EECS2311 – TalkBox Application

Testing Document

Group 6

Created: Feb 1, 2019

Last Updated: Feb 4, 2019

Draft Version 1.0

1. Introduction

This test document aims to describe the implementation and framework that will be used to facilitate the testing of the TalkBox configuration application and the TalkBox simulator. This includes:

* **Test coverage**: a comprehensive list of all tests used with a brief summary will be included for both the configurator and simulator respectively
* **Test case derivation & sufficiency**: how the test cases were derived will be explained
* **Testing implementation**: an explanation will be provided on how exactly how the test cases are implemented
* **Testing coverage metrics**:

1. Test Coverage

| **Application Testing Checklist** | | | |
| --- | --- | --- | --- |
| Application Name | TalkBox Configurator | | |
| Procedure | | Pass/Fail  (P/F) | Actual Results/Comments |
| Application Functionality | | | |
| Application executable launches successfully | |  |  |
| UI Layout dynamically adjusts to screen resolution/size | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

| **Application Testing Checklist** | | | |
| --- | --- | --- | --- |
| Application Name | TalkBox Simulator | | |
| Procedure | | Pass/Fail  (P/F) | Actual Results/Comments |
| Application Functionality | | | |
| Application executable launches successfully | |  |  |
| Upon running, application should display a file browser to user where they can select a configuration file to load into the simulator | |  |  |
| After selecting a configuration file, the simulator loads the right amount of panel buttons | |  |  |
| After selecting a configuration file, the simulator loads the right amount of toggle buttons | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |

1. Test Case Derivation & Sufficiency

The methodology used to obtain the test cases was to go through the user flow of both the configurator and simulator application from a user perspective. By going step by step through the applications, comprehensive testing can be guaranteed as each step in the user flow is accounted for.

For configurator testing, each possible flow will be walked through, and each step will be documented as a test. For example, there is a flow for creating a preset, loading a preset, and testing a preset. In the case of creating a preset, each step from clicking each button to go to the next screen to clicking a button to upload a sound file will be noted as a test.

The simulator testing will be similar, albeit with less user flows. The simulator tests are derived from the first step being loading in a configuration file from a file browser to the last step being clicking the simulator’s buttons to emit audio.

1. Testing Implementation

The testing will be implemented in the JUnit 5 framework within the Eclipse IDE.

Os version and system

1. Testing Coverage Metrics