Test Plan Document

For

American Printing House for the Blind

By

Team MGN

1 Introduction

This is a test plan for the O and M Trivia App by American Printing House for the Blind, which is an app designed to teach visually impaired or blind students Orientation and Mobility skills they need for daily life. The game will utilize Google Home devices to ask students Orientation and Mobility trivia questions, and then will provide feedback regarding whether each question was answered correctly. This test plan will be used to prepare the app for handoff to a software development team at American Printing House for the Blind

2 Business Background

The app system consists of 3 primary parts:

1) The action package that defines how users may invoke the O & M Trivia App
2) The server that carries on the conversation with user, asking questions and providing feedback to question answers
3) The service that provides O & M trivia questions to the conversation server.

The kinds of testing each component will require will be cases such as:

a) The server and service components will require response time testing to ensure they meet the 5 second response times required by Google Assistant
b) Load testing will need to occur on the server and service components to see how many trivia questions they can respond with, how many concurrent users they can handle, and so forth
c) The Google Action package that defines how to invoke the app will need to be tested via well-defined verbal commands to ensure the app can be properly started under various circumstances and commands

3 Test Objectives

- To ensure business and software usability requirements are met, as set forth by the American Printing House for the Blind
- To ensure the end users are granted a pleasurable experience throughout the usage of the system
- To find defects created during the initial development phase

4 Scope

Inclusions

- The Google Action package will be tested, to ensure that initial app invocations are triggered correctly
- The conversation server will be tested thoroughly to ensure smooth conversations occur with the user(s)
- The micro-service that provides trivia questions will be tested to ensure invalid requests are handled gracefully and to ascertain what limits may be imposed upon how much data it can provide and how many users can request at one time

Exclusions

- Firebase functions and the functionality it provides will not be tested
    - Currently the conversation server and the micro-service are hosted in Firebase functions, for development purposes
    - However, APH will not use Firebase functions when they release the app – thus it has little bearing on the long-term sustainability of the system

5 Test types Identified

- Server load testing
- Micro-service load testing
- Server response speed testing
- Server unit testing
- App invocation testing
- Exploratory system testing

6 Problems Perceived

- Problems have occurred with getting Google Assistant to recognize vocal invocations defined in the project's action package
- Problems have occurred with multiple team members with setting up a Google Actions project
- Temporary scaling problems exist due to the usage of free development servers with Google Firebase and Mongo, which may impact load testing to some degree

7 Architecture

- We have a startup, a questions and answers section, a database, and a microservice to serve data to the questions and answers part from the database.

8 Environment

- There are three major components to our system. Two are on one environment, the node server to fulfil requests and Mongo Database that holds the questions and answers database.
- Local Server: Hosts the node server and MongoDB.
- MongoDB has distributions for all major operating systems. The setup, imports, and exports follow the same schema, so migration is a low-cost operation.
- Node.js is pseudo platform independent. The packages used by Node are dependent on the OS however. Installing node and its package manager gives access to platform correct packages. Therefore, setting up the server is relatively low cost, and works so long as platform specific packages are not used. We are using packages that are supported on Linux and Windows systems alike. Migration is as simple as installing the node packages in the project directory then copying over the developed files.

9 Assumptions

We assume the target understands how to use the Google Home and everything is set up. We also assume they have an active internet service. These assumptions are requirements to use the Google Home in the first place.

For our project we also assume that users have a basic understanding of orientation and mobility and what difficulties to select for various students.

10 Functionality

Constraints and Resolutions

Parameter	Customer Constraints	Infosys Limitations
Setup Interface	Need to set player count, and difficulty	The customer and the speech recognition need to be able to accurately confirm and select settings.
Trivia Game	The user needs to be able to give answers well.	We need to compensate for Google's speech recognition

Risk Identified & Mitigation Planned

The data object passed between functions can act fairly volatile and be unpredictable at times. There is no way to mitigate this. A loss of data is a critical error and can even lead to catastrophic system failure.

Test Strategy

Exploratory testing alongside unit tests will likely be the best way of finding issues with the functionality. A unique problem presents itself with the way the dataflow works with a Google Home app. It requires us to enter and exit the function at a minimum of two times for one test of sample data.

Automation Plans

As the helper functions in the app are launched by a dispatcher by way of a state machine the best way to test this is to use dependency injection to test the data object.

Since there are only 3 valid difficulties, and 4 valid number of players we can test using test data and valid data to ensure successful entry and exit.

Deliverables
Verified functionality in the various functions that help the intent handlers fulfil requests from the user will be delivered. A request is when a user answers a question or when they answer the Google Home itself.

11 Security

Constraints and Resolutions

Parameter	Customer Constraints	Infosys Limitations

Risk Identified & Mitigation Planned
There is no sensitive information going into or out of the server. The only attack vectors available to malicious users or code is the Google Home, our server's post request fulfillment and the data going between the two.

The biggest possible risk is a middle man sitting between the Google Home and the server and modifying data that arrives to the Google Home or to the server. It is out of our hands at this point as this communication is managed entirely by the Google Actions API, and its actions-on-google node module.

Test Strategy
There is nothing to test.

Automation Plans
There's nothing to test.

Deliverables
There's nothing to deliver.

12 Performance

Constraints and Resolutions

Parameter	Customer Constraints	Infosys Limitations
Internet Connection	Need to be able to access the server.	After a 5 second delay the google home will report a non-responsive server and terminate the session.

Risk Identified & Mitigation Planned

If the user's internet connection, or the host's internet connection is compromised then the session will be terminated which will result in a loss in progress for the user. There is currently no mitigation for such a loss of data. We can store the data and session information and tie it to an account later if this occurs often enough. Architecture is implemented and designed in such a way to help make it easy to migrate to such a system.

Test Strategy

Timing each function and testing how long it takes for each function to fulfil the current request can be valuable as the time each function takes can threaten the connection terminating. Ideally, we can keep each function's sigma time below one second to compensate for connection interruptions.

Automation Plans

Run multiple and asynchronous instances of each helper function simultaneously and record the time it takes for each to execute.

Deliverables

A time table for average run time of each helper function so that we know out sigma time and how large of a server load the system can handle.

13 Usability

Constraints and Resolutions

Parameter	Customer Constraints	Infosys Limitations
Actions SDK documentation is lacking. Piecing together from sparse resources is time consuming	This is only a constraint on the time of delivery to our sponsor.	The time constraint will have to be met with team work to overcome the limitation.
Google Assistant does not always recognize speech correctly.	Customer, upon rendering their spoken response, may possibly be misunderstood and graded incorrectly.	Limitation in the system would be the Google Assistant's speech recognition. Our resolution would be to take such limitation into account when grading customer responses.

Risk Identified & Mitigation Planned

The risk here is that the users will not be able to accurately use the system. This would result in visually impaired persons not gaining the proper educational experience, which could have undesirable consequences.

Our mitigation is to take into consideration the possible scenarios in which the Google Assistant will incorrectly recognize speech. By thorough testing, we believe we can lessen the cognitive load on the user and provide a more comprehensive learning experience.

Test Strategy

The test strategy is a mixture of unit testing and manual testing. For some things, we will be able to unit test that our application receives and sends correct data. However, testing the speech recognition of the Assistant in relation to our application will have to be tested with a combination of unit testing and manual testing, including testing different dialects and different answer combinations (e.g. such as the case where the article 'a' should be a recognized entry preceding the answer).

Automation Plans

The automation is planned to be executed using the AVA JavaScript unit testing tool. This is installed from the command line using NPM and, currently, is run using the command line, although it may be able to be triggered from a voice command.

Deliverables

The deliverables in this case are an application and database of question data. The application will be one that will take questions sent to us by APH, read the question to the user as well as the possible answers, take the users' spoken feedback, and provide them with feedback on how they've performed. The database is currently a MongoDB housing the questions as delivered to us by APH after their performed field testing.

14 Test Team Organization

Our test team will be a smash and grab organization. In this case, we must simulate multiple dialects, multiple user scenarios and many questions to validate the behaviour of our application in a live situation.

15 Schedule

Our testing schedule will be a "flow-as-you-go" due to class and work schedule. Currently, it appears that our best time to get both teams (Home and Alexa) together is either on Thursdays or Saturdays. This is subject to change but as of now, this will be our best course of action. Individual testing is possible due to the Google Assistant being available via and Android phone. Any team members who have access to a Home or Alexa device to which they can connect their account can perform individual testing as well.

16 Defects Classification Mechanism

Type of Defects	Functionality	Performance	Security	Usability	Compatibility
Critical	Not allowing user to start app or submit answer.	Should allow user to start a trivia game and submit verbal responses.	Do not allow a trivia game to be activated by unauthorized user	Easy set up and answer submissio n.	Should be able to have system delivered from our own accounts to APH servers.
Major	Not catching every input for answer submission	Should catch the verbal submission accurately. However, Google Assistant has issues correctly recognizing certain input.	Should be limited to admins being able to adjust the verbiage input checking options.	Accuratel y capture input, yielding a smooth experienc e.	This should not be an issue, as the files will be passed off to APH in the form that we have tested them.

Minor	This will require more testing to be done to find issues of this level.	N/A	Not a security issue that we are currently aware of.	N/A	N/A
Cosmetics	Voices, pitch, volume needing adjustment to give more satisfying experience.	Should provide the most pleasant experience possible for each user.	This is not a security issue	Presentation can make a better experience.	May need adjustment to meet expectation of APH and end users.

Defects Logging and Status Changing Mechanism

APH has set up a space for us in their tracking system to record these issues. Prior to the project being handed over, we will be hosting this on our own accounts. During this time, we may also use Trello to track defects and remedies of such.

Turn Around Time for defect fixes

Turn-around time will be dependent on the severity and time constraint of our schedules. This will be a learning experience for all of us and thus will likely have a loosely large time frame to begin with until we have a better understanding of the process time frame.

17 Configuration Management

The configuration of our system will have to be one that we choose to most accurately mimic the situation for which it will be used, present the system to APH and confirm that it meets the needs as they see it. If it does not in any way, we will have to make necessary adjustments. This will be best to do as early as possible and perhaps multiple times.

18 Release Criteria

Release criteria will, as in most instances, be a validation process through our sponsor, APH. The system must be able to handle multiple users, be able to shift difficulty levels per game, accurately dictate the question and, where applicable, answers to the user(s), and, upon receiving user response, provide accurate and directive feedback to the user. Without any of the above, the system will fail.