System Verification and Validation Plan for Solar Cooker Energy Calculator

Deesha Patel

February 14, 2023

1 Revision History

Date	Version	Notes
February 14, 2023	1.0	Add General Information section
	1.1	Add further details in different section

Contents

List of Tables

List of Figures

Remove this section if it isn't needed

2 Symbols, Abbreviations and Acronyms

symbol	description
MG	Module Guide
MIS	Module Interface Specification
SRS	Software Requirement Specification
SCEC	Solar Cooker Energy Calculator
VnV	Verification and Validation

For complete symbols used within the system, please refer the section 1 in SRS document.

This document provides the road-map of the Verification and Validation plan for Solar Cooker Energy Calculator for ensuring the requirements and goals of the program (found in SRS document). The organization of this document starts with the General Information about the Solar Cooker Energy Calculator in section 3. A verification plan is provided in section 4 and section 5 describes the system tests, including tests for functional and non-functional requirements.

3 General Information

3.1 Summary

This document reviews the validation and verification plan for Solar Cooker Energy Calculator (SCEC), a program that calculate the balance temperature at recipient and cooking power in it using user inputs.

3.2 Objectives

The purpose of the validation plan is to define how system validation will perform at the end of the project - the strategy will use to assess whether the developed system accomplishes the designed goals. Also, the verification plan includes test strategies, definitions of what will be tested, and a test matrix with detailed mapping connecting the testing performed to the system requirements. This verification plan ensures that all requirements specified in the System Requirements Specification(SRS) document have been met and reviewed. The specific goal of this document is to demonstrate the adequate usability of the system.

3.3 Relevant Documentation

The relevant documentation for the SCEC includes Problem Statement, System Requirements Specifications, VnV Report, MG and MIS (found in Github Repository)

4 Plan

This section describes the plan for the Solar Cooker Energy Calculator system. The planning starts with the Verification and Validation team, followed by the SRS verification plan, Design verification plan, Implementation verification plan, Automated testing and verification tools, and Software validation plan.

4.1 Verification and Validation Team

This section describes the members of Verification and Validation plan.

Name	Document	Role	Description
Dr. Spencer Smith	all	Instructor/ Reviewer	Review the documents, design and documentation style.
Deesha Patel	all	Author	Create all the documents, provide the VnV plan, test case and test execution, verify the implementation.
Mina Mahdipour	all	Domain Expert Reviewer	Review all the documents and review the VnV plan.
Karen Wang	SRS	Secondary Reviewer	Review the SRS document
Lesley Wheat	VnV Plan	Secondary Reviewer	Review the VnV plan.
Sam Joseph Crawford	MG + MIS	Secondary Reviewer	Review the MG and MIS document.

Table 1: Verification and Validation team

4.2 SRS Verification Plan

The SCEC SRS document shall be verified in the following way:

1. Initial review from the assigned members (Dr. Spencer Smith, Mina Mahdipour, Karen Wang, and Deesha Patel) will be performed. For this, the manual review will perform using the given SRS Checklist, designed by Dr. Smith.

2. Reviewer can give feedback to the author by creating the issue in Github.

3. Author (Deesha Patel) is responsible to address the issues created by the primary and secondary reviewers. Also, need to address the suggestions given by the instructor (Dr. Spencer Smith).

4.3 Design Verification Plan

The design documents, Module Guide (MG), and Module Interface Specification (MIS) will be verified through the static technic of document inspection by the Domain/ Primary expert (Mina Mahdipour) and Secondary Reviewer (Sam Joseph Crawford). Also, the class instructor (Dr. Spencer Smith) will review both documents. Reviewers can give feedback to the author by creating the issue in Github. The author is responsible to solve the issues and address the suggestions. The reviewer will assess this document with the help of MG Checklist and MIS Checklist designed by Dr. Smith.

4.4 Implementation Verification Plan

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

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.

4.5 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.

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

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

4.6 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.

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

This section might reference back to the SRS verification section.

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.

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

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.

Title for Test

1. test-id1

   Control: Manual versus Automatic

Initial State:

Input:

Output: The expected result for the given inputs

Test Case Derivation: Justify the expected value given in the Output field

How test will be performed:

2. test-id2

   Control: Manual versus Automatic

   Initial State:

   Input:

   Output: The expected result for the given inputs

   Test Case Derivation: Justify the expected value given in the Output field

   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

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

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

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.

6 Unit Test Description

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

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.

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.

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.

1. test-id1

   Type: Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic

   Initial State:

   Input:

   Output: The expected result for the given inputs

   Test Case Derivation: Justify the expected value given in the Output field

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic

Initial State:

Input:

Output: The expected result for the given inputs

Test Case Derivation: Justify the expected value given in the Output field

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.

These tests may involve collecting performance data from previously mentioned functional tests.

6.3.1 Module ?

1. test-id1

   Type: Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic

   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.

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.

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?