

Sistemas de Computação Móvel e Ubíqua

José Costa, Rodrigo Albuquerque e Rafael Mira
62637, 70294 e 59243
Faculdade de Ciências e Tecnologia
Universidade NOVA de Lisboa

June 8, 2025

1 Introduction

The Smart Aquarium is an IoT-powered system designed to simplify the care and maintenance of multiple aquariums, such as those found in pet shops or collectors setups. By continuously monitoring environmental conditions and responding automatically to any undesirable changes, the system helps ensure a healthy and stable habitat for aquatic life with minimal manual effort.

It is composed of several sensors and actuators, complemented by a mobile application that enables users to remotely monitor and manage their aquariums, offering convenience and control.

2 General Overview

Each user has their own aquarium and a mobile application. A user can manage multiple aquariums.

The system supports multiple clients. An aquarium owner can share visibility of their aquarium with other users.

The system uses a Spring Boot backend running on a server that communicates with both the mobile application and the ESP32 microcontroller.

Each aquarium requires one microcontroller.

- TDS Sensor
- Temperature Sensor
- pH Sensor
- Ultrasonic Sensor
- LDR Sensor

The actuators are:

- OLED Display
- Water Pump
- Buzzer

Each microcontroller must be connected to at least one of the supported sensors and one of the actuators.

After the initial setup is complete, sensor readings are displayed on the OLED screen.

If any reading exceeds the user-defined thresholds, the buzzer will alert the user.

To refill the aquarium, the user can activate the water pump either when the TDS threshold is exceeded or manually through a button in the mobile application.

3 System Architecture

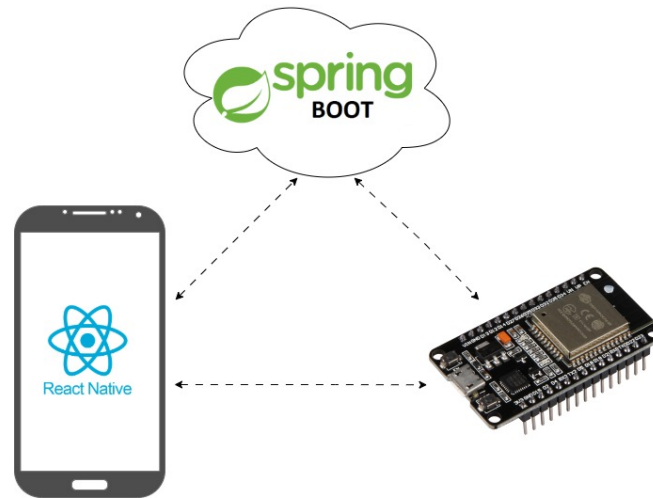


Figure 1: System Architecture

3.1 Hardware Layer

This layer is centered around microcontrollers (e.g., ESP32) connected to a set of environmental sensors and actuators. Each microcontroller gathers sensor data and communicates with the backend server to report readings and receive control commands. In our setup, two ESP32 devices are deployed in parallel, each managing its own set of sensors and actuators while communicating with the rest of the system over the network.

For initial setup, the system uses the Bluetooth Low Energy (BLE) protocol to pair with the user's smartphone and receive Wi-Fi credentials. BLE was chosen for its low power consumption and wide support on mobile devices. Once connected to the internet, the microcontrollers communicate with the backend server, registering their unique MAC address to an aquarium.

3.2 Backend Server

The backend layer is implemented using Spring Boot and serves as the central hub for communication between the microcontrollers and the mobile application. It exposes a set of secure RESTful APIs to receive sensor data from the hardware, process control commands from the mobile app, and coordinate system behavior. A MySQL database is used to store historical sensor readings and device states.

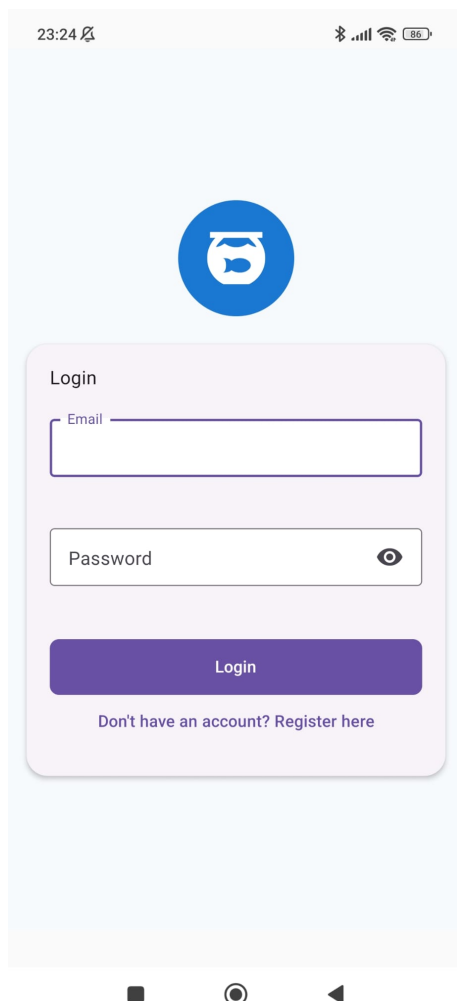
In the current setup, the backend server is deployed in a Docker container running on a local server. To enable remote access, we use a Cloudflare Tunnel, which securely exposes the local service to the internet through a public DNS address.

3.3 Mobile Application

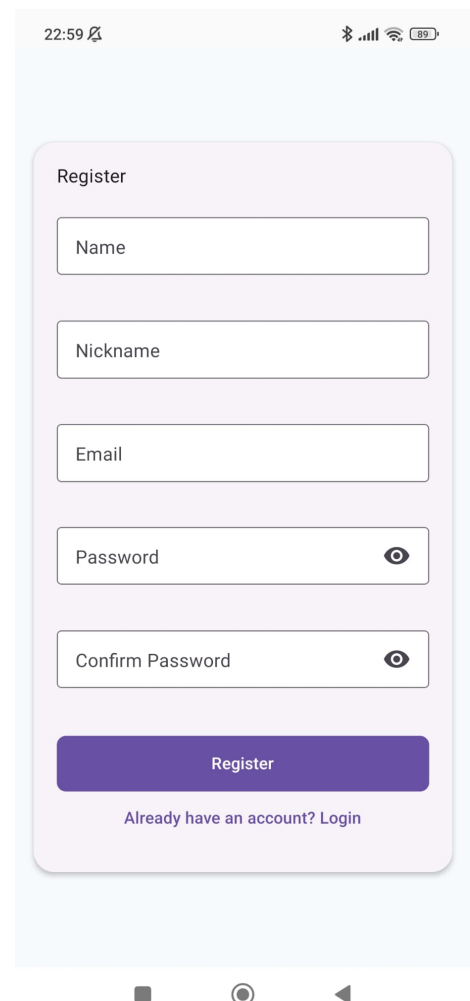
The mobile application, developed using React Native and TypeScript, serves as the user interface for the system. It allows users to monitor live sensor data, control devices, receive real-time alerts via push notifications, and configure system parameters. All interactions with the backend are performed securely over HTTPS, ensuring data integrity and privacy.

4 Mobile Application

4.1 Login and Register forms

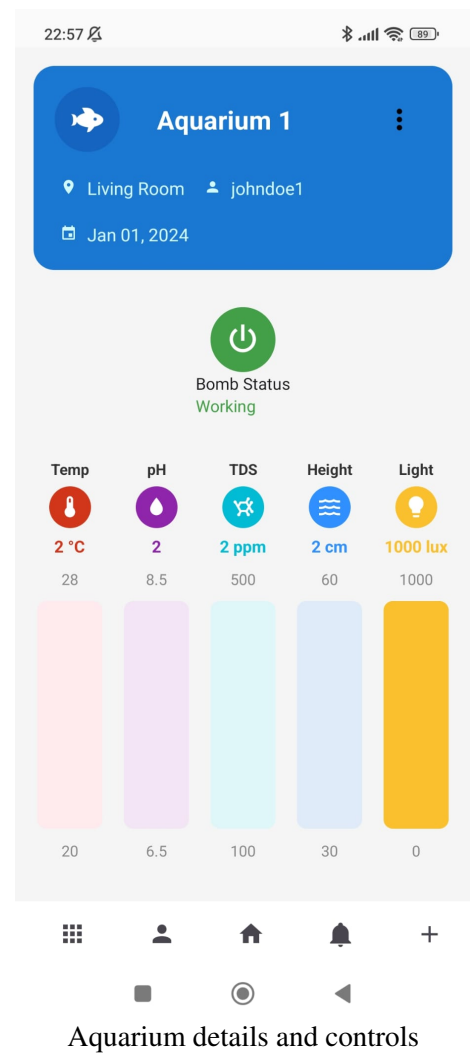
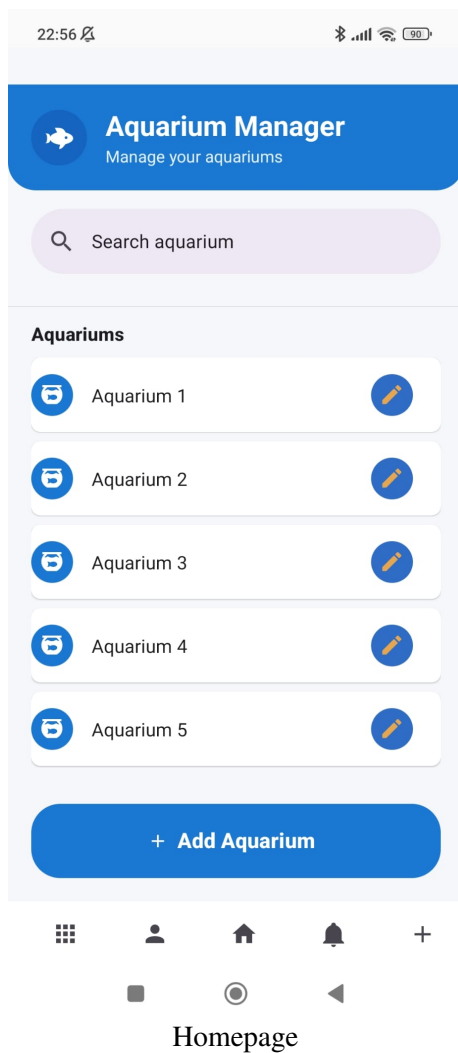
A mobile app login screen. At the top, there's a blue circular logo with a white stylized 'B'. Below it, a light purple rounded rectangle contains the 'Login' form. The form has an 'Email' input field, a 'Password' input field with an eye icon for toggling visibility, and a purple 'Login' button. At the bottom of the form, there's a link that says 'Don't have an account? Register here'. The phone's status bar at the top shows the time 23:24, signal strength, Wi-Fi, and 88% battery.

Login form

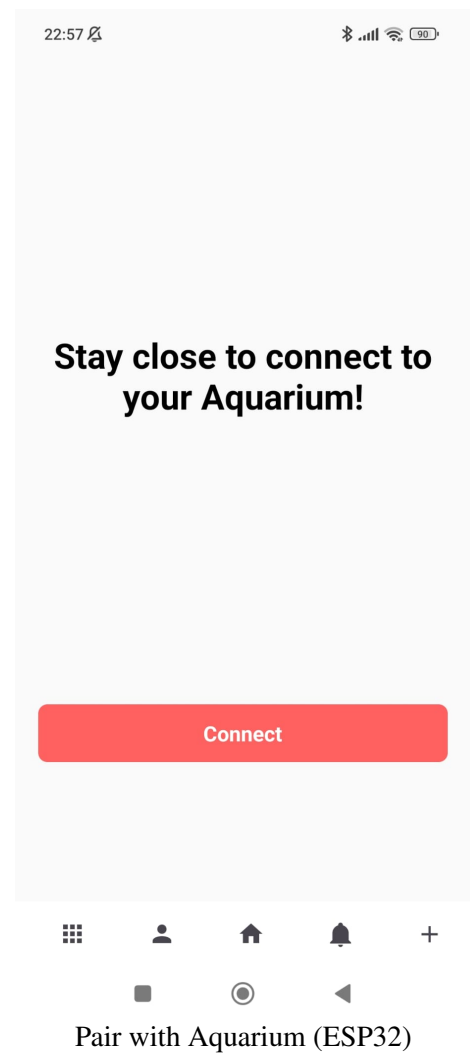
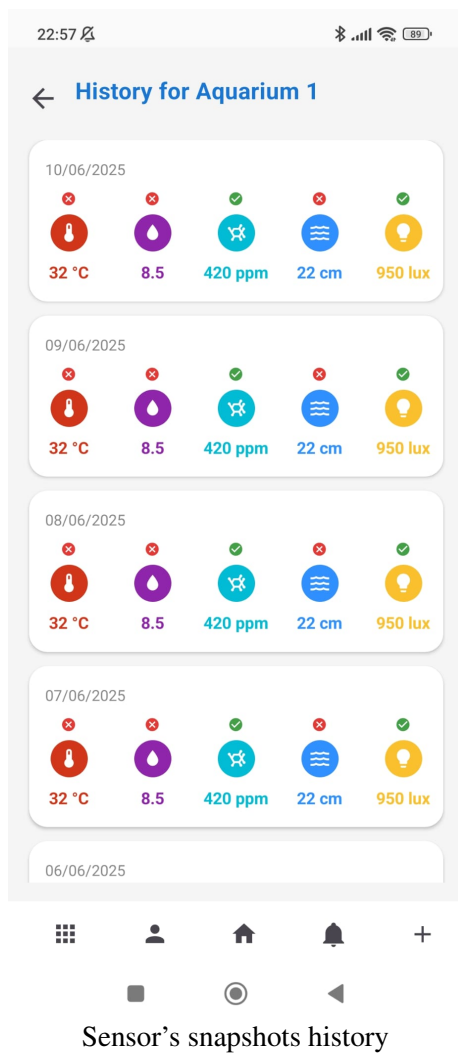
A mobile app registration screen. It features a light purple rounded rectangle with the 'Register' form. The form includes input fields for 'Name', 'Nickname', 'Email', 'Password' (with an eye icon), and 'Confirm Password' (with an eye icon). A purple 'Register' button is at the bottom of the form, with a link 'Already have an account? Login' below it. The phone's status bar at the top shows the time 22:59, signal strength, Wi-Fi, and 89% battery.

Register form

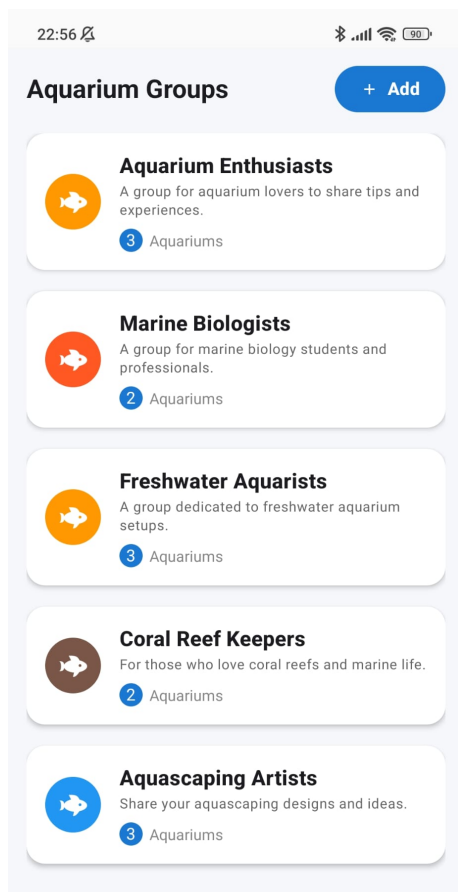
When entering the app for the first time, the user is greeted with login and registration pages, allowing them to either sign into an existing account or create a new one to start managing and monitoring their aquariums. Each aquarium is associated with an account, being possible for an aquarium's owner to share it with other users.



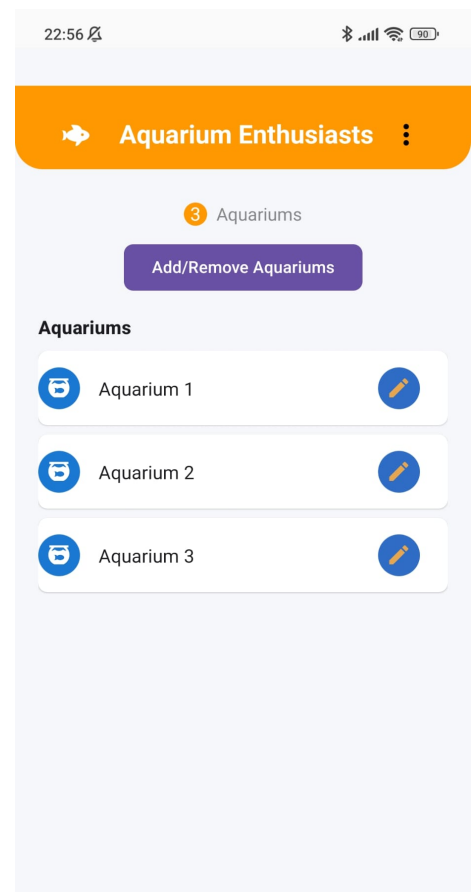
Once we log in, we go into the home page that contains the list of our own aquariums and those who we also have view access. We have a search bar that allow us to search for aquariums and we have a "Add Aquarium" button that triggers the flow to register with the esp32.



The system stores sensor snapshots, allowing users to view the historical data for each aquarium and track environmental changes over time. Additionally, the mobile app guides users through the pairing process with a new ESP32 device using Bluetooth, allowing for a simple initial setup.



Aquarium Groups



Add or remove Aquariums in a Group

To improve organization and efficiency, the app includes a Groups feature, which allows users to categorize aquariums into shared spaces. This can be useful for grouping aquariums from a specific pet shop or organizing them by fish type or environmental conditions.

23:46

Aquarium 4

Edit Thresholds

Min Temperature (°C) 20

Max Temperature (°C) 28

Min pH 6.5

Max pH 8.5

Min TDS (ppm) 100

Max TDS (ppm) 500

Min Height (cm) 30

Max Height (cm) 60

Grid User Home Notifications +

Edit Aquarium Thresholds

23:46

Aquarium 4

Bedroom johndoe Jan 01, 2024

Share Aquarium with

Username

Share Aquarium

Cancel

20 6.5 100 30 0

Grid User Home Notifications +

Share Aquarium

Each aquarium has configurable threshold values for parameters such as temperature, pH, TDS, light and water level. These thresholds determine when the system triggers warnings or automatic actions. Additionally, users can easily share aquariums with others by entering their username, allowing for collaborative monitoring and management.

5 Sensing and Reacting

The system continuously monitors the aquarium environment using various sensors. The TDS (Total Dissolved Solids) sensor measures the concentration of dissolved substances in the water, while the temperature sensor tracks the water temperature. The pH sensor monitors the acidity or alkalinity of the water, and the ultrasonic sensor measures the water level. The LDR (Light Dependent Resistor) sensor detects ambient light levels.

When any sensor reading exceeds predefined thresholds, the system reacts by activating the buzzer to alert the user. Additionally, if the TDS level is too high, the water pump will automatically turn on, to refill the aquarium and clear the water. Alternatively, the user can turn on or off the water pump, through a button in the mobile application.

5.1 Waiting screen



For better user experience using the aquarium, the OLED display shows a little clock while the user is completing the initial setup. This is done to avoid having a blank screen while the user is waiting for the BLE and Wi-Fi connection to be established.

This is done using two threads: one for the setup and another for the clock. The clock thread updates the OLED display every second.

5.2 Saving Wi-Fi credentials

The initial setup of the system requires the user to provide Wi-Fi credentials to connect the microcontroller to the internet. This is done using Bluetooth Low Energy (BLE) for secure and efficient communication with the mobile application.

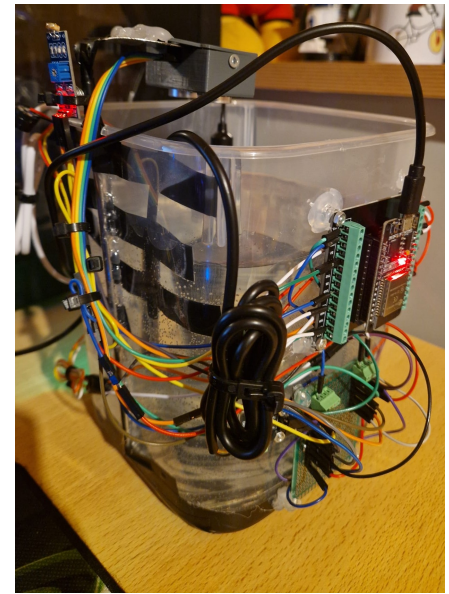
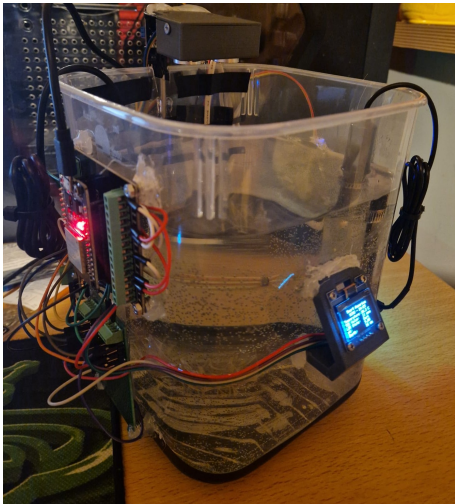
When the ESP32 receives the Wi-Fi credentials from the mobile app, it saves them in non-volatile memory (NVM) using the Preferences library to ensure they persist across reboots. This allows the microcontroller to automatically reconnect to the Wi-Fi network without requiring user intervention each time it starts up.

6 Experiments

During the development of the Smart Aquarium system, we conducted several experiments to validate the functionality and performance of the hardware and software components. These experiments focused on sensor accuracy, actuator responsiveness, and overall system reliability.

Each sensor was tested individually to ensure it provided accurate readings.

6.1 Final prototype



7 Conclusions and Lessons Learned

This project demonstrated the potential of using ESP32 microcontrollers to create a smart, compact and automated system. In a real-world deployment, key areas to improve would include better security, support for firmware updates over the air and explore GPS capabilities that could be used in the hardware.

One of the main lessons learned was the effectiveness of using BLE for initial configuration and the importance of reliable communication within services. We also gained practical experience in sensor integration, actuator control, and real-time data handling on resource-constrained devices. Overall, the ESP32 proved to be a powerful and flexible platform for IoT development.