System Verification and Validation Plan for Measuring Microstructure Changes During Thermal Treatment

Team #30, ReSprint
Edwin Do
Joseph Braun
Timothy Chen
Abdul Nour Seddiki
Tyler Magarelli

March 4, 2023

Table 1: **Revision History**

Date	**Developer**	**Notes/Changes**
Oct 31, 2022	Timothy Chen	Added to 5.2, 5.3, 7.2
Oct 31, 2022	Edwin Do	Added section 4 for V&V Plan
Nov 1, 2022	Abdul Nour Seddiki	Added to 5.1, 5.3
Nov 2, 2022	Joseph Braun	Added Section 3
Nov 2, 2022	Edwin Do	Added more content to section 4
Mar 4, 2022	Edwin Do	Revised tests for non functional requirements

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]

1 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
SRS	Software Requirements Specifications
VS	Visual Studio
UWP	Universal Windows Platform
MIN_USER_ACCEPT_RATE	90% - minimum acceptance rate

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

2 General Information

2.1 Summary

The purpose of this document is to provide a detailed plan for the testing of our system. This will include:

- Verification and Validation Plan
- System Test Description
- Unit Test Description

2.2 Objectives

The objectives of testing are to ensure that all functional and non-functional requirements of the system are being met. It is important to include both unit tests as well as system tests, as issues may arise when components are connected together in the system.

2.3 Relevant Documentation

Relevant documentation includes:

- SRS
- MIS
- MG

3 Plan

In this section of planning, it will outline our approaches to cover the requirements outlined in various areas such as the SRS document, Hazard analysis document, our implementation and design. Tools that will be used for automated unit testing and linting will also be introduced.

The following topics will be covered:

- The verification and validation team along with their respective responsibilities
- Our approach towards the SRS Verification plan
- Our approach towards the Design Verification plan
- Implementation verification plan
- Any testing and verification tools we plan to use

3.1 Verification and Validation Team

Below is a table outlining the members of the Verification and Validation team along with their respective responsibilities. Note that the listed responsibilities are only used as a guideline, responsibilities can shift between team members on a as-per-needed basis.

Table 2: Team and Responsibilities

Team Member	Role Name	Responsibilities
Edwin Do	Software and SRS Tester	Ensures that all requirements are valid and verified under the scope of software capabilities in this project and the SRS
Timothy Chen	Software Tester	Ensures that all requirements are met and verified under the scope of software capabilities in this project
Tyler Magarelli	Software Tester	Ensures that all requirements are met and verified under the scope of software capabilities in this project
Joseph Braun	Hardware Tester	Ensures that all requirements are met and verified under the scope of hardware capabilities in this project
Abdul Nour Seddiki	Hardware Tester	Ensures that all requirements are met and verified under the scope of hardware capabilities in this project
Dr. Hatem Zurob	Supervisor	Ensures that all requirements are valid and meets the expected result

3.2 SRS Verification Plan

To verify our SRS, our team intends on revisiting the SRS document on a bi-weekly basis to verify that the requirements are up to date and in sync with the project goal. This will also allow us to cover any newly discover risks or hazards which will also be reflected in the Hazard Analysis document. Any new changes within the two week window will be noted and discussed at the end to see if additional changes to the SRS document would be necessary. At each bi-weekly review, the team also plans on using the SRS checklist as a guideline throughout the meeting.

In addition, the team will use ad hoc feedback from reviewers such as classmates from our 4GA6 Capstone class as well as instructor, supervisor and

teaching assistants. This will act as a supplementary addition if any element of the SRS is out of date or missing.

3.3 Design Verification Plan

Our plans to verify our design document includes using the MIS checklist and reviews from our classmates. The MIS checklist will help our team ensure that the design logic helps us meet the requirements specified in the SRS document and cover any hazards or risks outlined in the Hazard analysis. The team will conduct an internal design review, by going over all the outlined requirements, risks and hazards and verifiying that the design does not contain any logical flaws to the best of our abilities.

In addition, the team will use the feedback from reviewers such as classmates from our 4G06 Capstone class as well as instructor, supervisor and teaching assistants. This will help further assist our team to ensure that the design is free from any logical flaws and is able to help meet the outlined requirements.

3.4 Implementation Verification Plan

Throughout the development phase of this project, the team will use GitHub Issues and pull requests to implement various features. Each pull request will require at least two other team members to inspect and review the code ensuring that it meets the requirement and design discussed. A pipeline can also be implemented in GitHub to ensure that the build in the main branch is always stable. Unit tests will also be used to ensure that the implementation of the product is verified.

3.5 Automated Testing and Verification Tools

The final product will be an Universal Windows Platform (UWP) application built using Microsoft Visual Studio (VS) in C# and XAML.

Majority of the project's testing will be conducted within VS. For unit testing, the team will use the built-in features of Unit Test Applications in VS to create unit tests projects and units tests.

Code coverage tests is also covered within the suite of tools available within VS. The team will be able to create testing suites and unit tests for the VS project. The team also plans on using the results of the code coverage tests to identify which portion of the project's code not covered by our tests. Uncovered blocks of code can be colour-coded to signify to the developer that no current test covers that block of code. Other metrics such as number of lines covered, % of a block covered will be summarized per code coverage project to help indicate where more testing efforts are needed.

Using Visual Studio's IDE extensions, the team plans on using SonarLint and CSharpier as its linter and formatting tool for C# respectively.

3.6 Software Validation Plan

To verify that the software will work as intended and designed, the team will use the sample data provided by Dr. Zurob and see if the results are within a reasonable margin of error. This sample data is obtained from a collection of existing materials with known results. If the actual results are outside a reasonable range, then the team shall conclude that the results are not valid.

4 System Test Description

4.1 Tests for Functional Requirements

FR-T1. Control: Dynamic & Manual
Initial State: Entire system is set up and the application is running.
Input: Developers will set up a sample and demonstrate the main function of the application.
Output: The application shall display the voltage, the current, the temperature and measure the conductivity of sample materials in real-time. Updates to values on the measurement tools should match with updates on the application and changes to the value of conductivity.
Test Case Derivation: Since the measurement tools are using a unified communication bus with the control computer, measurements are expected to be synchronised and displayed in real-time.
How test will be performed: Test will be performed by developers. Simultaneously monitoring both the values displayed on the external

measurement tools and the values in the application and observing for latency in the display and calculation.

FR-T2. Control: Dynamic & Manual
Initial State: Entire system is set up and the application is running.
Input: Developers will set up a sample and perform thermal treatment on it.
Output: The application shall make note of critical changes in conductivity.
Test Case Derivation: While the application is constantly monitoring the state of conductivity of samples in real-time, any major change in the conductivity indicates a transition in the phase of the material.
How test will be performed: Test will be performed by developers. Performing thermal treatment and observing changes in conductivity as calculated by the application.

FR-T3. Control: Dynamic & Manual
Initial State: Entire system is set up and the application is running.
Input: Developers will modify the setting that controls the data sampling rate in the application.
Output: The sampling rate of inputs is modified. Therefore, the display rate of inputs and outputs is going to change.
Test Case Derivation: The application is expected to be able to sample the data at variable rates, either by changing the actual rate of data acquisition or by changing the interval at which the data is displayed and analysed.
How test will be performed: Test will be performed by developers, manually testing this function of the system.

FR-T4. Control: Dynamic & Manual
Initial State: Entire system is set up and the application is running.
Input: Developers will set up a sample and perform thermal treatment on it.
Output: The application shall automatically calculate the slopes of resistivity-temperature in the phase change diagram, identify changes in these slopes and attributing slope changes to phase transitions. This information is highlighted with appropriate phase transition labels.
Test Case Derivation: While the application is constantly monitoring the resistivity of samples in real-time, major changes in the resistivity

to temperature ratios correlate to phase changes of the material.
How test will be performed: Test will be performed by developers. Performing thermal treatment and observing notifications and labels on graphs or logs made by the application.

FR-T5. Control: Dynamic & Manual
Initial State: Entire system is set up and the application is running.
Input: Developers will prompt a wireless connection to control computer and navigate the application.
Output: The application is able to be controlled using a proxy/web application.
Test Case Derivation: The control computer is expected to be connected to the network so that when the system is set up the application is operable remotely.
How test will be performed: Test will be performed by developers communicating remotely with the control computer.

4.2 Tests for Nonfunctional Requirements

4.2.1 Appearance Test

NF-AT1. Type: Static & Manual
Initial State: Application is opened and ready for use.
Input/Condition: User will follow a simple set of instructions to explore application, followed by a survey
Output/Result: Survey result will indicate MIN_USER_ACCEPT_RATE of users agree or strongly agree application is uncomplicated.
How test will be performed: Test will be performed by the user. They will receive a survey with a question. If MIN_USER_ACCEPT_RATE of users indicate agree or strongly agree, then it will be considered successful. (Refer to Appendix for Sample Survey)

NF-AT2. Type: Static & Manual
Initial State: Application is opened and ready for use.
Input/Condition: Developers will navigate the application by exploring every screen and state.
Output/Result: If all sections of the applications is in English, then this test is considered a pass. Otherwise, it is considered a fail.

How test will be performed: Different sets of instructions are used to cover a subset of the application screens and states, each set of instruction is to be carried out by a developer to verify the result.

4.2.2 Usability Test

How test will be performed:
Type: Dynamic & Automated
Initial State: Application home screen will be open.
Input/Condition: Invalid and edge case parameters will be enter into the application.
Output/Result: Application will flag the parameters that are not accepted and prevent the program from running.
How test will be performed: Test will be performed by a script which will enter and check for the flag messages after parameters have been given.

NF-UT2. Type: Dynamic & Manual
Initial State: Application will be closed.
Input/Condition: Users will be ask to open the application and perform a task. User then will be asked to return the next day to perform the same task.
Output/Result: Time from the users opening the application to interacting with it should be within 5 seconds.
How test will be performed: Test will be performed by the users. The test will pass only if the time is within 5 seconds and the task the user is asked to complete is completed without any mistakes the second time.

NF-UT3. Type: Static & Manual
Initial State: Application will be open.
Input/Condition: Developers will navigate and explore the application.
Output/Result: No calculations will be found when exploring the application.
How test will be performed: Test will be performed by the developers.

NF-UT4. Type: Static & Manual
Initial State: Research device will not have the application on it.

Input/Condition: Users will install the application on the device.
Output/Result: Application capacity on the device will be less than 8 GB.
How test will be performed: Test will be performed by the users.

NF-UT5. Type: Staic & Manual
Initial State: Application will be installed on the device.
Input/Condition: Users will set up the application with the measurement equipments according to instructions provided.
Output/Result: Application will not indicate further set up required after the initial set up of the measurement devices on the first uses.
How test will be performed: Test will be performed by users. User will be ask to set up the measurement equipment. The next time the user uses the appliaction, there will be no inidication that there will be a need to perform further set up.

4.2.3 Performance Test

NF-PT1. Type: Dynamic & Automated
Initial State: Measurement devices will be connected to sample and to device with application open.
Input/Condition: Start thermal treatment of sample.
Output/Result: Should return 100 readings a second for 95 precent of the time with the lowest reading being 98 per second.
How test will be performed: Test will be performed by a script to start the treatment while measuring the readings per second.

NF-PT2. Type: Dynamic & Automated
Initial State: Application will have defualt parameters.
Input/Condition: Parameters will be changed
Output/Result: Application will relfect changes by within 1 second for 95 percent of the time with the longest time being 2 seconds.
How test will be performed: Test will be performed by a script where parameters will be changed and the change will be timed and verified.

NF-PT3. Type: Dynamic & Automated
Initial State: Measurement devies will be connected to device.
Input/Condition: Thermal treatment sample data will be read into

the device.
Output/Result: The sample reading from the thermal treatment will be accurate to 3 decminal places 99.99 percent of the time.
How test will be performed: Test will be performed by a script where the data being read in will be compared to an exising data set of the same sample with the same parameters. The data will be identical for 99.99 percent of the readings.

NF-PT4. Type: Dynamic & Automated
Initial State: Application will be open with measuremnt devices connceted.
Input/Condition: Readings for thermal treatment sample.
Output/Result: The calculations displayed on the application will be accurate to 3 decimal places for 99.99 percent of the time.
How test will be performed: Test will be performed by a sscript performing the caluations and comparing the result to the result on the application.

NF-PT5. Type: Dynamic & Automated
Initial State: Application will be closed.
Input/Condition: Application will be opened and have modification applied to the it.
Output/Result: The application will be up and running 99 percent of the time.
How test will be performed: Test will be performed by a script making modifications on the application throughout 12 hours and seeing if the application gives a healthly response for 99 percent of the time.

4.2.4 Operational Test

NF-OT1. Type: Static & Manual
Initial State: Application is opened and ready with keyboard and mouse connected to the device.
Input/Condition: User will interact with the application with mouse and keyboard.
Output/Result: Application will accurately and correctly reflect the inputs from the mouse and keyboard provided by the user.
How test will be performed: User will be asked to perform a simple

set of instructions using the mouse and keyboard such as clicking and typing.

NF-OT2. Type: Static & Manual
Initial State: A machine/computer that does not have the application installed.
Input/Condition: Users will be asked to install the application on the machine/computer.
Output/Result: Survey results should indicate that the installation was easy to follow for MIN_USER_ACCEPT_RATE of the users.
How test will be performed: Test will be performed by the user. User will be given a survey including a question where 5 indicates easy to use, straightforward, and 1 indicating confusing, and/or difficult. If MIN_USER_ACCEPT_RATE of users respond with 5, the test will be considered a pass.

4.2.5 Maintainability/Support Test

NF-MT1. Type: Static & Manual
Initial State: A machine/computer that does not have the application installed.
Input/Condition: Application will be installed on the device and connected to the required measurement devices.
Output/Result: The application is able to operate smoothly without error with the connected measuring devices.
How test will be performed: Test will be performed by a developer installing the application onto a lab computer running Windows 10.

4.2.6 Security Test

NF-ST1. Type: Static & Manual
Initial State: Application is opened and ready to use.
Input/Condition: User will attempt to enter or modify data/parameters that they currently do not have permission for.
Output/Result: Application will display a message informing the user that they do not have permission.
How test will be performed: Test will be performed by users with missing permissions, or a test account by a developer.

NF-ST2. Type: Static & Manual
Initial State: Application is opened and ready to use.
Input/Condition: User will be asked to change protected/hidden settings given the appropriate permissions.
Output/Result: The changes are accepted and saved by the application. Settings are also accurately reflected on the application's interface.
How test will be performed: Test will be performed by users who have been granted permission to access these settings, or test account by a developer.

4.2.7 Cultural Test

NF-CT1. Type: Static & Manual
Initial State: Application is opened and ready to use.
Input/Condition: Users will explore the application through a demo/test account.
Output/Result: Survey result will indicate MIN_USER_ACCEPT_RATE of users agree or strongly agree application is appropriate.
How test will be performed: Test will be performed by the user. They will receive a survey with a question. If MIN_USER_ACCEPT_RATE of users indicate agree or strongly agree that application is appropriate, then it will be considered successful. (Refer to Appendix for Sample Survey)

4.2.8 Health and Safety Test

NF-HT1. Type: Static & Manual
Initial State: Application is opened and ready to use.
Input/Condition: Users will explore the application through a demo/test account.
Output/Result: Survey result will indicate MIN_USER_ACCEPT_RATE of users agree or strongly agree application does not cause/pose any noticeable health concerns.
How test will be performed: Test will be performed by the user. They will receive a survey with a question. If MIN_USER_ACCEPT_RATE of users indicate agree or strongly agree that application is free from health and safety concerns, then it will be considered suc-

cessful. (Refer to Appendix for Sample Survey).

4.3 Traceability Between Test Cases and Requirements

Table 3: Requirements Traceability

Requirements	**Tests**
FR1	FR-T1
FR2	FR-T2
FR3	FR-T3
FR4	FR-T4
FR5	FR-T5
NFR-L1	NF-AT1
NFR-L2	NF-AT2
NFR-U1	NF-UT1
NFR-U2	NF-UT2
NFR-U3	NF-UT3
NFR-U4	NF-UT4
NFR-U5	NF-UT5
NFR-U6	NF-UT6
NFR-U7	NF-UT7
NFR-P1	NF-PT1
NFR-P2	NF-PT2
NFR-P3	NF-PT3
NFR-P4	NF-PT4
NFR-P5	NF-PT5
NFR-O1	NF-OT1
NFR-O2	NF-OT2
NFR-O3	N/A
NFR-M1	N/A
NFR-M2	NF-MT1
NFR-M3	NF-MT1
NFR-S1	NF-ST1
NFR-S2	NF-ST2
NFR-C1	NF-CT1
NFR-H1	NF-HT1
NFR-H2	NF-HT1
NFR-I1	NF-MT1

5 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]

5.1 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]

5.2 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]

5.2.1 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. ...

5.2.2 Module 2

...

5.3 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]

5.3.1 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:

5.3.2 Module ?

...

5.4 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.

6 Appendix

This is where you can place additional information.

6.1 Symbolic Parameters

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

6.2 Survey Questions

User Experience Survey

Survey to be completed after finishing corresponding non-functional testing

1. Appearance Question

   How was the feel of the application?

   Mark only one oval.

	1	2	3	4	5	
Uncomplicated	◯	◯	◯	◯	◯	Complicated

2. Operational Question

   How was the installation of the application?

   Mark only one oval.

	1	2	3	4	5	
Simple	◯	◯	◯	◯	◯	Difficult

3. Cultural Question

   Is the graphics or terms included appropriate?

   Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Inapproptaie	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Appropriate

4. Health Question

   Do the colours or graphics used in the application cause any health concerns?

   Tick all that apply.

   - ☐ Yes
   - ☐ No

5. Health Question

   Are the colours too bright?

   Mark only one oval.

   - ◯ Yes
   - ◯ No

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.

2.