Software Requirements Specification
LiDart

Team 10
Jonathan Casella
Kareem Elmokattaf
Michaela Schnull
Neeraj Ahluwalia

October 5, 2022

Contents

Revision History

Date	Version	Authors	Notes
05\Oct\2022	1.0	Michaela Schnull Jonathan Casella Kareem Elmokattaf Neeraj Ahluwalia	Initial Release

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	**unit**	**SI**
m	length	metre
kg	mass	kilogram
s	time	second
W	power	watt ($\mathrm{W} = \mathrm{J\,s^{-1}}$)

1.2 Abbreviations and Acronyms

symbol	**description**
A	Assumption
CAD	Computer Aided Design
CMM	Coordinate Measuring Machine
FSM	Finite State Machine
GS	Goal Statement
GUI	Graphical User Interface
IM	Instance Model
LiDAR	Light Imaging, Detection, and Ranging
LC	Likely Change
NFR	Nonfunctional Requirement
R	Requirement
RAM	Random Access Memory
SRS	System Requirements Specification
VR	Virtual Reality

1.3 Terminology and Definitions

Term	Definition
Term	Definition

2 Introduction

Businesses and individuals are increasingly using 3D scanning technologies to collect 3D data for modeling and analysis. The 3D scanning market is rapidly growing, with a wide range of applications such as VR, rapid prototyping, reverse-engineering, and inspection technologies. Many current scanning technologies require that objects are brought to specialized scanning facilities, where fixed devices such as CMM machines and robotic arms are installed. Hand-held scanning devices are also available, however these technologies are expensive and require human operation. The cost and domain specific knowledge required are barriers to many users. There are limited solutions available for portable, remotely operated, and inexpensive 3D scanning solutions. A low cost, portable, and user-friendly scanning solution would make scanning accessible to a wider audience.

This document defines the scope of the LiDart project, specifies the scanning system, and defines requirements. The purpose of the SRS is discussed in detail in Section 2.2. Section 2.3 describes key project deliverables, as well as exclusions that are outside the scope of work. Readers of this document should be familiar with concepts outlined in Section **??**.

2.1 Problem Description and Goals

2.2 Purpose of Document

The purpose of this document is to provide specifications and requirements for the LiDart 3D scanning system. This document describes functional and non-functional requirements, undesired event handling, and start-up behavior. The functional behaviour of the system is modeled through system diagrams and instance models. The requirements defined in this document will drive design decisions and will be referenced throughout the design phase to ensure requirements are being met. The requirements will be a direct input to the verification and validation plan.

2.3 Scope of Requirements

LiDart aims to design and build a low cost 3D scanning robot, paired with a user application that displays a model of the scanned data. The robot is expected to operate indoors on flat surfaces in a controlled environment. A robot that is capable of operating outdoors or on rough terrain is not in the scope of this project. The user software application will allow users to view 3D data. However, tools for analyzing and modifying 3D models generated by the application will not be implemented. Software considerations such as licensing, user authentication, security, and data storage are also not within the scope of this project.

2.4 Organization of Document

The rest of this document provides detailed specifications and requirements for the LiDart 3D scanning system. The document is organized as follows:

Section 3: General System Description
: A general overview of the system is provided using a system context diagram. The interactions between the system, the environment,and its users are identified.

Section 4: Specific System Description
: The problem description and project goals are given, followed by system specifications including assumptions, theories, definitions, and instance models.

Section 5: Requirements
: This section defines the functional and non-functional requirements of the system.

Section 7: Likely Changes
: This section describes changes to system components that are likely to occur as a result of new features or changes in scope.

Section 8: Unlikely Changes
: This section describes the system requirements that are not likely to change.

Section 9: Development Plan
: A plan outlining the steps that will be taken to create the LiDart system is given.

Section 10: Values of Auxiliary Constants
: This section provides values for symbolic parameters used in this document.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

Figure 1 is a system context diagram of the LiDart 3D scanning system.

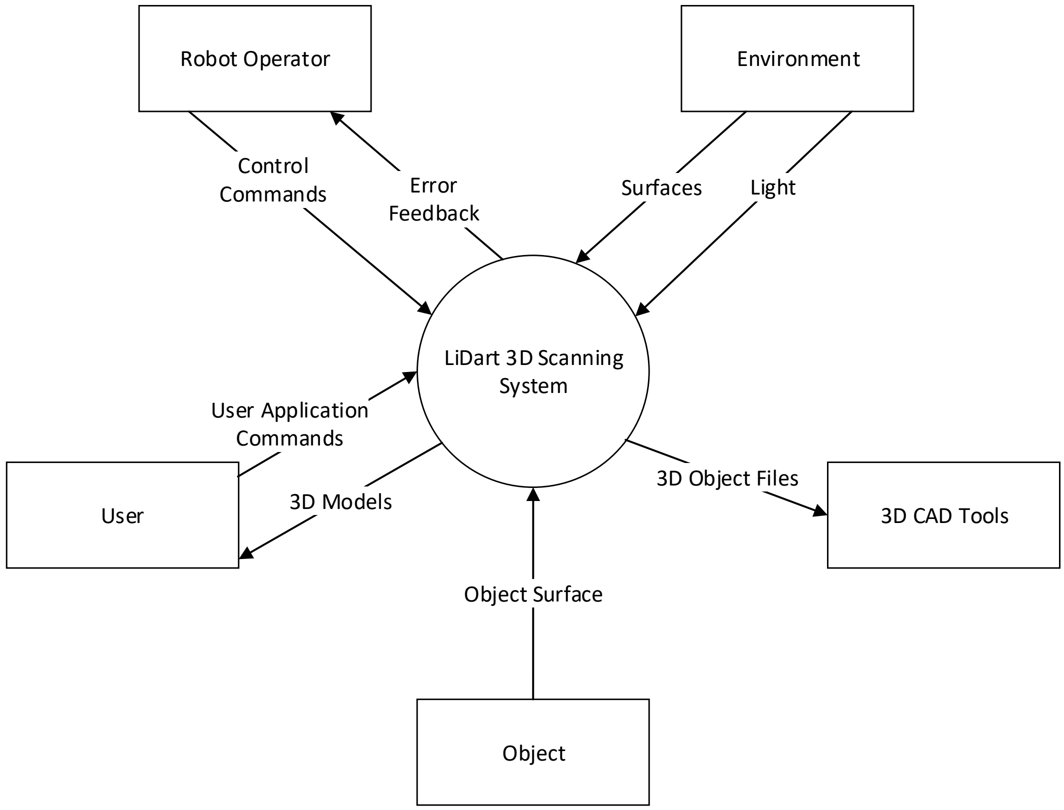


Figure 1: System Context Diagram

- Robot Operator:
  - The operator remotely controls the robot by sending commands to the robot. It is the operator's responsibility to instruct the robot to move and scan the desired object.
- Environment:
  - Information extracted from the surrounding environment is an input to the scanning system.
- User:
  - The user is responsible for providing input commands to the user application.
- Object:
  - The desired object that is being scanned is an input to the system.

- 3D CAD Tools:
  - External 3D CAD tools are able to import 3D object files that are outputted by the LiDart system. Further analysis and modification of 3D data can be carried out external CAD applications.
- LiDart 3D Scanning System:
  - The system will provide the robot operator with alerts about errors that occur during scanning operations.
  - The system is responsible for executing commands provided by the operator.
  - The system will check that the data it receives from the surrounding environment is valid.
  - The user application will respond to commands provided by the user.
  - The system outputs 3D Object Files that are supported by external CAD tools.
  - The user application will provide the user with view-able 3D models.

3.2 User Characteristics

The LiDart 3D scanning system does not require any domain specific knowledge. Individuals and business using the system are expected to have the following prior knowledge:

- An understanding of how to install applications on personal computers
- An understanding of 3D data file types and CAD programs
- Basic computer skills, such as how to navigate through application menus

3.3 Constraints

The following constraints are imposed on the LiDart system:

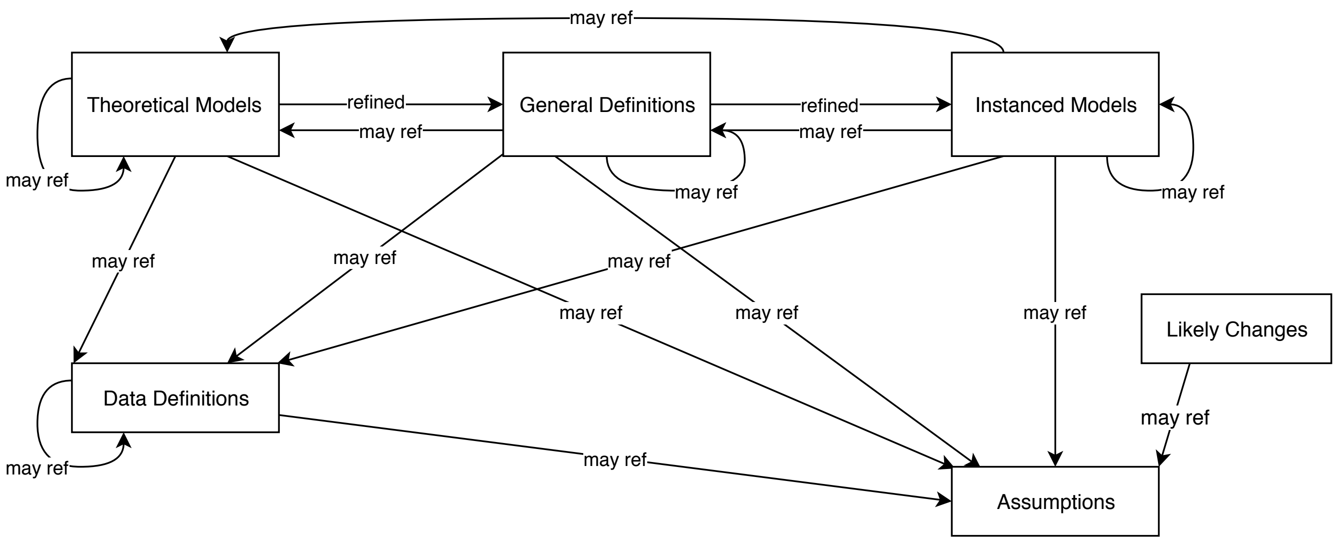
C1 The total cost of the system must be less than $750.

C2 The project must be completed by April 2022.

4 Specific System Description

This section presents assumptions, required behaviour, and a functional decomposition of the system.

4.1 Solution Characteristics Specification



The instance models that govern LiDart are presented in Subsection **??**. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.1.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1:

4.2 Behaviour Overview

4.3 Functional Decomposition

4.4 Subsystem Descriptions

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: The system shall take input from the user through a keyboard and standard pointing device such as a mouse.

R2: There shall be a kill-switch mounted on the robot, which immediately deactivates the the robot when pressed in emergency situations.

R3: The robot shall be able to move in all four directions: turn left, turn right, move forwards, and move backwards.

R4: The robot shall be stationary if no input commands are given.

R5: The robot shall be able to be connected to over a wireless network.
Rationale: A wireless connection is required for remote operation.

R6: The robot shall be able to be operated remotely.

R7: The robot shall be able to be connected to over WiFi.

R8: The robot shall be able to be operated remotely.

R9: The robot shall have a power on/off switch.

R10: The robot shall have a means of being charged.

R11: The robot shall be able to verify the inputs before processing the output. (needs to be more specific)

R12: The system shall be able to perform state estimation calculations based on landmarks in the surrounding environment.
Rationale:

R13: The system shall calculate distance and rotation between each landmark.
Rationale:

R14: The system shall transform point-cloud data into a 3D object model.

R15: The robot shall verify closed-loop calculations. (needs to be more specific)
Rationale:

R16: The GUI shall output the scanned data to a 3D model file type.
Rationale: The exported files should be compatible with existing CAD applications.

R17: The GUI shall display live video feed of the environment surrounding the robot. Rationale: This video feed can be used by the remote operator.

R18: The GUI shall display a real-time visualization of the raw scanned data.
Rationale: A real-time representation of the data can be used by the robot operator to ensure the scanner is functioning properly.

R19: The GUI shall display the current state of the system, as depicted in Figure 2.
Rationale: The states can be used by the robot operator to troubleshoot operational issues.

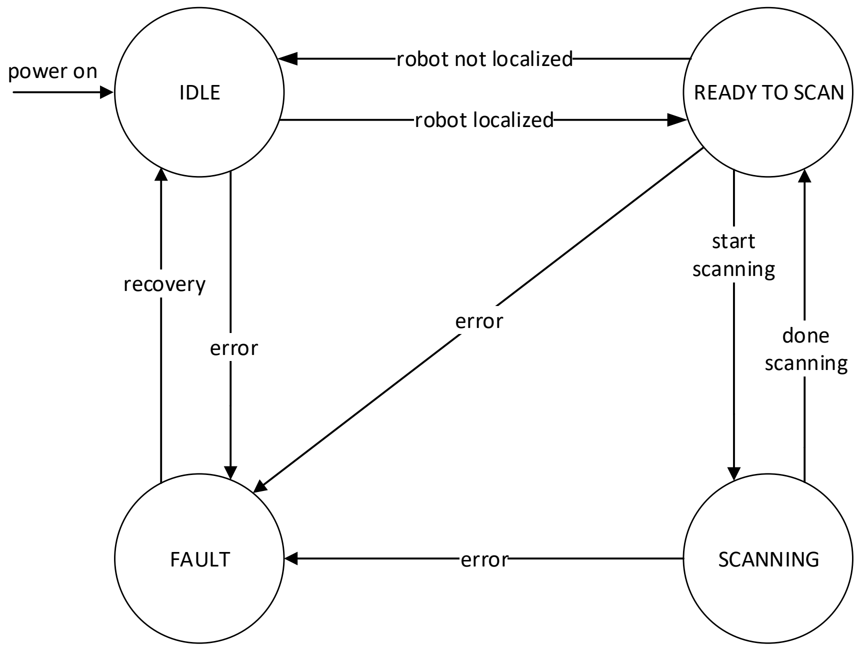


Figure 2: Robot FSM

5.2 Nonfunctional Requirements

5.2.1 Accuracy Requirements

NFR1: The accuracy of the scans shall be within SCAN_TOL. This level of accuracy meets the needs of parties interested in low-cost, easily accessible scanning.

5.2.2 Performance Requirements

NFR2: The robot shall be able to move at a maximum speed of ROBOT_SPEED.

NFR3: The robot shall be able to run for RUN_TIME of time without being charged.

NFR4: The system must be able to scan 1 square meter within SCANNING_TIME of time.

NFR5: The system shall require a maximum of MAX_PROCESSING_TIME to process raw data.

NFR6: The robot shall require a maximum of MAX_RESPONSE_TIME to respond to user input commands.

NFR7: The video feed shall have a minimum resolution of MIN_VIDEO_RESOLUTION.

NFR8: The video feed displayed on the GUI shall have a maximum delay of MAX_VIDEO_DELAY.

5.2.3 Usability Requirements

NFR9: The robot shall be easy to operate with very little knowledge. (needs to be measureable)

NFR10: The robot shall weigh less than ROBOT_MAX_WEIGHT.

NFR11: The robot shall be able to be stored within the dimensions ROBOT_MAX_DIMENSIONS.

NFR12: The GUI shall be easy to navigate. The number of levels of navigation shall not exceed MAX_NUM_LEVELS.

NFR13: The font style and size shall be consistent throughout the GUI. Fonts must be sans-serif and have a minimum font size of MIN_FONT_SIZE.

NFR14: The GUI shall be intuitive to use. Users must be able to execute desired tasks within EXECUTE_TIME seconds of accessing the user interface, for at least every 9/10 tasks.

5.2.4 Error-Handling Requirements

NFR15: The system shall display informative messages that alert the user of errors that have occurred. The messages may provide suggested steps to resolve the error.

NFR16: The system shall log error information. Error information should be specific and descriptive.

5.2.5 Maintainability Requirements

NFR17: Robot parts should be easily replaceable. It should take a maximum of REPLACE_TIME to replace any part on the robot.

NFR18: All hardware and electronic components shall be easily accessible. No special tools must be required to access hardware.
Rationale: This will facilitate activities such as debugging and reprogramming the robot.

NFR19: Standard, off-the-shelf components shall be used where possible.

5.2.6 Portability Requirements

NFR20: The user application shall run on a Windows operating system.

NFR21: The user application must be able to run on a standard personal computer. For example, the application must be able to run on a system with an IntelCore i5 processor and 8 GB of RAM.

5.2.7 Safety Requirements

NFR22: There shall not be any exposed electrical components or wiring.

NFR23: The operator shall be able to stop the robot at any time.

NFR24: The robot shall be placed in a safe-state if communication with the operator is lost.

5.2.8 Standards Requirements

NFR25: The system shall be in conformace with the following standards:

- CSA 22.1:21, Canadian Electrical Code [1]
- CSA Z434, Industrial robots and robot systems [2]

5.3 Requirements Dependencies

6 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table **??** shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table **??** shows the dependencies of instance models, requirements, and data constraints on each other. Table 1 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T??	X																		
T??																			
T??																			
GD??		X																	
GD??			X	X	X	X													
DD??							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM??											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 1: Traceability Matrix Showing the Connections Between Assumptions and Other Items

7 Likely Changes

LC1:

8 Unlikely Changes

LC2:

9 Development Plan

1.
2.
3.

10 Values of Auxiliary Constants

Constant	**Description**	**Value**
ROBOT_SPEED	Maximum speed of robot in navigation mode.	0.5m/s
SCAN_TOL	Maximum deviation between sufficient scans.	+/- 1cm
RUN_TIME	Maximum time robot can run before needing to be recharged.	3hrs
SCANNING_SPACE	Unit used for defining robot scan speed	1 square meter
SCANNING_TIME	Time elapsed to complete scan of a specified scanning space.	2mins
MAX_PROCESSING_TIME	Time needed for the robot to process raw scanned data.	10min
MAX_RESPONSE_TIME	Time needed for robot to respond to navigational commands given by user.	1s
MIN_VIDEO_RESOLUTION	Minimum resolution of video output feed in GUI.	1280x720p
MAX_VIDEO_DELAY	Max time between change in camera and output video feed in GUI.	1s
ROBOT_MAX_WEIGHT	Maximum weight of the fully assembled robot.	50kg
ROBOT_MAX_DIMENSIONS	Maximum size of the fully assembled robot.	1m x 1m x 3m (LxWxH)
MAX_NAV_LEVELS	Maximum number of layers required to access any function in GUI.	4
MIN_FONT_SIZE	The minimum font size used in the GUI for accessibility purposes.	14pt
EXECUTE_TASK_TIME	Time it would take an average user to execute 9/10 tasks within the GUI.	10s

References

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

[2] 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.

[3] W. S. Smith, "Software requirements specification template," 2022.

[4] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3d shape scanning with a time-of-flight camera," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1173–1180, 2010.

Appendix — Reflection

Throughout the LiDart project, the team will be collectively working on gaining knowledge as well as experience in order to get the project completed. Ranging from programming to electrical to mechanical knowledge each team member will be gaining a lot of knowledge in their respective domains depending on their responsibilities.

Jon and Kareem will need to understand how to get the proper communication between the robot and the machine for the LIDAR sensors. Throughout this process of figuring out the parameters, Jon will be able to expand on his knowledge on communications between the LIDAR sensors and the connected machine by working through the available documenation online. However, Kareem will be working on the controls of the robot by working through trial and error scenarios on the movement of the robot. Even though Kareem will be primarily working through trial and error, he will be using a lot of resources online to help learn how to properly figure out the routes of communication that are going to be used. Jon has picked reading through online documentation since he prefers gathering information from the original source and then applying his understandings. Kareem has picked working through trial and error as that will be the best approach to seeing how the different parameters will affect the robot and its controls. Reading through documentation would not be very effective when trying to get the right controls for the robot as it would be based on theoretical values which won't directly apply to the robot. Through theory, values can be used as a starting point.

While the software is being configured, Neeraj will be working on making the chassi for the robot. Making the chassi for LiDart will require a lot of knowledge based on weight distribution and stabilizing the different moving components. The robot will have a variety of components such as the LIDAR sensors, mounting components and motors to name a few. A lot of theory could be applied to designing the robot, but there will be a lot of trial and error with the process. That is why Neeraj is opting for using the trial and error method to figure out the best design for the robot.

Lastly, the electrical components will be a very critical domain of the robot. Getting the correct electrical equipment depending on the needs of LiDart will require a lot of work with the rest of the team. This is due to how LiDart needs to be configured. The required components such as the motor will require input from Kareem, Jon and Neeraj. The motor will depend on the weight of the chassi (Neeraj) alongside how the motor will be controlled (Kareem and Jon). The equipment picked will affect the mechanical design as the size of the equipment will be very critical to the design of LiDart. Michaela will be exploring online documentation to find the correct equipment that best matches what is required for the robot. Michaela figured that by searching online for the proper equipment she will be able to find the best matches for the requirements needed. This would also be the most effective strategy due to the limitation on the budget that has been given.