Software Requirements Specification

For

SWOOSH: Smart Metal Waste Segregation System for Housekeeping

Version<2.0>

Prepared by

Group Name: JOJOMAS

Jacob, John Joseph	2021-131050	jacobjc@students.nu-fairview.edu.ph
Lim, Jana	2021-131424	limjg@students.nu-farview.edu.ph
Magtibay, Mark Jannel	2021-130543	magtibaymm@students.nu-fairview.edu.ph
Mapalo, Jose Kyle	2021-130414	mapalojc@ students.nu-fairview.edu.ph
Ramirez, Jerome Denzel	2021-130095	ramirezjf@students.nu-fairview.edu.ph

Instructor:	Mr. Herminiño C. Lagunzad
Course:	SOFTWARE DEVELOPMENT
Date:	September 18, 2024

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Document Purpose

This document outlines the SWOOSH: Smart Waste Organization System for Housekeeping software requirements. The features, user interface, technical specifications of SWOOSH, and their functional and non-functional requirements are described in this paper. It encompasses the fundamental features of the system, such as waste management, processes for optimization, and instruments for organization that improve housekeeping effectiveness. The software specifications are meant for the product's first release and will act as a basis for further updates.

The SWOOSH system, which streamlines trash sorting, collecting, and disposal for housekeeping employees in both residential and commercial settings, is described in depth in this Software Requirements Specification (SRS). SWOOSH seeks to increase operational effectiveness, save waste-related expenses, and promote sustainability through automating waste management and offering real-time optimization ideas. The essential components of the system are covered by this SRS; however, hardware specs and current integration with external systems are not.

1.2. Product Scope

A software program called SWOOSH: Smart Waste Optimization & Organization System for Housekeeping was created to enhance waste management procedures in commercial, industrial, and residential settings. Automating and streamlining waste collection, sorting, and disposal is its main goal. It also provides real-time optimization recommendations to eliminate operational inefficiencies, increase recycling rates, and decrease waste. The system analyzes waste patterns, recommends effective disposal methods, and offers data-driven insights to lessen environmental impact using algorithms.

The main advantages of SWOOSH include improved sustainability initiatives by encouraging appropriate waste segregation and lowering waste quantities, cost savings through optimized waste management, and higher housekeeping efficiency. In addition to saving time and effort when it comes to manual waste management, the system assists organizations in meeting their environmental goals by automating common processes and offering practical recommendations. Supporting greener practices, reducing the amount of waste dumped in landfills, and streamlining housekeeping procedures are the goals of SWOOSH in order to enhance resource management and cleanliness.

1.3. Intended Audience and Document Overview

The intended audience for this Software Requirements Specification (SRS) includes individuals involved in the development, assessment, and evaluation of the SWOOSH: Smart Waste Organization System for Housekeeping. The primary readers of this document are:

- **Client**: The client commissioning the development of SWOOSH will use this document to ensure that the system requirements align with their vision and goals.
- **Professor**: As the reviewer or evaluator of the project, the professor will assess the completeness, accuracy, and alignment of the software requirements with the project's objectives.
- **Developers**: The development team will use this SRS as a blueprint for building the system, including its features, functions, and technical constraints.
- **Project Managers**: Responsible for ensuring that the project stays on schedule and within scope, project managers will rely on this document to track progress and manage changes to the requirements.
- **Testers**: The testing team will refer to the hardware and software prototype to execute test cases.
- **Documentation Writers**: Writers will use the SRS to create user manuals and other supporting documentation for the system.

This SRS is divided into multiple pieces, starting with an introduction that summarizes the goals and purposes of the system. The functional requirements are then described in detail, defining how each system component is to behave. Non-functional requirements, which deal with security, usability, and performance, come next. The limitations, presumptions, and dependencies that affect SWOOSH development and implementation are also included in the document. To get the most out of their reading, readers should start with the overview sections to grasp the scope and purpose of the product. From there, they should read sections that are pertinent to their roles—for instance, developers and testers might concentrate on the functional and non-functional requirements, while clients and professors might focus on the goals and advantages of the system.

2. Overall Description

2.1. Product Overview

The Smart Waste Optimization and Organization System for Housekeeping (SWOOSH) is a waste management system that automatically sorts metal, plastic, and paper in public places. It has an automatic touchless lid for better hygiene and sensors to track how full the bins are. When a bin is nearly full, it alerts housekeeping staff via SMS, making sure it is emptied on time. The system's processor manages these alerts, reducing the need for manual checks and improving recycling.

The product has two main subsystems: an embedded hardware system and a software application. The hardware uses sensors to automatically sort waste into categories like paper, plastic, and metal, ensuring each type goes into the correct bin. The software, accessible through a web application, allows users to monitor bin levels in real-time. It also includes a notification system that alerts housekeeping staff when a bin is full, helping ensure waste is collected promptly and managed efficiently.

2.2. Product Functionality

The SWOOSH: Smart Waste Optimization & Organization System for Housekeeping is a smart system that helps the housekeeping staff to sort the metal waste from other waste. It automatically separates waste like metal, plastic, and paper into different bins. The system has a lid that opens and closes automatically to reduce the risk of germs. It also alerts housekeeping staff when the bins are nearly full, so they can empty them on time and avoid overflow.

2.2.1. Major Functionalities

Metal Waste Segregation System: The system is designed to provide functional, effective, and accurate segregation of paper, plastic, and metal waste for NU Fairview housekeeping. It uses various sensors to identify and classify different types of waste materials, ensuring proper segregation into separate bins. This allows for more efficient waste management and recycling efforts, reducing manual labor, and improving environmental sustainability.

Notification System via Web Application: The system includes a web application that allows housekeepers to track which bins are filled, providing real-time monitoring for efficient waste management.

Notification System via Text Message: The system sends text message notifications to alert housekeepers when bins are full, ensuring timely waste collection and preventing overflow.

Automatic Opening and Closing Lid Mechanism: The system features an automatic opening and closing mechanism to minimize physical contact with the waste bin, enhancing hygiene. The lid is designed to open fully within 2 seconds after detecting motion above it, ensuring quick and efficient access for users while maintaining cleanliness.

2.3. Design and Implementation Constraints

Hardware Limitations

Sensor Limitations: Metal waste detection sensors need to accurately identify if a metallic material is present. If it makes mistakes, the metal waste might be sorted wrong, which can reduce the system's effectiveness.

Network Dependence: For the metal waste segregation system to work properly, both system: software which is a web application, and the hardware (Arduino with ESP01 Wi-Fi module) needs to be connected to the same Wi-Fi connection to handle data transmission and notifications. If the internet connection is unreliable, it can slow down or disrupt the system. The hardware also needs to transfer data quickly and without delays to keep everything running smoothly.

Power Requirements: Sensors and notification systems need a steady power supply to function properly. Battery-operated components require regular recharging or replacement. To keep operational costs low and battery life longer, the hardware should be designed to be energy efficient.

2.4. Assumptions and Dependencies

2.4.1. Assumptions

- **Sensors and Actuators:** It is assumed that sensors for waste identification, along with reliable motors for sorting, are available, affordable, and easy to integrate into the system.
- **Connectivity:** Stable and reliable network connection, such as Wi-Fi or wired internet, will be available for the system. This assumption impacts whether the system can perform tasks.

- **Environment:** The system is expected to operate in indoor or semi-controlled outdoor environments with moderate temperature, humidity, and dust levels. It is not designed to handle extreme conditions such as high humidity, rain, or extreme temperatures on a regular basis.
- **Waste Volume**: The system is designed to handle a moderate amount of daily waste, making it suitable for small to medium facilities, households, or offices.

2.4.2. Dependencies

- **Internet Connectivity**: The system relies on a reliable internet connection for database storing and connection of the Arduino to the web application.
- **Sensors and Actuators:** The project relies on the availability and compatibility of specific sensors. Reliable actuators, including motors for waste sorting, are essential for smooth operation. Any problems with sourcing or integrating these parts could affect the system's performance.
- **Software Licenses:** The project depends on the licensing terms of third-party software and libraries. These licenses include restrictions or fees that could impact how the software is used, modified, or distributed within the system.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interface

This this project focused on the automatic segregation of plastic, metal, and paper using an Arduino Mega R3 2560, the hardware setup involves a range of sensors and actuators integrated with a central control system. The sensors include an optical or infrared sensor for detecting plastic, an inductive proximity sensor or metal detector for identifying metal, and an optical or capacitance sensor for sensing paper. Each sensor provides a binary output to the Arduino, indicating whether the respective material is present. The actuators responsible for sorting the waste include a servo motor or solenoid for plastic, a servo motor or pneumatic actuator for metal, paper, and plastic bin. These actuators receive control signals from the Arduino, which directs them to move the waste to the appropriate bins. The Arduino serves as the central control unit, processing inputs from the sensors and sending commands to the actuators based on the detected waste type.

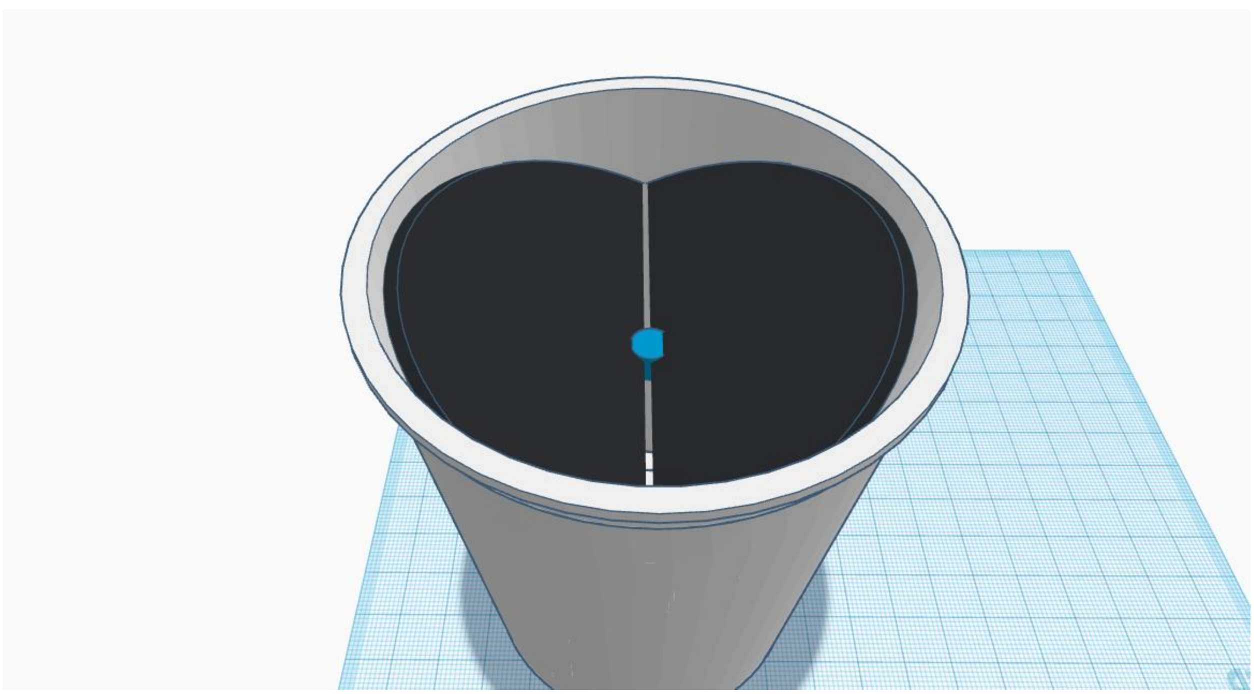


Figure 1 Top View of SWOOSH Bin

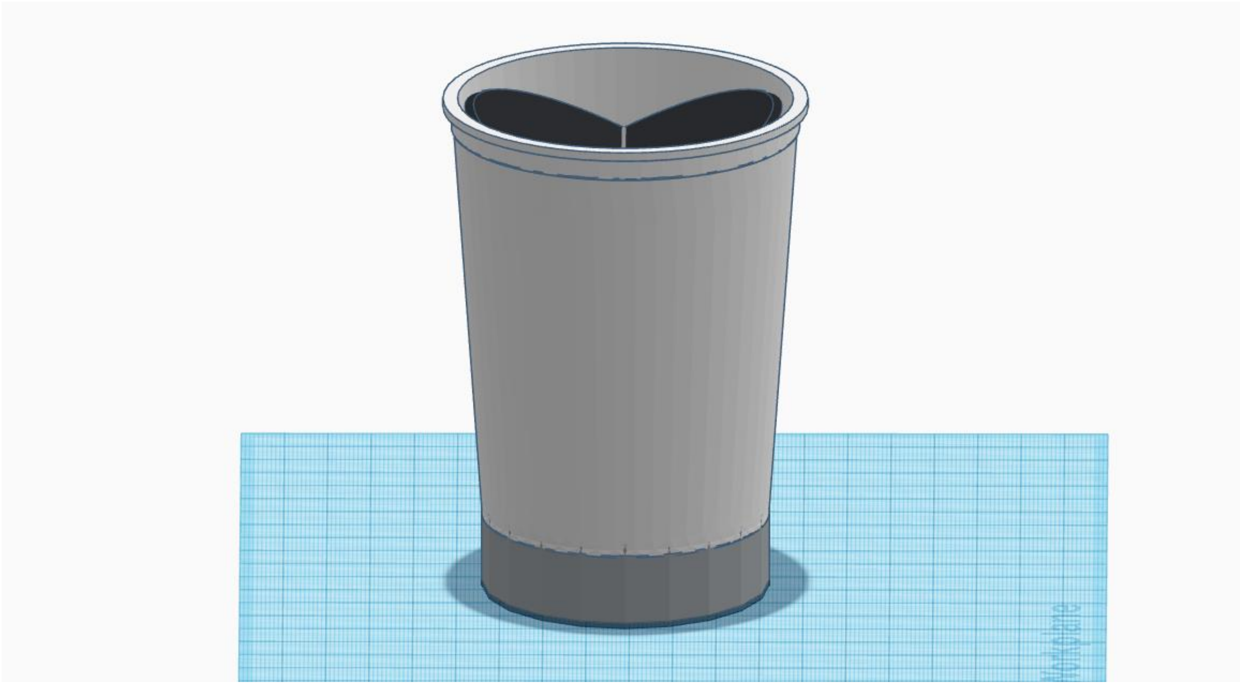


Figure 2 Side View of SWOOSH Bin

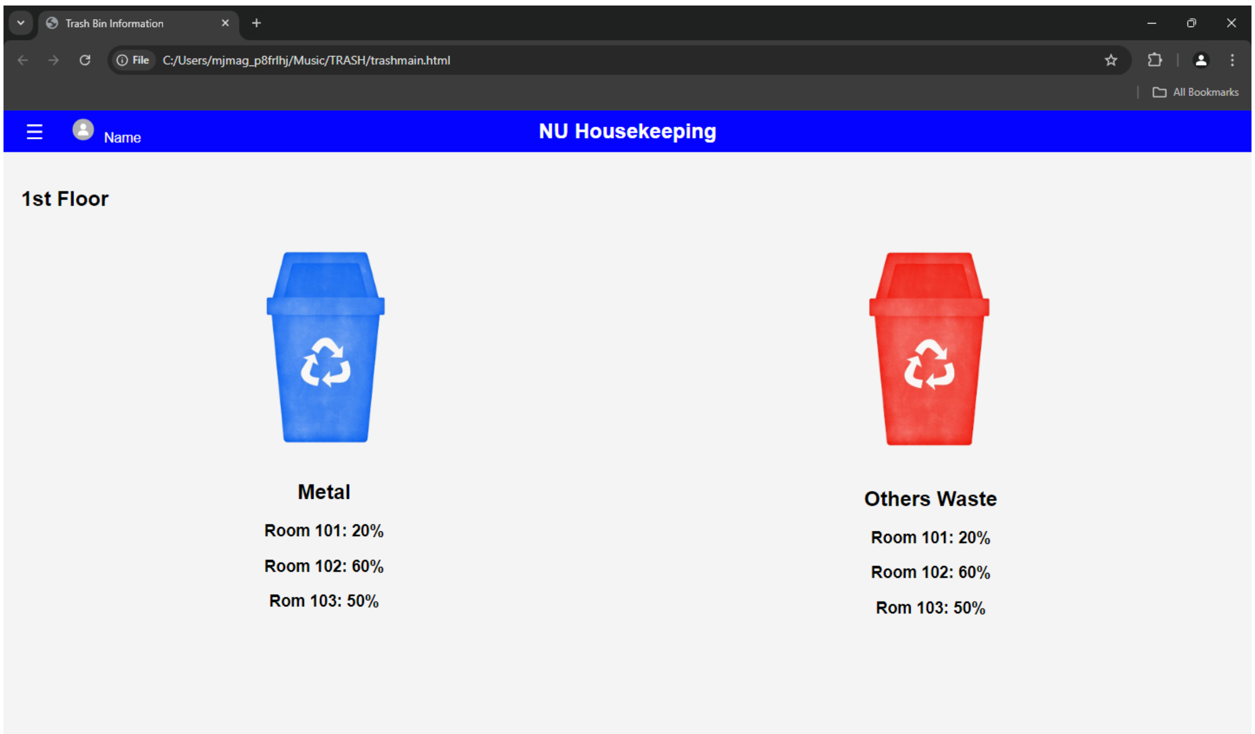


Figure 3 home page

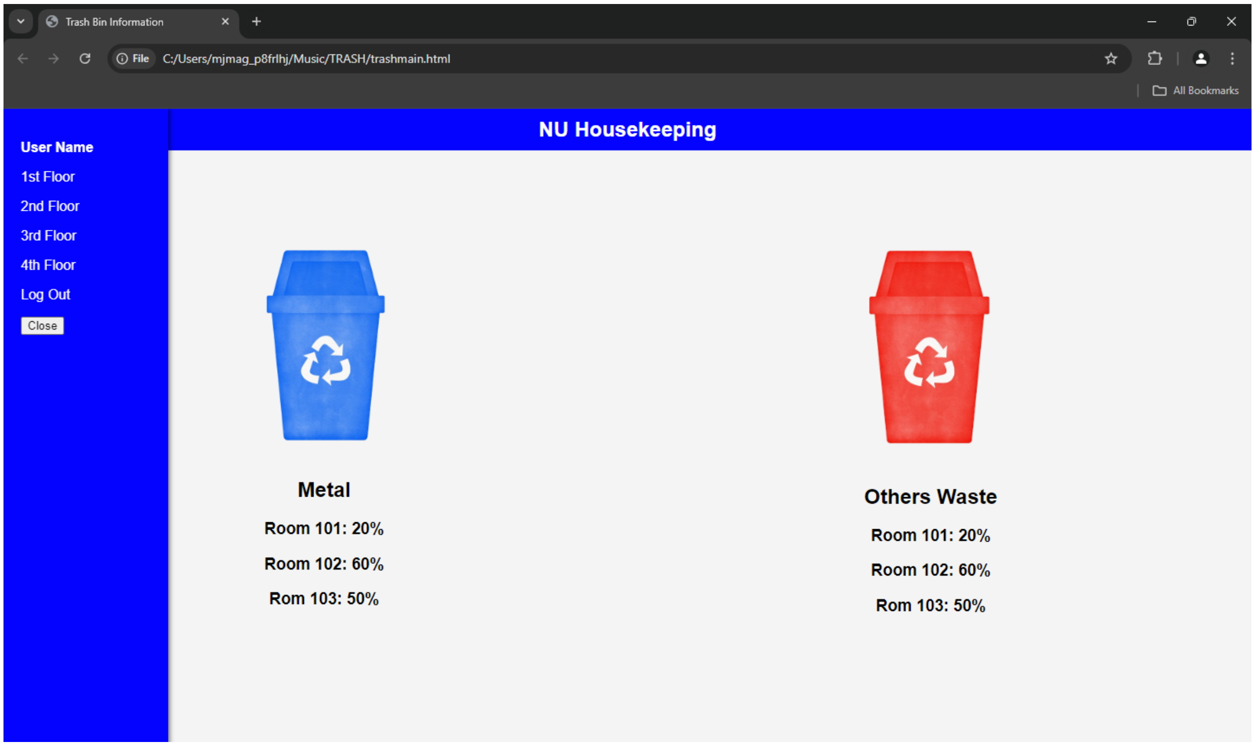


Figure 4 Menu Bar

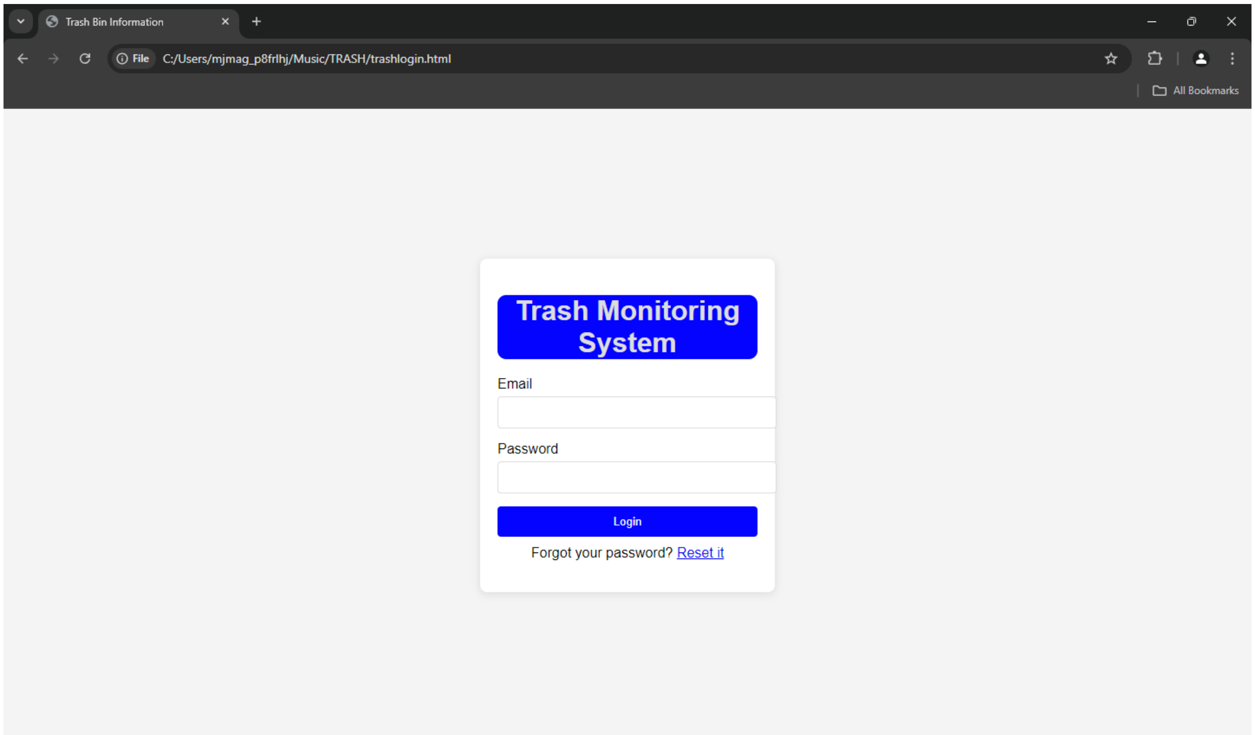


Figure 5 Login page

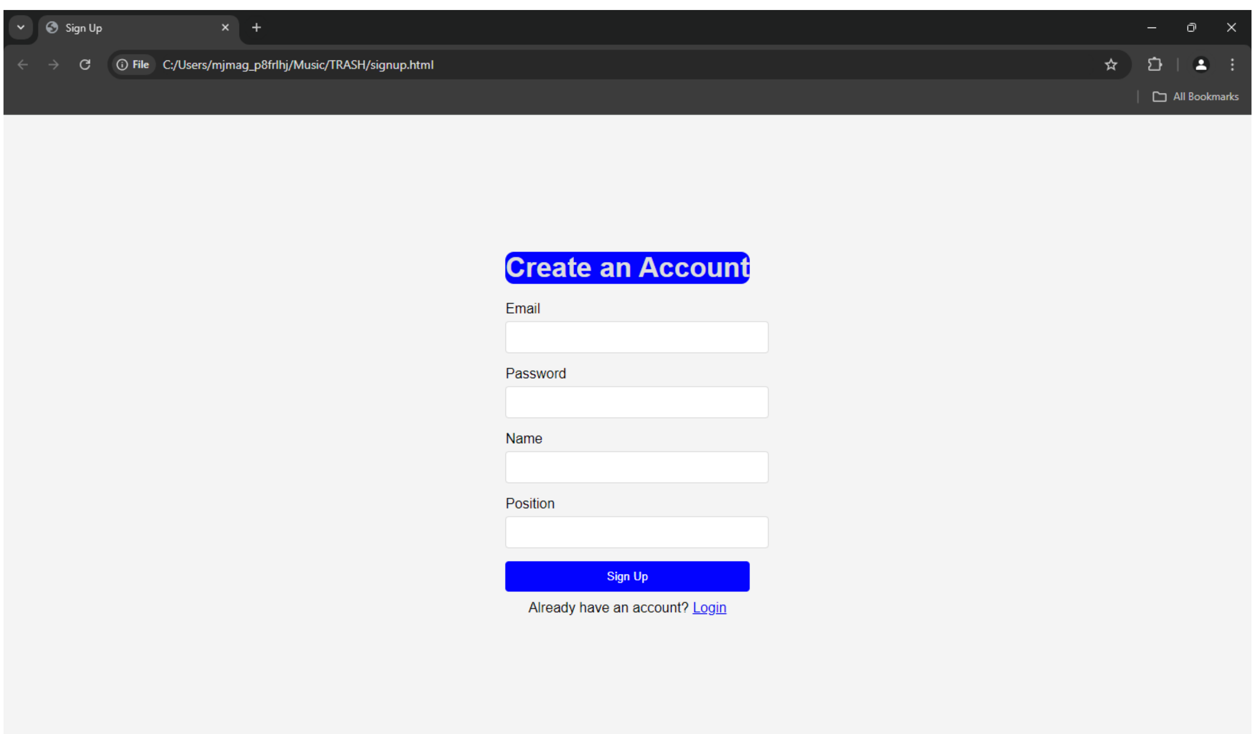


Figure 6 Sign-in page

This project involves automatic segregation of plastic, metal, and paper using an Arduino R3, here is a breakdown of the hardware interfaces will be using. This will include the devices you will interact with and their basic characteristics:

1. Sensors:

a. Infrared IR Proximity Sensor Module
- Type: Optical or Infrared sensor
- Function: To determine if waste is detected.
- Interface: Provides a binary output (waste detected or not detected).

b. Inductive Proximity Metal Detection Sensor Switch (LJ12A3-4-Z/BX NPN)
- Type: Inductive proximity sensor or metal detector
- Function: Identifies the presence of metal in the waste stream.
- Interface: Provides a binary output (metal detected or not detected).

c. Ultrasonic Sensor Distance Measuring Module (HC-SR04)
- Type: Distance sensor
- Function: Identifies the presence of the waste in the bin if it is at full capacity.
- Interface: Provides a binary output (waste detected or not detected).

2. Actuators:

a. Metal Sorting Mechanism
- Type: Servo motor
- Function: Moves the metal waste to the designated bin or area.
- Interface: Receives control signals from the Arduino to actuate sorting.

b. Other Waste Sorting Mechanism
- Type: Servo motor

- Function: Moves the other waste to the designated bin or area.
- Interface: Receives control signals from the Arduino to actuate sorting.

3. Control System:

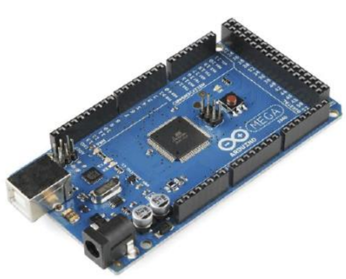
a. Arduino Mega R3 2560

- Function: Central control unit that processes input from the sensors and sends commands to the actuators.
- Interface: Connects to sensors and actuators via digital/analog pins. Uses specific code (sketch) to interpret sensor data and control actuators.

3.1.2. Hardware Interface

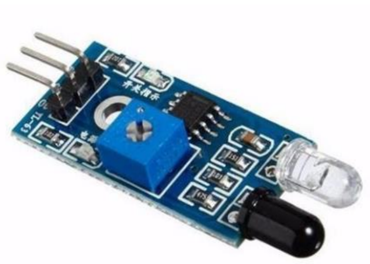
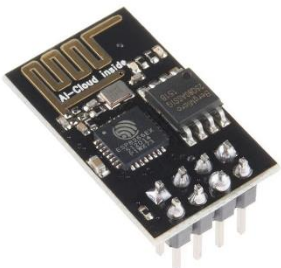
The Smart Waste Segregator, powered by an Arduino Mega R3, automatically sorts waste using three key sensors: an Inductive Proximity Sensor, an Ultrasonic Sensor, and an Infrared Sensor. This system enhances waste management by efficiently categorizing materials, reducing manual effort, and promoting eco-friendly disposal practices.

Table 1 Hardware Interface

Components	Logical Characteristics	Physical Characteristics
Arduino Mega R3 2560 	The Arduino Mega R3 features the ATmega2560 microcontroller, operating at 5V with an input voltage range of 7-12V. It has 54 digital I/O pins (15 of which support PWM) and 16 analog input pins with 10-bit resolution. The board is equipped with 256 KB of flash memory, 8 KB of SRAM, and 4 KB of EEPROM, all operating at a clock speed of 16 MHz	Physically, the Mega R3 measures 101.52 mm x 53.4 mm and weighs about 37 grams. It includes a standard USB-B port for programming and power, a barrel jack for external power, and various LED indicators. The board is designed for easy mounting with four holes, making it suitable for complex projects that

		require multiple inputs and outputs.
Ultrasonic Sensor (HC-SR04) 	The HC-SR04 ultrasonic sensor operates at 5V DC and measures distances by sending a 10 µs TTL pulse to initiate the measurement. It can detect distances ranging from 2 cm to 400 cm with an accuracy of typically ±3 mm, using a 40 kHz frequency. The sensor features a resolution of 0.3 cm and communicates via a simple digital interface that includes a trigger and echo signal.	Physically, the HC-SR04 measures approximately 45 mm x 20 mm x 15 mm and weighs about 13 grams. The sensor comes with mounting holes for easy attachment and usually has four pins: VCC, Trig, Echo, and GND. Some models also include an LED indicator for status, making the HC-SR04 widely used in robotics and distance measurement applications due to its reliability and straightforward integration.
Inductive Proximity Metal - Detection Sensor (LJ12A3-4-Z/BX NPN) 	The LJ12A3-4-Z/BX NPN inductive proximity metal detection sensor operates at 6V to 36V DC, with a sensing distance of approximately 4 mm. It features an NPN output that switches the ground connection when detecting metal, allowing easy interfacing with microcontrollers. The sensor has a response time of about 1 MS and operates in temperatures ranging from -25°C to +75°C.	Physically, the sensor has a cylindrical shape, measuring about 12 mm in diameter and 50 mm in length, with a durable metal casing. It is rated IP67, making it dust-tight and water-resistant. The sensor connects via a 3-wire configuration (VCC, ground, output), making it suitable for various industrial and automation applications.
Micro Servo (SG90)	The SG90 micro servo operates at a voltage range of 4.8V to 6V and has a torque rating of about 1.8 kg/cm at 6V. It features a 180-degree range of motion and is controlled using standard PWM signals, typically with a pulse width of 1 to 2	The SG90 micro servo measures about 22.5 mm x 11.5 mm x 31 mm and weighs around 9 grams. It has a plastic housing and includes three mounting holes for easy installation. The servo comes with a 3-pin connector for power (VCC), ground, and control

	milliseconds. The response time is approximately 0.1 seconds, making it suitable for applications that require quick movements.	signal, making it compatible with various microcontrollers and robotic systems.
24V/12V to 5V Buck Converter 	The 24V/12V to 5V buck converter efficiently steps down input voltages of 12V or 24V to a stable 5V output. It typically has an efficiency rate of around 85-95%, minimizing power loss. The converter can provide a maximum output current, often ranging from 2A to 5A, depending on the specific model. It usually features built-in protections such as overcurrent and thermal shutdown.	The buck converter is typically housed in a compact case, measuring around 50 mm x 30 mm x 15 mm. It includes multiple input and output terminals (V-in, GND, V-out) for easy connections and often has mounting holes for secure installation in various applications.
12V 3A Power Module 	The 12V 3A power module provides a regulated output of 12V with a maximum current of 3A and typically boasts an efficiency of over 85%. It often includes protection features like overvoltage, overcurrent, and thermal protection for safe operation.	Physically, the module measures around 80 mm x 40 mm x 20 mm and includes input and output terminals for easy wiring. Many models are housed in durable casings with mounting holes for secure installation in various applications.
Infrared IR Proximity Sensor	The Infrared (IR) proximity sensor detects nearby objects by emitting infrared light and measuring reflections. It typically operates at 3V to 5V, with a detection range of 10 cm to 80 cm. The sensor outputs a digital signal	Physically, the IR proximity sensor is compact, measuring around 30 mm x 20 mm x 10 mm. It features a plastic housing and includes three pins for power (VCC), ground (GND), and output. Many models also have

	when an object is detected, allowing easy interfacing with microcontrollers.	mounting holes for easy installation in various applications.
ESP8266 ESP-01 Wi-Fi Module 	The ESP8266 ESP-01 Wi-Fi module operates at 3.3V and enables wireless connectivity, supporting 802.11 b/g/n standards. It features a maximum data rate of 2 Mbps and can handle up to 15 connections simultaneously. The module includes GPIO pins for integration with microcontrollers and can be programmed using the Arduino IDE.	The ESP-01 module is compact, measuring about 24.8 mm x 14.5 mm, with two rows of pins for easy connection to breadboards. It has an onboard antenna and is housed in a plastic casing, making it ideal for various IoT applications.

3.1.3. Software Interface

The SWOOSH software features an easy-to-use dashboard and real-time data monitoring, along with tools for data logging and performance tracking to ensure smooth operation.

Table 2 Software Interface

Software	Description	Project Relevance

Arduino IDE	The Arduino IDE (Integrated Development Environment) is a user-friendly platform for programming Arduino boards. It offers an accessible interface for writing, compiling, and uploading code using a simplified version of C/C++. With built-in libraries and example sketches, users can quickly start projects, ranging from simple to complex applications.	The Arduino IDE is important to our project as it offers a user-friendly platform for programming the Arduino boards used in the waste segregation system. It simplifies coding, enabling developers to easily integrate sensors and components for automated sorting. Its built-in libraries and examples facilitate rapid prototyping, making it essential for enhancing the project's functionality.
Visual Studio Code IDE	Visual Studio Code (VS Code) is a lightweight code editor developed by Microsoft for building and debugging web and cloud applications. It supports multiple programming languages and features code completion, debugging tools, and an integrated terminal.	Visual Studio Code (VS Code) is important to our project as it provides a powerful environment for developing and debugging software components. Its support for multiple languages and extensions allows for a customized workflow, while integrated version control and debugging tools enhance collaboration and streamline development.

3.2. Functional Requirements

In the automatic segregation system for sorting metal, plastic, and paper, various interfaces collaborate for efficient operation. Optical sensors capture visual data to identify materials based on color or shape, while weight sensors measure item weight to aid in sorting accuracy. Proximity sensors detect object presence on the bin, sending binary data to track item positions and trigger sorting. Pneumatic or electric actuators then move or divert items according to sorting decisions based on sensor data. The user interface (UI) on a control panel allows manual monitoring and adjustments. Data communication ports facilitate command and data exchange between software and hardware. Each interface is crucial for precise and effective material sorting.

1. Material Detection and Classification

   a. Detection

   - Function: The system must detect the materials (metal and other waste) through sensors.
   - Details: The software should process the detection of the materials to ensure the accuracy of the segregation.

   b. Proximity Sensor

   - Function: The system must use proximity sensors to determine if the bin is full.
   - Details: Proximity sensors will provide real-time feedback on item location, ensuring that sorting mechanisms are triggered at the correct times.

2. Sorting and Separation

   a. Sorting Mechanism Control

   - Function: The system must control sorting actuators to divert items into the correct bins or channels.
   - Details: Based on the classification results from sensors, the software should send commands to pneumatic or electric actuators to move items into designated areas for metal, plastic, or paper.

   b. Actuator Coordination

   - Function: The system must coordinate the actions of multiple actuators to handle high throughput efficiently.

- Details: The software should ensure that actuators operate coordinated and handle the timing and sequence of sorting operations to prevent cross-contamination of materials.

3. System Monitoring and Control

   a. Real-Time Monitoring

- Function: The system must provide real-time status updates on the sorting process.
- Details: The user interface should display current sensor readings, sorting statistics, and system health information, allowing operators to monitor performance and detect issues.

   b. Manual Override

- Function: The system must allow operators to manually control sorting mechanisms if needed.
- Details: The user interface should provide options to override automatic sorting, adjust settings, and perform manual inspections or adjustments.

4. Data Logging and Reporting

   a. Data Logging

- Function: The system must log data related to sorting operations.
- Details: The software should record information such as the number of items sorted, material types detected, and system performance metrics. This data should be stored for historical analysis and troubleshooting.

   b. Reporting

- Function: The system must generate reports based on logged data.

- Details: The software should produce reports summarizing sorting efficiency, accuracy, and other key performance indicators. Reports should be available for review by operators and managers.

5. System Calibration and Maintenance

   a. Calibration

- Function: The system must allow for periodic calibration of sensors and actuators.
- Details: The software should provide calibration routines and guidelines to ensure that sensors and actuators maintain accuracy over time. Calibration should be easy to perform and require minimal downtime.

   b. Maintenance Alerts

- Function: The system must monitor hardware components for signs of wear or malfunction.
- Details: The software should issue alerts when maintenance is required, such as when sensors are out of calibration or actuators show signs of mechanical failure. Maintenance logs should be maintained for reference.

6. Integration and Communication

   a. Device Communication

- Function: The system must facilitate communication between the software and hardware components.
- Details: The software should handle data exchange with sensors, actuators, and other hardware devices through established communication protocols. This includes sending commands, receiving sensor data, and updating system status.

b. External Interface Integration

- Function: The system must integrate with external systems if required.
- Details: Warehouse management systems or data analytics platforms. Integration should be flexible and configurable based on external system requirements.

These functional requirements outline the essential capabilities and behaviors needed for an effective automatic metal, plastic, and paper segregation system, ensuring accurate sorting, reliable operation, and efficient management.

3.3. Use Case Model

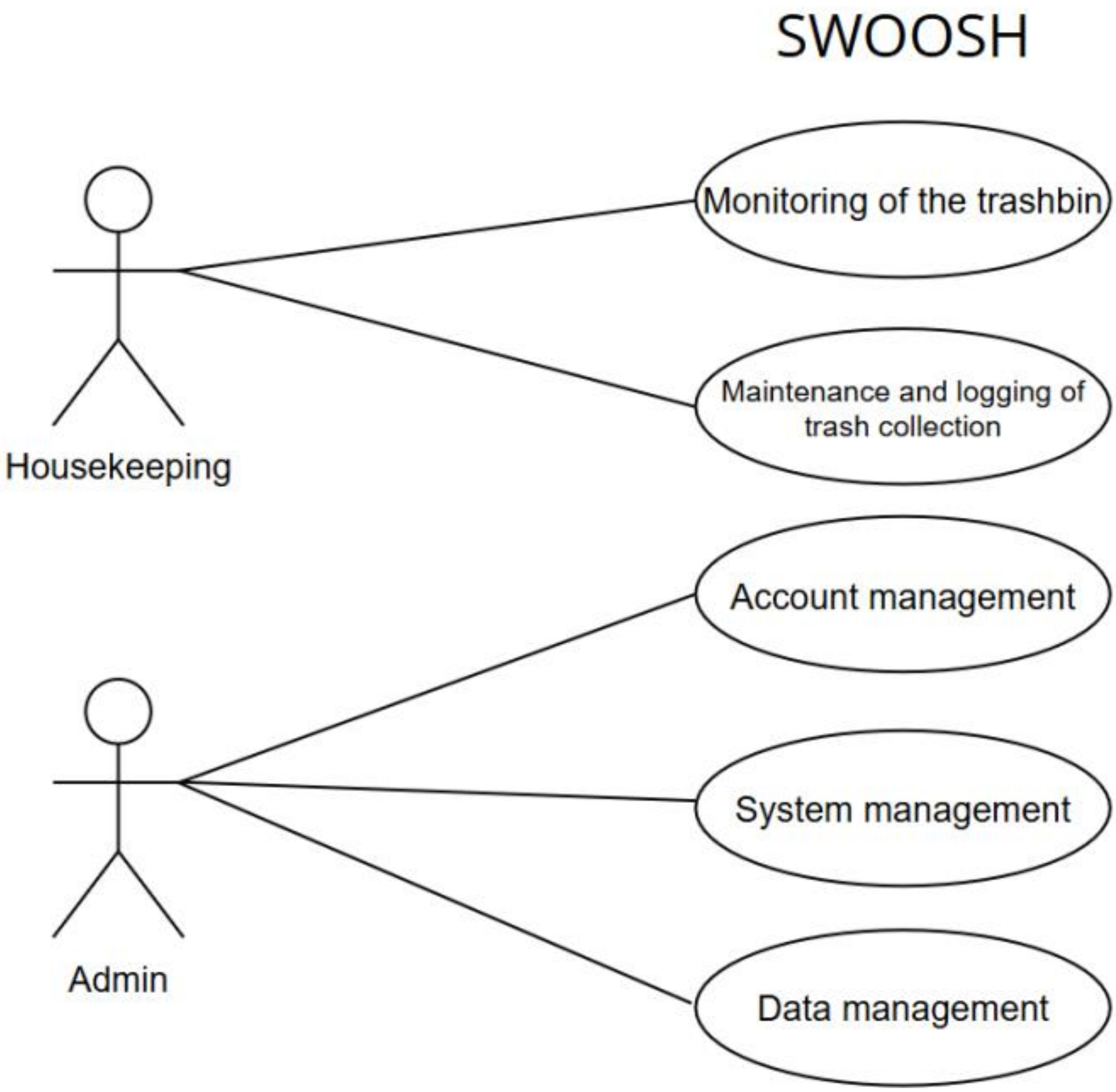


Figure 7 Use Case Model

3.3.1. Use Case #1

Table 3 Use Case #1

UC001	**User**
Author	**Jose Kyle Mapalo**
Purpose	Enable users to dispose of plastic items into a waste bin for automatic identification and sorting into the appropriate waste category.
Requirements Traceability	• The item must be near the sensors to identify if the object is plastic. • Items should be placed individually to ensure precise detection and sorting.
Preconditions	• The bin is designed for easy access, allowing users to deposit items. • The item is appropriate for detection, fitting within the system's size and weight limits and matching the data stored in the system. • The item can be detected properly if it meets the system's size and weight requirements and aligns with the information programmed into the system. • The waste bin has sufficient space for the item.
Postconditions	• The system detects the item placed in the bin and can identify it as plastic. • The disposal process follows according to the item's classification. • The system is continuously ready to identify more plastic items.
Actor	User, System

Flow of Activities	Basic Flow	
	Actor	System
	1. User places non-metal items in the bin.	1. The system activates detection of waste material.
	Alternative Flow	
	1. User placed multiple items.	1. The system might imply detection confusion.
	Exceptions Flow	
	1. User did not place any item.	1. The system will not detect anything
Notes/Issues		

Table 3 UC001

3.3.2. Use Case #2

Table 4 Use Case #2

UC001	**User**
Author	**Jose Kyle Mapalo**
Purpose	Enable users to dispose of metal items into a waste bin for automatic identification and sorting into the appropriate waste category.

<table>
<tr><td>Requirements Traceability</td><td colspan="2">The item must be near the sensors to identify if the item is Metal.
Items should be placed individually to ensure precise detection and sorting.</td></tr>
<tr><td>Preconditions</td><td colspan="2">The bin is designed for easy access, allowing users to deposit items.
The item is appropriate for detection, fitting within the system's size and weight limits and matching the data stored in the system.
The item can be detected properly if it meets the system's size and weight requirements and aligns with the information programmed into the system.
The waste bin has sufficient space for the object.</td></tr>
<tr><td>Postconditions</td><td colspan="2">The system detects the object placed in the bin and can identify it as plastic.
The disposal process follows according to the item's classification.
The system is continuously ready to identify more Metal items.</td></tr>
<tr><td>Actor</td><td colspan="2">User, System</td></tr>
<tr><td rowspan="3">Flow of Activities</td><td colspan="2">Basic Flow</td></tr>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>1. User places metal items in the bin.</td><td>1. The system activates detection of waste material.</td></tr>
</table>

	Alternative Flow	
	1. User placed multiple items.	1. The system might imply detection confusion.
	Exceptions Flow	
	1. User did not place any item.	1. The system will not detect anything
Notes/Issues		

Table 4 UC002

3.3.3. Use Case #3

UC001	**User**
Author	**Jose Kyle Mapalo**
Purpose	To track the waste bin’s fill level and alert users when it hits 75% capacity, helping manage waste promptly and avoid overflow.
Requirements Traceability	• The system needs sensors that can accurately measure the waste bin’s fill level. • The system should reliably gauge and notify when the bin reaches 75% of its capacity. • The waste bin needs to have suitable sensors installed for monitoring its level.
Preconditions	• The waste classification system is operational and functioning.

	• The waste bin is equipped with a level monitoring sensor. • The system's monitoring module is active and capable of processing fill level data. • The waste bin is easy to access for monitoring and has sufficient room for sensor installation.
Postconditions	• The system accurately detects and monitors the fill level of the waste bin. • The system sends an alert once the bin reaches 75% capacity. • The system is prepared to keep tracking the waste bin's fill level.
Actor	User, System
Flow of Activities	Basic Flow
	1. The system initiates its item detection feature. 2. The system analyzes the item and determines its category. 3. The system sends an alert or notification to notify the admin about the bin's fill level. 4. The system continues monitoring the bin and remains ready for future updates on the fill level.
	Alternative Flow
	The system identifies incorrect fill level sensor readings caused by environmental conditions or sensor issues.
	Exceptions Flow

	If the waste bin level sensor is malfunctioning or disconnected, the system will not be able to track the fill level.
Notes/Issues	

Table 5 UC003

3.3.4. Use Case #4

UC001	**User**
Author	**Jose Kyle Mapalo**
Purpose	To allow the administrator to manage the waste bin by removing waste when it reaches 75% capacity or is full, ensuring proper waste disposal and preventing overflow.
Requirements Traceability	• The system needs to keep track of the waste bin's fill level. • The system needs to keep track of the waste bin's fill level. • The administrator must have access to the waste bin for removal of waste.
Preconditions	• The waste classification system is operational and functioning. • The waste bin is equipped with an ESP8266 Wi-fi module. • The sensor is calibrated and functions correctly. • The system’s monitoring and notification modules are active.

Postconditions	• The system accurately monitors the waste bin’s fill level. • The system generates alerts when the bin reaches full. • The waste bin is managed accordingly, with waste removed as needed. • The system continues to monitor the bin and remains ready for future management actions.	
Actor	User, System	
Flow of Activities	Basic Flow	
	Actor	System
		1. The system issues an alert or notification to update administrators on the bin’s fill level.
	2. Housekeeping staff are notified and proceed to remove the waste from the bin.	
		3. The system refreshes the waste management status and gets ready for ongoing monitoring.
	Alternative Flow	

		1. The system did not send a notification that the bin has reached capacity.
	2. Housekeeping staff did not remove the waste from the bin.	3. The system will not change the status of the bin.
	Exceptions Flow	
	1. The ultrasonic sensor fails to operate or is disconnected 2. The Wi-Fi module fails to operate or is disconnected.	
Notes/Issues		

Table 6 UC004

3.4. Performance Requirements

The performance requirements for the Waste Segregation System ensure that the system operates efficiently and responsively. It must quickly classify waste items and provide timely feedback to users, maintaining accuracy and speed in sorting and system interactions. These requirements ensure that the system handles waste processing effectively and meets user expectations.

1. Wi-Fi connectivity

- **Requirements**: The web application and the system must be connected to the same network for proper functionality. If they are on different networks, communication between the application and the system will lead to issues with data transmission, delayed responses, or failure in synchronizing essential operations.

- **Rationale:** To maintain connection

2. Real Time Waste Detection

- **Requirements:** The system must classify each item as metal or other wastes.
- **Rationale:** Ensures prompt waste sorting to sustain efficient operation.

3. Servo Motor Response Time

- **Requirements**: The servo motor must act upon receiving signal from the sensor within 5 seconds below.
- **Rationale:** This ensures the proper flow of accurate waste sorting, helping maintain efficiency and reduce errors in the process.

4. Sensor Accuracy and Response Time

- **Requirements:** The sensor must detect the items properly and accurately.
- **Rationale:** Ensuring accuracy in detecting waste objects based on their size and positioning on the platform.

3.5. Safety and Security Requirements

Safety Requirements:

1. User Safety

- Incorporate automatic shutoff and clear warnings for moving parts to prevent injury.
- Ensure electrical components are insulated and meet safety standards.

2. Operational Safety

- Implement fail-safe to handle malfunctions safely, reverting to manual mode if necessary.
- Provide easy access for maintenance with clear, documented safety procedures.

3.6. Software Quality Attributes

4. Other Non-functional Requirements

4.1. Performance Requirements

The performance standards for the Smart Metal Segregation System guarantee efficient and responsive operation. It needs to swiftly categorize waste items and offer prompt feedback to users while maintaining both accuracy and speed in sorting and interactions. These standards ensure effective waste processing and align with user expectations.

Real-Time Metal Detection

Requirement:

The system must be capable of detecting metal objects in real time during the waste sorting process, ensuring immediate identification and separation.

Rationale:
Real-time detection is essential for maximizing efficiency and minimizing contamination of sorted materials. Quick identification allows for timely responses, reducing the risk of processing errors and improving overall system performance. This capability ensures that valuable metals are not lost and enhances the recycling process.

Sensor Processing Time

Requirement:

The system must have a sensor processing time of less than a specified threshold (e.g., 100 milliseconds) to ensure rapid detection and sorting of metal objects.

Rationale:

Minimizing sensor processing time is crucial for maintaining a high throughput in waste sorting operations. Faster processing allows for real-time adjustments and immediate actions, which helps prevent bottlenecks in the sorting line and enhances overall operational efficiency. This speed also contributes to better accuracy in sorting, ensuring that materials are classified correctly without delays.

Sensor Accuracy and Response

Requirement:

The sensors must achieve an accuracy rate of at least 95% in detecting and classifying metal objects, with a response time of less than 50 milliseconds.

Rationale:

High sensor accuracy is vital for ensuring that metals are correctly identified and sorted, reducing the likelihood of contamination in recycled materials. A quick response time enhances the system's efficiency by allowing for immediate sorting actions, minimizing delays, and improving throughput. Together, these factors contribute to r reliable waste processing and meet user expectations for precision and speed.

4.2. Safety and Security Requirements

The Metal Waste Segregation System must incorporate safety features to protect users from physical injury. This includes enclosing moving components, such as motors, and equipping the system with emergency stop buttons. Regular safety inspections and maintenance should be performed in accordance with safety standards like ISO 12100 and relevant local regulations for mechanical and electronic devices.

4.3. Software Quality Attributes

The Waste Segregation System should be user-friendly, ensuring that users can operate it with minimal training. It must also be dependable, consistently performing accurate waste classification and capable of integrating future upgrades or new features smoothly.

1. **Usability**

   Requirement: The system should be easy for users to operate with minimal training. The users should be able to start, stop, and monitor the system.
   Achieved By: Designing an intuitive user interface with clear instructions and feedback. Providing a user manual and conducting usability testing to ensure ease of use.

2. **Maintainability**

Requirement: The system must be maintainable with minimal downtime. Maintenance tasks should be documented clearly and be understandable within 30 minutes.
Achieved By: Implementing modular components that can be easily replaced or repaired. Providing comprehensive maintenance guides and training for support personnel.

3. **Reliability**

Requirement: The system must operate continuously for at least 8 hours and maintain a classification accuracy of 60-70% or higher.
Achieved By: Utilizing high-quality, tested components, and incorporating error-handling process in the software. Conducting an active testing to ensure consistent performance.

4. **Adaptability**

Requirement: The system should support integration with different or consistency of the sensors or additional features with minimal modification.
Achieved By: Designing the system with modular and configurable hardware and software interfaces. Providing clear documentation for integrating new components or features.

Appendix A – Data Dictionary/ ERD

Data Dictionary

Field Name	Data Type	Description
waste_id	Date	Unique identifier for each waste item.
collection_date	String	Date when the waste was collected.
location	String	Specific location on campus where the waste was generated (e.g., building name, floor).
weight	Float	Weight of the collected metal waste in kilograms.
segregation_status	String	Status of segregation (e.g., segregated, mixed, need manual sorting).
recycling_bin_id	Integer	Identifier for the recycling bin where the waste was deposited.
time_of_collection	Time	Time when the waste was collected.
collector_id	Integer	Identifier for the staff member who collected the waste.

Figure 8 Data Dictionary

ERD/ Entity-relationship diagrams

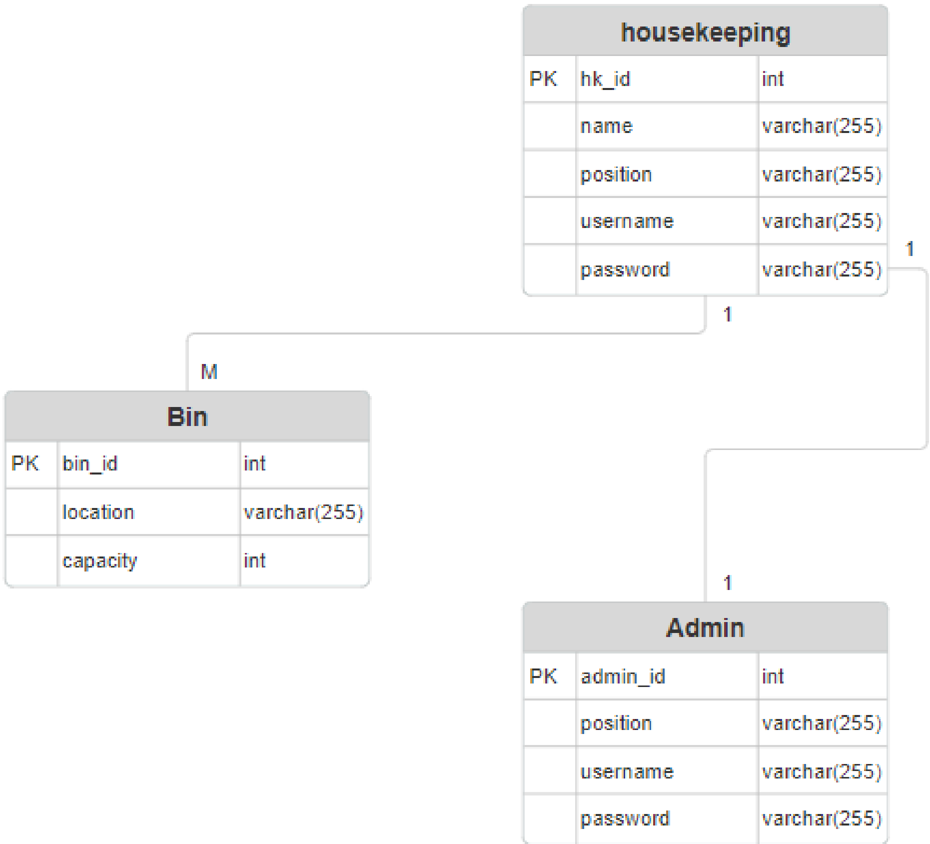


Figure 9 ERD/ Entity- Relation Diagram

Appendix B – Group Log

Schedule Control

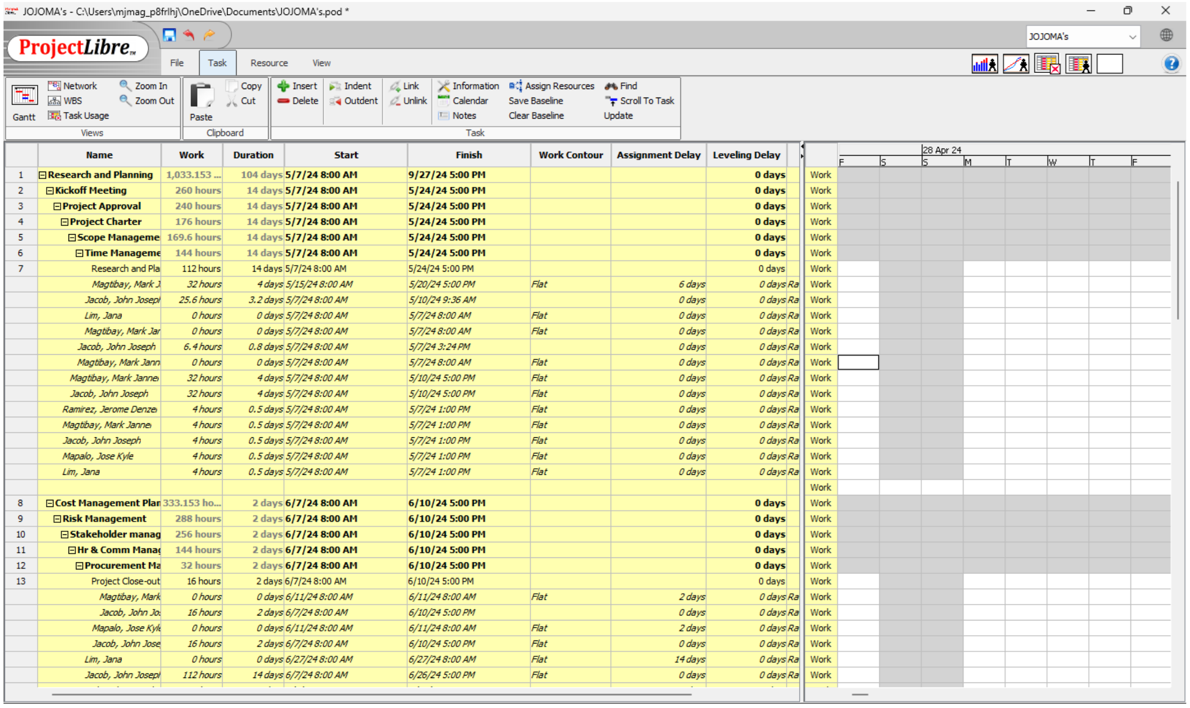
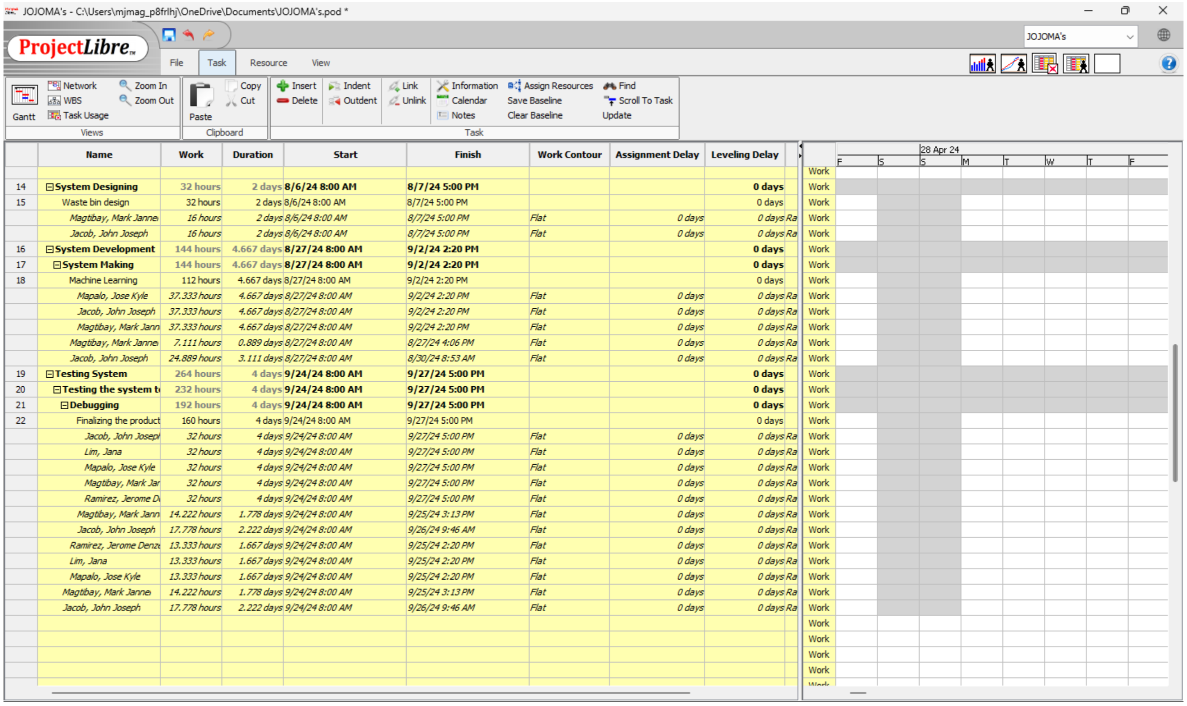
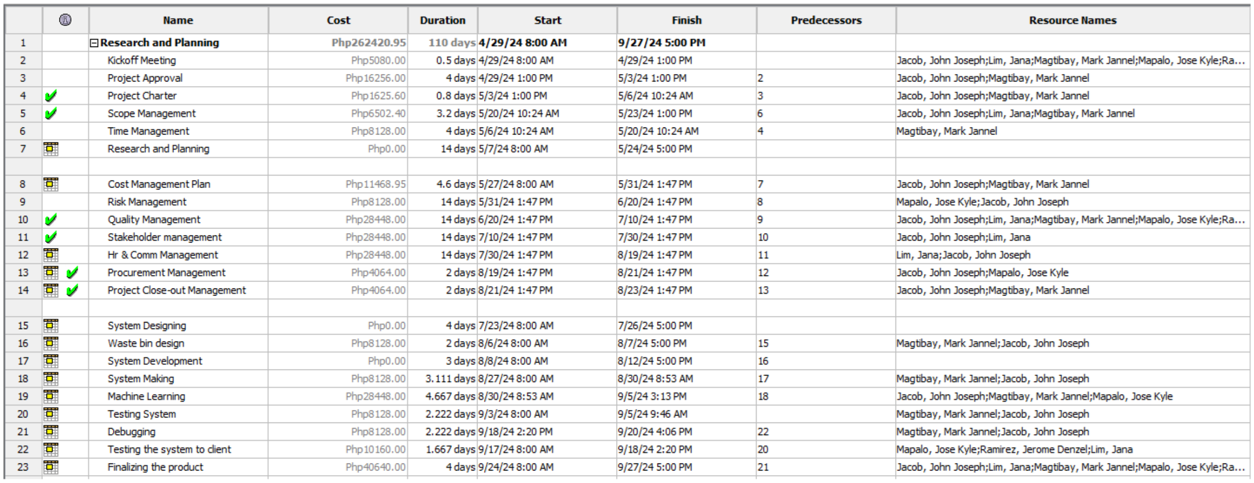


Figure 100 Project Libre Schedule Control



	Name	Work	Duration	Start	Finish	Work Contour	Assignment Delay	Leveling Delay		
										Work
14	⊟System Designing	32 hours	2 days	8/6/24 8:00 AM	8/7/24 5:00 PM			0 days		Work
15	Waste bin design	32 hours	2 days	8/6/24 8:00 AM	8/7/24 5:00 PM			0 days		Work
	Magtibay, Mark Jannel	16 hours	2 days	8/6/24 8:00 AM	8/7/24 5:00 PM	Flat	0 days	0 days	Ra	Work
	Jacob, John Joseph	16 hours	2 days	8/6/24 8:00 AM	8/7/24 5:00 PM	Flat	0 days	0 days	Ra	Work
16	⊟System Development	144 hours	4.667 days	8/27/24 8:00 AM	9/2/24 2:20 PM			0 days		Work
17	⊟System Making	144 hours	4.667 days	8/27/24 8:00 AM	9/2/24 2:20 PM			0 days		Work
18	Machine Learning	112 hours	4.667 days	8/27/24 8:00 AM	9/2/24 2:20 PM			0 days		Work
	Mapalo, Jose Kyle	37.333 hours	4.667 days	8/27/24 8:00 AM	9/2/24 2:20 PM	Flat	0 days	0 days	Ra	Work
	Jacob, John Joseph	37.333 hours	4.667 days	8/27/24 8:00 AM	9/2/24 2:20 PM	Flat	0 days	0 days	Ra	Work
	Magtibay, Mark Jann	37.333 hours	4.667 days	8/27/24 8:00 AM	9/2/24 2:20 PM	Flat	0 days	0 days	Ra	Work
	Magtibay, Mark Jannel	7.111 hours	0.889 days	8/27/24 8:00 AM	8/27/24 4:06 PM	Flat	0 days	0 days	Ra	Work
	Jacob, John Joseph	24.889 hours	3.111 days	8/27/24 8:00 AM	8/30/24 8:53 AM	Flat	0 days	0 days	Ra	Work
19	⊟Testing System	264 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM			0 days		Work
20	⊟Testing the system t	232 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM			0 days		Work
21	⊟Debugging	192 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM			0 days		Work
22	Finalizing the product	160 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM			0 days		Work
	Jacob, John Joseph	32 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM	Flat	0 days	0 days	Ra	Work
	Lim, Jana	32 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM	Flat	0 days	0 days	Ra	Work
	Mapalo, Jose Kyle	32 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM	Flat	0 days	0 days	Ra	Work
	Magtibay, Mark Jar	32 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM	Flat	0 days	0 days	Ra	Work
	Ramirez, Jerome D	32 hours	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM	Flat	0 days	0 days	Ra	Work
	Magtibay, Mark Jann	14.222 hours	1.778 days	9/24/24 8:00 AM	9/25/24 3:13 PM	Flat	0 days	0 days	Ra	Work
	Jacob, John Joseph	17.778 hours	2.222 days	9/24/24 8:00 AM	9/26/24 9:46 AM	Flat	0 days	0 days	Ra	Work
	Ramirez, Jerome Denze	13.333 hours	1.667 days	9/24/24 8:00 AM	9/25/24 2:20 PM	Flat	0 days	0 days	Ra	Work
	Lim, Jana	13.333 hours	1.667 days	9/24/24 8:00 AM	9/25/24 2:20 PM	Flat	0 days	0 days	Ra	Work
	Mapalo, Jose Kyle	13.333 hours	1.667 days	9/24/24 8:00 AM	9/25/24 2:20 PM	Flat	0 days	0 days	Ra	Work
	Magtibay, Mark Jannel	14.222 hours	1.778 days	9/24/24 8:00 AM	9/25/24 3:13 PM	Flat	0 days	0 days	Ra	Work
	Jacob, John Joseph	17.778 hours	2.222 days	9/24/24 8:00 AM	9/26/24 9:46 AM	Flat	0 days	0 days	Ra	Work

Figure 11 Project Libre Schedule Control



		Name	Cost	Duration	Start	Finish	Predecessors	Resource Names
1		⊟Research and Planning	Php262420.95	110 days	4/29/24 8:00 AM	9/27/24 5:00 PM		
2		Kickoff Meeting	Php5080.00	0.5 days	4/29/24 8:00 AM	4/29/24 1:00 PM		Jacob, John Joseph;Lim, Jana;Magtibay, Mark Jannel;Mapalo, Jose Kyle;Ra...
3		Project Approval	Php16256.00	4 days	4/29/24 1:00 PM	5/3/24 1:00 PM	2	Jacob, John Joseph;Magtibay, Mark Jannel
4	✔	Project Charter	Php1625.60	0.8 days	5/3/24 1:00 PM	5/6/24 10:24 AM	3	Jacob, John Joseph;Magtibay, Mark Jannel
5	✔	Scope Management	Php6502.40	3.2 days	5/20/24 10:24 AM	5/23/24 1:00 PM	6	Jacob, John Joseph;Lim, Jana;Magtibay, Mark Jannel
6		Time Management	Php8128.00	4 days	5/6/24 10:24 AM	5/20/24 10:24 AM	4	Magtibay, Mark Jannel
7		Research and Planning	Php0.00	14 days	5/7/24 8:00 AM	5/24/24 5:00 PM		
8		Cost Management Plan	Php11468.95	4.6 days	5/27/24 8:00 AM	5/31/24 1:47 PM	7	Jacob, John Joseph;Magtibay, Mark Jannel
9		Risk Management	Php8128.00	14 days	5/31/24 1:47 PM	6/20/24 1:47 PM	8	Mapalo, Jose Kyle;Jacob, John Joseph
10	✔	Quality Management	Php28448.00	14 days	6/20/24 1:47 PM	7/10/24 1:47 PM	9	Jacob, John Joseph;Lim, Jana;Magtibay, Mark Jannel;Mapalo, Jose Kyle;Ra...
11	✔	Stakeholder management	Php28448.00	14 days	7/10/24 1:47 PM	7/30/24 1:47 PM	10	Jacob, John Joseph;Lim, Jana
12		Hr & Comm Management	Php28448.00	14 days	7/30/24 1:47 PM	8/19/24 1:47 PM	11	Lim, Jana;Jacob, John Joseph
13	✔	Procurement Management	Php4064.00	2 days	8/19/24 1:47 PM	8/21/24 1:47 PM	12	Jacob, John Joseph;Mapalo, Jose Kyle
14	✔	Project Close-out Management	Php4064.00	2 days	8/21/24 1:47 PM	8/23/24 1:47 PM	13	Jacob, John Joseph;Magtibay, Mark Jannel
15		System Designing	Php0.00	4 days	7/23/24 8:00 AM	7/26/24 5:00 PM		
16		Waste bin design	Php8128.00	2 days	8/6/24 8:00 AM	8/7/24 5:00 PM	15	Magtibay, Mark Jannel;Jacob, John Joseph
17		System Development	Php0.00	3 days	8/8/24 8:00 AM	8/12/24 5:00 PM	16	
18		System Making	Php8128.00	3.111 days	8/27/24 8:00 AM	8/30/24 8:53 AM	17	Magtibay, Mark Jannel;Jacob, John Joseph
19		Machine Learning	Php28448.00	4.667 days	8/30/24 8:53 AM	9/5/24 3:13 PM	18	Jacob, John Joseph;Magtibay, Mark Jannel;Mapalo, Jose Kyle
20		Testing System	Php8128.00	2.222 days	9/3/24 8:00 AM	9/5/24 9:46 AM		Magtibay, Mark Jannel;Jacob, John Joseph
21		Debugging	Php8128.00	2.222 days	9/18/24 2:20 PM	9/20/24 4:06 PM	22	Magtibay, Mark Jannel;Jacob, John Joseph
22		Testing the system to client	Php10160.00	1.667 days	9/17/24 8:00 AM	9/18/24 2:20 PM	20	Mapalo, Jose Kyle;Ramirez, Jerome Denzel;Lim, Jana
23		Finalizing the product	Php40640.00	4 days	9/24/24 8:00 AM	9/27/24 5:00 PM	21	Jacob, John Joseph;Lim, Jana;Magtibay, Mark Jannel;Mapalo, Jose Kyle;Ra...

Figure 122 Project Libre Schedule Control