Smart Aquarium

Software Engineering for
Intern of Things (2023 – 24)

Aarohi Garg | Aicha Moussaid | Mudita Shakya

University of L'Aquila, Italy
Professor Davide Di Ruscio
18 January, 2024

Table of Contents

Introduction

The IoT-Based Smart Aquarium System is an advanced solution designed to monitor and maintain optimal conditions within an aquarium. Utilizing a suite of sensors, this system will continuously monitor various environmental factors, such as temperature, pH levels, diffused oxygen, dissolved ammonia-nitrogen, and water level. The primary goal is to ensure a healthy habitat for aquatic life by providing real-time data monitoring and customised thresholds according to the species of fish present in the aquarium.

Functional Requirements

ID	Requirement Name	Description
1.	Sensor Integration	The system must be able to incorporate the use of multiple precise sensors for the measurement of temperature, pH, diffused oxygen, dissolved ammonia, and water level.
2.	Data Gathering	The system should be able to support continuous in-flow of data collected from the sensors.
3.	Efficient storage	The system should be able to store data for historical data analysis and trend prediction.
4.	Remote monitoring	The system should allow monitoring the collected data through a web or mobile-based application.
5.	Data Visualization	The system should be able to present the stored data through easily interpretable charts.

Non-functional Requirements

ID	Requirement Name	Description
1.	Reliability	The system must be able to function without a significant downtime.
2.	Scalability	The system should support easy integration of additional sensors or functionalities.
3.	Usability	The system should be user-friendly and provide intuitive visualization and interaction for users of varying technical expertise.
4.	Security	The system should support secure communication channels to prevent unauthorized access and data breaches.
5.	Maintainability	The system should be easily maintainable and should be flexible to account for updates and enhancements.
6.	Availability	The system should be available for allowing monitoring and alerting.

Technologies Used

The following technologies were used for the development of the project:

1. ***Python***: Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It comes with a large standard library and has a vast ecosystem of third-party packages, making it suitable for a wide range of tasks.

2. ***Docker***: It is an open-source platform used for developing, shipping, and running applications. It uses containerization technology to enable developers to package applications with all their dependencies into a standardized unit called a container. This containerization ensures that the application works seamlessly in any environment, be it development, testing, or production.

3. ***Node-Red***: It is a programming tool that provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette, which can be deployed to its runtime in a single click.

4. ***Mosquitto***: It is an open-source message broker that implements the MQTT. It facilitates the communication between IoT devices and applications by enabling them to publish and subscribe to messages over a network. MQTT (Message Queuing Telemetry Transport) protocol is a lightweight and efficient messaging protocol designed for minimal network bandwidth and device resource requirements.

5. ***InfluxDB***: It is an open-source time series database designed to handle high write and query loads. It is particularly well-suited for storing and analyzing time-stamped data generated by sensors, applications, or infrastructure that change over time, such as IoT device data and application metrics.

6. ***Grafana***: It is an open-source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources, including time-series databases like InfluxDB, Prometheus, and Graphite.

System Architecture

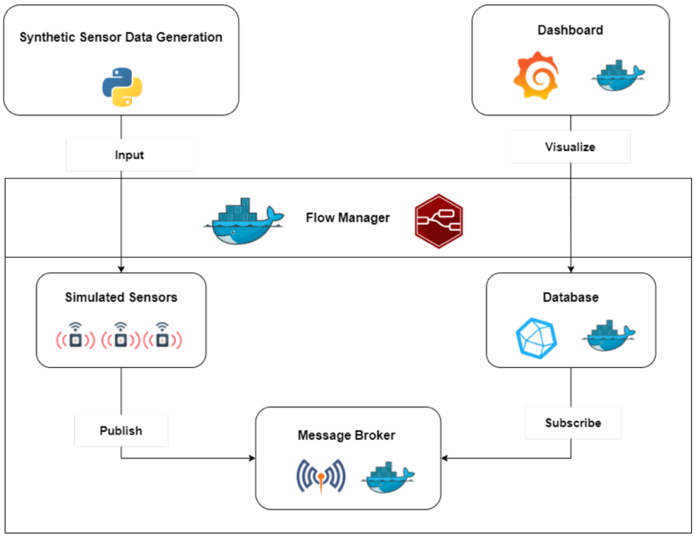


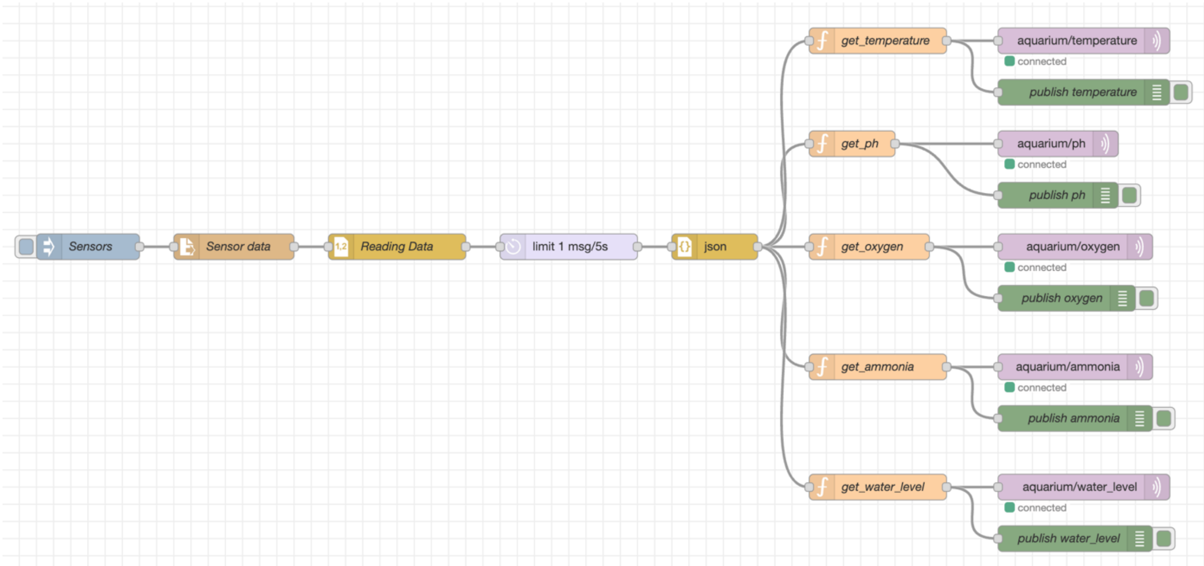
Figure 1 - System components with the technologies used

System Functionality

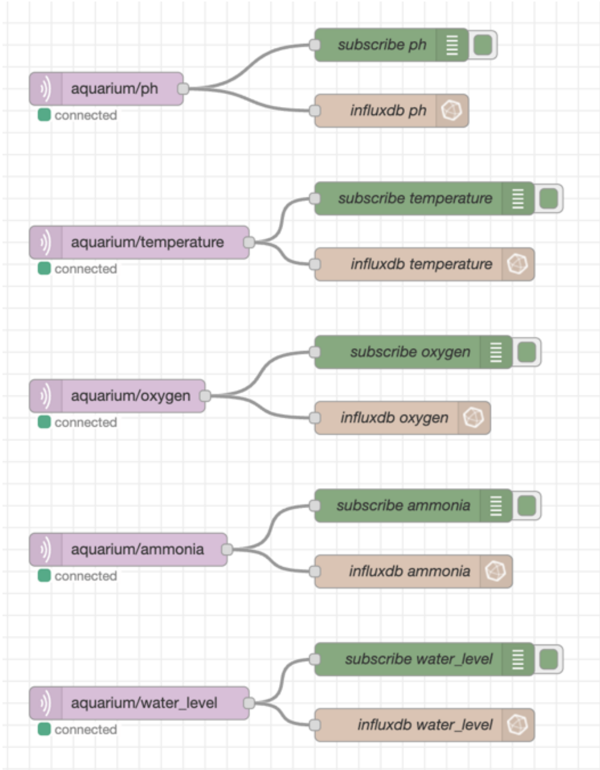
1. ***Data Generation***: The sensor readings are synthetically generated by using Python code and stored in .csv file.

2. ***Data processing***: Since the data is simulated, it is already pre-processed using Python.

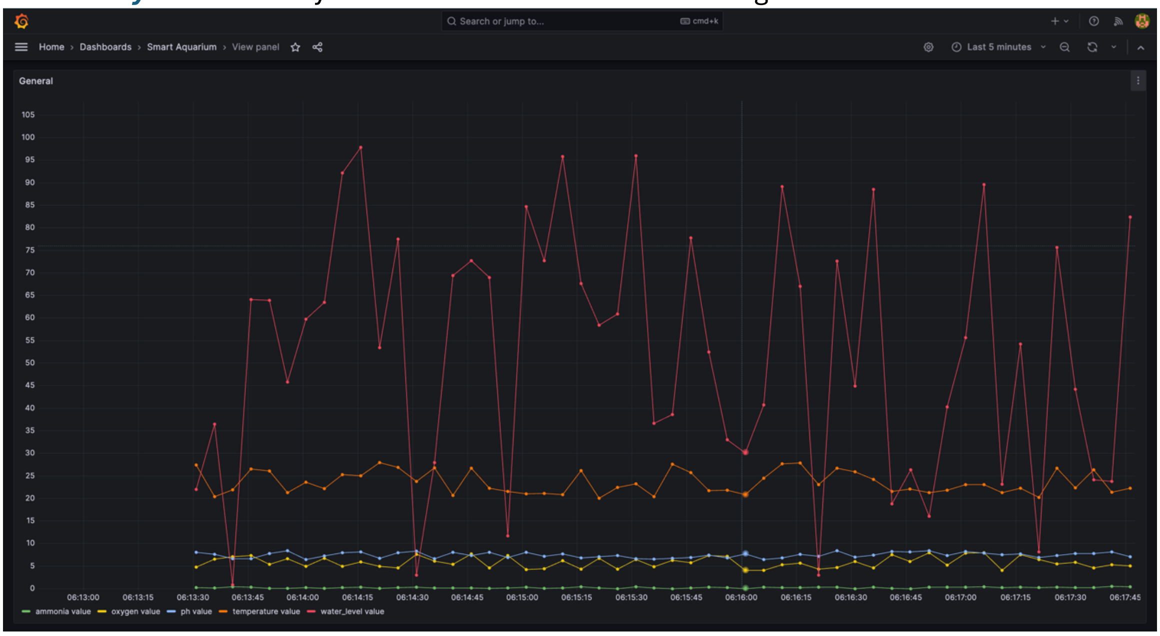
3. ***Data communication***: The entire data flow takes places inside Node-red. Firstly, the data is read from the synthetic data file (.csv) and published to MQTT broker – mosquito for further transferring.



4. ***Data Storage***: Our choice of database is influxdb which subscribes to the Mosquitto broker and stores the earlier published data.



5. ***Data Analytics***: We analyze the data stored in InfluxDb using Grafana dashboards.



6. ***Software Deployment***: Detailed representation of the sensor data is shown on the Grafana Dashboard. There are dashboard rows which give the thresholds for sensor values depending on specific fish type, hence giving customised thresholds according to the fish species.

Conclusion

In conclusion, the Smart Aquarium System represents an IoT-Based system in aquatic care technology. This project offers a comprehensive baseline solution for monitoring and maintaining the delicate ecosystem within an aquarium. This system not only ensures the well-being of the aquatic life by constantly monitoring critical parameters like temperature, pH, oxygen levels, and more, but it also simplifies the maintenance process for aquarium owners. The ability to set custom thresholds for different species of fish elevates the system's functionality, catering to the unique needs of each habitat. Moreover, real-time data monitoring allows for prompt responses to any environmental changes, preventing potential harm to the aquatic life. This project highlights the potential of IoT in transforming traditional practices into more efficient, sustainable, and user-friendly processes, paving the way for smarter and more responsive care in aquarium management.

Future Scope

Looking ahead, the future work on the IoT-Based Smart Aquarium System can focus on several enhancements to further elevate its capabilities. Integration with machine learning algorithms could be explored to predict to ensure more proactive care. Additionally, expanding the range of monitored parameters to include factors like lighting intensity and spectral composition could provide a more holistic approach to aquarium management. Finally, developing a user-friendly mobile application for remote monitoring and control would enhance accessibility and convenience for users.