香 港 大 學
THE UNIVERSITY OF HONG KONG

COMP4801 Final Year Project

Final Report – Front-End

Inbox Genius – Your Next Productive Email Client

Cheung Yan Ting (3035786988)

Date of Submission: 26 April 2024

Acknowledgement

We would like to express our gratitude our supervisor – Dirk Schnieders for guiding us throughout this process, as well as our second examiner – Bruno Oliveira. The constructive advice and help are truly appreciated.

Table of Contents

List of Abbreviations

Abbreviation	Full form
API	Application Programming Interface
CI/CD	Continuous Integration and Deployment
CSS	Cascading Style Sheets
E-2-E	End-to-End
ICS	Internet Calendaring and Scheduling
ML	Machine Learning
NLP	Natural Language Processing
SSO	Single Sign-On
UI	User Interface

List of Figures

1. Introduction

This is to develop the understanding of front-end development and accomplishments throughout the one-year construction. To expand in detail, the section is divided in three components:

- Development and Integration: This implementation encompasses three major aspects: 1) integration with a third-party service, 2) visual component rendering, and 3) back-end communication. The first one involves the integration of a login service for Microsoft Outlook connection. The second aspect focuses on the rendering of the User Interface (UI) components, while the final part pertains to the utilization of an Application Programming Interface (API).

- Highlighted Implementations: This part will dive into the various implemented features of the core web application, with a comprehensive analysis and discussion of each feature. Each functionality will be examined, highlighting its purpose and value that brings to the overall user experience. Additionally, the visual aspects associated with these functionalities will be explored along with their design on visual presences and UI considerations. It aims at a comprehensive understanding of the core web application and its capabilities.

- Further Improvement: This section will comprehensively cover the sectors in which enhancements have been made to further improve the current results within the domain of front-end development. It will delve into the specific areas where advancements have been implemented, detailing the enhancements and their impact on the overall functionality and user experience of the web application.

2. Development and Integration

2.1. Technological Solution Stack

2.1.1. Wireframing and UI Design: Figma

The initial draft on the email client UI was conducted on Figma, a wireframing tool. As one of the most popular and advanced interface design gadgets[1], it has a well-established community of designers with diverse online resources for the team to reference from. One highlight is its sample code plug-in which generates and provides codes to implement the components in the product with styling. Coupled with guidance from online UI community, it boosts the efficiency of development.

2.1.2. Engineering: React.js

As the product primarily serves as a web application, React.js has been selected for the UI build due to the team's prior knowledge and experience. It is a front-end framework built upon JavaScript and is used to develop server-rendered single-paged applications, which fulfills the project's goals. More essentially, due to being widely used[2], there is a vast supporting community with a wide range of UI component libraries which few have been incorporated into the email client and would be discussed in Section 2.2.

2.2. Development

2.2.1. Application

The product we developed utilizes ReactJS, which the previous section has discussed. It leverages various features of React.js, which includes:

- State variables

  React components provide developers with an efficient and organized way to handle and store dynamic data that undergoes frequent changes. In the specific application being discussed, OAuth tokens are employed to authenticate and authorize access, and live data is retrieved from external sources, as detailed in Section 2.2. Given the sensitive nature of this data, it

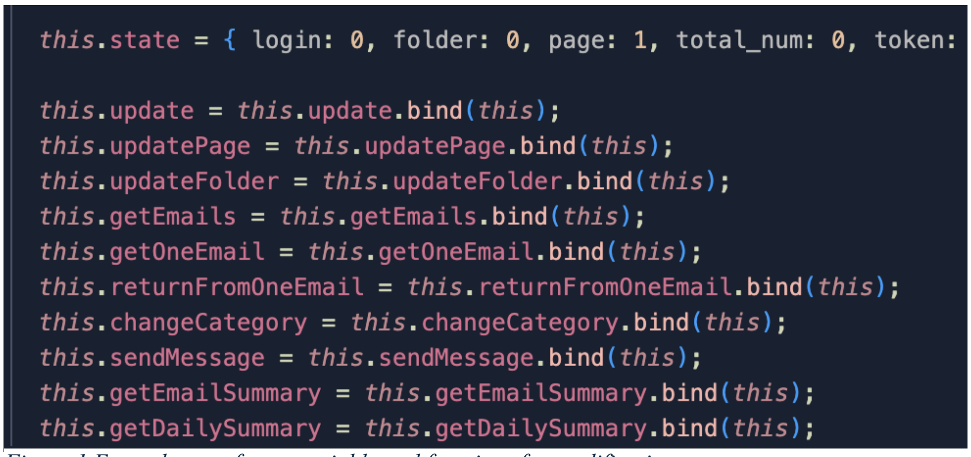
[1] Kopf B. The Power of Figma as a Design Tool. Toptal Design Blog. Published 2018. https://www.toptal.com/designers/ui/figma-design-tool

[2] Stack Overflow. Stack Overflow Trends. insights.stackoverflow.com. Accessed April 26, 2024. https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Csvelte%2Cangularjs%2Cvuejs3

is crucial to establish robust measures to protect temporary information from unauthorized access.

One of the key advantages of React components is their ability to maintain isolated state variables. Each component maintains its own dedicated stack of state variables, ensuring that they are securely stored and inaccessible to other threads or components[3]. This isolation mechanism prevents unauthorized access or modification of these variables, contributing to a higher level of data security and integrity.

Furthermore, to enhance the protection of user information, the implementation incorporates the use of highly sensitive access keys. These access keys are carefully managed and applied in the application, ensuring that only authorized entities can access and interact with the data[4]. By implementing this additional layer of security, the system effectively safeguards user information from any malicious or unauthorized intentions, providing users with peace of mind regarding the confidentiality and integrity of their data.



```
this.state = { login: 0, folder: 0, page: 1, total_num: 0, token:

this.update = this.update.bind(this);
this.updatePage = this.updatePage.bind(this);
this.updateFolder = this.updateFolder.bind(this);
this.getEmails = this.getEmails.bind(this);
this.getOneEmail = this.getOneEmail.bind(this);
this.returnFromOneEmail = this.returnFromOneEmail.bind(this);
this.changeCategory = this.changeCategory.bind(this);
this.sendMessage = this.sendMessage.bind(this);
this.getEmailSummary = this.getEmailSummary.bind(this);
this.getDailySummary = this.getDailySummary.bind(this);
```

Figure 1 Example use of state variable and functions for modification.

- React Hook

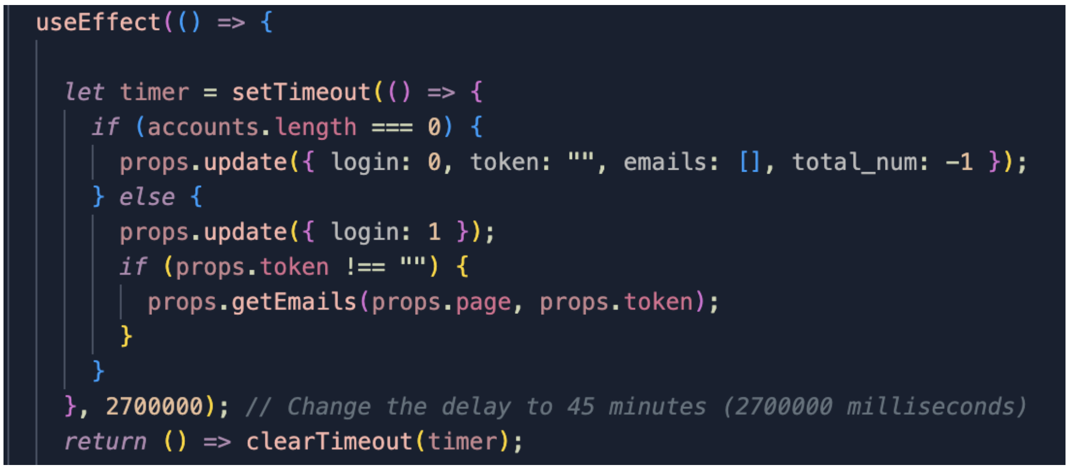
  React 16.8 brought about a breakthrough with the introduction of Hooks, a powerful feature that revolutionized how state and other React features are used within functional components. This innovation effectively eliminates the requirement for traditional class components, offering developers a more streamlined and intuitive approach to building React applications.

[3] MDN web docs. Call stack - MDN Web Docs Glossary: Definitions of Web-related terms | MDN. developer.mozilla.org. Accessed April 24, 2024. https://developer.mozilla.org/en-US/docs/Glossary/Call_stack

[4] MDN web docs. Call stack - MDN Web Docs Glossary: Definitions of Web-related terms | MDN. developer.mozilla.org. Accessed April 24, 2024. https://developer.mozilla.org/en-US/docs/Glossary/Call_stack

One of the standout Hooks is "useEffect," which plays a pivotal role in simplifying the management of side effects based on component lifecycles[5] [2]. In the past, developers had to handle different lifecycle methods like "componentDidMount" and "componentDidUpdate" separately, resulting in repetitive code patterns commonly known as boilerplate effects. These boilerplate effects often led to code redundancy and decreased code maintainability.

With the advent of "useEffect," developers can consolidate and centralize the management of side effects in a single place. This powerful Hook combines the functionalities of lifecycles, addressing the repetitive nature of handling similar code across multiple lifecycle methods[6]. By applying, it significantly reduces code duplication and enhances the overall maintainability and readability of the codebase.



```
useEffect(() => {

  let timer = setTimeout(() => {
    if (accounts.length === 0) {
      props.update({ login: 0, token: "", emails: [], total_num: -1 });
    } else {
      props.update({ login: 1 });
      if (props.token !== "") {
        props.getEmails(props.page, props.token);
      }
    }
  }, 2700000); // Change the delay to 45 minutes (2700000 milliseconds)
  return () => clearTimeout(timer);
```

Figure 2 Example use of React Hook in the data management of log-in process.

The introduction of Hooks marks a significant milestone in the evolution of React.js, empowering developers to write more elegant and concise code. The ability to use state and other React features directly in functional components not only simplifies the development process but also improves the overall efficiency and scalability of React applications.

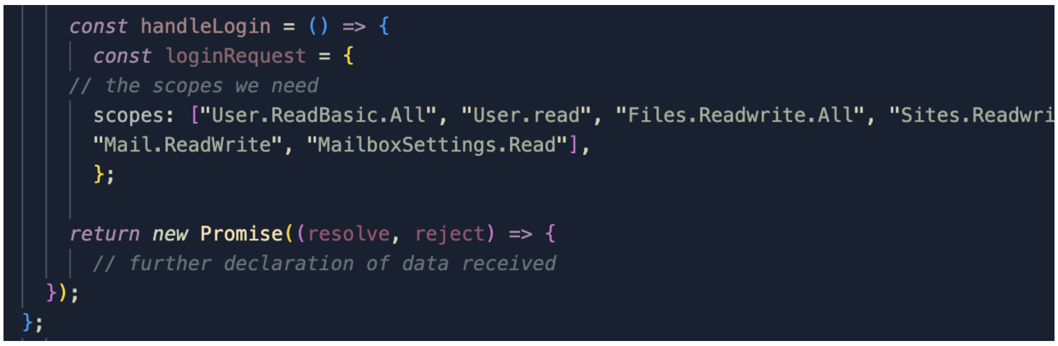
- React Async

React Async is a utility library within the React ecosystem that enables developers to handle asynchronous data fetching in a declarative manner,

[5] React. Synchronizing with Effects – React. react.dev. Accessed April 24, 2024. https://react.dev/learn/synchronizing-with-effects

[6] React. Synchronizing with Effects – React. react.dev. Accessed April 24, 2024. https://react.dev/learn/synchronizing-with-effects

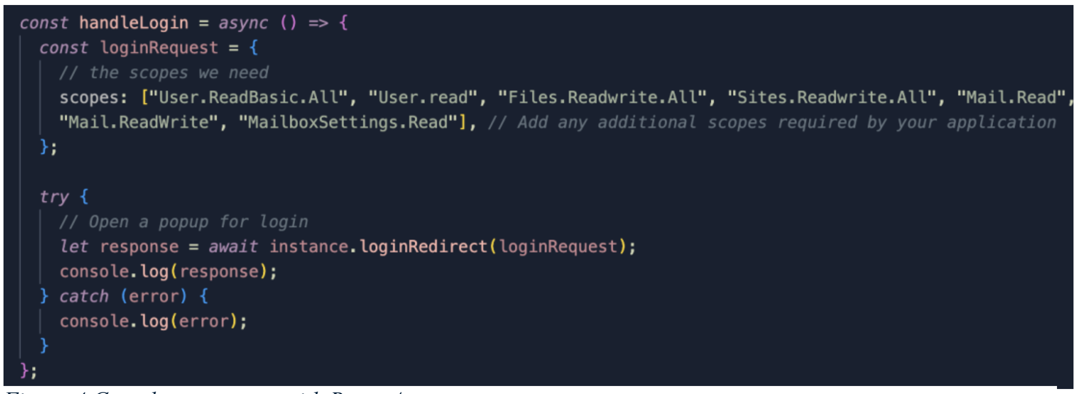
ensuring reliable resolution within React applications[7]. Unlike traditional approaches that often rely on assumptions about data structures and request properties, React Async introduces a set of React components and hooks that effectively manage the state of asynchronous UIs. These components and hooks allow developers to declaratively handle asynchronous operations at various stages, such as pending, resolved, and error states. The comparison has been demonstrated in Figure 3 and 4.



```
    const handleLogin = () => {
      const loginRequest = {
    // the scopes we need
      scopes: ["User.ReadBasic.All", "User.read", "Files.Readwrite.All", "Sites.Readwri
      "Mail.ReadWrite", "MailboxSettings.Read"],
      };

    return new Promise((resolve, reject) => {
      // further declaration of data received
  });
};
```

Figure 3 Example structure without the use of React Async



```
const handleLogin = async () => {
  const loginRequest = {
    // the scopes we need
    scopes: ["User.ReadBasic.All", "User.read", "Files.Readwrite.All", "Sites.Readwrite.All", "Mail.Read",
    "Mail.ReadWrite", "MailboxSettings.Read"], // Add any additional scopes required by your application
  };

  try {
    // Open a popup for login
    let response = await instance.loginRedirect(loginRequest);
    console.log(response);
  } catch (error) {
    console.log(error);
  }
};
```

Figure 4 Complete structure with React Async

One of the key advantages of using React Async is its ability to simplify asynchronous state management, while simultaneously enhancing composability and reusability. By adopting React Async, developers can implement an on-demand data retrieval mechanism at the component level[8]. This approach avoids the common practice of loading data in bulk at the page level, which can lead to excessive waiting times and potential network congestion. Instead, React Async decouples the data fetching process from the page stack and executes it in a more efficient manner.

[7] React Async. Introduction | next | React Async. React-async.com. Published December 8, 2020. Accessed April 24, 2024. https://docs.react-async.com/

[8] React Async. Introduction | next | React Async. React-async.com. Published December 8, 2020. Accessed April 24, 2024. https://docs.react-async.com/

Furthermore, React Async offers improved testing and usage experiences by being agnostic to the underlying routing model. Whether a project adopts a dynamic routing paradigm or follows a route-less application structure, React Async remains independent and adaptable. This flexibility allows developers to seamlessly integrate React Async into their existing projects without major architectural changes.

In summary, React Async is a powerful utility library for React that revolutionizes the handling of asynchronous data fetching. Its declarative approach, combined with enhanced composability and reusability, makes it an invaluable tool for building efficient and robust React applications. By leveraging React Async's on-demand data retrieval mechanism and its routing-agnostic nature, developers can optimize performance and improve the overall user experience of their applications[9].

2.2.2. Styling

- Color selection

To ensure a consistent and visually appealing color scheme throughout the project, a carefully curated range of colors was employed as displayed in Figure 5, accompanied by a set of guidelines dictating how these colors should be applied to various components. These guidelines encompassed factors such as shading techniques and the appropriate color selection for different types of components. By adhering to this systematic approach, the project aimed to create a harmonious and unified visual experience across its entire spectrum. In doing so, it sought to establish a cohesive aesthetic that would resonate with users and reinforce the project's overall design integrity.

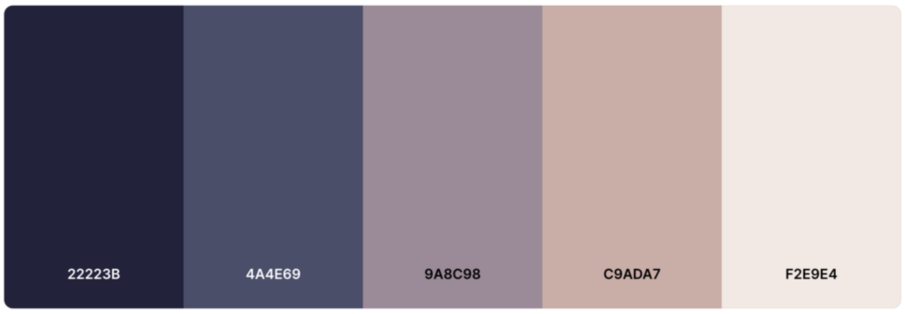


Figure 5 Selection of colours the project uses.

[9] React Async. Introduction | next | React Async. React-async.com. Published December 8, 2020. Accessed April 24, 2024. https://docs.react-async.com/

In addition to the often-cited advantage of enhancing brand recognition, adhering to industry standards in color selection and application serves multiple purposes. Notably, it significantly contributes to improved usability and navigation[10]. By implementing a standardized approach that assigns specific colors to each element, the overall intuitiveness of applications is greatly enhanced. This standardized color scheme allows users to swiftly grasp the functionalities of the product within a shorter timeframe, as they can easily associate similar components with the same color tone.

Moreover, the utilization of industry-standard color guidelines goes beyond superficial aesthetics. It establishes a solid foundation for user experience design[11], ensuring that visual cues and consistency align with established norms and best practices. This consistency engenders a sense of familiarity and ease of use for users, reducing cognitive load and facilitating efficient interaction with the application.

- Separation of Components

  To avoid overlap of defined styling, most components are functional components which are separated from the class component.

  A highlighted benefit is the improved testability. When styles are encapsulated within a component, it allows for focused testing of that specific component without the need to consider or account for the impact of global styles. This isolation provides an efficiently controlled testing environment where the behavior and appearance of the component can be thoroughly examined without interference from external styles. By testing components in isolation, developers can verify that the component's functionality and styling work as intended without being influenced by other parts of the application. This targeted testing approach helps identify issues or bugs specific to the component itself, enabling more precise debugging and troubleshooting.

[10] Jadhav S. The importance of color in web design and how to choose the right palette. Medium. Published March 27, 2023. Accessed April 24, 2024. https://medium.com/@siddhantjadhav445/the-importance-of-color-in-web-design-and-how-to-choose-the-right-palette-72f63dd7f6be

[11] Jadhav S. The importance of color in web design and how to choose the right palette. Medium. Published March 27, 2023. Accessed April 24, 2024. https://medium.com/@siddhantjadhav445/the-importance-of-color-in-web-design-and-how-to-choose-the-right-palette-72f63dd7f6be

Another asset of the isolation is the alignment of a core principle of React's component-based architecture – Separation of Concerns[12]. By separating component styling, the codebase becomes more organized and modular. Each component encapsulates its own styling rules, making it easier to locate and modify specific styles or behaviors. Changes to the styling can be made independently of the component's structure or functionality, reducing the risk of unintended side effects, or breaking existing functionality. This modularity also simplifies maintenance and future enhancements, as modifications to the styling can be isolated and applied without impacting the underlying component's functionality.

- Third-party Assistance: Tailwind Cascading Style Sheets (CSS)

Tailwind is an open-source CSS framework that follows the utility-first principle. It is in a pre-defined class structure that app components can designed directly based on the markups in their classes.

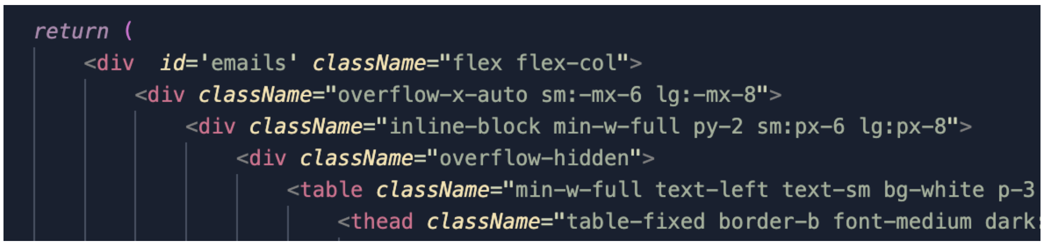
One of the key advantages of Tailwind CSS is its impact on loading efficiency. By automatically removing unused styling, Tailwind CSS enables faster loading times for web applications. This is achieved by reducing the file size that needs to be downloaded and parsed by browsers. With a lighter CSS file, the rendering performance improves as browsers spend less time digesting the styling rules. Additionally, Tailwind CSS provides benefits in terms of caching. Browsers cache CSS files to avoid redundant downloads[13], even if certain parts of the CSS are not utilized. By removing these unused portions, Tailwind CSS ensures that the cached CSS file is as small as possible, resulting in faster subsequent page loads for returning users.

Tailwind CSS provides significant benefits not only for end-users but also for developers. The framework's approach of using pre-defined utility classes promotes adherence to the "Separation of Concerns" principle in React, resulting in a more organized and maintainable codebase.

[12] Qawwas O. Separation of Concerns on the Front-end With React. Medium. Published July 28, 2022. Accessed April 24, 2024. https://engineering.teknasyon.com/separation-of-concerns-on-the-front-end-with-react-fd5d4afcc298

[13] Boroumand A. A Web Developer's Guide to Browser Caching. Medium. Published July 25, 2017. Accessed April 24, 2024. https://medium.com/@steelcityamir/a-web-developers-guide-to-browser-caching-cc41f3b73e7c

By leveraging Tailwind CSS's pre-defined classes, developers are encouraged to separate the concerns of styling from the logic and structure of their components as shown in Figure 6. This separation allows developers to focus on the functionality and purpose of each component, while the styling remains encapsulated within the appropriate utility classes. Developers can easily identify and modify the styles associated with a particular component by referencing the corresponding utility classes. This clear separation of concerns enhances code readability and maintainability, making it easier to understand and modify the application over time.



```
return (
    <div  id='emails' className="flex flex-col">
        <div className="overflow-x-auto sm:-mx-6 lg:-mx-8">
            <div className="inline-block min-w-full py-2 sm:px-6 lg:px-8">
                <div className="overflow-hidden">
                    <table className="min-w-full text-left text-sm bg-white p-3
                        <thead className="table-fixed border-b font-medium dark:
```

Figure 6 Example use of Tailwind CSS with classes precisely defined.

Furthermore, Tailwind CSS's utility-first approach enables developers to rapidly prototype and iterate on the UI. By composing and combining utility classes, developers can quickly assemble custom components and layouts without the need for writing extensive custom CSS. This modularity and reusability of styles promote a more efficient and scalable development process.

Tailwind CSS also provides a comprehensive set of configuration options, allowing developers to customize the default styles to match the specific design requirements of their project. This flexibility ensures that Tailwind CSS can adapt to various design systems and branding guidelines while maintaining its utility-first approach.

2.3. Integration

2.3.1. Microsoft Outlook Login System

In the context of this Microsoft Outlook-oriented project, the authentication process differs from the common approach of handling login information through the backend service. Instead, Outlook utilizes its own mechanism by redirecting users to sign in on the official website — Single Sign-On (SSO), followed by the usage of the Microsoft Authentication Library (MSAL) React[14].

[14] Microsoft. React single-page application using MSAL React to authenticate users against Azure AD for Customers - Code Samples. learn.microsoft.com. Published March 12, 2024. Accessed April 24, 2024. https://learn.microsoft.com/en-us/samples/azure-samples/ms-identity-ciam-javascript-tutorial/ms-identity-ciam-javascript-tutorial-1-sign-in-react/

To integrate this authentication process seamlessly into the product, the entire application is wrapped within a MsalProvider component. This component plays a crucial role in configuring and establishing the connection between the application and MSAL. By providing an instance of the MsalProvider, developers can set up the necessary authentication parameters and ensure a secure and reliable authentication flow.

The MsalProvider component encapsulates the logic and functionality required for authentication, such as handling token acquisition, refreshing tokens, and managing user sessions. It acts as a central hub for managing authentication-related tasks and provides a simplified interface for developers to interact with the MSAL library.

2.3.2. Communication with Back-end

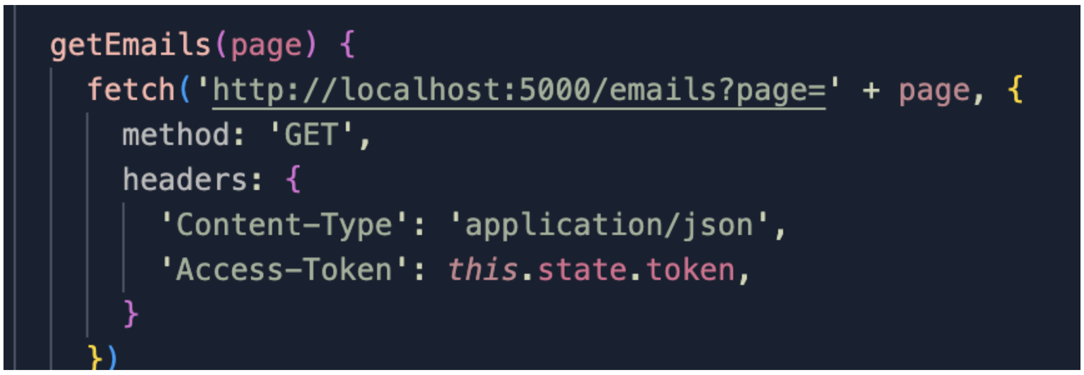
To maintain a continuous connection with the Microsoft Outlook server, a user-specific access token is acquired. This token is obtained as a result of the UI login handling, where the Outlook server returns it in the response to the frontend. Since there is no officially released API for retrieving the token from the backend, a state variable is allocated to store the token upon the user's initial sign-in. Additionally, a timer is implemented using React Hooks to periodically renew the token's value. Please refer to Figure 2 under Section 2.2.1.

Whenever users interact with the UI and trigger events, the token is included in the HTTPS request header. By leveraging the secure nature of HTTPS, the project ensures that the token is encrypted within the header and transmitted to the backend without any unauthorized access to the communication[15]. Figure 7 is a sample exercise of such.

This encryption mechanism provides an added layer of security, safeguarding the token from potential interception or tampering during transit. By including the token in the request header, the backend can validate and authorize the user's actions based on the provided token. This ensures that only authenticated users with valid tokens are granted access to protected resources or perform authorized operations within the application.

The project's approach of utilizing HTTPS for transmitting the token aligns with industry-standard security practices. HTTPS ensures the confidentiality and integrity of data exchanged between the frontend and backend, providing a secure channel for transmitting sensitive information such as access tokens.

[15] The HTTPS-Only Standard. The HTTPS-Only Standard - Introduction to HTTPS. https.cio.gov. Accessed April 24, 2024. https://https.cio.gov/faq/



```
getEmails(page) {
  fetch('http://localhost:5000/emails?page=' + page, {
    method: 'GET',
    headers: {
      'Content-Type': 'application/json',
      'Access-Token': this.state.token,
    }
  })
```

Figure 7 Partial content of getEmail function that retrieve access token from a state variable and includes it in the header information of the request.

3. Highlighted Implementations

3.1. Internet Calendaring and Scheduling (ICS) /iCalendar file Generation

This function is GPT-aided to identify event details in email body, namely topic, start and end date-time, and location of event with a click on button "Add to Schedule". The extracted information is then used to generate an ICS (iCalendar) file, which is a globally standardized format for storing and exchanging calendaring and scheduling data. The ICS format allows users to easily share event information across different calendar applications and platforms, visualized in Figure 8.

With the GPT-aided event identification function, users no longer need to manually search for and input event details when creating calendar entries which cause potential misread of information. Instead, the function automates the process by analyzing the email content and extracting the relevant information. This saves users time and minimizes the risk of errors or omissions when manually entering event details.

Once the event details are identified and extracted, the function generates an ICS file that contains all the necessary information, including the event's topic, start and end date-time, and location. This file can then be downloaded by the user and added to their calendar application with a simple click or import action.

By providing users with an automated and efficient way to generate ICS files from email content, the GPT-aided function simplifies the process of scheduling and managing events. It streamlines the workflow and ensures that important event details are accurately captured and integrated into the user's calendar system.

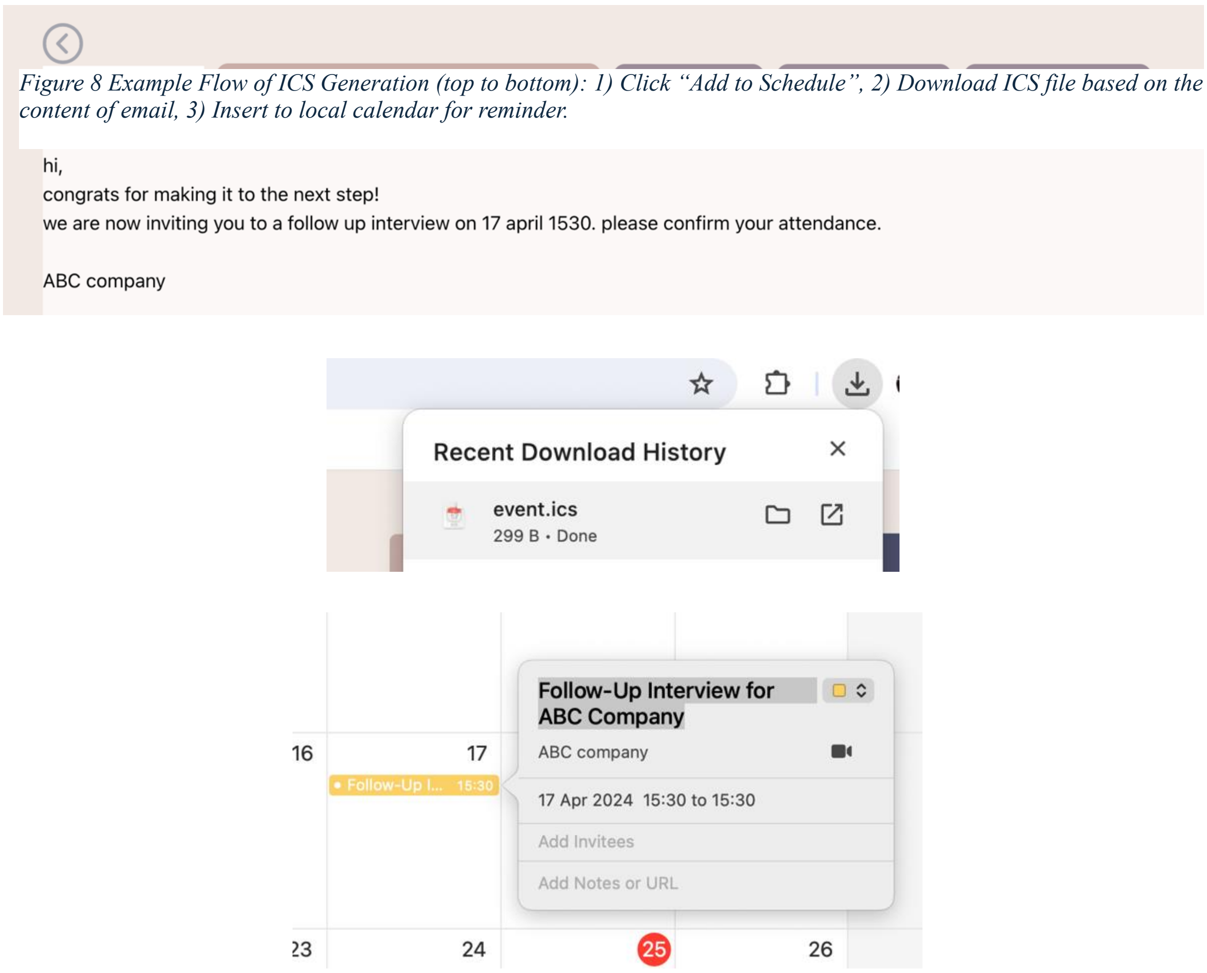


Figure 8 Example Flow of ICS Generation (top to bottom): 1) Click "Add to Schedule", 2) Download ICS file based on the content of email, 3) Insert to local calendar for reminder.

hi,
congrats for making it to the next step!
we are now inviting you to a follow up interview on 17 april 1530. please confirm your attendance.

ABC company

3.2. Email Summary

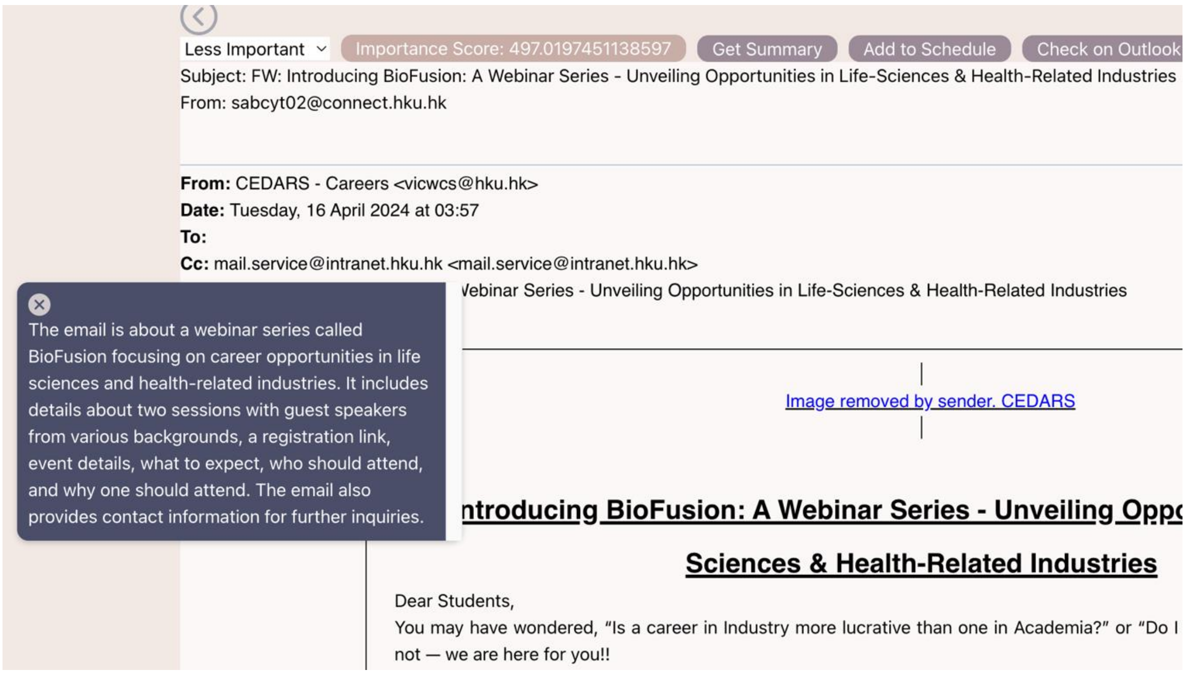
The email summarization feature utilizes GPT technology to provide users with concise summaries of lengthy emails. By breaking down the body content of the email into easily digestible chunks, this tool allows users to quickly scan through the summaries and assess the relevance and importance of the message. This helps to mitigate the risk of misinterpretation that can occur when reading long emails in their entirety.

For example, when a user clicks the "Get Summary" button, a 59-word abstract is generated for a 679-word email. This significant reduction in word count results in a substantial reduction in the time required to read through the content. Based on calculations, the time savings amount to approximately 91.3%.

The summarized email provides users with an overview of the key points and main ideas, enabling them to grasp the essential information without having to read the entire email. This allows for more efficient email management and decision-making.

After reviewing the summary, users can take appropriate actions based on their understanding of the email. For instance, they can choose to reply or delete the email directly from their Outlook Client by clicking the "Check on Outlook" button. This streamlines the workflow and eliminates the need to switch between different applications.

Additionally, if the email contains information about an event or appointment, users have the option to download an ICS file related to the event. By clicking the "Add to Schedule" button, the file can be saved on the user's device and imported into their calendar application. This simplifies the process of scheduling and organizing events mentioned in the email.



Subject: FW: Introducing BioFusion: A Webinar Series - Unveiling Opportunities in Life-Sciences & Health-Related Industries
From: sabcyt02@connect.hku.hk

From: CEDARS - Careers <vicwcs@hku.hk>
Date: Tuesday, 16 April 2024 at 03:57
To:
Cc: mail.service@intranet.hku.hk <mail.service@intranet.hku.hk>
Vebinar Series - Unveiling Opportunities in Life-Sciences & Health-Related Industries

The email is about a webinar series called BioFusion focusing on career opportunities in life sciences and health-related industries. It includes details about two sessions with guest speakers from various backgrounds, a registration link, event details, what to expect, who should attend, and why one should attend. The email also provides contact information for further inquiries.

Image removed by sender. CEDARS

ntroducing BioFusion: A Webinar Series - Unveiling Oppo

Sciences & Health-Related Industries

Dear Students,
You may have wondered, "Is a career in Industry more lucrative than one in Academia?" or "Do I
not — we are here for you!!

Figure 9 Example summary of long email in pop up box in the middle left.

3.3. Email Categorization

The email client incorporates a sophisticated feature that intelligently sorts incoming emails into three distinct importance levels: Most Important, Less Important, and Least Important. This functionality aims to solve situation where 67% – 82% e-mailbox users encounter fatigue and miss important information due to the clustered emails[16]. It assists users in efficiently prioritizing their email workflow and ensuring that critical messages receive the attention they deserve.

The email sorting process is not solely based on predefined rules or algorithms. Instead, the email client considers user activities on individual emails and the relevance of those emails to past correspondence which would be discussed in other reports.

In the UI, the importance levels are visually represented. It allows users to quickly identify and distinguish emails based on their importance. The Most Important category typically includes urgent or time-sensitive emails that require immediate attention. The Less Important category comprises emails that are relevant but may not require an immediate response. Lastly, the Least Important category includes emails that are considered less critical or of lower priority. For better recognition, a viewably darker color is applied to the selected folder as shown in Figure 10. While the email client automatically assigns an initial importance level to incoming

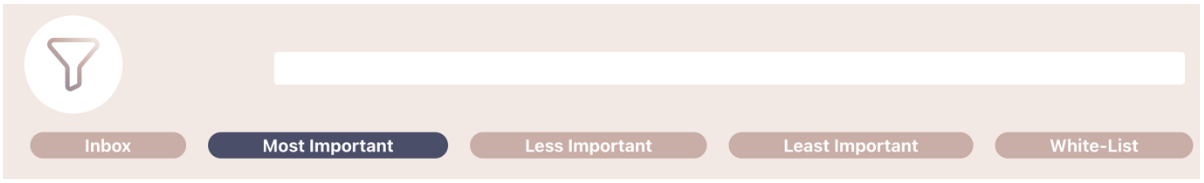
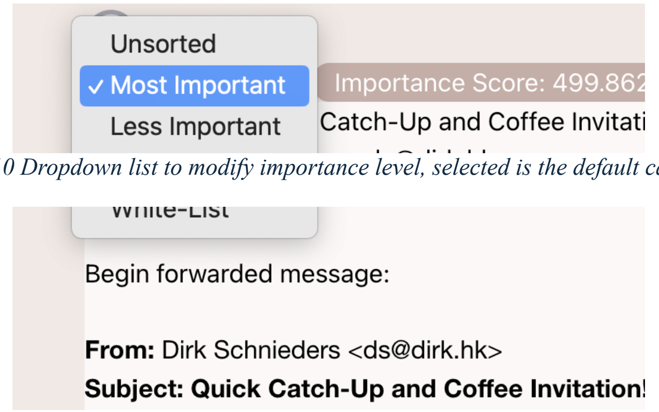


Figure 11 Range of Email Folders based on importance levels and specifications.

emails, users also have the flexibility to manually adjust the categorization based on their preferences which is showcased in Figure 11. This feature allows users to fine-tune the sorting process according to their individual needs and priorities.

By providing a comprehensive email sorting feature that takes into account user



Importance Score: 499.862

Catch-Up and Coffee Invitati

Begin forwarded message:

From: Dirk Schnieders <ds@dirk.hk>

Subject: Quick Catch-Up and Coffee Invitation!

Figure 10 Dropdown list to modify importance level, selected is the default category.

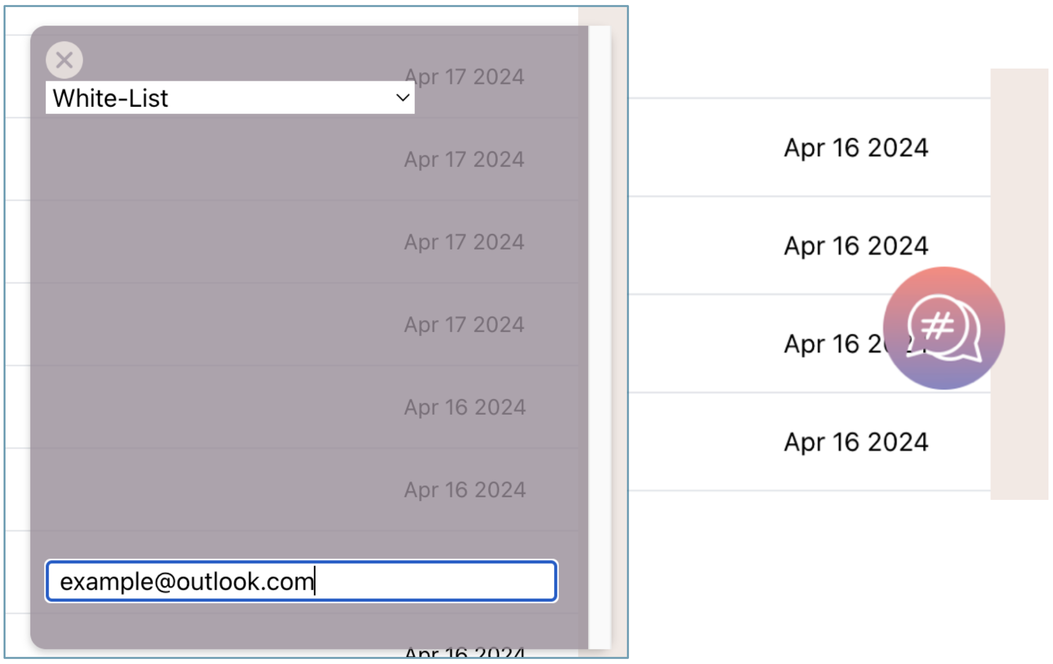
activities and historical context, the email client empowers users to efficiently manage their inbox and focus their attention on the most relevant and important messages. This helps to streamline email processing and ensures that critical

[16] Gated. 67 Percent of People Feel Overwhelmed by Their Email Inbox According to New Inbox Intelligence Report from Email Management Solution Gated. www.prnewswire.com. Published October 26, 2022. Accessed April 24, 2024. https://www.prnewswire.com/news-releases/67-percent-of-people-feel-overwhelmed-by-their-email-inbox-according-to-new-inbox-intelligence-report-from-email-management-solution-gated-301659242.html

communications are promptly addressed while minimizing distractions from less important emails.

3.4. Whitelist

The email client incorporates this that empowers users to selectively include email addresses that they deem important or trustworthy. By adding these email addresses to the whitelist located in the chat box as shown in Figure 12, the email client ensures that incoming emails from these sources are not overlooked or mistakenly categorized as spam.



example@outlook.com

Figure 12 Example use of Whitelist function by sending example@outlook.com to the chat box.

When a new email arrives from a sender listed in the whitelist, the email client automatically directs it to a designated mail folder called "Whitelist." This folder serves as a centralized location for all emails from whitelisted senders, making it easier for users to locate and prioritize these messages.

This feature is particularly beneficial for users who receive a large volume of emails on a daily basis or who may not always have sufficient time to thoroughly review every incoming email. By whitelisting specific email addresses, users can ensure that important content from trusted sources is readily accessible and not lost among the influx of other emails.

The whitelist function provides users with a proactive approach to email management, allowing them to prioritize and give special attention to emails from senders they have identified as relevant or significant. This ensures that critical information is not missed or mistakenly disregarded, even when time constraints prevent users from extensively reading emails from listed senders.

By incorporating the whitelist function, the email client enhances productivity and efficiency by streamlining the email filtering process. It eliminates the need for users to manually sort through numerous emails to identify and respond to important messages, as the whitelist automatically directs them to a dedicated folder for easy access and review.

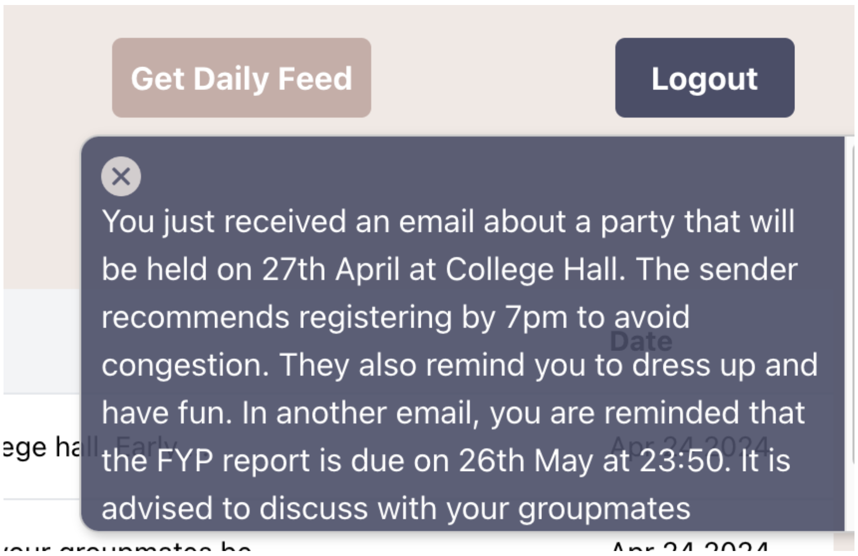
3.5. Daily Summary

Another notable feature is the daily summary function as pointed out in Figure 13. It aims to provide users with a concise overview of the most important emails received throughout the day, as determined by the underlying machine learning (ML) models, as well as emails from specific addresses defined in the whitelist.

The daily summary feature sets itself apart from email summaries in Section 3.2 by focusing exclusively on emails received within the current day. This ensures that users can effectively allocate their attention to the most relevant information in a timely manner, without being overwhelmed by a backlog of older emails.

Furthermore, the daily summary function takes into account a whitelist of specified email addresses. Emails originating from these addresses are automatically included in the summary, regardless of their ML-assigned importance level. This allows users to ensure that emails from specific individuals or organizations, such as colleagues, clients, or important contacts, are always highlighted and promisingly acknowledged within the summary.

By combining the ML-based email classification and the whitelist filtering, the daily summary function provides users with a curated selection of emails that demand immediate attention, allowing them to efficiently manage their inbox and prioritize their workflow. This feature enhances productivity by reducing the time and effort required to sift through a large volume of emails, ensuring that critical messages are promptly addressed.



You just received an email about a party that will be held on 27th April at College Hall. The sender recommends registering by 7pm to avoid congestion. They also remind you to dress up and have fun. In another email, you are reminded that the FYP report is due on 26th May at 23:50. It is advised to discuss with your groupmates

Figure 13 Daily Feed from incoming emails that are categorised as most important or from whitelisted email addresses.

3.6. Smart Search

The traditional search functionality in Microsoft Outlook is designed to help users find specific emails or related content based on their search queries which is also implemented as displayed in Figure 14. When a user enters a search term or keyword in the search bar, the email client performs a search operation within the user's mailbox or selected folders to identify relevant emails[17]. The search operation typically involves scanning various attributes of the emails, such as the sender's name, recipient's name, subject line, email body, and any associated metadata. The email client may also consider additional factors like date range, attachment names, or specific folders to narrow down the search results[18].

[17] Microsoft Outlook. How to search in Outlook - Microsoft Support. support.microsoft.com. Accessed April 24, 2024. https://support.microsoft.com/en-us/office/how-to-search-in-outlook-d824d1e9-a255-4c8a-8553-276fb895a8da

[18] Microsoft Outlook. How to search in Outlook - Microsoft Support. support.microsoft.com. Accessed April 24, 2024. https://support.microsoft.com/en-us/office/how-to-search-in-outlook-d824d1e9-a255-4c8a-8553-276fb895a8da

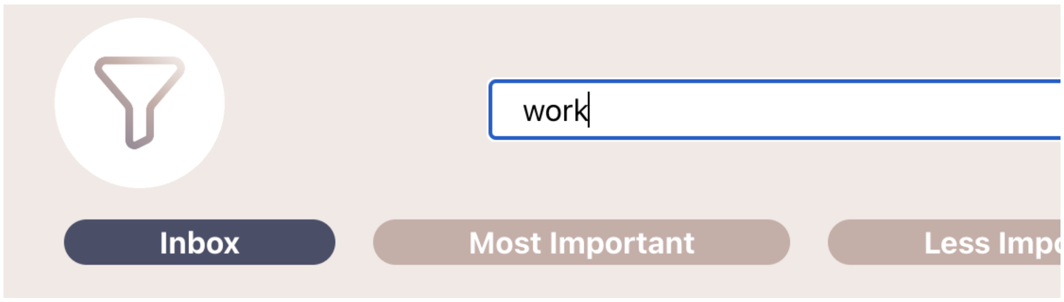
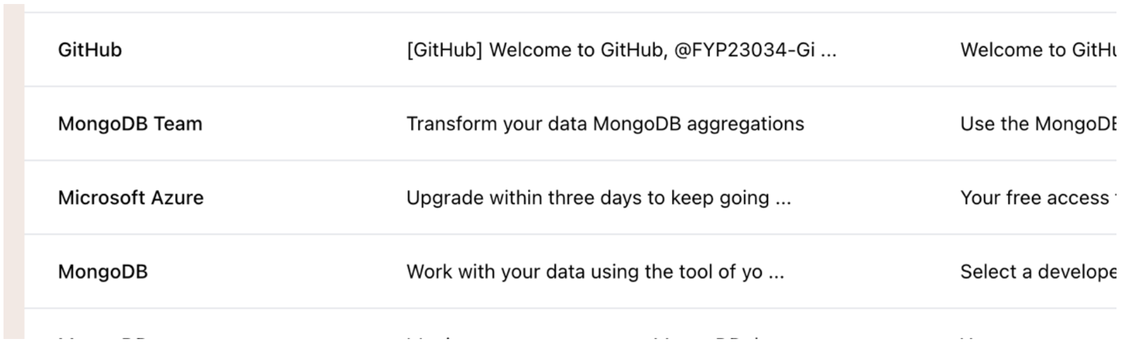


Figure 14 Traditional search bar with keyword "work".

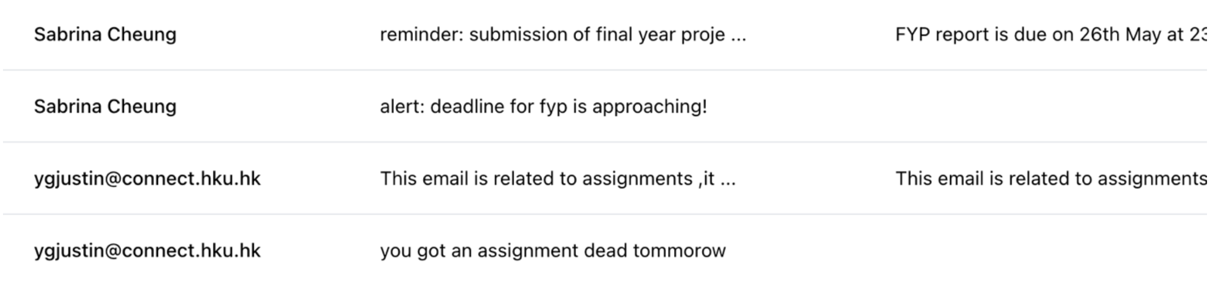
While traditional search relies on exact keyword matches, Smart Search takes into account the context and meaning behind the search query. For instance, if a user enters the term "work" in the search bar, the regular search returns any emails containing the word "work" in any context (Please Refer to Figure 15). However, Smart Search goes a step further by analyzing the user's search intent and generating additional keywords related to "work".



GitHub	[GitHub] Welcome to GitHub, @FYP23034-Gi ...	Welcome to GitHu
MongoDB Team	Transform your data MongoDB aggregations	Use the MongoDE
Microsoft Azure	Upgrade within three days to keep going ...	Your free access
MongoDB	Work with your data using the tool of yo ...	Select a develope

Figure 15 Tradition search result with keyword "work" as the query.

By understanding the context of the search query, Smart Search can identify the user's specific focus within the broader topic of "work." It recognizes relevant terms such as "project," "meeting," "deadline," or "colleague," and uses these generated keywords to refine the search results (Noted in Figure 16). This way, the Smart Search feature provides users with a more customized and targeted set of email results that are highly relevant to their specific needs.



Sabrina Cheung	reminder: submission of final year proje ...	FYP report is due on 26th May at 23
Sabrina Cheung	alert: deadline for fyp is approaching!	
ygjustin@connect.hku.hk	This email is related to assignments ,it ...	This email is related to assignments
ygjustin@connect.hku.hk	you got an assignment dead tommorow	

Figure 16 Smart Search result using keyword "work".

Smart Search goes beyond simple keyword matching by employing external techniques, which are discussed in other reports, to extract meaningful information from email content. By analyzing the metadata, Smart Search can identify the most relevant emails that align with the user's search intent.

The result is a more efficient and accurate search experience. Users can quickly find the emails they need without having to sift through irrelevant search results. Smart Search saves time and enhances productivity by presenting users with a curated selection of emails that are most likely to contain the information they are looking for.

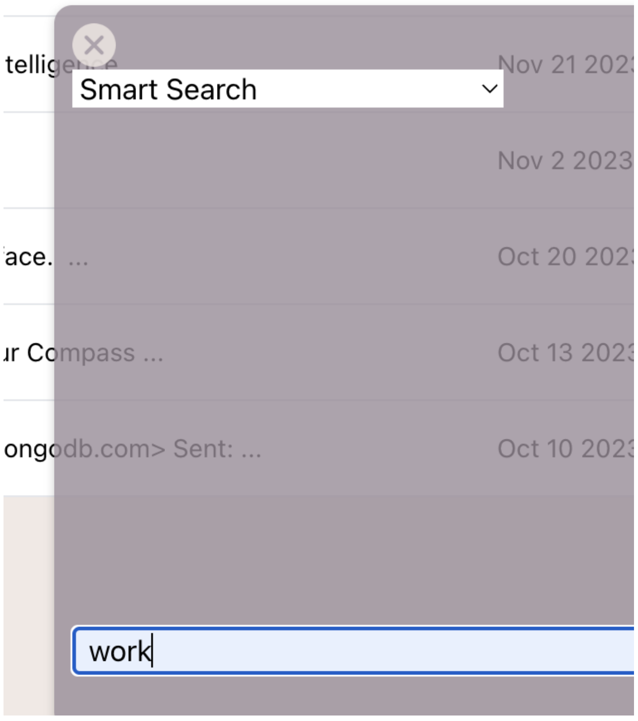


Figure 17 Demonstration of Smart Search

3.7. Preference update with Natural Language Processing (NLP) Input

In order to better meet the specific requirements of individual users, the web application features a preference update function. This functionality allows users to utilize NLP capabilities to specify their preferences and update the importance score rating calculation employed by the ML model in the chat box shown in Figure 18.

By leveraging NLP, users can provide explicit instructions or feedback regarding their desired importance levels for different types of emails. For example, they can specify that emails from certain senders or with specific keywords should be considered as highly important, while others may be categorized as less important or least important.

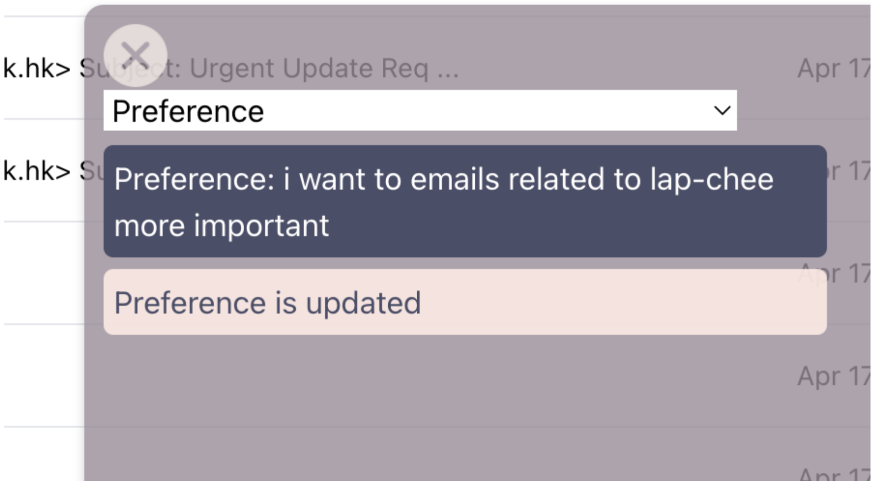

k.hk> Subject: Urgent Update Req ...
Apr 17
Preference
Preference: i want to emails related to lap-chee more important
Preference is updated

Figure 18 A text in chat box to adjust the importance level of "lap-chee" to a higher level by preference.

It enables users to gain more control over the ML model's rating system, ensuring that the importance scores align more closely with their personal preferences and priorities. This customization enhances the accuracy and relevance of the email client's classification of incoming emails. After the submission of preference described in Figure 18, emails related to "Lap-Chee" would receive a higher rating than before. As seen in Figure 19 and 20, the criticalness of "Lap-Chee" has been rated 147.39% higher than before.

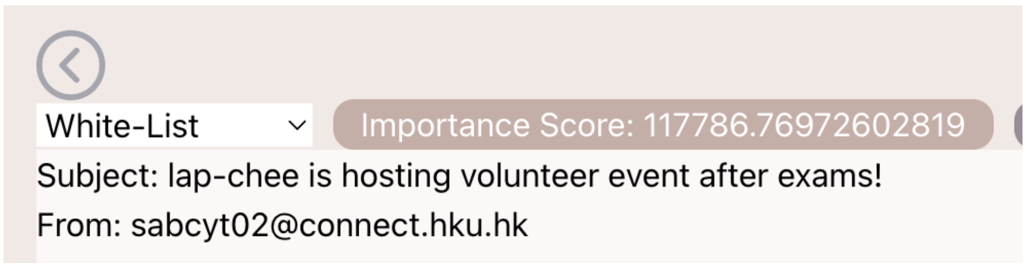

White-List
Importance Score: 117786.76972602819
Subject: lap-chee is hosting volunteer event after exams!
From: sabcyt02@connect.hku.hk

Figure 19 Importance rating of Lap-Chee related email after the preference update.

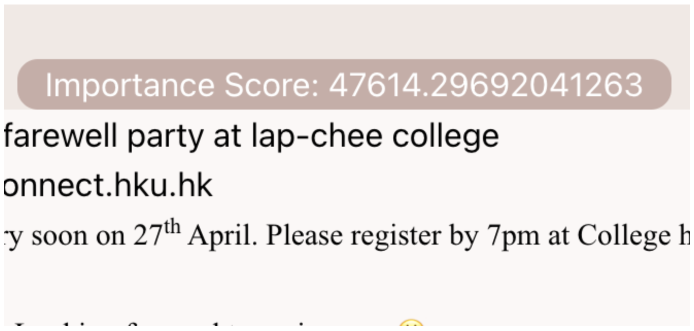

Importance Score: 47614.29692041263
farewell party at lap-chee college
onnect.hku.hk
y soon on 27th April. Please register by 7pm at College h

Figure 20 Importance rating of Lap-Chee related email before the preference update.

However, it is important to note that while the preference update feature allows users to modify the importance score rating system going forward, it does not retroactively update the importance ratings of previously received emails. This is because the importance scores for past emails are based on accumulated data and historical patterns and updating them would require significant computational resources and potentially disrupt the overall email classification system.

Nonetheless, the preference update feature empowers users to fine-tune the ML model's importance rating calculation, enabling a more personalized and tailored email experience. By incorporating user preferences through NLP, the email client can better adapt to individual needs and accurately prioritize incoming emails based on the revised importance scores.

4. Further Improvements

4.1. Testing System

The testing approach employed throughout the engineering process involved relying on the "Inspect" mode in the browser to manually identify areas of development that exhibited abnormalities or code redundancy. However, this approach was time-consuming and considered relatively naïve in terms of its effectiveness. One sort of correction is to adopt a comprehensive testing framework that covers different levels of testing pyramid.

Begin with unit testing, framework like Jest and Mocha can evaluate the behaviors of individual components in insolation to ensure their normal operations with dynamic states and inputs[19]. Unit tests have immediate execution that provides prompt feedback on correctness of components. This allows developers to capture and resolve issues at the earliest possible before the engineering is propagated to a higher level.

Next is End-to-End (E2E) Testing. This refers to the testing on entire application workflow (UI, data flow, and external integration) from an end-user's perspective[20]. It verifies the collaboration of components and business logic, guaranteeing the overall user experience. Cypress, Selenium and Puppeteers are some exemplars of E2E testing tools in market which automate the E2E testing process upon configurations.

Automation is the final procedure. Continuous Integration and Deployment (CI/CD) pipeline is critical to streamline the testing recurrently. Once new code changes are made, the CI/CD pipeline automatically triggers a series of tests to verify the integrity and

[19] Smolin T. Testing your React App with Mocha, Chai and other beverages…. Medium. Published January 29, 2020. Accessed April 24, 2024. https://medium.com/@tatismolin/testing-your-react-app-with-mocha-chai-and-other-beverages-e9a16ca7b9bb

[20] Schmitt J. What is end-to-end testing? CircleCI. Published April 5, 2022. https://circleci.com/blog/what-is-end-to-end-testing/

functionality of the application[21]. It promotes a more collaborative and efficient development environment. With its frequent code integration and deployment, it fosters a culture of continuous improvement and iteration. Developers can work in smaller increments and receive prompt validation of their changes, leading to faster and more reliable software delivery.

4.2. Package management

There were several failures of execution reported due to code integration and lack of package version support. To address these challenges, the project's package manager npm provides two plugin instances: "moment-locales-webpack-plugin" and "lodash-webpack-plugin". These plugins are specifically designed to optimize the major dependencies (Moment.js and Lodash.js) of React applications by removing unused locales and functions from the final bundle[22].

By doing so, they eliminate errors caused by conflicting packages and reduce load times by minimizing the file size, hence eliminates the errors triggered as a result of malfunction from clustering packages and shortens load time by lessening the file size.

5. Conclusion

This report covers the front-end development of Inbox Genius, including its implementations, and functionalities explained with results presented, finishing up with the areas of improvements. As important as how back-end service produces the service and serves as the core of the product, front-end is the ground where result and business logic are fully demonstrated. Despite the web application may not be strictly formatted as an industry product, it introduces a new form of email service that distinguishes itself from the existing third-party email assistants which implies its potential of leveraging and gaining recognition. With an appropriate amendment upon the development, the productivity of users will be profited and the diversity in the email service market will be maintained in long term.

[21] Fosco M. What is a CI/CD pipeline? CircleCI. Published October 6, 2022. Accessed April 25, 2024. https://circleci.com/blog/what-is-a-ci-cd-pipeline/

[22] Agnoletto F. Webpack mastery: How to bundle momentjs and lodash for production. Medium. Published November 12, 2018. Accessed April 24, 2024. https://medium.com/@francesco.agnoletto/webpack-mastery-how-to-bundle-momentjs-and-lodash-for-production-1384a7e853d2

Bibliography

1.
MDN web docs. Call stack - MDN Web Docs Glossary: Definitions of Web-related terms | MDN. developer.mozilla.org. Accessed April 24, 2024. https://developer.mozilla.org/en-US/docs/Glossary/Call_stack

2.
Stack Overflow. Stack Overflow Trends. insights.stackoverflow.com. Accessed April 26, 2024.
https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Csvelte%2Cangularjs%2Cvuejs3

3.
React. Synchronizing with Effects – React. react.dev. Accessed April 24, 2024.
https://react.dev/learn/synchronizing-with-effects

4.
React Async. Introduction | next | React Async. React-async.com. Published December 8, 2020. Accessed April 24, 2024. https://docs.react-async.com/

5.
Jadhav S. The importance of color in web design and how to choose the right palette. Medium. Published March 27, 2023. Accessed April 24, 2024.
https://medium.com/@siddhantjadhav445/the-importance-of-color-in-web-design-and-how-to-choose-the-right-palette-72f63dd7f6be

6.
Qawwas O. Separation of Concerns on the Front-end With React. Medium. Published July 28, 2022. Accessed April 24, 2024. https://engineering.teknasyon.com/separation-of-concerns-on-the-front-end-with-react-fd5d4afcc298

7.
Kopf B. The Power of Figma as a Design Tool. Toptal Design Blog. Published 2018.
https://www.toptal.com/designers/ui/figma-design-tool

8.
Boroumand A. A Web Developer’s Guide to Browser Caching. Medium. Published July 25, 2017. Accessed April 24, 2024. https://medium.com/@steelcityamir/a-web-developers-guide-to-browser-caching-cc41f3b73e7c

9.
Microsoft. React single-page application using MSAL React to authenticate users against Azure AD for Customers - Code Samples. learn.microsoft.com. Published March 12, 2024. Accessed April 24, 2024. https://learn.microsoft.com/en-us/samples/azure-samples/ms-identity-ciam-javascript-tutorial/ms-identity-ciam-javascript-tutorial-1-sign-in-react/

10.

The HTTPS-Only Standard. The HTTPS-Only Standard - Introduction to HTTPS. https.cio.gov. Accessed April 24, 2024. https://https.cio.gov/faq/

11.
Gated. 67 Percent of People Feel Overwhelmed by Their Email Inbox According to New Inbox Intelligence Report from Email Management Solution Gated. www.prnewswire.com. Published October 26, 2022. Accessed April 24, 2024. https://www.prnewswire.com/news-releases/67-percent-of-people-feel-overwhelmed-by-their-email-inbox-according-to-new-inbox-intelligence-report-from-email-management-solution-gated-301659242.html

12.
Microsoft Outlook. How to search in Outlook - Microsoft Support. support.microsoft.com. Accessed April 24, 2024. https://support.microsoft.com/en-us/office/how-to-search-in-outlook-d824d1e9-a255-4c8a-8553-276fb895a8da

13.
Smolin T. Testing your React App with Mocha, Chai and other beverages…. Medium. Published January 29, 2020. Accessed April 24, 2024. https://medium.com/@tatismolin/testing-your-react-app-with-mocha-chai-and-other-beverages-e9a16ca7b9bb

14.
Schmitt J. What is end-to-end testing? CircleCI. Published April 5, 2022. https://circleci.com/blog/what-is-end-to-end-testing/

15.
Fosco M. What is a CI/CD pipeline? CircleCI. Published October 6, 2022. Accessed April 25, 2024. https://circleci.com/blog/what-is-a-ci-cd-pipeline/

16.
Agnoletto F. Webpack mastery: How to bundle momentjs and lodash for production. Medium. Published November 12, 2018. Accessed April 24, 2024. https://medium.com/@francesco.agnoletto/webpack-mastery-how-to-bundle-momentjs-and-lodash-for-production-1384a7e853d2