2022

Project Plan

USPS informed delivery app – Visually impaired

**Presented by: Team Arch: Reshawna Sampson, Barry Gartrell, Sheena Johnpeter, Stanley De Jesus, Ananya Srinivasan, Arnold Detoito**

Table of Contents

[1 General Information 4](#_Toc110030246)

[1.1 Background 4](#_Toc110030247)

[1.2 Statement of Need 5](#_Toc110030248)

[1.3 Vision Statement 5](#_Toc110030249)

[1.4 Stakeholders 5](#_Toc110030250)

[1.5 Acronyms and Abbreviations 7](#_Toc110030251)

[2 Project Assignment 7](#_Toc110030252)

[2.1 Project Scope 7](#_Toc110030253)

[2.2 Project Objectives 8](#_Toc110030254)

[2.2.1 Project Requirements 8](#_Toc110030255)

[2.2.2 Project Assumptions 9](#_Toc110030256)

[2.2.3 Product Approval/Acceptance Criteria 10](#_Toc110030257)

[2.2.4 Project Cost 11](#_Toc110030258)

[2.2.5 Change Management 12](#_Toc110030259)

[3 Risk Analysis 13](#_Toc110030260)

[3.1 Risk Framework and Assessment 13](#_Toc110030261)

[3.2 Application Risks, Vulnerabilities and Mitigations 14](#_Toc110030262)

[3.3 Project Risks and Mitigation 16](#_Toc110030263)

[4 Organization of Project 17](#_Toc110030264)

[4.1 Communication 17](#_Toc110030265)

[4.2 Explanation of Methodology 18](#_Toc110030266)

[4.2.1 Scrum Artifacts 18](#_Toc110030267)

[4.3 Project Staffing 18](#_Toc110030268)

[4.4 Roles and Responsibilities 20](#_Toc110030269)

[4.5 Project Tools 21](#_Toc110030270)

[4.6 Project Storage 22](#_Toc110030271)

[4.7 Deliverables 22](#_Toc110030272)

[4.8 Project Schedule 0](#_Toc110030273)

[5 Software Test Plan 0](#_Toc110030274)

[5.1 Purpose 0](#_Toc110030275)

[5.2 Scope 0](#_Toc110030276)

[5.3 Test Items 0](#_Toc110030277)

[5.3.1 To Be Tested 0](#_Toc110030278)

[5.3.2 Not to be Tested 1](#_Toc110030279)

[5.4 Approach 1](#_Toc110030280)

[5.5 Pass/Fail Criteria 1](#_Toc110030281)

[5.6 Defect Severity Levels 1](#_Toc110030282)

[5.7 Risks and Contingencies 2](#_Toc110030283)

[5.8 Test Cases 2](#_Toc110030284)

[6 References 9](#_Toc110030285)

Project Name:

Date: 5/26/2022

Project Manager: Reshawna Sampson

Phase: Project Planning

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Description | Approved By |
| 1.0 | 6/1/2022 | Initial Document | Reshawna Sampson (Project Manager) |
| 1.1 | 6/4/2022 | Added Risk Management Framework table | Reshawna Sampson  (Project Manager) |
| 1.1.2 | 6/13/2022 | Added Milestone 1 feedback | Reshawna Sampson (Project Manager) |
| 1.2 | 6/16/2022 | Added Software test Plan | Reshawna Sampson (Project Manager) |
| 1.3 | 7/29/2022 | Update the tables and images to be APA. Cited in the references. | Reshawna Sampson  (Project Manager) |

# ****General Information****

## ****Background****

United States Postal Service (USPS) is an independent agency of the United States federal government which provides postal services, such as mail delivery, within the United States. In an effort to provide a robust experience for residential consumers, USPS launched the Informed Delivery service, which allows consumers to preview letter-size mail pieces that will be delivered to their residence for the day. This preview includes images of the mail pieces where the sender and return address are visible. The service also provides the tracking status of the consumer’s packages, and notes packages that are expected to arrive soon. After signing up for the Informed Delivery service and verifying their identity on the USPS website, the user can view the Informed Delivery Daily Digest as a dashboard on the USPS website or through an email notification sent to their provided email address. If there are links or QR Codes, the service offers the user the option to “Do more with your Mail” and redirect to the provided link.

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. As the current version of Informed Delivery is image and text-based, USPS intends to implement a version of the service that is accessible to visually impaired users, which allows these users to receive an audio description of their Informed Delivery Daily Digest emails. Users also have the option to utilize voice commands to perform the various functionalities in the application, such as receiving an audio description of their mail data.

**1.1.2 Purpose**

The primary purpose of this Project Plan Document is to provide a detailed description of the USPS Informed Delivery Application, which allows visually impaired users to view what is in their mailboxes such as view images of their mail, receive tracking updates for their incoming packages, as well as provide an audio explanation of their emails in a daily email format. The Project Plan document is intended to be used by the development team, which consists of Project managers, Software Developers/Testers, and Business Analysts who could improve the application.

This Project Plan Document is part of a set of documents created to aid in developing the USPS Informed Delivery Application and to provide artifacts with vital information for the application’s ongoing support and operation throughout its life cycle.

**The following documents are included in the entire documentation package:**

* Project Plan/Due by 06/03/2022
* Software Requirements Specification/Due by 06/03/2022
* Technical Design Document/Due by 06/17/2022
* Add Software Test Plan to Project Plan: Due by 06/17/2022
* Programmers Guide/Due by 07/29/2022
* Deployment and Operations Guide/Due by 07/29/2022
* Add Software Test Plan to Project Plan/ Due by 07/29/2022
* User Guide/Due by 08/09/2022
* Test Guide/Due by 08/09/2022
* Programmer Guide/Due by 08/09/202

## ****Statement of Need****

The purpose of the Informed Delivery mobile application is to provide visually impaired USPS consumers an audio summary of their Informed Delivery Daily Digest emails and camera-taken mail piece photos, as these emails solely consist of images and text. The Informed Delivery mobile application meets the standards outlined in Section 508 of the Rehabilitation Act of 1973, by providing visually impaired users an alternative method (audio descriptions) to receive the same content that is provided to all consumers.

## ****Vision Statement****

Team A’s vision is to enhance the image-based Informed Delivery experience for visually impaired users, by providing informative audio descriptions and follow-up actions for their Daily Digest emails and camera-taken mail images through an easy-to-use mobile application.

## ****Stakeholders****

In this project, the phrase “Stakeholder” is used to describe the group of people and the development team who is directly or indirectly involved in the project and who may be impacted or affected by the project’s outcome.

The professor, mentors, and development team are all internal stakeholders, since they are all directly involved in this software project. An external stakeholder is not directly involved in this project. However, the client determines the project’s needs and assists in the elicitation process during the requirement collecting phase. The main goal of this project is being created for the client.

**The following tables represent all stakeholders for the USPS Informed Delivery APP. The stakeholders include UMGC Associates, USPS Experts, and the Development Team.**

**Table 1**

*UMGC Associates*

|  |  |
| --- | --- |
| **Name** | **Role** |
| Mir Assadullah | Professor |
| Roy Gordon | Mentor |
| Robert Wilson | Mentor |

**Table 2**

*USPS Experts*

|  |  |
| --- | --- |
| **Name** | **Role** |
| Bob Dixon | Client |
| Roy Gordon | Mentor |

**Table 3**

*Development Team*

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

## ****Acronyms and Abbreviations****

API - Application Programming Interface

CIA - Confidentiality, Integrity, and Availability

FR - Functional Requirement

GCP – Google Cloud Platform

GUI – Graphical User Interface

iOS – iPhone Operating System

NFR – Non-functional Requirement

NIST – National Institute of Standards and Technology

OCR - Optical Character Recognition

QR – Quick Response

RMF – Risk Management Framework.

UI – User Interface

USPS – United States Postal Services

RACI – Responsible, Accountable, Consulted, Informed

# ****Project Assignment****

## ****Project Scope****

The project scope is to develop an application tailored for visually impaired users. USPS wants to improve user experience by using voice interaction to drive the application navigation and features. With the collaborative effort of Team B, Team A will develop the USPS Informed Delivery Mobile application for both Android and Apple iOS. As of 2015, there are 3.2 million Americans with visual impairment. These numbers are expected to be more than double by 2050 due to baby boomers hitting 65 by 2029 ("Visual impairment, blindness cases in the U.S. expected to double by 2050", 2016). The mobile cross-platform application will assist impaired users with retrieving daily information on mail and packages arriving at their homes by interacting with them via speech.

The application will signal the user when a daily digest email arrives and listen for keywords/phrases to interact with them. It will then convert those phrases to text to determine what actions to trigger on the application. The user will also have the ability to press the mic button to request details, scan mail to aid the user verbally and inform them of what mail they received. The application can also scan physical mail and verbally tell the user the sender and the recipient information displayed. It will also check for logos from daily digest emails and scan mail to provide additional information on the sender.

The application USPS Informed Delivery Mobile Application will be delivered using Flutter, which is an open-source framework for developing cross-platform applications using a single codebase and the Dart programming language. Google's Cloud Services Platform (GCP) will be extensively used to detect user inputs and provide options on information that becomes available.

**Within the scope of the project:**

* GCPs detect logos will be used to get more information on the organization that sent the mail or package.
* GCP's Optical Character Recognition (OCR) will be used to detect text or handwritten images to determine the sender and recipient's information.
* The application shall scan QR codes and mail envelopes using the camera.
* The Mobile application shall be compatible with the Android and iOS mobile operating systems.
* The application will be developed using flutter and Dart, and interface with Google Cloud API.
* The application shall be tested using a software test plan.
* The application shall have a User Interface (UI) that will enable these features and integrate with Team B’s Graphical UI efforts.

**Out of scope of the project**

* A database system will not be built/used for the application.
* The project will not include user profiles for the user (Team B's responsibility/ profile will be used to connect to email).
* An admin role will not be implemented in the application (focuses solely on the user).

## ****Project Objectives****

The objectives listed below adhere to the "SMART" criteria (specific, measurable, attainable, relevant, and time-bound).

* Develop the USPS Informed Delivery app by August 9th, 2022, to enable visually impaired users to get details on daily USPS mail and package delivery.
* Develop a 508-compliant application by August 9th, 2022.
* Turn in each Project Milestone deliverable on the scheduled time and date.
* Design and develop a functional graphical user interface (GUI) with integrated features from both teams by milestone 3 (7/29).
* Ensure all GCP features have the required security controls to allow necessary access to features deployed on the application.
* Ensure the application is deployed to both Android and IOS mobile platforms by the end of the project deadline (August 9th, 2022).

### ****Project Requirements****

The USPS informed Digest Application's functional and non-functional requirements are listed below.

**Functional Requirements (FR)**

F-1: The application shall be trained to recognize the user's voice.

F-2: The application shall feature a turn-on mic button to start capturing the user's speech.

F-3: The application shall recognize the user's unique voice and phrases. These phrases will trigger application actions.

F-4: The application shall have a user-friendly, flexible, functioning GUI.

F-5: The GCP-based features shall be a standalone widget/application.

F-6: The application shall notify the user when a new USPS delivery digest email is received and provide options for the user to take voice action.

F-7: The application shall get images from the informed delivery daily digest, run image processing software to detect logos, and type or handwritten text to determine the sender and recipient information. It will then read it audibly to the user.

F-8: The application shall scan an image of a mail through the phone camera and, using F-8, read aloud the contents of the scanned piece.

F-9: The application shall read any Quick Response (QR) codes or barcodes on the mail pieces and offer the user to visit those websites for additional information

F-10: The application shall have the key phrases for features mentioned in F-8 through F:10.

F-11: The application shall work on iOS and Android platforms and use the same interface for simplicity.

F-12: The application shall provide embedded training on using the app's functionalities.

F-13:   The user shall be able to use commands to use all application features.

**Non-Functional Requirements (NFR)**

NFR-1: The application shall integrate with speech API to support speech-to-text, text-to-speech, and wake-up words.

NFR-2: The application shall be built using the Dart programming language.

NFR-3: The application's user interface (UI) will be built utilizing Flutter, an open-source UI toolkit for cross-platform applications.

NFR-4: The mail images follow standard mail guidelines (sender's information is on the top left, and the recipient's information is in the center).

NFR-5: The application shall be Section 508 compliant for partially impaired users.

### ****Project Assumptions****

**Team**

* Team Members shall stay in their assigned roles throughout the project. In addition, they shall assist members in other roles.

**Scope**

* The scope may vary after acceptance and approval of the project plan by our customer, Robert Dixon, and class professor, Dr. Mir Assadullah. Due to time constraints (12 weeks), variation will be handled by the change management process via PMs, which will channel changes up to the approvers.

**Methodology:**

* The project will follow the Agile Scrum Methodology approach throughout its life cycle.
* Weekly sprints shall be conducted for this (12 weeks) project.
* Daily meetings via status update meetings, posts, emails, or chats shall be provided to keep track of team members' tasks and their progress.

**Budget**

* No budget will be allocated for hardware or software needs (all free/hardware – emulators, software – all open source)
* The labor estimated cost budget is **$79,000**. See section 2.4 for a breakdown.

**Technology**

* The mobile application will be written in the Dart programming language.
* The project will use the Flutter, the cross-platform, open-source UI framework to build the application.

**End-User**

* The user will have access to a mobile device.
* The application will be available for Android and iOS mobile device users.
* The user can assign the proper device permissions to grant application access to audio input and output, camera, and network components.
* The user's device can connect to the internet.
* The user has a basic understanding of smartphone functionality.

### ****Product Approval/Acceptance Criteria****

The project has four Milestones that Team A must accomplish by August 09, 2022. The project milestones are shown in the table below, along with their deadlines.

**Table 4**

*Project Milestones*

|  |  |  |
| --- | --- | --- |
| Milestone | Assignment | Due Date |
| 1 | Milestone 1 | Due Date: 06/03/2022 |
| 2 | Milestone 2 | Due Date: 06/17/2022 |
| 3 | Milestone 3 | Due Date: 07/29/2022 |
| 4 | Milestone 4 | Due Date: 08/09/2022 |

The professor, Dr. Mir Assadullah, and the client, Bob Dixon will analyze and evaluate each milestone. If the deliverable requires modification, the changes will be integrated into the Milestone deliverable that follows. In addition, all possible modifications are communicated to all team members. The client will approve the completion of the project. To ensure that all team members are on the same page, Microsoft teams, Google Hangouts, Meets, and emails are primarily used for communication.

### ****Project Cost****

Team A will not spend any money on the software as well as hardware tools in order to build the application for this project. All six members in Team A will devote a set number of work hours every day for seven days in order to complete the project in 11 weeks. The table below shows the predicted total hours and cost, as well as hourly rates based on glassdoor.com.

**The hourly rate is as follows:**

* Project Manager: $68/hour
* Lead Developer: $60/hour
* Developer: $52/hour
* Business Analyst: $45/hour
* Tester: $50/hour

**Table 5**

*Project Cost Breakdown*

|  |  |  |  |
| --- | --- | --- | --- |
| **Breakdown of the Project Cost** |  |  |  |
| **Category** | **Project Members** | **Total Hours** | **Cost** |
| *Project Manager* | Reshawna Sampson | 250 | $17,000 |
| *Lead Developer* | Stanley De Jesus | 250 | $13,500 |
| *Business Analyst* | Sheena Johnpeter | 250 | $11,250 |
| *Business Analyst* | Ananya Srinivasan | 250 | $11,250 |
| *Developer* | Arnold Detoito | 250 | $13,000 |
| *Developer* | Barry Gartrell | 250 | $13,000 |
| *Software/Hardware tools* | 0 | 0 | $0 |
| **Total** | **6 members** | **1500 total hours** | **$79,000 total cost.** |

**\*\*\*Note: All team members shall assume the role of Tester when applicable.**

### ****Change Management****

In case any issues are identified with the current project requirements by the stakeholders, the team will utilize the following change management process:

1. A change request is sent to the team through email by the stakeholder, with the title and description of the change that is to be made. Additionally, the email will include the submitter’s name and contact information (preferably email), if there is a need to follow up with them regarding the request.
2. The business analysts will perform an impact analysis to determine whether the change is necessary to the success of the project given the current requirements and resources. If it is deemed to be necessary, the analysts will determine the level of priority for the change. Each issue is assigned a priority level, based on the criteria in the table below.

**Table 6**

*Change Priority Types*

|  |  |
| --- | --- |
| **Priority** | **Description** |
| High | A requirement change is necessary for the successful implementation of the application |
| Medium | A requirement change should be implemented in order for the application to function as expected by users |
| Low | A requirement change that does not impact the success of the application, and can be implemented when time and resource availability allow for it |

1. Based on the analysis, the stakeholders will approve or reject the change request.
2. The project manager will create an issue for approved change requests in the project repository in GitHub. Each issue will include the issue details and priority level. This list will serve as a change log to keep track of all requests.
3. The issue is assigned to the development team for implementation.
4. The implemented solution for the request is validated by the testing team and stakeholders for final approval.
5. Once approved, the change is integrated into the project and submitted to the GitHub repository.

# ****Risk Analysis****

## ****Risk Framework and Assessment****

The USPS delivery digest application adopts the National Institute of Standards and Technology (NIST) Risk Management Framework (RMF) to assess, quantify and manage risks. It is used in both the public and private sectors to mitigate risk. There are seven steps to the NIST RMF. This process requires actions to be completed in sequential order.

1. **Prepare** - Determines critical roles in managing risk and who is responsible, determines risk appetite, assessing potential threats across the organization. It also identifies how continuous monitoring will be accomplished.

2. **Categorize** – Identifies and categorizes all risk of potential security breaches across the organization and how the CIA (Confidentiality, Integrity, and Availability) impacts will affect the organization.

3. **Select** – Determine what security controls are needed to protect each system that part of operations. Decisions are documented and tailored to companies' current and future needs.

4. **Implement** – This step implements the best security controls options and ensures it is implemented correctly and functioning as expected.

5. **Assess** – Implemented security controls must be evaluated to ensure the desired outcome is achieved. Any issues encountered must have a plan to address any deficiencies.

6. **Authorize** – Senior Management Official within the organization must be appointed to determine whether operations and security controls are sufficient. Authorization must be received before the system/application goes operational. Our appointed/approving official is our customer, Robert Dixon.

1. **Monitor** – Continuously monitor controls that were implemented and system risk.

See below for the RMF steps, current progress with comments, and overall status.

**RMF - Current Project/Application Risk Status**

**Table 7**

*Risk management Framework Identified for the Project*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Progress w/ comments** | **Milestone** | **Status** |
| Prepare | Identify decision maker  - Bob Dixon, USPS Director of products and innovations, and Professor Mir Assadullah.  Identify risk concerns/organizational risk appetite  - Stakeholders have identified that the product must be 508 compliant and use certain technologies.  - Stakeholder (customer) wants to avoid privacy liability by minimizing stored data. | 1 | **Complete** |
| Categorize | See sections 3.2 - 3.4 tables for details  3.2 – Application Risks, Vulnerabilities,  Mitigation  3.3-3.4 - Project Risks and Mitigation for analysis and overall impact level. | 1 | **Complete** |
| Select | See tables in sections 3.2 – 3.4 on how the team will mitigate application and project risks  3.2 – Application Risks, Vulnerabilities,  Mitigation  3.3 - 3.4 - Project Risks and Mitigation for analysis and overall impact level. | 1 | **Complete** |
| Assess | Completed during Milestone 1- Presentation | 1 | **Complete** |
| Implement | The team has started getting familiar with the required technologies and feature requirements. | 2 | *In progress* |
| Authorize | This step will start during the Milestone 4 presentation. The customer will decide if it meets his requirements and whether to deploy it as a USPS solution. | 4 | Not Started |
| Monitor | Once deployed for customer use, the application will be monitored, and USPS and futures teams will address any user concerns. This step is outside of the project scope due to the project timeline. |  | N/A |

## ****Application Risks, Vulnerabilities and Mitigations****

The following tables outline the United States Postal Service® (USPS) Informed Delivery application's possible risks and vulnerabilities.

**Table 8**

*Application Risks and Vulnerabilities*

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk No.** | **Risk** | **Risk Description** | **Vulnerability** |
| AR-1 | Compromised QR code | Damaging the end user's device due to malicious attack or malware brought on by compromised QR code. | United States Postal Service® (USPS) Informed Delivery application not being able to detect compromised QR code that could be redirected to a malicious website. |
| AR-2 | Image processing malfunction | Losing confidence of the end user due to inability of the application to run image processing correctly. | United States Postal Service® (USPS) Informed Delivery application not being able to run image processing. |
| AR-3 | Scanned image being hacked / stolen | Losing privacy or personal address information due to scanned image being hacked/stolen. | United States Postal Service® (USPS) Informed Delivery application not being able to secure or protect the scanned image. |
| AR-4 | End user not giving camera / audio usage permission | Disrupting application functionality due to missing camera / audio usage permission | United States Postal Service® (USPS) Informed Delivery application not being able to function due to missing permission. |

**Table 9**

Application Risk Mitigations

| **Risk No.** | **Risk** | **Risk Level** | **Likelihood Level** | **Impact level** | **Risk Mitigation** |
| --- | --- | --- | --- | --- | --- |
| AR-1 | Compromised QR code | High | Medium | High | Application developers and engineers should plan on incorporating technology that would detect compromised QR code. |
| AR-2 | Image processing malfunction | Medium | Low | High | Application developers and engineers should plan on performing comprehensive validation and testing. |
| AR-3 | Scanned image being hacked/stolen | High | Medium | High | Application should implement a mechanism to safeguard the scanned images. |
| AR-4 | End user not giving camera/audio usage permission | Medium | High | Medium | Application should show a screen that tells the end user to go to settings to give camera/audio usage permission. |

## ****Project Risks and Mitigation****

The following table outlines the United States Postal Service® (USPS) Informed Delivery application's possible project risks and mitigations.

**Table 10**

*Project Risks and Mitigation*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk No.** | **Risk** | **Risk Level** | **Likelihood Level** | **Impact Level** | **Risk Mitigation Method** |
| PR-1 | Project schedule not clearly defined | Low | Low | High | Develop a comprehensive project schedule that clearly defines the deliverables, start and end date, and who is responsible for delivering specified tasks. |
| PR-2 | Project scope creep | Low | Low | High | Ensure all stakeholders have been consulted and all the requirements have been clearly documented and approved. |
| PR-3 | Problem with project communications | Low | Low | Medium | Ensure the team members have agreement on the mode of communication before the start of the project. |
| PR-4 | Inadequate of knowledge / skills | High | High | High | Ensure the team members have enough time training and researching the development programming languages and tools. |
| PR-5 | Unsatisfied client | Low | High | Low | Ensure that the client is updated throughout the project and set expectations. |

**3.4. Project Risk Matrix**

Each project risk is reviewed and categorized to determine the likelihood of it occurring and the potential impact if this risk did occur. The likelihood and the potential impact are both measured as High, Medium, or Low.

The following table outlines the United States Postal Service® (USPS) Informed Delivery application's risk matrix and mitigation.

**Table 11**

*Project Risk Matrix*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Impact** | | |
| High | Medium | Low |
| **Likelihood** | High | * Inadequate of knowledge / skills |  | * Unsatisfied client |
| Medium |  |  |  |
| Low | * Project schedule not clearly defined * Project scope creep | * Problem with project communications |  |

# ****Organization of Project****

## ****Communication****

Microsoft teams, Google Hangouts, Meets, and emails are the most often used communication tools. For Milestone 1, Team A mainly interacts on a regular basis through Google Hangouts’ chat features. Team A may interact using Google Hangouts and Meets in a variety of methods, including Instant Messaging, Voice Calls, HD Video Calls, HD Group Conferencing, and Built-in screen sharing.

Team A has three meetings in a week on Wednesday, Friday, and Sunday to discuss the project’s present progress and what is required to accomplish it by the deadlines. The regular communication ensures that all members in Team A are on schedule to meet their milestones. Google Drive, which contains all documents, is used to track each project milestone for Team A. This promotes teamwork since all team members may view and modify all documents at the same time.

In Addition, Team A’s ultimate purpose is to succeed in this project through outstanding team communication, teamwork, and effective collaboration with Team B. Therefore, Team A utilizes the primary two communication methods: Cross-Team Communication and Stakeholder Communication approaches. The key advantage of using the Cross-Team Communication approach is that Team A will create the USPS Informed Delivery Mobile Application for both Android and iOS with the effective collaboration with Team B. Furthermore, it is of crucial importance to Team A’s Project Manager, Reshawna Sampson to work closely with stakeholders since their feedback/options and actions will have a direct impact on the project outcome. By adopting the Stakeholder Communication approach, time will be saved, obstacles will be removed, and the project will be completed on schedule.

Team B has been using the task tracking and document management platform Jira Software by Atlassian. This software allows team members to see what tasks they are assigned to, see other team members tasks, and be able to see the progress on the tasks assigned. By Milestone Two, Team A will also be utilizing this software. This is in hopes for more consistent collaboration between Team A and Team B.

## ****Explanation of Methodology****

For this project we will be using the Scrum methodology. Scrum is an iterative software development methodology where teams work in sprints to accomplish goals. Tasks are parts of overall requirements that are selective enough to be completed within the sprint. On top of the sprints, the team will meet every other day for a Scrum meeting that should be brief to discuss how each team member is progressing on their task for the sprint. For this project, our team will use sprints that last one week to meet the needs of the project.

### Scrum Artifacts

**Scrum produces a series of different documents that are associated with the development process. The first is the Product Backlog which is a list of tasks like new features, enhancements, bug fixes, or work requirements. A Product Backlog is created from needs relating to customer support or other forms of analysis. The team will use this to keep track of changes needed for the project. The second is the Sprint Backlog which is a set of tasks that have been approved to be worked on during the next sprint. The team will use this to keep track of work that will need to be completed. The final artifact is the Product Increment which is a deliverable to the customer that is produced by completing tasks from the Product Backlog during sprints.**

## ****Project Staffing****

Team A will be staffed with a team that has a Project Manager, a Business Analyst, Testers, and Developers. The team members will need to double up on some roles as there are a limited number of team members. The Project Manager will also have to function as a Scrum Master. Figure one illustrates the organizational chart of the SCRUM team roles.

**Figure 1**

*Scrum Organization Chart*

Diagram

Description automatically generated

**Table 11**

*Team Contact Information*

|  |  |
| --- | --- |
| Team Member | Email |
| Reshawna Sampson | reshawanasampson@gmail.com |
| Arnold Detioto | arnold.detoito@gmail.com |
| Stanley De Jesus | sdejesus@gmail.com |
| Ananya Srinivasan | aasrin99@gmail.com |
| Sheena Johnpeter | jsheena1927@gmail.com |
| Barry Gartrell | gartrellbarry@gmail.com |

## ****Roles and Responsibilities****

**Table 13**

*Team Members*

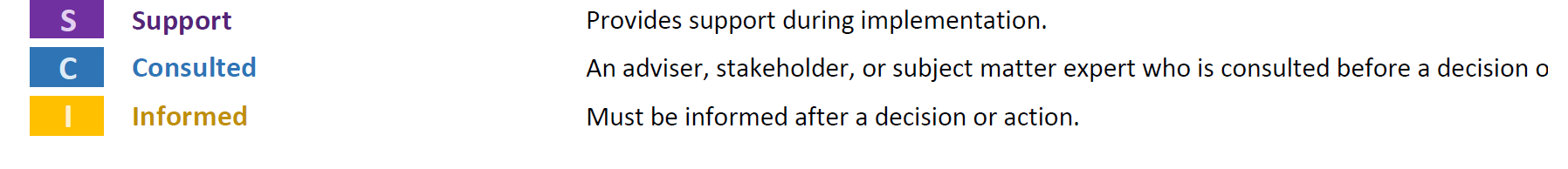
| **Name** | **Role** |
| --- | --- |
| Reshawna Sampson | Project Manager |
| Barry Gartrell | Developer |
| Stanley De Jesus | Developer |
| Ananya Srinivasan | Business Analyst |
| Sheena Johnpeter | Business Analyst |
| Arnold Detoito | Developer |

Figure two represents the roles and responsibilities of stakeholders throughout each milestone. **Figure 2**

*RACI Matrix*

A picture containing chart

Description automatically generated



## ****Project Tools****

The tools of the project are listed below:

**Table 14**

*Project Tools*

| Tool | Description |
| --- | --- |
| Android Studio | An IDE, integrated development environment, used to develop Android applications. Version 2021.2.1 Patch 1 |
| Dart | A language that allows for adaptive and quick development of multi-platform applications. Version 212.5744 |
| Flutter | UI toolkit developed by Google. Version 68.1.2 |
| GitHub | A Source Control service that allows us to organize the code throughout the development of the project. |
| Microsoft Teams | A communication platform that will allow us to organize meetings, share documents, and track progress during the project. |
| Microsoft Word | A document editing application that allows the team to work collaboratively on all documentation. |
| Google Drive | A platform that allows the team to share documents and collaboratively compose documents/presentations for the project. |
| Google Vision APIs | A google cloud service used to implement image recognition services. Version May 20, 2022 Release |

Table 14: Project Tools

## ****Project Storage****

**For this project, Team A will be utilizing Google Drive and Microsoft Teams Files as storage areas for all documentations. GitHub will be used as the source control for the project code.**

## ****Deliverables****

The following table outlines the United States Postal Service® (USPS) Informed Delivery application's milestones and the associated deliverables.

**Table 15**

*Deliverables*

| **Milestone** | **Deliverable** | **Description** | **Due Date** |
| --- | --- | --- | --- |
| 1 | Project Plan (PP) | The Project Plan (PP) is a document that defines the management, execution, and monitoring of the successful completion of the United States Postal Service® (USPS) Informed Delivery application. | Week 2 (6/3/2022) |
| Software Requirements Specification (SRS) | The System Requirements Specification (SRS) is a document that describes the features of the United States Postal Service® (USPS) Informed Delivery application and how the application will be expected to perform. |
| 2 | Technical Design Document (TDD) | The Technical Design Document (TDD) is a document that describes the design and implementation of the United States Postal Service® (USPS) Informed Delivery application. | Week 4 (6/17/2022) |
| Add Software Test Plan (STP) to the Project Plan (PP) | The Software Test Plan (STP) is a document that outlines how the testing activity of the United States Postal Service® (USPS) Informed Delivery application is going to take place in a project. It details the test plans, test cases and schedule of testing activities.  The Software Test Plan (STP) must be incorporated into the Project Plan (PP). |
| 3 | Programmer Guide (PG) | The Programmer Guide (PG) is a document that guides the future developers and engineers who will be supporting and maintaining the United States Postal Service® (USPS) Informed Delivery application. | Week 10 (7/29/2022) |
| Deployment and Operations Guide (Runbook) | The Deployment and Operations Guide (Runbook) is a document that guides the developers and engineers who are responsible for deploying and making the United States Postal Service® (USPS) Informed Delivery application operational and accessible in the Apple Store and Google Play Store. |
| Add Software Test Plan (STP) to the Project Plan (PP) (if needed) | The Software Test Plan (STP) is a document that outlines how the testing activity of the United States Postal Service® (USPS) Informed Delivery application is going to take place in a project. It details the test plans, test cases and schedule of testing activities  Additional changes to the Software Test Plan (STP) must be incorporated into the Project Plan (PP). |
| 4 | User Guide (UG) | The User Guide (UG) is a document that guides the end users how to operate and control the United States Postal Service® (USPS) Informed Delivery application. | Week 12 (8/9/2022) |
| Test Report (TR) | The Test Report (TR) is a document that details the United States Postal Service® (USPS) Informed Delivery application test objectives, test cases and test results.  It also summarizes the test results with a "pass" or "fail" notation. |
| Programmer Guide (PG) (if needed) | The Programmer Guide (PG) is a document that guides the future developers and engineers who will be supporting and maintaining the United States Postal Service® (USPS) Informed Delivery application.  Additional changes to the Programmer Guide (PG) must be made and submitted during this milestone. |

## ****Project Schedule****

**Figure 3**

Graphical user interface, application

Description automatically generated*Project Schedule*

Graphical user interface

Description automatically generated with medium confidence

**Figure 4**

Quick Reference Project Schedule

# ****Software Test Plan****

## Purpose

The purpose of the project test plan is to detail the testing approach used for the USPS Informed Delivery mobile application. The goal of the test plan is to identify potential issues with the application that prevent it from fulfilling the outlined project requirements. If issues are found with the project, the development team will mitigate the problem so that the user experience is not impacted.

## Scope

The scope of the test plan will cover the three testing approaches, test cases, and acceptance criteria that will be used to evaluate the system. The three testing approaches (discussed further in Section 5.4) utilized are:

* Unit/Widget Testing
* Integration/Regression Testing
* User Acceptance Testing/Focus Group

As the application is developed by two teams, integration testing is crucial for ensuring that the application frontend and backend are functioning accordingly. Unit testing will be utilized by the team throughout the development process to test the functionality of separate components. User Acceptance Testing (UAT) will be utilized to evaluate whether the application meets the requirements of the client and end-users – particularly users who are visually impaired. If issues in the system are found during the testing process, the issue is prioritized by severity and added as a defect in the project GitHub repository.

## Test Items

### To Be Tested

1. The application shall connect to the Google Vision API.
2. The application shall retrieve text from an image.
3. The application shall retrieve information from logos.
4. The application shall create a daily digest of the user’s Informed Delivery.
5. The application shall read a QR code.
6. The application shall read a barcode.
7. The application shall check if the image contains no gatherable information.
8. The application shall validate the clarity of the returned information.
9. The application shall open the device’s camera.
10. The application shall save any manually taken photos locally to the device.

### Not to be Tested

1. The application shall be built using the Flutter framework.
2. The application shall be built using the Dart programming language.
3. The application shall run on the Android Operating System.
4. The application shall run on iOS.

## Approach

As bugs are discovered they will be reported to the GitHub board. They will be logged with the information relating to their tests and the specific requirement that it is associated with.

**Unit/Widget Testing** - Unit tests will provide the bulk of the test that will be performed throughout the development of this project. These tests will represent most of the code coverage. These tests will need to be run manually and regularly when the functionality of features has changed.

**Integration/Regression Testing** - As components of the application are completed it will be vital for the teams to perform integration tests regularly. The teams must ensure that we have a robust testing schedule to minimize the amount of re-work and defects that could be present. Integration testing will be performed using Flutter’s package called integration test. This package functions similarly to how other self-driving tests function.

**User Acceptance Testing / Focus Group** - User Acceptance Tests will verify that the application meets the objectives of a requirement. This will allow the team to ensure that they can meet the needs of the end-users and the client. The application should be tested by real users so that feedback can be used to guide further versions of the application.

## Pass/Fail Criteria

* API tests will be considered a success if the results should remain consistent after 10 attempts.
* Text verification will be considered a success if more than 90% of the found information is comprehensible.

## Defect Severity Levels

1. Low - Intended for non-essential issues that do not hinder functionality or performance.
2. Moderate - These can apply to both functional and non-functional requirements that could have a slight impact on the objective of the application.
3. Critical - Functional requirement bugs that need to be addressed.
4. Blocker - Issues that prevent further work from being completed and halt development on a project.

## Risks and Contingencies

1. Some words that may be misinterpreted by the application may cause users to not understand the message even though it passed our 90% verification.
2. There may be times when the Google API will be unavailable due to circumstances beyond our control.

## Test Cases

**Table 16**

*Test Case 1: API Connection*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to connect to the Google Vision API. |
| Test Type | Functional |
| Requirement | The application shall connect to the Google Vision API. |
| Req. # | IDA001 |
| Prerequisite | The user can access the internet. |
| Steps | 1. The application makes a test request to the Google API. |
| Expected Output | The connection to the API is verified. |
| Assumption | None |

**Table 17**

*Test Case 2: Image Information Retrieval*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to retrieve text from an image. |
| Test Type | Functional |
| Requirement | The application shall retrieve text from an image. |
| Req. # | IDA002 |
| Prerequisite | The API is functioning properly. |
| Steps | 1. The application makes an API call to retrieve the information from the image. |
| Expected Output | The text results from the image. |
| Assumption | None |

**Table 18**

*Test Case 3: Logo Detection*

|  |  |
| --- | --- |
| Description | The application will be able to detect and retrieve information about a logo. |
| Test Type | Functional |
| Requirement | The application shall retrieve information from logos.  The API is functioning properly. |
| Req. # | IDA003 |
| Prerequisite | The user has already provided a logo image.  The API is functioning properly. |
| Steps | 1. The application makes an API call to retrieve the information about the logo. |
| Expected Output | Information about the logo. |
| Assumption | None |

**Table 19**

*Test Case 4: Daily Digest*

|  |  |
| --- | --- |
| Description | This test will verify that the application can create a daily digest of the user’s Informed Delivery information. |
| Test Type | Functional & User Acceptance |
| Requirement | The application shall create a daily digest of the user’s Informed Delivery.  The API is functioning properly. |
| Req. # | IDA004 |
| Prerequisite | The application has already gathered the images needed from the user’s email. |
| Steps | 1. For each image in the user’s Informed Delivery email, the application will make an API call to retrieve their information. 2. The application consolidated the information. |
| Expected Output | The user’s daily digest information. |
| Assumption | None |

**Table 20**

*Test Case 5: QR Code Reader*

|  |  |
| --- | --- |
| Description | This test will verify that the application is able to read QR codes and gather web addresses from them. |
| Test Type | Functional |
| Requirement | The application shall read a QR code. |
| Req. # | IDA005 |
| Prerequisite | The user has provided an image of a QR code.  The API is functioning properly. |
| Steps | 1. The application makes the API call to retrieve the web address from the QR code. |
| Expected Output | The web address associated with the QR code. |
| Assumption | None |

**Table 21**

*Test Case 6: Barcode Reader*

|  |  |
| --- | --- |
| Description | This test will verify that the application can read Barcodes. |
| Test Type | Functional |
| Requirement | The application shall read a barcode. |
| Req. # | IDA006 |
| Prerequisite | The user has provided an image of a barcode.  The API is functioning properly. |
| Steps | 1. The application makes the API call to retrieve the information from the barcode. |
| Expected Output | The information associated with the barcode. |
| Assumption | None |

**Table 22**

*Test Case 7: Image Verification*

|  |  |
| --- | --- |
| Description | This test will verify that the application can determine if an image was illegible. |
| Test Type | Functional |
| Requirement | The application shall check if the image contains no gatherable information. |
| Req. # | IDA007 |
| Prerequisite | The user has already provided an image.  The API is functioning properly. |
| Steps | 1. The application makes an API to get information about the image. 2. The application checks if the image returned any information was returned from the API. |
| Expected Output | A message saying that there was nothing returned from the image. |
| Assumption | None |

**Table 23**

*Test Case 8: Clarity Vest*

|  |  |
| --- | --- |
| Description | This test will make sure that the application is returning clear information. |
| Test Type | Functional |
| Requirement | The application shall validate the clarity of the returned information.  The API is functioning properly. |
| Req. # | IDA008 |
| Prerequisite | The application has already processed the text of an image. |
| Steps | 1. The application makes an API call to verify the quality of the returned text. |
| Expected Output | Either the correct text from an image or a message stating that the image text was not able to be processed. |
| Assumption | The image is not blank. |

**Table 24**

*Test Case 9: Open Camera*

|  |  |
| --- | --- |
| Description | This test will verify that the application has the ability to access the user’s camera. |
| Test Type | Functional |
| Requirement | The application shall open the device’s camera. |
| Req. # | IDA009 |
| Prerequisite | None |
| Steps | 1. The user opens the app 2. The user selects the open camera button. 3. On-screen the user's camera is opened. |
| Expected Output | The user’s device’s camera is open. |
| Assumption | The user’s device has a camera. |

**Table 25**

*Test Case 10: Photo Saved Locally*

|  |  |
| --- | --- |
| Description | This test will verify that images taken by the user’s device are saved locally to their device. |
| Test Type | Functional |
| Requirement | The application shall save any manually taken photos locally to the device. |
| Req. # | IDA010 |
| Prerequisite | The user has already opened their camera in the application. |
| Steps | 1. The user takes a picture. 2. The picture is saved locally to their device. |
| Expected Output | Location of the saved image. |
| Assumption | None |

**Table 26**

*Test Case 11: Built Using Flutter Framework*

|  |  |
| --- | --- |
| Description | This test verifies that the application was built using the flutter framework. |
| Test Type | Functional |
| Requirement | The application shall be built using the Flutter framework. |
| Req. # | IDA011 |
| Prerequisite | 1. The application meets all requirements and is developed using the Flutter Framework. |
| Steps | None |
| Expected Output | None |
| Assumption | None |

**Table 27**

*Test Case 12: Built Using Dart Language.*

|  |  |
| --- | --- |
| Description | This test verifies that the application is built using the Dart programming language. |
| Test Type | Functional |
| Requirement | The application shall be built using the Dart programming language. |
| Req. # | IDA012 |
| Prerequisite | None |
| Steps | 1. The application meets all requirements and is developed using the Dart programming language. |
| Expected Output | None |
| Assumption | None |

**Table 28**

*Test Case 13: Android Compatibility*

|  |  |
| --- | --- |
| Description | This test verifies that the application functions on an Android device. |
| Test Type | Functional |
| Requirement | The application shall run on the Android Operating System. |
| Req. # | IDA013 |
| Prerequisite | None |
| Steps | 1. The user opens the app 2. The user ensures that the app’s features are functional. |
| Expected Output | The application functions on an Android device. |
| Assumption | None |

**Table 29**

*Test Case 14: iOS Compatibility*

|  |  |
| --- | --- |
| Description | This test verifies that the application functions on an iOS device. |
| Test Type | Functional |
| Requirement | The application shall run on iOS. |
| Req. # | IDA014 |
| Prerequisite | None |
| Steps | 1. The user opens the app 2. The user ensures that the app’s features are functional. |
| Expected Output | The application functions on an iOS device. |
| Assumption | None |

# ****References****

Amazing. (2021, July 24). *Memory Enhancer Application - Project Plan*. Retrieved from https://umgc-cappms.azurewebsites.net/download/53068c5e-6a24-4414-90bc-16df208c6af8----ProjectPlan\_TeamAmazing\_v1.2.pdf

Amazing. (2021, July 2). Memory Enhancer Application - Software Requirements Specification. Retrieved from https://umgc-cappms.azurewebsites.net/download/b1497111-ddf1-4b9a-ac5d-71d2021f811f----SoftwareRequirementsSpecification\_TeamAmazing\_v1\_1.pdf

Davey, L. (2021, June 7). *Step-by-step guide to better cross-team collaboration*. Cross Team Collaboration**.**https://www.teamwork.com/blog/cross-team-collaboration/

Harris, C. (n.d.) Agile scrum artifacts. https://www.atlassian.com/agile/scrum/artifacts

Integrators, P. (2020, September 1). *Features of Google Hangouts*. Portal Integrators. https://portalintegrators.com/features-of-google-hangouts/

*NIST Risk Management Framework (RMF)( breakdown*. (n.d.). Retrieved from https://www.sailpoint.com/identity-library/nist-risk-management-framework/?utm\_id=414626188&msclkid=54c570fff8fb11fcdde9126b30139e8e

*Raci matrix template*. Vertex42.com. (2022, May 5). Retrieved June 3, 2022, from https://www.vertex42.com/ExcelTemplates/raci-matrix.html

*Salarissen per bedrijf*. (n.d.). Glassdoor. Retrieved May 31, 2022, from https://www.glassdoor.com/Salaries/index.htm

*Understanding and collaborating with stakeholders: 5 ways to work better together - @*. (2021, November 3). GatherContent. Retrieved June 2, 2022, from https://gathercontent.com/blog/five-ways-understand-collaborate-stakeholders

*Visual impairment, blindness cases in U.S. expected to double by 2050*. (2016, May 19). Retrieved from <https://www.nih.gov/news-events/news-releases/visual-impairment-blindness-cases-us-expected-double-2050>

*Widget Tests* (n.d.) https://docs.flutter.dev/testing#widget-tests

*ZipRecruiter*. (n.d.). Retrieved from <https://www.ziprecruiter.com>

Purdue Writing Lab. (n.d.). *APA tables and figures // Purdue Writing Lab*. Purdue Writing Lab. Retrieved July 29, 2022, from https://owl.purdue.edu/owl/research\_and\_citation/apa\_style/apa\_formatting\_and\_style\_guide/apa\_tables\_and\_figures.html