Software Testing Document

for

TalkBox

Prepared by Muhammad Danial Qureshi

York University
EECS2311 Software Development Project

04. February 2019

Contents

Revision History

Revision	Date	Author(s)	Description
1.0	04.02.2019	M.Qureshi	Chapter 1- Introduction
2.0	04.02.2019	M.Qureshi	Chapter 3 – Testing Checklist
3.0	05.02.2019	M.Qureshi	Chapter 3 – Testing Checklist
4.0	07.02.2019	M.Qureshi	Chapter 3,4,5
5.0	09.02.2019	M.Qureshi	Chapter 3,4,5
6.0	11.02.2019	M.Qureshi	Chapter 3,4,5
7.0	15.02.2019	M.Qureshi	Chapter 3,4,5,6
8.0	18.02.2019	M.Qureshi	Chapter 3,4,5,6
9.0	20.02.2019	M.Qureshi	Chapter 3,4,5,6
10.0	20.02.2019	M.Qureshi	Chapter 3,4,5,6

Chapter 1

Introduction

1.1 Purpose

This document provides information on test cases for the TalkBox application. This document covers the several test cases the application has, as well as justified derivation of each of these test cases. Test case derivation is justified thoroughly for both the simulator component of the TalkBox, as well as the configuration application component. The sufficiency of each test case is provided through this document as well as test coverage.

1.2 Testing Checklist

There are two testing checklists in the Testing Checklist chapter, one for the TalkBox configuration application and another for the TalkBox simulator application. These checklists are broken into two subsections of the chapter.

1.3 Testing Case Derivation

In the Test Case Derivation chapter of this document, a through description of how each testcase was derived is provided. There are two subsections to the Test Case Derivation chapter, one for the configuration application and another for the simulator application. In each of these subsections the respective applications test case derivation is provided.

1.4 Testing Case Sufficiency

In the Test Case Sufficiency chapter of this document, a justification for each test case from the Test Case Derivation chapter is provided. There are two subsections to the Test Case Sufficiency chapter, one for the configuration application and another for the simulator application. In each of these subsections the respective applications test cases are justified with proven sufficiency

1.5 Testing Case Implementation

In the Test Case Implementation section of this document, a description of how each test case is implemented is provided. There are two subsections to the Test Case Implementation chapter, one for the configuration application and another for the simulator application. In each of these subsections, the respective applications test case implementations are shown and justified.

1.6 Test Coverage

In the Test Coverage chapter of this document the entire Test Coverage will be shown after running all tests. Individual unit tests will also have their Test Coverages shown per test class.

Chapter 2

Overall Description

2.1 Testing Documents Description

Throughout this document there are several chapters that show different test cases as well as the respective derivation, sufficiency, and implementation of each of these test cases. A test coverage is also provided in this document. An understanding of the testing done and developed during the making of the TalkBox are thoroughly explored and conveyed throughout this document.

Chapter 3

Testing Checklist

3.1 TalkBox Configuration Application

Recorder Class Tests:

Tests	Pass / Fail	Comments
setup()	pass	Setting up objects and fields.
testInitialFields()	pass	Testing the initial fields of recording class
testRecording()	pass	Testing recording an audio file using the system and saving it to a directory
testButtons()		Testing the feature of changing the amount of buttons of the application

Simulator Preview (SimPreview) Class Tests:

Tests	**Pass / Fail**	**Comments**
setup()	pass	Setting up objects and fields.
testingUpdatingButtons()	pass	Testing update button method for the sim preview
testPlayingSound ()	pass	Testing audio output of application audio files
TestErrorPlayinSound()	Pass	Testing failure to play audio file due to wrong file or mic not found
testPlayinMissingSoundFile()	Pass	Testing playing audio if file is missing

PlayEditToggle Class Tests:

Tests	**Pass / Fail**	**Comments**
setup()	pass	Setting up objects and fields.
testAddingAudioToButtons()	pass	Testing adding audio to buttons in “Edit Mode”
testChangingButtonLabel()	pass	Testing changing the name of a button using naming feature in “Edit Mode”

ProfilesPanelTest Class Tests:

Tests	**Pass / Fail**	**Comments**
setup()	pass	Setting up objects and fields.

TalkBoxConfig Class Tests:

Tests	**Pass / Fail**	**Comments**
setup()	pass	Setting up objects and fields.
testSetNumAudioButtons()	pass	Testing update number of audio buttons on the TalkBox
testInitialFields	pass	Testing the intial set fields of the TalkBoxConfig app. (ex. Number of buttons)
TestJPaneSplits	Pass	Testing the different instances that are created by the configApp through JSplitPane components

3.2 TalkBox Simulator Application

TalkBox Simulator Tests (TalkBoxSimTest):

Tests	Pass / Fail	Comments

Chapter 4

Test Case Derivation

4.1 How Test Cases Were Derived: TalkBox Configuration Application

RecorderTest:
The Recorder tests cases are specifically tailored toward this class and its features. Derivations are as follows:

testInitialFields(): The initial fields of the recorder class are tested as upon launch they should have a certain state. For example, isRecording field should be false when the application is first launched to ensure there is no recording in progress.

testRecording(): The user is able to record from the recording panel, and this method tests the applications recording functionality.

testButtons(): The user is able to change the number of buttons from the recorder panel, and this method allows a test to see if this functionality is working correctly, by preforming a change on the amount of buttons.

SimPreviewTest:
The SimPreview tests cases are specifically tailored toward this class and its features. Derivations are as follows:

setup(): a configurator object is created and used to create a simulator preview object. These objects are needed to test the PlayEditToggle class.

testingUpdatingButtons(): The user is able to change the number of buttons from the recorder panel, and this method allows a test to see if this functionality is working correctly, by preforming a change on the number of buttons. This updates the number of buttons on the simulator preview.

testPlayingSound(): This method tests playing sound from a button, which is an essential feature of the simulator preview as well as the simulator in general.

testErrorPlayingSound(): This method tests what happens if there is an error playing back audio, for example if a file is null, the correct handled.

testPlayingMissingSoundFile(): This method tests what happens if there is an error playing back audio due to a missing audio file.

PlayEditToggleTest:

The PlayEditToggle tests cases are specifically tailored toward this class and its features. Derivations are as follows:

setup(): a configurator object is created and used to create a simulator preview object and recorder object. These objects are needed to test the PlayEditToggle class.

testChangingButtonLabel(): when in edit mode, the user is able to change the labels on each button. This method tests that functionality using the respective JTextFields and JButtons.

testAddingAudioToButtons(): When in edit mode, the user is able to add audio to a button of their choice. This method allows a testing of this feature by using the respective JTextFields and JButtons.

ProfilesPanelTest:

The ProfilesPanel tests cases are specifically tailored toward this class and its features. Derivations are as follows:

TalkBoxConfigTest:

The TalkBoxConfig tests cases are specifically tailored toward this class and its features. Derivations are as follows:

setUp(): a configurator object is created. This object is needed to test the main configurator class. A call to the main method of the configuration app is allow done in this method.

testSetNumAudioButtons: This method sets the number of audio buttons for the configuration app and tests if this change took place.

testInitialFields(): This method tests all initial fields of the TalkBoxConfiguration application to ensure they are correctly set upon launch of the application.

testJPaneSplits(): This method tests if all objects created by the TalkBoxConfiguration application are correctly initialized. This intasiation of objects takes place in the controlsProfileSplit class and the SimRecorderSplit class.

4.2 How Test Cases Were Derived: TalkBox Simulator Application

TalkBoxSimTest:

The TalkBoxSim tests cases are specifically tailored toward this class and its features. Derivations are as follows:

Chapter 5

Test Case Sufficiency

5.1 Why Test Cases Are Sufficient: TalkBox Configuration Application

RecorderTest:

testInitialFields(): This method is sufficient in testing the initial fields as it correctly tests all initial fields of the recorder class.

testRecording(): This method is sufficient in testing the recording feature as it creates a new audio file that is 2 seconds in length and correctly places it in the TalkBoxData directory.

testButtons(): This method is sufficient in testing the change in the amount of buttons, as it will update the number of buttons then check the simulator to see if the number of buttons has actually changed.

SimPreviewTest:

setup(): This method correctly sets up all necessary fields needed to thoroughly test the SimPreview class, and thus it is respectively sufficient.

testingUpdatingButtons(): This method is sufficient in testing the change in the amount of buttons, as it will update the number of buttons then check the simulator to see if the number of buttons has actually changed.

testPlayingSound(): This method is sufficient in testing audio playback from a button as it will correctly play audio from a given button, and test if the audio has been played correctly using bytes.

testErrorPlayingSound(): This method is sufficient in testing the possibility of an error in playing sound due to an incorrect file name, or null file name. It correctly tests the error that is handled during this special case.

testPlayingMissingSoundFile(): This method is sufficient in testing the possibility of a missing audio file that is supposed to be associated with a button. It correctly tests the error that is handled during this special case.

PlayEditToggleTest:

setup(): This method correctly sets up all necessary fields needed to thoroughly test the PlayEditToggle class, and thus it is respectively sufficient.

testChangingButtonLabel(): This method is sufficient as it correctly changes the button label, and then check the simulator preview to ensure that the change has taken place.

testAddingAudioToButtons(): This method is sufficient as it correctly goes through the process of adding an audio recording to a button. It then checks to see if the audio file has correctly been associated with the button, hence this method is sufficient.

ProfilesPanelTest:

TalkBoxConfigTest:

setUp(): This method correctly sets up all necessary fields needed to thoroughly test the TalkBoxConfig class, and thus it is respectively sufficient.

testSetNumAudioButtons: This method is sufficient as it changes the number of buttons on the simulator preview via the TalkBoxConfig class, and then tests to see if this change took place, hence allowing sufficiency for this method.

testInitialFields(): This method is sufficient in testing the initial fields as it correctly tests all initial fields of the TalkBoxConfig class upon launch of the application.

testJPaneSplits(): This method is sufficient as it correctly creates references to the objects that are created when a TalkBoxConfig. It then tests to see if these references are to the correct objects, hence allowing sufficiency for this testing method.

5.2 Why Test Cases Are Sufficient: TalkBox Simulator Application

TalkBoxSimTest:

Chapter 6

Test Case Implementation

6.1 How Are Test Cases Implemented

RecorderTest:

testInitialFields(): The initial fields of the recorder class are tested as upon launch they should have a certain state. For example, isRecording field should be false when the application is first launched to ensure there is no recording in progress.

testRecording(): The user is able to record from the recording panel, and this method tests the applications recording functionality.

testButtons(): The user is able to change the number of buttons from the recorder panel, and this method allows a test to see if this functionality is working correctly, by preforming a change on the amount of buttons.

SimPreviewTest:

setup(): a configurator object is created and used to create a simulator preview object. These objects are needed to test the PlayEditToggle class.

testingUpdatingButtons(): The user is able to change the number of buttons from the recorder panel, and this method allows a test to see if this functionality is working correctly, by preforming a change on the number of buttons. This updates the number of buttons on the simulator preview.

testPlayingSound(): This method tests playing sound from a button, which is an essential feature of the simulator preview as well as the simulator in general.

testErrorPlayingSound(): This method tests what happens if there is an error playing back audio, for example if a file is null, the correct handled.

testPlayingMissingSoundFile(): This method tests what happens if there is an error playing back audio due to a missing audio file.

PlayEditToggleTest:

setUp(): a configurator object is created and used to create a simulator preview object and recorder object. These objects are needed to test the PlayEditToggle class.

testChangingButtonLabel(): when in edit mode, the user is able to change the labels on each button. This method tests that functionality using the respective JTextFields and JButtons.

testAddingAudioToButtons(): When in edit mode, the user is able to add audio to a button of their choice. This method allows a testing of this feature by using the respective JTextFields and JButtons.

ProfilesPanelTest:

TalkBoxConfigTest:

setUp(): a configurator object is created. This object is needed to test the main configurator class. A call to the main method of the configuration app is allow done in this method.

testSetNumAudioButtons: This method sets the number of audio buttons for the configuration app and tests if this change took place.

testInitialFields(): This method tests all initial fields of the TalkBoxConfiguration application to ensure they are correctly set upon launch of the application.

testJPaneSplits(): This method tests if all objects created by the TalkBoxConfiguration application are correctly initialized. This intasiation of objects takes place in the controlsProfileSplit class and the SimRecorderSplit class.

TalkBoxSimTest:

Chapter 7

Test Coverage

7.1 Test Coverage Metrics