Operation RM
Northern Arizona University
Flagstaff, AZ

William Rogers (Team Lead): wpr29@nau.edu
Isaac Faulkner: lwf2@nau.edu
Andrew Milizia: am5275@nau.edu
Nick Henderson: nsh67@nau.edu

Software Testing Plan

April 5, 2024

Operation RM

Client:

General Dynamics Mission Systems

Mentor:

Tayyaba Shaheen

Team Members:

William Rogers

Isaac Faulkner

Andrew Milizia

Nicholas Henderson

Version:

1.2

Overview:

This document outlines the comprehensive testing plan, including unit testing, integration testing, and usability testing, to ensure the reliability and functionality of Operation RM's application.

Table of Contents

1 Introduction

In today's world, the need for secure, efficient, and tactical communication in the defense, public safety and intelligence communities can not be understated. General Dynamics Mission Systems (GDMS) has tasked Operation RM with developing a solution to solve the problem of inefficient communication with a radio modem on a mobile device.

Our solution is an Android application that can provide two-way communication with a radio modem, in addition to being able to select a specific waveform to use for transmitting and receiving. Furthermore, our solution is able to display composed and received emails containing images and videos in an embedded container. Incoming emails are displayed as a notification that opens the email when clicked upon. The application has integration with the default Android camera capabilities, and photos are able to be quickly shared with the application and added to composed emails. The application has color-blind considerations, and can be configured to a dark mode to avoid eye strain.

To ensure that our solution meets the requirements and expectations set forth by GDMS, a comprehensive testing plan was developed. This testing plan will serve to describe the software testing that will be performed. There are various kinds of testing that will be performed, such as unit testing, integration testing, and usability testing.

Unit tests for the application will be developed to ensure that functionality is not inadvertently altered when modifying the application. In addition, unit testing will provide a method for automatically testing edge cases for inputs to the functions and logic of the application.

Integration testing will be developed to ensure that the various components of the application work together correctly. Our application contains an interface between the front-end Java code and the back-end C code via JNI. It is crucial to ensure this interface functions properly as all calls to the radio device are made through this

interface. In addition UDP packets are sent between the mobile application and the radio simulator. This necessitates the need for testing on both the mobile application and the radio simulator to ensure proper functionality.

Usability testing is important for validating the non-functional requirements and ensuring that the application meets the end user needs. The usability tests will be conducted primarily through feedback from our client GDMS. Due to the sensitive nature of our application's end user, Operation RM can not be directly involved in direct end-user feedback and iteration. However, our client can provide feedback based on their usage and experience with the end user.

The nature of our application necessitates prioritization on the integration and usability tests. Due to the application being heavily integration based, this is the most important area of concern. While unit tests are important, because users primarily interact with the system indirectly via a GUI, direct user input is only used on the composed screen. Therefore testing edge cases will not be as important. Finally, due to the sensitive nature of the use case and end user of the product, usability testing will not be able to be conducted directly. Therefore, while usability testing is arguably the most important for validating the solution, this will be primarily done by the client.

2 Unit Testing

In order to provide validation after making changes to the software, unit tests are used. Unit testing provides an automated way of testing edge cases as well as validating the output of functions. This type of testing can provide valuable insight into the robustness of the software as well as ensuring that any changes made to the solution do not affect already written functionality.

Due to limited user input, unit testing will only need to be conducted for the composition methods, although extended coverage will be implemented to account for any unexpected impacts. Unit testing will be conducted using JUnit5. JUnit5 is a testing framework for unit tests on the Java Virtual Machine (JVM). Android Studio has built-in compatibility with JUnit5. In addition Mockito will be used to simulate objects. This is important for testing Android, since Contexts are not normally included in unit tests.

Primary focus will be on creating unit tests for the ComposeFragment. To begin, the subjectText field will be evaluated to ensure that the subject text is equal to the amount of characters that the user inputs, or if the text is more than 64 characters, then the first 64 characters. For unit tests, this will consist of manually calling the associated method and setting the text. Then, the contentText field will be evaluated to ensure that if the text is set to the amount of characters the user inputs, or if the text is more than 3000 characters, then the first 3000 characters. This will be evaluated by manually calling the method and setting the text. Both of these tests will test both the standard text size (less than the maximum), and a number greater than the maximum to ensure it works correctly.

The next unit test will consist of verifying that an email attachment is less than 100 MB. The requirements dictate that the solution must be able to support emails with attachments up to at least 20 MB, and the limit is set to 100 MB to ensure that the emails can be sent in a reasonable amount of time. This limit can be changed in the configuration file. For unit tests, the method checking the file size will be called and either return success or failure based on the size of the attachment the user selects.

The method will be tested by attempting to add two files, one 20 MB and the other 150 MB to ensure that the 20 MB file is able to be added, while the 150 MB file returns with an error.

The next unit test will be conducted on the MainActivity. The current time is set and displayed on the status bar of the app. To ensure this time is correct within reason, the function to get the current time of the app will be called and checked against the current time of the phone. If the two times are within 5 seconds of each other then the test will pass. Another unit test will be called that uses an incorrect time, and asserts that the time of the system is not equal to it.

3 Integration Testing

Integration testing is the method of testing a piece of software for the interaction between each of the project's main components. This is to ensure that data transferred from each piece of the application is successful and does not have any errors. In regards to emails, this is crucial. Integration testing is a crucial form of testing for this project since there are many components that must work together. Therefore, establishing a robust testing plan for how all the components connect is paramount for confirming the application's reliability. To do so, integration testing will be conducted between each component that the Operation RM team has developed and the additional libraries including the library provided by General Dynamics Mission Systems that provides functions that interact with the simulator.

The integration testing strategy revolves around testing the proper communication between significant modules to verify their interactions and data exchanges. The team will focus on the critical integration points, such as those involving connecting to the simulator in both directions of communication. These primary components are listed on the following page.

The primary components include:

- **Main Activity**: Handle email lists and primary back-end
- **Application Fragments**: Display emails and their content
- **List Adapters**: Properly display email lists and allow for email viewing
- **File Handler Module**: File transferring, sending, and receiving
- **C and Java Intermediate Layer**: Communication to the radio modem

The following diagram Figure 1.1 shows the data flow of all the project's components:

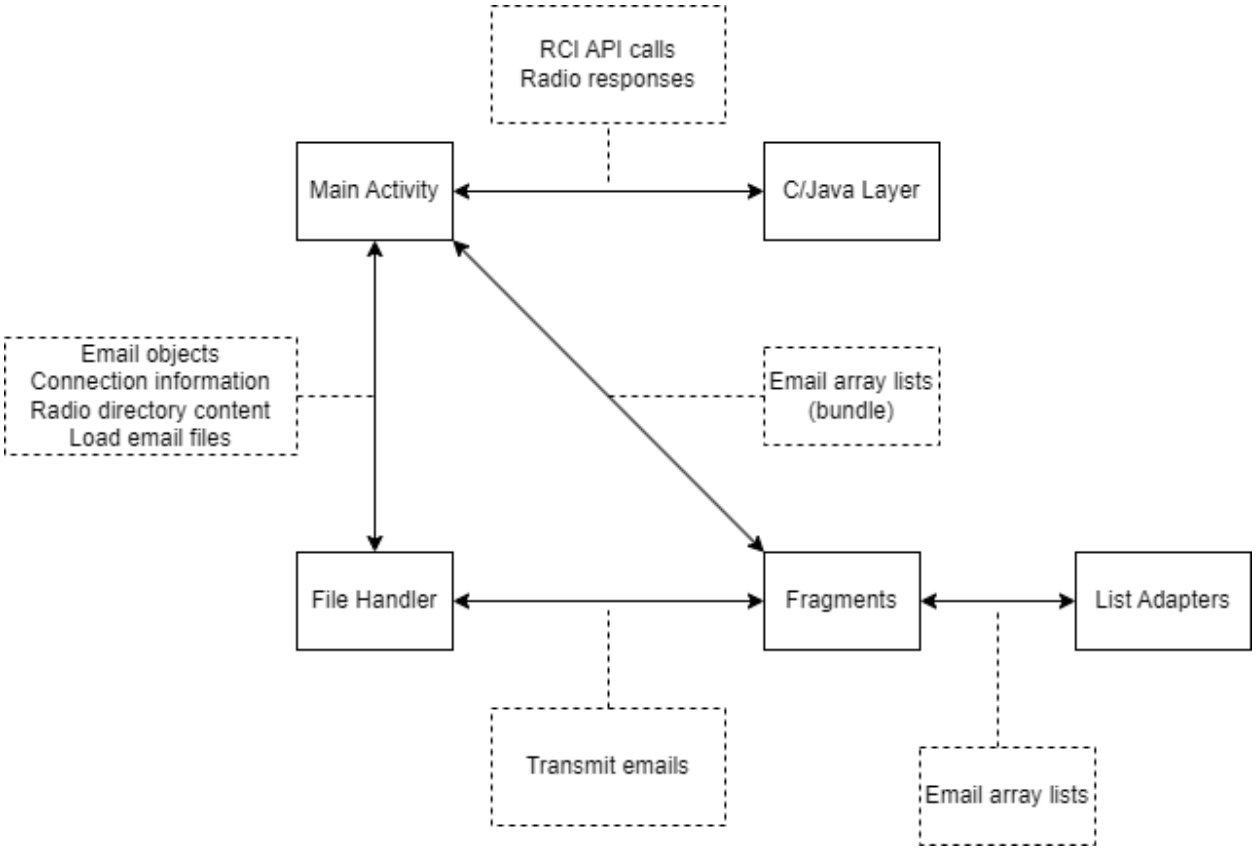

Figure 1.1 Component data flow diagram

All of these components must be able to communicate with each other so the management of email data can be properly sustained along with sending and receiving emails from the radio modem. The plan for testing these components is to use the testing harness integrated in Android Studio along with creating error cases by using the application. Some tests for errors will include:

- **Empty email composition**
- **Incorrect email file formats**
- **Lack of connection to the radio modem simulator**
- **Lack of response from the radio modem simulator**

To execute our integration testing plan the team will utilize various tests and methodologies.This will consist of a combination of automated testing and manual testing procedures. Each integration point will be tested when implemented together in the git lab where each design requirements were added together individually. After this integration the “plumbing” of the application is all set. Allowing for the testing of mock objects such as emails to be sent from and to the simulator.

The testing front and back end the creation of emails with the ability to access photos and documents on the application. Next is to prove the communication pipeline from the mobile application to the simulator again through emails that will be sent from and to the simulator. The hope of this testing is that it provides the conclusion that communication is harmonious between front-end, back-end, and the simulator.

During this testing stage we will also be stress testing the communication between the mobile application and simulator. This will be done by sending a vast amount of emails from the simulator to the phone ensuring that the phone will be able to deal with a vast amount of emails at one time as well as keep desired speed of the application.

Overall, the integration testing approach aims to validate the robustness and reliability of the system, guaranteeing that all components work harmoniously together

to deliver the desired functionality of the application. Through this careful, meticulous testing Operation RM hopes to not only uncover but address any bugs or potential issues that might arise. Through integration testing Operation RM will be able to

4 Usability Testing

To provide the best experience for users to use the application, there must be multiple tests. Usability for the application is extremely important as it is one of the main requirements of the application. Usability testing is about how well the application provides a positive user to experience. With this in mind, testing the Android application to ensure it runs smoothly and provides ease of use for new users is required. The requirements that will guide the usability tests are the reliability and the responsiveness of the application.

There are minimal options for usability tests as the project is under an NDA and has to go through the client for testing. The usability testing will be primarily conducted by the clients and the Operation RM team. The testing will not include a large subject pool, but will still be efficient since the testers will understand what is required for the application. The client has expertise in the project's scope and has a thorough understanding of what is required from the application. There is an engineering team at General Dynamics that the Operation RM team has been coordinating with that will be able to test this application. This is extremely insightful when providing feedback for improvements and potential issues that need to be resolved which will allow the Operation RM team to deliver the best possible product.

One of the major requirements the client desires is for Android application layout to resemble their current web based application. Testing cases for the Android application in this regard will have to be tested. To test for this, the Android application layout and the client provided screenshots of their web based application will be compared. This will ensure that both platforms have the same overall layout.

Throughout testing, confirming that all screens have a nearly identical appearance to the GDMS web based application will be conducted. The testing for the functionality of each screen and user interface component is as follows:

- **Navigation**: Testing that the navigation buttons lead to the correct destinations

- **Composition**: The “Compose” screen will be tested for all the required information and is able to send an email to the outbox.
- **Inbox**: The “Inbox” screen will be tested to make sure emails can be received with all the correct information in adherence to the web based application’s email format.
- **Outbox**: The “Outbox” screen will be tested to ensure the email has the same correct information and format along with the ability to transmit the selected email(s).
- **Sent**: The “Sent” screen will be tested the same as the previous two screens in regards to email content.
- **Transfer History**: The “Transfer History” screen will be tested and checked to ensure that the complete list of all emails that have been sent or received are there.
- **Settings**: The “Settings” screen will be tested to ensure the “Status” and “Dark Mode” buttons will work properly. Within the status page the testing for the presets button will be tested.
- **Presets**: The “Preset” screen will be tested to make sure the dropdown has to correct information within it and the activate button works properly and sets an active preset. Then we will test the deactivate button making sure it deactivates the current preset.
- **Dark Mode**: Testing dark mode will be conducted by navigating the application and confirming that all screens are correctly changed with dark mode.
- **Status Bar**: While navigating through the application, testing the status bar for displaying the correct connection status, current preset, time, and sending or receiving status icons will be conducted.

This will provide insight on the usability of the application. This is important since it is one of the key requirements that was required from the client. The data and feedback received from this testing procedure will prove that the application has high usability and fulfills all of the client's requirements. This data will assist the Operation RM team in making the application adhere to the appearance requirement which leads to lower training times for this version of the radio modem platform.

To test the responsiveness and reliability of the application there will be tests conducting various daily application operations that could potentially put stress on the application. The test aims to see how responsive the application is under high load and its ability to maintain stability. These will be multiple aspects to test in this regard. The tests for the non functional requirement is as follows:

- **Received Emails**: How quickly the application can recognize received email(s) on the radio simulator and the quantity of emails it can download from it.
- **Device Loading**: How quick the application loads on different mobile devices.

- **Radio Modem Connection**: How quick the application can connect to the radio simulator and how long it can st7ay connected. Also, checking for connectivity while the mobile device is in sleep mode.

With the data received from these tests, it can be compared to the non-functional requirements specified by the client to ensure that they are all met. A primary component of these tests is speed and ensuring that the application does not crash.

Overall the usability testing will highlight if there are any important flaws in the application and where they reside. Most of this testing will be started in the middle of April, 2024 since a majority of the implementation has been completed so the major components of the application can be tested. This will allow for the application to demonstrate how the user experience is and if the application can crash in any operation that will be commonly conducted. This will ensure that the application will be robust upon delivery to the client and can be used without any issues. Usability testing

will provide data that the application adheres to the responsiveness and reliability non-functional requirements along with the usability and dark mode functional requirements.

5 Conclusion

The development of a secure, efficient, and tactical communication solution is imperative for defense, public safety, and intelligence is of high importance in today's world. Operation RM's goal is to address the current inefficiencies in communication and hope to represent a significant step toward satisfying these needs.

Throughout the document, we have outlined Operation RM's approach to software testing, encompassing unit testing, integration testing, and usability testing, Unit testing serves to validate individual components and functions, ensuring its dependability and robustness. Integration testing focuses on verifying the interactions between modules and components, enabling seamless communication and functionality. Usability testing ensures that the application meets the end user needs and provides an intuitive and efficient user experience.

With the adherence to this in depth testing process and leveraging the knowledge that is presented from the testing. The team hopes to deliver a highly functional, error-free, and user friendly software product that meets the requirements and expectations of GDMS.

At the time of developing this document, testing in all three categories has been conducted. The unit tests that have been conducted are the tests specified in the respective section of this document. In regards to integration testing, the lack of connection, lack of response from the radio simulator, and empty email content have all been tested successfully. For usability testing, the application's user interface has been tested at the Engineering Fest which proved successful and this was also tested during the alpha prototype demonstration. Responsiveness has been tested throughout the implementation process and was also demonstrated with the alpha prototype.

In conclusion, the successful development and deployment of this communication solution will undoubtedly enhance the capabilities of defense, public safety, and intelligence personnel, enabling them to communicate effectively and securely in a vast amount of situations. Operation RM status dedicated to delivering that reliable solution to improve the communication of our client at GDMS