UNIVERSITY OF INFORMATION TECHNOLOGY

VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

---o0o---

Advanced Internet of Things Technologies

TOPIC

IOT SYSTEM SUPPORTS TRASH MANAGEMENT FOR APARTMENTS

Lecturer: Assoc. Prof. Le Trung Quan

KS. Nguyen Van Bao

Class: NT532.O22.CTTT

Group: 8

Number	Student Name	Student ID
1	Nguyễn Nhất Thưởng	20522000
2	Lê Văn Sỹ	20521854

- 04-04-2024 -

Contents

I. INTRODUCTION

1. Reason choose this topic

Smart bin was chosen as the project topic because of the practical benefits it brings. Thanks to the ability to automatically open the lid and warn that the trash can is full, the system makes waste collection effective and ensures apartment hygiene. Limiting manual inspection and optimizing the garbage collection process saves time and effort for both residents and management staff.

2. Description function

Function	**Description**
Automatically open the lid	Provides hands-free operation and improves hygiene by eliminating the need to touch the lid. When someone approaches the trash can keep a distance of 20cm. The trash can will automatically open.
Warn that the trash can is full	Warns users when the trash is nearly full. Notifications are sent to users through a website. There are three levels of trash fullness: low, medium and high. Corresponding to the three colors green, yellow, and red.

3. Technical

a. Hardware platform

Number	Name	Function
1	Arduino Uno R3	Microcontroller board responsible for controlling the overall operation of the system.
2	SRF04 Ultrasonic Distance Measuring	Detects the presence of objects within a certain range for proximity sensing.
3	Servo SG90	Controls the movement of the lid of the trash bin, enabling automatic opening and closing.
4	NODEMCU-ESP8266 Wifi	Provides WiFi connectivity to the system, enabling communication with remote servers or devices.

b. Software platform

Number	Name	Function
1	Visual Studio Code	Integrated Development Environment (IDE) used for writing, compiling, and uploading code to server and client.
2	Arduino IDE	Text editor used for writing and editing code for the Arduino Uno R3 and NODEMCU-ESP8266 module.

II. SYSTEM ARCHITECHTURE

1. Description

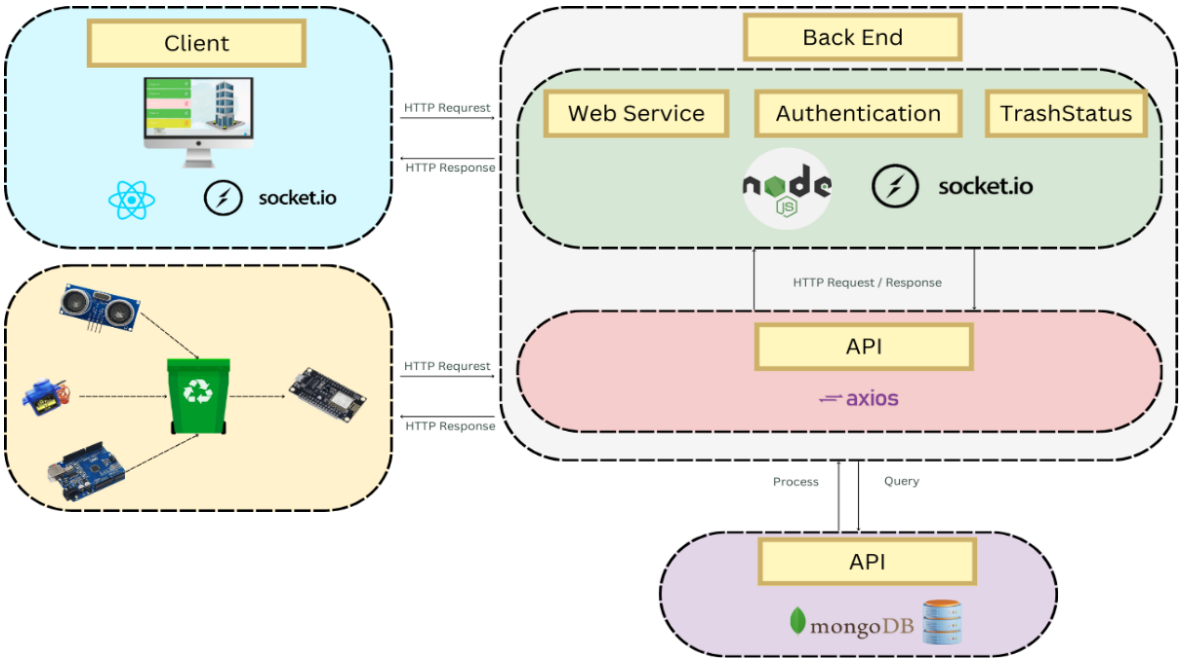

Figure 1: System architecture

The system's operating process includes 4 main parts: Smart Bin, website client (UI), backend, and MongoDB Database.

- Smart Bin: We use Arduino R3 and ESP8288 as the main modules with Arduino R3 used to physically operate the trash can, and ESP8266 with the function of using the internet and connecting to the backend via HTTP. request to update the status of the trash.
- Backend: Backend is used to receive and process requests from the trash and website clients, we use the Nodejs framework to deploy API construction, and socket.io is used to notify and receive information in real-time. Real-time and Axios for clients to call.
- Database: We use MongoDB to store trash status information.
- Website (UI): Used to display trash information for users to see and interact with, the website is built with Reactjs and uses socket.io for real-time execution, the trash can is displayed in 3 levels: there's more space, there's more space, it's full.

Here is the implementation process of the system

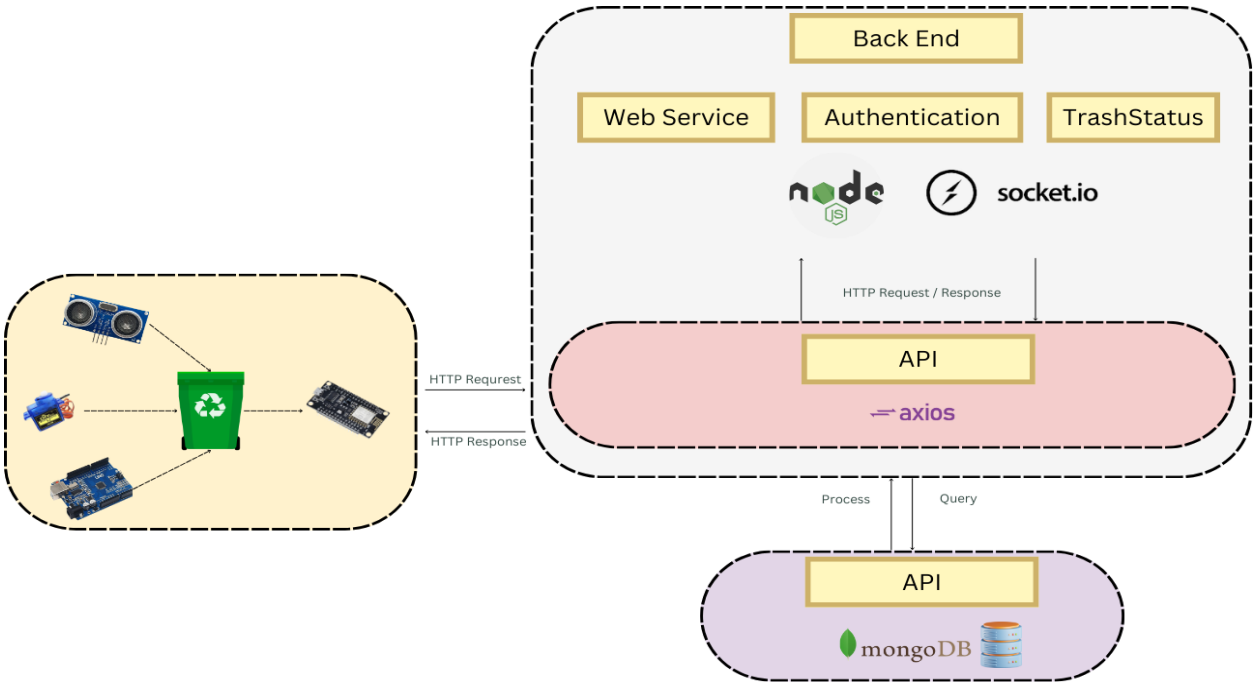

Figure 2 System architecture of trash, backend and database

The trash first will check the status of the trash, every 2 seconds the trash will update the status. ESP8266 will send the status of the trash through the PUT method to update the trash to the backend (API), and After receiving the backend, it will perform two main tasks: process the PUT request and make a call to the database to update the status. At the same time, a socket signal will be sent to the website client to request to update the information. , helps the website client update the latest status data of the trash.

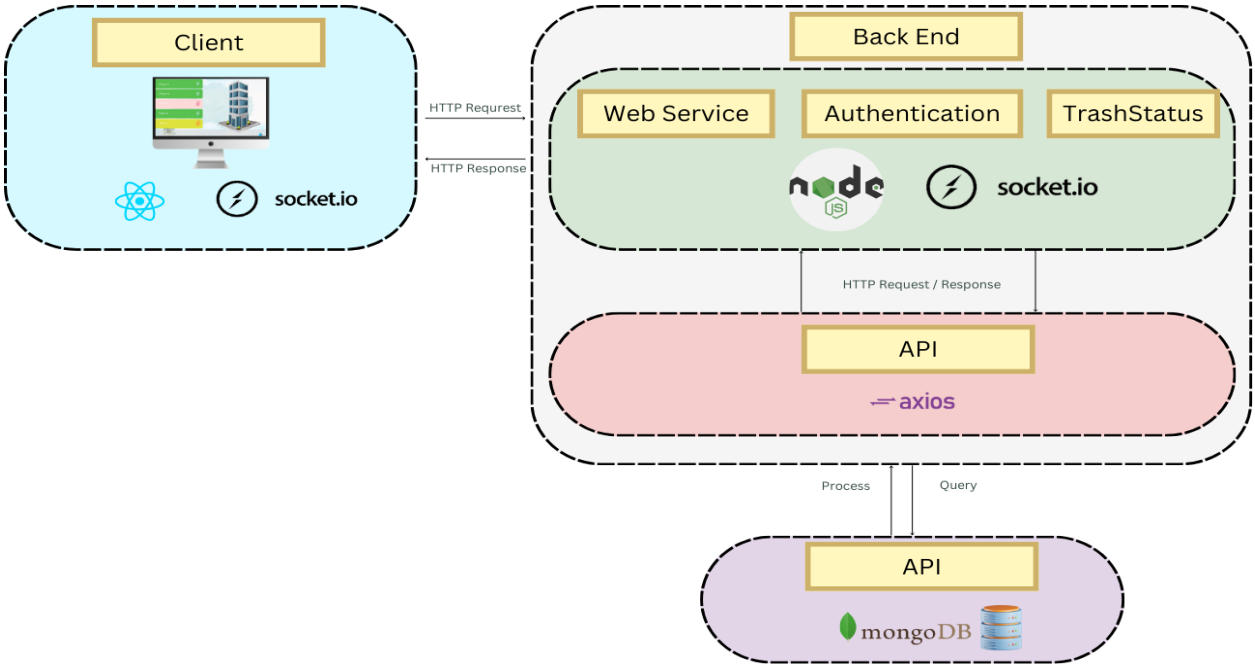

Figure 3: System architecture flow of website client, backend and database

The process of the client website, when receiving the signal of Socket.io requesting to update information, will make API calls through the GET method of information here the client will use axios to call the backend and the backend will call the database to return data to the client update data

2. Website Client (UI)

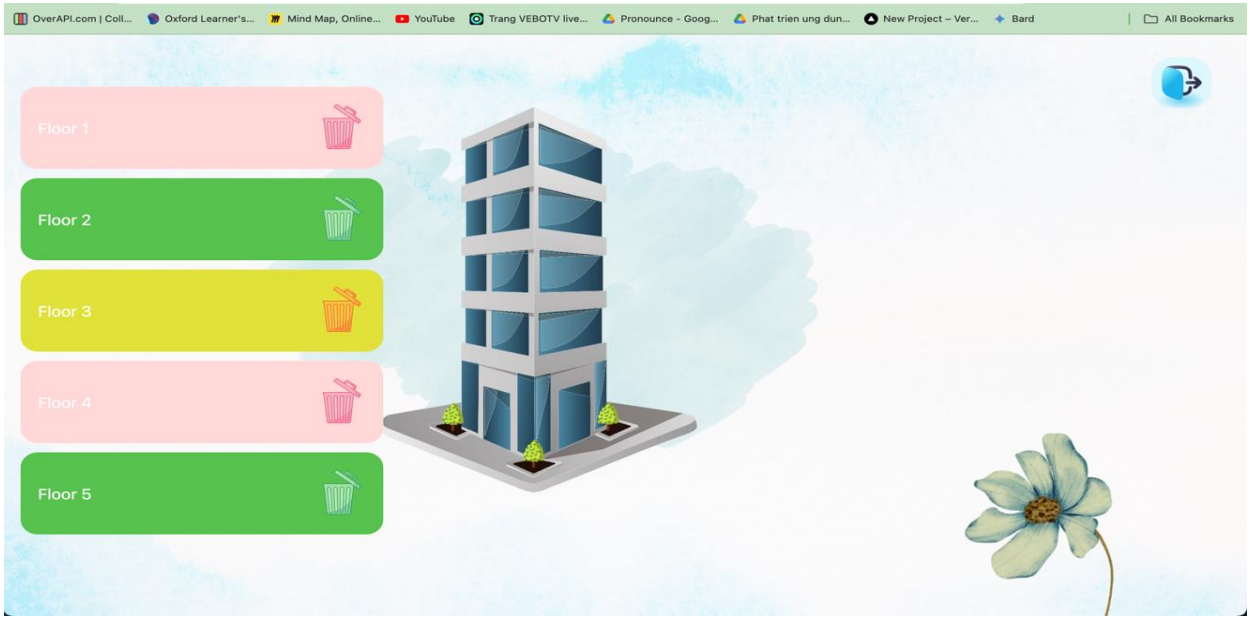

***Figure 2**: Client UI*

III. EXECUTION

1. Hardware Setup

a. Connect Arduino Uno R3:

- Connect the Arduino Uno R3 board to your computer using a USB cable.
- Open the Arduino IDE on your computer.

b. Connect SRF04 Ultrasonic Distance Measuring Sensor:

- Connect the VCC pin of the SRF04 sensor to the 5V pin on the Arduino.
- Connect the GND pin of the SRF04 sensor to the GND pin on the Arduino.
- Connect the Trig pin of the SRF04 sensor to digital pin 2 on the Arduino.
- Connect the Echo pin of the SRF04 sensor to digital pin 3 on the Arduino.

c. Connect Servo SG90:

- Connect the VCC pin of the Servo SG90 to the 5V pin on the Arduino.
- Connect the GND pin of the Servo SG90 to the GND pin on the Arduino.

– Connect the signal pin (usually white or yellow) of the Servo SG90 to digital pin 9 on the Arduino.

d. Connect NODEMCU-ESP8266 Wifi Module:

– Connect the NODEMCU-ESP8266 module to your computer using a USB cable.

– Ensure that the appropriate drivers are installed for the NODEMCU-ESP8266 module on your computer.

2. **Software Setup**

a. Download Required Libraries:

– Open the Arduino IDE.

– Install the required libraries for the SRF04 sensor and Servo SG90 if not already installed. You can find these libraries in the Arduino Library Manager.

b. Upload Arduino Sketch:

– Open the Arduino sketch file provided for the project in the Arduino IDE.

– Verify and upload the sketch to the Arduino Uno R3 board.

c. Setup WiFi Connection:

– Open the Arduino sketch file for WiFi communication using NODEMCU-ESP8266.

– Enter your WiFi SSID and password in the appropriate fields.

– Verify and upload the sketch to the NODEMCU-ESP8266 module.

3. **Testing Procedure**

a. Power On the System:

– Connect the Arduino Uno R3 board and NODEMCU-ESP8266 module to power sources

b. Test Proximity Sensor:

– Place an object within the proximity of the SRF04 sensor.

– Ensure that the servo motor rotates to open the lid of the trash bin automatically.

c. Test Overflow Detection:

– Fill the trash bin to the designated level.

– Verify that the system sends a notification to the designated endpoint indicating that the trash bin is full.

d. Verify WiFi Connection:

– Check the WiFi status LED on the NODEMCU-ESP8266 module to ensure that it is connected to the WiFi network.

e. Remote Monitoring:

– Access the designated endpoint web server to monitor the status of the trash bin remotely.

– Ensure that the status updates are reflected accurately on the monitoring interface.

4. **Troubleshooting**

a. Check Connections:

– Ensure all connections between hardware components are secure and correctly wired.

– Check for loose connections or faulty wiring.

b. Verify Power Supply:

– Ensure that all components receive an adequate power supply.

– Check for any power-related issues such as voltage drops or fluctuations.

c. Debugging Code:

– Use serial debugging in the Arduino IDE to monitor sensor readings and debug any issues in the code.

– Check for errors or inconsistencies in the Arduino sketch.

d. Check Network Configuration:

– Verify that the WiFi network credentials entered in the Arduino sketch are correct.

– Ensure that the NODEMCU-ESP8266 module is within range of the WiFi network.

VI. REFERENCES

[1] Random Nerd Tutorials. (n.d.). ESP8266 NodeMCU HTTP GET and HTTP POST with Arduino IDE (JSON, URL Encoded, Text). Retrieved from

[2] How To Mechatronics. (n.d.). Ultrasonic Sensor HC-SR04 and Arduino – Complete Guide.

[3] Config Mongodb for storage database of Bin.