Rowan Student Email Filtration Application

Test Plan

Shayne Colomy

CS 07321 Software Engineering I, Section 1

Professor Myers

April 19th, 2020

**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 3

**2** **TEST METHODOLOGY 4**

2.1 Overview 3

2.2 Test Levels 3

2.3 Test Completeness 4

**3** **TEST DELIVERABLES 5**

3.1 Test Cases 5

*3.1.1 Log In Test 5*

*3.1.2 Home Screen Test 5*

*3.1.3 Settings Screen Test 6*

*3.1.4 Notification Test 7*

**4** **RESOURCE & ENVIRONMENT NEEDS 8**

4.1 Test Environment 8

**5** **TERMS/ACRONYMS 8**

# **Introduction**

For this project, a number of testing strategies will be employed. There will be Unit testing of individual features as they are developed and completed. System testing of several features interacting with each other will then take place. Throughout the entire development process we will also be implementing Ad Hoc testing on the fly as features are being developed. This is to catch any problems early and iron out as many as possible.

## 1.1 **Scope**

### **1.1.1 In Scope**

All features that were ~~developed~~  created by the development team are considered in scope for testing and will be tested thoroughly. Features within our scope include logging into the application with accurate credentials, fetching relevant information from an email inbox based on ~~pre-set and~~ user configurable parameters, as well as managing all information in a neatly designed interface. The tests will insure that the user experience is fully upheld to the standard set by the project requirements.

### **1.1.2 Out of Scope**

Certain features of the application will be used without any form of testing taking place. These are standard features that the app will use to drive the features that have been created. One such feature is the JavaMail API which will be used to fetch a user's email inbox ~~and relevant information.~~ This API has been thoroughly tested by its developers and will not require testing ~~outside of successful usage of the API~~ for use in this application. ~~Another~~ One other feature that will go untested is the IMAP protocol employed by the JavaMail API for the process of fetching the email inbox. IMAP is an industry standard web protocol and is more than reliable for the purposes of this app.

## 1.2 **Quality Objectives**

* The email filtration app will meet the specifications laid out in the requirements document.
* All functional aspects of the email filtration app will be tested thoroughly upon completion.

## 1.3 **Roles and Responsibilities**

* System Owner (Jeffery Bonfield) - The system owner will review the testing plans to oversee ample meeting of the quality requirements. Any additional testing may be requested from the System Owner.
* Development Team (Ed Callihan, Nick Setaro, Kyle Takach, Collin Wisser) - The development team will be responsible for the Ad Hoc testing portions of the application. They will test features as they are developed prior to the features reaching a completed status.
* Testers (Ed Callihan, Shayne Colomy, Chris Hicks, Nick Setaro, Kyle Takach, Collin Wisser) - The testers will comprise the entire project team including the Development Team, Product Owner, and Scrum Master. This is to optimize the testing process and provide as much manual interaction as possible over the course of testing. The job of the testers is to administer tests, document the results, and establish what features of the application do not meet the requirements laid out. ~~This is~~ Testing will be done this way to ensure that all aspects of the application in scope are accounted for and tested prior to production.

# 2 **Test Methodology**

## 2.1 **Overview**

The email filtration app will be tested in numerous ways both during development and upon completion of ~~several~~ app features. ~~This~~ Testing will be done by all members of the project team to maximize the amount of test cases that can be deployed.

## 2.2 **Test Levels**

For the Rowan Email Filtration app, 3 different levels of testing will be employed.

* Ad Hoc - As features of the app are developed, testing will be done by the developers to establish boundary cases and steer the feature into a completed state.
* Unit Testing - Once features of the app are completed, they will be tested individually through the front end user interface to establish any bugs that will need to be repaired. This will determine the completed nature of all features.
* System Testing - System testing will then take place once enough features are present to do a full runthrough of the app as well as it’s functions. By this stage, all individual features should be fully tested and this stage would only highlight the interaction of those features to find any errors in place.

## 2.3 **Test Completeness**

Testing will be defined as complete once the following circumstances are met:

* All test cases have been executed and documented.
* All known bugs are fixed or will be fixed by the next release.

3. Test Deliverables

## 3.1 Test Cases

## 3.1.1 Log In Test

Test ID Step # Input Procedure Expected Result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 |  | Press login button with empty fields | Invalid credentials message |
|  | 2 | Invalid email | Press login button with invalid email | Invalid email message |
|  | 3 | Invalid password | Press login button with incorrect password | Invalid credentials message |
|  | 4 | Correct email and password | Press login button with correct credentials in fields | Successful login and redirection to home screen. |

## 3.1.2 Home Screen Test

Test ID Step # Input Procedure Expected Result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 1 |  | Scroll vertically on home screen | All home screen entities appear in a neat manner |
|  | 2 |  | Tap dropdown menus | Buckets on the home screen expand to show notifications when clicked on. |
|  | 3 |  | Click add bucket button | A new screen appears allowing the user to enter information for a new bucket |
|  | 4 | Contact info for new bucket | Enter contact information and click enter | Bucket is created and visible on the home screen |
|  | 5 |  | Click the settings button | The screen redirects to the settings page |
|  | 6 |  | Click the refresh button | All the content on the homescreen will update if any new information is available |

## 3.1.3 Settings Screen Test

Test ID Step # Input Procedure Expected Result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 1 |  | Click the email list button | A new screen appears showing all the current emails |
|  | 2 |  | Click the notification lifetime setting | A dropdown menu appears showing the various length options before a notification will expire |
|  | 3 |  | Snooze Settings | A dropdown will appear showing the various snooze time settings |
|  | 4 |  | Turn off received notifications | Disable push notification functionality |
|  | 5 |  | Turn off daily notifications | Hides the daily notifications feature from the home page |
|  | 6 |  | Switch the color themes | App will turn from light theme to dark and vice versa |

## 3.1.4 Notification Tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 1 |  | Create a bucket for test email | Bucket will be created and shall receive absolute notifications upon population |
|  | 2 | Test Email | Send an email to user email address from email address added in Step 1 | A notification will populate in the user created bucket & a push notification will arrive on the users device |
|  | 3 |  | Add keyword to the bucket from Step 1 | Keywords will now be active for this bucket. Push notifications will only be triggered by emails containing the keyword |
|  | 4 | Test email with keyword | Send an email to user email address containing the keyword added in Step 3 from the email address entered in Step 1 | A notification will populate in the user created bucket & a push notification will arrive on the users device |

# 4 **Resource & Environment Needs**

## 4.1 **Test Environment**

* The test environment will ~~be~~ take place on popular Android devices running ~~the Android APK version 29~~ Android version 10 (API Version 29).

# 5 **Terms/Acronyms**

|  |  |
| --- | --- |
| **TERM/ACRONYM** | **DEFINITION** |
| API | Application Program Interface |