TEST PLAN FOR

GROUP 9 EMAIL CLIENT

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
1.0	10/23/2020	Entire team	Initial input for each section

1 Introduction

Brief introduction of the test strategies, process, workflow and methodologies used for the project

This is the test plan for Group 9's email client. The test strategy on the project is ensuring that all functionalities (registering, logging in, viewing emails, composing, editing, sending, searching, forwarding, and sending attachments) of the app run smoothly.

1.1 Scope

Need to have 10-15 tests

1.1.1 In-Scope

- Launch the email client
- Go to the registration page and input and submit registration info
- Go to the login page, input, and submit email/password pair that was registered on previous page; confirm login was successful
- Select "sync mail" button and confirm emails have populated the page
- Go to "Compose Message" in the navbar and input "To" information, "Subject" information, input "Content" information, and edit content using "Summernote" text editor
- Use the "Send" button to finalize and send the email; confirm that the Email has been sent
- Use "search" to find an email to forward, select email, and use "forward" to forward email; confirm that email has been forwarded
- Go to "Compose Message" in the navbar and use "Attach File"
- Select file from the file explorer and submit
- Send email with file attachment; confirm email was sent with attachment

1.1.2 Out-of-Scope

- anything not specifically listed in "in-scope" criteria
- additional stakeholder requests, needs, desires, etc.

1.2 Quality Objective

- Ensure the Application Under Test conforms to functional and nonfunctional requirements
- Ensure the AUT meets the quality specifications defined by the client
- Bugs/issues are identified and fixed before go live
- Perform incremental tests each sprint
- Ensure bugs are fixed/mitigated before the next sprint

1.3 Roles and Responsibilities

Name	Net ID	GitHub username	Role
Jonathan Storey	jks263	jon-kane	Group Lead, Tester
Terryuntae Griffin	tmg361	crankiestflame2	Group Member, Programmer
Nicholas Hester	nh785	rojemo888	Group Member, Test Analyst
Brett Hutto	bh1379	BrettHutto	Group Member, Programmer

2 Test Methodology

2.1 Overview

We have adopted an agile paradigm. As the project is being developed in four sprints with each sprint having a particular set of deliverables, it is necessary to perform incremental testing towards the completion of each sprint. This allows us to ensure that bugs in the system are fixed or mitigated before the next sprint.

2.2 Test Levels

We are performing three testing levels: Unit Testing, Integration Testing, and System Testing. Each component that is introduced into the AUT is tested to confirm it fulfills the functionalities it is meant to accomplish (Unit Testing). Before new code is merged with the master, it is confirmed that the additional code does not interfere with previously developed code (Integration Testing). Towards the end of each sprint, the entire system is tested to confirm it satisfies deliverables for that particular sprint (System Testing).

2.3 Bug Triage

Below is a list for each known bug in the AUT. For each bug we provide the following:

- The resolution that we'll be pursued for the bug.
- A priority is assigned to the bug and a schedule given as to when the bug is to be fixed.

Current known bugs in system:

1. HTML messages from some received emails are not displayed correctly (top of UI is cut off).

    a. Resolution: Messages will be placed in a “box” to prevent interfering with UI elements.
    b. Priority: Low; does not interfere with reading messages or navigating the AUT. Bug will be fixed by the end of sprint 4.
2. On the account page, user emails are not sorted by date.
    a. Resolution: Store date from emails as a datetime object. Sort emails by date.
    b. Priority: Low; does not interfere with user ability to search and find emails. Bug will be fixed by the end of sprint 4.
3. Searching emails uses exact word/phrase matching.
    a. Resolution: Use partial word matching without capitalization needed for email subject.
    b. Priority: Medium; search feature works but it’s not particularly useful. Bug will be fixed by the end of sprint 4.
4. Problem with matching file paths to download attachments locally on Windows system.
    a. Resolution: Identify appropriate method for setting file path.
    b. Priority: Medium; it seems a Windows system may be incapable of downloading attachments. Bug will be fixed by the end of sprint 4.

2.4 Test Completeness

The following are the criteria by which testing will be deemed complete:

- All manual & automated test cases executed.
- All open bugs are fixed or will be fixed in the next release.
- 100% requirements coverage for current sprint.
- All severe/showstopper defects are fixed.

3 Test Deliverables

- **Test Plan :** a document which contains the plan for all the testing activities to be done to deliver a quality product
- **Test Strategy :** A document which captures the approach on how we go about testing the product and achieve the goals
- **Test Cases :** Scripts: Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions and actual results.
- **Bug Reports :** convey detailed info to developers about bugs.
- **Test Status Reports :** to track the testing status

- **Test Closure Report :** gives a detailed analysis of the bugs found, bugs removed and discrepancies found in the software.

4 Resource & Environment Needs

4.1 Testing Tools

- Requirements tracking tool -- Github
- Bug Tracking tool -- Github
- Automation Tools -- Travis CI

4.2 Test Environment

Minimum hardware requirements that will be used to test the Application.

- 64 bit processor
- 4 GB RAM

The following software is required in addition to client-specific software:

- Windows 10, macOS, or Linux
- A web browser
- Python 3.6

The following python libraries via requirements.txt:
- flask
- flask_sqlalchemy
- flask_bcrypt
- flask_login
- flask_wtf
- wtforms
- email_validator

5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test