Voice Campus

Test Plan Specification Document

Team 6

Kang Hyunmuk

Kim Jihye

Park Jiye

Shin Wonchul

Oh Seungjun

Date: 2021.05.30.

CONTENTS

LIST OF TABLES

1. Introduction

1.1. Purpose

This document is a Test Plan for Voice Campus (Blind-Friendly Learning Application) project. All people who related to testing are readers of the document. In this project, a testing team is same to a development team. Then, team 6 is the main reader.

This document mainly focuses on how to test Voice Campus project in detail. Why should the test be done? Test is used for dynamic verification and validation, which, as you know, is very important in developing a system. Therefore, testing and planning how to conduct the test is also essential.

Basically, testing is the process of putting input in and checking the output of the system to see how the system works. The test mainly has two objectives: first to see if it works properly, and second to find out where it works strangely. Testing for the former is called validation testing, and testing for the latter is called defect testing. This also changes the type of input that needs to be entered. For a successful testing, this document focuses on both. A person who has read this document will be able to create an artificial data based on the content of the document. By viewing the results and comparing to this document's expected result, successful testing can be carried out.

1.2. Scope

There are development testing, release testing, and user testing as types of software testing. In particular, this document deals with a development testing. Release testing and user testing are beyond the scope of this document. Development testing includes unit testing, component testing (interface testing), and system testing. But this document deals with unit testing and component testing, except system testing.

In unit testing, functional features are the main test targets. And in interface testing, components' interfaces are the main test targets.

1.3. Definition, Acronyms, and Abbreviation

[Table 1] Acronyms, Abbreviation

Acronyms & Abbreviations	Explanation
VA	Voice Assistance service, Android's voice assistant and iPhone's voice over
TTS	Text To Speech
SST	Speech To Text
RAM	Random Access Memory
JSON	JavaScript Object Notation
OS	Operating System
SRS	System Requirement Specification
API	Application Programming Interface
UI	User Interface
HTTP	Hypertext Transfer Protocol
TA	Teaching Assistance

1.4. Overview

The remainder of document consists of four sections: section 2, section 3, section 4, and section 5. Among them, section 3 and section 4 are the core contents of this document.

Section 2 is the overall description of how to implement test about Voice Campus. It describes test methods like Software unit test methods and Software interface test methods that will be implemented in testing the Voice Campus. After you read the section, you can imagine how the testing methods will be practiced. Section 3 is the Software Unit Test. This section is a detailed description of software unit test that must be followed to test a Voice Campus. Therefore, it is a part that developers should pay particular attention to. Section 4 is the Software Interface Test. This section is a detailed description of software interface test that

must be followed to test a Voice Campus. So, please pay attention to this chapter too. Section 5 is about our record of document.

Explaining the structure of Test Plan Document is done. Please refer to it for your reading.

2. Approach

2.1. Test method

2.1.1. Software unit test methods

Unit test is a test that cover the smallest parts of code to check if it works correctly. It is conducted by the developer during code development process. It takes in a sample input, and checks if outputs are same as expected. The inputs and outputs should be boundary values if they vary on some regions. If there are external dependencies, mocking should be used. Mocking creates an object that imitates the object with external dependencies. For implementation, Jest 27.0.1 will be used.

Chapter 3 describes the unit test plans, chosen by following rules:

- UIs are not tested.
- Networking, external system, file or database I/O, interaction between modules are not tested.
- Each functions can have multiple tests.
- Avoid using logics in test; Divide the test if it checks multiple states.

2.1.2. Software interface test methods

Software interface tests external interfaces between systems and components such as data transmission between Learning X server and the application. Test aims to find out whether the interface enables to function properly in the appropriate preconditions and action. Moreover, the test would also check the proper error message pops out in the failure conditions and unfulfilled preconditions.

Automated test would be performed for the test and therefore, executable for the driver

program that can run the test should be built in advance. Because steps of tests will be executed automatically by the code, test can be performed repeatedly, and it would help the developers to test the interfaces' functionality, performance, and stress resistance.

Chapter 4 describes the software interface test, followed by the rules below.

- Each interface is tested multiple times.
- Stress testing is used.
- Also design test which cause the component to fail.

3. Software Unit Test

3.1. Login unit

[Table 2] Login unit test

Item to Test	Test cases	Input values	Expected outputs
Login()	Check if getting authentication from the LearningX is possible when accurate user information is given.	Appropriate user login information	Login successful
Login()	Check if authentication is blocked when wrong user information is given.	Invalid user login information	Login denied

3.2. To-do list unit

[Table 3] To-do list unit test

Item to Test	Test cases	Input values	Expected outputs
GetListInfo()	Check if GetListInfo function brings accurate data from LearningX server when given right input.	Appropriate authenticated user information	Data is brought by the function
GetListInfo()	Check if GetListInfo function output message when the function fails to connect to the LearningX server.	Unauthorized information	Failure message

ShowList()	Check if list is constructed from the data sent from LearningX server in the right order.	Sample data of user class information	List is constructed in the right order
SortPriority()	Check if list's sorting priority changes according to the user's input.	Sample json data and data driven from the ShowList function	Priority change

3.3. Play lecture unit

[Table 4] Play lecture unit test

Item to Test	Test cases	Input values	Expected outputs
GetLecture()	Check if the function brings desired lecture from the user input when appropriate link was given	Sample data holding valid link	Video and learning progress data are driven from the function
GetLecture()	When the unavailable link is given to the function warning message should pop out.	Sample data with invalid link	Failure message
OpenLecture()	Considering the video data and learning progress data, lecture should be open from the position where user stopped.	Lecture data and leaning progress data sent from LearningX server	Lecture player opens considering the learning progress
ProgressUpdate()	Check whether progress gets updated when user watches the lecture from the application	Data input renewed by each second	Learning progress is updated.

3.4. VoicePlay unit

[Table 5] VoicePlay unit test

Item to Test	Test cases	Input values	Expected outputs

GetPlaySetting()	Check GetPlaySetting function if user's input is parsed correctly to the VoicePlay class attribute.	Sample intent	Attribute "playsetting" of VoicePlay class matching to the sample intent data
GetPlaySetting()	Check GetPlaySetting function if user's input is parsed correctly to the VoicePlay class attribute when conflicting options are on.	Sample intent ('including memo' option is off, 'only memo listening' is selected)	Error message
VoicePlay()	Check VoicePlay function executes correctly in a normal situation.	N/A	Attribute "status" of VoicePlay class is PLAY
VoicePause()	Check VoicePause function executes correctly in a normal situation.	N/A	Attribute "status" of VoicePlay class is PAUSED

3.5. VoiceMemo unit

[Table 6] VoiceMemo unit test

Item to Test	Test cases	Input values	Expected outputs
AlertMemoStart()	Check AlertMemoStart function alerts (as sound) correctly.	N/A	Success message

3.6. SoundFile unit

[Table 7] SoundFile unit test

Item to Test	Test cases	Input values	Expected outputs
GetSoundFile()	Check GetSoundFile function returns sound file list correctly in a basic case.	Sample playsetting ('playspeed' is 1.0, 'playstartpage' is 1, 'playconsecutive' is True,	Sample sound file list

		‘playmemo’ is False, ‘playonlymemo’ is False), sample fileid (any available file id)	
GetSoundFile()	Check GetSoundFile function returns sound file list correctly if user selects ‘play only memo’ option.	Sample playsetting (‘playspeed’ is 1.0, ‘playstartpage’ is 1, ‘playconsecutive’ is True, ‘playmemo’ is True, ‘playonlymemo’ is True), sample fileid (any available file id)	Sample sound file list
GetSoundFile()	Check GetSoundFile function returns sound file list correctly if user selects ‘page listening’ option.	Sample playsetting (‘playspeed’ is 1.0, ‘playstartpage’ is 1, ‘playconsecutive’ is False, ‘playmemo’ is False, ‘playonlymemo’ is False), sample fileid (any available file id)	Sample sound file list
GetSoundFile()	Check GetSoundFile function returns sound file list correctly if ‘including memo’ option is on.	Sample playsetting (‘playspeed’ is 1.0, ‘playstartpage’ is 1, ‘playconsecutive’ is True, ‘playmemo’ is True, ‘playonlymemo’ is False), sample fileid (any available file id)	Sample sound file list
GetSoundFile()	Check GetSoundFile function returns sound file list correctly if play speed is slower than 1.	Sample playsetting (‘playspeed’ is 0.25, ‘playstartpage’ is 1, ‘playconsecutive’ is True, ‘playmemo’ is False, ‘playonlymemo’ is False),	Sample sound file list

		sample fileid (any available file id)	
GetSoundFile()	Check GetSoundFile function returns sound file list correctly if play speed is faster than 1.	Sample playsetting (‘playspeed’ is 2.0, ‘playstartpage’ is 1, ‘playconsecutive’ is True, ‘playmemo’ is False, ‘playonlymemo’ is False), sample fileid (any available file id)	Sample sound file list
GetSoundFile()	Check GetSoundFile function returns sound file list correctly if start page is different from 1.	Sample playsetting (‘playspeed’ is 1.0, ‘playstartpage’ is 2, ‘playconsecutive’ is True, ‘playmemo’ is False, ‘playonlymemo’ is False), sample fileid (any available file id. File page should bigger than 2)	Sample sound file list

3.7. Test unit

[Table 8] Test unit test

Item to Test	Test cases	Input values	Expected outputs
GetTest()	Check GetTest function if test data from server is parsed correctly to test class variable.	Sample test json data	Test class attribute matching the test json data
GetTest()	Check GetTest function when the test data from server is not correct.	Wrong test json data	Failure message
ShowProblem()	Check ShowProblem function if the problem is correctly divided into description and	Sample problems for each type	Order of problem description and answer

	answer parts according to problem type.		
ShowProblem()	Check ShowProblem function when the problem data from server is not correct	Wrong problems for each type	Failure message
ShowTestDescription()	Check ShowTestDescription function if test description data from the server is parsed correctly to test class variable.	Sample test json data	Test description matching the test json data
ShowTestDescription()	Check ShowTestDescription function when the test description from server is not correct	Wrong test json data	Failure message
SendAnswer()	Check SendAnswer function if the answer data from user which satisfies the condition is correctly sent to the server	Sample answers for each type	Next problem
SendAnswer()	Check SendAnswer function when the answer data from user does not satisfies the condtion	Wrong answers for each type	Failure message
SendRecord()	Check SendRecord function when there is no recording file	Empty file	Failure message

3.8. STT unit

[Table 9] STT unit test

Item to Test	Test cases	Input values	Expected outputs
StartRecord()	Check StartRecord function when there is no recording permission.	No recording permission	Failure message

EndRecord()	Check EndRecord function when StartRecord function was not previously called	No previously called StartRecord function	Failure message
EndRecord()	Check EndRecord function when interval between StartRecord function exceeds the time limit 3 min.	StartRecord function called at 2 min 59 sec before, 3 min 1 sec before	Failure message for only exceeding 3 min.

3.9. TTS unit

[Table 10] TTS unit test

Item to Test	Test cases	Input values	Expected outputs
SaveTTSFile()	Check SaveTTSFile function if the document file is translated correctly to string.	Sample documents for each supporting type: .doc, .pdf, .ppt	String corresponding to document data.
SaveTTSFile()	Check SaveTTSFile function when the document type is not supported.	Sample document of unsupported document type	Failure message

4. Software Interface Test

4.1. Learning X Server interface test

4.1.1. Login

[Table 11] Login interface test

Description	Login to user's account
Precondition	User must have learning X account.
Action	When user opens the Voice Campus, the application automatically performs authentication from Learning X.
Expected Result	User successively logins to his or her account.
Failure Condition	1. Fail to connect to Learning X server 2. Fails to get authentication from the Learning X server

4.1.2. To-do List

[Table 12] To-do List interface test

Description	Make to-do list from the user information
Precondition	Voice campus is already logged in. Connection to Learning X application is available.
Action	When user clicks to-do list, user id is brought to the Learning X server. After the automatically performed authentication process, user information is sent to the Voice Campus application. With the data brought from the previous process, to-do list is constructed in the right order.
Expected Result	To-do List in constructed
Failure Condition	1. Fails to get authentication from the Learning X server. 2. Fails to bring data from Learning X server. 3. Fails to construct list from the data.

4.1.3. Open Lecture

[Table 13] Open Lecture interface test

Description	Open desired lecture in the Voice Campus application
Precondition	Voice campus is already logged in. Connection to Learning X application is available. User is authorized student to open the link of the lecture.
Action	Learning X server successfully delivers desired lecture information to the application. Information contains lecture video and learning process data.
Expected Result	Application opens desired lecture and plays it regarding on the learning process data.
Failure Condition	1. Fail to bring desired lecture to the application. 2. Fail to open lecture in the application. 3. Fail to open lecture regarding learning process data

4.1.4. Close Lecture

[Table 14] Close Lecture interface test

Description	Close lecture and deliver modified learning process data
Precondition	User is playing lecture in the application. Connection to Learning X application is available.
Action	When user close the lecture, learning process data is modified and the data is sent to Learning X server.
Expected Result	User successively logins to his or her account.
Failure Condition	1. Fails to deliver modified data to the Learning X server.

4.1.5. Get Test

[Table 15] Get Test interface test

Description	Get Test data from Learning X server.
Precondition	The user should be logged in
Action	When user requires test data, the test data is fetched from Learning X server. The problems of the test is also fetched.
Expected Result	Application successfully gets the test and problems data from Learning X server.
Failure Condition	1. Fail to bring desired test from server. 2. Fail to bring problems of desired test from server.

4.1.6. Send Answer

[Table 16] Send Answer interface test

Description	Send answer of problem to Learning X server.
Precondition	The user should be logged in and started taking test.
Action	When user prompts to send answer, the application sends the answer of specific problem to the Learning X server.

Expected Result	Learning X server receives the answer and application receives the success message.
Failure Condition	1. Fail to send the answer data. 2. Send wrong problem id of the answer. 3. Fail to receive success message from server.

4.1.7. Get Text File List

[Table 17] Get Text File List interface test

Description	Get text file to translate into voice file from Learning X server.
Precondition	User should be logged in to Learning X server.
Action	User select lecture and select course material text files.
Expected Result	Files that can be translated into voice files are listed.
Failure Condition	1. When user device's network is disconnected. 2. When Learning X server is not working.

4.2. Local DB interface test

4.2.1. Save Voice Memo

[Table 18] Save Voice Memo interface test

Description	Save voice memo
Precondition	VoiceMemo's SaveMemo function is called.
Action	SaveMemo function gets sample memofile and playstatus as input. It processes input into the proper form and save to the DB.
Expected Result	New data is added properly to the DB
Failure Condition	1. Saved data does not same to the sample DB data. 2. Data is not added to the DB because of wrong instruction.

4.2.2. Load Sound File

[Table 19] Load Sound File interface test

Description	Load sound files
Precondition	SoundFile's GetSoundFile function is called.
Action	GetSoundFile function gets sample playsetting as input. It gets data from the local DB according to the input.
Expected Result	Data from the local DB is same to the sample data
Failure Condition	1. Loaded data does not same to the sample data. 2. Data is not loaded from the local DB because of wrong instruction.

4.3. STT interface test

4.3.1. STT

[Table 20] STT interface test

Description	Translate text into speech by google service
Precondition	The user should be connected to the network.
Action	When the STT service is requested by user, the speech data is sent to the google STT cloud and text is responded.
Expected Result	The translated text is successfully returned.
Failure Condition	1. Fail to send data to google STT cloud. 2. Wrong format of STT request. 3. Fail to receive data from google STT cloud.

5. Supporting Information

5.1. Document History

[Table 21] Document History

Date	Version	Description	Writer
5/20	1.0	5.1	Kang Hyunmuk
5/22	1.1	1.1, 1.2	Kim Jihye
5/23	1.2	1.3, 1.4	Park Jiye
5/25	1.3	2.1.2	Shin Wonchul
5/26	1.4	2.1.1	Oh Seungjun
5/27	1.5	4.1, 4.3	Kang Hyunmuk
5/27	1.6	3.4, 3.5, 3.6, 4.2	Kim Jihye
5/28	1.7	3.7	Park Jiye
5/28	1.8	3.1, 3.2, 3.3, 4.1	Shin Wonchul
5/29	1.9	3.8, 3.9, 4.1, 4.3	Oh Seungjun