Software Requirements Specification

for

Email Client Project

<u>Prepared by</u>
CS 6214 - Lab Group 13
William Giddens
Joe Howard
Trey O'Neal
Chad Whitney

09/04/2020

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to properly document and detail the process to be employed by the team while creating the requested email software. The software will be designed and tested to meet user stories and specifications. The software will be built using the Agile Software Development approach. The timeline for completion of the project shall be one semester. Initial release will be version 1.0 and subsequent releases will follow the standard point release model.

1.2 Document Conventions

Hyperlink	Provide extensions to view external sources
Bold	Indicates required materials or specifications as defined in the requirements.
Underlined	Indicates a specific task that must be completed prior to beginning the next step(s).

1.3 Intended Audience and Reading

This project is a prototype for the email client system and it is restricted to the CS 6214 Lab environment. This project will be implemented under the guidance of college professors and is for project purposes only. This project is useful for the Software Engineering Lab team as well as to the Computer Science department as a whole.

2. Product Scope

This project aims to provide an applicable solution for handling the storage and retrieval of emails. The use of advanced object-oriented techniques will provide a foundation for further developers to create and deploy advanced versions of the tool. The foundation provided by the development of this project will be a step toward creating additional enhancements and features in subsequent releases; with each focusing on one specialized part of the email tool. The scope of the project is to meet the features requested from the user community in the initial project request and allow for two small incremental changes to the software.

2.1 References

- ❖ GitHub VCS code repository
- ❖ GoogleDocs team documentation
- ❖ Jamboard for Collaborating on user stories.

2.2 Project Team Responsibilities

The team has set out to build strong working relationships that will allow this project to be completed accurately and on time. We have made the following team Responsibility and implementation decisions.

William Giddens will serve as our team lead and will be responsible for running overall collaboration of the team. William will serve to keep the meeting records.txt file up to date for meetings.

Meetings should be limited to no more than 1 hour unless it involves direct collaboration on product features. The team documents are contained in Google Docs and GitHub.

Team communication has been decided to be over email, GroupMe, GoogleDocs, & GitHub.

Make every effort to ensure that all team members feel comfortable about each others' contributions; be open about grievances. It is a good idea to be collegial and professional in your communications and avoid conflicts. However, you want to be •

If for whatever reason a team members contribution falls short the other team members agree to do extra work for the subsequent deliverable(s).

Plan for disaster scenarios: often team members depend on each other's individual contributions. We have set firm internal deadlines (well ahead of the actual deadline) for contributing individual parts, so that if a team member fails to deliver, the others will have time to take over and complete the missing objectives.

General policy: If in doubt, communicate! Redundancy is OK. Every time we send an email related to the project we copy the email to all team members.

3. Overall Description

3.1 Product Perspective

The software product being created will be a web based email platform that will allow users to login and view their emails. Emails can be created and sent with or without attachments. A user will also be able to search email by subject and then forward an email to another email address.

3.2 Product Functions

Product Function 1 - Authenticated Logon
A user should be able to access and view their email messages with their unique account.

Product Function 2 – Compose, Edit, Send, and Receive Email
User should be able to send and receive emails from other users.

Product Function 2 - Forward Email
A user should be able to forward emails to other users.

Product Function 2 - Search Email
A user should be able to search their received emails.

Product Function 2 – Attach Files to Email
A user should be able to send attachments (image, files etc.) along with an email.

3.3 User Classes and Characteristics

This product will only include a standard email user class.

3.4 Operating Environment

The operating environment on this project will be a Docker container based on the Alpine operating system. This will be hosted on a TBD host for Docker containers. It will use a standalone MYSQL database from a database cluster hosted by Digital Ocean.

3.5 Design and Implementation Constraints

- This software is to be a web application.
- It will be supported by a MYSQL cluster with separate management and the model will not be able to be migrated.
- The firewall will be implemented after development is complete and production begins.
- Software maintenance ends once customer acceptance is complete and software moves to production.

4. System Features

4.1 System Feature 1 - Authenticated Logon

A user should be able to login to the web application for emails with their unique account.
- This function has a **HIGH PRIORITY**
- **Functional Requirements**
  - User should use email address as username
  - User should use password to authenticate login
    - If username and/or password is incorrect it should notify user of failed login
    - Successful Login should open a view of all emails.

4.2 System Feature 2 - View Emails

A user should be able to access and view their email messages with their unique account.
- This function has a **HIGH PRIORITY**
- **Functional Requirements**
  - User should be able to view
    - From
    - To
    - Subject
    - Body
    - Attachment
  - User should be able to view new emails
  - User should be able to view already viewed emails
  - User Should be able to open an email

4.3 System Feature 3 - Compose, Edit, Send, and Receive Email

A user should be able to compose, edit and send an email to other users.
- This function has a **HIGH PRIORITY**
- **Functional Requirements**
  - User Should be able to add email address of recipient
  - User should be able to add subject line
  - User should be able to add text to the body
  - User should be able to add address for sending
  - User should be able to click Send Email
    - This should process email send
    - This should verify email address for sending
    - This should process email send
    - This should verify email address for sending

4.4 System Feature 4 - Forward Email

A user should be able to forward emails to other users.
- o This function has a **MEDIUM PRIORITY**
- o **Functional Requirements**
  - ▪ User should be able to click forward email
  - ▪ User should be able to click send to forward email
    - • This should process email send
    - • This should verify email address for sending

4.5 System Feature 4 - Search Email

A user should be able to search emails.
.
- o This function has a **MEDIUM PRIORITY**
- o **Functional Requirements**
  - ▪ User should be able to enter text in the search bar and press the search button.
    - • Once a search is initiated the resulting emails will be filtered based on the search terms.

4.6 System Feature 5 - Send Email with File Attachment

A user should be able to send attachments (image, files etc.) along with an email
.
- o This function has a **HIGH PRIORITY**
- o **Functional Requirements**
  - ▪ User should be able to add attachment for sending
  - ▪ When pressing the "Attachment" button the user is presented with a file explorer window to chose a file for attaching.
  - ▪ The attachment process will validate the file is under 1MB in size.
    - • This is the only current restriction.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

There exist no unusual requirements or performance needs for use of this software.

5.2 Safety Requirements

The use of this software poses no safety or compliance concerns.

5.3 Security Requirements

A firewall should be installed on the server and standard password maintenance and security practices should be undertaken.

5.4 Software Quality Attributes

Ongoing use of this software requires the continued subscription to Digital Ocean and Nylas as service providers and that all required components remain operational.

6. Other Requirements

Nylasis a cloud based service that deals with api email delivery. The service manages various types of email accounts for applications.

DigitalOcean is an American cloud infrastructure provider that provides developers cloud services that help to deploy and scale applications that run simultaneously on multiple computers. Database interface specifications are here.

Appendix A: Glossary

- ❖ **To:** The "To" header field contains line is for the primary recipient.

- ❖ **From:** The From header field contains the messages author. The authors email address is always included in the From field.

- ❖ **Body message:** The body of an email is the sum of the message contents. It can include text, images, links and media including attachments.

- ❖ **Header:** The header precedes the content (the body) of the message and comprises fields that contain information about the sender and receiver, subject and date fields.
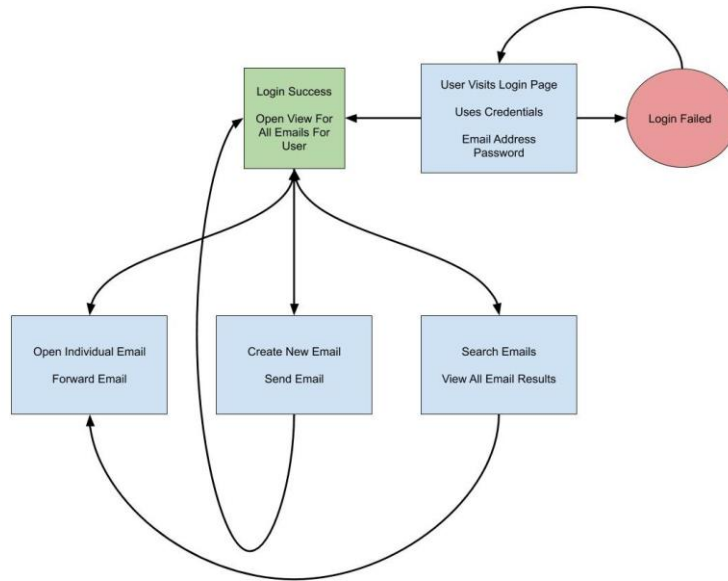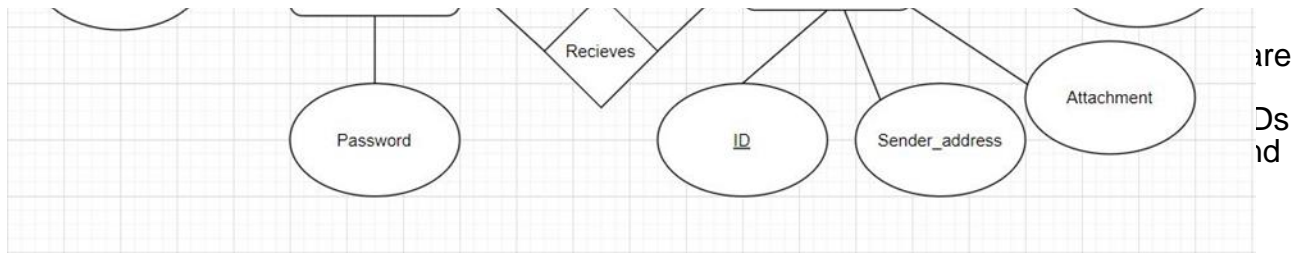
Appendix B: Analysis Models

Figure 3.1 – Simple flowchart of the functionality of the email software

Appendix C: To Be Determined List

6.1.1.1 Provider Emails
Digital Ocean

6.1.1.2 Hardware
VPS Server with 1GB of RAM and 1 Virtual CPU with 25GB of SSD Hard Drive

6.1.1.3 Operating System
Alpine

6.1.1.4 Webserver
Nginx

6.1.1.5 Connectivity
300Mbps available bandwidth (Up and Down)
1 TB of transfer

6.1.1.6 Version Control System (VCS)
Github

6.1.1.7 Security
SSL
UFW Firewall