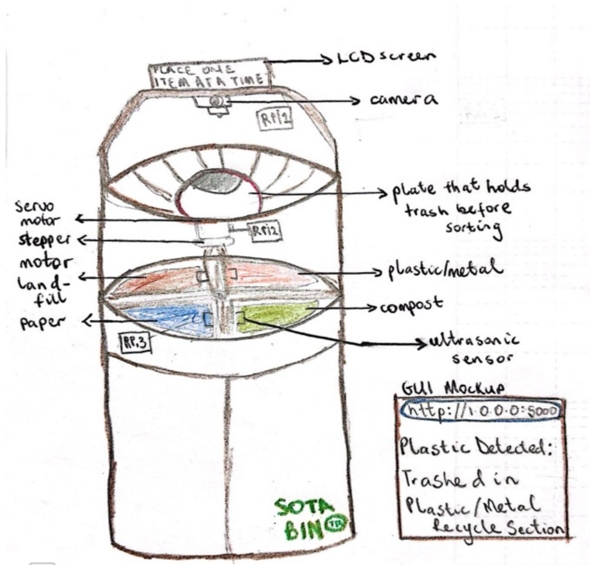
SYSC3010
Computer Systems Development Project

Sota Bin:
State-of-the-Art Sorting for a Sustainable Future

Final Project Report



Group L2-G3
Uchenna Obikwelu, 101241887
Adeyehun Folahanmi, 101237546
Dearell Tobenna Ezeoke, 101245819
Emeka Anonyei, 101209704
Tobiloba Ola, 101244412

TA: Afsoon Khodaee

April 7th, 2025

Table of Contents

1. Project Description

1.1 Problem Statement

Waste misclassification remains a persistent issue that contaminates recyclable streams, increases landfill contributions, and reduces overall recycling efficiency. A major reason behind this is a lack of public knowledge about proper sorting. Conventional multi-compartment bins rely solely on user discretion, often resulting in incorrect disposal and frequent sorting errors.

1.2 Motivation

Our goal was to design an intelligent waste bin that automates sorting and educates users through real-time feedback. Sota Bin addresses this issue by classifying waste into the correct category and instantly informing users of their sorting decisions via a connected web interface.

During development, we also identified another critical issue: in high-traffic public spaces, bins can overflow quickly if not monitored in real time. Overflowing bins not only lead to unhygienic and unpleasant public environments, but also discourage proper waste disposal - causing otherwise recyclable materials to end up as litter or landfill. To address this, we implemented a bin fullness detection feature that alerts both users and managers when any section of the bin is nearing capacity, ensuring timely maintenance and clean, efficient operation.

1.3 System Features and Users

The key features of Sota Bin include:

1. Classifies waste into four categories: landfill, paper, plastic/metal, and compost.
2. Provides real-time feedback to users through the website to educate them on how their waste was classified.
3. Alerts users and managers when a bin section is full, using the Sense HAT (on-device LED), the manager web dashboard, and automated email notifications.
4. Displays analyzed sorting data over time for both users and managers.
5. Allows managers to reset the bin status remotely from the website after the bin has been emptied.

Key Users:
Sota Bin is intended for use in public spaces such as schools, offices, malls, and libraries. There are two main user types:

- Everyday users who interact with the bin and receive feedback.
- Managers who monitor bin status, receive alerts, and review usage data through the web interface.

1.4 Overview of Design Solution

The system consists of four interconnected Raspberry Pi modules: RPi1, RPi2, RPi3, and RPi4. Each module performs a specific role in the waste classification and sorting process.

RPi1 captures images of the waste item using a camera and applies OpenCV processing (HSV, edge detection, brightness filtering) to classify the material. It also uses a moisture sensor (via ADS1115) to detect compost and an inductive proximity sensor to detect metal. Once classification is complete, it logs the data locally, sends the result to RPi2 via WebSocket, and uploads it to Firebase for real-time display on the web interface.

RPi2 receives the classification and controls a stepper motor and servo to move the item into the appropriate bin section.

RPi3 and RPi4 monitor bin fill levels using infrared reflective (IR) sensors. RPi3 handles compost and paper sections, while RPi4 manages landfill and plastic/metal. Both display bin status using Sense HATs, log data locally, and sync with Firebase for remote access.

When a bin section becomes full, the system updates Firebase and sends an email alert to the manager. A web interface shows the classification result to users in real time, displays live bin status, provides charts for analyzing sorting data, and allows managers to reset the bin after it is emptied.

2. Final Design Solution

2.1 Deployment Diagram

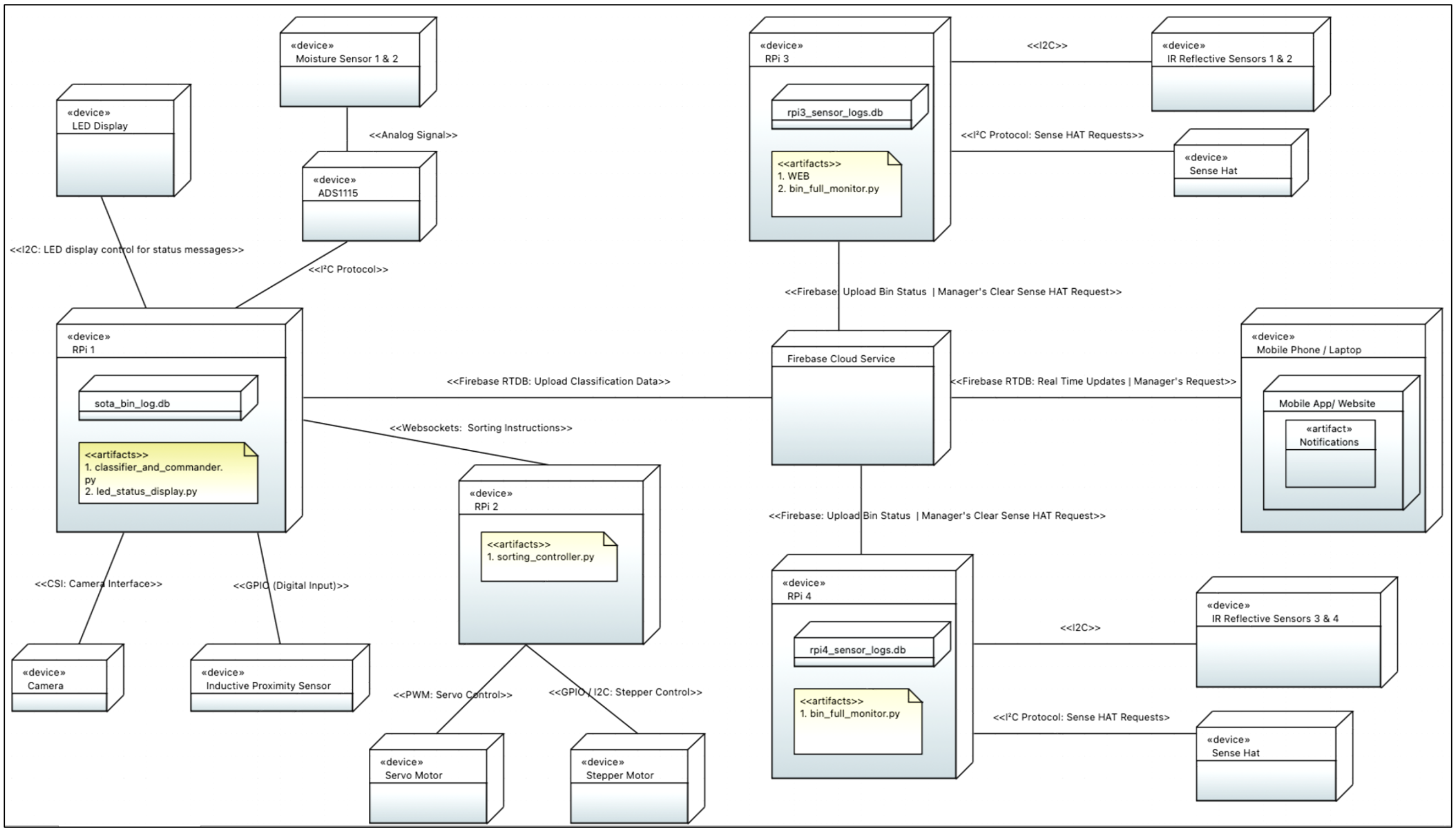


Figure 1: Sota Bin - System Deployment Diagram

This deployment diagram shows the hardware-software architecture of the Sota Bin system, consisting of four Raspberry Pi modules (RPi1–RPi4). Each Pi is assigned specific tasks: RPi1 handles classification, RPi2 controls sorting, and RPi3/RPi4 monitor bin fill levels.

Peripheral connections use the following protocols:

- **CSI**: Camera to RPi1
- **I2C**: Moisture sensors (via ADS1115), IR reflective sensors, and Sense HATs
- **GPIO**: Inductive proximity sensor
- **PWM/I2C**: Servo and stepper motor control
- **WebSocket**: RPi1 sends classification to RPi2
- **Firebase RTDB**: Syncs classification data, bin status, and manager actions to the cloud

RPi1, RPi3, and RPi4 log data locally using SQLite. Firebase enables real-time updates to the web interface, where users see live classification results and bin status. Managers can view sorting analytics, receive email alerts, and remotely reset bin states.

2.2 Message Protocol Table

2.2.1 RPi1- Collect and Classify

Sender	**Receiver**	**Purpose**	**Format / Parameters**	**Storage Location**
Camera	RPi1	Capture image on motion detection	Image (OpenCV processed: HSV, edges, brightness)	/RPi1/images/
Moisture Sensors (via ADS1115)	RPi1	Read moisture values for compost	JSON: {sensor1, sensor2}	/RPi1/sota_bin_log.db
Inductive Proximity Sensor	RPi1	Detect presence of metal	JSON: {metal: 1 or 0}	/RPi1/sota_bin_log.db
RPi1	Local DB	Log sensor data and classification	JSON: {material, bin}	/RPi1/sota_bin_log.db
RPi1	Firebase	Upload classification data for real time updates on website	JSON: {material, bin}	Firebase Realtime Database

2.2.2. RPi2 – Sort and Actuate

Sender	**Receiver**	**Purpose**	**Format / Parameters**	**Storage Location**
RPi1	RPi2	Instruct sorting based on classification	JSON: {type, metal, bin}	Not stored
RPi2	Motors (Servo/Stepper)	Move sorting plate	Angle / Step pulses	Not stored

2.2.3. RPi3 and RPi4 – Bin Full Detection & Alerts

Sender	**Receiver**	**Purpose**	**Format / Parameters**	**Storage Location**
IR Reflective Sensors	RPi3 / RPi4	Detect if bin section is full	Digital: 1 = Full, 0 = Not Full	/RPi3/sota_bin_log.db /RPi4/sota_bin_log.db
RPi3 / RPi4	Firebase	Send bin full status update	JSON: {bin, full}	Firebase Realtime Database
RPi3 / RPi4	Sense HAT	Light up LED for full bin	LED (Red/Green/…)	Not Stored

2.2.4. Firebase – Dashboard Sync and Remote Commands

Sender	**Receiver**	**Purpose**	**Format / Parameters**	**Storage Location**
Firebase	Web Interface	Update dashboard with latest data (Classification and bin full status)	JSON (automated sync)	Firebase Realtime Database
Web Interface	Firebase	Manager clears full bin alert	JSON: {action: reset, bin}	Firebase Realtime Database
Firebase	RPi3 / RPi4 Sense HAT	Clear Sense HAT	LED (Clear)	Direct Execution

2.3 Top-Level Sequence Diagram of Main Scenario(s)

2.3.1 Sequence Diagram 1: Classification + Sorting Flow

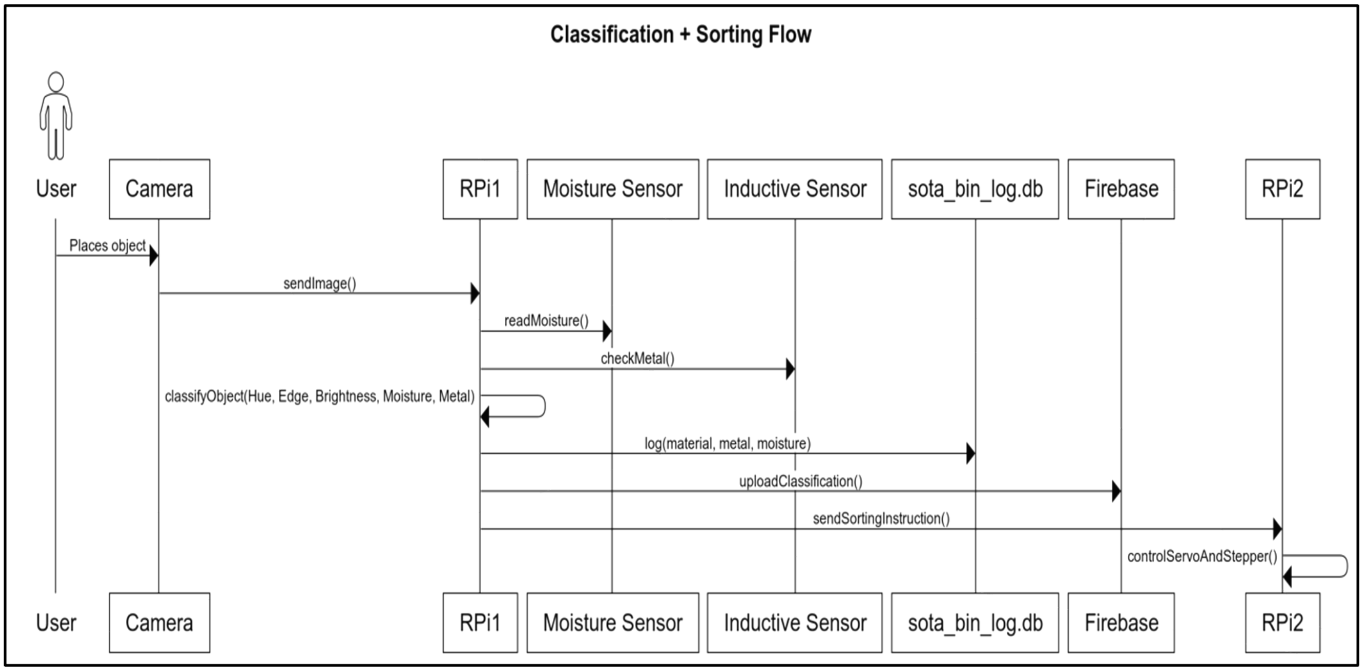


Figure 2: Classification + Sorting Flow Sequence Diagram

2.3.2 Sequence Diagram 2: Bin Full Detection and Sense HAT Alert

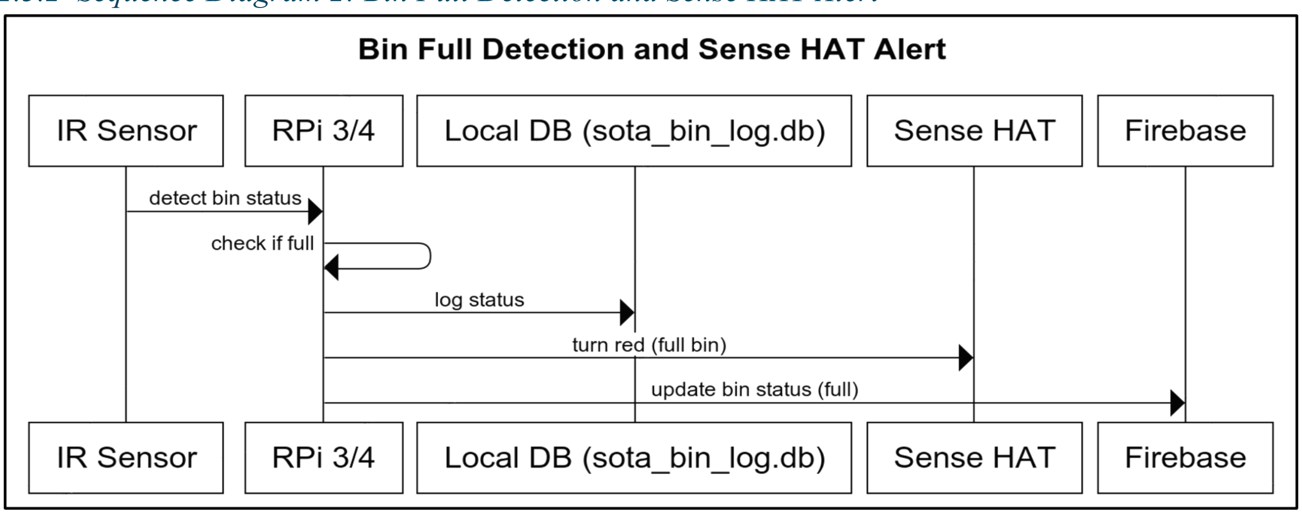


Figure 3: Bin Full Detection and Sense HAT Alert Sequence Diagram

2.3.3 Sequence Diagram 3: Real-Time Updates to Web Interface

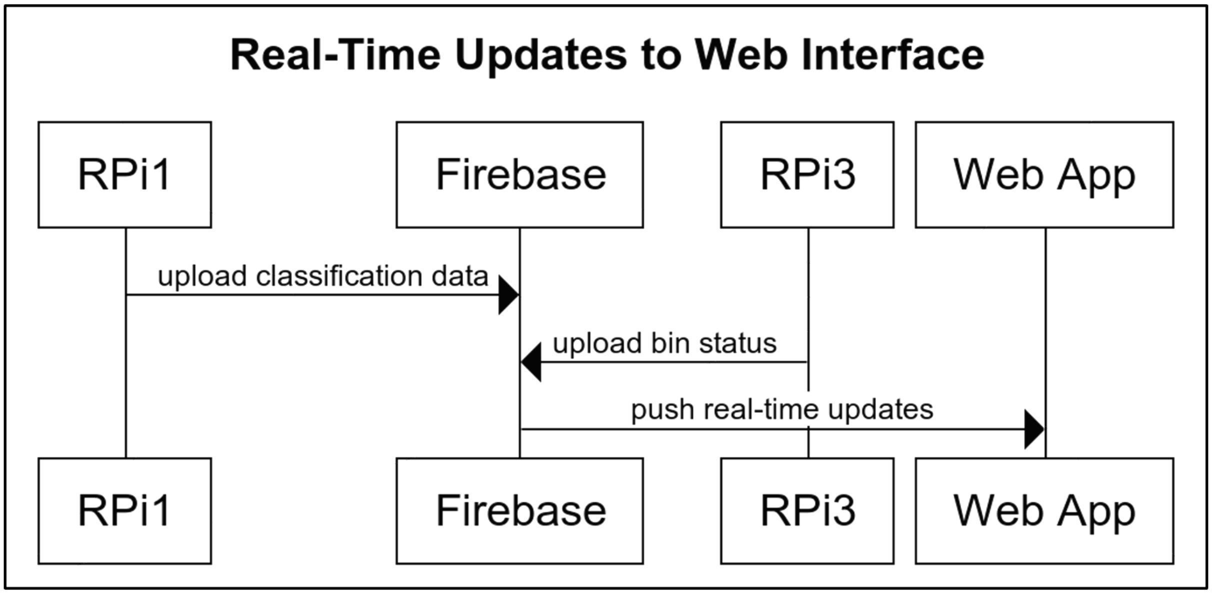


Figure 4: Real-Time Updates to Website

2.3.3 Sequence Diagram 4: Manager Resets Full Bin Alerts

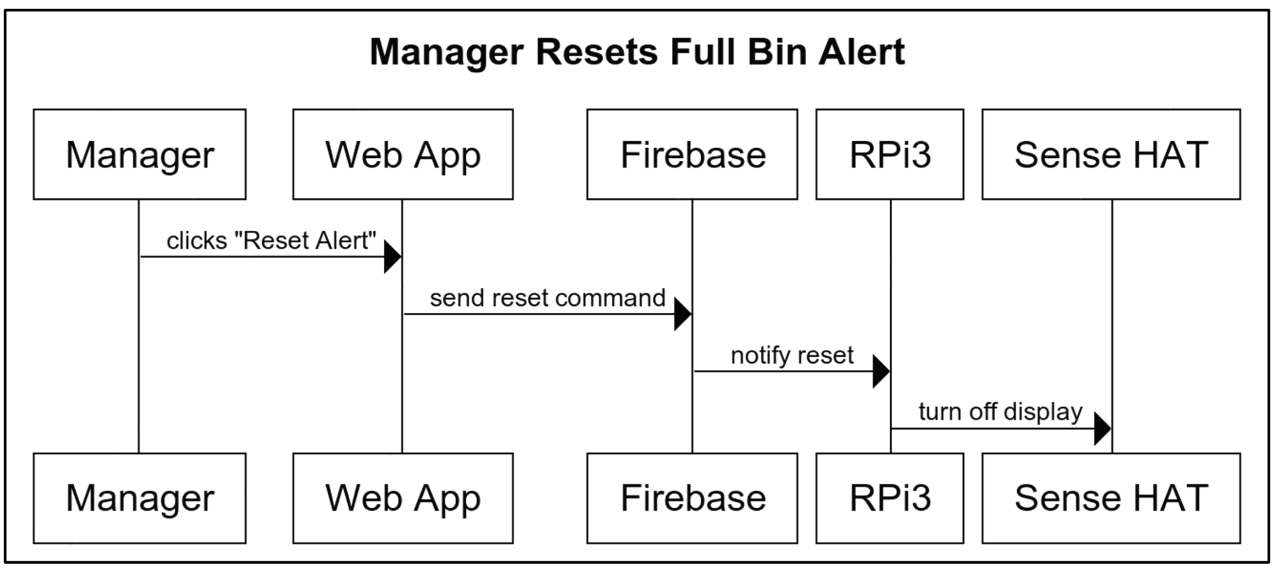


Figure 5: Manager Resets Full Bin Alerts

3. Discussion of Final Design

3.1 Final Design Overview

The final system uses a rectangular plastic bin divided into four sections: landfill, compost, paper, and plastic/metal. Waste is placed on a cardboard classification plate equipped with a moisture sensor, metal sensor, and a camera. Classification is performed using sensor readings and OpenCV image analysis. A stepper motor rotates the sorting plate while a servo controls a flap to direct the item. Infrared reflective sensors monitor bin fullness. Data is logged locally and synced with Firebase, enabling real-time classification feedback, bin status updates, and analytics via the website. The build uses cardboard, tape, and glue for structural simplicity and accessibility. See Figures 12 – 16 for images of physical implementation.

3.2 Changes from the Original Proposal

While the overall system architecture remained consistent, a few critical changes were made to improve accuracy, performance, and usability:

1. **IR Sensor Replaced Ultrasonic**: Chosen for more stable short-range detection and simpler wiring.
2. **Camera Replaced Color Sensor**: Due to better accuracy and flexibility with OpenCV.
3. **Moisture Sensor Added**: Improves classification by distinguishing compost from paper.
4. **Fourth RPi Added**: Offloaded IR processing, reducing lag and improving reliability.
5. **Website & Database Redesign**: Added bin-level tracking, real-time updates, charts, and remote reset options.

3.3 Testing Observation and Findings

1. **IR vs. Ultrasonic:** IR sensors gave more consistent readings and easier integration.
2. **Camera vs. Color Sensor:** Color sensors failed during testing; camera + image processing worked reliably.

3. **System Performance:** With three RPis, the system lagged. Adding a fourth improved responsiveness and stability.

4. Contributions

4.1 Code Contributions

Authors	Code Piece	Description
Uchenna Obikwelu	Classifier_and_commander.py	Handles classification via camera & moisture sensor; updates Firebase and sends sorting command to RPi2.
Folahanmi Adeyehun	Web_design, lcd_simulation.py	Developed the web interface and managed LED display messaging on RPi1.
Tobiloba ola	Final_motor.py	Receives commands and controls stepper + servo motors for waste sorting.
Dearell Tobenna Ezeoke, Emeka Anonyei	Bin_full_monitor.py	Detects bin fullness with IR sensors, logs data, and triggers notifications via Firebase.

4.2 Author of Sections in This Document

Document section	Author
Project description	Dearell Tobenna Ezeoke
Final design solution	Uchenna Obikwelu
Discussion of Final Design	Folahanmi Adeyehun
Contributions	Tobiloba ola
Reflection	Emeka Anonyei
Appendices	Dearell Tobenna Ezeoke and Tobiloba ola

All team members contributed to system testing and overall implementation.

5. Reflections

The Sota Bin project was an incredibly rewarding learning experience for our team. Overall, it went well, though we faced a few challenges. One key lesson was that integrating sensors and actuators into a working system takes more time than expected. We struggled with the ADS sensor initially, but after spending time troubleshooting, we realized the issue was with the lack of a properly soldered version. Rather than wasting more time, we bought a pre-soldered one and moved forward.

Another challenge was with the motor. Initially, we planned to use a larger NEMA motor for the sorting mechanism, but we were unable to get it working with our system. After much trial and error, we switched to a smaller motor, which worked well enough to allow us to proceed with the project. It was a good reminder that in engineering, things don't always go as planned, and being flexible is key.

We also had to make a switch in sensors. The ultrasonic sensors, initially chosen for bin fullness detection, weren't providing consistent results, so we replaced them with infrared (IR) reflective sensors. The IR sensors proved to be more reliable and easier to integrate, and they worked well in detecting when the bins were full.

Through this project, we learned how essential teamwork and communication are for success. Dividing the work based on our strengths helped us move efficiently, and using four Raspberry Pis allowed us to manage the system's complexity by splitting the workload. We also realized the importance of considering the physical space and wiring constraints in our design. Initially, we thought the bin would be small and simple, but once we started, we found that the wiring and components required more space. We opted for a larger bin, which made everything easier to manage.

If we had more time, we would focus on integrating AI for better waste classification using the camera, improving the physical design with 3D modeling for durability, and exploring additional features like automatic weight tracking. This project also showed us how technology can make a real environmental impact, and we're excited about the potential for future improvements and extensions.

This project reinforced that engineering requires problem-solving, adaptability, and collaboration. We encountered challenges along the way but overcoming them together was one of the most valuable aspects of the project.

6. Appendix A: README

Repository Structure

Folder	Description
SOTA BIN MAIN/	Contains Python implementation for each Raspberry Pi subsystem and website code. • RPi1: Object detection and classification • RPi2: Motor control • RPi3: Bin full monitoring and website scripts • RPi4: Bin full monitoring • configs/: Firebase credentials and constants.
SOTA BIN TEST/	Contains unit tests for hardware and software. • HARDWARE_TEST/ and SOFTWARE_TEST/: Unit test scripts for hardware and software components. • Other folders for end-to-end testing.
Project_Images/	Contains key photos from the project (system diagram, peripheral wiring, bin full detection, motor setup, team images).
WeeklyUpdates/	Weekly logs of team members' progress, tasks, and milestones.
README.md	Top-level documentation for understanding and running the system.

Installation Instructions

1. Hardware Setup

- **RPi1**: Camera, IR reflective sensor, metal sensor, moisture sensor, LCD display
- **RPi2**: Stepper motor + A4988 driver, servo motor
- **RPi3**: IR reflective sensors, Sense HAT
- **RPi4**: IR reflective sensors, Sense HAT
- **Power**: 5V USB for RPis, 12V adapter for metal sensor

2. Software Setup

```
# Clone the repository
$ git clone https://github.com/CU-SYSC3010W25/sysc3010-project-l2-g3

# Navigate to the appropriate directory (example for RPi1)
$ cd "SOTA BIN MAIN/RPi1"

# Install dependencies (adjust based on your Pi)
```

$ pip install opencv-python firebase-admin RPi.GPIO sense-hat websocket-client

- Add your Firebase firebase_key.json to the correct directory (e.g., RPi1/, RPi3/, etc.)
- Update scripts with your Firebase Realtime Database URL
- Ensure Wi-Fi connection is stable on all RPis.

3. Firebase Setup

- Create Firebase project
- Enable Realtime Database
- Download firebase_key.json and add to each Pi folder except for RPi2 (doesn't communicate with Firebase)

How to Run the System

1. On RPi1 (Object Classification):

```
cd "SOTA BIN MAIN/RPi1"
python3 classifier_and_commander.py
```

Then, in another terminal:

```
cd "SOTA BIN MAIN/RPi1"
python3 lcd_simulation.py
```

2. On RPi2 (Motor Control):

```
cd "SOTA BIN MAIN/RPi2"
python3 final_motor.py
```

3. On RPi3 (Bin Full Monitoring & Web Interaction):

```
cd "SOTA BIN MAIN/RPi3"
python3 monitor_bins.py
```

4. On RPi4 (Optional Monitoring):

```
cd "SOTA BIN MAIN/RPi4"
python3 bin_full_monitor.py
```

5. On RPi3 (Web Dashboard):

```
cd "SOTA BIN MAIN/RPi3/WEB"
```

Open index.html in browser or serve via local server

How to Know It’s Working

- Running the SOTA BIN TEST/HARDWARE_TEST and SOTA BIN TEST/SOFTWARE_TEST scripts should pass.
- The LCD screen continuously displays "Place one item at a time."
- When waste is placed:
    - Object is detected and classified.
    - Classification appears in RPi1 local DB, Firebase, and Web UI.
    - Motors activate (RPi2) to direct the waste to the correct bin section.
- When a bin is full:
    - Sense HAT displays an alert.
    - Web interface shows bin status, and the manager is notified by email.
- When reset is clicked (via web):
    - Firebase logs the reset.
    - RPi3 clears the Sense HAT display.

7. Other Appendices

7.1 Appendix B: System Design Diagrams

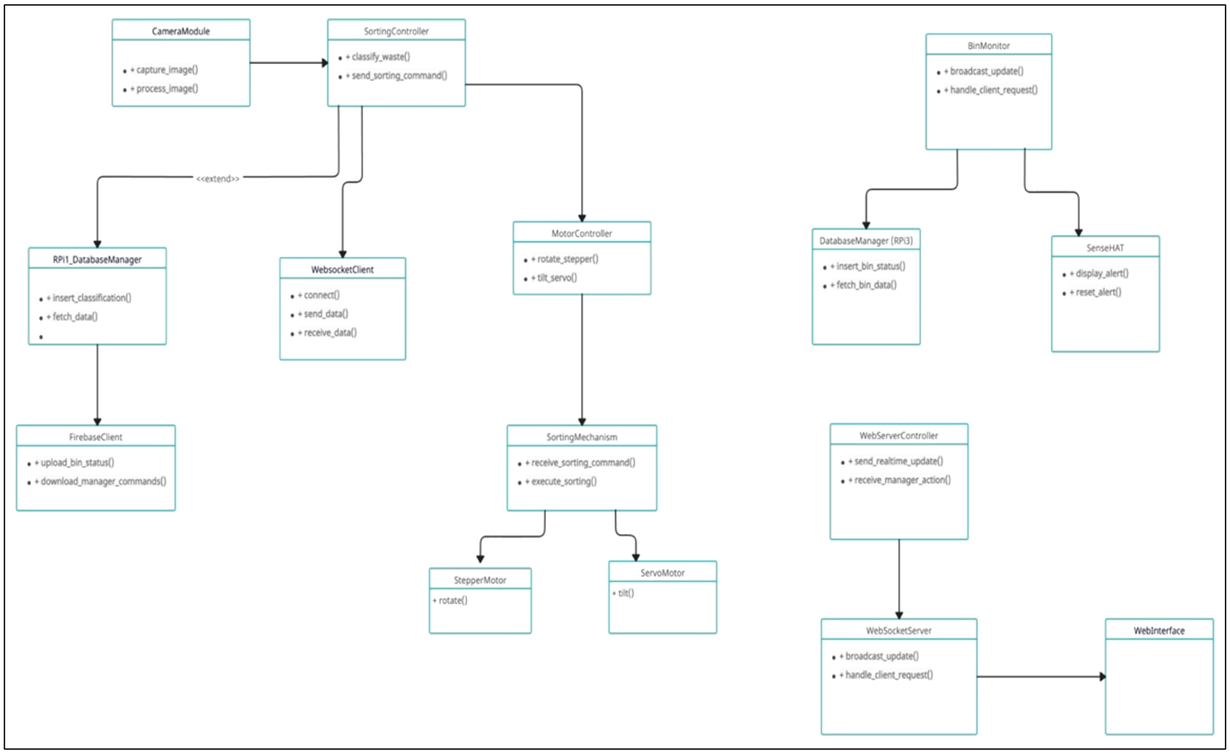


Figure 6: Sota Bin Class Diagram

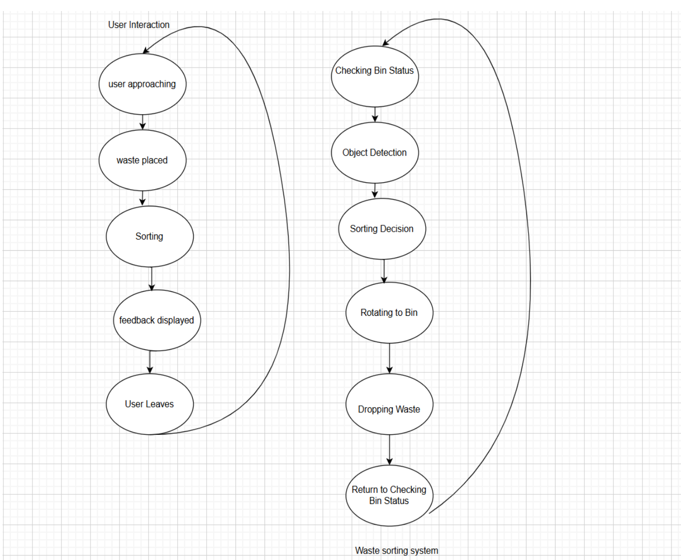


Figure 7: Sota Bin Flow Chart Diagram

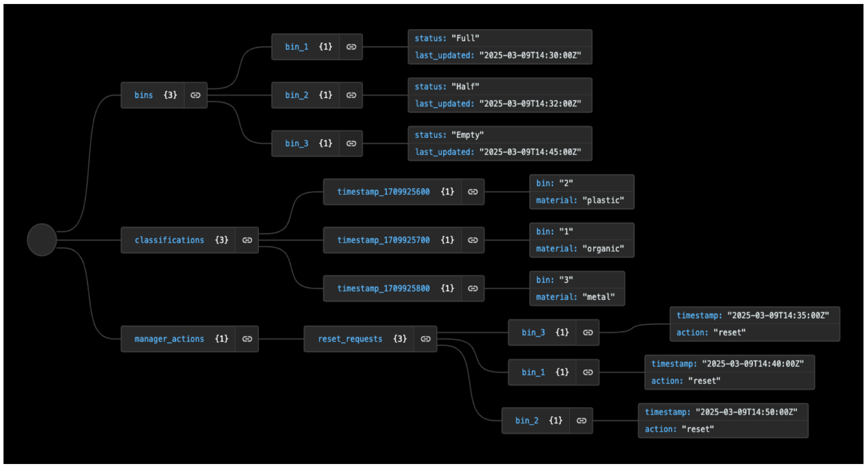


Figure 8: Firebase Database Schematic

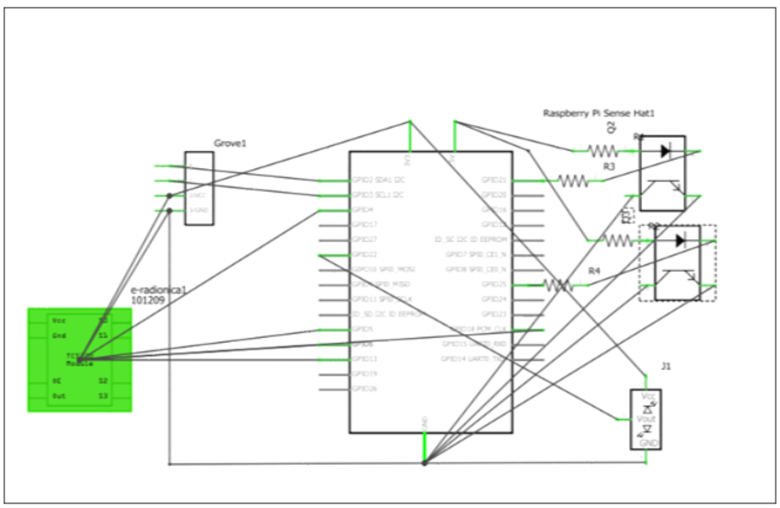


Figure 9: RPi1 and connected devices schematic

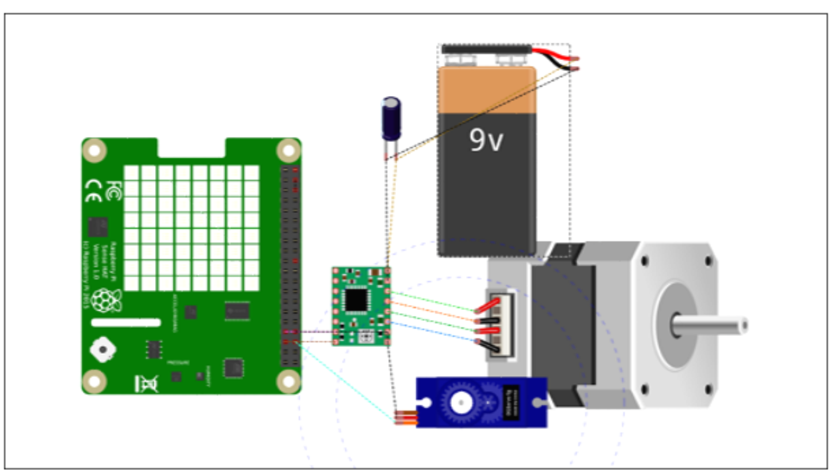


Figure 10: RPi2 and connected devices schematic

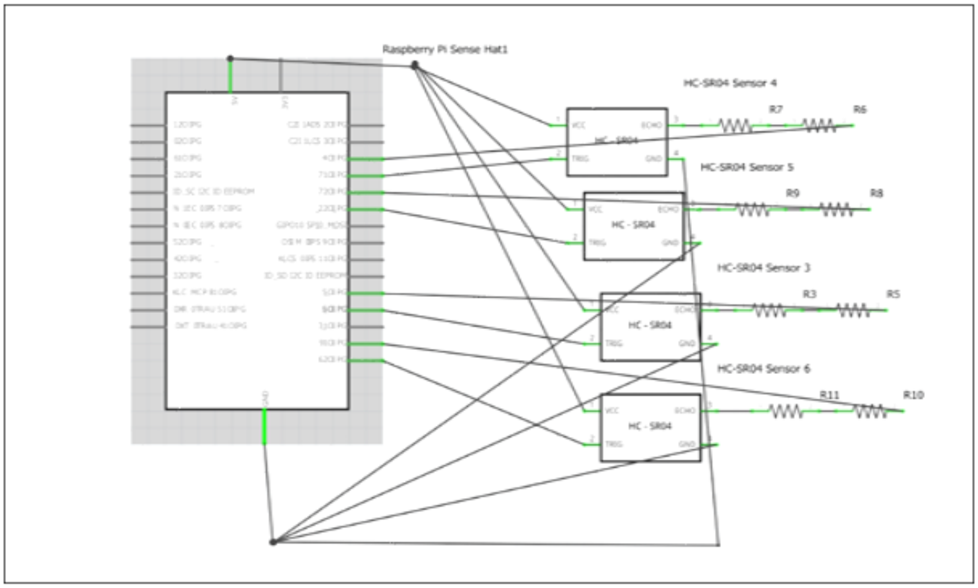


Figure 11: RPi3/RPi4 connected devices schematic

7.2 Appendix C: Images of Physical Implementation Sota Bin

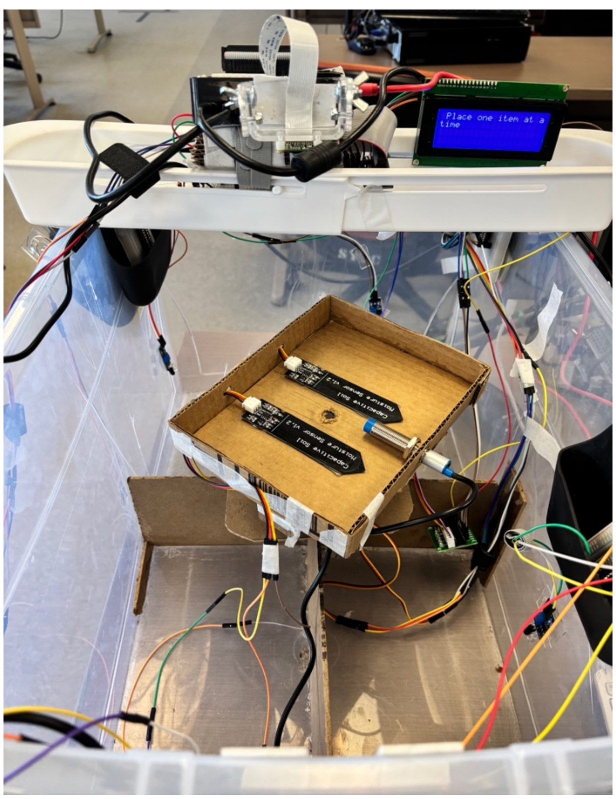


Figure 12: Sota Bin Top view

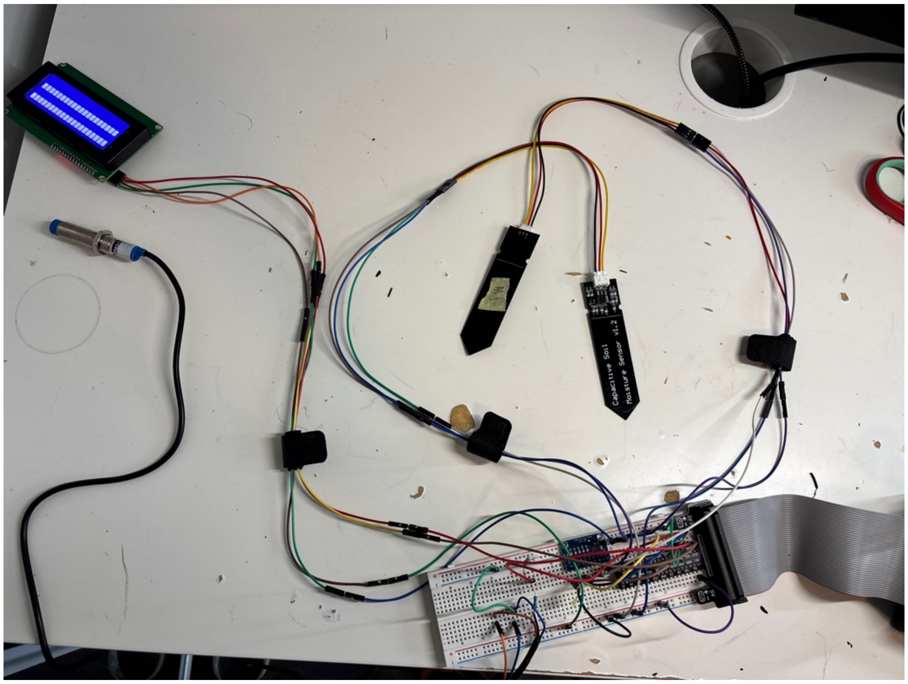


Figure 13: RPi1 peripheral circuit


Figure 14: Sota Bin actuator setup

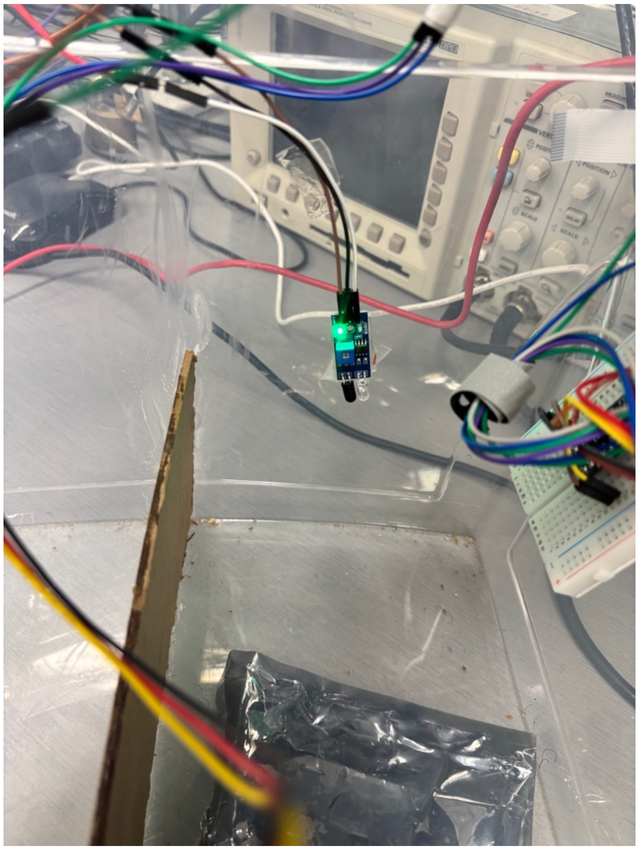


Figure 15: IR sensor placed to detect when bin section is full


Figure 16: Sense Hat for bin full