Functional Test Plan

For

Case Western Reserve University Baseball Alexa Skill

Version 4.0.0

Prepared by Steven Barker, Mark Gross, and Gavin Markowitz

December 7, 2018

Table of Contents:

Revision History

Date	**Reason for Change**	**Version**
11/6/2018	Initial Document with outline and 3 sections written	1.0.0
11/8/2018	Review Document and write final sections	2.0.0
11/9/2018	Review Document and submission	3.0.0
11/25/2018	Revisions on testing due to capabilities of Amazon Web Services built in testing environment	4.0.0

1. Introduction

The purpose of this document is to provide the outline and goals for the testing strategy of the Case Western Reserve University Baseball Alexa Skill. This document will list what our functional tests are for the Alexa Skill and how we plan to implement these tests. Functional testing areas include interpreting string input to determine what information to retrieve, fetching information for the team, individual team participants, or team schedule, and forming a response.

a. Objectives
   i. Specify tests and provide guidelines for functional testing plan
   ii. Make sure test cases verify that the software fulfills the requirements that we laid out in the SRS document.

2. Project Scope

This document outlines the goals which need to be accomplished in order for the Case Western Reserve University Baseball Alexa Skill to be properly tested.

a. Scope of Testing
   i. The CWRU Baseball Alexa Skill will be tested on a virtual Amazon Alexa device hosted on Amazon Web Services (AWS). The virtual Alexa must receive requests and generate responses.The specific tests are listed in the Test Specifications section of this document.

b. Out of Scope of Testing
   i. We will not be testing on an actual Amazon Alexa device, as the virtual Alexa system will be sufficiently similar in terms of the way it acts to negate the need for such a test.
   ii. Due to the nature of requests given to Amazon Alexa and the nearly infinite number of input requests that would be possible from a user, it is impossible the test the full range of requests in any reasonable amount of time with the resources available to us. So we will be testing a limited but comprehensive sample of requests, specified in our Test Specifications section.

c. Testing Schedule
   i. We will begin testing for each package outlined in our SRS immediately after completing the code for all classes that package. Once testing on a package is complete, the coding of another package or adding of additional features may begin.

3. Test Design Specification

a. Specification Format

Different tests are specified for each different function in the software. Each individual step of the tests will begin with the action by the user and go through the steps to get the expected outcome of the software. If all of the steps are completed correctly then the function passes the test.

b. Presumption and Suspension Criteria

Any tests that does not reach the final step of testing will be considered to have failed that test.

4. Test Specifications

a. Request team statistic for statistic that does exist
1. User activates Alexa using “Alexa ask Case Baseball...”
2. User asks Alexa for a specific team statistic
3. Alexa takes the user input
4. Input is checked for keywords to determine what user is asking for
    - i. If no year is given, default to current year
    - ii. If keywords are insufficient to determine what user is asking for Alexa will tell user that it was unable to determine what they were asking for and exit the program. User will then have to ask Alexa for a different query.
5. Keywords are used to pull correct information from the CWRU Athletics website
6. The requested information is formatted in order to make it easier for Alexa to deliver it as output
7. Alexa outputs the requested information as a sentence

b. Request team statistic for statistic that does not exist
1. User activates Alexa using “Alexa ask Case Baseball...”
2. User asks Alexa for a specific team statistic
3. Alexa takes the user input
4. Input is checked for keywords to determine what user is asking for
    - i. If no year is given, default to current year
    - ii. If keywords are insufficient to determine what user is asking for Alexa will tell user that it was unable to determine what they were asking for and exit the program. User will then have to ask Alexa for a different query.
5. Since not enough keywords are found to match a supported type of statistics, program will not request any data from the CWRU Athletics Website
6. No fetch method will be called
7. Alexa tells the user that the data does not exist or the question was not correctly understood

c. Request individual team participant statistic that exists

1. User activates Alexa using “Alexa ask Case Baseball...”
2. User asks Alexa for a specific team participant’s statistic
3. Alexa takes the user input
4. Input is checked for keywords to determine what user is asking for
    i. If no year is given, default to current year
    ii. If keywords are insufficient to determine what user is asking for Alexa will tell user that it was unable to determine what they were asking for and exit the program. User will then have to ask Alexa for a different query.
5. Keywords are used to pull correct information from the CWRU Athletics website
6. The requested information is formatted in order to make it easier for Alexa to deliver it as output
7. Alexa outputs the requested information as a sentence

d. Request individual team participant statistic that does not exist
   1. User activates Alexa using “Alexa ask Case Baseball...”
   2. User asks Alexa for a specific team participant’s statistic
   3. Alexa takes the user input
   4. Input is checked for keywords to determine what user is asking for
       i. If no year is given, default to current year
       ii. If keywords are insufficient to determine what user is asking for Alexa will tell user that it was unable to determine what they were asking for and exit the program. User will then have to ask Alexa for a different query.
   5. Since not enough keywords are found, program will not request any data from the CWRU Athletics Website
   6. No fetch method will be called
   7. Alexa tells the user that the data does not exist or the question was not correctly understood

e. Request previous / next upcoming game
   1. User activates Alexa using “Alexa ask Case Baseball...”
   2. User asks Alexa for previous or next upcoming game
   3. Alexa takes the user input as a string
   4. String is checked for keywords to determine what user is asking for
       i. If keywords are insufficient to determine what user is asking for Alexa will ask user to repeat the question
   5. Keywords are used to pull correct information from the CWRU Athletics website
   6. The requested information is formatted in order to make it easier for Alexa to deliver it as output
   7. Alexa outputs the requested information as a sentence

f. Request information about a specific game by date that exists
    1. User activates Alexa using “Alexa ask Case Baseball...”
    2. User asks Alexa for a specific game by date for the CWRU baseball team,
    3. Alexa takes the user input as a string
    4. String is checked for keywords to determine what user is asking for
        i. If no year is given, default to current year
        ii. If keywords are insufficient to determine what user is asking for Alexa will tell user that it was unable to determine what they were asking for and exit the program. User will then have to ask Alexa for a different query.
    5. Keywords are used to pull correct information from the CWRU Athletics website
    6. The requested information is formatted in order to make it easier for Alexa to deliver it as output
    7. Alexa outputs the requested information as a sentence

g. Request information about a specific game by date that does not exist
    1. User activates Alexa using the “Alexa ask Case Baseball…”
    2. User asks Alexa for a specific game for the CWRU baseball team,
    3. Alexa takes the user input as a string
    4. String is checked for keywords to determine what user is asking for
        i. If no year is given, default to current year
        ii. If keywords are insufficient to determine what user is asking for Alexa will tell user that it was unable to determine what they were asking for and exit the program. User will then have to ask Alexa for a different query.
    5. Keywords are used to pull correct information from the CWRU Athletics website and thus determine that game does not exist
    6. Alexa delivers a response to the user informing them the game does not exist

5. Functional Tests

a. Receive and Response Package Tests

Tests run on this package will be run using Amazon Web Services built in test environment. The test will consist of a preset request by Alexa and ensuring manually that the output is an expected English response.

Test	Description	Setup	Expected Result
RCT-1	Use a valid request to produce a valid response	Input a valid request to receive a valid response	A correct English sentence answering the request is returned
RCT-2	Use an invalid request to produce an invalid response	Input an invalid request by asking for something not supported by the application	Return an English sentence saying that the information could not be understood or could not be returned

b. Fetch Package Tests
Below are all functional tests associated with classes in the Fetch Package. All tests in this package will be run using Python unit testing.

i. Team Class Tests

Test	Description	Setup	Expected Result
TCT-1	Fetching Team statistic without specified year	Create instance of Team with no specified year	Created instance is of the current year
TCT-2	Fetch Team Statistic for specified year, object creation	Create instance of Team for specified year	Created instance is of the specified year
TCT-3	Fetch team statistic for a specified year	Create instance of Team for a specified valid year and fetch a statistic	Return requested team statistic from the specified year
TCT-4	Attempt to fetch team statistic for invalid year	Create instance of Team with an invalid year that cannot be retrieved from website	Return error stating that the information is unavailable for the specified year

ii. Team Participant Class Tests

Test	Description	Setup	Expected Result
TPT-1	Fetching Team Participant statistic without specified year	Create instance of Team Participant with no specified year	Created instance is of the current year
TPT-2	Fetch Team Participant Statistic for specified year, object creation	Create instance of Team Participant for specified year	Created instance is of the specified year
TPT-3	Fetch Team Participant statistic for a specified year for a specified player number	Create instance of Team Participant for a specified valid year and fetch a statistic for the specified player number	Return requested team statistic from the website for the specified year for the specified player number
TPT-4	Attempt to fetch team statistic for invalid year	Create instance of Team Participant with an invalid year that cannot be retrieved from website	Return error stating that the information is unavailable for the specified year
TPT-5	Attempt to fetch team statistics for valid year but player number who cannot be found	Create instance of Team participant, attempt to fetch statistic for specified player number who cannot be found on website	Return None

iii. Schedule Class Tests

Test	Description	Setup	Expected Result
SCT-1	Fetch Schedule without specified year	Create Schedule object without specified year	Created object is of the current year
SCT-2	Fetch Schedule for specific valid year,	Create Schedule for specific valid year	Created object is of specified year

	object creation		
SCT-3	Fetch Schedule for specific valid year, fetch a game for a date that does contain a single game	Create Schedule object, fetch game for specified date	Returns list containing only correct game for the specified date
SCT-4	Fetch Schedule for specific valid year, fetch a games for a date that contains multiple game	Create Schedule object, fetch games for specified date with multiple games	Returns list containing all correct games for specified date
SCT-5	Fetch Schedule for specific valid year, fetch a games for a date that contains no games	Create Schedule object, attempt to fetch games from date that has no games	Returns empty list

6. Defect Responsibility & Resolution

a. Issue Tracking

To track issues, our group will used the “Issues” section of the Github version control. This will easily allow all group members to see current issues as well as communicate discovered issues to the individual that will need to address the problem.

b. Responsibilities

The following table outlines the parts of the system and the developer who is primarily responsible for the planning, implementation, and testing of the system. The primary developer is responsible for the design, but all team members are responsible in assisting if requested by the primary developer.

System Part	**Developer**	**Responsibilities**
“Receive” Package	Gavin Markowitz	● Interpret Question ● Use Fetch package to fetch correct data for the input ● Pass all data to “Respond” package
“Fetch” Package	Mark Gross	● Connect to Case Athletics Website for given year

		● Extract all data to be used ● Provide interface to return the fetched data
“Respond” Package	Steven Barker	● Choose correct type of response based on data fetched ● Have response be a grammatically correct sentence ● Provide feedback when a fetched response could not be given.

7. **Exit Criteria**
    a. All tests have been run and all tests have passed.
    b. No stopping in main loop (requests through responses can be taken in succession without stopping to to error or other stopping condition).