System Verification and Validation Plan for LiDart

Team 10
Jonathan Casella
Kareem Elmokattaf
Michaela Schnull
Neeraj Ahluwalia

November 1, 2022

1 Revision History

Date	Version	Authors	Notes
02/Nov/2022	1.0	Michaela Schnull Jonathan Casella Kareem Elmokattaf Neeraj Ahluwalia	Initial Release

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
CAD	Computer Aided Design
DRC	Design Rule Check
ERC	Electrical Rule Check
GUI	Graphical User Interface
HA	Hazard Analysis
LiDAR	Light Imaging, Detection, and Ranging
MG	Module Guide
MIS	Module Interface Specification
SLAM	Simultaneous Localization and Mapping
SRS	System Requirements Specification
V&V	Verification and Validation
T	Test
TA	Teaching Assistant

[symbols, abbreviations or acronyms — you can simply reference the SRS [1] tables, if appropriate —SS]

[Remove this section if it isn't needed —SS]

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

This document presents a verification and validation plan for the LiDart system. It will be used to establish verification and testing procedures and create a plan to determine if LiDart meets it's goals as defined in the Problem Statement and Goals document.

The remainder of this document is structured as follows:

- **Section 3** provides background information about the LiDartsystem which will be subject to verification and validation activities. It also outlines the objectives of this plan and lists relevant documents.
- **Section 4** defines the verification and validation team roles and responsibilities. It presents verification methodologies and tools that will be used.
- **Sections 5** and **6** define what will be tested, and provide specific test cases. Furthermore, traceability matrices are provided to link test cases to requirements.

3 General Information

3.1 Summary

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

LiDart is a low cost, simple to use, 3D scanning robot. The LiDart system uses of low cost LiDAR sensors, consumer grade web cams, and inexpensive location markers. The user interfaces with the robot through GUI that allows them to remotely drive the robot and perform 3D scans.

3.2 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]

The purpose of this document is to create a plan to demonstrate that LiDart satisfies requirements as specified in the SRS and meets the goals of the project. This includes following objectives:

- Validate that the LiDart system meets its goals
- Create a plan to asses if the LiDart system is in conformance with both functional and non-functional requirements as specified in the SRS
- Identify the methodologies, tools, and equipment that will be used to perform V&V activities
- Create test cases to execute the V&V plan

3.3 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]

Project documentation such as the SRS and design documents are referred to throughout the V&V plan. Relevant documents are included below for reference.

- System Requirements Specification (SRS) [1]
- Problem Statement and Goals [2]
- Module Interface Specification (MIS) [3]
- Module Guide (MG) [4]
- Hazard Analysis (HA) [5]

4 Plan

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

This section introduces methods and techniques used to verify design requirements and provides a high-level plan to validate the LiDart system.

4.1 Verification and Validation Team

[Your teammates. Maybe your supervisor. You should 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]

Table 1 summarizes the roles and responsibilities of the verification and validation team.

Table 1: Team Member Roles

Team Member	**Roles and Responsibilities**
Jonathan Casella	- Software testing lead - Implementation of automated software testing methods - Design review lead
Michaela Schnull	- Verification of project documents including the SRS and V&V plan - Reporting of verification and validation activities - Printed circuit board verification
Kareem Elmokattaf	- Responsible for the design and execution of testing procedures - Maintenance of records documenting results from testing activities - Code verification
Neeraj Ahluwalia	- Preparation and maintenance of testing equipment - Hardware verification
Independent Reviewers (i.e. Teaching Assistant)	- Quality assurance and independent review

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]

The SRS will be verified to ensure that requirements are complete, unambiguous, and meet the goals of the LiDart system. The following methods shall be used to verify the SRS:

- Verify that the SRS follows the SRS checklist [6]
- Review from all team members using GitHub pull request reviews
- Independent review from the TA and classmates

4.3 Design Verification Plan

[Plans for design verification —SS]

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

[Create a checklists? —SS] The design of the LiDart system will be verified throughout development to ensure the system meets specifications and functions as intended. The following methods shall be used to verify the design:

- Hold internal design reviews before the proof of concept demo and prior to manufacturing of the system for Revision 0 and Revision 1 project phases
- Perform a failure modes and effects analysis
- Independent review from teaching assistants and classmates
- Verify that the design documentation follows the MIS [7] and MG [8] checklists

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]

The V&V plan will be verified to ensure that the plan to verify and validate the LiDart system is complete and feasible. The following methods shall be used to verify the V&V plan:

- Independent review from the TA and classmates
- Review from all team members using GitHub pull request reviews
- Verify that the V&V plan follows the V&V plan checklist [9]

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]

The implementation verification plan will be used to ensure the LiDart system meets all requirements as specified in the SRS. Verification methods that will be used include review, analysis, demonstration, and testing.

- **Review**: Review can be used when meeting a requirement is evident to a trained observer. Review of engineering drawings, code, or the physical device may be used. Techniques include code walk-throughs, code inspection, and drawing reviews.
- **Analysis**: Analysis can be used to verify design requirements where physical testing is not necessary, for example through mathematical and computer modeling. Analysis of data obtained through testing may be used to verify requirements. This verification method must be conducted by qualified individuals.
- **Demonstration**: Demonstration can be used to show that the system functions as intended. Unlike testing, demonstration does not require further analysis to determine if the system meets a requirement.
- **Testing**: Testing can be used to verify the behaviour of the system. Testing is conducted in a controlled environment with defined inputs and outputs. Test results must be analyzed to determine if tests pass or fail. Techniques that will be used include unit testing, automated testing, regression testing, and integration testing.

Section 5 specifies system test cases and Section 6 specifies unit test cases.

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]

4.6.1 Rust

Software for the LiDart project will be written in Rust, which natively includes testing facilities. The testing facilities built into Rust are explained in chapter 5 of the rustc book [10].

4.6.2 Rustfmt

The primary linter for Rust is Rustfmt. Rustfmt will be used to ensure a consistent programming style across all contributors.

4.6.3 Autodesk Inventor

LiDart's CAD suite of choice, Autodesk Inventor, will be used to ensure no mechanical conflicts exist in the design prior to manufacturing and assembly. This will be done by first modeling the LiDart robot in Autodesk Inventor then using tools within Inventor to detect any collisions or interpenetrating components.

4.6.4 EAGLE

Printed circuit board design will be done in EAGLE, which has built-in error checking tools. ERC (Electrical Rule Check) will be used to test schematics for electrical errors. DRC (Design Rule Check) will be used to ensure there are no board layout errors.

4.7 System 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]

This section lays out a series of validation exercises that will be performed before the system is deemed suitable for general users. The following exercises are designed to test the 3 major goals set out in the problem statement.

4.7.1 Accurate 3D Scanning

To validate the accuracy of the 3D scans a test environment will be constructed and scanned. Said test environment will have objects with known shapes at predetermined positions. The resulting 3D scan will be compared against the expected values for the test environment. If the point cloud matches the expected result within some tolerance this validation exercise will be deemed complete.

4.7.2 Low Cost Hardware

To validate that the goal of low cost, easily accessable hardware was met two exercises will be performed.

The first exercise will be to compare LiDart's price to other options in the market with equivalent features and comparable build quality. LiDart's price should fall below the median price of the existing solutions.

The second exercise will be to evaluate the accessibility of the hardware used by LiDart. During this exercise a list of alternative hardware components will be created for all major components of the LiDart system (e.g. motor controller, motors, cameras). This exercise will be deemed complete when all major components have at least one alternate part listed.

4.7.3 Ease of Use

To validate LiDart's ease of use a focus group will be used. The focus group will consist of subjects with no prior knowledge of the system. Each member of the focus group will be provided with the user manual then asked to scan an object. The time required to complete the scan and the percentage of successful subjects will be used to determine the success of this exercise.

Usability survey questions

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

    1. R1 Test
    Initial State: Nothing has been sent to the robot to output
    Input: Arrow key movement (example: right arrow key)
    Output: Robot moves in the direction specified (example: to the right)
    How test will be performed: The robot will be given a series of inputs from the keyboard such

as Right,Left and forward. The robot will then be observed and make sure that it follows the motions specified. This will verify that the robot is taking the input from the user as specified

2. R2 Test

Initial State: Robot is operating normally

Input: Emergency stop button is pressed

Output: Robot should completely shut down

How test will be performed: Inputs will be sent to the robot and the robot should not action any of them. So the inputs can be movement (left, right and forward) or it can be to scan the object.

3. R3 Test

Initial State: Robot is stationary

Input: From the control, there will be input specified to move the robot forward, backwards, left and right

Output: Robot should move in all specified directions

How test will be performed: Inputs will be sent to the robot and the robot should operate accordingly. The robot should move forward, backward, and rotate left and right.

4. R4 Test

Initial state: Robot is stationary

Input: No inputs are given to the robot

Output: Robot does not move or do any functions

How test will be performed: Robot will be left without any inputs given to it and its behaviour will be monitored/observed

5. R5 Test

Initial state: Robot is stationary

Input: Through a remote device connected over the internet, there will be commands given to the robot (example: Move left and Scan)

Output: Robot should move left and then start scanning

How test will be performed: Robot will be remotely connnected to a control source and the robot will then be given commands from the control source and the actions of the robot will be observed

6. R6 Test

Initial state: Robot will be idle (TO BE REVIEWED)

Input: Through the internet connection established, parameters will be sent to the robot (example: robot status)

Output: Robot will respond over wifi connection with the robot status

How test will be performed: Over a wifi connection a control will send certain paramters to the robot and the expected output should be recieved from the robot

7. R7 Test

Initial state: Robot is currently operating

Input: Robot switch will be flicked to go from "On" to "Off"

Output: Robot should power down and not have any power

How test will be performed: Robot will be powered on working. After confirming that the robot is powered on and performing actions, the robot's switch will be flicked to off and the robot should turn off

8. R8 Test

Initial state: Robot battery is low

Input: Robot batter will be connected to a battery charger
Output: Robot battery percentage should start increasing
How test will be performed: Robot will have a low battery. The battery will then be connected to a charger and tester will observe as the battery percentage increases on the GUI.
9. R9 Test (TO BE REVIEWED)
Initial state: Robot is operating without having scanned anything
Input: Robot scan
Output: Output from the robot to the remote connection will be the scans coming from the LiDAR senso
How test will be performed: Robot will be placed close to an object and instructed to scan the object. The tester will then observer the output from the robot on the remote connection.
10. R10 Test
Initial state: Robot uncalibrated
Input: Landmarks in the surrounding and the instruction to clibrate the position of the robot
Output: Coordinate of the robot in comparison to the landmarks
How test will be performed: Landmarks will be placed a specific distance away from the robot at a specific orientation. The robot will be asked to calibrate. The tester will verify the measurements from the robot and verify that they are correct
11. R11 Test
Initial state: Robot has scanned an object
Input: User identifes that they are done scanning and what to see the results
Output: Robot will send the output file to the remote connection
How test will be performed: (TO BE REVIEWED) Robot will have files preinstalled and then asked to communicate them in the same that they will communicate the results of the scan. Those files will the be verified to be correct and uncorrupted
12. R12 Test
Initial state: Robot moving with the camera installed
Input: User moving the robot around
Output: Live feed from the camera onto the remote connection
How test will be performed: Tester will place different objects infront of the robot and watch them change on the live feed on the GUI
13. R13 Test
Initial state: Robot scanning the object
Input: User continuing the scan of an object
Output: Current information on the scan. The current 3D scan model of the scan
How test will be performed: Tester will observe as the scan is happening and look at the output on the GUI from the robot. The tester will be able to confirm that the scan is coming as the robot is scanning
14. R14 Test
Initial state: Robot is ready to scan
Input: User asking for the current state of the robot
Output: Robot outputs that it is ready to scan
How test will be performed: Robot will be put in different states (Idle, ready to scan, etc..) and then asked to output the state to the GUI. The tester will be able to confirm that the output is matching what is expected

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 Usability Testing

Title for Test

1. test-id1

2. NFR10

   Verification Method: Testing
   Initial State: Robot is fully assembled
   Input: n/a
   Output: The robot must weight less than ROBOT_MAX_WEIGHT.
   How test will be performed: The robot shall be weighed on a scale. The weight of the robot and the accuracy of the measuring device must be recorded.

3. NFR11

   Verification Method: Testing
   Initial State: Robot is fully assembled
   Input: n/a
   Output: The robot dimensions must be less than ROBOT_MAX_DIM.

How test will be performed: The robot dimensions (length x width x height) shall be measured with a measuring tape. The robot dimensions and the accuracy of the measuring device must be recorded.

4. NFR12

   Verification Method: Review
   Initial State: The GUI main page is open
   Input: n/a
   Output: n/a
   How test will be performed: Each page will be navigated to from the main screen. The number of navigation levels will be recorded for each page. It must take less than MAX_NUM_LEVELS to navigate to any page on the application

5. NFR13

   Verification Method: Review
   Initial State: The GUI main page is open
   Input: n/a
   Output: n/a
   How test will be performed: A qualified individual will review the website to ensure all text is consistent, has a minimum font size of MIN_FONT_SIZE, and is sans-serif.

6. NFR9

   Verification Method: Testing
   Initial State: The robot is powered-on and connected to the GUI through WiFi
   Input: Module Guide document
   Output: At least 9/10 users with no prior knowledge of the system are able to successfully drive the robot, perform a scan, and save the scan data
   How test will be performed: A usability test shall be performed. Users will be given the Module Guide and instructed to drive the robot, perform a scan, and save the scanned data. user manual.

7. NFR14, NFR12, NFR9

   Verification Method: Usability Survey
   Initial State:
   Input:
   Output:
   How test will be performed:

8. NFR15

   Verification Method: Testing
   Initial State: The robot is powered-on and connected to the GUI through WiFi
   Input: User begins scanning and then attempts to move the robot
   Output: Error message stating the robot cannot move until scanning is complete
   How test will be performed:

9. NFR20, NFR21

   Verification Method: Testing
   Initial State: The GUI application is installed
   Input: Open the application
   Output: The application must run without crashing or performance issues
   How test will be performed: The application will run on a computer with the specifications IntelCore i5 processor and 8 GB of RAM or equivalent.

10. NFR22

    Verification Method: Review
    Initial State: The robot is fully assembled
    Input: n/a
    Output: n/a
    How test will be performed: A qualified individual shall inspect the robot to ensure there is no exposed wiring.

11. NFR25

    Verification Method: Review
    Initial State: The robot is fully assembled
    Input: n/a
    Output: n/a
    How test will be performed: A qualified individual shall inspect the robot to ensure it is in conformance with CSA 22.1:21 (Canadian Electrical Code) [11] and CSA Z434 (Industrial robots and robot systems) [12]

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

[1] N. Ahluwalia, J. Casella, K. Elmokattaf, and M. Schnull, "Software requirements specification," 2022.

[2] J. Casella, "Problem statement and goal," 2022.

[3] N. Ahluwalia, J. Casella, K. Elmokattaf, and M. Schnull, "Module interface specification," 2022.

[4] N. Ahluwalia, J. Casella, K. Elmokattaf, and M. Schnull, "Module guide," 2022.

[5] M. Schnull and K. Elmokattaf, "Hazard analysis," 2022.

[6] S. Smith, "SRS and CA checklist," 2022.

[7] S. Smith, "MIS checklist," 2022.

[8] S. Smith, "MIS checklist," 2022.

[9] S. Smith, "System verification and validation plan checklist," 2022.

[10] Rust Programming Language, *Testing*. https://doc.rust-lang.org/rustc/tests/index.html.

[11] CSA Group, "CSA C22.1:21 Canadian electrical code, part I (25th edition), safety standard for electrical installations," tech. rep., 2021.

[12] CSA Group, "CAN/CSA-Z434-14 Industrial robots and robot systems (adopted ISO 10218-1:2011, second edition, 2011-07-01, with Canadian deviations and ISO 10218-2:2011, first edition, 2011-07-01, with Canadian deviations)," tech. rep., 2019.

7 Appendix

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

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?