University of Maryland Global Campus

Presented By

Reshawna Sampson, Barry Gartrell, Stanley De Jesus,   
Arnold Detoito, Sheena Johnpeter, Ananya Srinivasan

Test report

USPS Informed Delivery Visual Assistance App

Table of Contents

[1. Introduction 3](#_Toc110974594)

[1.1 Purpose 3](#_Toc110974595)

[1.2 Application Overview 3](#_Toc110974596)

[1.3 Testing Scope 4](#_Toc110974597)

[1.4 Intended Audience 7](#_Toc110974598)

[1.5 Technical Stakeholders 7](#_Toc110974599)

[2. Overview of Test Results 8](#_Toc110974600)

[2.1 Test Roles and Responsibilities 8](#_Toc110974601)

[2.2 Overall Assessment and Test Summary 8](#_Toc110974602)

[3. Testing 9](#_Toc110974603)

[3.1. Unit Tests 9](#_Toc110974604)

[3.2. Integration Tests 20](#_Toc110974605)

[4. End-to-end Testing 22](#_Toc110974606)

[4.1. Usability Tests 22](#_Toc110974607)

[4.2. Performance Tests 22](#_Toc110974608)

[5. Test Summary for Vision API Google ML Kit QRcode/Barcode 23](#_Toc110974609)

[5.1 Vision API 23](#_Toc110974610)

[5.2 Google ML Kit QRcode/Barcode 23](#_Toc110974611)

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Description | Approved by |
| 1.0 | 8/9/22 | Initial Document | Reshawna Sampson (PM) |
|  |  |  |  |
|  |  |  |  |

1. Introduction

This report explains the software test process for the USPS Informed Delivery Application designed and developed by Team Arch and Team B. The USPS Informed Delivery App is the first iteration of this software as more users are becoming visually impaired. The application interacts with the user via speech recognition capabilities using a mobile phone. To measure and ensure the overall quality of the developed application for its correctness, completeness, usability and performance, the following testing were performed.

* Functional testing: It checks for the functional requirements of the system.
* Non-functional testing: It checks for requirements such as performance, usability, scalability of the software.
* Regression testing: These checks are performance after upgrade such as adding and removing features and any essential upgrades to ensure changes do not affect existing functionality.
* End to End testing: It tests the application workflow from beginning to end by manually testing different use case scenarios to validate integration functionality and data integrity.

1.1 Purpose

The purpose of testing is to evaluate the system for any potential bugs and if the application meets all stated requirements. The test report provides a summary of test objectives, activities, and test results for the USPS Informed Delivery 1.0 Application. The application is developed based on initial and evolving requirements determined throughout all four milestones. Users should use this application to confirm all features worked as discussed. This document should be used by all stakeholders to understand the application quality and decide whether the application feature meets their expectation for the final release of the application for general users to operate.

1.2 Application Overview

The USPS Informed Delivery Visual Assistance App is developed to help visually impaired people know the mail that will be in their mailbox for that day, as well as be read all emails they received. This mobile application was developed using speech to text recognition technology to read these emails to the user. Through a series of voice commands, the user can navigate through the app. Two commands used to read emails are “Lates digest” to get the USPS informed delivery email read, and “Latest Email” to have all emails read. There is a mic feature that the user needs to use to say this command. Press and holding the mic button will activate the speech o text listening feature that will listen for the user’s voice command. Upon release of the mic button the command will execute. The app is designed to be an email reader as well as a mail reader for the USPS Informed Delivery emails.

1.3 Testing Scope

The scope of this testing is to identify what is to be tested to ensure the quality, performance, usability, and completeness of this application as per the system requirements specification. This report is to highlight all required tests completed in order to evaluate the traits mentioned. The following table lists all the system requirements for the USPS Informed Delivery App.

**Table 1**

*Requirements Matrix*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code** | **Requirement** | **Team Arch**  **is responsible for** | **Team B is responsible for** | **Mandatory or Optional** |
| R.01 | The application shall use the user’s email address to collect information for the user’s Informed Delivery Digest | ☑ | ☑ | Mandatory |
| R.02 | The application shall show a screen that includes the user’s personal information |  | ☑ | Optional |
| R.03 | The application shall have access to the voice command feature |  | ☑ | Mandatory |
| R.04 | The application shall identify the user’s voice |  | ☑ | Mandatory |
| R.05 | The application shall have access to the mic button for requesting the user’s information |  | ☑ | Mandatory |
| R.06 | The application shall listen to the user’s voice when the user selects the mic button |  | ☑ | Mandatory |
| R.07 | The application shall notify the user when a daily digest email arrives |  | ☑ | Optional |
| R.08 | The application shall listen for keywords/phrases to interact with the users |  | ☑ | Mandatory |
| R.09 | The application shall turn keywords/phrases into text in order to identify what actions should be taken |  | ☑ | Mandatory |
| R.10 | The application shall have access to the user’s phone camera | ☑ |  | Mandatory |
| R.11 | The application shall use Google Vision API OCR and logo features for image processing | ☑ |  | Mandatory |
| R.12 | The application shall extract an image from the user’s email | ☑ |  | Mandatory |
| R.13 | The application shall recognize handwritten content | ☑ |  | Mandatory |
| R.14 | Upon recognizing handwritten contents, the application shall extract an image from the user’s email and verbally read the handwritten content out to the user | ☑ | ☑ | Mandatory |
| R.15 | Upon extracting an image of an email using the user’s phone camera, the application shall read the text aloud to the user |  | ☑ | Mandatory |
| R.16 | The application shall detect logos from daily digest emails | ☑ |  | Mandatory |
| R.17 | Upon checking for logos, the application shall provide additional information about the sender to the user | ☑ |  | Mandatory |
| R.18 | The application shall have access to the QR/barcode reader functionality | ☑ |  | Mandatory |
| R.19 | The application shall read QR/barcodes | ☑ |  | Mandatory |
| R.20 | The application shall have a “Play Latest” feature and allow users to locate the latest email |  | ☑ | Mandatory |
| R.21 | Data shall be transmitted and received using JSON format | ☑ |  | Mandatory |
| R.22 | The application shall operate with iOS and Android Operating Systems | ☑ | ☑ | Mandatory |

1.4 Intended Audience

The intended audience of this test report document are the development team, project manager, DevOps engineers, and testers responsible for implementing the USPS Informed Delivery Visual Assistance app. This document serves as a reference for the testing activities used to validate the application. Additionally, other stakeholders, such as the project client, can refer to this document to understand how the application meets the specified software requirements.

1.5 Technical Stakeholders

Each member of Team Arch has taken on two or more roles since the beginning of this project to contribute toward reaching the project’s objective and meeting the expectations of the stakeholders. This document defines stakeholders as individuals with an interest and expertise in this project’s design and development. Since the third milestone, taking on many responsibilities has been much more intense. The USPS Informed Delivery Vision Assistance Mobile Application’s project stakeholders are including in the table below, along with all the responsibilities the development team that Team Arch has played:

**Table 2**

*Team Members*

| **Team Member** | **Role** | **Email** |
| --- | --- | --- |
| Reshawna Sampson | Project Manager | reshawanasampson@gmail.com |
| Stanley De Jesus | Lead Developer | sdejesus@gmail.com |
| Barry Gartrell | Developer | gartrellbarry@gmail.com |
| Ananya Srinivasan | Business Analyst/Developer | aasrin99@gmail.com |
| Sheena Johnpeter | Business Analyst/Tester | jsheena1927@gmail.com |
| Arnold Detoito | Developer/Tester | arnold.detoito@gmail.com |

1. Overview of Test Results

2.1 Test Roles and Responsibilities

The Team Arch roles and responsibilities for testing the application are as follows:

**Table 3**

*Roles and Responsibilities*

|  |  |  |
| --- | --- | --- |
| Team Member Name | Role | Responsibility |
| Reshawna Sampson | Project Manager | Functional/Integration Test Whole Application for demo |
| Stanley De Jesus | Lead Developer | Unit/Function Test Vision |
| Barry Gartrell | Developer | Unit/Function Test USPS Validation WebAPI |
| Ananya Srinivasan | Business Analyst | Unit/Function Test QR/Barcode |
| Arnold Detoito | Developer | Function Test Camera on both iOS/Android |
| Sheena Johnpeter | Business Analyst | Unit/Function Test QR/Barcode |

2.2 Overall Assessment and Test Summary

Team Arch had a total of 19 out of the 39 unit/integration tests generated for the USPS informed delivery application.  The tests had a 100% pass rate. Team Arch implemented the project requirements as stand-alone features that can be added where needed with an understanding of its input and output. We did have two functional requirements that failed due to the text-to-speech feature (Team B requirement) not reading all texts that are queued into the feature for the iOS physical device.  It would run for five seconds before stopping text. These issues could not be troubleshot further due to compatibility issues with hardware (iPhone 7 / 8) that the team had at their disposal. Also, the camera does not work using the simulator.  The application also requires a minimum of iPhone 10.  Observations were made for functional and end-to-end testing and all requirements from the stakeholders were satisfied.  We will discuss these test cases further in the testing section.

1. Testing

3.1. Unit Tests

The following tests were conducted by the testers and developers in segments as each major system requirement was completed.

**Table 4**

*Test Case 1: Digest Email Parsing IMAP Client Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to create an IMAP Client for the user’s email. |
| Test Type | Functional |
| Requirement | The application shall receive mail piece images from the Informed Delivery Daily Digest email in the user’s inbox. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  The user has signed in. |
| Steps | 1. The application creates an IMAP client. 2. The application makes a IMAP command call. |
| Expected Output | The application can receive data from the IMAP Client. |
| Assumption | None |

**Table 5**

*Test Case 2: Digest Email Parsing Email Retrieval Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to receive data from an IMAP client. |
| Test Type | Functional |
| Requirement | The application shall receive mail piece images from the Informed Delivery Daily Digest email in the user’s inbox. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  The user has signed in. |
| Steps | 1. The application opens an IMAP client. 2. The application makes an IMAP Search command. |
| Expected Output | The application has received data from the IMAP client about the emails matching the search criteria. |
| Assumption | None |

**Table 6**

*Test Case 3: Digest Email Parsing Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to parse the data from the IMAP client. |
| Test Type | Functional |
| Requirement | The application shall receive mail piece images from the Informed Delivery Daily Digest email in the user’s inbox. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  The user has signed in. |
| Steps | 1. The application opens an IMAP client. 2. The application makes an IMAP Fetch command. 3. The application receives data from the IMAP client. 4. The application parses the data into a consumable object. |
| Expected Output | The application has created an object based on the data from the IMAP Client. |
| Assumption | The returned data from the IMAP client is a single Informed Delivery Daily Digest Email. |

**Table 7**

*Test Case 4: USPS Address Verification Address String Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to accept an address string for parsing. |
| Test Type | Functional |
| Requirement | The application recognizes text from a mail piece and interprets the information. |
| Req. # | SRS-02 |
| Prerequisite | The user can access the internet. |
| Steps | 1. The application has found an address that it wants to verify. 2. The application checks if the address is in the correct format. |
| Expected Output | The application approves or denies the address for verification. |
| Assumption | None |

**Table 8**

*Test Case 5: USPS Address Verification XML Creation Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to |
| Test Type | Functional |
| Requirement | The application recognizes text from a mail piece and interprets the information. |
| Req. # | SRS-02 |
| Prerequisite | The user can access the internet.  The application has approved the address for verification. |
| Steps | 1. The application parses the address for its components. 2. The application creates an XML using the address’ components. |
| Expected Output | The application has successfully created an XML for the web request. |
| Assumption | None |

**Table 9**

*Test Case 6: USPS Address Verification Web API Connection Test*

|  |  |
| --- | --- |
| Description | This test will ensure that the application is able to connect to the USPS Address Verification Web API. |
| Test Type | Functional |
| Requirement | The application recognizes text from a mail piece and interprets the information. |
| Req. # | SRS-02 |
| Prerequisite | The user can access the internet.  The application has already created an address XML. |
| Steps | 1. The application creates a web request using the address XML. 2. The application calls the web request. |
| Expected Output | The application has connected to the USPS Web API. |
| Assumption | None |

**Table 10**

*Test Case 7: USPS Address Verification Web Request Interpretation Test*

|  |  |
| --- | --- |
| Description | This test will ensure that the application is able to interpret the information returned from the USPS Web API. |
| Test Type | Functional |
| Requirement | The application recognizes text from a mail piece and interprets the information. |
| Req. # | SRS-02 |
| Prerequisite | The user can access the internet.  The application has already created an address XML. |
| Steps | 1. The application creates a web request using the address XML. 2. The application calls the web request 3. The application determines the validity of the Address based on the returned data from the Web Request. |
| Expected Output | The application produces a value for the validity of the address. |
| Assumption | None |

**Table 11**

*Test Case 8: Other Email Parsing IMAP Client Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to create an IMAP Client for the user’s other email. |
| Test Type | Functional |
| Requirement | The application shall receive other email in the user’s inbox. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  The user has signed in. |
| Steps | 1. The application creates an IMAP client. 2. The application makes an IMAP command call. |
| Expected Output | The application can receive data from the IMAP Client. |
| Assumption | None |

**Table 12**

*Test Case 9: Other Email Parsing Email Retrieval Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to receive other email data from an IMAP client. |
| Test Type | Functional |
| Requirement | The application shall receive other email in the user’s inbox. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  The user has signed in. |
| Steps | 1. The application opens an IMAP client. 2. The application makes an IMAP Search command. |
| Expected Output | The application has received data from the IMAP client about the emails matching the search criteria. |
| Assumption | None |

**Table 13**

*Test Case 10: Other Email Parsing Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to parse the other email data from the IMAP client. |
| Test Type | Functional |
| Requirement | The application shall receive other email in the user’s inbox. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  The user has signed in. |
| Steps | 1. The application opens an IMAP client. 2. The application makes an IMAP Fetch command. 3. The application receives data from the IMAP client. 4. The application parses the data into a consumable object. |
| Expected Output | The application has created an object based on the data from the IMAP Client. |
| Assumption | None |

**Table 14**

*Test Case 11: Camera Scan Mail Piece Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application has the ability to scan mail pieces using the user’s camera. |
| Test Type | Functional |
| Requirement | The application shall scan mail pieces using the user’s camera. |
| Req. # | SRS-02 |
| Prerequisite | The user can access the internet.  The user has signed in. |
| Steps | 1. The user opens the application. 2. The user selects the Scan Mail button to open the camera. 3. The user scans a mail piece. 4. The user accepts the scanned mail image. |
| Expected Output | The application read aloud the contents of the scanned piece. |
| Assumption | None |

**Table 15**

*Test Case 12: API Credential Test*

|  |  |
| --- | --- |
| Description | This test will verify that the application has the credentials file and loads its content |
| Test Type | Unit Test |
| Requirement | The application shall connect to the Google Vision API. |
| Req. # | SRS-01 |
| Prerequisite | * The user can access the internet. * Google API credentials must be in the assets folder. |
| Steps | 1. The application opens up the credentials file located under the assets file and loads its content. |
| Expected Output | No error is displayed |
| Assumption | None |

**Table 16**

*Test Case 13: Process mail piece image for text using google vision*

|  |  |
| --- | --- |
| Description | This test will verify that the application can process image for text using google vision |
| Test Type | Functional |
| Requirement | 1. API credentials Test must pass 2. The application must have a mail piece labeled mail.test.01.jpg under the assets folder. |
| Req. # | SRS-01 |
| Prerequisite | * The user can access the internet. * Google API credentials must be in the assets folder. |
| Steps | 1. The application establishes a cloud vision object.  The object constructor will initialize and the file to use to connect to vision 2. The application loads a test mail piece labeled mail.test.01.jpg. 3. The application will convert the image to bytes and encode it into base64 4. The application makes a call to cloud vision to process the image. 5. The application will receive the call response and compare its results. |
| Expected Output | The response will be the same as the implanted object. |
| Assumption | None |

**Table 17**

*Test Case 14: Process mail piece image for logo using google vision*

|  |  |
| --- | --- |
| Description | This test verifies that the application can process image for logos using google vision |
| Test Type | Functional |
| Requirement | 1. API credentials test must pass 2. The application must have a mail piece labeled mail.test.01.jpg under the assets folder. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  Google API credentials must be in the assets folder. |
| Steps | 1. The application establishes a cloud vision object.  The object constructor will initialize and the file to use to connect to vision 2. The application will load a test mail piece labeled mail.test.01.jpg. 3. The application will convert the image to bytes and encode it into base64 4. The application makes a call to cloud vision to process the image. 5. The application will receive the call response and compare its results. |
| Expected Output | The response will be the same as the implanted object information. |
| Assumption | None |

**Table 18**

*Test Case 15: Process illegible mail piece; returns empty object*

|  |  |
| --- | --- |
| Description | This test is to demonstrate what is returned if both text/logo object is illegible or not recognized |
| Test Type | Functional |
| Requirement | 1. API credentials test must pass 2. The application must have a mail piece labeled horusx.jpg under the assets folder. |
| Req. # | SRS-01 |
| Prerequisite | * The user can access the internet. * Google API credentials must be in the assets folder. |
| Steps | 1. The application establishes a cloud vision object.  The object constructor will initialize and the file to use to connect to vision 2. The application will loads a test mail piece labeled horusx.jpg. 3. The application will convert the image to bytes and encode it into base64 4. The application will invoke the cloud vision process method using the encoded image as parameter. 5. The application will receive the object and compare its results. |
| Expected Output | The response will be an mail Object with address/logo lists, each with a length of zero. |
| Assumption | None |

**Table 19**

*Test Case 16: Process a mail piece image through barcode scanner widget using Google ML Kit*

|  |  |
| --- | --- |
| Description | This test verifies that the application can pass an image through a barcode scanner widget using Google ML Kit and determines if the link is valid |
| Test Type | Functional/Unit Testing |
| Requirement | 1. API credentials test must pass 2. The application must have a mail piece labeled mail.test.03.jpg under the assets folder. |
| Req. # | SRS-01 |
| Prerequisite | * The user can access the internet. * Google API credentials must be in the assets folder. |
| Steps | 1. The user opens the application 2. The application opens the credentials file located under the assets file. 3. The application will load a test mail piece labeled mail.test.03.jpg. 4. The application will convert the image to bytes and encode it into base64 5. The application makes a call to Google ML Kit to process the image. 6. The application validates if the link is valid. 7. The application determines if bounding and scaling is necessary. |
| Expected Output | The response will validate links, if valid and (if applicable) match link on mail piece. In addition, it will also determine if bounding and scaling is necessary. |
| Assumption | None |

3.2. Integration Tests

The following tests were conducted to ensure that the integration with the other team was working as designed.

**Table 20**

*Test Case 17: process image for text and logo and consolidate into one object*

|  |  |
| --- | --- |
| Description | This test verifies that the application can process image for logos using google vision |
| Test Type | Integration |
| Requirement | 1. API credentials Test must pass 2. The application must have a mail piece labeled mail.test.01.jpg under the assets folder. |
| Req. # | SRS-01 |
| Prerequisite | The user can access the internet.  Google API credentials must be in the assets folder. |
| Steps | 1. The application establishes a cloud vision object.  The object constructor will initialize and the file to use to connect to vision 2. The application will load a test mail piece labeled mail.test.01.jpg. 3. The application will convert the image to bytes and encode it into base64 4. The application makes a call to cloud vision to process the image using the process method. 5. The application will receive the call response and compare its results. |
| Expected Output | The response will be same to the implanted object information. |
| Assumption | None |

**Table 21**

*Test Case 18: Load Digest Page Test*

|  |  |
| --- | --- |
| Description | This test verifies that the application can access the Digest information and that the created object is consumed by the User Interface. |
| Test Type | Integration |
| Requirement | 1. API credentials Test must pass. |
| Req. # | SRS-01 |
| Prerequisite | * The user can access the internet. * Google API credentials must be in the assets folder. * The user is logged in. |
| Steps | 1. The user prompts the application to open latest digest. 2. The application gets the Digest Email for that date. 3. The application gets the Google API information for the individual mail pieces. 4. The application parses the remaining email information into a single object 5. The application navigates to the *Digest Details* page where the object is then consumed on the User Interface. |
| Expected Output | The application has navigated to a new page an begun consuming the data in the object. |
| Assumption | There is a valid USPS Informed Delivery Email for the chosen date. |

1. End-to-end Testing

End-to-end testing has been executed to verify that the application is working as expected. The front-end process was tested with the back-end process which includes the functionalities such as the image parser, text-to-speech, etc. These have been done using Android emulator, iOS Simulator, physical Android mobile devices and physical iOS mobile devices.

4.1. Usability Tests

The United States Postal Service (USPS), as an agency of the federal government, is required to comply with Section 508 of the Rehabilitation Act of 1973, which requires agencies to give individuals with disabilities access to information that is comparable to the access available to other users of a service.

The application provides the visually impaired users an audio description of the USPS Informed Delivery Daily Digest email details, other email details and camera-taken mail photos, as these emails solely consist of images and text.

The application also provides the visually impaired users an option to utilize voice commands to perform the various functionalities in the application, such as receiving an audio description of the Daily Digest email details, other email details, navigating from email pages to another and retrieving the list of help commands.

The USPS Informed Delivery Visual Assistance application is deployed to the physical Android and iOS mobile devices to perform these various usability tests.

4.2. Performance Tests

Performance testing has been performed on an Android emulator, iOS Simulator, physical Android mobile devices and physical iOS mobile devices

Stress testing is the type of performance testing that the team has used. The application was placed under higher-than-expected requests to measure the efficiency and reliability. One example is the application was given a request/voice command then was given another request while the first request was still processing. There was a slight delay in response from the application, but it was able to handle and perform all the requests successfully.

1. Test Summary for Vision API Google ML Kit QRcode/Barcode

5.1 Vision API

The following tests were conducted using 96 images of mail pieces extracted from USPS Informed Delivery emails over a four-month period. As depicted in the table only some had logos.

**Table 22**

*Vision API Test Summary*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Total tests conducted | 151 | Passed | % | Failed | % |
| OCR | 92 | 84 | 91% | 8 | 9% |
| Logo | 61 | 53 | 86% | 8 | 14% |

5.2 Google ML Kit QRcode/Barcode

Testing for barcode was sparse as receiving any USPS Informed delivery mail pieces with a barcode or Qr code on them. The following numbers were conducted using physical pieces, USPS Informed Delivery png images, and single qr codes.

**Table 23**

*Barcode Success Rate*

|  |  |  |
| --- | --- | --- |
| Total tests conducted | Passed | Failed |
| 100 | 76 | 24 |

Figure depicts the success rate percentage for the different types of mail pieces and QR codes that were scanned.

**Figure 1**

*Barcode Scanner Success Rate*

Chart, pie chart

Description automatically generated