UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

Good!!

Computer Science Honours
Final Paper 2018

See comments
-h.

Title: A Portable Large Volume Email Retrieval System

Author: Monyemoratho Breyden

Project Abbreviation: FINDMAIL

Supervisor: Associate Professor Hussein Suleman

Category	Min	Max	Chosen
Requirement Analysis and Design	0	15	0
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	20
System Development and Implementation	0	10	10
Results, Findings and Conclusion	0	20	20
Aim Formulation and Background Work	0	10	10
Quality of Paper Writing and Presentation	0	10	10
Adherence to Project Proposal and Quality of Deliverables	0	10	10
Overall General Project Evaluation [requires explicit motivation from the project supervisor]	0	0	0

A Portable Large Volume Email Retrieval System

Breyden Monyemoratho
University of Cape Town, South Africa
Computer Science
mnybre002@myuct.ac.za

CCS CONCEPTS

• **Information systems → Information retrieval →** Specialized information retrieval • **Software and its engineering** → Software libraries and repositories • **Software and its engineering** ➝ Software Portability

KEYWORDS

Portable; Searchable; Email formats; Archives; Indexing; Parsing; User interface; Query System

ABSTRACT

Email is present in all facets of daily life. A remarkable amount of information resides in email archives. This paper describes an attempt to develop~~ment of~~ a portable email system that allows users to input large email archives in maildir or mbox format and accurately and efficiently browse and search over the archive offline. This approach to designing systems for preservation and offline access is useful in areas with limited Internet bandwidth such as in most African countries. Experimental results confirm that users were satisfied with the general design of the system and moreover, that this system is effective and efficient.

1. INTRODUCTION

In the working world, there is often a need to search and browse through very large collections of emails, to track down individuals or to verify decisions, etc. For convenience, many users will either delete or archive email after it has been handled. If they choose to archive their email, these archives can later become large and cumbersome to search through, especially after a long period of time has passed. This can be attributed to many factors, including poor personal information management and large amounts of high priority email. Whittaker et al. [15] termed this "email overload".

With the problem of "email overload", there is also the issue of archives becoming obsolete through *software aging* [8]. In order to combat obsolescence and improve longevity, various preservation strategies need to be considered.

A possible solution to address the email overload and obsolescence issues, is to use a portable offline searchable email archive that ~~which~~ handles multiple email formats (such as mbox and maildir). The searchability feature would allow for specific emails to be retrieved from the large archive (managing email overload), whilst e the portable and offline features would make the archive less likely to become obsolete in the short-term. This solution is the one proposed in this paper.

Taking the above into consideration, we created a Web application to facilitate portability and offline searchability, that allows for multiple email formats as inputs.

The project is divided into two logical sections which when used in conjunction, solves the overall project. The two sections are as follows:

1. *Pre-Processing:*

   This involves parsing and indexing of the inputted archives of various email formats. Parsing will extract and structure relevant information from the inputted archive, whilst e indexing involves creating indices from the parser output.

2. *Email Processing:*

   This involves the user interface and a query system that allows for fast and efficient

YOU CANNOT HAVE AN OVERLAPPING REPORT WITH SHIVAAN!
IT WILL BE CONSIDERED AS PLAGIARISM! REWRITE IN YOUR OWN WORDS!

retrieval of emails. The user interface should display emails clearly to the user and allow for ease-of-use. The query system should be able to handle various queries.

WHICH BIT IS YOURS?

1.1. Project Significance

We hope this project will help individuals better manage their emails from large archives and provide them with fast and accurate search and browse functions, making it more likely for the information to be retrievable in later years. ~~In addition, by releasing the resulting tool from this project as open source, the community will be able to freely use our work and expand and improve on our tools.~~

1.2. Project Aims

This project had the primary aim of developing a distributable repository which consists of the following features:

1. *Allow for input of large email archives in multiple email formats:* large maildir and mbox archives.
2. *Portable:* Is hardware independent, requires minimal software installation and only requires a standard Web browser.
3. *Provide fast and accurate offline search and browse functions over the archive/collection.*
4. Has user interface that represents emails in a easy to understand way, and is usable.
5. Implement a reliable preservation policy to ensure preservation against hardware and/or software obsolescence.

1.3. Project Structure

The rest of this paper present background work and how THE findmail system was designed. Various experimental results are then discussed to illustrate how findmail is effective and efficient. Finally, ethical considerations, conclusions and future work are presented.

2. BACKGROUND

2.1. Digital Collections:

For developed countries many preservation techniques can be implemented, however this is not the case for developing countries (such as in Africa) [12]. In developing countries, most preservation techniques cannot be implemented due to insufficient resources and poor/expensive cost of Internet bandwidth.

A particular way of preserving digital collections(including email archives) that works for developing countries is through using the principle of simplicity [8]. An illustration of this could be the use of XML plain text documents to store information and metadata, making it more likely for the information to be retrievable in later years. Focusing on simplicity also provides easier interconnection, extension and modification of the features of a specific system, allowing for the system to function on multiple platforms(portable). The concept of portability is important for email, as email users use multiple platforms to access their email, and the email itself can be stored in different formats.

Suleman et al. [13] developed CALJAX, a generic hybrid (online-offline) repository management and access system based on a strong AJAX foundation. It allows integration of content from a local source with content from a remote source, with the only requirement being a Web browser. XML plain text documents were used to store information, making it more likely for the information to be portable, preservable and accessible through a Web browser.

Expanding on the issue of poor Internet bandwidth, is the idea of having hybrid online-offline digital collections to counteract this issue. Online and offline collections present both advantages and disadvantages, thus a hybrid digital collection(online-offline repository) could interleave advantages from both, and potentially aid in preservation [13]. A hybrid system will however not be in the scope of this project, as there are many complications that come with creating such a system.

SUCH AS?

2.2. Email Archives:

Some existing software projects around email archives include Windows Mbox Viewer[ ](MV), Mairix [10] and Mailpile [6]. WMV [11] displays mbox files on the

REF?

user's screen via a simple user interface. It runs offline but is a program specifically for Windows. It also does not provide search functionality over the archive. The other downsides are the fact that it does not cater for other email formats and is not portable across operating systems.

ROOF!

Mairix and Mailpile include indexing and search functionality, but are not suitable either in terms of preservation, portability or offline use. Mairix [10] is an email indexing and searching tool that works with maildir, MH or mbox formats. It works offline but is mainly for Linux systems. Since it involves installation and is not portable across non-Linux operating systems, it is unusable in this project.

Mailpile [6] is similar to Mairix; it also indexes mbox and maildir formats, however Mailpile is an email client and personal Web mail server. It also has a much better user interface(in comparison to WMV) that is based on Gmail. It works on multiple browsers but does not have specific offline usability. It was made using Python, JS and HTML5, and is the closest work to the one we propose in this paper.

3. DESIGN OF FINDMAIL

Our approach was split into two main sections of pre-processing and email processing.

3.1. Pre-processing

3.1.1. Parser

Before indexing can occur, email archives (sources/inputs) of different formats, need to be streamed into the application. The parser extract relevant data (constituents) from the email archives and pass this data on to the indexer. The parser was created using the Python mailbox module [9].

NOT YOUR PART

3.1.2. Indexer

The indexer then generates indices for searching and browsing of the archive. In order to enable access via a Web browser, all indices were stored as XML files that can be parsed from within a Javascript environment. Pre-indexing is necessary to speed up access to result set. The indices for browsing are in sorted order of various fields such as from, date and subject. Pre-indexing for search generate inverted files that are used with an extended boolean search implementation in the access Web interface. The indexers was also created using Python.

3.2. Email Processing

3.2.1 User Interface(UI)

The access Web interface is a standard email web interface offering the user search and browse functions. It was developed using a user centered approach in order to understand users' needs and preferences. It consists of mainly static HTML, CSS and Javascript to display the relevant result when a user invokes one of the services.

W

Sp.

3.2.2 Query

The query system, using extended boolean implementation retrieves relevant emails from the email archive, by using the indices generated by the search indexer.

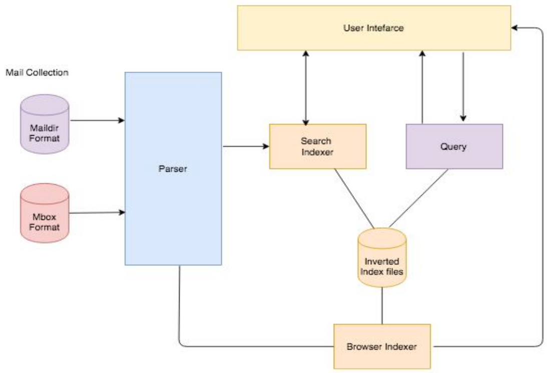


Fig. 1. Overview of the FINDMAIL system

In Fig.1, we show the popular email formats: maildir and mbox, being inputted to the parser. The parser then sends its output to the two indexers. The browser indexer will create indices to facilitate browsing of the email, whilst the search indexer will create indices for the search functionality. Both of these indexers will interact will the user interface to provide the services of browsing and searching to the user.

The design scheme helps to simply the preservation of the archive by making it easy for the archive to be migrated to other hardware environment or software platforms. This design scheme also highlights how simple repository storage and service architectures can be designed.

SAME AS SHIVAAN!

X EXPERIMENTAL DESIGN?

— WHAT EMAIL DID YOU USE?
— HOW WERE USERS SAMPLED?
— WHAT WERE THE TASKS?

4. EXPERIMENTAL RESULTS

4.1. System Usability Testing:

The usability test was conducted on a near final version of the software. The test was conducted with a total of 20 users to assess attributes of the system that make it understandable, learnable, easy-to-use and attractive. The test lasted approximately 20 minutes. Ethical clearance was obtained from the Science Faculty Research Ethics Committee and the Department of Student Affairs. Before taking part in the usability test, participants were asked to sign a consent form informing them of the anonymity of their results. On completion of the task scenarios, users were asked to fill in a system usability score questionnaire to determine the usability of the system . On completion of the usability study, users were compensated for their time with a standard fee as specified by the Department. Tests were conducted in an uncontrolled environment of the Computer Science Senior laboratory. Participants accessed a Web page (standard email page) that presents a browse view of the collection, using a laboratory computer through Chrome browser.

Ref?

Rather than observe users throughout the test process, users were allowed to conduct tasks and answer questions independently within the 20 minutes of the usability test session. The reason for this is that users who are observed will alter their behavior and may become nervous, resulting in mistakes and errors affecting results [52]. However, if users experienced particular difficulties in completing a task or found the instructions to be ambiguous, the facilitator could be asked for help or clarification. Most users completed tasks successfully.

? Sp.

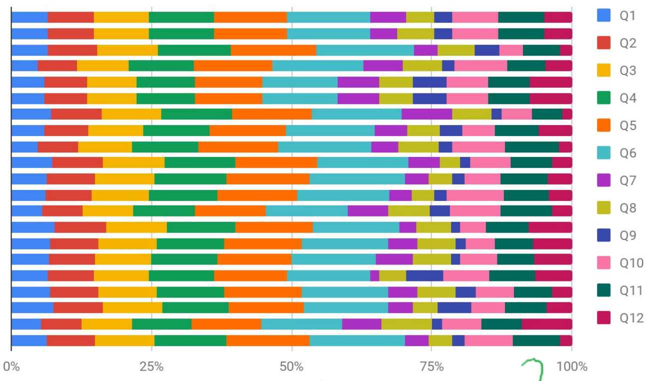
Responses were constrained to a Likert scale that ranges from ‘strongly disagree’ to ‘strongly agree’. The raw data and mode data from the usability test is provided in A appendix A. Figure 1 shows the cumulative distribution of the responses for each question. Users wanted clearly defined visuals and graphics. This was apparent when they struggled to identify chained mails, wanted a button near the search area and failed to identify emails with attachments. Thus, More information should be added for ease of understanding. When looking at design, all the users were happy with the basic and minimalistic design. A Majority of users (>52%) ~~however;~~ believed the system was not lacking in intuitivity. The system had maximal cognitive flow (little friction and confusion when the user was using the system). ~~This was a good benchmark for the Project.~~

THIS SEEMS SMALL?

Generally the ~~raw~~ data, ~~as well as the mode responses;~~ ES do not appear to indicate any particular usability issues.

The overall feedback was positive, with all criticism being constructive and leading to consistent improvements and updates to the design of the user interface.

WHAT DO COLOURS MEAN?



AXIS LABELS?

Fig. 2. Cumulative distribution of the system usability score.

4.2 Performance Testing:

Experiments were conducted to measure the time it takes to download and display the entire content of a web page for both the search and browse functions over collections of various sizes. The data collection used were simple text files filled with test data. This allowed tight control over the number of files, as the exact number of files could be generated for each test. The test was conducted on collections containing various number of files (2000, 4000, 6000, 8000, 10000, 12000) and all the files contained the same email items, as the load time is affected by the number of http connections needed to download items, item size and types.

uc

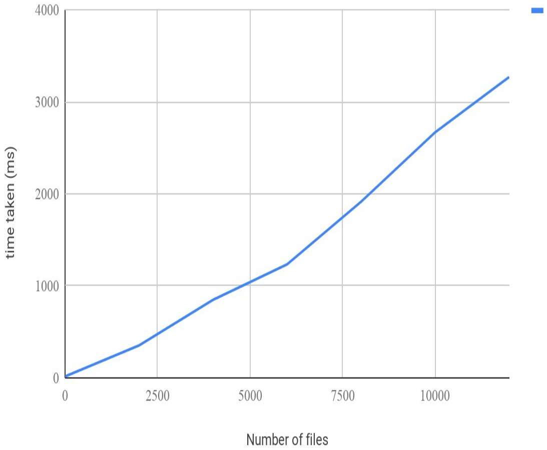
The browsing test was conducted by loading 3 pages and the load times were recorded and averaged. The results of the time taken to generate a browse view ARE is presented in Figure 3. This time is linear but does not increase much with increases in collection size. Similar results were obtained for the search function.

DATA!

MORE DETAIL NEEDED

All performance testing was done on A Mac Book Pro 5.2. There are a few caveats that we were aware of for this kind of performance measurement (,) as listed below:

1. The available system memory and CPU.
2. The browser used affected the Javascript execution and rendering speed.



?

Figure 3. Time take to browse a collection

4.3 Relevance of the search results:

To measures the relevance of the search results, The test was conducted using our personal collections containing 10 files. For a certain random query term(s) present within the collection, precision and recall were measured. Table 1 shows that the system returns approximately all the relevant result sets. It stands to reason that for large databases with millions of documents (,) recall and precision cannot be easily determined, as it depends on knowing the precise number of relevant documents in the collection and result set. ~~Moreover, User happiness can only be measured by relevance to an information need, not by relevance to queries.~~

Table 1: Precision and recall for ~~random~~ SAMPLED query terms present in collection.

Query/Term	***Precision***	***Recall***
gary	1	1
projects	1	1
technical	1	1
Research	1	2/3
participant	1	1

4.4 Portability Testing:

The developed systems' look and feel with respect to the UI and functional features are similar on multiple OS and multiple browsers.It can be run on Windows 10, mac OS, Linux and any standard Web browser versions without any change in the behavior, thus can be ported from one environment to another.

DATA?

5. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

Ethical issues were identified in the testing, software implementation and data handling stages of the project. Each will be discussed in further detail below.

5.1. Testing:

We applied to the Faculty of Science Research Ethics Committee for ethical clearance, in order to test the usability of the system on students. All user testing was conducted through a simple surveys and usability testing, which did not raise any ethical issues.

5.2. Software:

This project is declared open source. This is to encourage further development and improvement to our software.

5.3. Data:

We sourced data from our own personal Gmail inboxes (Shivaan Motilal has a 680 MB inbox unzipped, Breyden Monyemoratho has a 670 MB inbox unzipped) and compiled from the Enron email dataset containing approximately 1.5 million emails (423MB, tarred and zipped) [2], which did not raise any ethical issues.

6. CONCLUSIONS AND FUTURE WORK

Experiments have confirmed that the developed proof of concept is intuitive, portable and effective for browsing and searching over email archives. Findmail has demonstrated that it is possible to leverage a simpler architecture and Web technology to enable fast and accurate browsing and searching over email archives in developing countries with limited Internet bandwidth.

This simple approach can be extended further in the following ways :

116

1. Leveraging the feature of AJAX framework to enable integration of content from a local source with content from a remote source, thus allowing the user full access to the most current content.
2. Integrating the current solution with tools and services that facilitate preservation, such as logging and integrity checking.
3. For greater efficiency, splitting browsing and search indices into shards. Thus the speed of both operations will be constant irrespective of the size of the collection.
4. Some (advertising) email messages contain unsightly links that are bundled together with the message body. There is no easy way to ascertain if these links will be useful to the user or not, however the display of these links can be improved on.
5. The display of chained email to the user can be improved on. Chained messages are currently separated into parts but not displayed as such to the user.
6. Using Dublin Core as the metadata scheme to ensure conformance to international standards and a universal understanding of the metadata.

7. ACKNOWLEDGEMENTS

I would like to thank my project partner Motilal Shivaan and project supervisor, Associate Professor Hussein Suleman for his commitment and help throughout the course of this project. Finally, my sincere thanks and appreciation are extended to University of Cape Town's Computer Science department for funding our participant remuneration and supplying experimental space.

8. REFERENCES

[1] CALO Project. Enron Email Dataset, 2015.DOI: https://www.cs.cmu.edu/~enron/

[2] Centre for Curating the Archive. The Digital Bleek and Lloyd, 2018. DOI: http://lloydbleekcollection.cs.uct.ac.za/

[3] Jakob Nielsen and Rolf Molich. 1990. Heuristic evaluation of user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90), Jane Carrasco Chew and John Whiteside (Eds.). ACM, New York, NY, USA, 249-256. DOI=http://dx.doi.org/10.1145/97243.97281

[4] JSDOM. JavaScript browser simulator, 2018. DOI: https://github.com/sttk/jsdom-browser/

[5] Facebook. Jest. Javascript testing tool, 2018. DOI: https://facebook.github.io/jest/

[6] Mailpile. An email client, 2018. DOI: https://www.mailpile.is/

[7] David L. Parnas. Software aging. In *Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on* (pp. 279-287). IEEE. May, 1994.

[8] Lighton Phiri and Hussein Suleman. In search of simplicity: Redesigning the digital bleek and lloyd. *DESIDOC Journal of Library & Information Technology,* (pp 32-34), 2012.

[9] Python 3 Standard Library. Mailbox module, 2018. DOI: https://docs.python.org/3/library/mailbox.html/

[10] SourceForge. Mairix. Programme for indexing and searching mail, 2009. DOI: https://github.com/rc0/mairix/

[11] SourceForge. Windows Mbox Viewer, 2015. DOI: https://sourceforge.net/projects/mbox-viewer/

[12] Hussein Suleman. An African Perspective on Digital Preservation. In *Multimedia Information Extraction And Digital Heritage Preservation* (pp. 295-306), 2008.

[13] Hussein Suleman, Marc Bowes, Matthew Hirst, and Suraj Subrun. Hybrid online-offline digital collections. In *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '10*, ACM Press, 421-425, 2010.

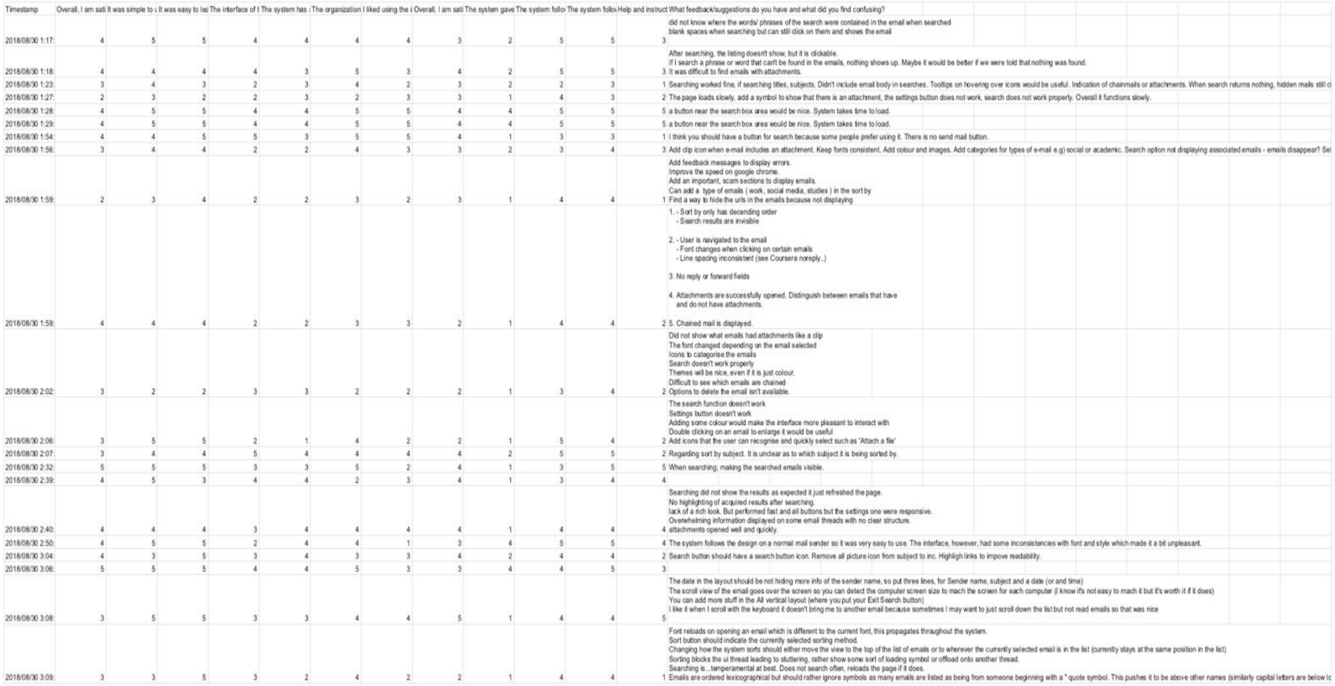
[14] Visual Studio Code. A compact code editor and IDE, 2018. DOI: https://code.visualstudio.com/

[15] Steve Whittaker, and Candace L. Sidner. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 276-283). ACM. April, 1996.

Notes

- Where are the research questions?
- Need a detailed separate experimental design section
- Need actual data to support conclusions
- More detail for experiments
- Make sure to only write on your part of the project

APPENDIX A: The raw data and mode data from the usability test:



Timestamp	Overall, I am sati	It was simple to	It was easy to lea	The interface of t	The system has	The organization	I liked using the i	Overall, I am sati	The system gave	The system follo	The system follow	Help and instruct	What feedback/suggestions do you have and what did you find confusing?
2018/08/30 1:17:	4	5	5	4	4	4	4	3	2	5	5	3	did not know where the words/ phrases of the search were contained in the email when searched blank spaces when searching but can still click on them and shows the email
2018/08/30 1:18:	4	4	4	4	3	5	3	4	2	5	5	3	After searching, the listing doesn't show, but it is clickable. If I search a phrase or word that can't be found in the emails, nothing shows up. Maybe it would be better if we were told that nothing was found. It was difficult to find emails with attachments.
2018/08/30 1:23:	3	4	3	2	3	4	2	3	2	2	3	1	Searching worked fine, if searching titles, subjects. Didn't include email body in searches. Tooltips on hovering over icons would be useful. Indication of chainmails or attachments. When search returns nothing, hidden mails still c
2018/08/30 1:27:	2	3	2	2	3	2	3	3	1	4	3	2	The page loads slowly, add a symbol to show that there is an attachment, the settings button does not work, search does not work properly. Overall it functions slowly.
2018/08/30 1:28:	4	5	5	4	4	5	5	4	4	5	5	5	a button near the search box area would be nice. System takes time to load.
2018/08/30 1:29:	4	5	5	4	4	5	5	4	4	5	5	5	a button near the search box area would be nice. System takes time to load.
2018/08/30 1:54:	4	4	5	5	3	5	5	4	1	3	3	1	I think you should have a button for search because some people prefer using it. There is no send mail button.
2018/08/30 1:56:	3	4	4	2	2	4	3	3	2	3	4	3	Add clip icon when e-mail includes an attachment. Keep fonts consistent. Add colour and images. Add categories for types of e-mail e.g) social or academic. Search option not displaying associated emails - emails disappear? Se
2018/08/30 1:59:	2	3	4	2	2	3	2	3	1	4	4	1	Add feedback messages to display errors. Improve the speed on google chrome. Add an important, scam sections to display emails. Can add a type of emails (work, social media, studies) in the sort by Find a way to hide the urls in the emails because not displaying
2018/08/30 1:59:	4	4	4	2	2	3	3	2	1	4	4	2	1. - Sort by only has decending order - Search results are invisible 2. - User is navigated to the email - Font changes when clicking on certain emails - Line spacing inconsistent (see Coursera noreply..) 3. No reply or forward fields 4. Attachments are successfully opened. Distinguish between emails that have and do not have attachments. 5. Chained mail is displayed.
2018/08/30 2:02:	3	2	2	3	3	2	2	2	1	3	4	2	Did not show what emails had attachments like a clip The font changed depending on the email selected Icons to categorise the emails Search doesn't work properly Themes will be nice, even if it is just colour. Difficult to see which emails are chained Options to delete the email isn't available.
2018/08/30 2:06:	3	5	5	2	1	4	2	2	1	5	4	2	The search function doesn't work Settings button doesn't work Adding some colour would make the interface more pleasant to interact with Double clicking on an email to enlarge it would be useful Add icons that the user can recognise and quickly select such as 'Attach a file'
2018/08/30 2:07:	3	4	4	5	4	4	4	4	2	5	5	2	Regarding sort by subject. It is unclear as to which subject it is being sorted by.
2018/08/30 2:32:	5	5	5	3	3	5	2	4	1	3	5	5	When searching; making the searched emails visible.
2018/08/30 2:39:	4	5	3	4	4	2	3	4	1	3	4	4	
2018/08/30 2:40:	4	4	4	3	4	4	4	4	1	4	4	4	Searching did not show the results as expected it just refreshed the page. No highlighting of acquired results after searching. lack of a rich look. But performed fast and all buttons but the settings one were responsive. Overwhelming information displayed on some email threads with no clear structure. attachments opened well and quickly.
2018/08/30 2:50:	4	5	5	2	4	4	1	3	4	5	5	4	The system follows the design on a normal mail sender so it was very easy to use. The interface, however, had some inconsistencies with font and style which made it a bit unpleasant.
2018/08/30 3:04:	4	3	5	3	4	3	3	4	2	4	4	2	Search button should have a search button icon. Remove all picture icon from subject to inc. Highligh links to impove readability.
2018/08/30 3:06:	5	5	5	4	4	5	3	3	4	4	5	3	
2018/08/30 3:08:	3	5	5	3	3	4	4	5	1	4	4	5	The date in the layout should be not hiding more info of the sender name, so put three lines, for Sender name, subject and a date (or and time) The scroll view of the email goes over the screen so you can detect the computer screen size to mach the screen for each computer (I know it's not easy to mach it but it's worth it if it does) You can add more stuff in the All vertical layout (where you put your Exit Search button) I like it when I scroll with the keyboard it doesn't bring me to another email because sometimes I may want to just scroll down the list but not read emails so that was nice
2018/08/30 3:09:	3	3	5	3	2	4	2	2	1	4	4	1	Font reloads on opening an email which is different to the current font, this propagates throughout the system. Sort button should indicate the currently selected sorting method. Changing how the system sorts should either move the view to the top of the list of emails or to wherever the currently selected email is in the list (currently stays at the same position in the list) Sorting blocks the ui thread leading to stuttering, rather show some sort of loading symbol or offload onto another thread. Searching is...temperamental at best. Does not search often, reloads the page if it does. Emails are ordered lexicographical but should rather ignore symbols as many emails are listed as being from someone beginning with a " quote symbol. This pushes it to be above other names (similarly capital letters are below lo