Group 9
Karla Chuprinski
Montserrat Alonso

Assignment 8 - Building an End-To-End IoT System

Approach and Summary of System Architecture

To process and analyze user requests, our system combines a MongoDB database, a TCP client-server paradigm, and data from IoT sensors. The server fetches the required data from MongoDB after receiving appropriate requests from the client. The client receives the results of the server's calculations and data processing. To guarantee that device-specific queries were appropriately handled, Dataniz's metadata for IoT devices was included. Among the essential architectural elements are:

- TCP Client: Communicates with the server, shows results to the user, and accepts specified queries.
- TCP Server: Manages computations, communicates with MongoDB, and processes queries.
- Sensor readings and metadata from IoT devices are stored in the MongoDB database.

Research Findings on IoT Sensors and Data

We have three devices, data types, unit types, and one timezone type:

Devices	**Data**	**Units**	**Time Zones**
Kitchen Fridge	Moisture readings	Relative Humidity (RH%)	PST
Smart Dishwasher	Water consumption per cycle	Gallons	PST
Second Fridge	Electricity consumption	kWh (kilowatt-hours)	PST

The IoT system incorporates 4 types of sensors: 1. Moisture sensor - measures the relative humidity inside the kitchen fridge. It monitors the fridges' environmental conditions and provides data in RH%. The data is recorded in regular intervals. Here we computed the average humidity over a three-hour period. 2. Water consumption sensor - measures the amount of water used per cycle. Data is collected at the end of each cycle, with timestamps that indicate when the cycle happened. 3. Electricity meter - measures the power consumption of the kitchen fridge, second fridge, and smart dishwasher. We record the kWh values of all three in order to see the usage patterns and conclude which device had the highest electricity consumption. 4. Temperature sensor- measures the temperatures of the fridges. Although this sensor is not utilized in one of the queries it is in our IoT system to represent an actual Fridge and records in Celsius.

Metadata used in Dataniz

Dataniz contains metadata about every IoT device, such as its distinct assetUid, data type, and unit of measurement. This information was essential for connecting database queries to certain devices, ensuring accurate association of units and data types. No explicit unit conversions were required as the metadata matched the units specified in the assignment requirements. Using parentAssetUid to connect sensors to their corresponding boards and devices, the metadata's hierarchical structure allowed for a clear arrangement of devices, boards, and sensors. For gadgets like the Smart Dishwasher, which has an ammeter and a water consumption sensor, this proved crucial. Validating sensor data and making sure values stayed within operational ranges—such as the water consumption sensor's range of 0 to 50 gallons—required the use of fields like minValue, maxValue, and desiredMinValue. Furthermore, elements like generationDate and unit guaranteed consistency in data processing; generationDate secured the use of the most recent metadata.

Details on Algorithms and Calculations

A binary tree efficiently manages the refrigerator's moisture data, enabling sorted storage and retrieval to compute the average relative humidity (RH%) using in-order traversal. Dishwasher water usage was grouped into cycles by identifying gaps greater than an hour between timestamps, with averages calculated per cycle. The device with the highest electricity consumption was identified by adding the usage for each device and finding the rightmost node in the binary tree, which held the highest total consumption. Units (%RH, kWh, gallons) matched Dataniz metadata, requiring no conversions. The binary tree's efficiency ensured fast, accurate data processing and retrieval for all queries.

Challenges Faced

The main challenge in this assignment was learning how to effectively use Dataniz and editing sensors to meet the requirements. Ultimately, we decided to erase everything and start from scratch for better alignment with the project goals. Another difficulty was researching sensor specifications, particularly when determining the minimum and maximum values required to fill out the sensor forms.Additionally, creating the queries was challenging, as we wanted to ensure the calculations were accurate while efficiently utilizing the binary search tree for sorting and fast data retrieval.

Dataniz Feedback

As mentioned earlier, Dataniz lacks features that could significantly improve user efficiency. Adding the ability to edit sensor configurations would save time and enhance usability. Additionally, providing a tutorial or onboarding guide for first-time users would make the website more user-friendly.