

TEST PLAN FOR STAR MAIL

EMAIL CLIENT PROJECT
TANNER HESS, CONNOR DOUGLAS, CLARENCE BARLOW

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
0.1	10/06/2021	Connor Douglas	Team members added a sign in function along with viewing emails
1.0	10/30/2021	Full Team	Added composing and sending emails.

1. INTRODUCTION	2
1.1. SCOPE.....	2
1.1.1. <i>In-Scope</i>	2
1.1.2. <i>Out-of-Scope</i>	2
1.2. QUALITY OBJECTIVE.....	2
1.3. ROLES AND RESPONSIBILITIES	3
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

1. Introduction

Test strategies for this project include compiling the code and ensuring every feature works. If a feature within the application is not working the team's members analyze the code. When the bug is found the team will resolve the bug. Team members research the different ways to implement a function to resolve the issues within the project.

1.1. Scope

1.1.1. In-Scope

The features that will be tested in this scope are all the features documented in the user stories. Features tested will consist of the following login, viewing and accessing emails, compose emails, sending and viewing attachments, sending emails, and the logout function. The features that our project use rely directly on IMAP Protocols. Our team will spend time testing the applications to ensure its functionality. We will also continue to test the application when adding new features so they do not interfere with the functionality of existing features.

1.1.2. Out-of-Scope

Features that are out of scope and will not be tested are the server load times regarding the IMAP protocol.

1.2. Quality Objective

- Each feature is implemented correctly
- The UI (User Interface) easy to learn, use, and understand
- 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

1.3. Roles and Responsibilities

Name	Net ID	GitHub username	Role
Connor Douglas	cbd211	con505doug	Developer, Test Manager
Tanner Hess	tmh648	TanHess	Developer, Test Manager
Clarence Barlow	cb3407	barlowec	Developer, QA Analyst

2. Test Methodology

2.1. Overview

The reason for adopting extreme programming for this project is because it works best for the particular project we are working on. We research different functions to implement and check for functionality. If the addition of the new code is not functioning correctly we step through the code to find the bug. When bug is discovered we debug the code, step through it again, compile it; we do this to ensure it is running successfully.

2.2. Test Levels

Unit and acceptance testing are the types of testing that will be executed on the Application Under Test (AUT). The Testing Levels primarily depends on the scope of the project, time and budget constraints.

2.3. 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”.

2.4. 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. While conducting Unit Testing if there is a success we move on to find bugs. When it reports a failure, the test will be suspended until we analyze and fix the bugs that caused the issue.

2.5. Test Completeness

Criteria that will deem complete testing:

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed
- Any other bugs will be fixed in next release

3. Test Deliverables

The Test Artifact that will be delivered during the different phases of the testing lifecycle is the Test Plan and the Test Cases .

4. Resource & Environment Needs

4.1. Testing Tools

Tools we are using to test our project:

- Requirements Tracking Tool
- Bug Tracking Tool

4.2. Test Environment

Hardware

- Minimum of 1 GB of Ram
- Minimum 500 MB of disk space
- 1.6 GHz of faster processor
- Computer (i.e. Laptop, Desktop)

Software

- Windows 8 and above
- Mac OS X El Capitan and above
- Office 2013 and above
- Visual Studio Code
- Python 3 or above
- Any Web browser

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
UI	User Interface