Project Title: System Verification and Validation Plan for The Nursery Project

Aaron Billones, billonea
Gillian Ford, fordg
Juan Moncada, moncadaj
Steven Ramundi, ramundis

November 2, 2022

Date	Version	Notes
2022-11-02	Juan Moncada, Aaron Billones, Steven Ramundi, Gillian Ford	Initial release

Contents

List of Tables

1 Symbols, Abbreviations and Acronyms

symbol	description
CR	Conveyor Functional Requirement
NFR	Non-Functional Requirement
PDR	Pot Dispensing Functional Requirement
TDR	Tray Dispensing Functional Requirement
VR	Verification Functional Requirement
SRS	Software Requirements Specification
TDST	Tray Dispenser Subsystem Test
PDST	Pot Dispenser Subsystem Test
CST	Conveyor Subsystem Test
VST	Verification Subsystem Test
SCT	Safety Critical Test
PT	Precision Test
RT	Reliability Test
EPET	Expected Physical Environment Test
LED	Light Emitting Diode
LCD	Liquid-Crystal Display
SRT	Speed Requirements Test
ART	Accessibility Requirements Test
LRT	Learning Requirements Test

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 (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]

?

3 Plan

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

3.1 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]

3.2 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]

3.3 Design Verification Plan

[Plans for design verification —SS]

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

[Create a checklists? —SS]

3.4 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]

3.5 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 walk-throughs, code inspection, static analyzers, etc. —SS]

3.6 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]

3.7 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]

4 System Test Description

4.1 Tests for Functional Requirements

The following section includes system test cases for functional requirements. The tests are designed in such a way to ensure that all the functional requirements are met. For reference of the functional requirements, please review the SRS document.

4.1.1 Pot-pulator Complete System Testing

[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:

4.1.2 Tray Dispenser Subsystem Testing

1. TDST-01: **Tray Stack Detection**

   Control: Static, Manual

   Initial State: No trays present in the stack. Trays present in the stack.

   Input: Sensor reads the status of tray stack.

   Output: Sends a signal/bit to microprocessor that tells the system there are/aren't trays present.

   Test Case Derivation: The observed signal/bit is the expected value. The subsystem does not operate when no trays are present.

   How test will be performed: All other sensors and subsystems will be switched off. All trays will be removed from the stack. The detection bit will be observed. Then trays will be placed in the stack, and the detection bit will be observed.

2. TDST-02: **Operation from Tray Stack Detection**

   Control: Dynamic, Manual

   Initial State: Some amount of trays in the stack.

   Input: Sensor reads the status of tray stack.

   Output: Subsystem operates or remains idle.

   Test Case Derivation: If no trays are present, the subsystem will not operate and remain ready in the idle state. Otherwise, operate normally.

   How test will be performed: All other sensors and subsystems will be switched off. Trays will be removed from the stack and operation will be observed. Trays will be put in the stack and operation will be observed.

3. TDST-03: **Tray from Stack to Conveyor**

   Control: Dynamic, Manual

   Initial State: There is a stack of trays beside the vacant conveyor with the subsystem in idle position.

   Input: Stack of trays.

   Output: One tray from the stack is placed onto the end of the conveyor and returns to idle position.

   Test Case Derivation: There is a tray in the correct designated position. The subsystem moves into the ready idle state to retrieve more trays.

   How test will be performed: All other sensors and subsystems will be switched off. The system will be manually activated to retrieve one tray from its stack. The success or failure will be observed.

4. TDST-04: **Verify Tray Status on Conveyor**

   Control: Dynamic, Manual

   Initial State: Tray put on conveyor.

   Input: Sensor reads the status of tray on conveyor.

   Output: Subsystem continues operation or stops.

Test Case Derivation: Subsystem continues operation (when successful) or stops (when tray is stuck/fails to move on conveyor).

How test will be performed: Trays will be fed onto the conveyor correctly. Results will be observed. then trays will be placed stuck on purpose. Results will be observed.

4.1.3 Pot Dispenser Subsystem Testing

1. PDST-01: **Pot from Stack to Tray**

   Control: Dynamic, Manual

   Initial State: Pot dispenser loaded with two pots

   Input: Simulated sensor input, two pot locations of tray directly below pot dispenser

   Output: Pot dispenser will dispense two pots into designated pot locations on tray

   Test Case Derivation: Pot dispenser will dispense pots into correctly positioned tray as it is prompted to

   How test will be performed: Tray will be manually placed directly below pot dispenser with pot locations directly below pot stack. Machine will be turned on. Once pots are dispensed, pot dispenser will queue next two pots and tray will be removed.

2. PDST-02: **Tray Sensing**

   Control: Dynamic, Manual

   Initial State: Mounted sensor with no object being sensed

   Input: Manual placement of trays in front of sensor

   Output: Sensor will output a signal when the presence of a tray is sensed

   Test Case Derivation: Sensor will recognize that a tray is beneath the pot dispenser

   How test will be performed: Tray will be manually placed directly in front of the mounted sensor. Signal output from sensor will be analyzed to determine sensor is aware of tray presence. Tray will then be moved forward and output from sensor will be analyzed to confirm sensor is aware that tray is moving. Tray will then be moved forward out of view of sensor and output from sensor will be analyzed to confirm snesor is aware that tray is no longer present.

3. PDST-03: **Ability to Dispense 4” Diameter Pots**

   Control: Static, Manual

   How test will be performed: All specifications of pot dispenser will ensure that a 4” diameter pot is able to be dispensed. Measurements and reviews will be conducted by another member of the group any time a change is made to the dispenser during design and build phases. During build phase, test will be conducted on both pot dispensers.

4. PDST-04: **Ability to Store/Sispense Multiple Pots**

   Control: Dynamic, Manual

   Initial State: Pot dispenser loaded with pots

   Input: Ten pots, simulated sensor input

   Output: Pot dispenser will dispense two pots, reload with two pots from stack, dispense two pots, etc. until pot storage is empty

   Test Case Derivation: Pot dispenser will complete 5 cycles of dispensing, storing and dispensing 10 pots in total

   How test will be performed: Pot dispenser will be loaded with 10 pots, 5 per side. Sensor input will be simulated to indicate presence of tray. Pot dispenser will complete 5 cycles of dispensing, at which point pot storage will be spent.

5. PDST-05: **Pot Storage Detection**

   Control: Dynamic, Manual

   Initial State: Pot dispenser with no pots in storage

   Input: N/A

   Output: Pot storage sensor will output a signal when no trays are detected in pot storage

Test Case Derivation: Sensor will recognize that no pots are sensed in pot storage

How test will be performed: All pots will be removed from pot storage. Signal output from sensor will be analyzed to confirm sensor is aware that pot storage is empty.

4.1.4 Conveyor Subsystem Testing

1. CST-01: **Conveyor Ability to Move Trays**

   Control: Dynamic, Manual

   Initial State: Conveyor with tray placed at start

   Input: Simulated inputs indicating conveyor can start

   Output: Constant speed of conveyor motor and belt

   Test Case Derivation: Conveyor will recognize tray is present on belt and able to move forward

   How test will be performed: A single tray will be placed at the start point on the conveyor belt. The conveyor will receive signals indicating that there are no issues with any other subsystems and the tray can be moved forward. Behaviour of conveyor will be observed to confirm conveyor has moved tray from start to end with no stopping. Test will be interrupted if tray is unable to move forward due to physical interferance or if conveyor stops.

2. CST-02: **Conveyor Ability to Stop**

   Control: Dynamic, Manual

   Initial State: Conveyor moving tray along belt

   Input: Simulated signals from pot dispenser indicating tray is beneath pot dispenser

   Output: Conveyor motor and belt come to a stop

   Test Case Derivation: Conveyor will receive signal from pot dispenser, indicating the tray is beneath the pot dispenser, and stop movement of tray

How test will be performed: A single tray will be placed on the conveyor while conveyor is moving. A signal will be sent to the conveyor, simulating a signal from the pot dispenser sensor which indicates that the tray is beneath the pot dispenser. Behaviour of conveyor will be observed o confirm conveyor brings tray to a stop when signal is recognized.

3. CST-03: **Conveyor Belt Friction**

   Control: Static, Manual

   Input: Mass of tray, tilt angle of conveyor belt

   Output: Maximum acceleration of conveyor belt

   Test Case Derivation: Maximum acceleration based on friction between conveyor belt and tray will be calculated and set acceleration/decceleration values will be determined

   How test will be performed: 6 trays will be weighed and the mean mass will be calculated. Each tray will be placed on the conveyor belt one by one. For each tray, the conveyor belt will be tilted until the tray begins to slip, at which point the angle at which the belt is tilted will be recorded. The mean of these 6 angles will be calculated. These values will then be used to approximately determine the maximum acceleration the trays can undergo without slipping, and the acceleration of the conveyor motor will be set to not exceed 70% of this value.

4.1.5 Verification Subsystem Testing

1. VST-01: **Verify Correct Number of Pots in Tray**

   Control: Dynamic, Manual

   Initial State: One tray filled with some pots placed on the conveyor.

   Input: Tray filled with a number of pots.

   Output: Returns a count of the number of pots in the tray.

   Test Case Derivation: The count read by the subsystem matches the actual number of pots in the tray.

How test will be performed: All other sensors and subsystems will be switched off. The subsystem will be manually activated to count the number of pots in the given tray as it moves on the conveyor. The success or failure will be observed.

2. VST-02: **Operation from Verification Status**

   Control: Dynamic, Manual

   Initial State: Tray has completed counting the number of pots in the tray and deemed it success or fail.

   Input: Status bit for success or fail of the pot verification step.

   Output: Signal to tell the system to continue/stop operation based on status bit.

   Test Case Derivation: The subsystem should signal the main processsor to turn off other subsystems when there is a problem in verifying the number of pots (ie. $actual \neq target$).

   How test will be performed: All other sensors and subsystems will be switched off. The subsystem will be manually activated to count the number of pots in the given tray as it moves on the conveyor. The success or failure will send a status bit to the main processor. The status bit will be observed.

4.2 Tests for Nonfunctional Requirements

4.2.1 Safety Critical Testing

1. SCT-01: **Tray Dispenser Failure**

   Type: Dynamic, Manual

   Initial State: Tray dispenser functioning normally

   Input: Tray dispenser disconnect

   Output: System flags tray dispenser failure

   How test will be performed: Tray dispenser will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the tray dispenser is recognized.

2. SCT-02: **Pot Dispenser Failure**

   Type: Dynamic, Manual

   Initial State: Pot dispenser functioning normally

   Input: Pot dispenser disconnect

   Output: System flags pot dispenser failure

   How test will be performed: Pot dispenser will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the pot dispenser is recognized.

3. SCT-03: **Conveyor Failure**

   Type: Dynamic, Manual

   Initial State: Conveyor functioning normally

   Input: Conveyor disconnect

   Output: System flags conveyor failure

   How test will be performed: Conveyor will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the conveyor is recognized.

4. SCT-04: **Verification Failure**

   Type: Dynamic, Manual

   Initial State: Verification functioning normally

   Input: Verification disconnect

   Output: System flags verification failure

   How test will be performed: Verification will be manually disconnected from the system during operation. System response will be analyzed to confirm that a failure in the verification is recognized.

4.2.2 Precision Testing

1. PT-01: **Tray Dispenser Precision**

   Type: Static, Manual

How test will be performed: Tray storage will be filled to maximum capacity. Tray dispenser will place tray onto conveyor belt. Centre line of tray will be established and measured relative to centre line of conveyor. Test will be repeated for all trays until tray storage is empty. Average offset measurement will be calculated.

2. PT-02: **Pot Dispenser Precision**

   Type: Static, Manual

   How test will be performed: Pot storage will be filled to 50% capacity. Pot dispenser will dispense pots into trays. Centred position of each opening in trays will be established and measured relative to centre line of pots. Test will be repeated for all pots until pot storage is empty. Average offset measurement will be calculated.

4.2.3 Reliability Testing

1. RT-01: **Function Under Vibration**

   Type: Static, Manual

   How test will be performed: Pot-pulator will run continuously for 8 hours, with a tester ensuring pots and trays are available to the machine at all times so operation is never interrupted. Machine will be subject to vibrations resulting from conveyor motor, tray dispenser motors, and pot dispenser motors. Behaviour will be observed to ensure machine is able to function under long-term exposure to small amplitude vibration.

4.2.4 Expected Physical Environment Testing

1. EPET-01: **Function Under Aerial Pollution**

   Type: Static, Manual

   How test will be performed: Pot-pulator will run continuously for 8 hours in an environment with an amount of aerial pollution similar to what is expected of the environment at Sheridan Nursery. A tester will be present to ensure that pots and trays are available to the machine at all times so operation is never interrupted. Behaviour will be observed to ensure machine is able to function for a long period of time in an environment with aerial pollution.

4.2.5 Speed Requirements Testing

1. SRT-01: **Acceleration Displacement of Trays**

   Type: Dynamic, Manual

   Initial State: Multiple trays (2-3) on the conveyor.

   How test will be performed: The conveyor speed and acceleration will be modified while trays are on it. The trays should not show perpendicular axis movement along the conveyor while in higher than normal speeds.

2. SRT-02: **Pot Dispensing Rate**

   Type: Dynamic, Automatic

   Initial State: Subsystem in idle state ready to dispense pots.

   Input: Stack of pots.

   Output: Pots dispensing.

   Test Case Derivation: The preceding system for filling pots with soil operates at a certain rate which should be met by the Pot-purlator.

   How test will be performed: Pots will be placed in the stack. When turned on, the pots will dispense at the desired rate (10 pots / 30 sec) and timed using a stopwatch. If the dispenser can meet these requirements then it passes.

3. SRT-02: **Tray Dispensing Rate**

   Type: Dynamic, Automatic

   Initial State: Subsystem in idle state ready to dispense trays.

   Input: Stack of Trays.

   Output: Trays dispensing.

   Test Case Derivation: The preceding system for filling pots with soil operates at a certain rate which should be met by the Pot-purlator.

   How test will be performed: Trays will be placed in the stack. When turned on, the trays will dispense at the desired rate (1 tray / 30 sec) and timed using a stopwatch. If the dispenser can meet these requirements then it passes.

4.2.6 Learning Requirements Testing

1. LRT-01: **Operational Simplicity**

   Type: Static

   How test will be performed: An individual will be trained on operating the entire system. After training is complete, the individual must demonstrate that they are able to successfully and safely handle all possible scenarios that may occur during operation.

4.2.7 Accessibility Requirements Testing

1. ART-01: **Audio and Visual Indicators**

   Type: Static, Manual

   Inputs: Trigger signal.

   Outputs: A corresponding light, sound, or screen display.

   How test will be performed: LEDs, speakers, and LCD screens will be placed in the system and sent a signal to activate in a specified way.

4.3 Traceability Between Test Cases and Requirements

The following table outlines all of the system tests and how they relate to the relevent requirements. The requirements can be referenced in the SRS document.

Table 1: Corresponding Test IDs and Requirements

Test ID	**Supporting Requirements**
TDST-01	TDR3, TDR5
TDST-02	TDR4, TDR5
TDST-03	TDR2
TDST-04	TDR2
PDST-01	PDR2
PDST-02	PDR2
PDST-03	PDR3

PDST-04	PDR4
PDST-05	PDR5, PDR6
CST-01	CR2
CST-02	CR3
CST-03	CR4
VST-01	VR1
VST-02	VR2
SCT-01	NFR12
SCT-02	NFR12
SCT-03	NFR12
SCT-04	NFR12
PT-01	NFR13
PT-02	NFR14
RT-01	NFR17
EPET-01	NFR20
LRT-01	NFR6
ART-01	NFR7
SRT-01	NFR8
SRT-02	NFR9
SRT-03	NFR10

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.

   (a) Steven:
   I will acquire skills related to dynamic testing of software and hardware.

   (b) Juan:
   This project presents a unique challenge when it comes to group management and teamwork, this being one of few interdisciplinary capstones it is the only capstone project(from my knowledge) that includes groups from vastly different engineering streams. While ECE and CAS Share many common teachings, engineering physics lacks that familiarity. this, however, in my opinion, is a great representation of the future. When working in the industry there are very few instances that you will be surrounded by those with similar knowledge sets and backgrounds, most of the time the team you will work with will be of diverce experience and backgrounds. Understanding how to work as a team is detrimental to good productivity. I believe throughout this project I will further develop my project management skills, they will be put to the test as each member of the team will have their expertise and we will be fusing knowledge to create a solution. Team management and general coding.

   Throughout this project I also wish to enhance my general coding skill. Which I have always had an interest in coding and have made attempts to use it whenever possible throughout my university career. I am excited to take on a project with a defined goal and no clear solution to improve my coding ability.

(c) Aaron:
Communication, team management, and time management are crucial to the success of the project. Working towards improving these skills will greatly benefit me and my team members in the future. I also expect to further my knowledge in both hardware and software while working on this capstone project so it can propel me into the future of becoming an engineer.

(d) Gillian:
Throughout this project, I am looking forward to learning skills from the other members of the team. As an Engineering physics student, the areas I would like to improve the most about are software and writing. I would like to be prepared for the industry and potentially working in a job that requires coding because it's something I enjoy, but I don't think it's something I am experienced enough in yet. In terms of writing and communication, it isn't often where I am required to write long reports anymore, and I like the challenge provided by creating this extent of documentation. I look forward to my writing skills to improve over the course of this capstone project.

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?

   (a) Steven:
   An approach to acquiring dynamic testing skills is practicing simulating testing conditions with our microcontrollers. An alternative approach is familiarizing myself with microcontroller documentation and creating testing conditions from the knowledge I obtain. I will pursue the former approach, as I feel the trial-and-error this will enable will be a better approach to learning how to effectively test specific portions of the project, while having the added bonus of allowing me to learn more about the software and hardware I will be working with.

(b) Juan:
Project management skills can be developed throughout this project in many different ways. The first will be in taking a heavy role in the logistical and administrative side of the team, another is to run the project management through an issue tracking system like the one provided by git hub. This might be tedious as it adds an extra step to all tasks as they will have to be updated and status regularly, but the pros outways the cons as this will help with the overall management of the team. Of the two solutions, both will be attempted however more emphasis will be put on the task tracking software as it adds the added benefit of simulating how large corporations are running their business.
To improve general coding two approaches can be taken on. the first being that I can take on a software dominant portion and be responsible for its fruition, or I can volunteer to take one of these software challenges on with another member in my group who is better versed in coding so that we may work hand in hand. This way if there will be support and guidance throughout and could offer a solution faster. The second option, in this case, seems more reasonable as there are hard deadlines for this project and the system itself is already ambitious.

(c) Aaron:
Working towards improving my team communication is something that I strive to do while participating in this capstone. This can be achived in several ways. One approach that I will take is to strengthen my bond with the team members. In doing so, I will feel more comfortable in sharing ideas and being constructive when necessary. Another approach is to document what each team member is working on and some of the problems they have run into. This will increase my team awareness and ultimately lead to the success of the project. I chose to pursue this skill because I know sometimes I want to do a lot of work on my own without developing trust towards my team members. This skill will help me when I become an engineer and work with large teams where communication and collaboration is crucial.

(d) Gillian:
I know I can improve my coding skills effectively by working on something closely to another member of the team who is more skilled in that area. I think this will be very helpful because I can work on a portion of software on my own, and if I have questions or get stuck, I will have someone working on something similar who will understand the problem and can help me troubleshoot. I can improve writing and communication skills by reviewing and editing not only my own work, but the writing of my peers as well. After each deliverable is finished, I think it is very useful when we go over all the writing as a team and come up with ideas on how each section of the documentation can possibly be improved.