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

November 2, 2022

1 Revision History

Date	Version	Notes
2022-11-02	1.0	Initial Documentation

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 (Developers, 2022e) 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

Researchers at the School of Rehabilitation Sciences (SReS) at McMaster University are interested in performing Ecological Momentary Assessment (EMA) for victims of spinal disorders and back pain. EMA aims to study the thoughts, experiences, and behaviours of a participant's daily life by repeatedly collecting data in an individual's normal environment, at or close to the time they carry out that behaviour.

The type of EMA that the SReS is interested in is focused on analyzing the daily activities and symptoms of mostly-older adults with mobility and spinal issues. They wish to track that participant's walking activity as they go about their daily life, along with prompting participants with questions when relevant events occur. The answers to event-based prompts will be combined with activity monitoring data to form a better picture about the experience this participant has with their spinal and mobility issues.

The [name of device] will perform EMA analysis in a manner which even older participants can use, integrated into one package which gathers relevant data about a participant's activities and allows them to easily report what is currently going on and how they feel. They also are looking for a way to access EMA data in various ways. This includes graphical representations of the data which are meaningful to researchers, along with the raw data itself. This data could be activity data, symptoms reporting data, both types of data collated together, and so on.

This document is intended to describe the plan for verification and validation of the device. Here, verification and validation are both technical terms with very specific meanings.

Verification involves checking whether or not the specifications which were described in the planning phase of the project are implemented correctly by

the final system designed. Validation involves checking whether or not the product actually fulfills the needs of the end user of the device. In the words of Barry Boehm, verification asks, "Are we building the product right?" and validation asks, "Are we building the right product?"Pham (1999).

3.2 Objectives

The objectives aimed to be accomplished by this verification and validation are to:

- Validate if user requirements truly represent the goals of the stakeholders of [name of device].
- Validate if the input provided by the operators of [name of device] meet the established rules and constraints.
- Verify if the high-level design of the device correctly fulfills the specifications of the functional and non-functional requirements.
- Verify if the components of the device (e.g. source code, database, physical construction, user interface, etc.) fulfill the specifications of the design.
- Discover any faults, failures, or malfunctions in the [name of device].

3.3 Relevant Documentation

Please refer to the following documentation for reference and more information: [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]

- Development Plan: Developers (2022a)
- Problem Statement & Goals: Developers (2022c)
- SRS: Developers (2022e)
- Hazard Analysis: Developers (2022b)

- VnV Plan: Developers (2022g)
- Reflection: Developers (2022d)
- User Guide: Developers (2022f)

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 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.

   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:

5.2.2 Area of Testing2

...

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 (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

Back End Developers. Development plan. https://github.com/zakerl/Capstone_Project/blob/main/docs/DevelopmentPlan/DevelopmentPlan.pdf, 2022a.

Back End Developers. Hazard analysis. https://github.com/zakerl/Capstone_Project/blob/main/docs/HazardAnalysis/HazardAnalysis.pdf, 2022b.

Back End Developers. Problem statement & goals. https://github.com/zakerl/Capstone_Project/blob/main/docs/ProblemStatementAndGoals/Team1_ProblemStatement%20%26%20Goals.pdf, 2022c.

Back End Developers. Reflection. https://github.com/zakerl/Capstone_Project/blob/main/docs/Reflection/Reflection.pdf, 2022d.

Back End Developers. System requirements specification. https://github.com/zakerl/Capstone_Project/blob/main/docs/SRS/SRS.pdf, 2022e.

Back End Developers. User guide. https://github.com/zakerl/Capstone_Project/blob/main/docs/UserGuide/UserGuide.pdf, 2022f.

Back End Developers. Vnv plan. https://github.com/zakerl/Capstone_Project/blob/main/docs/VnVPlan/VnVPlan.pdf, 2022g.

Hoang Pham. page 567–567. John Wiley & Sons, Inc, 1999.

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

1. *What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.*

   Approximately a third of this project's timeline has passed at this point. So far, team members have spent the vast majority of their time generating documentation for this project. In the process of doing so, the team has garnered a good sense of the learning styles, working styles, and habits of each individual team member.

   Now that the planning phase has largely passed, now is the right moment to utilize this newfound knowledge about the team to maximum effect. Team #1 is lucky; each of its team members is deeply committed to achieving excellence in this project. As a result, they are willing to use knowledge about the strengths, weaknesses, and differences of each member in the aim of working as efficiently and effectively as possible. Synergy has become the greatest strength for Team #1, on top of the large collection of diverse skills held by Team #1's individual members.

   Naturally, identifying the advantages of collaboration in Team #1 also sheds light on potential blind spots. For this specific team, these blind spots mostly rest within the "soft skills", as team members already are quite proficient with the tools and hard skills necessary to bring this project to fruition.

   These soft skills are:

   - **Jessica:** Critical Thinking
   - **Oliver:** Engagement During Meetings
   - **Jonathan:** Time Management

- **Anish:** Positive Attitude
- **Nish:** Leadership
- **Labeeb:** Communication

Developing these skills will be essential to the success of the project. It is the responsibility of each individual team member to work on their skills, but it is also necessary for the rest of the team to support them appropriately and make considerations to help fill any holes left behind.

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?*

   - **Critical Thinking:**
     (a) *For Jessica:* Consider the "why" for each important decision made in the project.
     (b) *For the Team:* Host design review sessions regularly to look back and determine whether or not the project is headed in the right direction.
   - **Engagement During Meetings:**
     (a) *For Oliver:* Chair a third of all meetings going forward.
     (b) *For the Team:* Assign a portion of every meeting to questions and feedback, without any other tasks preempting this section.
   - **Time Management:**
     (a) *For Jonathan:* Set up automated reminders and schedules for capstone related tasks.
     (b) *For the Team:* Parcel work into sections in which the workload required is easily understood and planned for.
   - **Positive Attitude:**
     (a) *For Anish:* Identify 3 things about each team member for which they are grateful for.
     (b) *For the Team:* Actively celebrate the successes of the team, and regularly recognize the achievements of each individual team member.

- **Leadership:**
  (a) *For Nish:* Ensure input at least once into every major decision made by the team.
  (b) *For the Team:* Whenever making an important decision, ask each individual member for their opinion.
- **Communication:**
  (a) *For Labeeb:* Raise every concern that comes to mind. Regardless of size or importance.
  (b) *For the Team:* Create a section on the Trello board specifically for non-objective related concerns.