**Test plan for**

**dogPile Email Client**

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| V0.1 | 10/21/21 | Morgan Reece | Initial Draft |
| V0.2 | 10/21/21 | Dusty Piper | Updated Draft – Sect 3&4 |
| V1.0 | 10/22/21 | Morgan Reece | Minor update – Sect 3 |

1 Introduction 2

1.1 Scope 2

1.1.1 In-Scope 2

1.1.2 Out-of-Scope 2

1.2 Quality Objective 2

1.3 Roles and Responsibilities 2

2 Test Methodology 3

2.1 Overview 3

2.2 Test Levels 3

2.3 Bug Triage 3

2.4 Suspension Criteria and Resumption Requirements 4

2.5 Test Completeness 4

3 Test Deliverables 4

4 Resource & Environment Needs 4

4.1 Testing Tools 4

4.2 Test Environment 5

5 Terms/Acronyms 5

# Introduction

The dogPile Email Client will be tested using the interative method. Testing will be conducted following user stories and functionality requirement specifications. Results of the testing will be documented in line with the environemnt and procedures.

## Scope

### In-Scope

The scope of the testing of the dogPile Email Client will be limited to specified functionality within the SRS document. These features are as follows:

1. creation of user account for client access
2. sending and receiving emails through securely integrating with network-based email service provider
   1. compose & edit email body text
3. user authentication/secure connection to network-based email services
4. message management
   1. forwarding
   2. deleting
5. sending and receiving/opening attachments
6. search messages

### Out-of-Scope

Out-Of-Scope defines the features, functional or non-functional requirements of the software that **will NOT be** tested. The functionality and non-functional requirements not listed in the Scope section above will not be tested. A few of the items out of scope are:

1. *Security*
2. *Performance*
3. *Load*
4. *Capacity*
5. *Code Coverage*
6. *Stability*

## Quality Objective

Some objectives of testing of our dogPile Email Client project are:

* Ensure the Application Under Test conforms to functional and non-functional requirements
* Ensure the AUT meets the quality specifications defined by the client
* Bugs/issues are identified and fixed before go live

## Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

* QA Analyst – individual responsible for test execution and result documentation
* Test Manager – individual who will coordinate and direct the testing activities
* Configuration Manager – individual who will specific and confirm the configuration of the environment/system under test
* Developers – individuals who will contribute to the creation of the application to align with the design specification
* Installation Team – individuals who will be responsible for deploying and verifying the application has been deployed in alignment with the installation/deployment guidelines

Amongst others.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Net ID** | **GitHub username** | **Role** |
| Dusty Piper | dp1662 | dp1662msstate | Developer, Conf Mgr & Installation |
| Morgan Reece | mlr687 | morgan-at-MSU | Developer, QA Analyst & Test Mgr |
|  |  |  |  |
|  |  |  |  |

# Test Methodology

## Overview

The iterative testing methodology will be used to exercise the testing of the dogPile Email Client. This methodology was chosen because of the type of process that was used in the development of the application. The functional requirements were defined at the beginning of the project, but they were implemented incremenatally. The application/web service was funtioning very early in the project, but with limited functionality. As more features were implemented, the more test mechanisms were added.

## Test Levels

The dogPile Email Client will be testing in following manner:

1. Unit testing
2. Integration testing
3. Application Under Test

## Bug Triage

The goal of the triage is to

* To define the type of resolution for each bug
* To prioritize bugs and determine a schedule for all “To Be Fixed Bugs’.

Severity of bugs/issues will be scored from 1(low)-10(high).

## Suspension Criteria and Resumption Requirements

Suspension criteria define the criteria to be used to suspend all or part of the testing procedure while Resumption criteria determine when testing can resume after it has been suspended.

Testing will be suspended a level 10 issue is identified. This usually presents itself as the application is unable to run or compile. Testing will resume once the level 10 issue has been resolved.

## Test Completeness

Testing will be deemed complete when the folowing criteria is met:

* 100% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release
* CI build reports will be reviewed and bugs/issues addressed

# Test Deliverables

Upon completion of testing, the following documents will be added to the project collateral. The inclusion of these documents will contribute to the understanding of the testing methodology and the depth and breadth of the testing that has been conducted.

|  |
| --- |
| * Test Plan: Plan containing testing actions and datasets to be used to verify product functionality. * Test Cases: Functional test cases to test application routing and access rights, data scenarios for creation, validation and rejection of application datasets. * Bug Reports: Document containing any known bugs that are pending resolution, research, or are documented as unresolvable. * Test Status Report: Report containing various testing cases and the present status. |

# Resource & Environment Needs

## Testing Tools

* Requirements Tracking Tool: **Github**
* Bug Tracking Tool: **Github**
* Automation Tools: **Travis CI**

## Test Environment

Following **software’s** are required in addition to client-specific software.

1. Windows 8 and above; Mac OS X and above
2. Web Browser
3. Python 3.9+
4. Python Libraries
5. Pipreqs
6. Pytest

Python libraries required for the project can be installed with pipreqs defined in requirements.txt

# 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 |
| SMTP | Simple Mail Transfer Protocol |
| IMAP | Internet Message Access Protocol |