

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

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	iii
3	General Information	1
3.1	General Information	1
3.2	Objectives	1
3.3	Relevant Documentation	1
4	Plan	2
4.1	Verification and Validation Team	2
4.2	SRS Verification Plan	2
4.3	Design Verification Plan	2
4.4	Implementation Verification Plan	3
4.5	Automated Testing and Verification Tools	3
4.6	Software Validation Plan	3

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
(?) 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.