# System Verification and Validation Plan for LiDart

Team 10
Jonathan Casella
Kareem Elmokattaf
Michaela Schnull
Neeraj Ahluwalia

October 31, 2022

# 1 Revision History

| Date | Version | Authors | Notes |
|---|---|---|---|
| 02/Nov/2022 | 1.0 | Michaela Schnull<br>Jonathan Casella<br>Kareem Elmokattaf<br>Neeraj Ahluwalia | Initial Release |

# Contents

# List of Tables

# List of Figures

# 2 Symbols, Abbreviations and Acronyms

| **symbol** | **description** |
|---|---|
| GUI | Graphical User Interface |
| LiDAR | Light Imaging, Detection, and Ranging |
| MIS | Module Interface Specification |
| SLAM | Simultaneous Localization and Mapping |
| SRS | System Requirements Specification |
| V&V | Verification and Validation |
| T | Test |
| TA | Teaching Assistant |

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

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

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 fulfills its intended use and meets the needs of users
- 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
- Identify key V&V activities
- Create a roadmap to execute the V&V plan

## 3.3 Relevant Documentation

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 [5]

# 4 Plan

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

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 unit testing lead<br>- Implementation of automated software testing methods<br>- Design review lead |
| Michaela Schnull | - Verification of project documents<br>- Reporting of verification and validation activities |
| Kareem Elmokattaf | - Responsible for the design and execution of testing procedures<br>- Maintenance of records documenting results from testing activities |
| Neeraj Ahluwalia | - Preparation and maintenance of testing equipment<br>- Execution of test cases |
| Independent Reviewers (i.e. Teaching Assistant) | - Quality assurance and independent review |

## 4.2 SRS Verification Plan

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

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 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 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

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 analysis, review, 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.

## 4.6 Automated Testing and Verification Tools

## 4.7 Software Validation Plan

# 5 System Test Description

## 5.1 Tests for Functional Requirements

### 5.1.1 Area of Testing1

**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:

   Test Case Derivation:

   How test will be performed:

#### 5.1.2 Area of Testing2

...

## 5.2 Tests for Nonfunctional Requirements

### 5.2.1 Area of Testing1

**Title for Test**

1. test-id1

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

## 6 Unit Test Description

### 6.1 Unit Testing Scope

### 6.2 Tests for Functional Requirements

#### 6.2.1 Module 1

1. test-id1

   Type:

   Initial State:

   Input:

   Output:

   Test Case Derivation:

   How test will be performed:

2. test-id2

   Type:

   Initial State:

   Input:

   Output:

   Test Case Derivation:

   How test will be performed:

3. ...

#### 6.2.2 Module 2

...

## 6.3 Tests for Nonfunctional Requirements

### 6.3.1 Module ?

1. test-id1

   Type:

   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

# 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.

# 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?

# 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?