Project Title: 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

October 31, 2022

Table 1: **Revision History**

Date	**Developer**	**Notes/Changes**
Oct 31, 2022	Timothy Chen	Added to 5.2, 5.3, 7.2

Contents

1 Symbols, Abbreviations and Acronyms iv

2 General Information 1

2.1 Summary 1
2.2 Objectives 1
2.3 Relevant Documentation 1

3 Plan 1

3.1 Verification and Validation Team 1
3.2 SRS Verification Plan 2
3.3 Design Verification Plan 2
3.4 Implementation Verification Plan 2
3.5 Automated Testing and Verification Tools 2
3.6 Software Validation Plan 2

4 System Test Description 3

4.1 Tests for Functional Requirements 3
4.1.1 Area of Testing1 3
4.1.2 Area of Testing2 4
4.2 Tests for Nonfunctional Requirements 4
4.2.1 Appearance Test 4
4.2.2 Usability Test 4
4.2.3 Performance Test 6
4.2.4 Operational Test 7
4.2.5 Maintainability/Support Test 8
4.2.6 Security Test 8
4.2.7 Cultural Test 9
4.2.8 Health and Safety Test 9
4.3 Traceability Between Test Cases and Requirements 10

5 Unit Test Description 11

5.1 Unit Testing Scope 11
5.2 Tests for Functional Requirements 11
5.2.1 Module 1 11
5.2.2 Module 2 12
5.3 Tests for Nonfunctional Requirements 12

5.3.1 Module ? 12
5.3.2 Module ? 13
5.4 Traceability Between Test Cases and Modules 13

6 Appendix **14**
6.1 Symbolic Parameters 14
6.2 Survey Questions 15

List of Tables

1 **Revision History** i
2 Requirements Traceability 10
[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

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

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

2 General Information

2.1 Summary

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

2.2 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]

2.3 Relevant Documentation

[Reference relevant documentation. This will definitely include your SRS and your other project documents (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)

3 Plan

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

3.1 Verification and Validation Team

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

3.2 SRS Verification Plan

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

[Remember you have an SRS checklist —SS]

3.3 Design Verification Plan

[Plans for design verification —SS]

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

[Remember you have MG and MIS checklists —SS]

3.4 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]

3.5 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]

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

3.6 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]

4 System Test Description

4.1 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. —SS]

4.1.1 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. 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:

4.1.2 Area of Testing2

...

4.2 Tests for Nonfunctional Requirements

4.2.1 Appearance Test

NF-AT1. Type: Static & Manual
Initial State: Application home screen will be open.
Input/Condition: Users will explore the application then take a survey.
Output/Result: Survey result will indicate 90 percent of users feel application is uncomplicated.
How test will be performed: Test will be performed by the user. They will receive a survey with a question. If 90 percent of users inidicate a 1 out of 5, it will be considered successful. (Appendix)

NF-AT2. Type: Static & Manual
Initial State: Application home screen will be open.
Input/Condition: Developers will navigate the application and explore all sections.
Output/Result: Developers will ensure all section of the application is in English.
How test will be performed: Test will be performed by developers.

4.2.2 Usability Test

NF-UT1. Type: Dynamic & Manual
Initial State: Application home screen will be open.
Input/Condition: Users will be asked to perform a set of tasks.
Output/Result: Time to perform each task will be within 1 minute 90 percent of the time.
How test will be performed: Test will be performed by the users.

Each task will be timed from the moment the user starts interacting with the application and end when the user complete the task.

NF-UT2. Type: Static & Manual
Initial State: Application home screen will be open.
Input/Condition: Users will be asked to make any modifications to any parameters.
Output/Result: Time to perform the modifcation will take 5 seconds 90 percent of the time.
How test will be performed: Test will be performed by the users. Each task will be timed from the moment the user starts interacting with the application and end when the parameter has been modified.

NF-UT3. 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-UT4. 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-UT5. 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-UT6. 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-UT7. 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 will be open with keyboard and mouse connected to the device.

Input/Condition: User will interact with the application with mouse and keyboard.
Output/Result: Application will reflect inputs provided by mouse and keyboard.
How test will be performed: Test will be performed by the users.

NF-OT2. Type: Static & Manual
Initial State: Application will be uninstalled from the device.
Input/Condition: Users will be asked to installed the application.
Output/Result: Feedback surveys from the installtion should indicate the installation was simple for 90 percent of of the users.
How test will be performed: Test will be performed by the user. User will be given a survey a question where 1 being simple and 5 being difficult. If 90 percent of users answer 1, it will be considered a success.

4.2.5 Maintainability/Support Test

NF-MT1. Type: Static & Manual
Initial State: Application will be uninstalled for device.
Input/Condition: Application will be installed on the device and connected to measurement devices.
Output/Result: The applications functions will be verified and ensure it is performing smoothly on the device.
How test will be performed: Test will be performed by the developer installing the application onto the lab computer running window 7.

4.2.6 Security Test

NF-ST1. Type: Static & Manual
Initial State: Application will be open on device.
Input/Condition: User will attempt to inject data or modify parameters they do not have permission for.
Output/Result: Application will have a indicator inform user does not have permission.
How test will be performed: Test will be performed by users who do not have certain permissions.

NF-ST2. Type: Static & Manual
Initial State: Application will be open on device.
Input/Condition: User will be asked to change hidden settings and calculations.
Output/Result: Changes should be accepted and saved by the application. Will be reflected on the application interface.
How test will be performed: Test will be performed by users who have permission to access these settings/parameters.

4.2.7 Cultural Test

NF-CT1. Type: Static & Manual
Initial State: Application will be open on device.
Input/Condition: Users will explore the appliaction then take a survey
Output/Result: Result from survey will return a score of 8.
How test will be performed: Test will be performed by users which will take a survey after interacting with the application, 1 being inappropriate and 10 being appropriate.

4.2.8 Health and Safety Test

NF-HT1. Type: Static & Manual
Initial State: Application will be open on device.
Input/Condition: Users will explore the appliaction then take a survey
Output/Result: Result from survey will indicate application interface does not pose any health concerns.
How test will be performed: Test will be performed by the user which will take a survey regarding the interface's colour and design.

4.3 Traceability Between Test Cases and Requirements

Table 2: Requirements Traceability

Requirements	**Tests**
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

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

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.

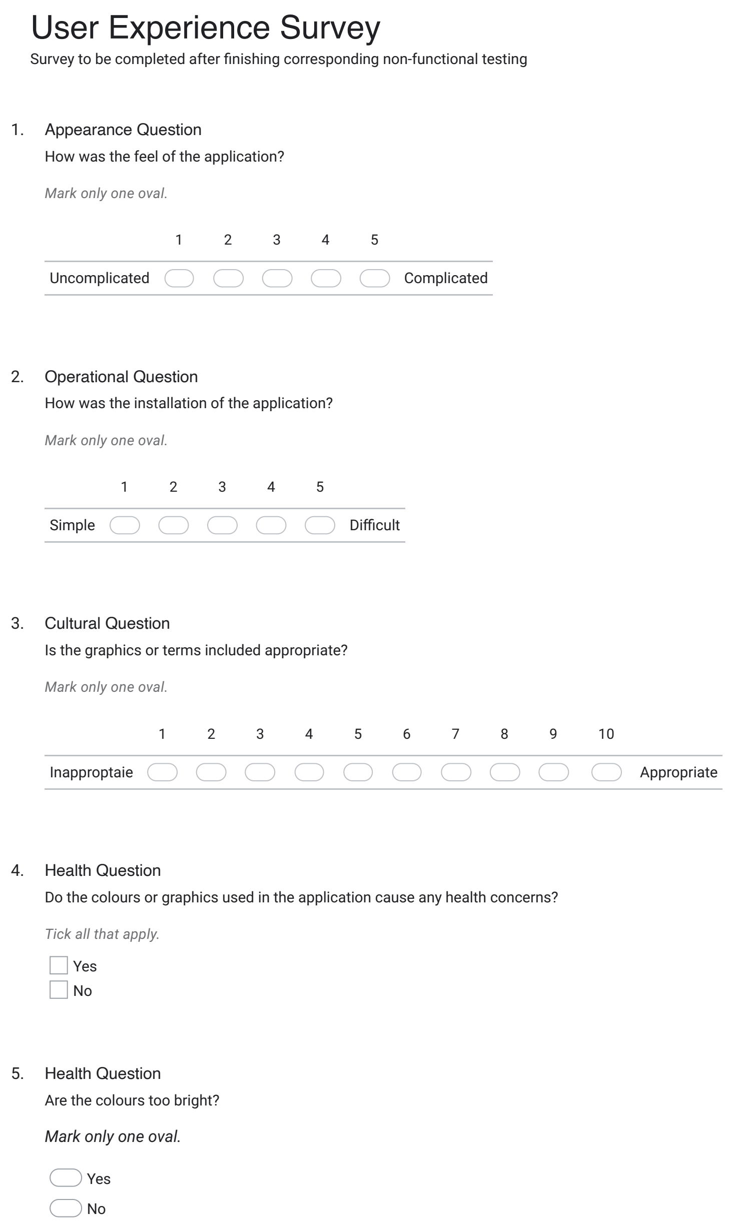
6 Appendix

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.