas. These can now be used as objectives in the implementation of the client.

ID	User Story
01	As an Accountant, I want to only be notified of the email that are flagged as important so that I am not distracted from the work I am carrying out for my client.
02	As an Accountant, I want to only be notified of emails from my client so that I am not distracted from the work I am carrying out for them.
03	As an Accountant, I don't want to turn my notifications off completely so that I am not missing important information or tasks.
04	As a semi-retired marketing consultant, I want to stop receiving email notifications outside of my designated working time so that I can relax when I am spending time with my family.
05	As a semi-retired marketing consultant, I want a simple way to alter my setting notifications so that I do not get confused.
06	As an architect I want to turn off my personal email notifications when I am presenting to others so that other people do not see sensitive information through my notifications.
07	As a marine biologist, I want to only receive notifications from emails that contain specified keywords so that I don't have to read emails that are not related to my work.
08	As a self-employed fashion designer, I want email notifications to come through in batches of a specified number so that I don't get overwhelmed with the number of new emails in my inbox.

Table 6 - User Stories

3.6 Design Requirements

The table shows the design requirements derived from the literature review, survey and evaluation of a similar product. Each requirement has a 'T-shirt' size estimation and they have been prioritised according to the MoSCoW prioritisation system. The 'Won't Have' requirements were considered as useful features, however since this project is just a proof of concept to demonstrate how modes can be implemented into email clients, they won't be included.

Design Requirement	Size Estimation	MoSCoW Prioritisation
View a user's emails within a folder	XL	Must Have
View and download attachments from an email	L	Must Have
Create new email	M	Must Have
Add attachments to user's emails	L	Must Have
Send user's email	S	Must Have
Save email as a draft	M	Must Have
Delete emails from inbox/folders	M	Must Have
'Reply', 'Reply All' and 'Forward' Features	L	Must Have
Load user's existing folders	M	Must Have
A 'Do Not Disturb' mode to stop all email notifications	L	Must Have
A 'Do Not Disturb' mode to stop all email notifications until a specified time	L	Must Have
A 'Do Not Disturb' mode to stop all email notifications for a specified amount of time	L	Must Have
A 'Restricted' mode to only allow email notifications from a list of specified senders	L	Must Have
A 'Restricted' mode to only allow email notifications from emails that are tagged as important	L	Must Have
A 'Restricted' mode to only allow email notifications from emails where the subject or body includes some keyword(s)	L	Must Have
A 'Batch Notifications' mode to only notify the user when there is a specified number of new emails in their inbox	L	Must Have
Create new folders	M	Should Have
Delete folders	S	Should Have
File emails into folders	XL	Should Have
A system to remind users to reply to emails	L	Could Have
Searching Feature	XL	Could Have
Email tagging features (e.g. 'Important' or giving emails 'Categories')	L	Could Have
Option to view all attachments in one email thread	M	Won't Have
Creation of rules to organise the inbox	L	Won't Have
An alternative view of email threads	L	Won't Have
Integrated To-Do List	L	Won't Have
Automatic suggestions to add tasks found in to To-Do List	M	Won't Have
Automatic prioritisation of inbox	XL	Won't Have
Automatic organisation of inbox e.g. into folders	L	Won't Have

Table 7 - Design Requirements

3.7 Use Case Diagram

The following use case diagram was created to help visualise how the system will be used and interacted with. It is based on the design requirements identified in the previous section. The user is the primary stakeholder in the system and the only one who will be interacting with the system. The colours of each use case corresponds to their MoSCoW prioritisation. The 'extends' relationship show that a use case extends the behaviour of another.

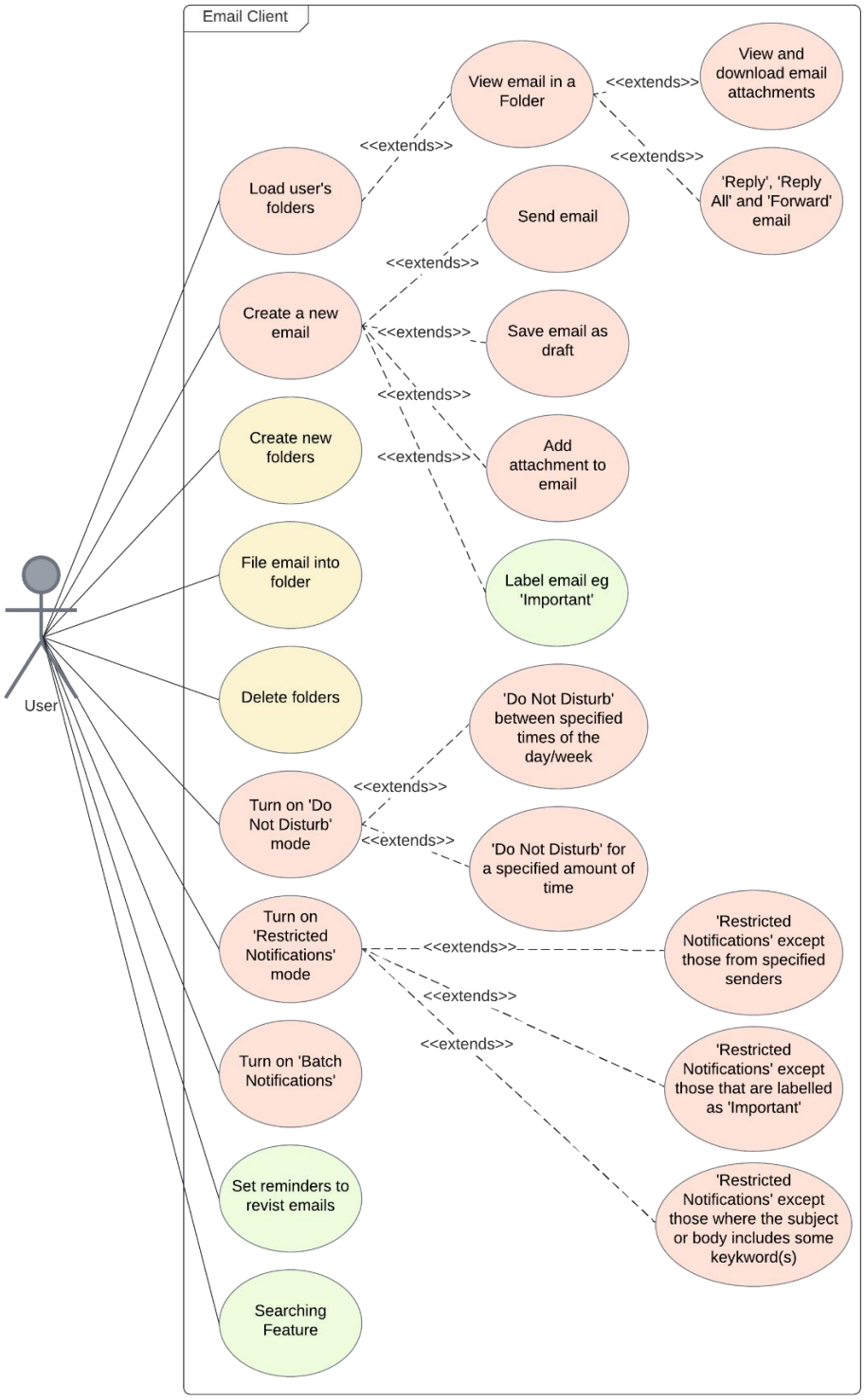


Figure 1 - Use Case Diagram

Chapter 4 - Design

This chapter describes the design of the system including the structure of the code and the user interface.

4.1 Final Design of the System

Informed by the literature review, survey and evaluation of a similar product, the project will be focussed on the creation of modes in an email client. This has been chosen because of all the areas of potential improvement identified, modes that can finely tune notifications depending on specific users desires is the least explored in popular email clients. By developing modes, the email client could help users to reduce the feeling of 'email overload' when interrupted by email notifications. The email client will include four modes:

- Normal Mode – In this mode the user's inbox will update according to a specified 'Sync Frequency' and they will receive notifications accordingly

- Do Not Disturb Mode – in this mode the user will have the option to:
    - Turn off email notifications until turned back on
    - Turn off email notifications until a specified time
    - Turn off email notifications for a predefined amount of time (e.g. for the next hour)

- Restricted Notifications Mode (Holiday Mode) – in this mode you will have the option to:
    - Allow notifications from a specified sender list
    - Allow notifications from emails labelled as 'Important'
    - Allow notifications from emails where the subject or body includes some keyword(s)

  This mode was named 'Holiday' as it would be most useful when users would like to take time away from their work, whilst still being informed on urgent matters or topics of interest.

- Batch Notifications Mode (Concentrated Mode) – in this mode you will be able to allow a notification when there is a specified number of new and unread emails in your inbox.

  This mode was named 'Concentrated' as this feature will allow the user to focus on work outside of their emails and therefore reducing the negative effects of email interruptions. However, as the user is notified when the number of new emails reaches their threshold, it should help the user to reduce the feeling of 'email overload' – where the user feels overwhelmed at the number of new emails in their inbox.

Java and Microsoft's Mail Outlook API will be used to create the email client. In order to create the UI, JavaFX, FXML and Scene Builder will be used, which will allow quick and efficient UI development and leave more time for backend development. Maven will be used to automate the building of the project and manage the project's dependencies. Microsoft's Mail Outlook API will be accessed through Microsoft Graph. There are two versions of Microsoft Graph – V1 and Beta, for this project V1 will be used because it is production ready and has all the capabilities required.

To call the Mail API, the application needs an access token from the Microsoft Identity Platform. Microsoft Authentication Library for Java (MSAL4J) will be integrated into the application and this allows it to easily obtain an OAuth2 access token which can, in turn, be used to access Microsoft Graph. In order to integrate Microsoft Identity Platform into the application, the application has to be registered with Microsoft Azure Active Directory, which generates a unique application ID and a redirect URI. The Azure Active Directory is used to configure which permissions the application will require. The full authentication process is displayed in the Sequence Diagram (Figure 12).

When the user and application have been authenticated, the email client will be loaded. The email client will be initialised with the user's folders and email messages by sending an HTTP GET request to the Microsoft Graph Outlook Mail API and these will be displayed to the user. When the user interacts with the application, it will continue to interact with the Microsoft Graph Outlook Mail API using HTTP requests. For example, if the user chooses to create a new email message and save it as a draft, the application will send an HTTP POST request to the Microsoft Graph Outlook Mail API with the new message object and it will be saved in the user's 'Drafts' folder. This process and more are displayed on the Sequence Diagram (Figure 12).

4.2 Storyboarding the Key Design Choices

In order to plan out the user interface, wireframes were created on Balsamiq.

Main Screen – Home Tab:

This is the design for the main screen of the email client – it will be displayed after the user has logged in. It takes inspiration from main stream clients such as Outlook. From the survey, users expressed that they used email because it was what they were used to and therefore redesigning the layout of traditional email clients was unnecessary.

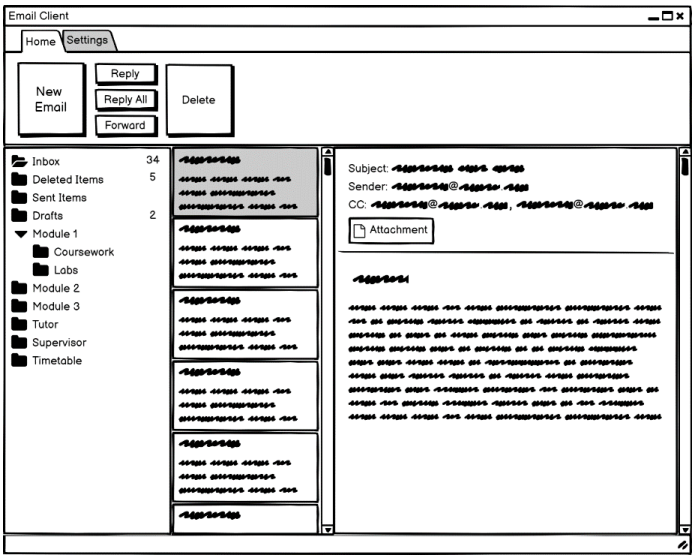


Figure 2 - Main Screen Home Tab

Main Screen – Settings Tabs:

This wireframe depicts the settings tab where users can navigate to in order to change which mode the email client is currently in. Since this project shows a proof of concept, other settings haven't been included in this tab, however this is where those options would be placed if this email client was to be developed in to a viable product.

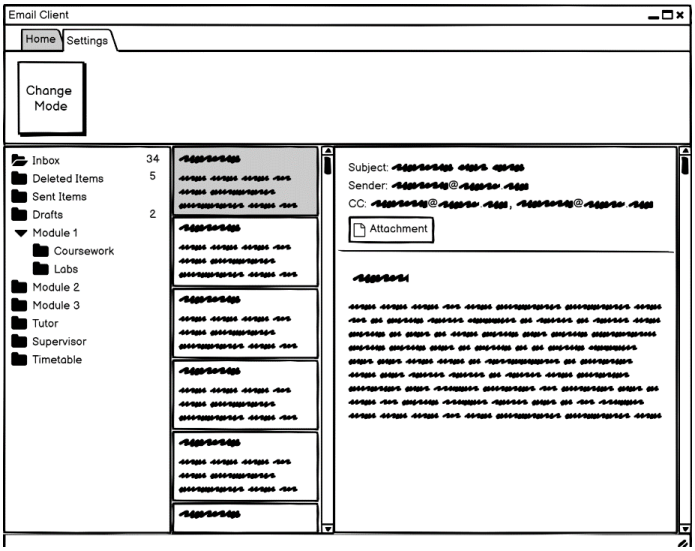


Figure 3 - Main Screen Settings Tab

Change Mode Dialog:

This is the dialog that users will be shown when they click the ‘Change Mode’ button from the settings tab of the home screen.

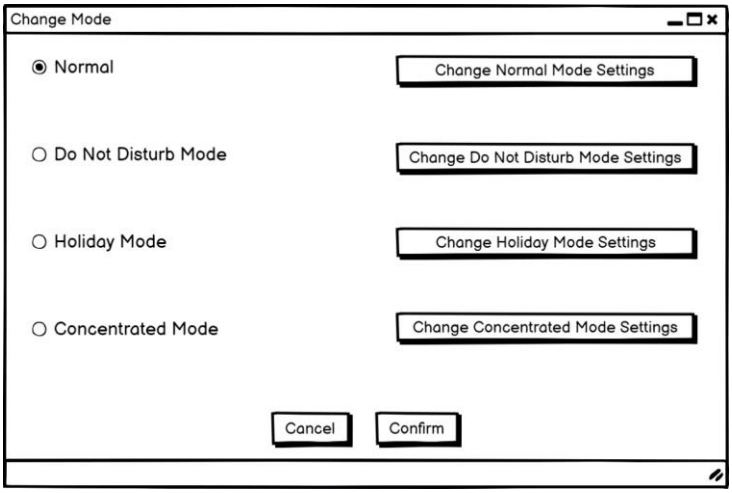


Figure 4 - Change Mode Dialog

Normal Mode Settings Dialog:

This is the dialog the users will be shown after clicking the ‘Change Normal Mode Settings’ button from the ‘Change Mode’ dialog. This is where users can edit how often the email client will sync with the server.

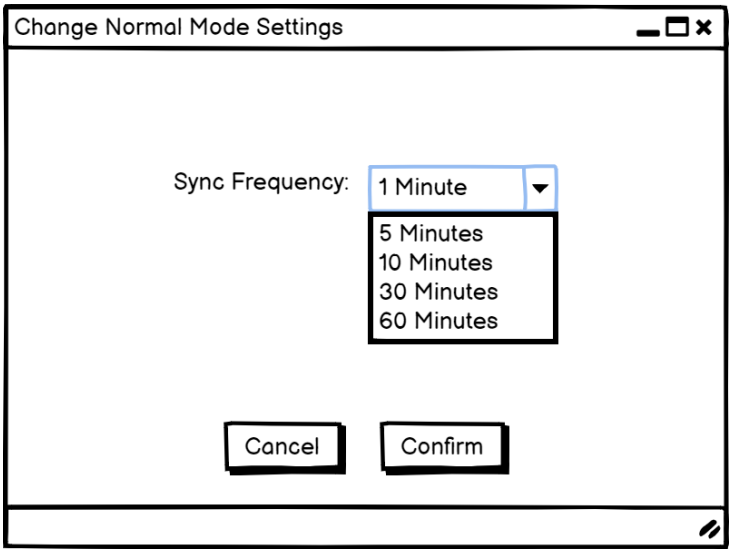


Figure 5 - Normal Mode Settings Dialog

Do Not Disturb Mode Settings Dialog:

This is the dialog that will be shown to users after they click the ‘Change Do Not Disturb Mode Settings’ on the ‘Change Mode’ dialog. Users will have the option to turn on ‘Do Not Disturb Mode’ until they turn it off and they will not receive any notifications during this time. They can also choose to turn it on for one of the predefined time periods: 1 hour, 8 hours or 24 hours.

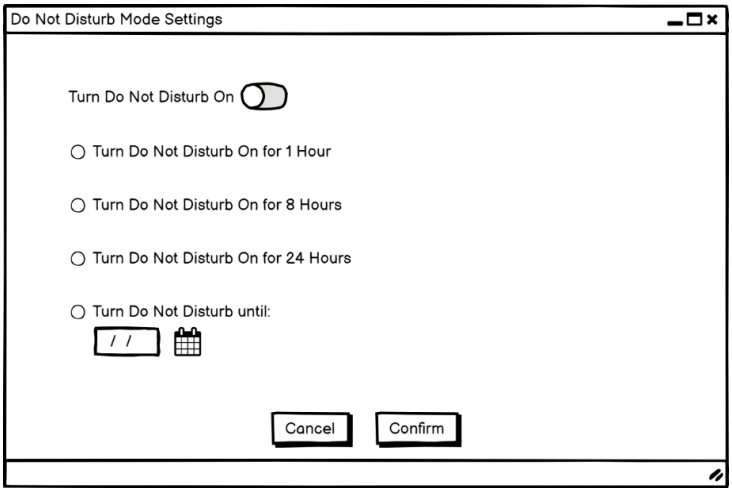


Figure 6 - Do Not Disturb Mode Settings Dialog

For more finely tuned control, the user can select a specific time and date that ‘Do Not Disturb Mode’ will be turned off.

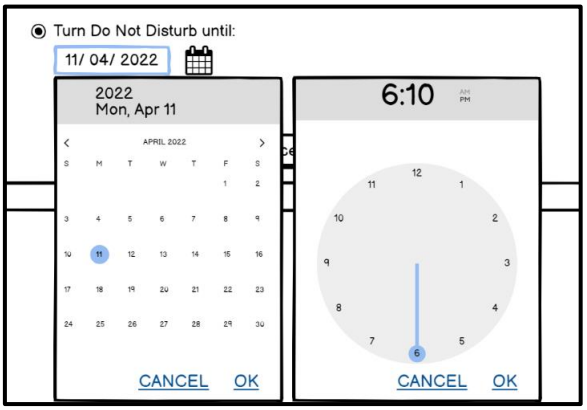


Figure 7 - Do Not Disturb Mode Settings Date Picker

Holiday Mode Settings Dialog:

This is the dialog that the user will be shown after they click the 'Change Holiday Mode Settings' on the 'Change Mode' dialog. Holiday mode will allow the user to choose to receive notifications from specific senders, emails that contain specified keywords in the subject or body and/or emails that are labelled as important. Therefore the user can select one or more of these options to apply the rules in combination.

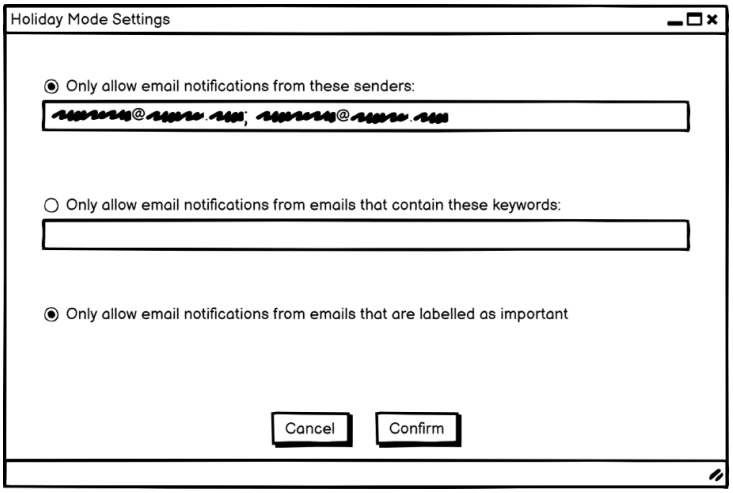


Figure 8 - Holiday Mode Settings Dialog

Concentrated Mode Settings Dialog:

This is the dialog that the user will be shown after they click the 'Change Concentrated Mode Settings' on the 'Change Mode' dialog. In 'Concentrated Mode', the user can choose how many new emails can arrive in their inbox before they are notified and therefore this dialog will display a number spinner where the user can specify this number

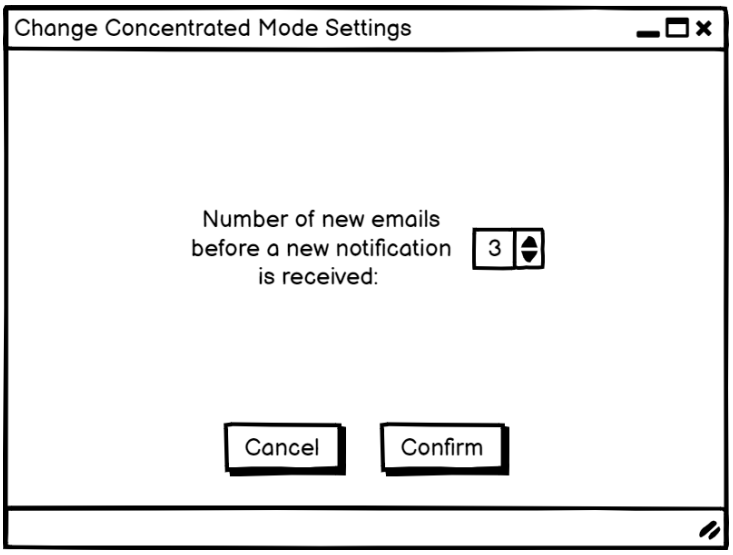


Figure 9 - Concentrated Mode Settings Dialog

4.3 Design Architecture – MVC Approach

To ensure the code for the project is easy to maintain, the Model-View-Controller (MVC) architectural approach will be implemented. This approach separates the 'Model' and 'View' completely and all interactions between them happen through the 'Controller'. The 'Model' in the application is responsible for all interactions with Microsoft Graph, which calls the Outlook Mail API. To create the 'View' of the application JavaFX and FXML will be used to create the UI, and this will be separate from the logic of the rest of the application. The 'Controller' of the application handles all interactions between the 'Model' and 'View' and triggers updates for both 'Model' and 'View'.

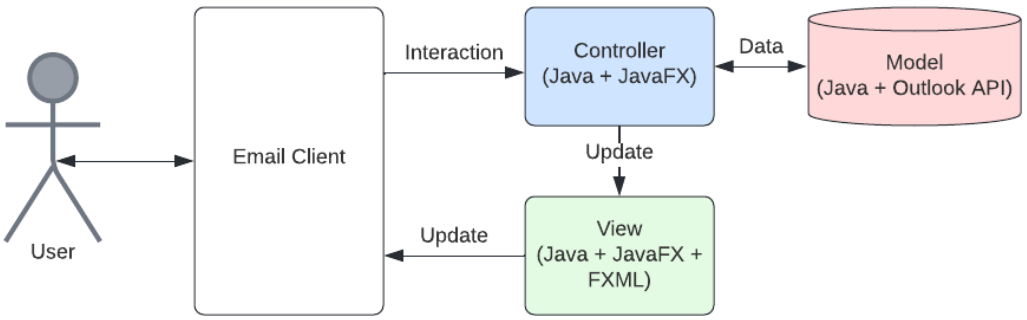


Figure 10 - MVC Diagram

4.4 UML Class Diagram

A UML diagram was created in order to plan out the structure of the code. In accordance with the MVC diagram (Figure 10 – MVC Diagram), the UML class diagram below shows the 'Model' classes in pink, the 'View' class in green and the 'Controller' classes in blue. In addition to these, the diagram also shows the resources used by the application in purple.

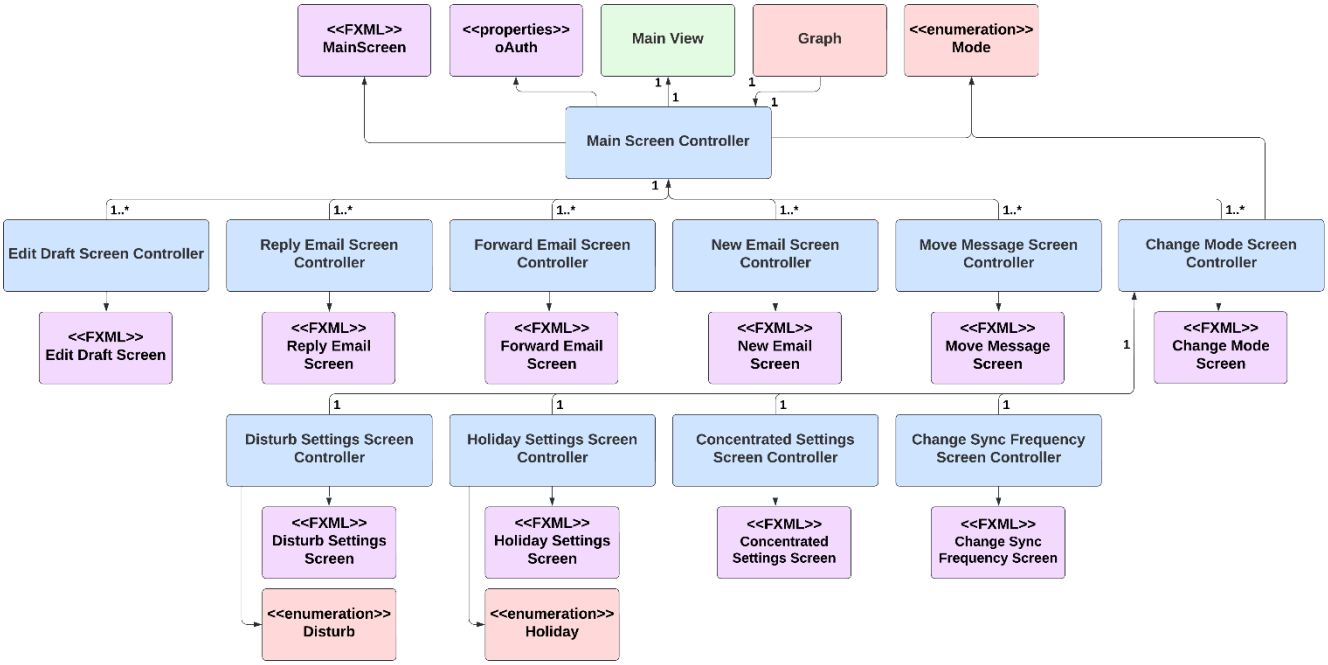


Figure 11 - UML Class Diagram

4.5 Full System Sequence Diagram

This sequence diagram describes the interactions between the entities in the system. There is only one actor in the diagram as the user will be the only one interacting with the system. The sequence diagram specifies how the access token is obtained when the user logs into the system. During the creation of this diagram the Microsoft Azure Documentation [16] and a YouTube video explaining OAuth Authorisation code flow for Microsoft Graph [17] were referenced. This diagram shows a small sample of actions the user can take and is by no means exhaustive.

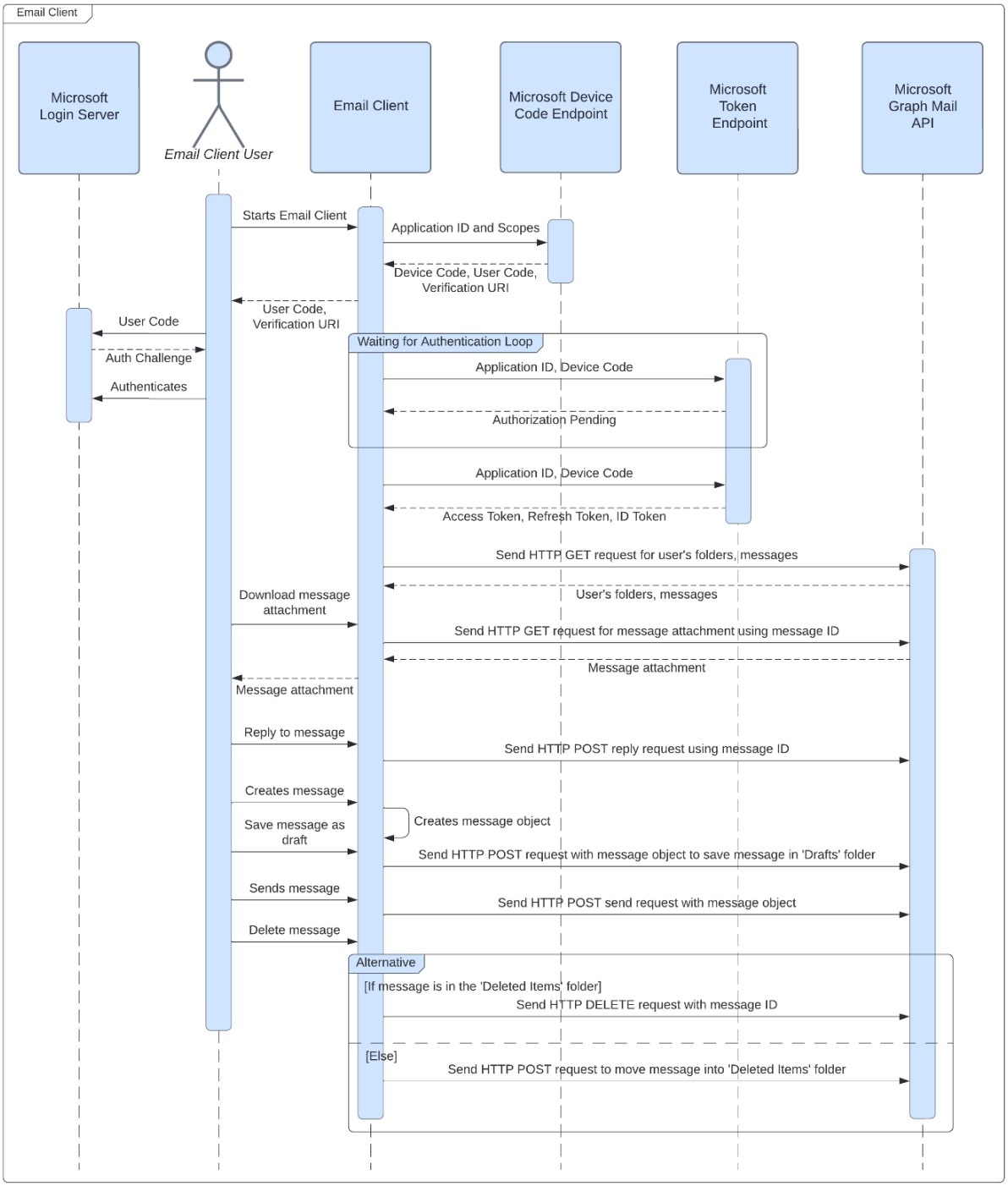


Figure 12 – Full System Sequence Diagram

Chapter 5 - Implementation of the Application

This chapter explains how the application was implemented, including the logic for the modes using pseudocode and how the UI was implemented.

5.1 Environment and Tools Used

IntelliJ was the chosen IDE for this project as it includes excellent support for Java and Maven, as well as great integration for GitHub version control. Microsoft Graph was used to access Microsoft's Outlook Mail API. Scene Builder was used to quickly generate a UI in FXML and Java was the language used to build the application. The project was version controlled using GitHub and therefore was easily accessible if the data or tools used for development were lost.

During development, the following tutorials were referenced: How to Open a File in Java [18]; Build Java apps with Microsoft Graph [19]; How to Use Regular Expressions to Replace Tokens in String in Java [20]; Preserving Line Breaks When Using Jsoup [21].

5.2 Final Design

The following sections display the main screens of the application created. When adding CSS to the application the JavaFX CSS Reference Guide was referenced [22].

5.2.1 Logging into the Application

When the application is started the user is shown a user code and a link. When entering the user code into the link provided they are then redirected to a Microsoft log in page. Once logged in, they are asked to consent to the permissions required for the application. After having granted the permissions the application launches.

5.2.2 The Main Screen of the Application

Once the application is launched, the user is met with the main screen of the application:

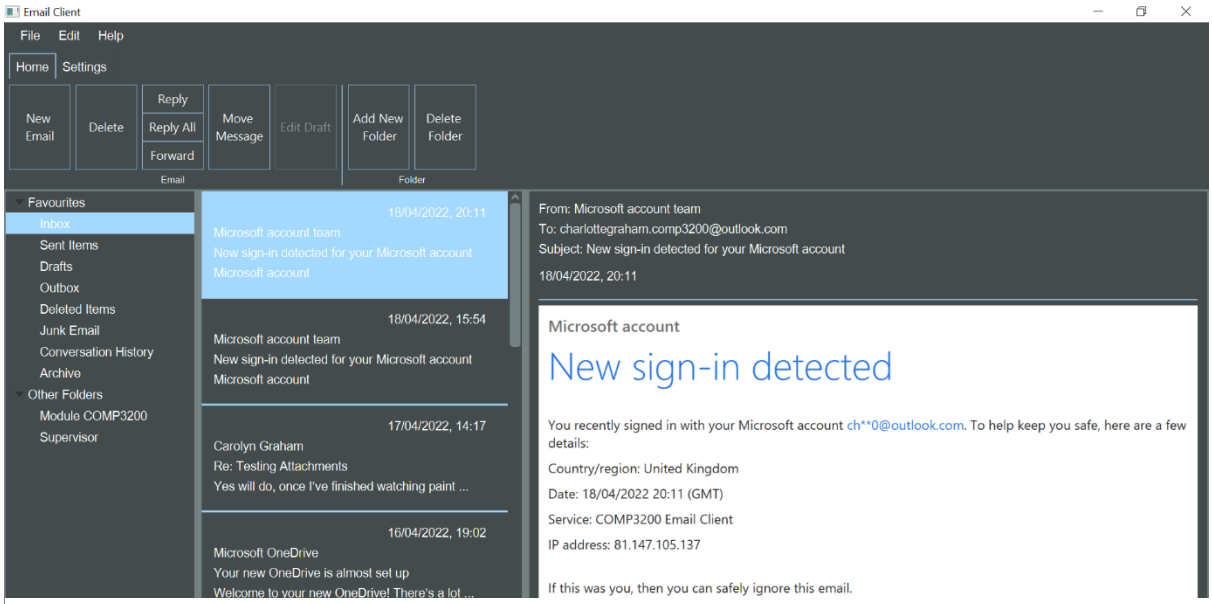


Figure 13 - Main Screen of Application

5.2.3 New Email Screens

If the user clicks on the 'New Email' button on the main screen they will be shown this screen:

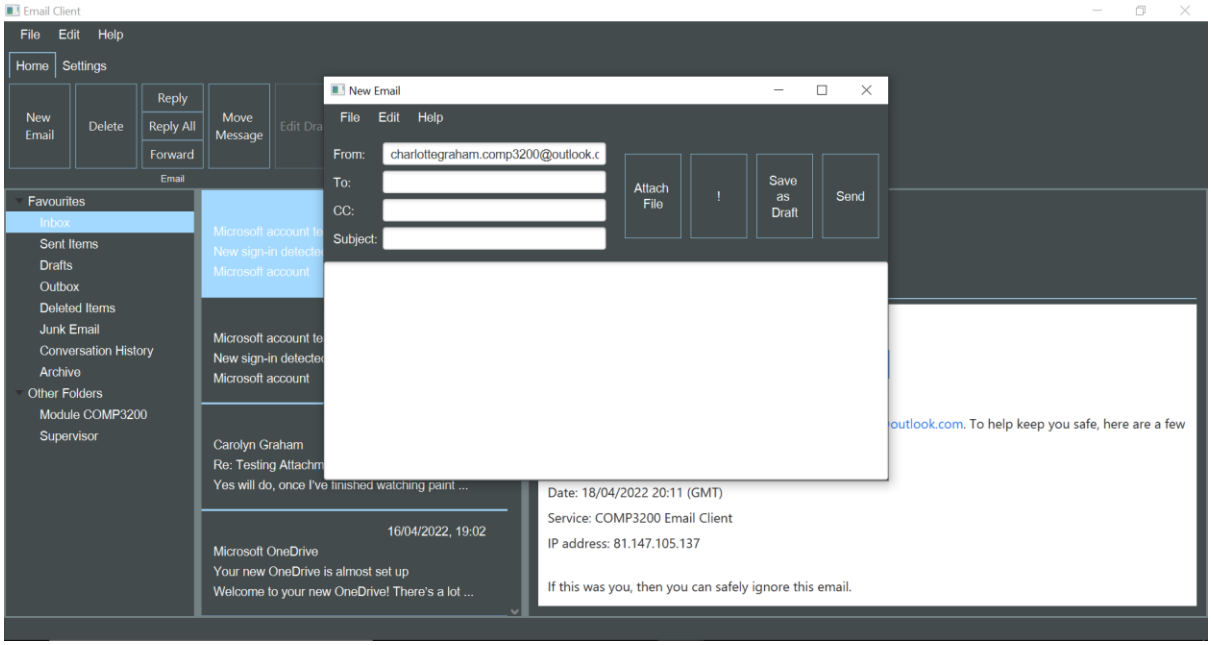


Figure 14 - New Email Screen of Application

The user can attach a file of their choosing, label the email as important, save it as a draft or send the email. A similar process happens for the replying and forwarding of emails:

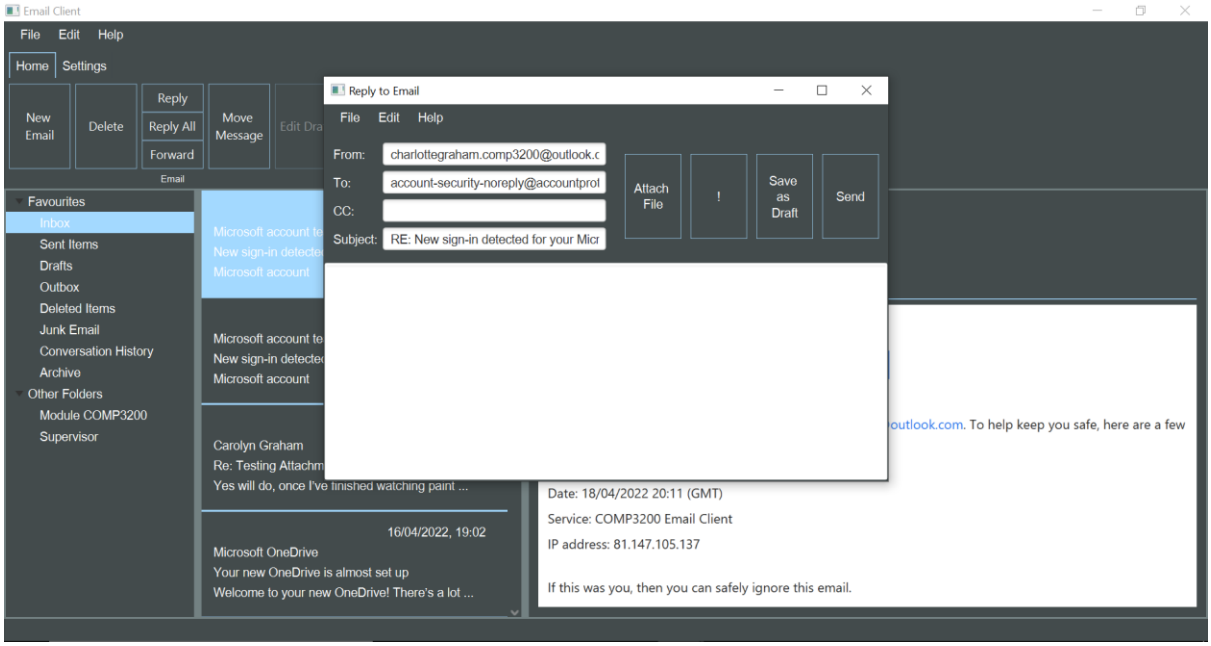


Figure 15 - Reply to Email Screen of Application

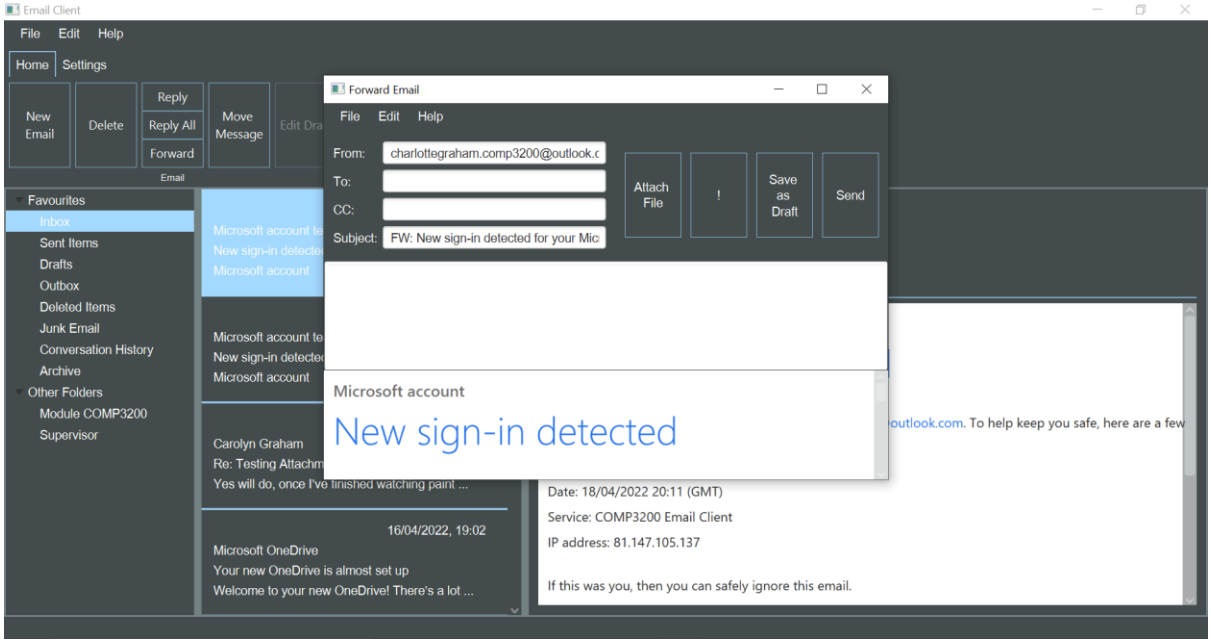


Figure 16 - Forward Email Screen of Application

5.2.4 Change Mode Screen

When the user navigates to the 'Settings' tab and clicks on the 'Change Mode' button they are shown this screen:

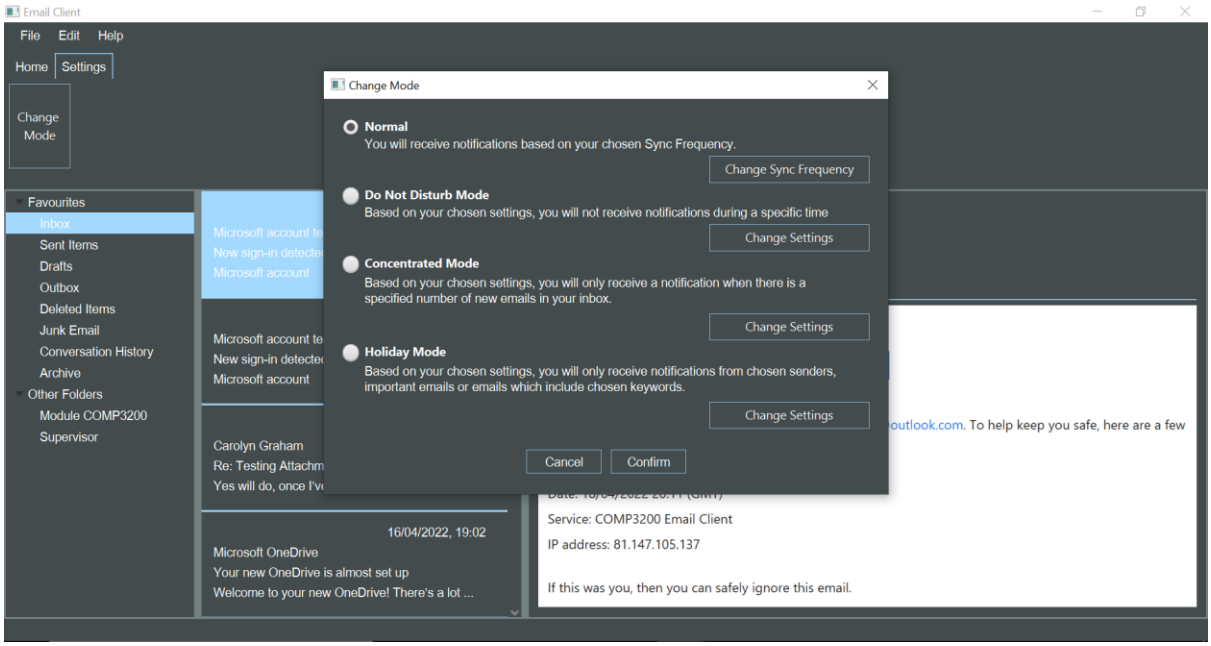


Figure 17 - Change Mode Screen of Application

This is where the user can choose which mode they would like the email client to be in.

5.2.5 Normal Mode Settings

If the user clicks on the 'Change Sync Frequency' button they will be shown this dialog:

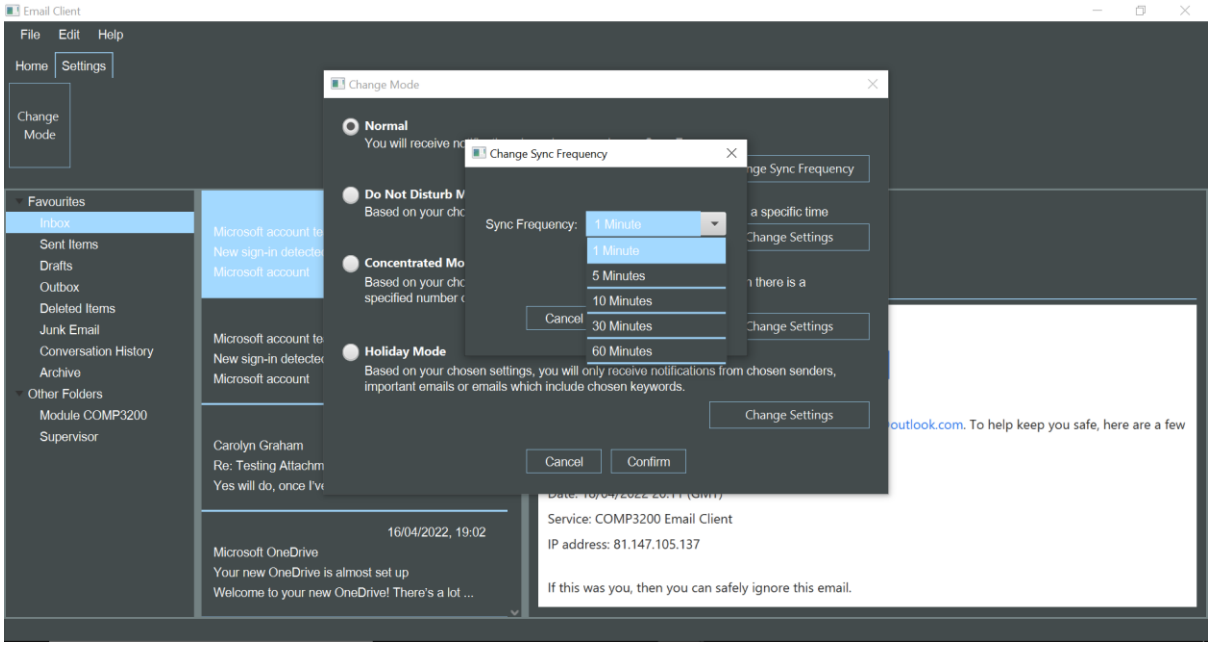


Figure 18 - Normal Mode Settings Screen of Application

The user can choose how often they would like the email client to sync with the server when the client is in 'Normal' mode.

5.2.6 Do Not Disturb Mode Settings

From the 'Change Mode' window, the user can click on the 'Change Settings' button underneath the 'Do Not Disturb Mode' radio button and they will be shown this screen:

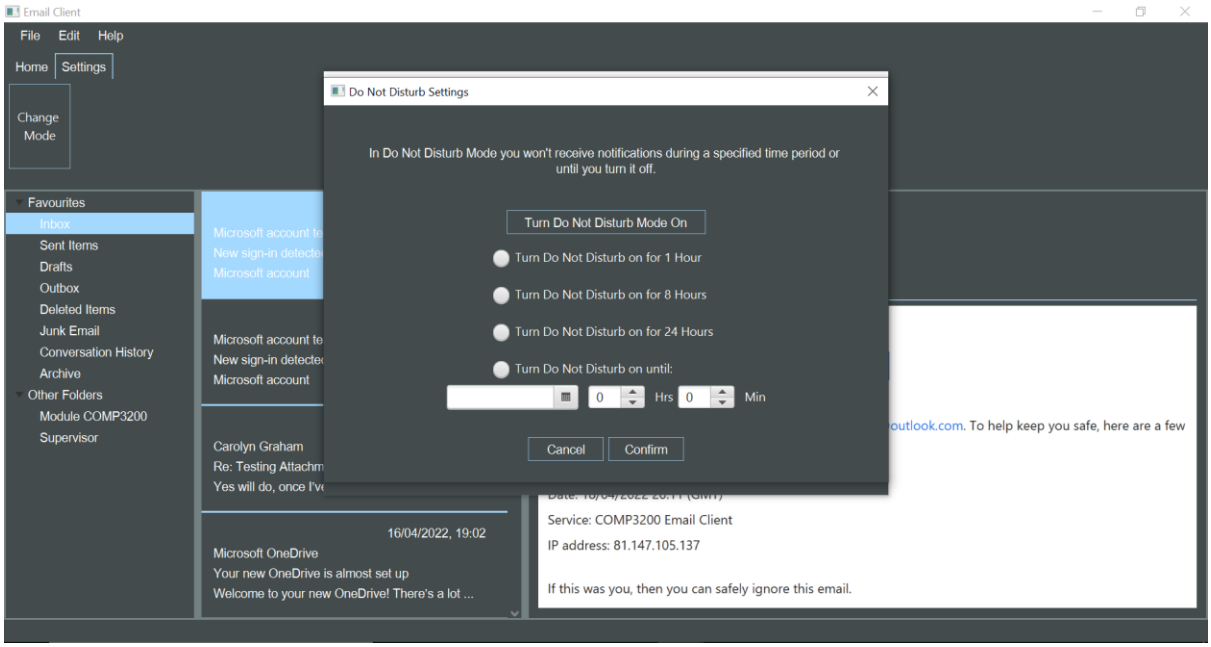


Figure 19 - Do Not Disturb Mode Settings Screen of Application

From this window the user can choose to turn ‘Do Not Disturb Mode’ on with toggle button, choose one of the radio buttons that turns ‘Do Not Disturb Mode’ for one of a specified amount of time or choose the radio button that allows them to specify when ‘Do Not Disturb Mode’ will be turned off.

5.2.7 Concentrated Mode Settings

From the ‘Change Mode’ window, the user can click on the ‘Change Settings’ button underneath the ‘Concentrated Mode’ radio button and they will be shown this screen:

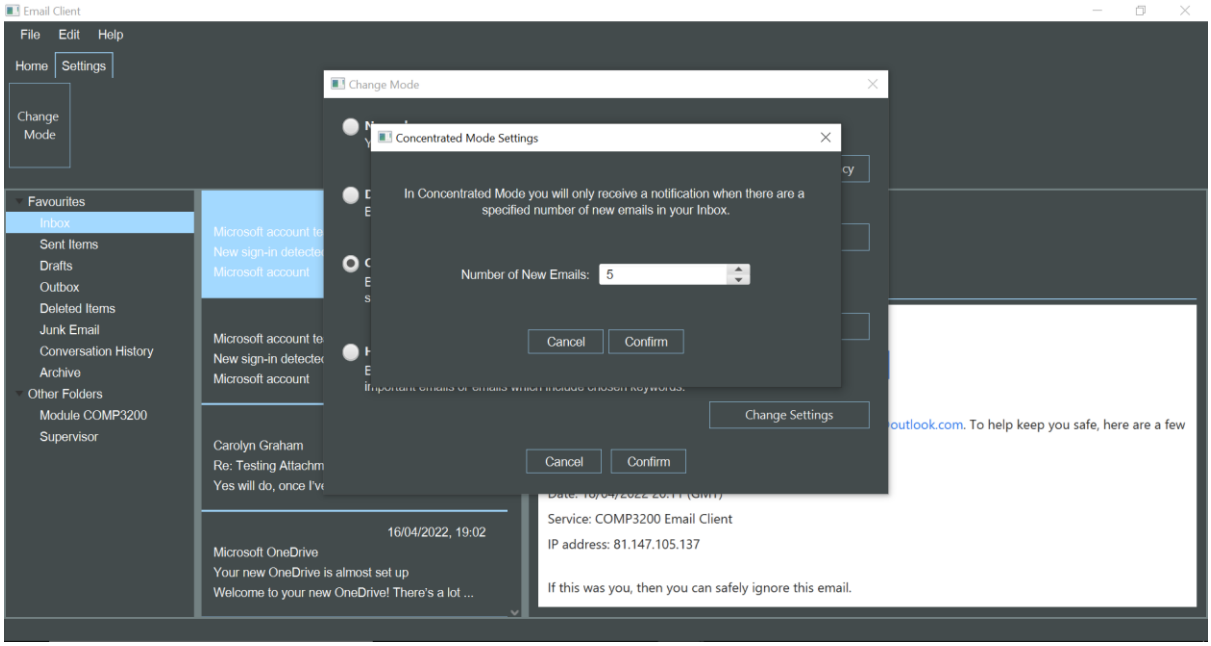


Figure 20 - Concentrated Mode Settings Screen of Application

From this window the user can choose how many emails they will receive before they are notified.

5.2.8 Holiday Mode Settings

From the ‘Change Mode’ window, the user can click on the ‘Change Settings’ button underneath the ‘Holiday Mode’ radio button and they will be shown this screen:

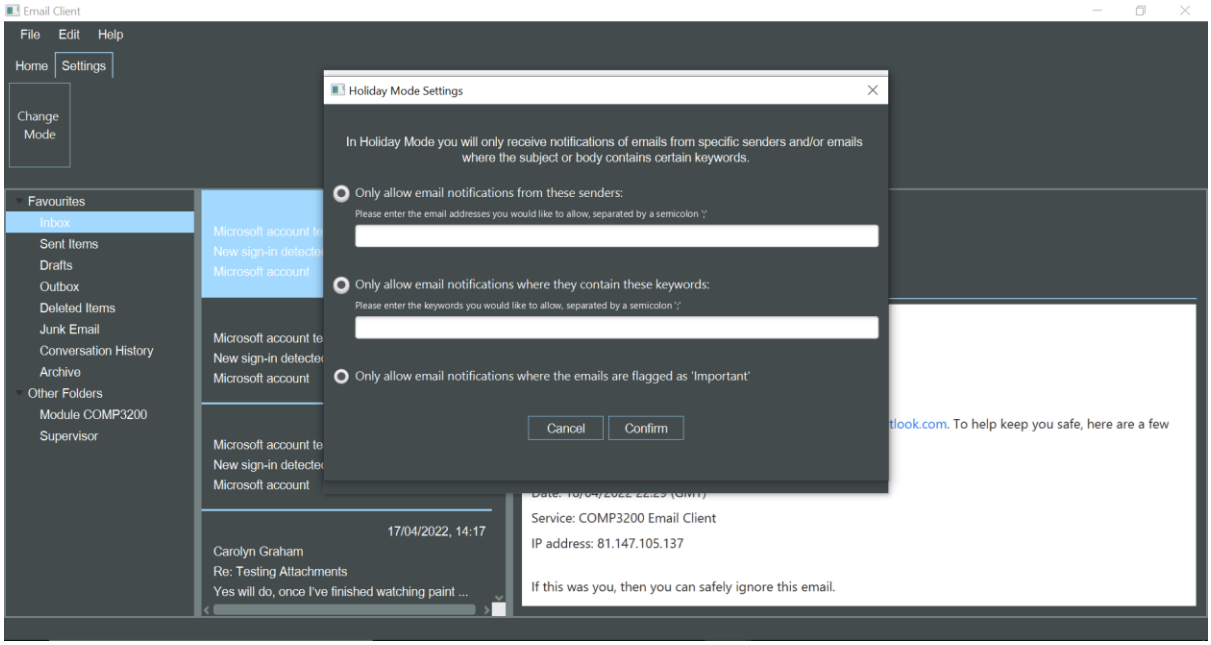


Figure 21 - Holiday Mode Settings Screen of Application

For this window the user can choose to receive email notifications from specified email addresses, emails that contain specified keywords, emails that are labelled as ‘Important’ or any combination of the three.

5.3 Pseudocode for the Mode Algorithms

The follow sections contain the pseudocode for the algorithms implemented to the application to create the modes.

5.3.1 Normal Mode

In 'Normal Mode' the user's inbox syncs according to their chosen 'Sync Frequency'. The can choose between:

- 1 Minute
- 5 Minutes
- 10 Minutes
- 30 Minutes
- 60 Minutes

The default value is 1 minute as shown in the pseudocode below:

```
#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in normal mode
    If mode == NORMAL
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0
            Foreach message in difference
                sendNotification(message)
            Endfor

            #update inbox contents
            inbox.setContents(graphMessages)
        Endif
    Endif
Endtimer
```

Figure 22 - Normal Mode Pseudocode

5.3.2 Do Not Disturb Mode

When selecting Do Not Disturb Mode the user can choose between:

- Turning Do Not Disturb Mode on until they turn it off
- Turning Do Not Disturb Mode on for a specified amount of time:
    - 1 Hour
    - 8 Hours
    - 24 Hours
- Turning Do Not Disturb Mode on until a specified date and time

The following three subsections show the pseudocode for the three options.

5.3.2.1 Do Not Disturb Mode On

When Do Not Disturb Mode is turned on until further notice, the user will not receive any email notifications, however the inbox will continue to sync according to the sync frequency.

```
#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in Do Not Disturb Turned on mode
    If mode == DISTURB.ON
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0
            #update inbox contents but don't send notifications
            inbox.setContents(graphMessages)
        Endif
    Endif
Endtimer
```

Figure 23 - Do Not Disturb Mode On Pseudocode

5.3.2.2 Do Not Disturb Mode on for Specified Period of Time

When Do Not Disturb Mode is turned on for a specified period of time, the user can choose from these options:

- Turn Do Not Disturb on for 1 hour
- Turn Do Not Disturb on for 8 hours
- Turn Do Not Disturb on for 24 hours

In this pseudocode, the user has chosen to turn Do Not Disturb on for 1 hour and therefore 3600000ms is assigned to the variable 'turnedOnFor'.

```
#how long DO NOT DISTURB Mode is until on for in milliseconds
#depends on user selection
turnedOnFor <- 3600000

#when DO NOT DISTURB Mode should be turned off
turnedOnUntil <- currentTime + turnedOnFor

#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in DO NOT DISTURB Turned on mode
    If mode == DISTURB.TIMED
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0
            #update inbox contents but don't send notifications
            inbox.setContents(graphMessages)
        Endif

        #if current time is more than or equal to when DO NOT DISTURB
        Mode should be turned off
        If turnedOnUntil >= currentTime
            #mode gets set back to NORMAL
            mode = NORMAL
        Endif
    Endif
Endtimer
```

Figure 24 - Do Not Disturb Mode On for a Specified Period of Time Pseudocode

5.3.2.3 Do Not Disturb Until a Specified Date and Time

When the user choses to turn Do Not Disturb on until a chosen date and time, the application shows the user a date picker, an hour selector and a minute selector. The application then convert this to a Date object which is represented in this pseudocode by the variable 'turnedOnUntil'.

```
#when DO NOT DISTURB Mode should be turned off
#depends on user selection
turnedOnUntil <- selectedTime

#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in DO NOT DISTURB Turned on mode
    If mode == DISTURB.UNTIL
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0
            #update inbox contents but don't send notifications
            inbox.setContents(graphMessages)
        Endif

        #if current time is more than or equal to when DO NOT DISTURB
        Mode should be turned off
        If turnedOnUntil >= currentTime
            #mode gets set back to NORMAL
            mode = NORMAL
        Endif
    Endif
Endtimer
```

Figure 25 - Do Not Disturb Mode On Until a Specified Date and Time Pseudocode

5.3.3 Holiday Mode

When selecting Holiday Mode the user can choose to only allow email notifications:

- From a list of specified senders
- Where emails contain at least one of their specified keywords
- Where emails are labelled as 'Important'
- Or any combination of the three

The following three sections show the pseudocode for the three options.

5.3.3.1 Holiday Mode for Specified Senders

When the user chooses to be notified when they receive emails from specific senders, the application allows them to enter a semi-colon-separated list and this is converted into an array of email addresses, which is represented by the 'senders' variable in this pseudocode.

```
#list of allowed senders
#depends on user input
senders[] <- textInput

#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in HOLIDAY MODE
    If mode == HOLIDAY.SENDERS
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0
            Foreach message in difference

                #send notification if message is from one of the
                specified senders
                If senders.contains(message.getSender())
                    sendNotification(message)
                Endif
            Endfor

            #update inbox contents
            inbox.setContents(graphMessages)
        Endif
    Endif
Endtimer
```

Figure 26 - Holiday Mode for Specified Senders Pseudocode

5.3.3.2 Holiday Mode for Specified Keywords

When the user chooses to be notified when they receive emails that contain specific keywords, the application allows then to enter a semi-colon-separated list and this is converted to an array of strings. In this pseudocode the array of keywords is represented by the 'keywords' variable.

```
#list of allowed keywords
#depends on user input
keywords[] <- textInput

#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in HOLIDAY MODE
    If mode == HOLIDAY.KEYWORDS
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0
            Foreach message in difference

                #for each specified keyword check if it exists in the
                message
                Foreach keyword in keywords

                    #send notification if keyword exists in email and
                    break loop
                    If (message.getContent()).contains(keyword)
                        sendNotification(message)
                        BreakFor
                    Endif
                EndFor
            Endfor

            #update inbox contents
            inbox.setContents(graphMessages)
        Endif
    Endif
Endtimer
```

Figure 27 - Holiday Mode for Specified Keywords Pseudocode

5.3.3.3 Holiday Mode for Important Emails

When the user chooses to be notified of important emails, the application will check each new email received for an 'Important' label and only notify the user if it is present, as shown by the pseudocode below.

```
#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in HOLIDAY MODE
    If mode == HOLIDAY.IMPORTANT
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0
            Foreach message in difference

                #send notification if message is labelled Important
                If message.IsImportant() == true
                    sendNotification(message)
                Endif
            Endfor

            #update inbox contents
            inbox.setContents(graphMessages)
        Endif
    Endif
Endtimer
```

Figure 28 - Holiday Mode for Important Email Pseudocode

5.3.4 Concentrated Mode

When selecting Concentrated Mode the user can specify how many new emails they would to receive before they are notified. The application allows the user to choose this number through a number spinner and this number is then assigned to the 'notificationThreshold' variable. In this example, the user has chosen to be notified when they receive 10 new emails.

```
#how often the email client syncs the inbox in milliseconds
syncFrequency <- 60000

#how many new emails the user will receive before being notified
#depends on user input
notificationThreshold <- 10

#new email counter
emailCounter <- 0

#timer carries out task in intervals of given argument
Timer(syncFrequency)

    #if the email client is currently in HOLIDAY MODE
    If mode == CONCENTRATED
        #list of messages current in inbox
        inboxMessages <- inbox.contents()

        #list of messages in inbox called from Microsoft Graph
        graphMessages <- Graph.inboxContents()

        #list difference
        difference <- graphMessages - inboxMessages

        #if difference is more than 0 there are new emails
        If difference.size() > 0

            #increment emailCounter with number of new emails
            emailCounter = emailCounter + difference.size()

            #if number of new emails received is more than or equal to
            the user's threshold then send notification
            If emailCounter >= notificationThreshold
                sendNotification("You have new emails")

                #set email counter back to 0
                emailCounter <- 0
            Endif

            #update inbox contents
            inbox.setContents(graphMessages)
        Endif
    Endif
Endtimer
```

Figure 29 - Concentrated Mode Pseudocode

Chapter 6 - Testing

Since the Email Client Application uses Microsoft's Graph Outlook Mail API to access the user's Outlook email account, it was very difficult to write unit tests since the application could potentially be different every time it is viewed. It is recommended that unit testing should cover 80% of code written [23] and since this would be very difficult to achieve, the application was tested manually.

6.1 Testing Using Scenarios

Some realistic scenarios that the personas might experience while using the Email Client application were created. The scenarios and tests are described below:

Lorea Oakley
• Lorea is a 45-year-old accountant who would like to filter her email notifications so that she is only notified of emails that are labelled as important and/or emails from specified senders • Lorea logs into the email client application using her Outlook account • Lorea clicks on the 'Settings' tab and clicks on the 'Change Mode' button • The 'Change Mode' dialog is shown on the screen and after reading the descriptions of the different modes, Lorea navigates to the 'Holiday Mode Settings' button • The 'Holiday Mode Settings' dialog is shown on the screen • Lorea selects both the radio buttons: 'Only allow email notifications from these senders' and 'Only allow email notifications where the emails are flagged as 'Important'' and enters the email addresses she would like to receive notifications from • Lorea confirms her choice on the 'Holiday Mode Settings' dialog • Lorea confirms her choice on the 'Change Mode' dialog • She will now only receive notifications of emails from her specified senders list and those emails that are flagged as important
Corresponding User Stories
01, 02, 03
Tests Performed

Type of Test: Manual **Preconditions:** Email Client is not yet loaded and user is not yet logged in	• All buttons work as expected • Choices for settings were saved • Email notifications were only received when emails were from specified senders or labelled 'Important'

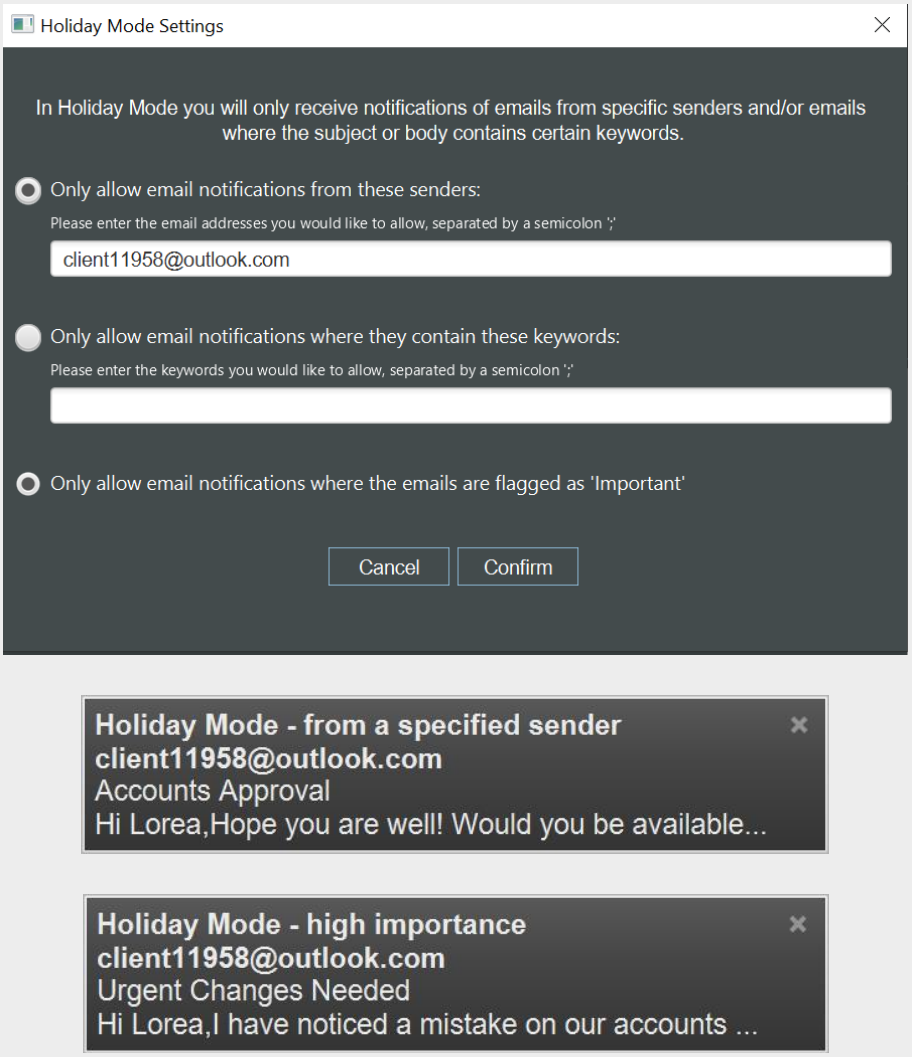
Screenshots


Table 8 - Scenario Test 1

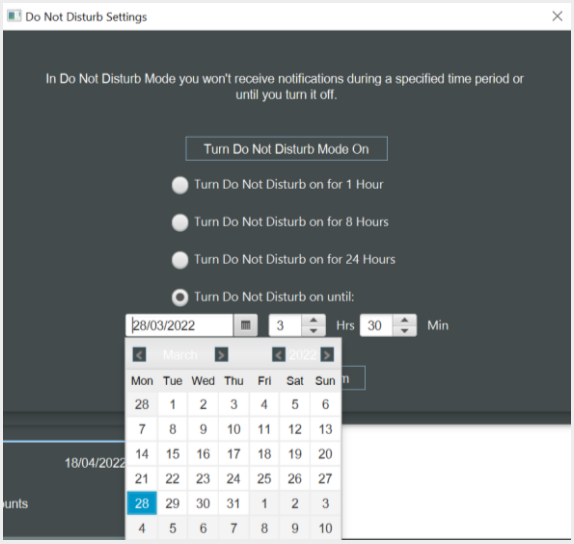
Roman Alesio
• Roman is a 57-year-old marketing consultant who would like to stop receiving email notifications outside of his working hours • Roman logs into the email client application using his Outlook account • Roman clicks on the 'Settings' tab and clicks on the 'Change Mode' button • The 'Change Mode' dialog is shown on the screen and after reading the descriptions of the different modes, Roman navigates to the 'Do Not Disturb Mode Settings' button • The 'Do Not Disturb Mode Settings' dialog is shown on the screen • Roman selects the radio button that allows him to choose when he will start receiving notifications again • Roman selects the date and time he would like to start receiving notifications again using the date and time pickers • Roman confirms his choice on the 'Do Not Disturb Mode Settings' dialog • Roman confirms his choice on the 'Change Mode' dialog • He will now not receive email notifications until the date and time he has chosen
Corresponding User Stories
04, 05
Tests Performed
Type of Test: Manual **Preconditions:** Email Client is not yet loaded and user is not yet logged in • All buttons work as expected • Choices for settings were saved • Email notifications were only received after the specified date and time
Screenshots


Table 9 - Scenario Test 2

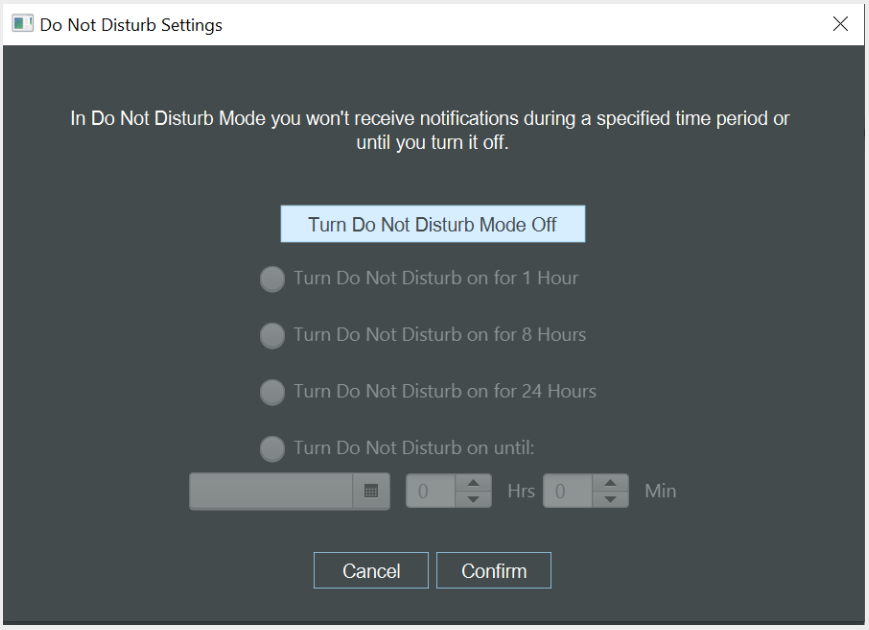
Hira Popov
• Hira is an architect who would like to turn on her email notifications when she knows that other people will be looking at her screen • Hira logs into the email client application using her Outlook account • Hira clicks on the 'Settings' tab and clicks on the 'Change Mode' button • The 'Change Mode' dialog is shown on the screen and after reading the descriptions of the different modes, Hira navigates to the 'Do Not Disturb Mode Settings' button • The 'Do Not Disturb Mode Settings' dialog is shown on the screen • Hira toggles the 'Do Not Disturb' button to 'ON' • Hira confirms her choice on the 'Do Not Disturb Mode Settings' dialog • Hira confirms her choice on the 'Change Mode' dialog • Hira will not receive email notifications until she turns 'Do Not Disturb Mode' off
Corresponding User Story
06
Tests Performed
Type of Test: Manual **Preconditions:** Email Client is not yet loaded and user is not yet logged in • All buttons work as expected • Choices for settings were saved • Email notifications weren't received while 'Do Not Disturb Mode' was turned on
Screenshots


Table 10 - Scenario Test 3

Tasia Abhiskek
• Tasia is a marine biologist who would like to only be notified of emails that contain specified keywords • Tasia logs into the email client application using her Outlook account • Tasia clicks on the 'Settings' tab and clicks on the 'Change Mode' button • The 'Change Mode' dialog is shown on the screen and after reading the descriptions of the different modes, Tasia navigates to the 'Holiday Mode Settings' button • The 'Holiday Mode Settings' dialog is shown on the screen • Tasia selects both the radio button: 'Only allow email notifications where they contain these keywords' enters the keywords of which she would like to be notified • Tasia confirms her choice on the 'Holiday Mode Settings' dialog • Tasia confirms her choice on the 'Change Mode' dialog • She will now only receive notifications of emails that contain her specified keywords
Corresponding User Story
07
Tests Performed
Type of Test: Manual **Preconditions:** Email Client is not yet loaded and user is not yet logged in • All buttons work as expected • Choices for settings were saved • Email notifications were only received if emails contained the keywords in the subject or body
Screenshots

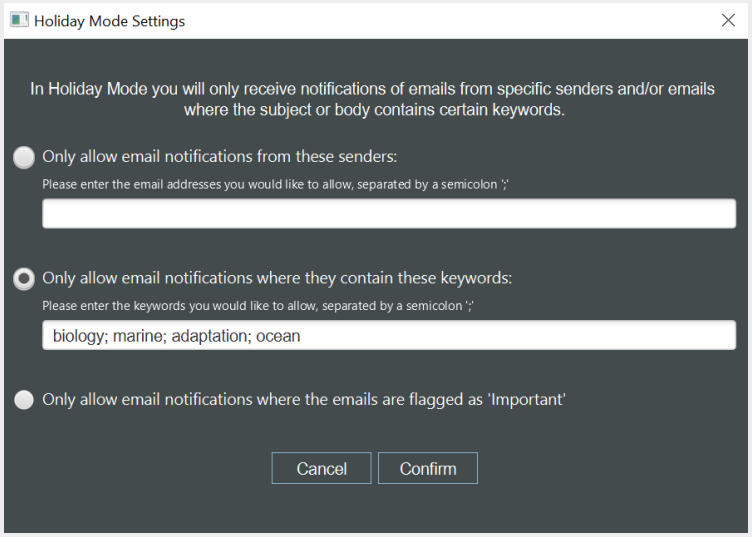


Table 11 - Scenario Test 4

Dan McCrory
• Dan is a self-employed fashion designer who would like to only receive notifications when he has a specified number of new emails in his inbox • Dan logs into the email client application using his Outlook account • Dan clicks on the 'Settings' tab and clicks on the 'Change Mode' button • The 'Change Mode' dialog is shown on the screen and after reading the descriptions of the different modes, Dan navigates to the 'Concentrated Mode Settings' button • The 'Concentrated Mode Settings' dialog is shown on the screen • Dan selects the number of new emails he would like to receive before he is notified • Dan confirms his choice on the 'Concentrated Mode Settings' dialog • Dan confirms his choice on the 'Change Mode' dialog • He will now not receive email notifications until he receives the specified number of new emails in his inbox
Corresponding User Story
08
Tests Performed
Type of Test: Manual **Preconditions:** Email Client is not yet loaded and user is not yet logged in • All buttons work as expected • Choices for settings were saved • Email notifications were only received after the specified date and time

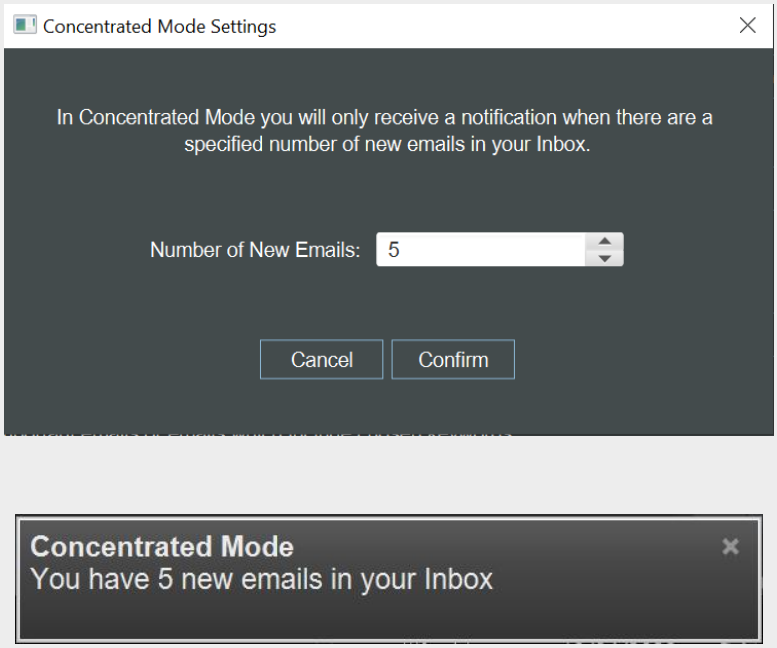
Screenshots


Table 12 - Scenario Test 5

Chapter 7 - Conclusion

The aim of this project was to develop an email client with modes that would allow the user greater control over their notifications. As demonstrated by the testing, this goal was achieved and therefore this project can be regarded as a success.

7.1 Evaluation of Design Requirements

Before building the project design requirements were created and prioritised using the MoSCoW prioritisation system. The table below shows the actual size of the requirements and which of them were implemented into the final design of the application.

Design Requirement	Size Estimation	Actual Size	MoSCoW Prioritisation	Implemented
View a user's emails within a folder	XL	XL	Must Have	Yes
View and download attachments from an email	L	XL	Must Have	Yes
Create new email	M	M	Must Have	Yes
Add attachments to user's emails	L	XL	Must Have	Yes
Send user's email	S	S	Must Have	Yes
Save email as a draft	M	S	Must Have	Yes
Delete emails from inbox/folders	M	S	Must Have	Yes
'Reply', 'Reply All' and 'Forward' Features	L	XL	Must Have	Yes
Load user's existing folders	M	S	Must Have	Yes
A 'Do Not Disturb' mode to stop all email notifications	L	M	Must Have	Yes
A 'Do Not Disturb' mode to stop all email notifications until a specified time	L	L	Must Have	Yes
A 'Do Not Disturb' mode to stop all email notifications for a specified amount of time	L	L	Must Have	Yes
A 'Restricted' mode to only allow email notifications from a list of specified senders	L	XL	Must Have	Yes
A 'Restricted' mode to only allow email notifications from emails that are tagged as important	L	M	Must Have	Yes
A 'Restricted' mode to only allow email notifications from emails where the subject or body includes some keyword(s)	L	XL	Must Have	Yes
A 'Batch Notifications' mode to only notify the user when there is a specified number of new emails in their inbox	L	M	Must Have	Yes
Create new folders	M	S	Should Have	Yes
Delete folders	S	L	Should Have	Yes
File emails into folders	XL	M	Should Have	Yes
A system to remind users to reply to emails	L	-	Could Have	No
Searching Feature	XL	-	Could Have	No

Email tagging features (e.g. 'Important' or giving emails 'Categories')	L	-	Could Have	Yes
Option to view all attachments in one email thread	M	-	Won't Have	No
Creation of rules to organise the inbox	L	-	Won't Have	No
An alternative view of email threads	L	-	Won't Have	No
Integrated To-Do List	L	-	Won't Have	No
Automatic suggestions to add tasks found in to To-Do List	M	-	Won't Have	No
Automatic prioritisation of inbox	XL	-	Won't Have	No
Automatic organisation of inbox e.g. into folders	L	-	Won't Have	No

Table 13 - Evaluation of Design Requirements

7.2 Proof of Concept

The background research for this project demonstrated that there was a desire for email clients to include modes so that they could adapt to the users' needs at different times and during different situations. Therefore an email client application has been developed that includes four modes: Normal, Do Not Disturb Mode, Holiday (Restricted) Mode and Concentrated (Batch) Mode.

7.3 Future Development

The research carried out and literature on email client development has shown that there are many areas for development. To further develop this project, requirements that were placed in the "Won't Have" category of the MoSCoW prioritisation could be introduced and developed further. If this software was to be developed for release, then many production-level features such as sorting of folders' content, a searching feature, customisation of the user interface, and persistence of login and chosen settings could be included.

An area in which more development is needed is the presentation of email threads [3]. This issue was found in the background research literature review, with some users suggesting a tree like structure and others suggesting an instant message style interface like Facebook Messenger or Slack [3]. An instant messaging style interface would probably appeal to a younger audience and ensure that email stays modern. This style of interface could better help users to keep track of when an attachment is sent back and forth and could act as a version control if changes needed to be reverted.

7.4 User Feedback

The finished application was presented to 5 end users. As per the ERGO reference ERGO/FEPS/41626.A2, these end users were recruited through social networks or face to face and they are all students of the University of Southampton. They were shown the 'participants information sheet' and 'consent form' included in the appendices. All the data they provided was anonymous. To assess the usability of the application I based a questionnaire on the System Usability Scale (SUS) [24][25]. The users were also asked to evaluate how useful each of the modes would be to them.

To assess the usability of the application the users were asked to score the following 10 statements with a response that range from 'strongly agree' to 'strongly disagree':

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in the this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

To assess how useful the added modes are, the user were asked to score the following statements with a response that range from 'extremely useful' to 'not at all useful':

1. I would find the 'Do Not Disturb Mode' feature useful.
2. I would find the 'Concentrated Mode' feature useful.
3. I would find the 'Holiday Mode' feature useful.

To work out the SUS for each user the following steps were taken [25][26]:

- The answers are ranked from 1-5, where 'strongly disagree' is 1 and 'strongly agree' is 5.
- Add up the total score for all odd-numbered questions, then subtract 5 from the total to get the new score X
- Add up the total score for all-even numbered, then subtract that total from 25 to get the new score Y
- Add X and Y and multiply the answer by 2.5 to get the SUS score

This will give a SUS score out of 100. The average score is 68 and therefore if it falls below this the usability is below average [26]. In order to combine the scores of the users that evaluated the application, I averaged their scores. The average SUS score from the users was 95. This is above average and therefore the application is very usable.

To evaluate how useful the users found each individual mode, the answers were ranked 1-5, where ‘not at all useful’ is 1 and ‘extremely useful’ is 5. The score from all users was then summed and averaged into a percentage:

- ‘Do Not Disturb Mode’ scored 88%
- ‘Concentrated Mode’ scored 76%
- ‘Holiday Mode’ scored 88%

Therefore users found both ‘Do Not Disturb Mode’ and ‘Holiday Mode’ the most useful within the application. Since all three modes scored highly, it is clear that users found them useful and would therefore benefit from integration of modes into popular and main stream email clients.

Chapter 8 - Project Management

Good project management is crucial for developing and delivering a research project according to the time constraints and therefore a risk assessment was carried out and time management planned with a Gantt Chart.

8.1 Risk Assessment

A risk assessment was carried out before project development started. I predicted any issues that I may face during the development of my research project and the actions I could take to combat the issues. The risk is calculated by multiplying the loss and probability.

Problem	Loss (1-5)	Probability (1-5)	Risk	Plan
Laptop may be lost or damaged	4	2	8	Code will be version managed on a git repository. There are machines in the labs that I could use to develop my project should be laptop be unusable.
COVID-19 causes in-person supervisor communication to halt	1	3	3	My supervisor and I communicate a lot via email and video calls therefore this would not impact our communication greatly.
I fall ill with COVID-19 or another illness	4	3	12	This may cause me to halt my work for a few days whilst I recover. If this happens and severely impacts my time spent on the project then I will only complete the 'Must Have' design requirements from my MoSCoW prioritisation.
Very few people answer my survey	3	3	9	I will ensure that at least some people answer the survey so I have at least some data to work with. If I get very few responses I will rely more on my literature review and alternative products review to inform my design choices.
Underestimation of Task Size	3	4	12	If tasks take longer than I anticipated then I will only complete the 'Must Have' design requirements from my MoSCoW prioritisation.
Poor time management leading to slow progress	4	2	8	I will keep a Gantt Chart in order to keep track of my time management however if I am running short on time I will only complete the 'Must Have' design requirements from my MoSCoW prioritisation.

Table 14 - Risk Assessment

8.2 Time Management

To effectively manage my time I created a Gantt Chart at the beginning of the project as a guide of how to allocate my time and made every effort to adhere to it, however slight deviations were inevitable and are demonstrated on the charts below.

8.3 Gantt Chart

I used a Gantt Chart to plan the time spent on each element of my project. Below is the planned Gantt Chart and the Actual Gantt Chart which depicts how my time was actually allocated. As shown by the actual Gantt Chart, I overestimated how much time I would need to design my project and underestimated how much time I would need for the implementation and testing of the application.

8.3.1 Planned Gantt Chart

Figure 30 - Planned Gantt Chart

8.3.2 Actual Gantt Chart

Figure 31 - Actual Gantt Chart

8.4 Issues Faced During the Project

Although my overall time management for this project was good and I am proud of the result, there were some inevitable difficulties along the way. In February of 2022, I caught COVID and was ill for a week, however I had already accommodated for this in my risk assessment and since I had started development for this project in October I was able to catch up quickly with minimal impact to my progress.

8.4.1 Technical Issues Faced

When starting the project the main issues faced revolved around learning how to properly integrate Microsoft's Graph Outlook Mail API into my application. I chose to write my application in Java since this was the language I had the most experience with and therefore would enable me to start developing the project without delay. Java also had the advantage of JavaFX and furthermore FXML in the form of Scene Builder which allowed me to develop a UI quickly and easily so that more of my attention could be placed on the back end of the application. Since Microsoft own the programming language C#, a lot of the support and tutorials on the Microsoft Graph documentation was for C#, however since Java is quite a similar language and some examples for Java were provided, I was able to eventually overcome my difficulties and integrate the API into my code.

Another issue faced was an account problem with Microsoft's Azure. An Azure account was required so that the application could be registered and a unique application ID could be generated. The initial set up of the application registration happened without issue, the issue only started on 24/03/2022. When logging into the Microsoft Azure Active Directory, two errors were shown:

Figure 32 – Azure AD error 1

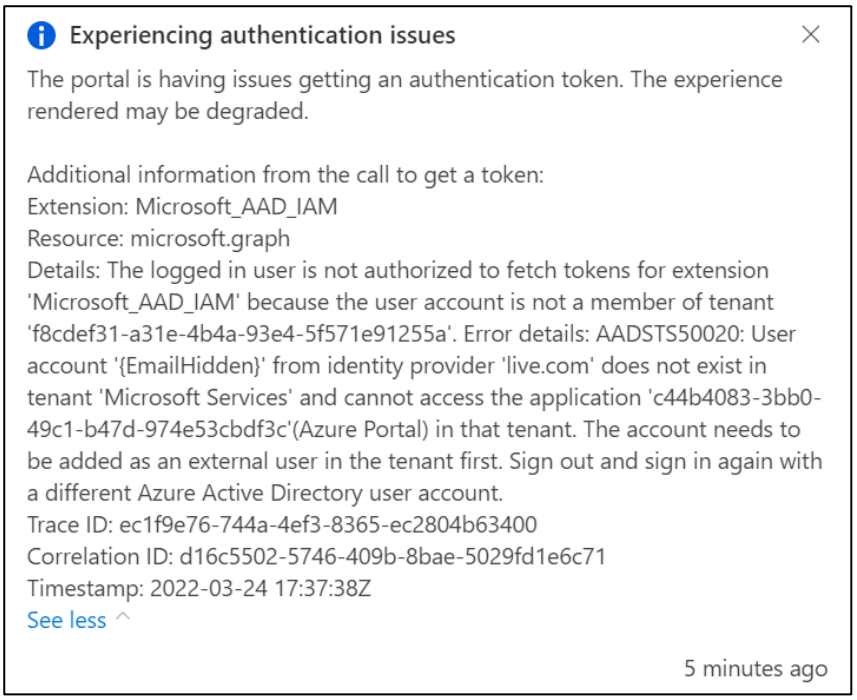


Figure 33 – Azure AD error 2

It appears that the account used to login was under the 'live.com' domain and this was not accepted by the Azure Active Directory (AD). This was confusing since the same account had been used to access Azure AD throughout the project. At first I thought it was a browser issue, so I tried clearing all my cache and cookies, logging in using a private browser window and trying a completely different browser, however none of these fixed the issue. After researching the error code, I found that I had to create an Azure account using the same email address I had been using before. After creating a new Azure account, I was able to access the app registration under 'Applications from Personal Account'. I am still unsure as to what caused this issue, however after creating an Azure AD account I was able to carry on developing the application as normal.

Chapter 9 - References

[1] "What is Focused Inbox?," *Microsoft*. https://support.microsoft.com/en-us/office/what-is-focused-inbox-16b24373-dfa9-4139-ab19-08aa753a6055#:~:text=Focused%20Inbox%20intelligently%20pre%2Dsorts,email%20accounts%2C%20personal%20and%20professional. (accessed Apr. 01, 2022).

[2] A. M. Szóstek, "'Dealing with My Emails': Latent user needs in email management," *Computers in Human Behavior*, vol. 27, no. 2, pp. 723–729, Mar. 2011, doi: 10.1016/j.chb.2010.09.019.

[3] S. Park, A. X. Zhang, L. S. Murray, and D. R. Karger, "Opportunities for automating email processing: A need-finding study," May 2019. doi: 10.1145/3290605.3300604.

[4] C. Blank *et al.*, "Emotional Footprints of Email Interruptions," Apr. 2020. doi: 10.1145/3313831.3376282.

[5] B. v. Hanrahan and M. A. Pérez-Quĩnones, "Lost in email: Pulling users down a path of interaction," in *Conference on Human Factors in Computing Systems - Proceedings*, Apr. 2015, vol. 2015-April, pp. 3981–3984. doi: 10.1145/2702123.2702351.

[6] INC. THE RADICATI GROUP, "Email Statistics Report, 2021-2025," 2021. [Online]. Available: http://www.radicati.com

[7] B. Sarrafzadeh, A. H. Awadallah, C. H. Lin, C. J. Lee, M. Shokouhi, and S. T. Dumais, "Characterizing and predicting email deferral behavior," in *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, Jan. 2019, pp. 627–635. doi: 10.1145/3289600.3291028.

[8] B. Sarrafzadeh, A. H. Awadallah, and M. Shokouhi, "Exploring email triage: Challenges and opportunities," in *CHIIR 2019 - Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, Mar. 2019, pp. 325–329. doi: 10.1145/3295750.3298960.

[9] A. H. Awadallah, S. Dumais, and T. Alrashed, "The lifetime of email messages: A large-scale analysis of email revisitation," in *CHIIR 2018 - Proceedings of the 2018 Conference on Human Information Interaction and Retrieval*, Feb. 2018, vol. 2018-March, pp. 120–129. doi: 10.1145/3176349.3176398.

[10] D. Carmel, G. Halawi, L. Lewin-Eytan, Y. Maarek, and A. Raviv, "Rank by time or by relevance? Revisiting email search," in *International Conference on Information and Knowledge Management, Proceedings*, Oct. 2015, vol. 19-23-Oct-2015, pp. 283–292. doi: 10.1145/2806416.2806471.

[11] S. Whittaker, T. Matthews, J. Cerruti, H. Badenes, and J. Tang, *Am I wasting my time organizing email? A study of email refinding*. 2011.

[12] Y. Koren, E. Liberty, Y. Maarek, and R. Sandler, "Automatically Tagging Email by Leveraging Other Users' Folders," ACM, 2011.

[13] J. Wainer, L. Dabbish, and R. Kraut, “Should I Open this Email?: Inbox-Level Cues, Curiosity and Attention to Email,” 2011.

[14] Y. Kim, A. Fourney, and E. Kamar, “Studying preferences and concerns about information disclosure in email notifications,” in *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, May 2019, pp. 874–885. doi: 10.1145/3308558.3313451.

[15] H. Bota, P. N. Bennett, A. H. Awadallah, and S. T. Dumais, “Self-Es: The Role of Emails-to-Self in Personal Information Management,” 2017.

[16] “Microsoft identity platform and the OAuth 2.0 device authorization grant flow,” *Microsoft Azure Documentation*. https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-device-code (accessed Mar. 05, 2022).

[17] ConceptsWork, “Oauth 2.0 Authorization Code Flow | Microsoft Graph.” https://www.youtube.com/watch?v=AvNc1p2HywI&ab_channel=ConceptsWork (accessed Mar. 03, 2022).

[18] “How to Open a File in Java,” *Java T Point*. https://www.javatpoint.com/how-to-open-a-file-in-java (accessed Apr. 13, 2022).

[19] jasonjoh and nokafor, “Build Java apps with Microsoft Graph,” *Microsoft Graph Documentation*. https://docs.microsoft.com/en-us/graph/tutorials/java (accessed Mar. 12, 2022).

[20] A. Frieze, “How to Use Regular Expressions to Replace Tokens in Strings in Java,” *Baeldung*, Dec. 08, 2021. https://www.baeldung.com/java-regex-token-replacement (accessed Mar. 12, 2022).

[21] Baeldung, “Preserving Line Breaks When Using Jsoup,” *Baeldung*, Jul. 31, 2020. https://baeldung-cn.com/jsoup-line-breaks (accessed Mar. 23, 2022).

[22] “JavaFX CSS Reference Guide,” *Oracle*. https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html (accessed Apr. 02, 2022).

[23] S. Pittet, “An introduction to code coverage,” *Atlassian*. https://www.atlassian.com/continuous-delivery/software-testing/code-coverage (accessed Apr. 14, 2022).

[24] “System Usability Scale (SUS),” *usability.gov*. https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html (accessed Apr. 20, 2022).

[25] J. Brooke, “SUS: A quick and dirty usability scale Display design for fault diagnosis View project.” [Online]. Available: https://www.researchgate.net/publication/228593520

[26] A. Smyk, “The System Usability Scale & How It’s Used in UX,” *Adobe*, Mar. 17, 2020. https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/ (accessed Apr. 20, 2022).

Chapter 10 - Appendices

10.1 Original Project Brief

COMP3200 Individual Project Brief

Student Name: Charlotte Graham

Project Title: A Better Email Client

Supervisor Name: Les Carr

Problem:

Email has been around since the 1970s and although many improvements have been made, email for professional use has fallen behind many other communication methods such as team communication software like Teams and Slack. Some issues encountered are:

- Threads are presented in a way that makes communication hard to follow and keeping track of communication within a group of people is even harder.
- Algorithms designed to organise and filter inboxes often only serve to frustrate users.
- Files sent as attachments are often lost within threads.
- Urgent emails are read and forgotten about.

Email has so much potential, yet so many users experience frustration with it. Issues like these only serve to complicate communication and can hinder user productivity, especially in an academic or working environment.

Goals:

- To create an application that uses Microsoft Outlook API in order to help users better organise their inboxes, make it easier to follow email threads and include better support for organising emails within a group
- Extended goals include:
    - Implement a feature to keep track of files exchanged in an email thread
    - Implement a feature that allows users to choose whether to be reminded of an outstanding email at a late date or time
    - Have sub-inboxes to which emails from specific people are automatically forwarded – this will be down to the choice of the user to enable this or not
    - Offer the user statistics on the people they communicate with the most, how many emails they receive a day, how much of their time is spent reading and sending emails etc

Scope:

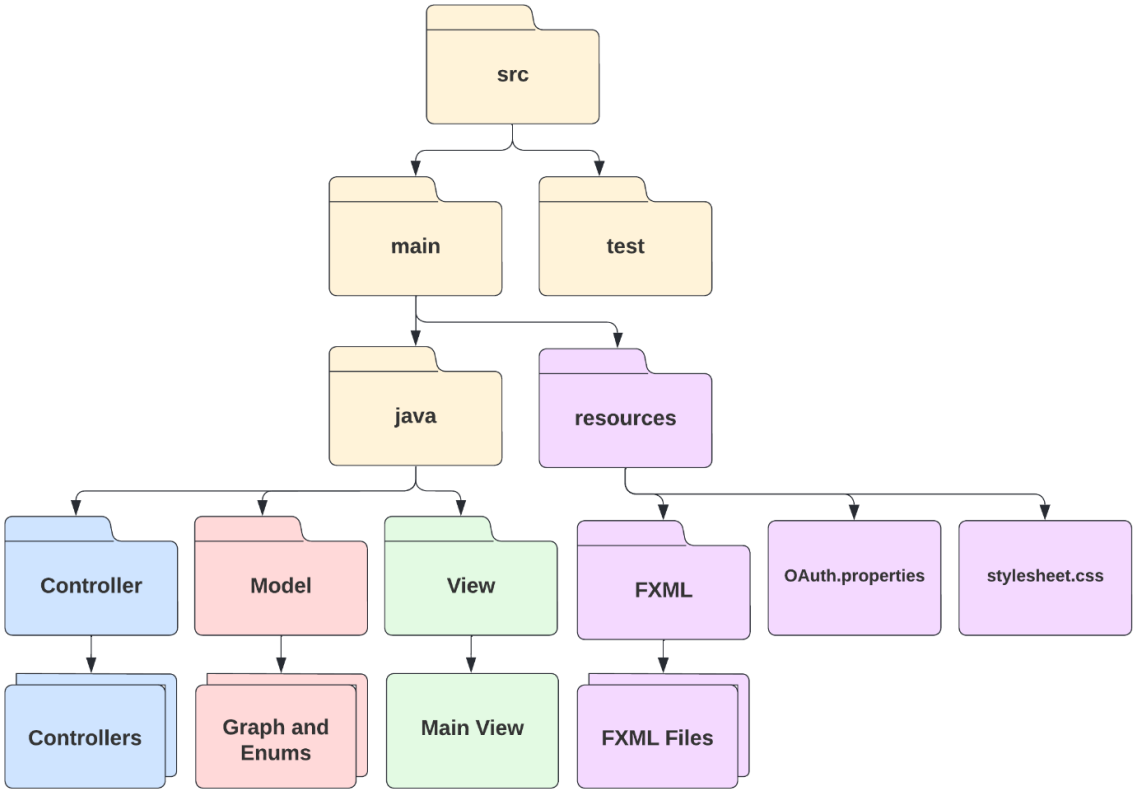
- The application will be written in Java as this is the language with which I am the most familiar
- Outlook's API will used to integrate the inboxes into the application
- Research will be done through methods such as surveys and reviews of similar products to see what features users enjoy and which they feel are missing

10.2 Table of Figures

10.3 Table of Tables

10.4 Contents of Design Archive

The design archive consists of the Maven project developed in Intellij. The code written for this project is within the 'src' folder and organised according to this diagram:



The colours shown in the diagram related to the MVC and UML class diagrams shown earlier, where the blue shows the location of the 'Controller' files, the red shows the 'Model' files, the green shows the 'View' files and the purple shows the resources.

- The files within the 'Controller' folder are responsible for handling interactions between the 'View' and 'Model' of the system.
- The Java file under the 'Model' folder named 'Graph.java' handles all interactions with the Microsoft Graph API which is used to interact with the Microsoft Outlook Mail API.
- The 'Main View' class is used to launch the application interface.
- The FXML files were generated using Scene Builder and are used to launch the different screens of the application.
- The 'OAuth.properties' file contains the application ID that was generated by Microsoft's Azure Active Directory when the application was registered and the scopes used by the application.
- The 'stylesheet.css' file contains the CSS that styles the whole application and is applied to all screens of the application.

10.4.1 How to run the project JAR

The JAR for the project is located under the 'target' directory and named 'EmailClientV3-1.0-SNAPSHOT-shaded.jar'. In order to run the JAR, open the CLI and enter the command:

```
'java -jar EmailClientV3-1.0-SNAPSHOT-shaded.jar'
```

The email client application will then show this line:

```
'To sign in, use a web browser to open the page
https://microsoft.com/devicelogin and enter the code XXXXXXXXX to
authenticate.'
```

After entering the device code into the link provided, the page redirects to a login page, where a user can log into their Microsoft Outlook Account and grant the application the necessary permissions.

When authentication has finished, the application will be launched.

10.5 Participants Information Sheet

Appendix (i) Participant Information

Participant Information

Ethics reference number: **ERGO/**FEPS**/41626.A2**	Version: 1	Date:25/03/2021
Study Title: Student project for COMP3200 – A Better Email Client		
Investigator: **Charlotte Graham**		

Please read this information carefully before deciding to take part in this research. If you are happy to participate you will be asked to indicate your consent to take part either verbally or by selection if an online questionnaire. Your participation is completely voluntary.

What is the research about? This research project is part of the COMP3200 project. The research will only involve :

Questionnaire

Why have I been chosen? You have been approached because you are known to the student(s) or because you have been identified by the students as being appropriate for the research

What will happen to me if I take part? You will take part in a short

Questionnaire

Are there any benefits in my taking part? The study will add to current knowledge, as well as being valuable practical learning for the COMP3200 project student

Are there any risks involved? No sensitive issues will be discussed and there are no risks beyond that which would normally be experienced in everyday life.

Will my data be confidential? Please do not give any identifiable personal information. The project student will retain anonymous data until the end of the Project. No video or audio recording will occur **What happens if I change my mind?** You may withdraw at any time and for any reason. You may decline to give your consent and not take part in the study without penalty.

What happens if something goes wrong? If you have any concern or complaint, contact Charlotte Graham – cb6g19@soton.ac.uk or Leslie Carr - lac@ecs.soton.ac.uk, otherwise please contact Research Governance Office (02380 595686, Rgoinfo@soton.ac.uk).

10.6 Consent Form

Appendix (ii) Consent Form

Consent Form

Ethics reference number: **ERGO/**FEPS**/41626.A2**	Version: 1	Date: 25/03/2021
Study Title: Student project for COMP3200- A Better Email Client		
Investigator: **Charlotte Graham**		

Please read the following and indicate

by selection on an online questionnaire

if you agree with the following statements:

I have read and understood the Participant Information and have had the opportunity to ask questions about the study.

I agree to take part in this study.

I understand my participation is voluntary and I may withdraw at any time and for any reason.

ONLY If the participant has verbally or by selection on an online questionnaire agreed to the above, and consented to take part in the research, the study may commence.