Rowan Student Email Filtration Application

Validation Plan (Revision)

Nick Setaro

CS 07321 Software Engineering I, Section 1

Professor Myers

04/26/2020

Table of Contents

[1. Introduction 3](#_Toc38788783)

[2. Overview 3](#_Toc38788784)

[2.1 Scope 3](#_Toc38788785)

[2.2 Depth 4](#_Toc38788786)

[2.3 Testing Methodology 4](#_Toc38788787)

[2.3.1 Unit Testing 4](#_Toc38788788)

[2.3.2 System Testing 5](#_Toc38788789)

[2.3.3 Ad Hoc Testing 5](#_Toc38788790)

[2.3.4 Installation Qualification (IQ) 5](#_Toc38788791)

[2.3.5 Operational Qualification (OQ) 5](#_Toc38788792)

[2.3.6 Performance Qualification (PQ) 7](#_Toc38788793)

[3. Testing Rationale 7](#_Toc38788794)

[3.1 Email address connection: 7](#_Toc38788795)

[3.2 Notifications not pushed to home screen: 7](#_Toc38788796)

[3.3 The user cannot download the application: 8](#_Toc38788797)

[4. Traceability 8](#_Toc38788798)

[5. Roles 10](#_Toc38788799)

[5.1 System Owner (Jeff Bonfield) 10](#_Toc38788800)

[5.2 Development Team (Ed Callihan, Nick Setaro, Kyle Takach, Collin Wisser) 11](#_Toc38788801)

[5.3 Testers (Ed Callihan, Shayne Colomy, Chris Hicks, Nick Setaro, Kyle Takach, Collin Wisser) 11](#_Toc38788802)

[5.4 Product Owner (Shayne Colomy) 11](#_Toc38788803)

[5.5 Scrum Master (Chris Hicks) 11](#_Toc38788804)

[References 12](#_Toc38788805)

# Introduction

The validation plan will outline the strategy used to test the Rowan Student Email Filtration Application to ensure that it meets the user requirements. This plan will detail the different testing methods used to provide a comprehensive test of the overall system and any necessary connected systems in the project scope. It will also justify how thorough each testing strategy will be relative to each component that is tested by assessing the risk of that component. The risk of each component or specified task will be assessed by considering the likelihood and severity of it failing, and the mitigation strategies to prevent it from failing.

The validation plan will ensure that each user requirement is tested by the testing strategy. The plan will reference each requirement to the appropriate test script which utilizes the appropriate strategy. The test scripts will be thoroughly organized and presented in the testing document. Each team member is associated with a different portion of the validation and testing strategy to make sure that the final application will meet the user and system owner requirements. These roles will be specified later in the validation plan.

# Overview

## 2.1 Scope

The scope for validation will be the Rowan Student Email Filtration Application. This includes the front end, or user interface, and the back-end java application. The front-end is a mobile application for viewing important notifications based on student emails. The back-end will parse through all of the user’s emails and relay the emails deemed as important to the front-end mobile application view.

The system will integrate directly with Gmail through the Internet Message Access Protocol (IMAP) using the JavaMail API. The JavaMail API has been thoroughly tested by the developers and therefore will not be tested as part of the validation plan. IMAP will also not be tested as it is the internet standard for messaging and has been thoroughly tested by outside parties. The system will integrate with the admins through a web server that will enter standard admin settings into the application. We will be unable to test this component due to the lack of availability of admins due to the current health crisis.

## Depth

The depth of testing will vary depending on the risk associated with each component, or function, failing. The risk will be assessed by considering the likelihood and severity of each component failing. This will result in a risk score from 1 to 5 where 1 is low severity and low likelihood, 3 is medium severity and medium likelihood and 5 is high severity and high likelihood. Components with a risk score of 4 or higher will be focused for testing to ensure that the application will not fail with high severity.

The following functions have the highest severity and/or likelihood. The risk score can be found in parentheses next to the item:

1. The emails address cannot be connected to. (5)
2. The notifications are not pushed to the home screen. (4)
3. The user is not able to download the application. (4)

These specific functions are vital to the performance of the application and will be more thoroughly tested than other functions.

## Testing Methodology

Scripted testing will be utilized in the testing strategy. Scripted testing will play a large part in unit testing. Test scripts will be created by the test team to validate each component and function individually. The test scripts will test individual features and methods within the code.

The following sections describe the testing methodologies that will be used for validation of the application.

### 2.3.1 Unit Testing

Unit testing will be done to validate the various components and functions in an isolated environment. A test script will be run on the individual components and functions. This will be validated by receiving the expected results from the test script for that component. Unit testing will not include any other connected components and will simply test individual functions. For example, unit testing will be performed on functions such as adding a bucket, adding a keyword, pressing the snooze button on a notification, etc.

### 2.3.2 System Testing

System testing will test the entire system as a whole upon completion. This will include all aspects of the system including connecting components such as JavaMail and local flutter notifications. This will be validated by ensuring that all components are connected correctly and pass the correct information. It will also include all of the unit tests performed to validate that all components receive the correct information, process the information correctly, and send that information to the correct components. System testing will be done by loading the application onto a developers’ Android devices and utilizing every feature implemented in the application. The results will be validated by the entire system’s ability to meet user requirements.

### 2.3.3 Ad Hoc Testing

Ad hoc testing was used throughout the entire software development cycle. It was used to ensure that components worked in the most basic use case and performed basic functions. This was accomplished informally by the developers while coding the different functions and components. This form of testing was not documented as it is not a formal testing strategy, however, it was utilized heavily by the development team. Errors resulting from ad hoc testing were resolved by using debugging software built into the developers Integrated Development Environments (IDEs). These errors are usually fatal to the function or component and are resolved prior to being processor by formal testing strategy.

### 2.3.4 Installation Qualification (IQ)

Installation qualification testing will be done to validate the correct components needed for installation. This includes correct or minimum versions for dependencies as well as operating systems. The following conditions must be met to adhere to the installation qualification:

* The device must run an Android API of 29 or higher.
* The device must be capable of running the JavaMail API version 1.5.5 or higher.
* The device must be capable of running the Jsoup API version 1.11.1 or higher.

### 2.3.5 Operational Qualification (OQ)

Operation qualification testing will use a user configuration that satisfies the conditions set forth in the installation qualifications. This configuration will test all functionality of the application from the user perspective. Test scripts will be run from the correct configuration and the results will be well documented. These will ensure that a user running the correct operating systems and dependencies will experience the correct functionality. The following important use cases will be tested:

|  |  |
| --- | --- |
| User Story | Validation |
| The user will be able to open the application by pressing the application logo on their mobile device’s home screen.  The user will be able to login to their student Gmail account using their username and password.  The user will be able to add a bucket to their list of buckets.  The user will be able to name their bucket and enter an associated email address.  The user has an important notification that they would like to always see. The user hits the pin notification button  The user has seen a notification on the home screen of his device but would like to be reminded of the notification later. The user presses the snooze button.  The user would like to change the name of a bucket or the email associated to a bucket. The user presses the edit bucket button and enters a new email or name. | The login page for the application will then appear.  Upon pressing the login button with correct credentials, the application will move the user to the homepage.  A menu will appear from the bottom of the screen which will contain a name and email field.  The home screen will show the new bucket with the user entered email address and name.  The notification is pinned to the home screen and is always shown as the first item behind previously pinned notifications.  The notification is pushed to the home screen again at after a user defined period of time.  The name and the email address of the bucket are changed on the home screen and the bucket shows notifications from the new email address if entered. |

### 

### 2.3.6 Performance Qualification (PQ)

Performance qualification validation will be completed by using actual user email addresses and checking real emails. The following conditions must be met to validate the performance of the application:

|  |  |
| --- | --- |
| user story | validation |
| The user wants to begin using the Rowan Email Filtration Application.  The user wants to login to the application to begin using it.  The user would like to view their important notifications.  The user would like to be notified via a push notification when a new important email is received. | The user will be able to search for the application on the Google Play Store and download it.  The user will enter their username and password and be taken to the home screen if they enter the correct credentials.  The home screen will show admin defined default buckets which will list important notifications from the user’s emails.  The application will send a push notification to the Android device when a new notification arrives. This will include the subject, date and body. |

# Testing Rationale

## Email address connection:

Email address connection has a high severity and a high likelihood. All functionality of the application requires access to user emails to function correctly. This aspect is critical to the application. Google email addresses will not allow third party applications to connect to a Google account without switching the “allow less secure apps” setting to be switched to on. Without this setting Gmail will deny access to the emails even with the correct email and password. This will be mitigated by utilizing OAuth2. OAuth2 is a secure authorization service used by Google that allows applications to access protected information in a way that is safer than just using a password.

## Notifications not pushed to home screen:

Notifications not being pushed to the home screen has a high severity and a medium likelihood. The user would only be able to see notifications when the application opens on their phone which would greatly inconvenience them. It could also lead to the user missing important notifications which would undercut the purpose of the application. To mitigate this issue, we will use local notifications through flutter. These notifications work seamlessly with flutter applications to deliver reliable and accurate notifications to users.

## The user cannot download the application:

The user’s ability to download the application has a high severity and a medium likelihood. If the user cannot download the application onto their device then they will not be able to use any of the features. This would be severely detrimental to the application. This is of medium likelihood due to users who have older versions of the Android API on their mobile devices. We will mitigate this by using functions that work with more versions of the API to reduce the minimum needed version for the user. This will make the application available to a wider range of users without requiring them to update their mobile device.

# Traceability

The following shows the traceability matrix for the validation plan:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Requirement | Card Number | Design Document Reference | Test Script Reference | Notes |
| The user would like to be notified via push notification when an email is received from a specified email that may or may not contain a specified keyword. | 1.1, 1.5 | 1.1, 1.5 | 3.1.4 |  |
| The user would like to press a push notification for an important email and be brought to that email. | 1.4 | 1.2 | 3.1.4 |  |
| The user wants to be reminded of a notification later, so they press the snooze button and the notification appears again after a set amount of time. | 1.3 | 1.3 | 3.1.4 |  |
| The user would like to set a severity level for a bucket to indicate how important the notifications will be. | 2 | 2 |  | Changed to simply allowing the user to change how notifications are received by the application. Will not be bucket specific. |
| The user would like to create a new bucket by pressing an add bucket button on the home screen and entering an email address and any desired keywords. | 3.1 | 3.1 | 3.1.2 |  |
| The user would like to change the email address or keywords associated with a bucket, so they press the edit button on a bucket and enter the new information. | 3.2, 3.3, 3.4, 3.5, 3.6 | 3.2, 3.3, 3.4, 3.5, 3.6 | 3.1.2 | Adding, deleting and editing keywords and email addresses in buckets will all be done in the same menu. |
| The user no longer wants to receive emails from a certain bucket. The user can then press the delete button on the bucket to delete it. | 3.7 | 3.7 | 3.1.2 |  |
| As a user, I would like to be able to change the settings of the application from a settings menu. I would like to be able to press a button on the home screen to access this menu. | 4.4 | 4.4 | 3.1.3 |  |
| As a user, I would like the settings menu to have an option to restore the buckets to their default values. | 4.10 | 4.10 | 3.1.3 |  |
| The user may not want to receive push notifications from certain buckets. They can go to the settings page and toggle which buckets push notifications. | 5.2 | 5.2 | 3.1.3 |  |
| The user would like to see their important notifications on the home page. | 6.1 | 6.1 | 3.1.2 |  |
| The user has a notification that is very important and that they would like to always be at the top. The user can press the pin notification button to pin it to the top. | 6.2 | 6.2 | 3.1.2 |  |
| The user would like to see notifications that they have recently received. The application will have a list of daily notifications that have been received in the past 24 hours right below pinned notifications. | 6.3 | 6.3 | 3.1.2 |  |
| The user would like to know which notifications have been snoozed for a later time. The application will have a list of the snoozed notifications below the daily notifications. | 6.4 | 6.4 | 3.1.2 |  |
| The user would like to press the application icon on their home screen to redirect them to the login screen. | 8.1 | 8.1 | 3.1.1 |  |
| The user would like to enter their rowan network username and password into the appropriate fields on the login screen. | 8.2 | 8.2 | 3.1.1 |  |
| The user would like to press the login button and be redirected to the home page if the correct credentials were entered. | 8.3 | 8.3 | 3.1.1 |  |
| The user would like to login using the secure Rowan CAS system. | 8.5 | 8.5 |  | This function will not be completed due to the current health crisis. |
| The user would like to download the application. They would like to be able to go to the Google Play Store and search for the app then download it. | 7.2 | 7.2 | Tested via Google Play on IQ device. |  |

# Roles

## 5.1 System Owner (Jeff Bonfield)

The system owner shall review the validation process and ensure that the validation requirements meet quality and usability that is required. They shall also review the results of testing the system and ensure that they agree with the documented results. The owner may relay any changes that must be made to the validation and testing plan to the product owner, who will in turn relay that to the development team to rectify the issue.

## 5.2 Development Team (Ed Callihan, Nick Setaro, Kyle Takach, Collin Wisser)

The development team will be responsible for the making sure that the application is ready to receive the test scripts. This will include proper set up of the testing environment and the correct version of the application. The development team will assist the testers to complete and document the running of all test scripts.

## 5.3 Testers (Ed Callihan, Shayne Colomy, Chris Hicks, Nick Setaro, Kyle Takach, Collin Wisser)

The testers will be responsible for running the formal test scripts and analyzing the results of these scripts. They will also be responsible for ensuring that the results are well documented and related to each of the user requirements. The testers will use this information to determine what areas of the application have not met user requirements and must be fixed. This will ensure that all requirements set forth in the requirements document have been fully met and work as expected by the user.

## 5.4 Product Owner (Shayne Colomy)

The product owner will be responsible for relaying the test results to the project system owner. The results will be clearly laid out to show the system owner how the results have correlated to the user requirements. This will include which user requirements have been met by the current version of the application as well as which could not be met. The product owner will ensure that the system owner knows the risk involved with each component failing and how that will affect how the end user experiences the application.

## 5.5 Scrum Master (Chris Hicks)

The scrum master will be responsible for the team’s adherence to validation through agile scrum methods. This includes proper documentation of test results and distribution of testing responsibility. In this case the scrum master will also serve as a tester in running test scripts and documenting results.

# References

[1] J. V. Emden and L. M. Becker, Writing for engineers. London: Palgrave, 2018.

[2] Shayne Colomy, Rowan Student Email Filtration Application Test Plan, 19 April 2020.

[3] Kyle Takach, Rowan Notification App Requirements Document, 10 March 2020.