

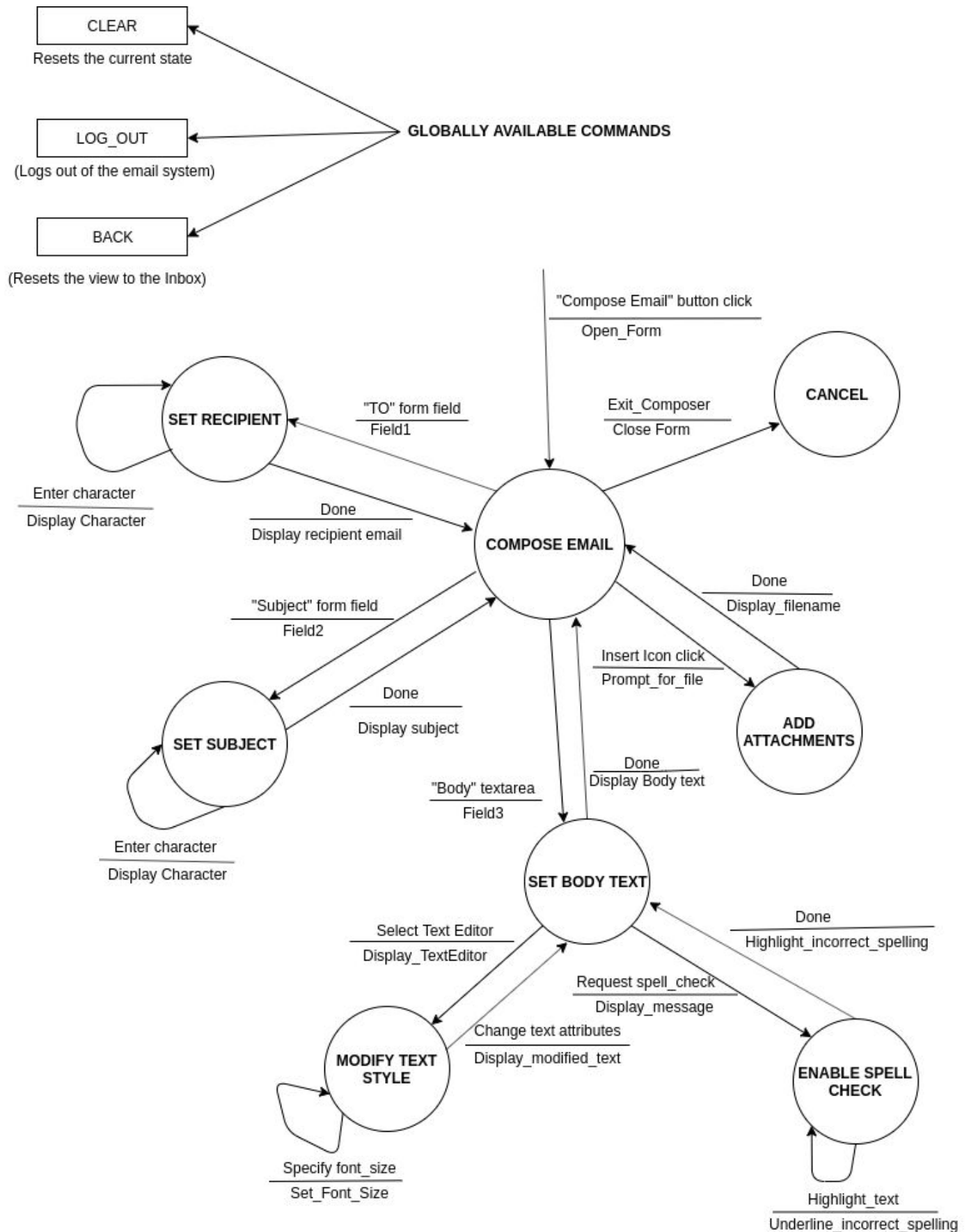
Design and Specification

9/29/19

Semantic Design -

Function Name	Parameters	Description	Feedback	Errors.
Login	email password	Lets the user access the email system and perform actions	A welcome message appears for the specific user who logs in	If the user enters an incorrect password or non-registered email address, a pop-over appears saying out what is incorrect.
Send email	Recipient's email address Subject Email body	Sends an email to the desired person or a group of people	"Email sent successfully" pop-up appears on successful delivery of the email.	If the recipient field is empty, a message appears stating that "Please enter the email of the recipient" If the file attached in the email is too large, a message appears saying "File too large. Please limit the files you send to 20MB"
Search for Email	Search string	Searches the inbox for an email	The set of emails whose text matches the search string are displayed as a list to the user	If none of the emails contain matching text as the search string, a message is shown to the user "No emails found".
Download Attachment	Email body	Downloads the attachment present in the email	A pop-up stating "File downloaded successfully" comes up.	If the file size is too big or the file contains harmful data, a warning message appears saying "Download cancelled. File contains harmful data" or "File size too big"

Syntactic Diagram -



CS5340 Assignment #3

Lexical Model -

1. Compose Email: Allows the creation/modification/formatting of an email to be sent to another person.
 - a. Set recipient's email: add the email address of the recipient of the email that is being composed to a text field
 - b. Set email subject: add the subject of the email being sent to a text field
 - c. Set Email text: Text can be added that will be a part of the email being sent.
 - d. Modify text style: The Email text can be modified, in parts or as a whole in terms of font size, font type, color, etc, with these user actions.
 - e. Add attachment: Allows attachment of a file up to 25 Mb to the email.
 - f. Enable Spell Check: when enabled, the words that the user types in the email or subject are checked in real time, and suggestions are given upon right-click that allow the user to pick alternate words
2. Globally Available Commands: Commands that are accessible from anywhere in the user interface
 - a. Clear: Clears the email being composed and brings the user to a blank "Compose Email" state
 - b. Log Out: Logs the user out of their account and takes them to the sign-in page
 - c. Back: Resets the view to the inbox and places any current work (if composing an email) into the drafts folder

Documentation -

EasyEmail is a new email service that is packed with features to make it extremely user friendly and hassle free. It has all the features offered by other email services such as a powerful search and filter tool, spell checking, and auto suggesting of frequently used phrases as well as several new unique features. Users can choose to use these features to help them keep their emails organized or the features can be turned off to offer a more simplistic experience to the user.

Easy Account Creation

- Create an account the traditional way by entering a minimum amount of information
- Create an account easily by clicking "Log in with Google" or "Sign in with Facebook."

Search Emails

- Easy to find search button at the top of your inbox
- When searching you can choose to search by sender, subject, content, or everything

Inbox Display

- You can choose to display your emails by date (first to last or last to first), or alphabetically by sender or content

Sending and Receiving Emails

- Users will be able to create compose and send emails with ease by clicking on a 'compose' button
- There will be spellcheck and grammar checks that the system does while the user composes the email.

CS5340 Assignment #3

- Once the email is composed user can send the email to any number of the recipient (some recipients can also be kept as hidden from other recipients.)
- The system will notify the user of the delivery of the email to the recipient.
- Users have the option of being notified when a new email is received
- Emails are automatically changed to 'read' once their content is shown for 3 seconds through the inbox. (or manually toggled)

Auto-Remove feature

- Option that allows users to choose if they would like emails to be moved to a certain folder or deleted after a specified amount of time so that your inbox stays uncluttered
- If this feature is activated, there will be 'countdown' displayed on each email in the user's inbox to inform them when each email will be archived or deleted
- Emails can be 'flagged' to save them instead of removing them

Guidelines -

EasyEmail Design Goals:

EasyEmail is going to be designed around the idea that one's Inbox should be a pleasure to look at. The user needs easy access to features that allow for customizable and automatable organization of their inbox, without having to go through excessive menus or complicated settings. The goal of this guideline is to provide a framework to think about when creating the user interface and coding the email client from the ground up.

EasyEmail Design Principles:

- **Minimal Effort:** The interface should prioritize fewer clicks, less mouse movement, more shortcuts (via keyboard or onscreen buttons) for any task the user wants to get done within the application. Automatic inbox cleaning and automatic email forwarding to folders should be simple and easy to set up. Simplicity should be favored over complexity in design wherever possible
- **Intuitiveness:** Traversing the UI of EasyEmail should be intuitive. There needs to be feedback, either textual or contextual, whenever the user makes an action to reinforce good or bad behavior. Related options and features should be grouped and found where it would be most expected. It is important to have user feedback for this design principle to be properly implemented.
 - **Metaphors:** If EasyEmail uses metaphors which reflect experiences/scenarios which are familiar to the user, it will increase the intuitiveness of the program.
- **Aesthetically Pleasing:** EasyEmail needs to be, well, easy to look at. The screen should not feel cluttered with an overwhelming number of options for the user to choose from at once. The colors need to be coordinated, and they should reflect the functionality of the product as opposed to just filling space.

EasyEmail Coding Principles:

- **Scalability:** The EasyEmail platform needs to be coded from the ground up in a way that is scalable to hundreds if not thousands of users. Each user would need to have their stored emails and settings stored separately from other users and their logins (usernames and passwords) would all need to be stored securely as well. As new users are added, they should have memory pre-allocated to account for the emails they will need to store in the future. Email servers should be able to handle a growing user base.

CS5340 Assignment #3

- Consistency: Similar objects and actions should have consistent names in the code. Users can thus learn the interface more easily without being slowed unnecessarily. Code should also have identical formatting, and comments across the board. The final product should also look similar across multiple devices of different screen sizes.
- Feedback: All code written must be reviewed by the rest of the team and be given feedback so that the product can successfully advance towards a common goal. Also included in 'Feedback' is feedback from the user to incorporate into our design, code and overall product.
 - Optimize code for Reader: One aspect of getting feedback is modifying code that one did not necessarily write oneself. Good comments and code organization can help someone who hasn't read that specific code before understand it much more quickly.

Task-Command Analysis

For predicting time required for the task, User tasks can be described by a series of actions. One approach is based on goals, operators, methods, and selection rules (GOMS), by decomposing goals into operators (actions) and then into methods. There are several GOMS based techniques that can be used to analyze and predict task performance. Keystroke-Level Model (KLM) is one of the GOMS predictive techniques for keyboard input. Estimating an execution time for a task is done by listing the sequence of operators and then totaling the execution times.

Execution time is the sum of the time spent executing the different operator types:

$$T_{\text{execute}} = TK + TP + TH + TD + TM + TR$$

K keystroking/ keypressing

P pointing with a mouse to a target

H homing the hand on the keyboard

D drawing a line segment on a grid

M stands for the time a user has to mentally prepare to do an action

R the user has to wait for the system.

Functionality 1 : Logging into the email application

KLM calculation

Action Sequence	Count
Reach for mouse H[mouse]	50 ms
Move pointer to "Name Field" P[menu item]	Click & Drag (1000 ms)

CS5340 Assignment #3

Click on "UserName" field K[mouse]	50 ms
Mental: Remember your name	50 ms
Type	(5-20 characters) 10 ms per character
Move pointer to "Password Field" P[menu item]	Click & Drag (1000 ms)
Click on "Password" field K[mouse]	50 ms
Mental: Remember your password	50 ms
Type	(5-20 characters) 10 ms per character
Move mouse, navigate, click...	50 ms
System time to validate the login to the application	500 ms
Analysis	<p>2900 - 3200 ms, This is optimal because the login to the application only takes approximately around ~ 3 s</p> <p>Here we can also think on improving the system validation time of 500 ms.</p>

Functionality 2 : Send/Receive email

KLM calculation for **send**

Action Sequence	Count
Reach for mouse H[mouse]	50 ms
Move pointer to "Compose Email" button P[menu item]	Click & Drag (1000 ms)
Click on "Compose Email" field K[mouse]	50 ms
Mental: Remember the text that need to be typed	1000 - 5000 ms(depends on the content and user)
Type	(5-20 characters) 10 ms per character
Move pointer to "Send" P[menu item]	Click & Drag (1000 ms)
Click on "Send" field K[mouse]	50 ms

CS5340 Assignment #3

Analysis	<p>Here the variable factors that need to be considered is the time required by the user to compose and structure the email. It can vary based on the content of the email and the user.</p> <p>3160 - 7160 ms, on an average case this is optimal because the receiving and reading a new email application only takes approximately around ~ 5 s</p>
----------	--

KLM calculation for **recieve**

Action Sequence	Count
Reach for mouse H[mouse]	50 ms
Move pointer to "Inbox" button P[menu item]	Click & Drag (1000 ms)
Click on "Inbox" field K[mouse]	50 ms
Mental: check for the unread email (Scan with eyes)	500 ms(depends on the content and user)
Move pointer to New email subject P[menu item]	Click & Drag (1000 ms)
Click on New email subject K[mouse]	50 ms
Mental: read the new email (Scan with eyes)	~ 1000-5000 ms (depends on the content and user)
Analysis	<p>Here the variable factors that need to be considered is the time required by the user to read the email. It can vary based on the content of the email.</p> <p>4750 - 8750 ms, on an average case this is optimal because the receiving and reading a new email application only takes approximately around ~ 6 s</p>

CS5340 Assignment #3

Functionality 3 : Search for an old email

KLM calculation

Action Sequence	Count
Reach for mouse H[mouse]	50 ms
Move pointer to "Search" Field P[menu item]	Click & Drag (1000 ms)
Click on "Search" field K[mouse]	50 ms
Mental: Remember the text in the email	50 ms
Type	(5-20 characters) 10 ms per character
User wait for system response (R)	1000 - 2000 ms (system calculations)
Mental: check for the email from retrieved list (Scan with eyes)	500 ms(depends on the content and user)
If not found, then navigate to the next page of the retrieved list. Move the pointer to "Page 2" If found, Move the pointer to retrieve email item P[menu item]	Click & Drag (1000 ms)
Click on the link field if not found or click on the email subject if found K[mouse]	50 ms
Mental: read the email (Scan with eyes)	~ 1000-5000 ms (depends on the content and user)
Move mouse, navigate, click 'Back'	1050 ms
Analysis	<p>~ 5800 - 10950 ms,</p> <p>This is semi optimal considering the system actions. There are many variable factors in this analysis. The time may largely vary based on the search query that user types in, if it is exact then user will be able to get the retrieve email in the first result page itself or else he might have to navigate to the other search result pages which may vary depending on the number of retrieved items. Second variable is the content of the email, if it is a large email then user might spend more time reading and scanning for the required content in the email.</p>

CS5340 Assignment #3

	<p>On an average case the application only takes approximately around ~ 10 s for this functionality.</p> <p>The system performance of 2s for search query execution can also be thought to be improved and the time can be reduced to 500 - 1000 ms</p>
--	---

Assumptions -

- We assume people check emails on their desktops and on mobile devices often, therefore performance metric analysis should be frequently done on those.
- Day time traffic will be more than night.
- We assume that composing an email may span more than a single login/ single day, therefore the options to save email as a draft is also added.
- People would love to categorize their emails (Eg, personnel, official etc).

Sources Cited -

1. <https://edu.gcfglobal.org/en/email101/common-email-features/1/>
2. <http://www.cs.loyola.edu/~lawrie/CS774/S06/homework/klm.pdf>

Team Details

Zach Rooney	rooney.z
Rajesh Sakhamuru	rajes95
Neenu Jacob	neenu
Deesha Shah	deesha