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

April 5, 2023

1   Revision History

Date	Version	Notes
11/1/2022	1.0	Added Section 6 - Unit Test Description
11/2/2022	1.1	Added Section 5 - System Test Description
11/2/2022	1.2	Added Section 4 - Plan
11/2/2022	1.3	Added Section 3 - General Information
11/2/2022	1.4	Added Section 2 and 7 - Symbols and Appendix
4/4/2023	1.5	Updated for feedback and consistency

Contents

List of Figures

List of Tables

2 Symbols, Abbreviations and Acronyms

Symbol	description
Age groups	(15-30, 31-50, 51-75, 75+)

3 General Information

3.1 General Information

Synesthesia Wear's goal is to create a wearable ~~product~~ device that ~~assists users with certain vocal tasks that need attention. These tasks can be generic or custom to the user as needed.~~ improves users' auditory awareness of their surroundings. Furthermore, the product will use signal processing to gather and analyze information to determine the best and most appropriate feedback (via vibrations) to send to the user. As a result, this gives users peace of mind knowing that if ~~an important call is being directed towards them~~ their attention is ever required, Synesthesia Wear will be able to 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. Furthermore, it is often the case where software may have bugs that is then experienced by the end-users. So, the tests stated in the document will describe ways of mitigating those issues and thus ensuring the underlying logic for all subsystems. This will be completed by rigorous unit testing on the functional and non-functional requirements. Lastly, the code and the circuitry being tested will incorporate the underlying logic that interacts with the database.

3.3 Relevant Documentation

The relevant documents include:

- *Hazard Analysis Document*

- *Software Requirements Specification Document*

- *System Design Document*

- *Module Interface Specification Document*

4 Plan

4.1 Verification and Validation Team

The following project members are responsible for all procedures of the verification and validation:

- Jordan Bierbrier - Embedded signal validation

- Udeep Shah - Hardware validation

- Taranjit Lotey - Application validation

- Abraham Taha - Application validation

- Azriel Gingoyon - Embedded signal validation

Responsibilities include executing and writing tests.

4.2 SRS Verification Plan

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

- **Review by teammates:** This plan involves making each member go through our SRS document and verifying that each requirement is still within our usability scope.

- **Review by stakeholders:** Our external stakeholders will get to go through the SRS and gain some perspective on what requirements are supposed to be met while they use the product. Also, insight will be provided by external stakeholders on missing requirements from our SRS.

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

4.3 Design Verification Plan

The following shows our plans for reviewing the Design verification:

- **Review by teammates:** The planned objective is to go through a high-fidelity or functional prototype to verify if the design meets the expected data of the SRS document.

- **Review by stakeholders:** This plan involves going through the design of the project with our stakeholders to see if the prototype meets expectations that were set in the SRS. This will be done three times during the course of the project. Firstly it will be done after POC. Then

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

4.4   Implementation Verification Plan

The following plans indicate our Implementation verification plan:

- **Static Analysis:** All test plans in sections 5 that have code will be using VSCode static analyser extension which reports bugs detected by the Clang Static Analyser.

- **Code Inspection:** This will be used for all relevant test plans that incorporate code in section 5.

- **Non-functional Testing:** Test plans for Non-functional Requirements are written in detail within section 5.2 of this document.

4.5   Automated Testing and Verification Tools

Automated Testing Tools:

- **Mocha:** Mocha is the oldest testing framework for Node.js and hence will be used for our project. It has also evolved with Node.js and the JavaScript language, giving users 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 system.

Verification Tools:

- **ESLint:** ESLint is a tool for identifying and reporting on patterns found in ECMAScript/JavaScript code with a purpose of making code consistent while 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

- FRT1

  Control: Manual

  References FR: FR2

  Initial State: Powered on device with chosen classified keywords "Jordan, Abraham, Azriel, John" and no input signal.

  Input: Sound recordings containing 4 keywords is supplied to the device.

  Output: Device is able to classify the keywords.

  Test Case Derivation: Device needs to react once per keyword therefore 4 keywords should give 4 reactions.

  How test will be performed: Tester will input randomized digital sound clips into the device and the device will be able to classify the 4 keywords.

- FRT2

  Control: Manual

  References FR: FR1, FR2, FR8

  Initial State: Powered on device with a chosen classified keyword and no input signal

  Input: The classified keyword is spoken in different environments (quiet room, busy library, quiet outdoors and a construction site)

  Output: Device is able to classify the keyword.

  Test Case Derivation: Device should be able to classify sound in different environments ~~and still produce a reaction~~.

  How test will be performed: Tester will speak the same keyword "Jordan" into the device in differing environments and manually check

4

that the device's output ~~reacts to the same amount of keywords in each environment~~.

- FRT3

  Control: Manual

  References FR: FR1, FR8

  Initial State: Powered on device with chosen classified keywords and no input signal.

  Input: Sound clip with varying intensities.

  Output: Device is able to classify the keyword at different sound intensities.

  Test Case Derivation: Device should still be capable of producing a reaction even if the input signal is diminished.

  How test will be performed: Tester will play the same sound with different intensities (45dB, 50dB, 60dB, 75dB) at a distance of 0.5m away from device and check if device picks up sound.

- FRT4

  Control: Automatic

  References FR: FR1

  Initial State: Powered on device with no input signal.

  Input: N/A

  Output: Device should output nothing.

  Test Case Derivation: Because there is no input to the device there should also not be any output.

  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

  References FR: FR2, FR8

Initial State: Powered on device with chosen classified keywords and no input signal

Input: Sound clips induced with digital noise

Output: Correct classifications

Test Case Derivation: Device should be able to classify the input data even if there is noise in the input.

How test will be performed: Tester will load sound clips (induced with random noise) manually onto device.

- FRT6

  Control: Manual

  References FR: FR1, FR2, FR6, FR7, FR8

  Initial State: Powered on device with chosen classified keyword "Jordan" and no input signal.

  Input: The classified keyword verbally spoken to the device.

  Output: Haptic feedback from device.

  Test Case Derivation: Device should provide a haptic feedback when it detects a keyword.

  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

  References FR: FR1, FR2, FR8

  Initial State: Powered on device with chosen classified keywords and no input signal

  Input: The classified keyword verbally spoken to the device by a male and a female around the age of 22.

  Output: Haptic feedback

  Test Case Derivation: The device should be able to classify the keywords regardless of inputted voice.

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

  References FR: FR3, FR8

  Initial State: Powered on device with chosen classified keywords and no input signal

  Input: Sound clips

  Output: N/A

  Test Case Derivation: The sound recognition algorithm should be able to correctly classify similar words as not being the keyword and not provide a reaction.

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

- FRT9

  Control: Manual

  References FR: FR2, FR3, FR4

  Initial State: Powered on device with classified keywords

  Input: New keywords classifications, Sound clip

  Output: Haptic Feedback

  Test Case Derivation: Device should recognize the new keywords and correctly react to the new words.

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

- FRT10

  Control: Manual

References FR: FR2, FR3, FR4

Initial State: Powered on device with classified keywords

Input: New keywords classifications, Sound clip

Output: No output

Test Case Derivation: Once a keyword is no longer classified the device should no longer provide an output for that keyword.

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

- FRT11

  Control: Manual

  References FR: FR3

  Initial State: Powered on device with classified keywords

  Input: Remove all classified keywords, Sound clip

  Output: No output

  Test Case Derivation: If there are no chosen keywords classified the device should not produce any output.

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

- FRT12

  Control: Manual

  References FR: FR2, FR5

  Initial State: Powered on device with no inputs

  Input: Keyword Classifications, Sound clip

  Output: Haptic Feedback

  Test Case Derivation: As keywords are classified the device should now react to those chosen keywords.

  How test will be performed: Tester will add 3 new classifications sequentially then say those 3 keywords (in any order) to see device correctly reacts to all keywords

- FRT13

  Control: Manual

  References FR: FR4

  Initial State: Powered on device with no inputs

  Input: Keyword Classifications, Sound clip, Reboot Device

  Output: Haptic Feedback

  Test Case Derivation: Device should still retain all chosen keywords in the event of a reboot or a power off.

  How test will be performed: Tester will set the classification then reboot the device and check that the device is classification system is meeting correctness requirements.

- FRT14

  Control: Manual

  References FR: FR6

  Initial State: Powered on device with keyword classifications

  Input: Trigger haptic feedback manually.

  Output: Haptic Feedback

  Test Case Derivation: Haptic feedback should be outputted by the device when detecting a keyword.

  How test will be performed: Tester will trigger the haptic feedback manually ~~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

  References FR: FR6

  Initial State: Powered on device with keyword classifications

  Input: Sound Clip

  Output: Haptic Feedback

Test Case Derivation: Haptic feedback needs to be easily recognizable by the average user.

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

  References FR: FR6

  Initial State: Powered on device with keyword classifications

  Input: Sound Clip

  Output: Haptic Feedback

  Test Case Derivation: Users need to be able to feel the haptic feedback through numerous wearing patterns of the device.

  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

  References FR: FR6

  Initial State: Powered on device with keyword classifications

  Input: Sound Clip

  Output: Haptic Feedback

  Test Case Derivation: Haptic feedback should be strong enough that it can be felt through clothing articles.

  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

  References FR: FR1, FR2, FR7

  Initial State: Powered on device with keyword classifications

  Input: Sound Clip

  Output: Haptic Feedback

  Test Case Derivation: When device recognizes a keyword it should react with haptic feedback in realtime.

  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

  References FR: FR1, FR2, FR4, FR5, FR7

  Initial State: Powered on device with no keyword classifications

  Input: Keyword classifications, Sound Clip

  Output: Distinct haptic feedback for distinct keywords.

  Test Case Derivation: Device should have compatibility for five different keywords and should provide unique haptic feedback patterns for each of the keywords.

  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

- NFRT1

  Type: Performance

  References NFR1

Initial State: Unloaded Application

Input/Condition: Open Application

Output/Result: Application Loads up

How test will be performed: Participants will open the application and manually check to see that a home page is loaded on opening of the application. Users should not be required to provide any input after initiating the opening of the application to get navigated to the home page

- NFRT2

  Type: Performance

  References NFR1

  Initial State: Unloaded Application

  Input/Condition: Open Application, click pair button

  Output/Result: Application pairing page

  How test will be performed: Participants should be able to identify the option to pair the external device with the application within the home page. Users will be tested to see if they can identify the pairing option within 10 seconds. Test will be performed manually with a tester observing that the users can meet these constraints.

- NFRT3

  Type: Manual

  References NFR1

  Initial State: Unloaded Application

  Input/Condition: Open Application

  Output/Result: N/A

  How test will be performed: Application will be run and the tester will manually check to ensure that the different buttons are color coded based on similar functionalities.

- NFRT4

  Type: Performance

References NFR1

Initial State: Open Application

Input/Condition: Click Pair Button

Output/Result: Application goes to pairing page

How test will be performed: Tester will launch the application and check to see if clicking the pair button on the homepage proceeds the application to a pairing page.

- NFRT5

  Type: Performance

  References NFR1

  Initial State: Open Application

  Input/Condition: Click Keyword Selection Button

  Output/Result: Application goes to keyword selection page

  How test will be performed: Tester will launch the application and check to see if clicking the keyword selector button proceeds the application from the homepage to the keyword configurations page.

- NFRT6 - Out of scope

  Type: Manual

  References NFR1

  Initial State: Turned off wearable device

  Input/Condition: N/A

  Output/Result: N/A

  How test will be performed: A survey will be conducted to a group of 10 participants and they will rate the finish of the product out of 10, they will also rate the accessibility/findability of the buttons on a scale of 10. They will also be asked about the distinguishability of the charging port and its ease of use. Answers will be averaged out and an average score of 8 will be considered a pass for the test.

- NFRT7 - Out of scope

  Type: Manual

References NFR2

Initial State: Opened Application

Input/Condition: N/A

Output/Result: N/A

How test will be performed: The style requirements of the device/application will all be tested through a participant study where they will be asked the following questions see appendix 7.2.7. An average score of 4 out of all the questions from the participants will be considered a pass.

- NFRT8 - Out of scope

    Type: Performance

    References NFR3, NFR7

    Initial State: Unopened Application, Unpaired device

    Input/Condition: Open application, click pair button on both device and application

    Output/Result: Paired Screen

    How test will be performed: The final product will be given to users from multiple different age groups and asked to open the application and connect the wearable device to the application over bluetooth. Testers will take note of any issues that arise and record them.

- NFRT9

    Type: Security Test

    References NFR3

    Initial State: Pairing Screen

    Input/Condition: Begin pairing

    Output/Result: Alert Message

    How test will be performed: Tester will attempt to pair the device without pressing the pairing button on the device. Tester will then check to see if the application alerts the user that a device is not found.

- NFRT10 - Removed due to design decisions
  ~~**Type:** Security Test~~
  ~~**References** NFR3~~
  ~~**Initial State:** Login page of application~~
  ~~**Input/Condition:** Username and Password~~
  ~~**Output/Result:** Alert Message~~
  ~~**How test will be performed:** Tester will attempt to log in with an unregistered account. Check to see that the program correctly identifies that the account does not exist and prompts the users to try again or register an account.~~

- NFRT11

  Type: Stress Test

  References NFR3

  Initial State: Keyword configuration page

  Input/Condition: Invalid Keyword

  Output/Result: Alert Message

  How test will be performed: Tester will attempt to configure an unrecognizable keyword. Check to see that the program alerts the users that the keyword is not supported.

- NFRT12

  Type: Manual

  References NFR4

  Initial State: Application Home Screen

  Input/Condition: Click keyword selection button

  Output/Result: Keyword configuration screen

  How test will be performed: Tester will check that the application has a page that allows the configuration of different keywords. This will be done manually and newly inputted keywords will be spoken and the appropriate reaction from the device will be recorded.

- NFRT13 - Out of scope

  Type: Manual

References NFR4

Initial State: Setup page

Input/Condition: Preferred Language

Output/Result: Application translated to preferred language

How test will be performed: Tester will check that the application prompts the user to choose a preferred language when setting up the device. This will be done by manually setting up a new device and visually checking if the prompt appears.

- NFRT14 - Out of scope

  Type: Manual

  References NFR4

  Initial State: Preferences Page

  Input/Condition: Change Language

  Output/Result: Application translated to chosen language

  How test will be performed: Tester will also check to see if a user has the option to change preferred language on an already set-up device. This will also be manually checked and verified by the tester checking that the language has changed.

- NFRT15 - Out of scope

  Type: Manual

  References NFR4

  Initial State: Base user manual

  Input/Condition: Team translates manuals

  Output/Result: translated manuals

  How test will be performed: The team will hire translators to ensure that each of the primary languages are correctly translated from a base user manual. The base user manual will be written in english.

- NFRT16

  Type: Performance

  References NFR5

Initial State: Unopened application, turned off device

Input/Condition: Open application, turn on device, pair device

Output/Result: Paired device and application

How test will be performed: Participants will be given device and application and timed to see if they can set up and use them within 5 minutes. This test will be conducted with 4 people from each age group, and will be considered pass if 3/4 participants from each age group can use the device within 5 minutes.

- NFRT17 - Out of scope

  Type: Visual

  References NFR6

  Initial State: Opened Application

  Input/Condition: N/A

  Output/Result: N/A

  How test will be performed: Test will be conducted with 4 participants from each age group. Participants will be given an application and asked which icon corresponds to which action/function. For each icon they answer correctly, they will receive one point. A total of 5 icons will be asked see appendix 7.2.8. A pass is achieved if all 5 icons are named by 3/4 participants from each age group.

- NFRT18

  Type: Recovery Testing

  References NFR8

  Initial State: Paired device

  Input/Condition: Take device out of range and back into range

  Output/Result: Device repairs

  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. This test will be performed 10 times with 2

different devices and it should be able to connect back more than 90% of the time for a pass.

- NFRT19

  Type: Performance

  References NFR9

  Initial State: Device preconfigured with keywords

  Input/Condition: Keyword

  Output/Result: Haptic Feedback

  How test will be performed: Tester will input keyword sound and record the amount of time to receive haptic feedback from the wearable device. Timer begins as soon as keyword sound is played and stopped when haptic feedback begins. Test is considered a pass if the time recorded for 8/10 measurements is less than 1 second.

- NFRT20

  Type: Performance

  References NFR9

  Initial State: Application home screen

  Input/Condition: Button click

  Output/Result: Corresponding page

  How test will be performed: Helper code will calculate the time between a user input detected and a corresponding change in the UI. The helper code will simulate 100 user inputs spread over all possible places of correct user inputs and record the response times. If the average of the response times is less than 1ms then the test is considered to be a pass.

- NFRT21

  Type: Performance

  References NFR9

  Initial State: Application pairing screen

  Input/Condition: Being pairing

Output/Result: Paired device

How test will be performed: A set of 5 new bluetooth devices will be introduced to the hardware, on performing the bluetooth connection procedure the connection should be established within a minute for all the devices for the test to pass.

- NFRT22

  Type: Recovery

  References NFR9

  Initial State: Paired Devices

  Input/Condition: Devices taken out of range and brought back into range

  Output/Result: Repairing of devices

  How test will be performed: A set of 5 pre-existing bluetooth devices will be brought into the pairing range of the device. A tester will time how long it takes each of the devices to reconnect to the application. For this test to pass the average time of all 5 should be $\leq$ 10 seconds.

- NFRT23

  Type: Visual

  References NFR10

  Initial State: Turned off device

  Input/Condition: N/A

  Output/Result: N/A

  How test will be performed: Visual inspection of finished devices should yield no sight of the battery.

- NFRT24

  Type: Stress Testing

  References NFR11

  Initial State: Device with preconfigured keywords

  Input/Condition: Sound clips

Output/Result: Haptic Feedback

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.

- NFRT25 - Out of scope

  Type: Stress Test

  References NFR12

  Initial State: Fully charged device

  Input/Condition: Sound clips

  Output/Result: Haptic Feedback

  How test will be performed: The battery will be fully charged on 10 separate devices and a group of 10 testers will use the device for a consecutive 15 hours. The time the devices run out of battery will be recorded. The average battery life of all 10 devices should be > 12 hours for the test to be considered a pass.

- NFRT26

  Type: Load Testing

  References NFR12

  Initial State: Powered on device with preconfigured keywords

  Input/Condition: Sound clips

  Output/Result: Haptic Feedback

  How test will be performed: Tester will power on the device for a duration of 5 hours. At 5 randomized time intervals throughout the 5 hours, the tester will insert a keyword and record whether or not the device reacts. Device should react at each interval for the test to be considered a pass.

- NFRT27 - Out of scope

  Type: Upgrade and Installation Test

  References NFR12

Initial State: Working Application

Input/Condition: Application goes down

Output/Result: Email sent to developers

How test will be performed: Helper code will be used to check if the application is up. If the application goes down the helper code will alert the development team. If the development team does not receive an alert over a one year period then the test will be considered a pass.

- NFRT28

  Type: Upgrade Testing

  References NFR15

  Initial State: Application keyword selection page

  Input/Condition: add 10 Keyword

  Output/Result: Keyword added successfully

  How test will be performed: Testers will manually check if the product supports up to 10 keywords 2 years after the launch of the device.

- NFRT29 - Out of scope

  Type: Visual and Stress Testing

  References NFR16

  Initial State: Turned off Device

  Input/Condition: N/A

  Output/Result: N/A

  How test will be performed: Product will be stress tested by a team of testers. Stress tests can include battery drain/charge cycles, Material wear and tear, extensive microphone use etc. Based on results the device should be rated for a lifetime of 5 years for the tests to pass

- NFRT30 - Out of scope

  Type: Performance

  References NFR17

  Initial State: Unpaired Device

Input/Condition: Setup Device

Output/Result: N/A

How test will be performed: Using a sample group of 10 participants each member will be asked to use the device for a duration of 3 days. Participants will then be asked to submit any times that the device inhibited their day to day lives. If 8/10 of the participants did not find the device to inhibit their daily lives then the test is considered a pass.

- NFRT31

  Type: Visual

  References NFR17

  Initial State: Turned off Device

  Input/Condition: Adjust size

  Output/Result: Size should be adjustable from 6-8 inches

  How test will be performed: Device will be manually inspected to ensure that it is adjustable to a multitude of different wrist sizes, 6 inches to 8.5 inches, (wrist sizes ranging from small to large).

- NFRT32

  Type: Performance and Installation Testing

  References NFR17

  Initial State: Uninstalled Application

  Input/Condition: Install Application

  Output/Result: Application Installs Correctly

  How test will be performed: Testers will try to download the application and pair a device on both an IOS and Android device. If both the devices successfully download and pair then the test is considered a pass.

- NFRT33 - Out of scope

  Type: Upgrade Test

  References NFR20

Initial State: Turned off device

Input/Condition: Turn on device, sound clip

Output/Result: Haptic Feedback

How test will be performed: Tester will manually try and use the device 24 hours after the software has been updated. If the device functions correctly then the test is considered a pass.

- NFRT34

  Type: Visual

  References NFR25

  Initial State: Unopened Application

  Input/Condition: Open Application

  Output/Result: N/A

  How test will be performed: A diverse sample group (varying in religion, ethnicity, culture) will be told to examine the code and will be asked to provide any relative feedback about references pertaining to their cultures. If all participants do not find any references the test will be considered a pass.

- NFRT35 - Out of scope

  Type: Regulation Testing

  References NFR26

  Initial State: Ready to ship product

  Input/Condition: N/A

  Output/Result: N/A

  How test will be performed: An independent team of lawyers will be used to check that the application, device, and the user manual all comply with their corresponding regulations (including licensing agreements). If the team does not find any issues then the test is considered a pass.

- NFRT36 - Moved to safety requirements
  Type: System Isolation Auto shut-off
  References

Initial State: ~~Powered On~~
Input/Condition: ~~N/A~~
Output/Result: ~~N/A~~
How test will be performed: ~~We will create an environment that will result in similar characteristics as rain in the outdoors.~~

5.3 Tests for Safety requirements

Added new test cases for safety requirements

- SRT1

  Type: Electrical system isolation

  References SIR1

  Initial State: Powered On

  Input/Condition: N/A

  Output/Result: N/A

  How test will be performed: The device will be powered on and using a handheld multimeter will be checked for live current at all possible contact locations with the user's hand.

- SRT2

  Type: System Auto shut-off

  References SIR2

  Initial State: Powered On

  Input/Condition: Overvoltage supplied to VCC

  Output/Result: Device shuts off

  How test will be performed: We will create an environment that will result in similar characteristics as the circuit getting shorted or overvoltage.

- SRT3

  Type: Device heat dissipation

  References SIR3

  Initial State: Powered On

Input/Condition: N/A

Output/Result: N/A

How test will be performed: The device will be powered on and using a thermocouple will be checked heat that is 5 degrees above the environment.

- SRT4

  Type: Reboot capability

  References SIR4

  Initial State: Powered On

  Input/Condition: Remove and reconnect the power

  Output/Result: Reboot as required

  How test will be performed: The device will be powered on, the power will be disconnected and reconnected without switching off the device.

- SRT5 - Out of scope

  Type: Application Testing

  References ACR1

  Initial State: Application opened

  Input/Condition: New vibration intensity changes

  Output/Result: Vibration setting should be updated

  How test will be performed: The device will be powered on connected to the application, then the user will try to update the vibration intensity.

- SRT6

  Type: Application Testing

  References ACR2

  Initial State: Application opened

  Input/Condition: Invalid user input

  Output/Result: Error message

How test will be performed: The device will be powered on connected to the application, then the user will try to perform invalid operations on the application. The application should not break and instead should output an error message of invalid input.

- SRT7 - Out of scope

  Type: Application testing

  References ACR3

  Initial State: Unopened application installed on incompatible device.

  Input/Condition: Open application

  Output/Result: Error message

  How test will be performed: The tester will load the application on to a device that is not compatible. The application should show an error message that the device is not compatible.

- SRT8

  Type: Communication testing

  References IR3

  Initial State: Opened application

  Input/Condition: Device connected with outdated settings

  Output/Result: New settings are automatically updated on connection

  How test will be performed: The tester will pair a device with outdated settings, the application should automatically sync with the device.

- SRT9

  Type: Application testing

  References IR4

  Initial State: Opened application

  Input/Condition: Same vibration settings for two different classes

  Output/Result: Error message

How test will be performed: The tester will try to have the same vibration setting for two different classifications. The device should pop up an error message that two classifications cannot have the same vibrations.

- SRT10

  Type: Application Testing

  References IR5

  Initial State: Opened application with device connected

  Input/Condition:

  Output/Result: No change in settings

  How test will be performed: The tester will move around the application, but it should only send a sync request when there is a change in settings.

- SRT11

  Type: Hardware testing

  References IR6

  Initial State: Powered device

  Input/Condition: Device supplied with extremely high intensity (90dB) sounds

  Output/Result: Device does a continuous green flashing

  How test will be performed: The tester will play a loud, sound the device should recognize its inability to classify correctly and alert the user.

- SRT12

  Type: Application testing

  References IR7

  Initial State: Opened application with device connected

  Input/Condition: Application receives an incorrect Bluetooth transmission.

  Output/Result: Application displays an error message.

How test will be performed: The tester will manually send a false Bluetooth signal from the device. The application should reject it.

- SRT13

    Type: Communication testing

    References IR8

    Initial State: Powered device connected with the application

    Input/Condition: Application disconnected

    Output/Result: Device vibrates

    How test will be performed: The tester will manually disconnect the device from the application, the device shall notice this and alert the user.

5.4 Traceability Between Test Cases and Requirements

Tests	Requirements
FRT1	FR-2
FRT2	FR-1, FR-2, FR-8
FRT3	FR-1, FR-8
FRT4	FR-1
FRT5	FR-2, FR-8
FRT6	FR-1, FR-2, FR-6, FR-7, FR-8
FRT7	FR-1, FR-2, FR-8
FRT8	FR-3, FR-8
FRT9	FR-2, FR-3, FR-4
FRT10	FR-2, FR-3, FR-4
FRT11	FR-3
FRT12	FR-2, FR-5
FRT13	FR-4
FRT14	FR-6
FRT15	FR-6

FRT16	FR-6
FRT17	FR-6
FRT18	FR-1, FR-2, FR-7
FRT19	FR-1, FR-2, FR-4, FR-5, FR-7
NFRT1	NFR-1
NFRT2	NFR-1
NFRT3	NFR-1
NFRT4	NFR-1
NFRT5	NFR-1
NFRT6	NFR-1
NFRT7	NFR-2
NFRT8	NFR-3
NFRT9	NFR-3
NFRT10	NFR-3
NFRT11	NFR-3
NFRT12	NFR-4, NFR-25
NFRT13	NFR-4
NFRT14	NFR-4
NFRT15	NFR-4
NFRT16	NFR-5, NFR-7
NFRT17	NFR-6
NFRT18	NFR-8
NFRT19	NFR-9
NFRT20	NFR-9
NFRT21	NFR-9
NFRT22	NFR-9
NFRT23	NFR-10
NFRT24	NFR-11
NFRT25	NFR-12
NFRT26	NFR-12

NFRT27	NFR-12
NFRT28	NFR-15
NFRT29	NFR-16
NFRT30	NFR-17
NFRT31	NFR-17
NFRT32	NFR-17, NFR-21
NFRT33	NFR-20
NFRT34	NFR-25
NFRT35	NFR-26, NFR-27
SRT1	SR1
SRT2	SR2
SRT3	SR3
SRT4	SR4
SRT5	ACR1
SRT6	ACR2
SRT7	ACR3
SRT8	IR3
SRT9	IR4
SRT10	IR5
SRT11	IR6
SRT12	IR7
SRT13	IR8

Table 1: Traceability Between Test Cases and Requirements

6 Unit Test Description

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*. Furthermore, the modules of this section can be found in Synesthesia Wear's *MIS Document* where information on their Uses, Syntax, and Semantics are explained in greater detail.

6.2 Tests for Functional Requirements

6.2.1 Microphone Module

1. UT1

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.

Table 2: UT1 Description

6.2.2 Bluetooth Module

1. UT2

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.

Table 3: UT2 Description

2. UT3

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.

Table 4: UT3 Description

6.2.3 Classification Module

1. UT4

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.

Table 5: UT4 Description

2. UT5

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.

Table 6: UT5 Description

6.2.4 Feedback Module

1. UT6

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.

Table 7: UT6 Description

6.3 Tests for Nonfunctional Requirements

6.3.1 Microphone Module

1. UT7

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%.

Table 8: UT7 Description

2. UT8

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%.

Table 9: UT8 Description

6.3.2 Bluetooth Module

1. UT9

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.

Table 10: UT9 Description

2. UT10

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.

Table 11: UT10 Description

6.3.3 Noise Filter Module

1. UT11

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.

Table 12: UT11 Description

6.3.4 Classification Module

1. UT12

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.

Table 13: UT12 Description

6.3.5 Feedback Module

1. UT13

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.

Table 14: UT13 Description

6.3.6 Interface Module

1. UT14

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.

Table 15: UT14 Description

2. UT15

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.

Table 16: UT15 Description

3. UT16

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.

Table 17: UT16 Description

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.

7.2.7 Additional Material for Non-Functional Requirement Test 7

Answer all questions on a scale of 1-6

1. Does the application contain consistent buttons throughout?

2. Does the device have a uniform finish and consistent buttons?

3. Do the colors of the application and the device follow a consistent color palette?

4. Does the device automatically begin a pairing process when the corresponding buttons are inputted?

5. Is the differing parts of the UI including different pages, buttons etc. easily identifiable and responsive?

7.2.8 Additional Material for Non-Functional Requirement Test 17

Specific icons participants will be asked to name:

- Bluetooth/setup icon

- Settings icon

- Go back icon

- Login icon

- Logout icon

Appendix — Reflection

This section deals with what knowledge, experiences, and approaches each team member will need to acquire/pursue so that the verification and validation of Synesthesia Wear can be completed successfully. To start off, the collective knowledge and skills that the team would need to acquire includes dynamic testing knowledge, static testing knowledge, Mocha skills, Mongo Orchestration skills, and ESLint skills. With that in mind, an approach that we could use to learn/master the new knowledge or skills would be scouring the internet for resources, videos, websites, blogs, or any other notable sources for relevant information. Another approach would be to look through relevant books at McMaster's library or any other library to see if there is any applicable details that could be used for this document. 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 completion of the Verification and Validation of our project.

Jordan Bierbrier

The knowledge that Jordan needs to acquire during this project is Static Testing knowledge. This is because he needs to be able to theoretically test whether the signal processing aspects of the product will work based on an examination of the code. To further clarify, the hardware needed for the signal processing may not always be set up for testing. So, Jordan should learn Static Testing knowledge so that he can still test at all times even when the hardware is not set up and the code is not running. To add on, although the actual results of the signal processing may not be able to be verified, Jordan can still test the logic of the code such as the flow of signals over bluetooth from the bracelet to the app. With all this in mind, the approaches Jordan should take would include scouring the internet for resources on static testing knowledge as well as asking experts with testing experience to see if they have any advice. These two approaches were chosen because they would be the most effective and efficient methods of learning static testing knowledge.

Azriel Gingoyon

The knowledge that Azriel needs to acquire during this project is Mocha skills. This is because he needs to test whether the hardware components are functioning properly and are able to communicate with the app as expected. To further clarify, all of the hardware needs to be incorporated properly so that everything is working properly. So, if the design and hardware are all good, the communication to the hardware should not have too many issues, and so this testing platform would help with that. With all this in mind, the approaches Azriel should take would include scouring the internet for resources on Mocha as well as trying trial-and-error/practice to further solidify the Mocha skills that he is learning.

Taranjit Lotey

To search the McMaster professor profolios form previous years to find experts in our relevant topics which require attention. Approach one would be asking the professor of our Captsone project to suggest professors which would align with our needs the best. Approach two would be going back to previous courses and find out which professors taught topics related to the issue/interest at hand.

Udeep Shah

Find information required to successfully finish research and understanding the topic I would reach for the help of relatable books. This would be possible by using online library search engines to find matching summaries. Also use the help of online resources to find which books have problems or solutions related to the one at hand.

Abraham Taha

The knowledge that Abraham needs to acquire during this project is Dynamic Testing knowledge. This is because he needs to be able to continuously test whether the app aspects of the product are working as expected. To further clarify, as the app is continuously being developed to meet all requirements and preferences of our users, it is essential to ensure that other aspects of the app are still functioning properly. With all this in mind, the approaches

Abraham should take would include scouring the internet for resources on dynamic testing knowledge as well as asking experts with testing experience to see if they have any advice. These two approaches were chosen because they are the most effective and efficient methods of learning dynamic testing knowledge.