Hamdard University
Department of Computing
Final Year Project

Luma Bot AI
(FYP-002/FL24)

Software Requirements Specifications

Submitted by Rayyan Shaikh (2277-2021)
Syed Ibtesam Ahmed (2174-2021) M.Asad (1613-2021)

Supervisor(s) Dr. Khurram Iqbal

Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

Fall 2024

Document Sign off Sheet

Document Information

Project Title	Luma Bot AI
Project Code	FYP-023/FL24
Document Name	Software Requirements Specifications
Document Version	1.0
Document Identifier	FYP-002/FL24-SRS
Document Status	Final
Author(s)	Rayyan Shaikh
Approver(s)	Dr.Khurram Iqbal
Issue Date	

Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

Revision History

Date	Version	Description	Author
11-jan-2025	1.0	details of the changes made	Rayyan Shiakh

Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

Definition of Terms, Acronyms, and Abbreviations

AI (Artificial Intelligence): The simulation of human intelligence processes by machines, especially computer systems, involving learning, reasoning, and self-correction.
Computer Vision: A field of AI that enables computers to interpret and make decisions based on visual data from the world.
IoT (Internet of Things): A network of physical devices embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet.
Ardiuno : A small, affordable computer used for programming and electronic projects, including the development of smart devices.
UC (Use Case): A list of actions or event steps typically defining the interactions between a role (actor) and a system to achieve a goal.
User: In the context of this project, a visually impaired individual using the Smart Vision Kit.
Visually Impaired Individual: A person with significant visual impairment that cannot be corrected fully with glasses or contact lenses, affecting their ability to perform daily tasks.
Real-Time Processing: The immediate processing of data as it comes in, without any significant delay, to provide timely feedback or results

Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

Table of Contents

Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

1) Introduction.

Luma Bot is a smart robot that can move on its own (autonomous mode) or be controlled using a mobile app. It uses AI and sensors (LIDAR, ultrasonic, cameras) to scan rooms, detect obstacles, and navigate smoothly.

1.1) Purpose of Document.

Technical Details – Information about its hardware (Raspberry Pi, Arduino) and sensors (LiDAR, ultrasonic, camera).

Uses – Where and how Luma Bot can be helpful in real life.

1.2) Intended Audience.

This document is meant for project sponsors, development teams, AI and robotics researchers, and possible users who may be evaluating Luma AI Bot system or wish to adopt it.

Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

2) Overall System Description

2.1) Project Background

This project is an automated cleaning system integrated with AI capabilities, which can perform intelligent cleaning tasks on the surface of a robotic device. The project uses the capability of AI to detect debris and clean it up, thus providing a cleaner environment with less manual effort and time.

2.2) Problem Statement

In modern households and workplaces, there is a growing need for autonomous robots that can navigate and operate efficiently in small indoor environments.
Many existing robots are designed for large spaces and lack precise room-level scanning and obstacle detection.

2.3) Project Scope

This project is to design, develop and deploy an AI-powered robotic vacuum cleaner, which can:

Room Scanning & Obstacle Detection:
AI-based system for detecting obstacles using ensors (ultrasonic, LiDAR, cameras).
AI-Based Navigation:
Uses path-planning algorithms (*A or Dijkstra's**) for efficient
movement. Autonomous Control
Operates in autonomous mode

Not In Scope

Outdoor Navigation**:** Designed for indoor use only.

Self-Repair & Maintenance: Cannot repair or replace its own parts.

Industrial Cleaning: Limited to home or small office environments (not for large-scale or industrial use).

Multi-Room Navigation (Current Version): Does not support multi-room navigation

2.5) Project Objectives

- Build a working prototype of the Luma AI Bot.
- Real-time Dust Detection and Removable Dust Box.
- Minimize system uptime, energy usage, and user effort.

2.6) Stakeholders & Impacted Parties

- **Key Stakeholders:** Project sponsors, development teams, and AI researchers.
- **Who Is Impacted:** Facility managers, home users, and commercial users that require automated cleaning solutions.

2.7) Operating Environment

The Luma AI Bot is specifically built to work in indoor atmospheres like homes,It should be able to work well, for example, on carpets, tiles, hardwood floors,

2.8) System Constraints

- The size of the hardware must remain portable.
- Battery capacity and recharging time.
- To guarantee commercial feasibility, cost-effectiveness must be maintained.

2.9) **Assumptions & Dependencies**

- The robot will work in a controlled, obstacle-free environment.
- Pre-trained Models for Object Detection and Debris Classification.
- The reliance on stable power and quality hardware elements.

3)External Interface Requirements

3.1) Hardware Interfaces

The Luma AI Bot system supports the following hardware components:

- **Sensors:** Infrared and ultrasonic sensors for obstacle detection, dust sensors for debris identification.
- **Microcontroller/Processor:** A processing unit for examining sensor data and controlling functions.
- **Battery Unit:** The system is powered by a rechargeable lithium-ion battery.
- **Structured Wiring:** This computer network is not specifically designed for a computer only.

It should be able to reach consensus on handling tasks, working collaboratively, and facilitating efficient task performance for cleaning operations.

3.2) Software Interfaces

The following software applications are integrated with Luma AI Bot:

- **AI Model for Dust Detection:**
  - **Model Name:** One based on TensorFlow or PyTorch.
  - **Outside Owner:** Open-source community.

- **Robot Operating System (ROS):**
  - **Ros Melodic/Noetic:**
  - Open-source Community → External Owner
  - **Interfaces:** The topic and the service to communicate with the hardware.

---

4) System Functions / Functional Requirements

4.1) System Functions

Ref #	Functions	Category	Attribute	Details & Boundary Constraints
R1.1	Detect and classify debris	Evident	System accuracy	Must accurately detect dust, wrappers, and cigarette butts with 90%+ precision.
R1.2	Navigate and avoid obstacles	Hidden	Navigation speed	The system must react to obstacles within 0.5 seconds of detection.
R1.3	Perform autonomous cleaning	Evident	Battery efficiency	Cleaning duration must exceed 2 hours on a single charge.
R1.4	Operate in multiple environments	Hidden	Environmental adaptability	The bot must adjust suction and navigation for different floor types (e.g., carpet, tile)

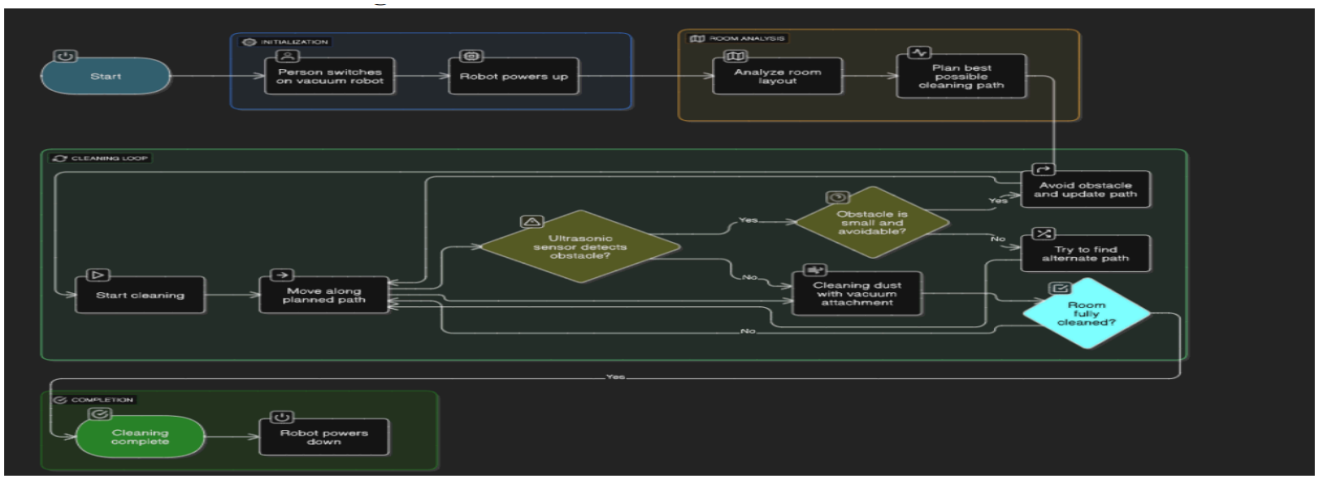
Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

System Attributes / Nonfunctional Requirements

Attribute	Details and Boundary Constraints	Category
Response time	Must detect debris and navigate obstacles within 0.5 seconds.	Mandatory
Concurrent User Load	Supports up to 5 simultaneous actions, including diagnostics and updates.	Mandatory
Energy Efficiency	Must consume less than 10% battery capacity in idle mode.	Optional
Noise Level	Cleaning operation should not exceed 65 decibels.	Mandatory

4.2) Use Cases

Use Case Diagram



Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

Description of Use Cases

Use Case: Start Cleaning

- Actors: User
- Objective: Initiate the cleaning process.
- Description: The user tells the robot to start the cleaning process. It switches on its sensors and starts the vacuum cleaner.
- See Cross References: Functions: F1 1
- Pre-condition The robot is in the power on, standby state.
- Criteria for Successful Post: The robot begins to clean.

Use Case: Move

- Actors: User
- Cleaning sequence — Move robot around the cleaning area.
- Name: Robot Moves to Areas or Traces a Path
- Cross References: Functions: F1 2
- Sensors and motors must be working.
- Post Prime: The robot is in the correct area.

Use Case: Charge Battery

- Actors: User
- Objective: To recharge the robot battery
- So, the user basically leads the robot to its charging spot.
- Cross References: Functions: F1 4
- Pre-Requisites: The robot should be alive
- Post-Conditions (successful): The robot is charging.
- Failure Condition Post-Charge: Does not start charging in case of power failure.

Use Case: Detect Dirt

- Actors: User
- Purpose: Detect dust and debris
- The robot used its AI system to identify and find dirt.
- Cross References: Functions: F1 5
- Content: They should have working AI algorithms and a camera.
- Post-conditions to the success: the dirt has been recognized and marked for cleaning.
- 4.5 Failure Post-conditions : it fails to detect dirt and logs an Error.

Use Case: Report Status

- Actors: User
- Purpose: Notify about cleaning status.
- What it does: The robot updates the user on its current cleaning progress.
- Cross References: Functions: F1. 6
- Pre-condition: Robot is operational
- The post-conditions which were completed successfully: status is returned to the user

Luma Bot AI	Version: 1.0
Software Requirements Specifications	11-Jan-2025
code	

Use Case: Stop Cleaning

- Actors: User
- End the cleaning operation.
- Action instructions: The user tells the robot to stop cleaning.
- Cross References: Functions: F1. 7
- Pre-Conditions: The robot is currently cleaning.
- Successful Post-Conditions: The cleaning process has stopped.
- Post aspects of failure: Robot does not stop cleanin

Typical Course of Event

Start Cleaning	User presses the "Start Cleaning" button.	The system activates the vacuum motor and sensors, begins cleaning, and confirms the operation has started.
Move	User instructs the robot to move to a specific location.	The system calculates the path, uses its motors to move to the specified location, and avoids obstacles during movement.
Navigate	The user allows the robot to navigate autonomously.	The system uses its AI and sensors to map the area, detect obstacles, and adjust its path to avoid collisions while continuing to clean.
Charge Battery	User directs the robot to return to the charging station.	The system stops current operations,
Stop Cleaning	User presses the "Stop Cleaning" button.	The system halts all motors, stops the cleaning operation, and enters standby mode, ensuring all operations are safely concluded.

5)Non-Functional Requirements

5.1) Performance Requirements

The bot has only 0.5 seconds to both detect and classify the debris upon encounter.

Cleaning coverage efficiency needs to be 95%+ for a standard-sized room (12 x 12 feet) in 10 minutes.

It should be capable of up to 30 minutesof running time on a single charge in standard cleaning operations.

Reporting and analytics data sync should happen in less than 5 seconds once the task is complete.

5.2) Safety Requirements

It has to cease all operations right away in order to avoid damaging itself or causing an accident.

Avoidance Handling: For avoiding obstacles Ultrasonic and infrared sensors are used to avoid collision with any obstacles (with a minimum gap of 10 cm).

5.3) Security Requirements

Our AES-256 encryption will ensure that no communications between the bot and other devices are intercepted.

5.4) **Reliability Requirements**

During cleaning operations, the system should maintain 99% uptime.

Hardware needs to last at least that with no major wear, ridicule or operational failure.

The sensors should work with 95% accuracy under normal usage conditions for a minimum of 12 months.

5.5) **Usability Requirements**

Since users may not have the technical expertise to navigate complex command sets, the bot user interface must be intuitive.

It would only take 5 minutes for first-time users to configure and set it up.

It must also be able to make announcements for the maintenance to be performed to its owner (emptying the dust box, re-charging) through sound, LED indicators, or a mobile app (future development).

5.6) **Supportability Requirements**

Over-the-air firmware updates should be enabled so that the new software can be compatible with the current system.

The modular design should allow easy replacement of critical components like sensors, batteries, and brushes.

5.7) **User Documentation**

A detailed user manual must accompany the product, including:

- A guide to setting up a dataset, step-by-step.
- Guides for troubleshooting common issues.
- How to care for optimal performance.

Digital and printed versions of the information should exist somewhere.

First-time users are to be provided with a Quick Start Guide.

It must have an online support portal with FAQs, video tutorials, and customer support contact options.

6) References

Open-Source Communities:

Frameworks of TensorFlow and PyTorch for AI model development.
Built on ROS (Robot Operating System) Melodic and Noetic for robot integration.

Hardware Manufacturers:

Sourced components from top brands like Bosch and Panasonic for sensors, microcontrollers, and lithium-ion batteries.

Guidelines and Standards set by Industry:

ISO 13482:2014: Personal care robots: safety standards. The wireless communication protocols were IEEE 802.11.

Scholar Anti-Articles and Anti-Research:

AI-based debris detection and classification papers.