
Software Requirements Specification

for

CSE-6214 Email Client Fall 2021

Version 1.0 approved

Prepared by

Jeffrey Tagen
Ronald Unz
Sean Douglas
Katie Phillips

Mississippi State University

September 2021

1. Introduction

1.1 Purpose

This document describes the software requirements needed for Version 1.0 of an email application system to perform based on customer expectations. The requirements are outlined for each functionality. This document will cover all the interfaces and functionality of the web-based email client and makes the assumption that all interfaces beyond the client are beyond the scope of this document.

1.2 Document Conventions

General topics and points of interest are notated by numbered, boldface headings. The priority of each product feature specified in Section 3 is listed with the description of that feature. Due to the size of this application, no specific priority is given to individual functional requirements.

1.3 Intended Audience and Reading

This SRS is intended to be read by the marketing staff, users, developers, project managers, testers, and documentation writers.

Once the purpose and scope of the product are understood, the marketing staff and users will benefit from starting with Section 2 of this document, which gives an overall description of the product, including an overview of the product features.

Section 3 of this SRS gives a detailed description of each functional requirement, including the priority, user story, an overview of the response sequence, and each individual requirement that must be met in order for that functionality to perform. This section is a good place for developers, project managers, testers, and documentation writers to begin reading. These readers should also be familiar with the nonfunctional requirements described in Section 4 and any other requirements that are described in Section 5.

1.4 Product Scope

The email application to be developed under this SRS will allow a user to be able to electronically communicate with other users of the application by sending text-only emails. The application will allow local login, authenticating against a preconfigured email server, retrieve and display the emails, and allow the user to send new emails. Server setup, configuration, and user management are beyond the scope of this application.

1.5 References

All interfaces will adhere to HTML5 (version 5.0) specification (<https://www.w3.org/TR/2011/WD-html5-20110405/>) as published on April 5, 2011.

2. Overall Description

2.1 Product Perspective

This email application is a web based system. It is created with the intent of being used inter-departmentally by businesses, but it will be accessible to anyone with internet access.

2.2 Product Functions

- A user can access emails by logging into his/her account using a unique username and password.
- A user can securely logout of his/her account.
- A user can compose, edit, and send emails to other users.
- A user can forward emails to other users.

2.3 User Classes and Characteristics

All users of this email application will be of the same class. These users will be able to send and receive emails, accessing the full functionality of the application. Administration and user management is outside the scope of this application.

2.4 Operating Environment

The operating environment can be split into two areas, client and server.

The server will be a hosted environment on the public internet. It will include a web server configured with the latest PHP 8 runtime environment (8.0.9 / July 29, 2021 as of the writing of this document) with the IMAP extensions enabled. The server is expected to have outbound email capabilities.

The client is expected to be the latest Chrome build on any operating system with JavaScript enabled.

2.5 Design and Implementation Constraints

The app is expected to be able to talk to connect to an IMAP service (IETF RFC5301) through the php-imap library.

3. System Features

3.1 User Login

3.1.1 Description and Priority

User inputs username and password into two separate fields and either hits enter or clicks on a submission button for submission to access his/her emails.

Priority: High

3.1.2 User Story

I want to access my emails as an authorized user so that my messages inbox with my messages are displayed.

3.1.3 Task Breakdown

- Create front-end user HTML login page containing inputs for username and password with a submit button
- Develop code for link between login page and login/password reference
- Develop the functionality to redirect to inbox page after successful login
- Develop code for prompting the user with an wrong username/password
- Write the functionality for a redirect back to the homepage after wrong username/password entry

3.1.4 Stimulus/Response Sequence

The user shall enter their username into a username field. The field should mirror the user's input as it is being typed. The user shall enter their password into the password field. The field shall mirror the user's input with asterisks to hide the entry from unwanted eyes. The user then hits enter or clicks on a submission button for submission of the fields to be checked against a database of username and password. The system either gives access to the user to his/her emails and displays their current "inbox" or the system notifies the user of an incorrect username or password, without specifying which.

3.1.5 Functional Requirements

REQ-1.1: Username - user shall be able to use and input a unique username into the username field.

REQ-1.2: Password - user shall be able to use and input a unique password that is tied to the username into the password field.

REQ-1.3: Submission button - a button that is user clickable for submission of the username and password for check against the email service

REQ-1.4: Database query and authentication- The login page will query the email service to validate the email/password combination is correct.

REQ-1.5: If login is successful, pull all of the user's messages and display inbox

REQ-1.6: If login is unsuccessful, display an error message stating that username or password was incorrect and the user returned to the login page.

3.2 Inbox

3.2.1 Description and Priority

This page will display the sender's name, message title, message receipt date. The number of messages displayed is fixed at 50 per page. This page will indicate if the messages are read or unread using some kind of marking for each. The method for marking is TBD, but could be as simple as a symbol or bold typed font. This page will contain a search function to search through email messages, titles, or senders. There shall be a functional button for the user to access the email editor on this page.

Priority: High

3.2.2 User Story

I want to see a listing of my emails sorted in chronological order on date received, new e-mails near the top. I want to be able to easily distinguish which emails are read and unread. I want to be able to access the email editor from this page.

3.2.3 Task Breakdown

- Create front-end user HTML interface for inbox with: page for displaying N (N integer number, TBD) number of emails in a list showing date sent, sender, and title of email
- Develop button to redirect to email editor
- Write a mechanism for logging out
- Develop code for hyperlinking to new page with entire email message

- Create HTML interface for displaying email messages: displays sender, date sent, and message
- Develop functionality that for adjusting unread/read email messages format on the inbox
- Develop functionality for searching through emails
- Create method for sorting email by date and sender

3.2.4 Stimulus Response Sequence

The users shall be brought to this page after a successful login. The user can identify which emails are unread using a marking system. Users can click on an email of their choosing and it will open a new page displaying the message in its entirety. The user can forward messages from the full messaging pane/window. The user can search through the emails using a search bar located on the inbox screen by entering text and hitting enter. This will refresh the inbox so that it contains only messages that satisfy the search result.

3.2.5 Functional Requirements

REQ-2.1 - Users are brought to this screen from the login page after a successful login.

REQ-2.2 - Display emails in a list format containing date received, recipient name, and title of email, with news emails at the top.

REQ-2.3 - Users can click on email he/she would like to read and it will be opened in new page, or displayed in a separate messaging pane

REQ-2.4 - Read emails are automatically marked.

REQ-2.5 - N number integer number of messages titles should be displayed, N is TBD.

REQ-2.6 - Read and unread emails should be clearly distinguishable.

REQ-2.7 - Users can forward emails to other users.

REQ-2.8 - A search function should be embedded in the page that will filter and display successfully searched emails in a list identical to those in the original inbox.

3.3 Email Editor

3.3.1 Description and Priority

Users have a large text field to add and manipulate text to send as their message. In addition, there is a second text field for the user to enter the email address of the recipient. A third text box will be available to add an optional message subject.

Priority: High

3.3.2 User Story

I want to draft an email and send it to a specific email address.

3.3.3 Task Breakdown

- Create front-end user HTML interface for Email editor: text box for editing text, text box for editing receiver, and text box for editing title
- Create code for sending the email to the correct user over SMTP, function should work with the press of “send” button
- Create functionality to check a receiver has been added to entered into the proper text box

3.3.4 Stimulus/Response Sequence

The user shall be able to enter text of up to 1MB into a message field. The user shall be able to enter the email address of a single user. The user shall be able to enter an email subject/title information into a text field if desired. When the user has completed their message and a recipient is entered, e-mail can be sent to the recipient over the Simple Mail Transfer Protocol (SMTP) protocol by pressing the “send” button. The user will be prompted with a message if the recipient address is blank.

3.3.5 Functional Requirements

REQ-3.1: Recipient - user shall be able to enter single email address to send

REQ-3.2: Subject - user shall be able to enter an email subject/title

REQ-3.3: Message - user shall be able to enter a message into a large text box for drafting

REQ-3.4: Send - The message from the messaging field, with the subject, is sent to the recipient over SMTP protocol.

REQ-3.5: If the recipient field is empty, the user shall be prompted that it needs to be

entered

3.4 Reply and Forward

3.4.1 Description and Priority

Reply and forward are special cases of drafting email messages based on messages already received into the Inbox. Buttons must be available in the Inbox to support corresponding special instances of the Email Editor. Reply messages will include automated pre-population of the Recipient, Subject, and Message fields with corresponding elements from the preceding message(s) as well as the automated inclusion of a prepended "RE:" in the subject field to indicate that the message is a Reply. For Forward messages, the Forward messages will include automated pre-population of the Recipient, Subject, and Message fields with corresponding elements from the preceding message(s) as well as the automated inclusion of a prepended "FW:" in the subject field to indicate that the message is a Forward.

Priority: High

3.4.2 User Story

I want to reply to or forward emails that I have already received. I want these emails to include the texts of the previous emails in the thread or chain. I want to be able to edit these emails before sending, and specify additional and/or other recipient email addresses.

3.4.3 Task Breakdown

For Replies:

- Create mechanism within the email displaying page for replying that redirects the user to the editor, button click
- Create mechanism that copies the original message and pastes it to the messaging text box
- Develop the function that automatically adds the original sender to the receiver text box

For Forwarding

- Create mechanism within the email displaying page for forwarding that redirects the user to the editor without adding the original sender to the recipient, button click

3.4.4 Stimulus/Response Sequence

Aside from the special conditions noted in Paragraph 3.4.1 above, the user experience will be identical to that described in Section 3.3.

3.4.5 Functional Requirements

REQ-4.1: Reply button - user shall be able to apply this special case of the Email Editor onto a selected message in the Inbox or from within an open message.

REQ-4.2: Forward button - user shall be able to apply this special case of the Email Editor onto a selected message in the Inbox or from within an open message.

3.5 User Logout

3.5.1 Description and Priority

A user should be able to logout of their account easily, quickly, and securely.

Priority: High

3.5.2 User Story

I want to be able to logout of my account easily, quickly, and securely without fear of unauthorized users accessing my emails after I do so.

3.5.3 Task Breakdown

- Create mechanism that disconnects the user from the email messaging service

3.5.4 Stimulus/Response Sequence

User clicks "Logout" button offered in Inbox. This results in all server-side sensitive information, including cached emails and credentials being erased.

3.5.5 Functional Requirements

REQ-5.1: Logout button - all server-side sensitive information, including cached emails and credentials being erased.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

REQ-6.1: All pages should load and process in under 5 seconds given a 1Mbps internet connection. An exception is granted in the event of a slow response from the IMAP or SMTP server.

REQ-6.2: Buttons should visibility respond to clicks from an external cursor device (mouse, trackpad, etc.)

REQ-6.3: Text input areas should respond and accept input from keyboard and number pad.

4.2 Safety Requirements

1. This email app/service is not guaranteed to be compliant with the Americans with Disabilities Act.
2. No function is currently provided to allow deletion of emails. This will result in ongoing expansion of storage requirements.
3. If a network connection is required for functionality, users should not depend on the system to support emergency communications unless appropriate redundancies are built into the network.
4. Parental controls and/or guidance are recommended for any children or minors using the system to prevent any influence by bad actors.

4.3 Security Requirements

REQ-7.1 - Users shall have their own unique individual username and password.

REQ-7.2 - Session data, including authentication information and cached emails will be erased upon logout.

4.4 Software Quality Attributes

Correctness - the app will perform the calculations indicated and expected without incomplete data.

Reliability - given support by underlying layers, the software will perform the same for every request

Efficiency - the system will minimize resource utilization and strive for scalability

Usability - The system will require minimal training and be intuitive for new users

Maintainability - The system will be designed in such a way as to have clear code and documentation for future revisions

Reusability - The software will be modularized and clearly documented in such a way to make use of content in the future.

5. Other Requirements

No additional requirements outside the items above have been identified so far.

Appendix A: Glossary

ERD - Entity-Relationship Diagram

HTML - Hyper-Text Markup Language

IMAP - Internet Message Access Protocol

Mbps - Mega-bits per second

PHP - PHP Hypertext Preprocessor

POP - Post Office Protocol

SMTP - Simple Mail Transfer Protocol

SRS - Software Requirements Specification

Appendix B: Analysis Models

Due to the extreme simplicity of this application, including the lack of local storage beyond session identifiers and cached data, no analysis models are provided. The application is merely a display layer that sits between an IMAP server and the user. There is no data dictionary, ERD, or relationship diagrams because there is no persistent local data.

Appendix C: To Be Determined List

REQ-1.7 - Security for use login

REQ-2.5 - Number of messages displayed

REQ-8.1 - Software Quality Attributes (as required)

REQ-9.1 - Other Requirements (as required)

Appendix B: Analysis Models