Synesthesia Wear: System Verification and Validation Plan for SE 4G06, TRON 4TB6

Team 26, STRONE
Jordan Bierbrier
Azriel Gingoyon
Taranjit Lotey
Udeep Shah
Abraham Taha

November 2, 2022

1 Revision History

Date	Version	Notes
10/31/2022	1.0	Added Section 6 - Unit Test Description
Date 2	1.1	Notes

Contents

List of Tables

[Remove this section if it isn't needed —SS]

List of Figures

[Remove this section if it isn't needed —SS]

2 Symbols, Abbreviations and Acronyms

symbol	**description**
T	Test

[symbols, abbreviations or acronyms – you can simply reference the SRS (Author, 2019) tables, if appropriate —SS]

This document ... [provide an introductory blurb and roadmap of the Verification and Validation plan —SS]

3 General Information

3.1 General Information

Synesthesia Wear goal is to create a wearable product that allows users to get assisted with certain vocal tasks needing attention. These tasks can be generic or custom to the user as needed. The product will use signal processing to gather information and make a calculated prediction of the required action. This will let the user reach a peace of mind, knowing that if an important call is being directed towards the user then the Synesthesia Wear will alert them.

3.2 Objectives

The objective of the document is to prove correctness of the system requirements and the system design documents by using unit and system testing for adequate usability. Often software may have bugs that is experienced by end-user. The tests stated in the document will show signs of mitigating those issues which will ensure the underlying logic for the subsystems. This will be completed by rigorous unit testing on the functional and non-functional requirements. The code and the circuitry tested will be the underlying logic which interact with the database.

3.3 Relevant Documentation

The relevant documents include:

- Hazard Analysis Document
- Systems Design Document

4 Plan

4.1 Verification and Validation Team

The following project members are responsible for all procedures of the verification and validation. Responsibilities can be executing and writing tests :

- Jordan Bierbrier
- Udeep Shah
- Taranjit Lotey
- Abraham Taha
- Azriel Gingoyon

4.2 SRS Verification Plan

The following plans indicate what our team intends to do for SRS verification:

- **Review by teammates:** This plan will make each member go through each SRS and verify if each SRS is still within our usability scope.
- **Review by stakeholders:** This will let our stakeholders to go through each SRS and get their perspective on the usage of the product.
- **Checklist:** This plan involves using previously set checklists in our SRS document which will verify conditions being met.

4.3 Design Verification Plan

The following show our plan to review the Design verification :

- **Review by teammates:** The planned objective is to go through a high-fidelity prototype or functional prototype to verify if the design meets expected data of the SRS.
- **Review by stakeholders:** This plan involves going through the design of the project with our stakeholders to see if the prototype meets expectations set in the SRS.

- **Checklist:** This plan involves using previously set checklists in our SRS document which will verify conditions being met.

4.4 Implementation Verification Plan

The following plans indicate our Implementation verification plan:

- **Static Analysis:** Test plans in sections 5.1.2 and 5.1.3 will be using this for test plans.
- **Code Inspection:** This will be used for the test plans in section 5.1.1.
- **Non-functional Testing:** Non-functional Requirements test plans are written in details in section 5.2.

4.5 Automated Testing and Verification Tools

Automated Testing Tools:

- **Mocha:** Mocha is the oldest testing frameworks for Node.js and hence will be used for our project. It has also evolved with Node.js and the JavaScript language, giving user the opportunity for callbacks, promises and async/await.
- **Mongo Orchestration:** Mongo Orchestration will be used to test our MongoDB database using the MongoDB process management.

Verification Tools:

- **EsLint:** ESLint is a tool for identifying and reporting on patterns found in ECMAScript/JavaScript code, with a goal to make our code consistent and avoiding bugs.

4.6 Software Validation Plan

Currently there is no available data that can help validate the software.

5 System Test Description

5.1 Tests for Functional Requirements

[Include a blurb here to explain why the subsections below cover the requirements. References to the SRS would be good. —SS]

- FRT1

  Control: Manual versus Automatic

  Refrences FR: FR1

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will input randomized sound clips into the device and the device will react x amount of times for x amount of keywords found in the sound clips

- FRT2

  Control: Manual versus Automatic

  Refrences FR: FR1

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will input the same sound clips into the device in differing environments and manually check that the device reacts to the same amount of keywords in each environment

- FRT3

  Control: Manual versus Automatic

  Refrences FR: FR1

Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed: Tester will play same sound at specific distances away from device and check if device picks up sound and reacts

- FRT4

  Control: Manual versus Automatic

  Refrences FR: FR1

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will keep device in a quiet environment and see if device reacts to no noise environments, exposing a false microphone input

- FRT5

  Control: Manual versus Automatic

  Refrences FR: FR2

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will input sound clips into the device while constant background noises are being played i.e with a ambient noise from a car in the background

- FRT6

  Control: Manual versus Automatic

  Refrences FR: FR2

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will speak into the device and determine if keywords are correctly classified by the device by observing that the device gives feedback when the keyword is spoken.

- FRT7

  Control: Manual versus Automatic

  Refrences FR: FR2

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will input sounds from different people saying same words to see if device can correctly classify between people

- FRT8

  Control: Manual versus Automatic

  Refrences FR: FR2

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed: Tester will input words that rhyme with keyword or sound similar to device can correctly classify as not the keyword

- FRT9

  Control: Manual versus Automatic

  Refrences FR: FR3

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will change the keyword classification of the device and input the sound clip with the newly set keyword. Tester will determine if the device correctly reacts to the spoken keyword.

- FRT10

  Control: Manual versus Automatic

  Refrences FR: FR3

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will change the keyword and then check to see that the prior set keyword no longer causes the device to react.

- FRT11

  Control: Manual versus Automatic

  Refrences FR: FR3

  Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed: Tester will delete all classifications and check that device never reacts

- FRT12

  Control: Manual versus Automatic

  Refrences FR: FR3

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will add x amount of classifications sequentially then input those x keywords (in any order) to see device correctly reacts to all keywords

- FRT13

  Control: Manual versus Automatic

  Refrences FR: FR3

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will set the classification then reboot the device and check that the keywords are still correctly reacted to.

- FRT14

  Control: Manual versus Automatic

  Refrences FR: FR4

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will input a sound clip to the device which contains a keyword. The tester will be wearing the device and will manually ensure that the device provides haptic feedback.

- FRT15

  Control: Manual versus Automatic

  Refrences FR: FR4

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will repeat the same test with a sample size of 10 people and check if all the participants can notice the haptic feedback from the device. A total of 9/10 participants must conclude that they have felt the feedback for the test to be a success.

- FRT16

  Control: Manual versus Automatic

  Refrences FR: FR4

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed: Tester will wear device at different orientations and places along wrist and keyword will be inputted to ensure device can provide noticeable feedback to user at different positions on wrist

- FRT17

  Control: Manual versus Automatic

  Refrences FR: FR4

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will wear device on top of clothing article on wrist and keyword will be inputted to verify sufficient feedback from device to the user

- FRT18

  Control: Manual versus Automatic

  Refrences FR: FR5

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will input sound clip to device that contains a specific keyword and will manually ensure that device provides specific corresponding haptic feedback. Test is repeated 10 times to ensure consistent haptic feedback. Test is a success if 9/10 times the correct haptic feedback is recorded.

- FRT19

  Control: Manual versus Automatic

  Refrences FR: FR5

  Initial State:

  Input:

  Output: [The expected result for the given inputs —SS]

  Test Case Derivation: [Justify the expected value given in the Output field —SS]

  How test will be performed: Tester will add multiple keywords to the device. Following, the tester will input sound that matches the keyword and manually ensure each haptic feedback is different.

5.2 Tests for Nonfunctional Requirements

[Tests related to usability could include conducting a usability test and survey. —SS]

- NFRT1

  Type:

  Refrences NFR1

  Initial State:

  Input/Condition:

  Output/Result:

  How test will be performed:

5.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

6 Unit Test Description

[Reference your MIS and explain your overall philosophy for test case selection. —SS] [This section should not be filled in until after the MIS has been completed. —SS]

6.1 Unit Testing Scope

The scope of the unit testing will involve evaluating the microphone, bluetooth, classification, feedback, noise filter, and interface modules to see if they adhere to respective functional and non-functional requirements found in Synesthesia Wear's SRS document.

6.2 Tests for Functional Requirements

[Most of the verification will be through automated unit testing. If appropriate specific modules can be verified by a non-testing based technique. That can also be documented in this section. —SS]

6.2.1 Microphone Module

[Include a blurb here to explain why the subsections below cover the module. References to the MIS would be good. You will want tests from a black box perspective and from a white box perspective. Explain to the reader how the tests were selected. —SS]

1. test-id1

Type	Functional, Dynamic, and Manual.
Initial State	No data in buffer and requesting microphone input.
Input	Sample Recording.
Output	The sample recording in the memory buffer.
Test Case Derivation	The output has to be the digital representation of the input.
How test will be performed	3 Different sample sounds will be supplied near the microphone. Will compare the output with expected output. The test succeeds if all the outputs match the expected outputs within some tolerance.

6.2.2 Bluetooth Module

1. test-id1

Type	Functional, Dynamic, and Manual.
Initial State	Data in buffer and send request received.
Input	Digital sound recording.
Output	The same digital sound recording at the receiver.
Test Case Derivation	The module is a communication module and no change has been made to the data. Hence the data has to be the same as the output.
How test will be performed	A large audio recording will be sent to the data buffer of the sender and send request will be asserted. The receiver should receive the data. The data will be compared manually to check if the test was passed.

2. test-id2

Type	Functional, Dynamic, and Manual.
Initial State	Classification detected asserted.
Input	Sample classification signal asserted on software.
Output	Feedback signal asserted on hardware.
Test Case Derivation	The module is a communication module, and the classification signal received from the software has to tie into its respective feedback signal.
How test will be performed	A classification signal will be asserted manually in the software, its respective feedback signal needs to be asserted in the hardware for the test to pass.

6.2.3 Classification Module

1. test-id1

Type	Functional, Dynamic, and Automatic.
Initial State	Sound classification settings already preconfigured.
Input	Stored sound data in the memory buffer.
Output	Classified sound data.
Test Case Derivation	The output should be digital sound data that has been classified under one of the categories that were preconfigured in the sound classification settings.
How test will be performed	Sound data from the Microphone module testing will be used for this test. The classification code ingrained in the Synesthesia Wear app will automatically try to classify stored sound data in memory. The test succeeds if all outputs are classified under their expected categories.

2. test-id2

Type	Functional, Dynamic, and Manual.
Initial State	Sound classification settings are empty or already preconfigured.
Input	New classification settings.
Output	Classification settings have been changed.
Test Case Derivation	The output should match the new sound classification settings verbatim.
How test will be performed	New sound classification settings will be inputted into a menu on the Synesthesia Wear app and a save button will be used to preserve those settings. The test succeeds if after going back to the sound classification settings menu, the newly inputted settings are displayed.

6.2.4 Feedback Module

1. test-id1

Type	Functional, Dynamic, and Manual.
Initial State	Classification received.
Input	A feedback signal is asserted.
Output	Vibration detected at the end that coincides with the feedback signal.
Test Case Derivation	Tests how our feedback structure performs.
How test will be performed	A feedback signal pertaining to a particular classification is asserted, the output has to be equal to the set vibration specified by the classification.

6.3 Tests for Nonfunctional Requirements

[If there is a module that needs to be independently assessed for performance, those test cases can go here. In some projects, planning for nonfunctional tests of units will not be that relevant. —SS]

[These tests may involve collecting performance data from previously mentioned functional tests. —SS]

6.3.1 Microphone Module

1. test-id1

Type	Dynamic and Manual.
Initial State	No data in buffer.
Input	Sample recording.
Output	The sample recording in the memory buffer.
Test Case Derivation	The output has to be within at least a 95% confidence level of the input.
How test will be performed	3 different sounds found online will be taken and played on some speakers that will project the sounds into the microphone. Taking the initial sound files and the sound data from the microphone, an online software tool will compare the sound data and measure their similarities/confidence level. The test succeeds if the similarities/confidence level is at least 95%.

2. test-id2

Type	Dynamic and Automatic.
Initial State	No data in buffer and the device is powered on.
Input	Random ambient sound.
Output	Continuously updated sound buffer with sampling frequency fs.
Test Case Derivation	Tests if the device is able to continuously update when turned on.
How test will be performed	Random sounds will be inserted into the microphone. The sound buffer will be copied at the frequency of the sampling frequency into a file. The device has to be able to update the sound buffer continuously until the device is turned off to receive a conditional pass. For a complete pass, all the sound data has to have a distortion of less than 5%.

6.3.2 Bluetooth Module

1. test-id1

Type	Dynamic and Manual.
Initial State	Bluetooth device not paired.
Input	Introduce a new bluetooth connection.
Output	Connect with the bluetooth connection in under a minute.
Test Case Derivation	The device has to be able to connect with the hardware easily.
How test will be performed	A new bluetooth device will be introduced to the hardware, on performing the bluetooth connection procedure the connection should be established within a minute for the test to pass.

2. test-id2

Type	Dynamic and Manual.
Initial State	Bluetooth device not connected but paired.
Input	Disconnect bluetooth abruptly.
Output	Auto-reconnection of the bluetooth.
Test Case Derivation	The device has to be able to reconnect without any issues.
How test will be performed	The device will be paired to the hardware initially, by taking the device out of range we will simulate abrupt interruption. It should automatically connect back when back in range, this should not take any longer than 10 seconds after the device is back in range.

6.3.3 Noise Filter Module

1. test-id1

Type	Dynamic and Automatic.
Initial State	Is empty and waiting for an input to process.
Input	Digital data with one or more sounds.
Output	The same digital sound recording but with less noise.
Test Case Derivation	The background noise in the sound file is reduced/removed and a main/singular sound is more notable than others.
How test will be performed	After receiving sound data over bluetooth, Synesthesia Wear's app will automatically send this data over to the corresponding device's noise filtering hardware that will process and return a filtered version of the data. This test passes if it is clear that there is notably less noise in the filtered sound file compared to the original one.

6.3.4 Classification Module

1. test-id1

Type	Dynamic and Automatic.
Initial State	Waiting for sound input and classification settings to be preconfigured.
Input	Sample sounds that fall into classifications and those that do not.
Output	Classification signals asserted for sounds that are in the classification.
Test Case Derivation	Tests the performance and effectiveness of the classification module to be able to distinguish classified and non-classified signals.
How test will be performed	A sample set of different sounds (6 different types of sounds with each one supplied 20 times, each time with a random distortion added to make them all digitally different) will be run through a pre-configured classification set. If the output of the module is correct 90% of the time, it is considered to be a pass.

6.3.5 Feedback Module

1. test-id1

Type	Dynamic and Manual.
Initial State	Classification received.
Input	A feedback signal is asserted.
Output	Vibration detected at the end that coincides with the feedback signal and is not intrusive.
Test Case Derivation	Tests how our feedback structure performs.
How test will be performed	A feedback signal pertaining to a particular classification is asserted such that the output has to be equal to the set vibration specified by the classification. A sample group of 5 will be asked to feel the vibration and then reply if said vibration was sufficient and non-intrusive. If 4 of the 5 answers are yes, the test is passed.

6.3.6 Interface Module

1. test-id1

Type	Structural, Dynamic and Manual.
Initial State	N/A.
Input	Sample group user inputs.
Output	UI should behave as required in terms of appearance and style.
Test Case Derivation	Gathering information on the wants of the users to ensure that the interface adheres to their preferences and that they are pleased with their user experiences.
How test will be performed	A sample group of 5 people will have the opportunity to interact with the UI and device first. They will later be asked questions regarding certain features of the product. These questions are listed in the appendix. If a satisfying result is obtained over the sample group, then the test is passed. This test has multiple subsections where each can be passed or failed separately.

2. test-id2

Type	Dynamic and Automatic.
Initial State	User interface opened up.
Input	User input.
Output	UI response within 1ms.
Test Case Derivation	The device has to be able to respond quickly to any user input.
How test will be performed	It will be too hard to measure the response time manually as most humans have a response time greater than 1ms. Hence this test will be done with the help of helper code which will calculate the time between a user input detected and a corresponding change in the UI.

3. test-id3

Type	Dynamic and Manual.
Initial State	User interface opened up.
Input	User input.
Output	Expected UI response on all the different devices.
Test Case Derivation	The UI has to be able to work the same on all platforms.
How test will be performed	The UI will be installed on different systems (Android, Windows, IOS). If it is capable of all functionality within all the platforms, it receives a pass.

6.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

References

Author Author. System requirements specification. https://github.com/..., 2019.

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

7.2.1 Appearance Requirements

1. How did the finish and look of the device appeal to you?

2. How was the appearance of different pages in the UI software?

Expected answers for pass condition: Satisfied or better for both questions above.

7.2.2 Style Requirements

1. Did you feel that there was consistency between different elements of the UI?

Expected answers for pass condition: Yes.

7.2.3 Ease of Use Requirements

1. Out of 10, how easy do you find it to interact with the UI?

2. Out of 10, what would you rate the usability of the system?

3. What do you find most frustrating about the system?

Expected answers for pass condition: For the first two questions, the average score has to be greater than 7. The last question should not have the same answer repeated between different members. If so, it would suggest an issue with the system.

7.2.4 Personalization and Internationalization Requirements

1. Were you satisfied with the personalization choices of the UI?

Expected answers for pass condition: Should be yes for 85% of the sample group.

7.2.5 Learning Requirements

1. How long did it take you to understand and use the software on your own?

Expected answers for pass condition: Should not be longer than 5 minutes for each person in the sample group.

7.2.6 Understandability and Politeness Requirements

1. How difficult was it to read information off the screen?

2. Were you satisfied with the arrangement of content on the screen?

3. Were you displeased with the language or content used on the UI?

Expected answers for pass condition: For the first condition, the difficulty should not be more than 6 out of 10. For the second condition, it should be yes for 85% of the sample group. For the third condition, it should be no for all members of the sample group.

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

This section deals with what knowledge and experiences each team member will need to acquire so that the capstone project can be completed successfully. With that in mind, before identifying what each member is going to learn/master, some approaches of learning need to be established. Firstly, we believe that one of our main approaches to learning/mastering new skills is by scouring the internet for resources, videos, websites, blogs, or any other notable sources for relevant information. Another approach would be to look through books at McMaster's library to see if there is any applicable details that could be used for this project. Furthermore, one could also master their new skills via practice and trial-and-error by following tutorials and then trying to do them in real-time. Lastly, a final approach could be to find someone with relevant expertise and ask them for advice or some lessons on relevant skills/knowledge that would be beneficial for the project as a whole.

1.

2.