Project Title: System Verification and Validation Plan for Mechatronics Engineering

Team #1, Back End Developers
Jessica Bae
Oliver Foote
Jonathan Hai
Anish Rangarajan
Nish Shah
Labeeb Zaker

October 31, 2022

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
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]

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

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

3 General Information

3.1 Summary

[Say what software is being tested. Give its name and a brief overview of its general functions. —SS]

3.2 Objectives

[State what is intended to be accomplished. The objective will be around the qualities that are most important for your project. You might have something like: "build confidence in the software correctness," "demonstrate adequate usability." etc. You won't list all of the qualities, just those that are most important. —SS]

3.3 Relevant Documentation

[Reference relevant documentation. This will definitely include your SRS and your other project documents (design documents, like MG, MIS, etc). You can include these even before they are written, since by the time the project is done, they will be written. —SS]

Author (2019)

4 Plan

[Introduce this section. You can provide a roadmap of the sections to come. —SS]

4.1 Verification and Validation Team

[Your teammates. Maybe your supervisor. You shoud do more than list names. You should say what each person's role is for the project's verification. A table is a good way to summarize this information. —SS]

4.2 SRS Verification Plan

[List any approaches you intend to use for SRS verification. This may include ad hoc feedback from reviewers, like your classmates, or you may plan for something more rigorous/systematic. —SS]

[Maybe create an SRS checklist? —SS]

4.3 Design Verification Plan

[Plans for design verification —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

4.4 Verification and Validation Plan Verification Plan

[The verification and validation plan is an artifact that should also be verified. —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

4.5 Implementation Verification Plan

[You should at least point to the tests listed in this document and the unit testing plan. —SS]

[In this section you would also give any details of any plans for static verification of the implementation. Potential techniques include code walkthroughs, code inspection, static analyzers, etc. —SS]

4.6 Automated Testing and Verification Tools

[What tools are you using for automated testing. Likely a unit testing framework and maybe a profiling tool, like ValGrind. Other possible tools include a static analyzer, make, continuous integration tools, test coverage tools, etc. Explain your plans for summarizing code coverage metrics. Linters are another important class of tools. For the programming language you select, you should look at the available linters. There may also be tools that verify that coding standards have been respected, like flake9 for Python. —SS]

[If you have already done this in the development plan, you can point to that document. —SS]

[The details of this section will likely evolve as you get closer to the implementation. —SS]

4.7 Software Validation Plan

[If there is any external data that can be used for validation, you should point to it here. If there are no plans for validation, you should state that here. —SS]

[You might want to use review sessions with the stakeholder to check that the requirements document captures the right requirements. Maybe task based inspection? —SS]

[This section might reference back to the SRS verification section. —SS]

5 System Test Description

5.1 Tests for Functional Requirements

[Subsets of the tests may be in related, so this section is divided into different areas. If there are no identifiable subsets for the tests, this level of document structure can be removed. —SS]

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

5.1.1 Area of Testing1

[It would be nice to have a blurb here to explain why the subsections below cover the requirements. References to the SRS would be good here. If a section covers tests for input constraints, you should reference the data constraints table in the SRS. —SS]

Title for Test

1. test-id1

   Control: Manual versus Automatic

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:

2. test-id2

   Control: Manual versus Automatic

   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:

5.1.2 Area of Testing2

...

5.2 Tests for Nonfunctional Requirements

[The nonfunctional requirements for accuracy will likely just reference the appropriate functional tests from above. The test cases should mention reporting the relative error for these tests. Not all projects will necessarily have nonfunctional requirements related to accuracy —SS]

[Tests related to usability could include conducting a usability test and survey. The survey will be in the Appendix. —SS]

[Static tests, review, inspections, and walkthroughs, will not follow the format for the tests given below. —SS]

5.2.1 Usability

Title for Test

1. test-id1

   Type: Manual, Static

   Initial State: The device is turned on and a question prompt is generated on screen.

   Input/Condition: Users are asked to answer the questionnaire set without assistance in less than 1 minute.

   Output/Result: All users are able to answer their prompt successfully.

   How test will be performed: A test group of people with age of 40 or older to be asked to receive a powered-on device and asked to respond to a set of 3 questions in 1 minute. These questions might relate to EMA studies but it's not a requirement that the test questions are EMA research specific.

2. test-id2

   Type: Manual, Static

   Initial State: The device is turned on and left on idle mode.

   Input: Users are asked to wear the device for 1 day.

   Output: Users fill out a survey form regarding the comfort of the deivce on their body.

   How test will be performed: At the end of the 1 day cycle, users will be asked to fill out an online form indicating how comfortable they felt the device was regards to weight, shape, stability, etc.

3. test-id3

   Type: Manual, Static

   Initial State: The device is turned on off.

   Input: Users are asked to clean the device.

   Output: The device is cleaned without any harm to it.

How test will be performed: A group of university students to be asked to read the manual and clean the device. All students should be able to understand the manual and clean the device succesfully without any assistance.s

5.2.2 Performance

1. test-id1

   Type:

   Initial State: The device has collected a complete set of data.

   Input: The device is connected to the host software and connection is established for data transfer.

   Output: Data transfer to finish within the specified parameter, TRASFERTIME.

   How test will be performed: 2 group members of Back End Developers to each perform data trasnfer test 3 times and record the total transfer time for each execution. All 6 tests should result in execution time less than or equal to TRANSFERTIME.

2. test-id2

   Type:

   Initial State: The device is turned on and left on idle mode.

   Input: Users wear the device for 10 days.

   Output: The total amount of time before the device runs out of battery is greater or equal to the parameter, BATTERYLIFE.

   How test will be performed: 2 group members of Back End Developers each wear the device for 10 days under their normal lives. At the end of each calendar day, they are asked to record their response to a simple question: "Does the device still have battery life left? If so, how much percentage?".

5.2.3 Saftey Tests

This section will test the saftey of the device concerning electircal components and sharps hazards involving hardware

1. ST-1

   Control: Manual

   Initial State: Device begins in "off" mode.

   Input: When device is turned on a voltage over expected voltage is applied.

   Output: Device detects high voltage and sends it to ground.

   Test Case Derivation: N/A

   How test will be performed: Use a power supply to over volt the system.

5.2.4 Reusability Tests

This section will test the reusability of the various systems of the device

1. RT-1

   Control: Manual Initial State: Battery is fully discharged

   Input: Connect battery to the charging cable

   Output: Battery is fully charged without issue

   Test Case Derivation: N/A

   How test will be performed: Battery component will be isolated from device and tested to charge and discharge in multiple cycles

5.2.5 Privacy Tests

This section test the privacy of the users data

1. PT-1

   Control: Manual

Initial State: Database is full of dummy data and device is powered on

Input: Deliver malicious code to the device through data cable.

Output: Data is not able to be accessed without administrative security key

Test Case Derivation: N/A

How test will be performed: Device will be given to a volunteer with technical knowhow who does not have access to the admin security key, they will attempt to access the device.

5.3 Traceability Between Test Cases and Requirements

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

	??	??	??	??
??				X
??		X		
??	X	X		X
??		X		X
??				
??			X	
??	X		X	
??		X		
??	X	X		X
??			X	
??		X	X	
??	X			
??		X		X
??		X		
??	X			X
??	X		X	X
??		X		

Table 1: Traceability Matrix Showing the Connections between NFR's etc.

6 Unit Test Description

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

6.1 Unit Testing Scope

[What modules are outside of the scope. If there are modules that are developed by someone else, then you would say here if you aren't planning on verifying them. There may also be modules that are part of your software, but have a lower priority for verification than others. If this is the case, explain your rationale for the ranking of module importance. —SS]

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 Module 1

[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, Manual, Automatic, Static etc. Most will be automatic —SS]

   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:

2. test-id2

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   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:

3. ...

6.2.2 Module 2

...

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 Module ?

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

6.3.2 Module ?

...

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?

[This is a section that would be appropriate for some projects. —SS]

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:

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:

1. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage etc. You should look to identify at least one item for each team member.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?