

Smart Waste Management System Using IoT and Real-Time Monitoring

CSS533 (Project Proposal template)

Name – Sahithi Chimakurthi

Student ID - 2303017

Project Description:

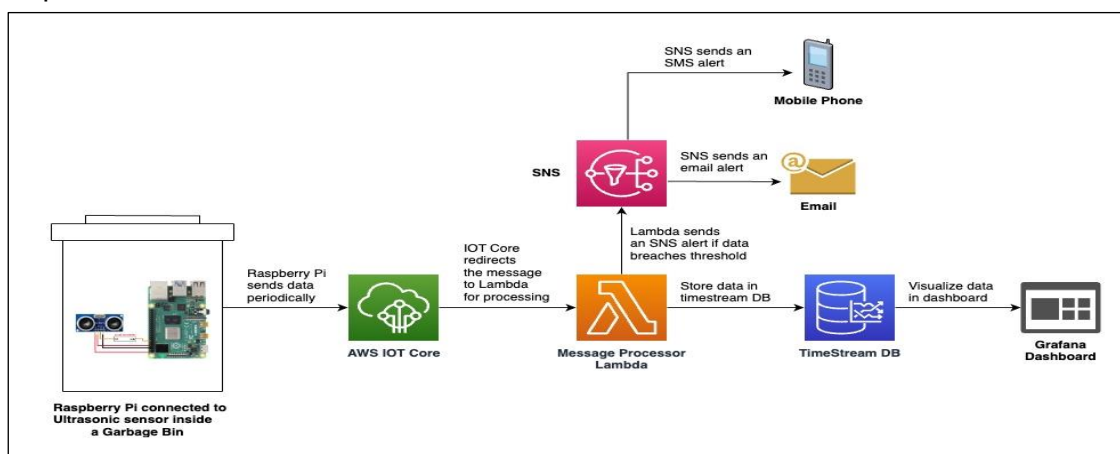
The proposed project is a smart waste management system that utilizes a Raspberry Pi and an ultrasonic sensor to monitor the fill level of a waste bin. The sensor will measure the distance between the sensor and the waste in the bin and send data to AWS IoT Core every 2 hours. IoT Core will forward the data to AWS Lambda, which will process it and store it in Amazon Timestream with a timestamp for tracking.

The system is configured with fixed thresholds, and when the waste level exceeds these thresholds, AWS Lambda will trigger alerts:

- **At 70% full:** A **Sev3 alert** will be triggered.
- **At 90% full:** A **Sev2 alert** will be triggered.
- **At 100% full:** A **Sev1 alert** will be triggered.

When these thresholds are breached, Lambda will send an SNS (Simple Notification Service) alert. The SNS service will send both an **email** and an **SMS** notification to the user's mobile phone, ensuring timely action when bins are near capacity.

The data, along with the alerts, can be visualized on a Grafana dashboard, providing a clear, real-time display of the bin's status, waste levels. The motivation for this project stems from the need to improve waste collection efficiency. Overfilled bins can lead to unsanitary conditions and environmental issues, while empty bins result in unnecessary trips for collection. By collecting and analyzing data in real-time, waste collection routes can be optimized, reducing costs and minimizing environmental impact.



Research issues include:

- **Sensor Accuracy:** Ensure the ultrasonic sensor provides precise readings, even in different environmental conditions or with varying waste types.
- **Data Management:** Efficiently handle and store large amounts of real-time sensor data in Amazon Timestream.
- **System Scalability:** In future if I want to monitor multiple bins, how will I scale the system and handle the increased data load?
- **Real-Time Updates:** Ensure that the data is transmitted from the sensor to the cloud and then visualized in Grafana with minimal delay.
- **Power Efficiency:** If the system needs to run for long periods on battery, how can I optimize power usage to avoid frequent recharging or downtime?

Expected Challenges:

1. **Sensor Calibration:** Ensuring the ultrasonic sensor provides consistent and accurate readings in different environments (e.g., indoor, outdoor, varying bin sizes).
2. **Data Latency and Transmission:** Handling any delays in transmitting data from the Raspberry Pi to the cloud service (Timestream) and ensuring real-time updates in Grafana.
3. **Power Efficiency:** The system needs to be power-efficient if used in remote or off-grid locations.
4. **Scaling the System:** In future if I want to integrate multiple bins across different locations and then, how efficiently can I manage the flow of data into a centralized system.

Estimated Time Schedule:

Week 1:

- 1.) Research and selection of components (Raspberry Pi, ultrasonic sensor, power management options). Setting up Raspberry Pi and installing necessary libraries.
- 2.) Connecting the ultrasonic sensor to the Raspberry Pi and writing the code to retrieve sensor data. Testing the accuracy of sensor measurements.

Week 2:

- 1.) Setting up Amazon Timestream and storing sensor data in a time-series format. Ensuring data is properly formatted and logged.
- 2.) Integrating Grafana for data visualization. Developing a dashboard to display bin status and any historical data trends.

Week 3:

- 1.) Testing the full system in a real-world scenario, troubleshooting any issues related to sensor data, communication, or data visualization.

Week 4:

- 1.) Final testing, optimization, and project documentation.