**Software Requirements Specification**

**for**

Email Client Web Application

**Version 1.0 approved**

**Prepared by Karan, Anjal, Alonzo**

**Mississippi State University**

**September 08, 2021**

# **Introduction**

## **Purpose**

The document presents a detailed description of an Email Client that will have a number of functionalities based on the customer’s expectations. It explains the purpose of the project, the interfaces used, system requirements, and the constraints under which the project is based.

## **Document Conventions**

Simply followed the SRS template provided by the TA. Default font as of now but, maybe update it to Times New Roman 12pts or Arial.

## **Intended Audience and Reading**

This document is intended for requirements engineer, customer, student developer and project leader. This document is especially designated for student developers to input the overall concept/overview which will be proposed to Mr. Rayhan, the TA for approval.

## **Product Scope**

This project is about an Email client that allows a user to login to his account, access his emails, compose, edit and send emails to others, search and forward emails, and send attachments with the email. The developers will be assigning certain task to each individual, similar to this documentation.

* User should be able to login to their account to access emails
  + The purpose is to prevent unauthorized personal from gaining access to private and confidential information
  + Will speed up communication between employees and their superiors, and interconnect with different departments
* User should be able to compose, edit, and send an email to other users
  + User can type email address of receipt (if known) along with an email title to send the email seamlessly
* User should be able to search and forward emails to other users
  + This allows corporations to easily search and pass refurbished emails down a chain to save time and not lose the context of the email
* User should be able to send attachments (images, files, maybe video) along with their email
  + Media and files can be easily transferred between individuals so as long as they are within the upload size

## **References**

No references used so far, except the provided SRS template.

# **Overall Description**

## **Product Perspective**

The email web application stores the following information.

* Emails:
  + It includes all the incoming and outgoing emails to and from the email client.
* Attachments:
  + It includes any files or zipped folders attached with the email.
* Users:
  + It includes the name, email, and various personal information regarding a particular user.

## **Product Functions**

* User should be able to login to their account to access emails
  + There needs to be a login page or somewhere for the user to enter their credentials to access their email and email history
* User should be able to compose, edit, and send an email to other users
  + The following elements will be required (or available) to compose an email: receiver email address, email title, textbox
  + User should have access to a textbox to draft their email before sending
  + There also needs to be a timestamp for when the email was sent
* User should be able to search and forward emails to other users
  + There should be an inbox available for emails to be sent to
  + The inbox should be searchable by title, date, or email address sent from
  + There also should be a Sent-Inbox for all emails sent to others, along with searching capabilities (title, date, email address sent to)
* User should be able to send attachments (images, files, maybe video) along with their email
  + Should be a clear button to attach media to emails
  + Will need to have a data capacity

## **User Classes and Characteristics**

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## **Operating Environment**

The software will operate in the following environment:

* Client/Server System
* Operating system: Windows
* Platform: Python/Flask
* Database: sql+ (not sure)

## **Design and Implementation Constraints**

* Who stores the emails? (Corporations or Developers)
  + For security concerns, corporations will most likely have their own email servers to store their emails, or at the very least host their email servers from an email agency
* What file sizes does the software handle/allow?
  + Emails might only be up to a certain size, the same can definitely be said for attachments
* Maintenance / Version Control?
  + Will the developers be responsible for it after the software has been finished or will the client take over all responsibilities afterwards
* Coding Language Requirements or others
  + Any specifications or standards they want us to adopt for them to integrate it into their profile
* Implementation to other already existing interfaces
  + Do they want the software to integrate with a specific application that they are already using

# **System Features**

## Login Page

4.1.1 Description and Priority

The email application requires an authentication page to access the email. This feature has high priority because it is the foundation of the email app. It provides security and multiple user support.

4.1.2 Stimulus/Response Sequences

* Input username
* Input password
* Signup for new account
* Login button and Signup button

4.1.3 Functional Requirements

REQ-1: Display error if username or password is incorrect

REQ-2: Display error if the user does not exist

## **System Feature 2 (and so on)**

# **Other Nonfunctional Requirements**

## **Performance Requirements**

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

Performance Requirements?

* Reasonable load time for the email client to launch
* Emails should be sent and load at a reasonable time-frame
* Inability for emails to simply disappear or not send at all without a notification of some sort to the sender
* Emails should syncronize with real-world time

## **Safety Requirements**

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## **Security Requirements**

The email application uses stored usernames and passwords so passwords must be hashed before they are stored. If a token such as from Google Authentication is to be used, vendors must authorize tokens only when the user is using the application.

## **Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

# **Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*

*Karan - 1.1, 1.4, 2.2, 2.5, 4.1, 4.4*

*Alonzo - 1.2, 1.5, 2.3, 3.1, 4.2*

*Angel - 1.3, 2.1, 2.4, 3.2, 4.3,*