

Project Plan

Version 1.4

Presented by

Stefon Williams

Caleb Crickette

Thomas Barton

Eugene Kim

Nicholas Ballo

Abdul Kamara

Sohail Sobhani  
Anthony Cedeno  
Geoffrey Dean   
Jean-Nae' Dedrick

February 24, 2021

SWEN 670

University of Maryland Global Campus

For approval: Stefon Williams

Stefon Williams Date: 2/24/2021

For approval: Dr. Mir Assadullah

Date: 2/24/2021

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **01-30-2021** | **1.0** | **First Revision** | **Thomas Barton, Stefon Williams, Caleb Crickett, Sohail Sobhani** |
| **01-31-2021** | **1.1** | **Added some suggestions.** | **Thomas Barton, Stefon Williams, Caleb Crickett, Sohail Sobhani** |
| **02-10-2021** | **1.2** | **Milestone 2 Changes** | **Thomas Barton, Stefon Williams, Caleb Crickett, Sohail Sobhani, Nicholas Ballo** |
| **02-22-2021** | **1.3** | **Test Plan Identifier added** | **Thomas Barton, Stefon Williams, Caleb Crickett, Sohail Sobhani, Nicholas Ballo** |
| **03-08-2021** | **1.4** | **Test Plan Identifier edits** | **Anthony Cedeno** |

Table of Contents

[Background 6](#_Toc65097004)

[Statement of Need 6](#_Toc65097005)

[Vision Statement 6](#_Toc65097006)

[Project Tools 7](#_Toc65097007)

[Microsoft Office and Teams 7](#_Toc65097008)

[DevCloud Azure 7](#_Toc65097009)

[Github 7](#_Toc65097010)

[Google’s Dialogflow development console 7](#_Toc65097011)

[Project Deliverables and Milestones 7](#_Toc65097012)

[Definitions of Acronyms and Abbreviations 9](#_Toc65097013)

[Scope Management Plan 9](#_Toc65097014)

[Project In-Scope Activities 9](#_Toc65097015)

[Project Out-of-Scope Activities 10](#_Toc65097016)

[Success Criteria 10](#_Toc65097017)

[Project Assumptions 11](#_Toc65097018)

[Project Dependencies 11](#_Toc65097019)

[Project Constraints 11](#_Toc65097020)

[Project Approach 12](#_Toc65097021)

[Project timeline (Gannt Chart) 12](#_Toc65097022)

[Project Work Breakdown 12](#_Toc65097023)

[Sprint 1 13](#_Toc65097024)

[Sprint 2 13](#_Toc65097025)

[Sprint 3 13](#_Toc65097026)

[Sprint 4 13](#_Toc65097027)

[Requirements Management Plan 13](#_Toc65097028)

[Pending Functional Requirements 13](#_Toc65097029)

[1. Use Case: Extract Name(s) from speech. 13](#_Toc65097030)

[2. Use Case: Understand the user’s different intents from speech. 14](#_Toc65097031)

[3. Use Case: Extract entities and data from speech 14](#_Toc65097032)

[4. Use Case: Insert extracted entities and data into a database. 14](#_Toc65097033)

[5. Use Case: Service is integrated with mobile application 14](#_Toc65097034)

[Integration Management Plan 14](#_Toc65097035)

[Change Management Plan 14](#_Toc65097036)

[Quality Management Plan 14](#_Toc65097037)

[Resource Management Plan 15](#_Toc65097038)

[Project Team Roles and Responsibilities 15](#_Toc65097039)

[Responsibility Assigned Matrix (RACI) 16](#_Toc65097040)

[Risk Management Plan 17](#_Toc65097041)

[Risk Matrix 17](#_Toc65097042)

[Risk Analysis 18](#_Toc65097043)

[Risk Mitigation Strategies. 19](#_Toc65097044)

[Communication Management Plan 20](#_Toc65097045)

[Test Plan Identifier 20](#_Toc65097046)

[Introduction 20](#_Toc65097047)

[Purpose 20](#_Toc65097048)

[Test Scope 20](#_Toc65097049)

[*In-Scope* 21](#_Toc65097050)

[*Out-of-Scope* 21](#_Toc65097051)

[Test Objective 21](#_Toc65097052)

[Roles and Responsibilities 21](#_Toc65097053)

[Testing Schedule 22](#_Toc65097054)

[Key Milestones 23](#_Toc65097055)

[Testing Approach 23](#_Toc65097056)

[Test Automation 23](#_Toc65097057)

[Functional Requirements of Test Cases 23](#_Toc65097058)

[Test Plan Metrics 24](#_Toc65097059)

[Test Cases 24](#_Toc65097060)

[Test Case 01: Isolate primary user’s voice for use within the service 24](#_Toc65097061)

[Test Case 02: Dialogflow service is integrated within the mobile application 25](#_Toc65097062)

[Test Case 03: Receive report identifier from user 26](#_Toc65097063)

[Test Case 04: Get report fields from database 27](#_Toc65097064)

[Test Case 05: Process report contents and fields. 27](#_Toc65097065)

[Test Case 06: Generate dynamic intents to capture report fields 28](#_Toc65097066)

[Test Case 07: Identify report contents extracted by the Dialogflow agent 29](#_Toc65097067)

[Test Case 08: Store extracted report contents into a database 30](#_Toc65097068)

[Test Case 09: Initiate connection with database 31](#_Toc65097069)

[Test Case 10: Dialogflow Help Support Response 31](#_Toc65097070)

[Testing Criteria 32](#_Toc65097071)

[Suspension Criteria 32](#_Toc65097072)

[Resumption Criteria 32](#_Toc65097073)

[Test Completeness 32](#_Toc65097074)

[Test Deliverables 33](#_Toc65097075)

[Resource & Environment Needs 33](#_Toc65097076)

[Testing Tools 33](#_Toc65097077)

[Appendix A WBS 36](#_Toc65097078)

# Background

The Form Scriber application project was created by Dr. Mir Mohammed Assadullah. Dr. Assadullah created this project for SWEN 670 the Software Engineering Project or often called the Capstone Project. This project is designed to measure what students have taught throughout their Master’s degree. Completion of this team project is designed to integrate knowledge and skills gained through previous study and provide experience of the constraints commonly experienced in industry (scheduling, vagueness of clients). Dialogflow team, Mobile team, and DevSecOps team are the teams that have been created to collaborate and complete the project. The customers for the project are any general professional whom fills out forms after interacting with clients and/or customers.

The Form Scriber project will develop an AI service to help professionals to prepare text reports while interacting with their clients. An example of this may be listening to a medical professional as they speak to a patient, and records important information that the medical professional says during their conversation with the patient before recording to a form and into the an integrated and internal Google Docs storage system.

Our team will focus on building out the AI Service using Google Cloud Platform’s (GCP) DialogFlow service. Our service will integrate with the web and mobile UIs that will be built and designed by other teams in the overall project.

At the end of the project, the application will continue to be hosted on Github to be examined and downloaded for others use.

# Statement of Need

Public service professionals are constantly filing paperwork which takes up a significant portion of their job. In order to maximize time and productivity public service professionals need a tool that can automatically take input from service workers and organize it within form templates.

# Vision Statement

The Form Scriber project will integrate an AI service to help professionals in public service to prepare text reports after performing a portion of their job. An example of this may be recording a conversation between a patient and physician to prepare potential notes that the physician can edit before saving them in an integrated and internal Google Docs storage system.

Team Dialogflow will focus on building out the AI Service using Google Cloud Platform’s (GCP) DialogFlow service. Our service will integrate with the web and mobile UIs that will be built and designed by other teams in the overall project. The Web app will offer a completed form with collected data. The Mobile app will generate a report based on the form template.

The intended audience for this document is the stakeholders which includes the development team, and the project approval personnel. This document provides the details required to execute and control the Form Scriber project successfully. Details regarding facilitating communication among stakeholders, establishing the project’s deliverables, and information specific to the project’s document planning, assumptions, decisions, scope, and schedule baseline can also be found within this document. Throughout this document, “project” refers to the Form Scriber project.

# Project Tools

There are several key tools that will be used within the project to accomplish creating and delivering the end product.

## Microsoft Office and Teams

Microsoft Office tools such as Microsoft Word and PowerPoint will be necessary for creating document deliverables, as well as Microsoft project for managing the project hours and creating the project timeline graphics. Microsoft Teams will be an integral collaboration platform for document sharing and collaboration and otherwise being used for communication purposes.

## DevCloud Azure

This tool will be used as a task tracking and overall plan fulfillment.

## Github

This cloud platform will be used to host the project code deliverables, as well as allow for team collaboration and version management.

## Google’s Dialogflow development console

The Dialogflow interpreter will be developed using googles specialty built Dialogflow console which will allow for the customization and training of the Dialogflow service.

## Google Docs

This cloud file repository allows for the storage of template documents and finalized doucments.

# Project Deliverables and Milestones

The below table shows the deliverables that will be submitted for each milestone. Deliverables may be revised based on client and stakeholder feedback and submitted again to the next milestone.

Table 1 - Project Deliverables

|  |  |  |
| --- | --- | --- |
| Milestone | Deliverables | Due date |
| Milestone 1 | Software Reference Specification  Project Plan  Team Roles and Presentation | 2/2/21 |
| Milestone 2 | Technical Design Document  Software Test Plan  Presentation | 2/23/21 |
| Milestone 3 | Deployment and Operations Guide  Programmer Guide  Presentation | 3/16/21 |
| Milestone 4 | User Guide  Test Report  Final Presentation | 3/30/21 |

# Definitions of Acronyms and Abbreviations

The below table has a list of acronyms and abbreviations found in this document.

Table 2 - Acronyms and Abbreviations

|  |  |
| --- | --- |
| Acronym or Abbreviation | Definition |
| GCP | Google Cloud Platform |
| PMBOK | Project Management Body of Knowledge |
| QA | Quality Assurance |
| UI/UX | User Interface/User Experience |
| UMGC | University of Maryland Global Campus` |
| SDLC | Software Development Lifecycle |
| PMI | Project Management Institute |
| IT | Information Technology |
| EMR | Electronic Medical Record |
| UI | User Interface |
| REST | Representational State Transfer |
| GB | Gigabyte |
| SRS | Software Requirements Systems |
| Form Scriber | The name of the system |
| HTTP | Hypertext Transfer Protocol |
| JSON | JavaScript Object Notation |
| ID | Identification |
| GCP | Google Cloud Platform |
| OS | Operating System |
| E2E | End to End |

# Scope Management Plan

The project will be built with GCP’s DialogFlow service. In-scope activities will include project management, design, validation, testing, and deployment of the AI service and any other artifacts needed to successfully fill all requirements and milestones.

## Project In-Scope Activities

Activities in-scope focus on the necessary components required for a system build and maintaining scope over the course of the project’s lifecycle; each project artifact is tailored for ensuring effective management of project activities (Project Management Institute, 2017).

At the minimum, the below deliverables will be delivered by the end of the project:

* Project Plan
  + This would include the document being presented currently
* Test Plan and Report
  + The test plan would include the things that would need to be tested in order for the project to be considered fulfilling requirements.
* Software Requirements Specification (SRS)
  + The software requirements are defined in detail within this document, interpreted from stakeholder and client resources and sources.
* Technical Design Document
  + This document will detail the different aspect of the software for the Form Scriber and get into greater
* Form Scriber intent training
  + This is a general task for the creation of intents within the Google Dialogflow service, so that data may be extracted.
* Web hook creation.
  + The webhook creation will require collaboration with the web server team, and allow communication between the dialog flow and web server service.
* User Guide
  + This guide shall detail in what ways the application is intended to be used by the users.
* Programmer Guide
  + This guide will allow other programmers to pick up the nuances of how to set up this project and the reasoning behind the decisions that went into the project.
* Deployment and Operations Handbook
  + This handbook shall detail how to set up and maintain this project.
* Research Paper documenting the project
  + Various lessoned learned and other pieces of information gleamed from working on this project.
* Software Demo/Presentation
  + This shall be the final product where the project shall be presented and demoed to the key stakeholder.

## Project Out-of-Scope Activities

Out-of-Scope activities include those activities that will be accomplished by other teams. This includes but is not limited to:

* Mobile UIProject post development hosting.
* Project infrastructure hosting setup and maintenance.

## Success Criteria

The following are the minimum project success criteria to be considered an organizational enabler (Project Management Institute, 2018):

* Delivered according to the approved schedule and budget.
* All required documentation is provided to the stakeholder.
* All completed code at the end of the project schedule is provided to the stakeholder and hosted on GitHub.
  + \*Note: due to time constraints, code may not be fully functional but documentation will exist on any missing components.

## Project Assumptions

Assumptions identified by the team are:

* Code needs to be delivered in a functional state with minimal requirements satisfied.
* Currently it is assumed that only the professional’s voice will be recorded, meaning that the intents within Dialogflow will need to be tailored to that sort of input (onesided conversations or otherwise). An end product will not be delivered to a specific customer, only hosted on GitHub.
* The application will need to have some sort of network connection (either an internal network, or full internet connection) for the application to be functional.
* UI will primarily be handled by the mobile team.
  + \*Note: A simple, demo UI will be created to showcase and validate this team’s work.

## Project Dependencies

The DevSecOps team will need to provide our team with the tools and resources to standardize, interoperate, and create best practices and procedures. Specifically, the following tools and process will need to be formalized:

* Code Repository in the form of GitHub.
* Deployment Pipeline
* Artifact Repository in the form of azure dev ops.

The project is codependent with the mobile team to ensure that a cohesive, unified product is released. The dialog flow program is integral to the success of the project, as it will allow for the main functionality needed for the mobile component to get any information into the report forms.

## Project Constraints

Significant project constraints identified by the team are the following:

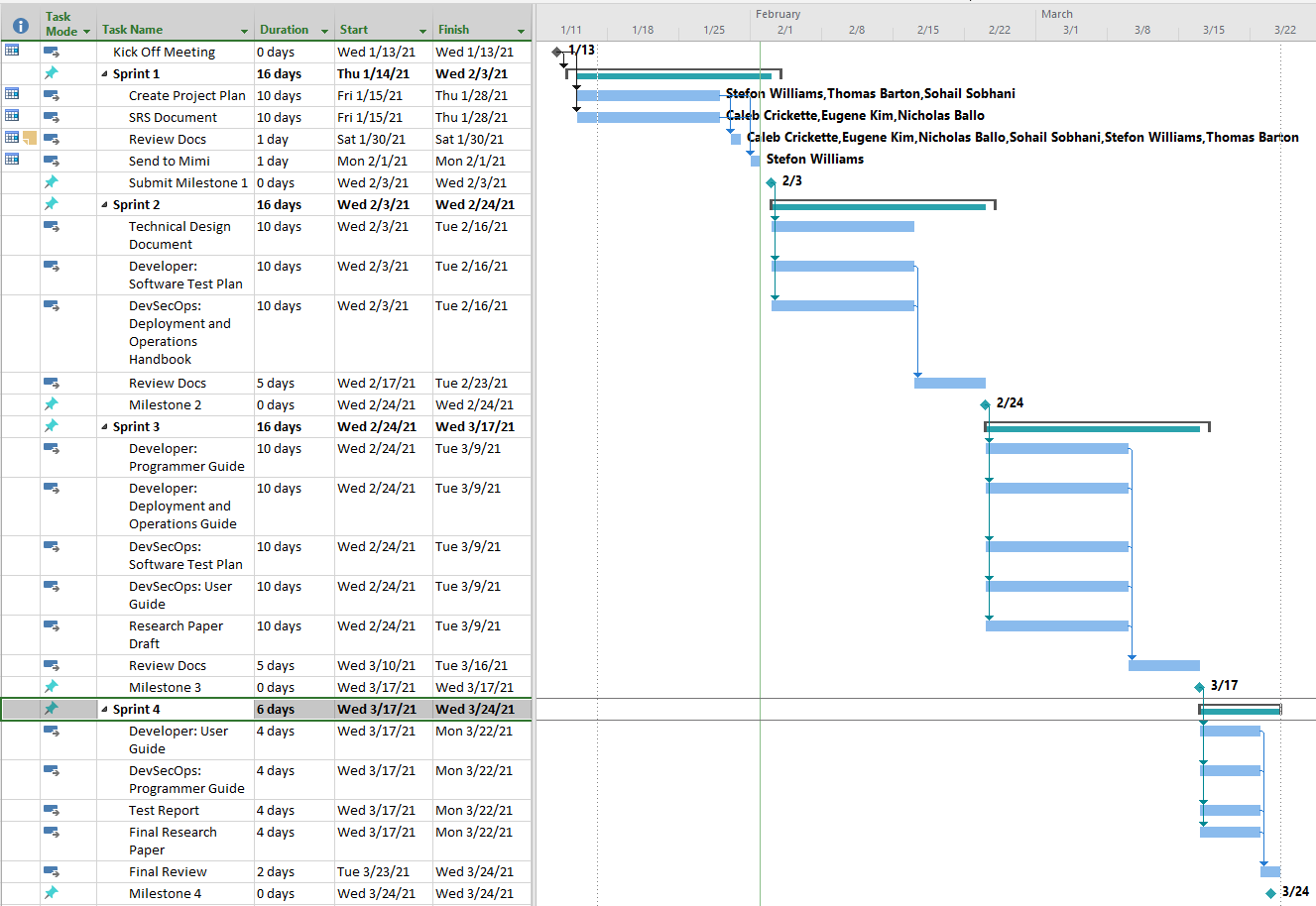
* Short timeline to deliver fully functional code.
* 300-dollar credit (free trial for dialog flow interpretation and webhook use) per team member.
* HIPPA requirements and other proprietary conditions prevent the project team from collecting examples of live data, and can only use mock data for testing purposes.
* Similar to above requirements, consideration needs to be taken to protect sensitive and private data.
* In order for the application to function, a server to run the webhook portions of the application will need to be set up.

# Project Approach

The project team will follow a standard Software Development Lifecycle (SDLC) of plan, design, develop, and implement. Using an agile methodology, the team will divide the work into four sprints. At the end of the final sprint, the team will close out the project with the final deliverables and artifacts being turned over to the project stakeholders.

The agile methodology for this project will consist of four sprints that align with the project milestones. Within a sprint there will be initial sprint planning for that sprint, and as needed or at minimum weekly team check in meetings to ensure that project objectives are being met. A normal staple of agile methodologies is the daily standup, which due to the asynchronous schedule of the team members, daily standups will be omitted. Sprint closeout meetings will also occur reviewing what occurred in that sprint, and the next sprint planning meeting will occur immediately following.

## Project timeline (Gannt Chart)



## Project Work Breakdown

The following Work Breakdown Structure (WBS) will be used as the initial guiding schedule for project development. This WBS in its simplest state specifies the individual milestones and their delivery date. In addition to the milestones, the WBS provides an intended format for the sprints that will be used throughout this project to deliver specific functionality to the stakeholders at the earliest opportunity. This work breakdown structure is detailed in overview below and in a more in-depth manner in Appendix A. The WBS will be maintained and updated throughout the life of this project.

## Sprint 1

This sprint consists of creating key project documents used to outline what the overall project process will entail. The major key documents include the Software Requirements Specification and this document, the project plan. There will also be research conducted to determine the best approach to the project.

## Sprint 2

This is largely a design sprint where details of how to accomplish completing this project are defined. The technical design document, software test plan, and deployment and operations handbook will be created. The project will also likely have code start to be laid down.

## Sprint 3

The bulk of the coding will occur in this sprint (as well as the next) with the previous document designs realized into the start of the ‘physical’ product. A programmer guide, and the start of the user guide will be created.

## Sprint 4

This is the final sprint in which the finished product will be delivered, outstanding documents will be finished and the research paper published.

## Post Agile activities

A short period of polish and wrapup will be conducted to finalize any loose ends after the development and testing completes during the agile development.

# Requirements Management Plan

It is DialogFlow’s responsibility to provide successful implementation of the following requirements: extract name(s) from speech, understand the user’s different intents from speech, extract entities and data from user’s speech, insert extracted entities and data into a database, and ensure service is integrated with mobile application. Due to time constraints all team members will be involved in contributing to ensuring that all requirements are implemented successfully. DevCloud Azure, Github, Dialogflow’s development console will be used in fulfilling the requirements throughout the project lifecycle.

## Pending Functional Requirements

### 1. Use Case: Extract Name(s) from speech.

**Summary:** DialogFlow is used to record information we are interested in using for a report. One of the fields is names, which can be a first name or last name.

### 2. Use Case: Understand the user’s different intents from speech.

**Summary:** DialogFlow is used to recognize the different intents from the user’s speech. Each intent corresponds to a form field that the user wants to capture data for.

### 3. Use Case: Extract entities and data from speech

**Summary:** DialogFlow is used to extract specific entities and data from the speech related to its intent. Each intent corresponds to a form field that the user wants to capture data for. Each entity corresponds to the data content of the form field.

### 4. Use Case: Insert extracted entities and data into a database.

**Summary:** After DialogFlow has extracted entities and relevant form data, the service needs to insert the data into a database so that other teams can take it and update the forms as needed.

### 5. Use Case: Service is integrated with mobile application

**Summary:** DialogFlow will be integrated into the mobile application, thus our service needs to be able to be accessed by the mobile application.

# Integration Management Plan

To ensure that the various elements of the projects are properly coordinated team Dialogflow will abide by the project schedule displayed above. This will make certain that the due dates of all milestone deliverables of the project are aligned and met. Microsoft Teams provides a change log that allows team members to view changes made by fellow team members, which helps measure and monitor the project’s progress in between due dates. Contact will be made regularly with DevSecOps and Mobile teams, especially in the design phase to ensure all the different aspects of the overall project fit together correctly.

# Change Management Plan

Dialogflow’s change control board (Stefon Williams, Eugene Kim, Thomas Barton, Caleb Crickette) will review and authorize any changes that may occur during the SDLC. To ensure that change management is manageable, we will conduct a Teams meeting after receiving any change requests. This process will allow us to evaluate, authorize, and control the change requests. Team Dialogflow will maintain a change log collect and track any changes made during the SDLC within Microsoft Teams. In addition, the Dialogflow team will coordinate a harmonized change management plan with other teams and stakeholders in order to ensure all are aligned with said change. Project Managers from Web team, Mobile team, and Dialogflow team meet weekly on Microsoft Teams for status updates. If possible, change requests occur during the week, said change requests will be approved or unapproved during these weekly meetings. If a change request occurs and needs to be immediately addressed, an impromptu meeting will occur to discuss the change. Team developers meet as needed to discuss changes within the project.

# Quality Management Plan

The processes subject to quality review are the following: extract name(s) from speech, understand the user’s different intents from speech, extract entities and data from user’s speech, insert extracted entities and data into a database, and service is integrated with mobile application. Nicholas Ballo (Test Engineer) will conduct testing of the software during its development to ensure proper functionality. Dialogflow will record specific requirements and check off various quality management and testing activities.

# Test Plan Identifier

# Introduction

## Purpose

This section describes the plans for testing the functional requirements of REST calls for the **Form Scriber – Dialogflow Service.**Additionaly, this section describes the testing approaches that will be used to test the REST calls of the system. The test plan will help identify the testing scope, testing objectives, test responsibilities, test schedule, major milestones, testing approach, suspension and resumption criteria, test deliverables, and resources and environments needed for testing.

## Test Scope

This test plan will be used in testing the application's input and output in terms of meeting the expected results of the in-scope testing requirements. The following details the in-scope and out-of-scope testing. There will be an additional overall testing effort for integration testing which will require coordination with the mobie team.

### *In-Scope*

The scope of this test plan is to test the REST services of the **"Form Scriber- Dialogflow Service",** as described in the SRS. The following testing methods will be implemented: Unit Testing, Integration Testing, and Overall Integration Testing.

***Unit Testing***which involves testing a program unit independently. Developers of the RESTful web services will independently test each of their APIs, as well as independently across teams. Similarly, developers of the User Interface will also independently test their web pages. All these unit tests need to be completed prior to the next testing method: *Integration Testing*.

***Integration Testing***will involve testing all REST services in cooperation with the GCP Dialogflow service. Therefore, this occurs when the work of all REST services and setup of the Dialogflow service chat agents are completed. This will utilize GCP’s assistant service to fully test the system.

***End-to-End (E2E) Testing****.* At the time of this document’s delivery, a detailed test plan for **Form Scriber – Dialogflow Service** as it interacts with the mobile teams deliverables are not yet identified. However, it is worth-noting in this document of the intent to perform such testing since the data inputs of our application will depend on the requests of another team. Similarly, the mobile team will rely on the services that our team will provide. As the project evolves and gets completed, test cases for this particular testing method are to be determined as the process of identifying them will need further coordination with the mobile team .

### *Out-of-Scope*

Out-of-scope testing activities for the **Form Scriber – Dialogflow Service** include performance testing, stress testing, system testing, and user acceptance testing.

## Test Objective

The objectives of the testing of the **"Form Scriber Project"** are:

* Check to see if the intents within Dialogflow are functioning as expected.
* Ensure that all REST calls of the software are correctly working, and these REST calls' responses are as intended.
* Identify defects in the REST calls and be reported for correction.

Intended Audience

The intented audience of this test plan include the project manager, developers, testers, and other team members who will participate in the REST calls testing of the **Form Scriber – Dialogflow Service.**

## Roles and Responsibilities

Roles and responsibilities of different team members of the testing team are:

***Table 2:****Roles and Responsibilities*

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Nicholas Ballo | Test Manager | Ensure proper planning and management of test resources.  Assess the progress and effectiveness of the testing effort.  Support the appropriate level of quality with the resolution of significant defects. |
| Eugene Kim  Stefon Williams | Test Analyst | Identify and define the necessary tests.  Detailed monitoring of the tests.  Collection and management of test data in each test cycle.  Overall quality assessment as a result of testing activity. |
| Sohail Sobhani  Abdul Kamara  Anthony Cedeno | Test Analysts / Tester | Perform responsibilities of a Test Analyst  Responsible for executing test cases.  Identify and report REST calls defects within the software.  Follow-up test resolutions and subsequent verification. |
| Caleb Crickette  Thomas Barton  Sohail Sobhani  Abdul Kamara  Geoffrey Dean | Developer | Will be responsible for fixing the identified UI and REST calls defects of the system. |

## Testing Schedule

The schedule of the testing is:

***Table 3:****Testing Schedule*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task No. | Task Activity | Start Date | End Date | Duration |
| 1 | REST Calls Testing | 3/17/2021 | 03/21/2021 | 5 days |
| 2 | Dialogflow Assistant Testing | 3/22/2021 | 3/26/2021 | 5 days |
| 3 | E2E Integration Testing | 3/27/2021 | 3/30/2021 | 3 days |

## Key Milestones

The key milestones of the testing activities will be:

***Table 3:*** *Key Milestones of Testing*

|  |  |  |
| --- | --- | --- |
| Task No. | Task Activity | Milestone |
| 1 | REST Calls Testing | Completed REST calls testing |
| 2 | Dialogflow Assistant Testing | Completed Dialogflow Assistant Testing |
| 3 | E2E Integration Testing | Completed E2E Testing |

## Testing Approach

As defined in the test scope, the following type of tests for the Form Scriber – Dialogflow Service will be performed:

* **REST Calls Testing**
  + The REST calls of the Form Scriber – Dialogflow Service will be tested to ensure all features are implemented in the system and working correctly, and responses of these REST calls are as intended.
* **Dialogflow Assistant Testing**
  + The Dialogflow agent will be tested using GCP’s assistant service to ensure that the system works as intended.
* **E2E Integration Testing**
  + An E2E test will be performed to ensure that the entire system from all teams are integrated appropriately and any issues or defects are addressed.

## Test Automation

Test automation tools will be used to test the REST calls and Dialogflow agents. The Fiddler tool will be used for automation testing of REST calls and responses. GCP has an automated test API that we can leverage to test the chat agents.

## Functional Requirements of Test Cases

1.  Isolate primary user’s voice for use within the service

2.  Dialogflow service is integrated within the mobile application

3.  Receive report identifier from user5.  Process report contents and fields.

4.  Process report contents and fields.

5. Generate dynamic intents to capture report fields.

6.  Identify report contents extracted by the Dialogflow agent.

Test Plan Metrics

These test metrics will be used to indicate the overall success status of the testing effort. The testing metrics are as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Cases Planned** | **Test Cases Executed** | **Test Cases Passed** | **Test Cases Failed** |
| 6 | 0 | 0 | 0 |

Statistics: This is a list of percentages that will visualize key project benchmarks.

Executed Tests (number of tests executed/planned): 0%

Passed Tests (number of passed tests/executed): 0%

Failed Tests (number of failed tests/executed): 0%

Fixed Defect percentage (Defects Fixed/Reported): 0%

Defects Deferred Percentage (Defects Deferred/Total Reported Defects): 0%

# Test Cases

## **Test Case 01**: Isolate primary user’s voice for use within the service

|  |  |
| --- | --- |
| **Description:** | Discern primary user’s voice as the input to the Dialogflow service. |
| **Requirements:** | The user initiates a request to record their voice as an input to the Dialogflow service through the Google Assistant. Only the primary user’s voice will be recognized and sent to the service for voice recognition. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide. |
| **Trigger** | The user will select a record button to record their voice or use a prebuilt voice trigger (i.e., Form Scriber do this) to recognize the user. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access |
| **System Response** | The system should recognize the user’s voice and pass to Dialogflow |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message. |
| **Alternate Path** | If Dialogflow does not understand or misinterprets the user, the system will prompt the user to try again. |

## **Test Case 02**: Dialogflow service is integrated within the mobile application

|  |  |
| --- | --- |
| **Description:** | DialogFlow engine can be used and accessed by the mobile application through an accessible API. |
| **Requirements:** | Dialogflow will provide an Assistants API to embed into the mobile application. This will allow the mobile team to access our services using this API. This also allows anyone to access the service via API if they choose to. |
| **Actor:** | The mobile app will embed the Assistants service. |
| **Trigger** | Mobile app startup. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access |
| **System Response** | The user should see the Google Assistant page when they go to the corresponding mobile page that embeds the API. |
| **Post Condition** | The user will be able to interact and speak with the assistant. |
| **Alternate Path** | If there is an error with the embedding of the API, the mobile team will display an error message indicating that it is unavailable. If there is an error with the API itself, an error message will appear from GCP. |

## **Test Case 03**: Receive report identifier from user

|  |  |
| --- | --- |
| **Description:** | Identify report identifier from user to receive report metadata. This could be an id or unique name. |
| **Requirements:** | The user will initiate a voice command to retrieve a specific report by id or name. For example, the user may say “Form Scriber, I want to fill out the new patient form”. The system will then retrieve the report metadata for the new patient form. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice command to fill out a specific report name. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access |
| **System Response** | The system will respond with the field names to fill out and ask for user confirmation. |
| **Post Condition** | If the user confirms, the system will respond and continue to listen to the user’s report field triggers. |
| **Alternate Path** | If there are multiple reports that match the query, the system will display the first available report. If the user denies, it will confirm the next available report. It will continue this pattern until the user confirms a report or does not confirm.  If the user does not confirm a report metadata and there are no other reports left to query, the system will respond to the user saying there is no available reports and to create the form on the web page. |

## **Test Case 04**: Process report contents and fields.

|  |  |
| --- | --- |
| **Description:** | Process report fields and content into internal data structure. |
| **Requirements:** | The user initiates a request to create a report and triggers voice commands for the report contents. The system will need to capture and store this information in a meaningful structure to send to other components in the system. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice trigger to fill out a specific report. They will then continue to say voice triggers to fill out the report contents. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access * Confirmation of user’s intent to create report * Confirmation of user’s intent to create report fields’ content |
| **System Response** | The system will respond with a success message. |
| **Post Condition** | The system will store the report ID, user, and report contents based on the recognized intents from Dialogflow. |
| **Alternate Path** | N/A |

## **Test Case 05**: Generate dynamic intents to capture report fields

|  |  |
| --- | --- |
| **Description:** | Use retrieved form fields to generate dynamic intents in Dialogflow. |
| **Requirements:** | The user initiates a request to create a report and triggers voice commands for the report contents. The system will need to capture the report metadata and dynamically generate the intents in Dialogflow to capture the relevant report fields. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice trigger to fill out a specific report. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access * Confirmation of user’s intent to create report |
| **System Response** | The system will respond that it is ready to accept report content. |
| **Post Condition** | The system will retrieve the report metadata and use GCP’s Dialogflow API to create generic intents that capture any of the report fields as keywords and store any other information that comes after the keywords. For example, if a report field requires ‘heart rate’, our service will generate an intent that looks for the key word heart rate. To capture data, the user will need to say ‘heart rate 60 bpm’, where the keyword is ‘heart rate’ and the value is ‘60 bpm’ |
| **Alternate Path** | **If an intent already exists due to a previous report, a new intent will not overwrite the existing. The system will use the existing intent.** |
|  |  |

## **Test Case 07**: Identify report contents extracted by the Dialogflow agent

|  |  |
| --- | --- |
| **Description:** | Identify report contents that are recognized from the user’s voice and dynamic intents using Dialogflow. |
| **Requirements:** | The user initiates a request to create a report and triggers voice commands for the report contents. The report contents are sent to our webhook service after they have extracted by the Dialogflow Agent. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice trigger to fill out a specific report. They will then continue to say voice triggers to fill out the report contents. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access * Confirmation of user’s intent to create report * Confirmation of user’s intent to create report fields’ content |
| **System Response** | The system will respond with a confirmation of the user’s voice recognition for the triggered report field. |
| **Post Condition** | If the user confirms, the system will store the report field as metadata. |
| **Alternate Path** | If the system does not understand the key word as a valid intent, it will ask the user to try again. |

## **Test Case 06**: Dialogflow Help Support Response

|  |  |
| --- | --- |
| **Description:** | Dialogflow should properly handle support requests from the end user and provide additional information accurately and properly within a conversational flow. |
| **Requirements:** | User requests support or help within the dialogflow conversation. Dialogflow shall respond accordingly and appropriately offering help and support to the user. |
| **Actor:** | User shall give a voice command asking with intent for support and help. |
| **Precondition:** | User has Google Assistant open and has a proper device such a smartphone or computer. Additionally, they need to be interfaced with Form Scriber by invoking the “Talk to Form Scriber dotcom” invocation. |
| **System Response** | The system will respond with a description on where to find help and help to steer the user through a support and how-to conversational flow. |
| **Post Condition** | The user receives appropriate help and link to the manuals and guides with additional information on how to engage and use the application. |
| **Alternate Path** | If the user mistakenly invokes the help & support intent, they can tell the bot to stop or return to the beginning to start a new conversational flow. |

# Testing Criteria

## Suspension Criteria

Formscriber’s suspension criteria for testing shall be guided by those itemized by (onestoptesting.com, n.d.): in an article entitled *Suspension criteria & resumption requirements*. All or a portion of testing shall be suspended if any of the following are met:

* *External dependent system unavailability*. For instance, testing shall be suspended if Formscriber’s required Google Cloud Services, while may be unlikely to happen, suddenly become available. In terms of End-to-End testing, this criterion is met if either the mobile app or the Formscriber service is unavailable.
* *When an identified defect cannot allow succeeding tests*. For example, in testing Formscriber’s web services, if testing for creating a session fails, succeeding tests dependent on sessions may not be worthwhile to continue because a session is crucial especially in identifying further actions (or REST calls) of a current user.
* *Contractual obligations are not met particularly in meeting a deadline*. In the academic setting, this could mean that the deadline has passed late submissions are unacceptable.

## Resumption Criteria

Similar to suspension criteria, Formscriber’s resumption criteria shall be guided by the aforementioned article. In essence, these criteria are corrective actions of each aforementioned suspension criterion. Hence, all testing shall resume under any of the following circumstances:

* Downtime of dependent system ceases and resumes normal operation.
* A fix is implemented to a defect and all test cases dependent on its related use case can possibly proceed testing.
* An extension of the delivery of the final product is established.

## Test Completeness

Testing will be considered complete when the following conditions have been met:

* When the Developers and Testing Team agree that the system meets all the UI requirements and REST calls testing, the system has been completed.
* All test cases have been executed.
* All priority 1, 2, and 3 defects have been closed after successful resolution.
* The test manager has accepted the testing and signed off testing as completed.

## Test Deliverables

During the testing of the project, the following specific deliverables will be provided:

* **Test Plan**
  + Test plan document, which is this document. It describes the testing objectives, test scope, resources, deliverables, estimations, roles and responsibilities, test environments.
* **Test Cases Specification**
  + This document will provide the test case specifications for all UIs and REST calls of the system. Each test case will:
    - Describes the objective of that test.
    - Specify the inputs for testing and expected outputs.
    - Provides step-by-step procedures for executing the test.
    - Outlines the pass/fail criteria of the test.
* **Test Case Report**
  + This document will present the results of the test case executions.
* **Defects Reports**
  + Defects reports will store the information about the identified defects and resolutions.
* **Customer Sign Off**
  + This is the document that the customer of this project will sign to validate that system meets all the REST calls requirements of the system.

# Resource & Environment Needs

## Testing Tools

***Table 4: Testing*** *Tools*

|  |  |
| --- | --- |
| Process | Tool |
| Test Case Specification | MS Office |
| Test Case Tracking | MS Excel |
| REST Calls Testing | Fiddler |
| Dialogflow Agent Testing | GCP Assistants API |
| Defect Management | MS Word |
| Test Reporting | MS Word |

 Test Environment

**Hardware:**

Laptop or PC with the following minimum specs:

* Intel Core i5
* 8GB RAM:
* 2.0 GHz Processor

**Software:**

The following **software**will be required for testing the system:

* **Windows 8:** Chrome (latest version)
* **Mac OS X:** Chrome (latest version)
* **Fiddler:**(latest version)
* **GCP Console**
  + **Dialogflow Service**
  + **Google Assistant**

# Resource Management Plan

The project team’s organization is shown in the table below.

Table 3 - Project Team

|  |  |
| --- | --- |
| Name | Role |
| Stefon Williams | Project Manager |
| Eugene Kim | Business Analyst |
| Caleb Crickette | Lead Developer/DevSecOps |
| Thomas Barton | Developer |
| Sohail Sobhani | Developer |
| Abdul Kamara | Developer |
| Nicholas Ballo | Test Engineer |

The outside resources that will contribute to the success are listed in the table below.

Table 4 - External Resources

|  |  |
| --- | --- |
| Name | Role |
| Dr. Mir Assadullah | Professor and Advisor |
| Roy Gordon | PM and Analyst Advisor |
| John Lockhart | Developer and DevSecOps Advisor |
| Menen (Mimi) Fantay | Lead Project Manager |
| Vincent Leung | DevSecOps |
| Ivy Pham | DevSecOps |

## Project Team Roles and Responsibilities

Project Manager: The project manager role will be responsible for overseeing the schedule and pace of the project, ensuring that the deliverables are on track, and that they are delivered to the appropriate persons.

Business Analyst: Will be responsible for determining the resources needed for the project to be successful, how to acquire said resources, and will recommend how to allocate the resources.

Lead Developer: Development lead will be responsible for guiding the overall program process according to the plans laid out in the project plan and that the requirements are satisfied adequately.

Developer: Developers will be directly responsible for implementing features to satisfy the requirements satisfactorily by any means available and appropriate. Resource usage shall be tracked, and techniques shall conform to best practices for the industry.

Test Engineer: Shall put together the test plans as well as implement some automated testing functionality to ensure the proper function of the end product based on the requirements.

## Responsibility Assigned Matrix (RACI)

The matrix below displays the responsibilities of each Dialogflow team member during the SDLC. All team members have assigned tasks but constant collaboration will occur due to the lack of resources. The four roles that are identified include the following: Responsible, Accountable, Consulted, Informed. Responsible persons must complete the task or objective or make the decision. The accountable person must sign or approve when the task or objective is complete. Success requires that there is only one person Accountable. Consulted persons need to provide input, remain in the loop, and actively participate. Lastly, Informed persons should be constantly updated on progress or decisions.

Chart

Description automatically generated

# Risk Management Plan

Risk mitigation occurs in five phases, identification, analyzation, prioritization, mitigation, and monitoring. Risk management occurs throughout the project management, and would normally be undertaken by a risk management officer, however due to the short time frame of this project, each member of each team will need to be aware of the risks and alert the team of new risks as they arise as well as when risk levels change due to evolving circumstances.

Risks will have two estimates associated with them, likelihood and severity of impact, both on a five-point scale of low, medium-low, medium, medium-high, and high. Likelihood is defined by how likely a risk is impacting the project over the entire project lifecycle by rough estimate, and severity is how much impact a risk would have if it were to come into play. Low likelihood means that a risk is unlikely to impact the project delivery, and high likelihood means that a risk is very likely (if not certain) to impact the project, where medium has around 50% chance of occurring. Severity is a general estimate of how many hours and/or delay will occur to mitigate the risk if it occurs or otherwise impact to the functionality overall, with low severity being little to no delay with some resource shifting to high severity being causing enough of a delay in the project that it would cause some requirements to not be completed by the project deadline, if not having the project fail on occurrence.

The overall risk association is the interaction of both the likelihood and severity, on a three point scale that gives an overall idea of where the risk would fall for prioritization for the project.

## Risk Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| High Likelihood |  |  |  | 1 | 1 |
| Medium-High Likelihood |  | 1 | 1 |  |  |
| Medium Likelihood |  |  | 2 |  | 1 |
| Medium-Low  Likelihood |  |  |  | 1 | 1 |
| Low Likelihood |  |  | 1 |  | 2 |
|  | Low severity | Medium-Low  severity | Medium severity | Medium-high severity | High severity |

## Risk Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Risk** | **Severity** | **Likelihood** | **Overall** |
| RS-1 | The short timeframe will likely cause various aspects of the project to be developed hastily. | Medium-high | High | High |
| RS-2 | There may be bugs that will cause the project to be restructured in some way, causing further delay. | High | High | High |
| RS-3 | Individual team members may become unavailable at certain points due to unforeseen circumstances (sickness, family issues, etc.) | High | Medium-low | High |
| RS-4 | Unfamiliar environments may cause delays in the form of developers needing to learn the environment. | Medium | Medium-high | Medium |
| RS-5 | The different authentication methods and encryption protocols in the different applications can stop working when deployed to production servers causing operations to fail. | Medium-high | Medium-low | Medium |
| RS-6 | Certain accounts or subscriptions may run out or be inaccessible which the project is dependent on to operate. | High | Low | Medium |
| RS-7 | Lack of experience or skill with certain new technologies which the architecture will require. | Medium | Medium | Medium |
| RS-8 | Scope Creep of Project Goals | Medium | Low | Low |
| RS-9 | Misinterpreted Requirements | Medium- high | Medium-low | Medium |
| RS-10 | Lack of Requirements Details | Medium | Medium | Medium |
| RS-11 | Breakdown of Team Communication | High | Low | Medium |
| RS-12 | Insufficient testing and testing resources | High | Medium | High |

## Risk Mitigation Strategies.

Risk mitigation strategies will take the form of any combination of prevention, mitigation, and acceptance, with a tie into all risks being monitored throughout the project lifecycle at a minimum. Prevention will aim to outright stop the issue from occurring in the first place, mitigation will lessen the impact of the risk, and acceptance will be simply acknowledging the risk occurs and making adjustments to the project to work around the impact.

The risks are addressed in the order that they appear within the table above

* RS-1: The short timeframe of the project cannot be avoided; the timeline is fixed so the only way to address this is to mitigate via adequate planning.
* RS-2: Bugs making their way into the code is nearly certain, the only way to address this is to have robust testing and fixing the bugs as quickly as possible.
* RS-3: Individual team members will become unavailable throughout the project for some reason or another, this must largely be an accepted risk, where the only way to address it is to balance out our resources for the project.
* RS-4: Unfamiliar development environment can be addressed partly by sharing resources for learning the environments, however it will largely be on each individual team member to learn what they need in order to develop the project.
* RS-5: This risk largely falls within bugs causing the project to be restructured, though this problem would be easier to identify and somewhat easier to resolve, hopefully by only changing the authentication methods to be compatible.
* RS-6: Account subscriptions running out or being inaccessible is something that can only be addressed when it happens, as if that were to happen there are no backup options aside from finding a completely different solution or attempting to recover the solution in some way, either of which would be detrimental to the project if not an automatic failure state depending on the phase in which this would occur. Thankfully the chances of this happening are low.
* RS-7: Any other lack of experience would need to be addressed in the same fashion as the unfamiliar development environment.
* RS-8: Scope creep can be addressed by sufficient monitoring of efforts and analysis of evolving requirements.
* RS-9: Requirement's misinterpretation is something that can only really be mitigated by ensuring understanding and checking against the requirements to ensure they are actually being fulfilled, there is no outside customer to ensure that the requirements are what they need to be.
* RS-10: A lack of requirements details can occur within the project which will cause additional time needed on clarification and delay in development.
* RS-11: A breakdown of team communication would be highly detrimental to the overall project workflow as there are three teams that need to be in contact with each other. The mitigation would be ensuring that the teams have access to alternate communication methods, as well as offering channels for communication remediation in the event that the communication breakdown is due to disagreement.
* RS-12: As mentioned in previous risks, the testing resources are needed to ensure an adequate working product, and that bugs are not present. If testing resources are not present, they will need to be acquired no matter what.

# Communication Management Plan

All communication will be facilitated by the use of Microsoft Teams and various chat rooms set up for those purposes. Other forms of communication may be used unofficially, however Microsoft Teams and Office will be the primary form of communication used for the purposes of the project. Inter-team communications will formally be handled by the PMs, while unofficial communication can occur between any team members. The lead developer shall also communicate with other teams regarding various approaches to the project solution on an as-needed basis. Internal team communication will consist of individual team member updates, and informal team member communication for collaboration. Formal internal communication shall occur using Microsoft team chats and any other form of requested team communication.

Project Managers from the Web team, Mobile team, and Dialogflow team will meet weekly on Microsoft Teams to discuss the project and to provide status updates. Developers will keep constant communication through the chat function within Microsoft Teams and conduct meeting as needed.

# Test Plan Identifier

# Introduction

## Purpose

This section describes the plans for testing the functional requirements of REST calls for the **Form Scriber – Dialogflow Service.**Additionaly, this section describes the testing approaches that will be used to test the REST calls of the system. The test plan will help identify the testing scope, testing objectives, test responsibilities, test schedule, major milestones, testing approach, suspension and resumption criteria, test deliverables, and resources and environments needed for testing.

## Test Scope

This test plan will be used in testing the application's input and output in terms of meeting the expected results of the in-scope testing requirements. The following details the in-scope and out-of-scope testing. There will be an additional overall testing effort for integration testing which will require coordination with the mobie team.

### *In-Scope*

The scope of this test plan is to test the REST services of the **"Form Scriber- Dialogflow Service",** as described in the SRS. The following testing methods will be implemented: Unit Testing, Integration Testing, and Overall Integration Testing.

***Unit Testing***which involves testing a program unit independently. Developers of the RESTful web services will independently test each of their APIs, as well as independently across teams. Similarly, developers of the User Interface will also independently test their web pages. All these unit tests need to be completed prior to the next testing method: *Integration Testing*.

***Integration Testing***will involve testing all REST services in cooperation with the GCP Dialogflow service. Therefore, this occurs when the work of all REST services and setup of the Dialogflow service chat agents are completed. This will utilize GCP’s assistant service to fully test the system.

***End-to-End (E2E) Testing****.* At the time of this document’s delivery, a detailed test plan for **Form Scriber – Dialogflow Service** as it interacts with the mobile teams deliverables are not yet identified. However, it is worth-noting in this document of the intent to perform such testing since the data inputs of our application will depend on the requests of another team. Similarly, the mobile team will rely on the services that our team will provide. As the project evolves and gets completed, test cases for this particular testing method are to be determined as the process of identifying them will need further coordination with the mobile team .

### *Out-of-Scope*

Out-of-scope testing activities for the **Form Scriber – Dialogflow Service** include performance testing, stress testing, system testing, and user acceptance testing.

## Test Objective

The objectives of the testing of the **"Form Scriber Project"** are:

* Check to see if the intents within Dialogflow are functioning as expected.
* Ensure that all REST calls of the software are correctly working, and these REST calls' responses are as intended.
* Identify defects in the REST calls and be reported for correction.

Intended Audience

The intented audience of this test plan include the project manager, developers, testers, and other team members who will participate in the REST calls testing of the **Form Scriber – Dialogflow Service.**

## Roles and Responsibilities

Roles and responsibilities of different team members of the testing team are:

***Table 2:****Roles and Responsibilities*

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Nicholas Ballo | Test Manager | Ensure proper planning and management of test resources.  Assess the progress and effectiveness of the testing effort.  Support the appropriate level of quality with the resolution of significant defects. |
| Eugene Kim  Stefon Williams | Test Analyst | Identify and define the necessary tests.  Detailed monitoring of the tests.  Collection and management of test data in each test cycle.  Overall quality assessment as a result of testing activity. |
| Sohail Sobhani  Abdul Kamara  Anthony Cedeno | Test Analysts / Tester | Perform responsibilities of a Test Analyst  Responsible for executing test cases.  Identify and report REST calls defects within the software.  Follow-up test resolutions and subsequent verification. |
| Caleb Crickette  Thomas Barton  Sohail Sobhani  Abdul Kamara | Developer | Will be responsible for fixing the identified UI and REST calls defects of the system. |

## Testing Schedule

The schedule of the testing is:

***Table 3:****Testing Schedule*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task No. | Task Activity | Start Date | End Date | Duration |
| 1 | REST Calls Testing | 3/17/2021 | 03/21/2021 | 5 days |
| 2 | Dialogflow Assistant Testing | 3/22/2021 | 3/26/2021 | 5 days |
| 3 | E2E Integration Testing | 3/27/2021 | 3/30/2021 | 3 days |

## Key Milestones

The key milestones of the testing activities will be:

***Table 3:*** *Key Milestones of Testing*

|  |  |  |
| --- | --- | --- |
| Task No. | Task Activity | Milestone |
| 1 | REST Calls Testing | Completed REST calls testing |
| 2 | Dialogflow Assistant Testing | Completed Dialogflow Assistant Testing |
| 3 | E2E Integration Testing | Completed E2E Testing |

## Testing Approach

As defined in the test scope, the following type of tests for the Form Scriber – Dialogflow Service will be performed:

* **REST Calls Testing**
  + The REST calls of the Form Scriber – Dialogflow Service will be tested to ensure all features are implemented in the system and working correctly, and responses of these REST calls are as intended.
* **Dialogflow Assistant Testing**
  + The Dialogflow agent will be tested using GCP’s assistant service to ensure that the system works as intended.
* **E2E Integration Testing**
  + An E2E test will be performed to ensure that the entire system from all teams are integrated appropriately and any issues or defects are addressed.

## Test Automation

Test automation tools will be used to test the REST calls and Dialogflow agents. The Fiddler tool will be used for automation testing of REST calls and responses. GCP has an automated test API that we can leverage to test the chat agents.

## Functional Requirements of Test Cases

1.  Isolate primary user’s voice for use within the service

2.  Dialogflow service is integrated within the mobile application

3.  Receive report identifier from user

4.  Get report fields from database

5.  Process report contents and fields.

6.  Generate dynamic intents to capture report fields.

7.  Identify report contents extracted by the Dialogflow agent.

8.  Store extracted report contents into a database.

9.  Initiate connection with database

## Test Plan Metrics

These test metrics will be used to indicate the overall success status of the testing effort. The testing metrics are as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Cases Planned** | **Test Cases Executed** | **Test Cases Passed** | **Test Cases Failed** |
| 10 | 0 | 0 | 0 |

Statistics: This is a list of percentages that will visualize key project benchmarks.

Executed Tests (number of tests executed/planned): 0%

Passed Tests (number of passed tests/executed): 0%

Failed Tests (number of failed tests/executed): 0%

Fixed Defect percentage (Defects Fixed/Reported): 0%

Defects Deferred Percentage (Defects Deferred/Total Reported Defects): 0%

# Test Cases

## **Test Case 01**: Isolate primary user’s voice for use within the service

|  |  |
| --- | --- |
| **Description:** | Discern primary user’s voice as the input to the Dialogflow service. |
| **Requirements:** | The user initiates a request to record their voice as an input to the Dialogflow service through the Google Assistant. Only the primary user’s voice will be recognized and sent to the service for voice recognition. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide. |
| **Trigger** | The user will select a record button to record their voice or use a prebuilt voice trigger (i.e., Form Scriber do this) to recognize the user. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access |
| **System Response** | The system should recognize the user’s voice and pass to Dialogflow |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message. |
| **Alternate Path** | If Dialogflow does not understand or misinterprets the user, the system will prompt the user to try again. |

## **Test Case 02**: Dialogflow service is integrated within the mobile application

|  |  |
| --- | --- |
| **Description:** | DialogFlow engine can be used and accessed by the mobile application through an accessible API. |
| **Requirements:** | Dialogflow will provide an Assistants API to embed into the mobile application. This will allow the mobile team to access our services using this API. This also allows anyone to access the service via API if they choose to. |
| **Actor:** | The mobile app will embed the Assistants service. |
| **Trigger** | Mobile app startup. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access |
| **System Response** | The user should see the Google Assistant page when they go to the corresponding mobile page that embeds the API. |
| **Post Condition** | The user will be able to interact and speak with the assistant. |
| **Alternate Path** | If there is an error with the embedding of the API, the mobile team will display an error message indicating that it is unavailable. If there is an error with the API itself, an error message will appear from GCP. |

## **Test Case 03**: Receive report identifier from user

|  |  |
| --- | --- |
| **Description:** | Identify report identifier from user to receive report metadata. This could be an id or unique name. |
| **Requirements:** | The user will initiate a voice command to retrieve a specific report by id or name. For example, the user may say “Form Scriber, I want to fill out the new patient form”. The system will then retrieve the report metadata for the new patient form. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice command to fill out a specific report name. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access |
| **System Response** | The system will respond with the field names to fill out and ask for user confirmation. |
| **Post Condition** | If the user confirms, the system will respond and continue to listen to the user’s report field triggers. |
| **Alternate Path** | If there are multiple reports that match the query, the system will display the first available report. If the user denies, it will confirm the next available report. It will continue this pattern until the user confirms a report or does not confirm.  If the user does not confirm a report metadata and there are no other reports left to query, the system will respond to the user saying there is no available reports and to create the form on the web page. |



## **Test Case 05**: Process report contents and fields.

|  |  |
| --- | --- |
| **Description:** | Process report fields and content into internal data structure. |
| **Requirements:** | The user initiates a request to create a report and triggers voice commands for the report contents. The system will need to capture and store this information in a meaningful structure to send to other components in the system. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice trigger to fill out a specific report. They will then continue to say voice triggers to fill out the report contents. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access * Confirmation of user’s intent to create report * Confirmation of user’s intent to create report fields’ content |
| **System Response** | The system will respond with a success message. |
| **Post Condition** | The system will store the report ID, user, and report contents based on the recognized intents from Dialogflow. |
| **Alternate Path** | N/A |

## **Test Case 06**: Generate dynamic intents to capture report fields

|  |  |
| --- | --- |
| **Description:** | Use retrieved form fields to generate dynamic intents in Dialogflow. |
| **Requirements:** | The user initiates a request to create a report and triggers voice commands for the report contents. The system will need to capture the report metadata and dynamically generate the intents in Dialogflow to capture the relevant report fields. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice trigger to fill out a specific report. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access * Confirmation of user’s intent to create report |
| **System Response** | The system will respond that it is ready to accept report content. |
| **Post Condition** | The system will retrieve the report metadata and use GCP’s Dialogflow API to create generic intents that capture any of the report fields as keywords and store any other information that comes after the keywords. For example, if a report field requires ‘heart rate’, our service will generate an intent that looks for the key word heart rate. To capture data, the user will need to say ‘heart rate 60 bpm’, where the keyword is ‘heart rate’ and the value is ‘60 bpm’ |
| **Alternate Path** | **If an intent already exists due to a previous report, a new intent will not overwrite the existing. The system will use the existing intent.** |
|  |  |

## **Test Case 07**: Identify report contents extracted by the Dialogflow agent

|  |  |
| --- | --- |
| **Description:** | Identify report contents that are recognized from the user’s voice and dynamic intents using Dialogflow. |
| **Requirements:** | The user initiates a request to create a report and triggers voice commands for the report contents. The report contents are sent to our webhook service after they have extracted by the Dialogflow Agent. |
| **Actor:** | The mobile app should perform the user’s request using the Assistant API that our team will provide, which will send it to Dialogflow and our webhook service. |
| **Trigger** | The user will say a voice trigger to fill out a specific report. They will then continue to say voice triggers to fill out the report contents. |
| **Precondition:** | The system should have:   * App installed with the proper OS version. * Voice recording capability * Internet Access * Confirmation of user’s intent to create report * Confirmation of user’s intent to create report fields’ content |
| **System Response** | The system will respond with a confirmation of the user’s voice recognition for the triggered report field. |
| **Post Condition** | If the user confirms, the system will store the report field as metadata. |
| **Alternate Path** | If the system does not understand the key word as a valid intent, it will ask the user to try again. |







## **Test Case 10**: Dialogflow Help Support Response

|  |  |
| --- | --- |
| **Description:** | Dialogflow should properly handle support requests from the end user and provide additional information accurately and properly within a conversational flow. |
| **Requirements:** | User requests support or help within the dialogflow conversation. Dialogflow shall respond accordingly and appropriately offering help and support to the user. |
| **Actor:** | User shall give a voice command asking with intent for support and help. |
| **Precondition:** | User has Google Assistant open and has a proper device such a smartphone or computer. Additionally, they need to be interfaced with Form Scriber by invoking the “Talk to Form Scriber dotcom” invocation. |
| **System Response** | The system will respond with a description on where to find help and help to steer the user through a support and how-to conversational flow. |
| **Post Condition** | The user receives appropriate help and link to the manuals and guides with additional information on how to engage and use the application. |
| **Alternate Path** | If the user mistakenly invokes the help & support intent, they can tell the bot to stop or return to the beginning to start a new conversational flow. |

# Testing Criteria

## Suspension Criteria

Formscriber’s suspension criteria for testing shall be guided by those itemized by (onestoptesting.com, n.d.): in an article entitled *Suspension criteria & resumption requirements*. All or a portion of testing shall be suspended if any of the following are met:

* *External dependent system unavailability*. For instance, testing shall be suspended if Formscriber’s required Google Cloud Services, while may be unlikely to happen, suddenly become available. In terms of End-to-End testing, this criterion is met if either the mobile app or the Formscriber service is unavailable.
* *When an identified defect cannot allow succeeding tests*. For example, in testing Formscriber’s web services, if testing for creating a session fails, succeeding tests dependent on sessions may not be worthwhile to continue because a session is crucial especially in identifying further actions (or REST calls) of a current user.
* *Contractual obligations are not met particularly in meeting a deadline*. In the academic setting, this could mean that the deadline has passed late submissions are unacceptable.

## Resumption Criteria

Similar to suspension criteria, Formscriber’s resumption criteria shall be guided by the aforementioned article. In essence, these criteria are corrective actions of each aforementioned suspension criterion. Hence, all testing shall resume under any of the following circumstances:

* Downtime of dependent system ceases and resumes normal operation.
* A fix is implemented to a defect and all test cases dependent on its related use case can possibly proceed testing.
* An extension of the delivery of the final product is established.

## Test Completeness

Testing will be considered complete when the following conditions have been met:

* When the Developers and Testing Team agree that the system meets all the UI requirements and REST calls testing, the system has been completed.
* All test cases have been executed.
* All priority 1, 2, and 3 defects have been closed after successful resolution.
* The test manager has accepted the testing and signed off testing as completed.

## Test Deliverables

During the testing of the project, the following specific deliverables will be provided:

* **Test Plan**
  + Test plan document, which is this document. It describes the testing objectives, test scope, resources, deliverables, estimations, roles and responsibilities, test environments.
* **Test Cases Specification**
  + This document will provide the test case specifications for all UIs and REST calls of the system. Each test case will:
    - Describes the objective of that test.
    - Specify the inputs for testing and expected outputs.
    - Provides step-by-step procedures for executing the test.
    - Outlines the pass/fail criteria of the test.
* **Test Case Report**
  + This document will present the results of the test case executions.
* **Defects Reports**
  + Defects reports will store the information about the identified defects and resolutions.
* **Customer Sign Off**
  + This is the document that the customer of this project will sign to validate that system meets all the REST calls requirements of the system.

# Resource & Environment Needs

## Testing Tools

***Table 4: Testing*** *Tools*

|  |  |
| --- | --- |
| Process | Tool |
| Test Case Specification | MS Office |
| Test Case Tracking | MS Excel |
| REST Calls Testing | Fiddler |
| Dialogflow Agent Testing | GCP Assistants API |
| Defect Management | MS Word |
| Test Reporting | MS Word |

 Test Environment

**Hardware:**

Laptop or PC with the following minimum specs:

* Intel Core i5
* 8GB RAM:
* 2.0 GHz Processor

**Software:**

The following **software**will be required for testing the system:

* **Windows 8:** Chrome (latest version)
* **Mac OS X:** Chrome (latest version)
* **Fiddler:**(latest version)
* **GCP Console**
  + **Dialogflow Service**
  + **Google Assistant**

**References**

1. Project Management Institute. (2017). *A Guide to the Project Management Body of*  *Knowledge (Pmbok Guide)* (6th ed.). Newtown Square, PA: Project Management Institute.
2. Project Management Institute. (2019). *The Standard for Risk Management in Portfolios,*  *Programs, and Projects*. Newtown Square, PA: Project Management Institute.
3. Project Management Institute. (2018). *The Standard for Organizational Project*

*Management*. Newtown Square, PA: Project Management Institute.

Onestoptesting.com (n.d.) *Suspension criteria & resumption requirements*  
 Retrieved from: http://www.onestoptesting.com/test-plan/resumption-   
 requirements.asp

# Appendix A WBS

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generatedTable

Description automatically generated