

Test Plan: Dialogflow

Version 1.2

By:

Stefon Williams

Nicholas Ballo

Abdul Kamara

Thomas Barton

Sohail Sobhani

Caleb Crickette

Eugene Kim

Page Break

**Project Plan Approvals**

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date |
| Approved by: |  |  |
| Approved by: "Stakeholder" |  |  |
| Approved by: "Project Manager" |  |  |

Page Break

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Name | Description of Change(s) |
| 1.0 | 2/23/2021 | Eugene Kim | Initial Test Plan for Formscriber - Dialogflow team |
| 1.2 | 2/23/2021 | Abdul Kamara | Reviewed, fix font size consistency and text alignment |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Introduction 5](#_Toc65008430)

[Purpose 5](#_Toc65008431)

[Test Scope 5](#_Toc65008432)

[In-Scope 5](#_Toc65008433)

[Out-of-Scope 5](#_Toc65008434)

[Test Objective 6](#_Toc65008435)

[Intended Audience 6](#_Toc65008436)

[Roles and Responsibilities 6](#_Toc65008437)

[Testing Schedule 7](#_Toc65008438)

[Key Milestones 7](#_Toc65008439)

[Testing Approach 7](#_Toc65008440)

[Test Automation 8](#_Toc65008441)

[Functional Requirements of Test Cases 8](#_Toc65008442)

[Test Cases 9](#_Toc65008443)

[**Test Case 01**: Isolate primary user’s voice for use within the service 9](#_Toc65008444)

[**Test Case 02**: Dialogflow service is integrated within the mobile application 9](#_Toc65008445)

[**Test Case 03**: Receive report identifier from user 10](#_Toc65008446)

[**Test Case 04**: Get report fields from database 10](#_Toc65008447)

[**Test Case 05**: Process report contents and fields. 11](#_Toc65008448)

[**Test Case 06**: Generate dynamic intents to capture report fields 11](#_Toc65008449)

[**Test Case 07**: Identify report contents extracted by the Dialogflow agent 12](#_Toc65008450)

[**Test Case 08**: Store extracted report contents into a database 12](#_Toc65008451)

[**Test Case 09**: Initiate connection with database 13](#_Toc65008452)

[Testing Criteria 14](#_Toc65008453)

[Suspension Criteria 14](#_Toc65008454)

[Resumption Criteria 14](#_Toc65008455)

[Test Completeness 14](#_Toc65008456)

[Test Deliverables 14](#_Toc65008457)

[Resource & Environment Needs 15](#_Toc65008458)

[Testing Tools 15](#_Toc65008459)

[Test Environment 15](#_Toc65008460)

[Appendix A: Terms / Acronyms 16](#_Toc65008461)

[Appendix B: References 17](#_Toc65008462)

# Introduction

## Purpose

This document describes the plans for testing the functional requirements of REST calls for the **Formscriber – Dialogflow Service.**This document 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 document will be used to test the application's input and output to meet the in-scope and out-of-scope requirements' expectations. This test plan will include the in-scope and out-of-scope testing, as described below.

### In-Scope

The scope of this test plan is to test the REST services of the **"Formscriber- 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 **Formscriber – Dialogflow Service** as it interacts with the other team’s deliverables is 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, other teams like the web team will also rely on data that our team will provide. As the project evolves and gets completed, test cases for this particular method are to be determined as the process of identifying them will need coordination with the other teams.

### Out-of-Scope

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

## Test Objective

The objectives of the testing of the **"Formscriber 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 fix them properly.

## Intended Audience

This test plan document's intended audience include the project manager, developers, testers, and other team members who will participate in the REST calls testing of the **Formscriber – 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 | 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 | Tester | 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 Formscriber – Dialogflow Service will be performed:

* **REST Calls Testing**
  + The REST calls of the Formscriber – 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., Formscriber 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 “Formscriber, 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**: Get report fields from database

|  |  |
| --- | --- |
| **Description:** | Retrieve report fields from database using report identifier. |
| **Requirements:** | The user will initiate a request to fill out a report with predefined fields. The user will trigger voice commands to capture content for each field. |
| **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 field. |
| **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 with user confirmation for the field content. |
| **Post Condition** | If the user confirms field content, the system will respond with a success message. |
| **Alternate Path** | If the user does not confirm valid field entry, the system will prompt the user to try again. |

## **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 08**: Store extracted report contents into a database

|  |  |
| --- | --- |
| **Description:** | Extracted report contents from user’s voice must be inserted into database to generate the final report. |
| **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 * Confirmation of the user’s completion of report |
| **System Response** | The system will respond stating that the report has been saved and to view the final report on the web application. |
| **Post Condition** | The system will store the report fields into a database for the web application to use when generating the final report. |
| **Alternate Path** | If the user does not confirm, the system will continue to listen to the user’s data requests. |
|  |  |

## **Test Case 09**: Initiate connection with database

|  |  |
| --- | --- |
| **Description:** | Database connection is initialized and any exceptions are handled and presented to the user. |
| **Requirements:** | The user initiates a request to create a report. The system will then try to query the database for the appropriate report. |
| **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 with a confirmation of the report metadata. |
| **Post Condition** | The system successfully connected with the database. |
| **Alternate Path** | If the system cannot connect to the database, the system will respond to the user notifying them of a connection error, and to notify DevSecOps. |

## **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. |
| **Trigger** | The user will say a voice trigger with intent of seeking support. |
| **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 Formscriber.com” 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

* The build contains many severe defects which seriously or limit testing progress.
* The client suggested significant changes in requirements.
* Software/Hardware problems.
* Assigned resources are not available when needed by the test team.

## Resumption Criteria

Resumption will only occur when the problem(s) that caused the suspension will be resolved.

## 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 Cases Results**
  + 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 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 Assistants**

# Appendix A: Terms / Acronyms

***Table 5: Terms*** */ Acronyms*

|  |  |
| --- | --- |
| TERM/ACRONYM | DEFINITION |
| UI | User Interface |
| REST | Representational state transfer |
| GB | Gigabyte |
| SRS | Software Requirements Systems |
| Formscriber | 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 |

# Appendix B: References

Jenkins, N. (2017). A software Testing Primer. *An Introduction to Software Testing v2.*

Software Requirements Specification for Formscriber – Dialogflow (version 2.0)

[www.guru99.com](http://www.guru99.com) (n.d). Retrieved February 20, 2021 from

*https://www.guru99.com/test-plan-for-project.html*