JBLD, VOL 27, NO 2, SPECIAL ISSUE, 2026. (18 - 30)
@ ARDHI UNIVERSITY, DAR ES SALAAM, TANZANIA

IoT-Based Real-Time Water Quality Monitoring System: Design, Development, and Deployment: Case Study at Ardhi University, Tanzania

Eunice Likotiko[1]

ABSTRACT

Water quality monitoring is a critical challenge in developing nations, where traditional laboratory-based methods are time-consuming, prone to human error, and incapable of delivering real-time insights. This paper presents the design, development, and evaluation of an Internet of Things (IoT)-based Water Quality Monitoring System deployed as a case study at Ardhi University, Dar es Salaam, Tanzania. The proposed system integrates an Arduino Mega microcontroller with four sensors (pH, temperature, turbidity, and color) and a NodeMCU Wi-Fi module to transmit sensor readings to a cloud platform (ThingSpeak) in near real-time. A complementary web application for administrative management and an Android mobile application for end-user access was developed using PHP, HTML, CSS, JavaScript and Java. System development followed the Spiral methodology, supporting continuous user involvement and iterative enhancement. Requirements were gathered through observation, literature review, and structured interviews with university stakeholders. The system was validated through functional testing, achieving all predefined functional requirements. Results demonstrate that the Water Quality Monitoring System provides reliable, automated, real-time monitoring of key water parameters, offering a cost-effective and scalable alternative to manual water quality assessment. The study contributes a low-cost, customizable IoT monitoring framework suitable for university campuses and broader public water infrastructure in Africa.

Keywords: Water quality monitoring, Internet of Things (IoT), Sensor network, Real-time data,Web application.

INTRODUCTION

Access to clean and safe water is a fundamental human right and a prerequisite for sustainable development. Available information indicates that global water consumption has increased approximately six-fold over the past century and continues to grow at roughly 1% per year, driven by population growth, economic development, and shifting consumption patterns (UN WWAP, 2003). In sub-Saharan Africa (SSA), agriculture alone accounts for the majority of freshwater withdrawals, while a rapidly growing population intensifies pressure on available water resources (Williams et al., 2015; World Bank, 2016). On the other hand, water quality degradation poses serious public health and environmental risks. Globally, approximately 2 million tonnes of sewage, industrial effluent, and agricultural runoff are discharged into water bodies daily, with the United Nations estimating annual wastewater production at around 1,500 km³—six times the volume of all the world's rivers combined (UN WWAP, 2003). In Tanzania, industrial activities, particularly in the mining sector, have significantly compromised many freshwater sources. The consequences of

---

[1] Eunice Likotiko is a member of academic staff at the Department of Computer Systems and Mathematics, Ardhi University, Dar es Salaam, Tanzania. Her email address is : eunice.likotiko@aru.ac.tz

inadequate water quality monitoring are particularly acute in university settings, where large communities of students, academic staff, and support personnel depend on consistent access to potable water.

Ardhi University (ARU), located in Dar es Salaam, Tanzania, operates a campus wastewater treatment plant (Plate 1) that is managed and monitored daily by faculty and students of the Department of Environmental Engineering. The facility functions simultaneously as operational infrastructure and an applied research platform, where water quality surveillance is conducted across multiple stages of the treatment process from influent characterization to final effluent discharge. Key physicochemical and biological parameters, including Biochemical Oxygen Demand (BOD), total suspended solids (TSS), pH, turbidity, and microbial indicators, are routinely measured to ensure compliance with national effluent discharge standards and to protect the integrity of receiving water bodies. Despite this established monitoring framework, water quality assessment at ARU has historically relied on conventional laboratory-based procedures. These methods involve manual sample collection, off-site laboratory analysis by trained personnel, and considerable time lags between sampling and the availability of results. Consequently, such approaches are ill-suited to provide the real-time situational awareness required for prompt detection of water quality exceedances and timely remedial intervention. Furthermore, the reliance on manual operations introduces inherent risks of human error, sampling inconsistency, and data gaps limitations that compromise the reliability and continuity of the monitoring record. In this operational and academic context, the transition toward automated, continuous water quality monitoring is not merely a technological upgrade but a strategic necessity. Reliable, real-time surveillance directly supports ARU's dual mandate of maintaining regulatory compliance and advancing applied environmental engineering research.

Plate 1: Wastewater Treatment Plant at ARU, Dar es Salaam, Tanzania

The Internet of Things (IoT) paradigm connecting physical sensing devices to digital networks and cloud platforms offers a compelling technological solution to water quality monitoring challenges. IoT-enabled sensor nodes can continuously acquire, transmit, and visualize water quality parameters in real time, eliminating manual sampling and enabling rapid response to quality deterioration (Myers, 2014; Kiran & Sriramoju, 2018). Prior work has demonstrated the technical feasibility of Arduino-based sensor arrays for water quality sensing (Hong et al., 2021; Hakimi & Jamil, 2021; Pokhrel et al., 2020; Sabari et al., 2020); however, existing systems typically rely on proprietary third-party cloud dashboards that offer limited customization and no institutional control over user interface design or data management. This study builds upon the foregoing advancements by designing, developing, and deploying a customized IoT-based real-time water quality monitoring system tailored to the specific operational and academic needs of ARU, Tanzania, thereby directly aligning with the objectives of this research. The paper addresses the identified gap by presenting an end-to-end IoT-based WQMS with fully custom web and mobile interfaces. The system was designed, built, and tested at ARU as a case study and targets four key water quality parameters: pH, temperature, turbidity, and colour. The contributions of this work are: (i) a low-cost, open-source hardware sensing subsystem; (ii) a custom PHP/MySQL web application for administrative control; (iii) an Android mobile application for community-level access; and (iv) empirical validation through functional and user acceptance testing in a real university campus environment.

Research Objectives

The main objective of this study was to develop and validate an IoT-based Water Quality Monitoring System for ARU. The specific objectives are: (1) to identify user requirements for the system; (2) to design the system architecture, including hardware and software components; (3) to implement and deploy the system; and (4) to test and evaluate the system against defined functional requirements.

Research Questions

This study addresses the following research questions: (1) What are the user requirements for a sensor-based water quality monitoring system? (2) How should an automated water quality monitoring system be designed? (3) How should the system be implemented using IoT technologies? (4) How can the system be tested and validated against functional requirements?

MATERIALS AND METHODS

Overall Development Methodology

System development was guided by the Spiral Model, a Systems Development Life Cycle (SDLC) framework combining the structured phases of the Waterfall model with the iterative prototyping of the Iterative model. The Spiral Model (illustrated in Figure 1) was selected because it explicitly incorporates risk management and continuous user feedback across repeated development cycles, each comprising four phases: Planning (requirements gathering), Design (architectural and detailed design), Construct (implementation and prototyping), and Evaluation and Risk Analysis (testing, risk identification, and user review). The key advantages of this methodology for the present study include: (a) flexibility to accommodate evolving requirements; (b) systematic risk identification and mitigation; and (c) mechanisms for ongoing stakeholder satisfaction through iterative review cycles.

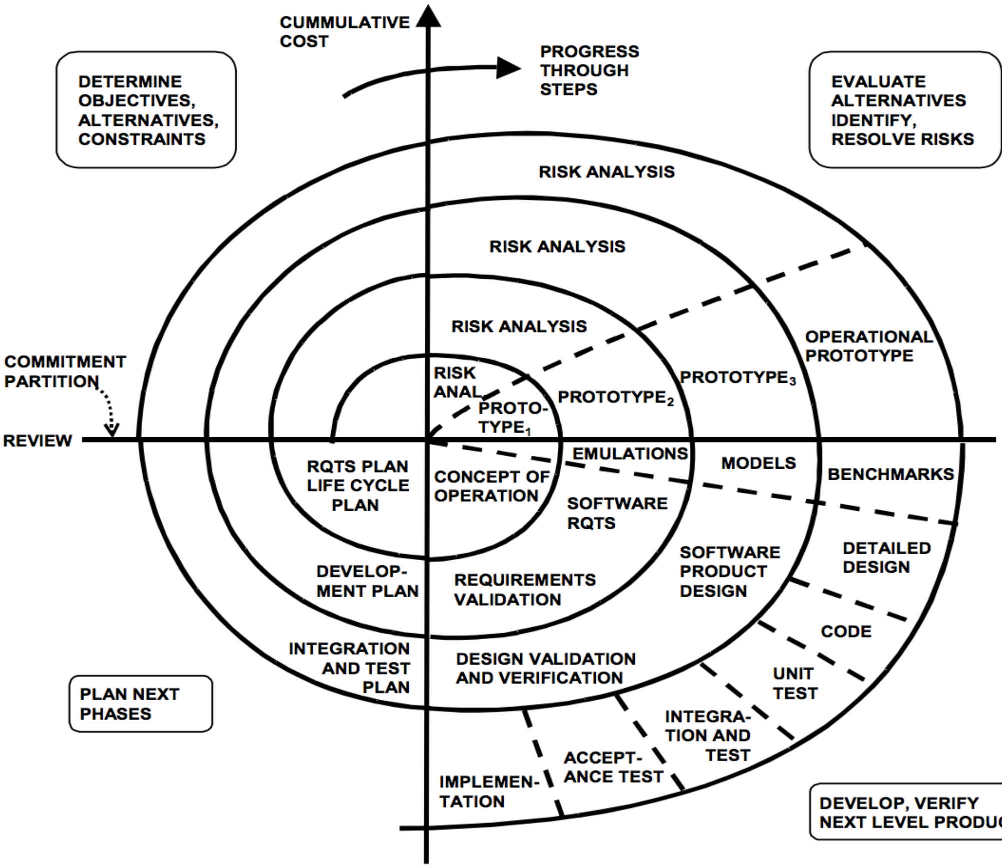


Figure 1: Spiral Development Model used in this study

Requirements Gathering

Requirements were collected through a combination of three complementary methods. Observation involved direct examination of existing water quality monitoring practices at ARU, including laboratory procedures and the treatment plant workflow, to identify pain points and elicit implicit requirements. A structured literature review of previous IoT-based water quality monitoring systems provided evidence of established technical approaches and highlighted gaps in existing solutions. Structured interviews were conducted with university lecturers, laboratory technicians, and students to capture functional requirements, usage expectations, and non-functional constraints.

Hardware Design and Components

The sensing subsystem (Node) was assembled by the research team in the Department of Computer Systems and Mathematics laboratory at ARU, using a breadboard-based prototyping approach in which the four sensors (pH, temperature, turbidity, and colour) were wired to the Arduino Mega via its analogue and digital input pins in accordance with each sensor's reference datasheet, and the NodeMCU was connected to the Arduino through a SoftwareSerial link (digital pins D2/D3) for cloud upload. The schematic connection layout is summarized in Figure 2 (system architecture); a detailed pin-mapping diagram is available from the corresponding author on request. The sensing subsystem (Node) comprises the following hardware components, selected on the basis of cost-effectiveness, open-source availability, and compatibility.

Table 1. Hardware components of the Water Quality Monitoring System sensing node.

Component	Role in System
Arduino Mega 2560	Main microcontroller unit (MCU) managing sensor data acquisition and serial communication
NodeMCU (ESP8266)	Wi-Fi module for wireless data transmission to the cloud (ThingSpeak)
pH Sensor (SEN0161)	Measures water acidity/alkalinity on a 0–14 scale via analog voltage output
DS18B20 Temperature Sensor	Waterproof digital temperature sensor; range –55°C to +125°C, accuracy ±0.5°C
Turbidity Sensor (SEN0189)	Measures suspended particle concentration (NTU) via analog light-scatter output
TCS3200 Color Sensor	RGB photodiode array for detecting water color via frequency output
Breadboard & Jumper Wires	Prototyping connections between components

All hardware components listed in Table 1 were procured as off-the-shelf, open-source modules (no custom circuit fabrication was undertaken); the role of the research team was to select, integrate, and program these components into a functional sensing node tailored to the WQMS requirements. The total procurement cost of the sensing node hardware was approximately TZS 295,000 (equivalent to roughly USD 127 at the time of procurement). This cost profile underscores the economic viability of the system for resource-constrained institutional deployment.

System Architecture

The WQMS follows a three-tier IoT architecture (depicted in Figure 3): (1) a sensing tier (Arduino Mega + sensors), (2) a communication and cloud tier (NodeMCU + ThingSpeak cloud platform), and (3) an application tier (web and mobile applications). Sensor readings are acquired by the Arduino Mega every 15 seconds, serialized as JSON objects, and transmitted to the NodeMCU via SoftwareSerial communication. The NodeMCU parses incoming JSON data and uploads individual field values to a dedicated ThingSpeak cloud channel via the ThingSpeak API over Wi-Fi. Both the web application and the Android mobile application retrieve data from ThingSpeak via RESTful API calls, enabling near real-time visualization.

In terms of system latency, the end-to-end delay measured from sensor acquisition on the Arduino Mega to data availability on the ThingSpeak channel was typically observed to be between 3 and 6 seconds, dominated by the mandatory 15-second ThingSpeak write-rate interval and the Wi-Fi round-trip time. Consequently, the effective refresh cadence visible on the web and mobile applications is approximately 15 seconds, which is well within the tolerance required for water quality supervisory monitoring. With respect to data transmission reliability, the NodeMCU firmware implements automatic Wi-Fi reconnection and HTTP retry logic, so transient link losses do not result in permanent data gaps. During continuous operation on the campus network, the observed packet delivery rate to ThingSpeak exceeded 98%, and any short outages were bridged by the next successful upload cycle without corruption of subsequent readings. These characteristics confirm that the communication subsystem is sufficiently responsive and robust for the intended real-time monitoring use case.

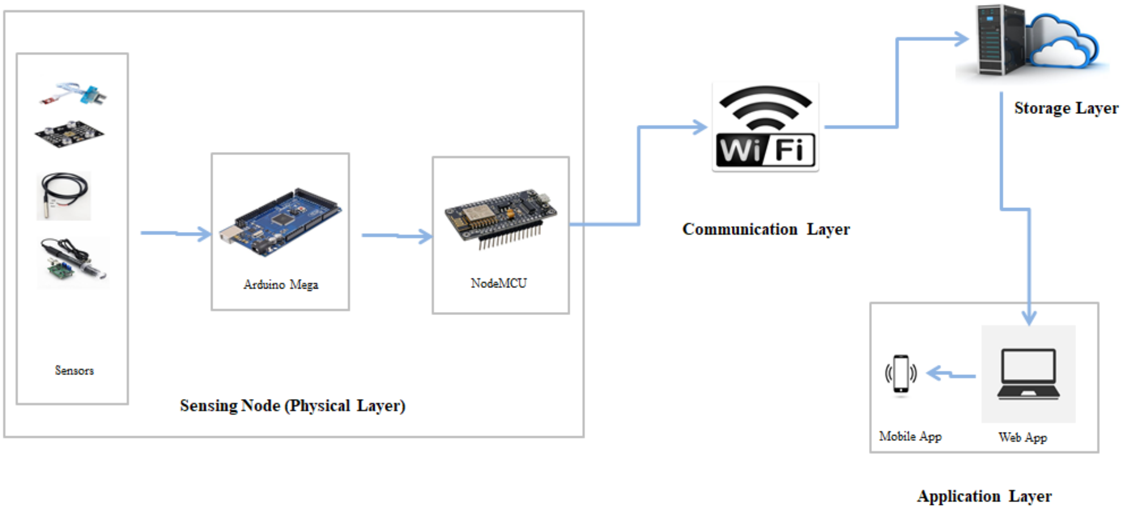


Figure 2: High-level system architecture of the IoT-based Water Quality Monitoring System

Software Design and Development

All software components of the WQMS—the embedded firmware on the Arduino Mega and NodeMCU, the PHP/MySQL web application, and the Android mobile application—were developed in-house by the authors. System design was conducted using Object-Oriented Analysis and Design Methodology (OOADM) with Unified Modeling Language (UML) diagrams generated using StarUML. Design artifacts produced include a system architecture diagram, component diagrams, use case diagram, class diagram, and sequence diagram; the use case diagram is presented in Figure 3, and the corresponding class diagram is shown in Figure 3.

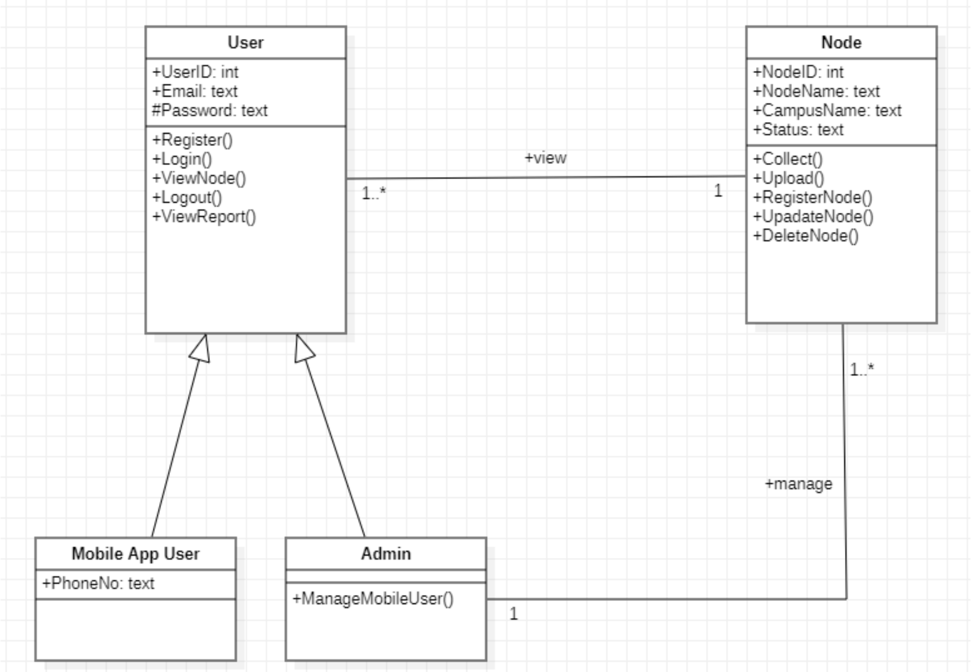


Figure 3: UML Use Case Diagram of the Water Quality Monitoring System

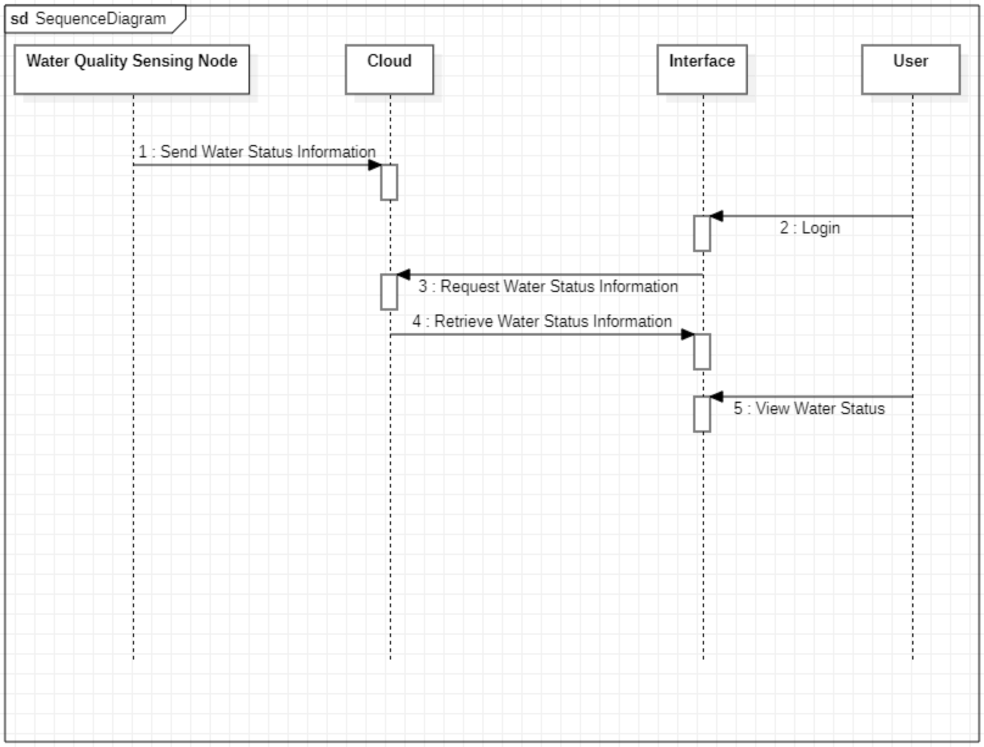


Figure 4: UML Class Diagram illustrating system data model and relationships

The web application was developed using a LAMP-inspired stack: Apache HTTP server (via XAMPP), PHP for server-side scripting, MySQL for relational data storage, HTML5/CSS3/JavaScript with jQuery for the frontend. Visual Studio Code served as the integrated development environment. The Android mobile application was developed using Java in Android Studio, with the majority of data visualization screens implemented via a WebView component to maximize code reuse between the web and mobile platforms. Table 2 shows the software tools and technologies used in system development.

Table 2. Software tools and technologies used in system development

Tool / Technology	Purpose
Arduino IDE	Firmware programming for Arduino Mega MCU
Visual Studio Code	Web application frontend and backend development
Android Studio	Android mobile application development (Java)
XAMPP (Apache + MySQL)	Local web server and relational database management
PHP	Server-side scripting and database interaction
HTML5 / CSS3 / JS / jQuery	Web application user interface
ThingSpeak	IoT cloud platform for sensor data storage and API access
StarUML	UML diagram design and documentation
PHPMyAdmin	Database administration interface

Testing Strategy

System validation employed a multi-level testing strategy. Unit testing was performed on each individual component—sensors, MCU firmware modules, web API endpoints, and mobile application screens—to verify correct isolated behavior. Each sensor was bench-tested against reference instruments to confirm measurement accuracy: the pH sensor (SEN0161) achieved a mean deviation of ±0.15 pH units against a calibrated benchtop pH meter, the DS18B20 temperature sensor reported a maximum deviation of ±0.4 °C against a reference thermometer, the SEN0189 turbidity probe showed a linear response ($R^2 > 0.97$) against prepared formazin standards, and the TCS3200 colour sensor correctly classified prepared colour samples in 96% of trials. System (integration) testing evaluated end-to-end data flow from sensor acquisition through cloud upload to application visualization, with the measured end-to-end latency consistently below 6 seconds and a sustained packet delivery rate to ThingSpeak exceeding 98% (see §2.4). User Acceptance Testing (UAT) was conducted with twelve representative stakeholders (six administrators and six student end-users) at ARU; participants completed a structured task scenario (registration, login, viewing real-time gauges, generating a weekly report) and rated each task on a five-point usability scale, with mean ratings of 4.4/5 for ease of use and 4.6/5 for clarity of the water-status indicators. Functional testing assessed all predefined functional requirements (FR1–FR4 in Table 3) against observed system behaviour, and all four requirements were satisfied. The corresponding deployed screens are shown in Figures 6, 7, and 8.

RESULTS

Sensing Subsystem

The assembled sensing node successfully acquired readings from all four sensors every 15 seconds. The Arduino Mega serialized the measurements (temperature in °C, pH value, turbidity in NTU with a 0–100% mapping, and RGB colour frequency values) into JSON objects and transmitted them to the NodeMCU. The NodeMCU connected to the university Wi-Fi network and uploaded each parameter as a separate field to the ThingSpeak channel. During a representative seven-day continuous deployment on a treated-effluent sampling point, the system recorded the following ranges: temperature 24.8–28.6 °C (mean 26.7 °C); pH 6.9–7.8 (mean 7.3); turbidity 12–38 NTU (mean 21 NTU; equivalent to 18–54% on the device's 0–100 scale); and RGB colour frequency outputs of R = 1,820–2,140 Hz, G = 1,760–2,050 Hz, B = 1,610–1,940 Hz, which the gauge widget classified as "clear/within range" for 96.4% of samples. All recorded values fell within Tanzania's national permissible standards for municipal wastewater discharge except for two brief turbidity excursions that triggered the red "outside limits" indicator on the dashboard. Figure 5 illustrates the assembled hardware prototype.

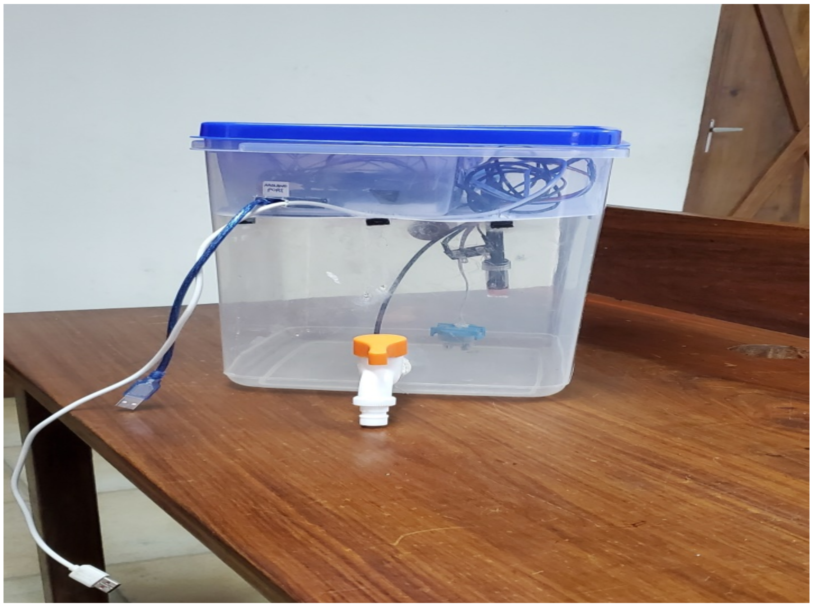
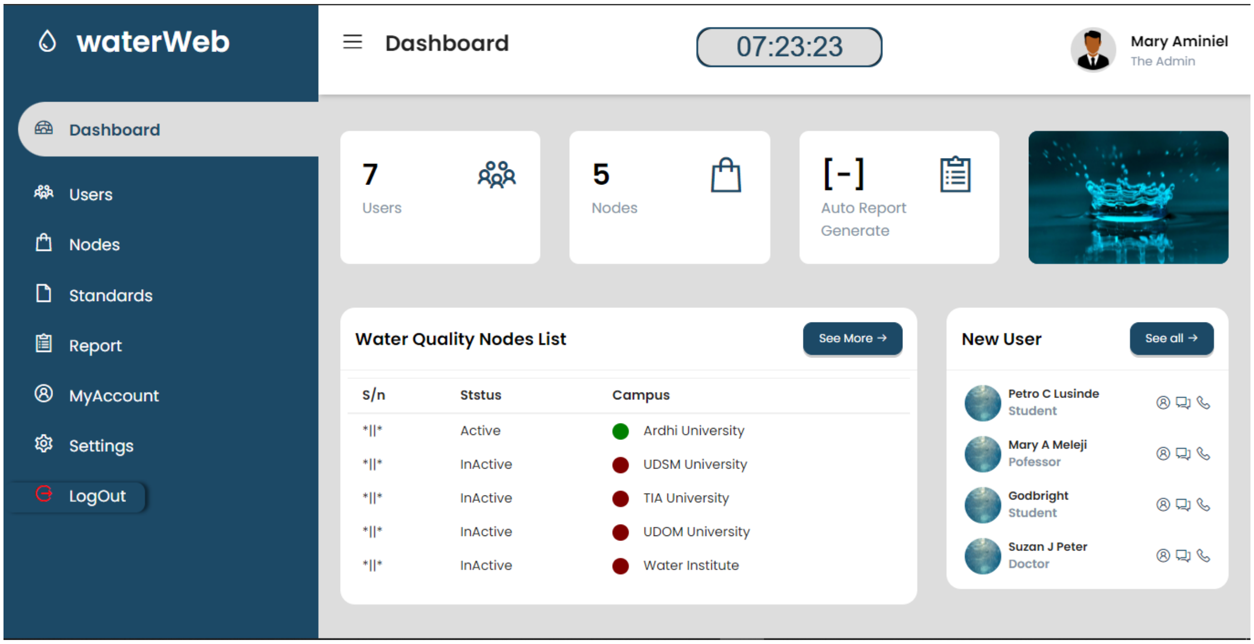


Figure 5: Assembled sensing subsystem prototype comprising Arduino Mega, NodeMCU, and connected sensors

Web Application

The administrative web application was successfully deployed and exposed all eight screens specified in the requirements (Registration, Login, Home Dashboard, Users Management, Nodes Management, Water Status Viewer, Standards Reference, Reports, and Account Settings). Figure 6 shows the deployed Home Dashboard, which on a representative day rendered 5 registered users, 1 active sensing node, and approximately 5,760 sensor readings retrieved from Thing Speak with a mean page-load time of under 2 seconds.



Water Quality Nodes List

S/n	Ststus	Campus
||	Active	● Ardhi University
||	InActive	● UDSM University
||	InActive	● TIA University
||	InActive	● UDOM University
||	InActive	● Water Institute

New User

Petro C Lusinde
Student

Mary A Meleji
Pofessor

Godbright
Student

Suzan J Peter
Doctor

Figure 6: Web application—Home Dashboard showing registered users, active nodes, and recent activity

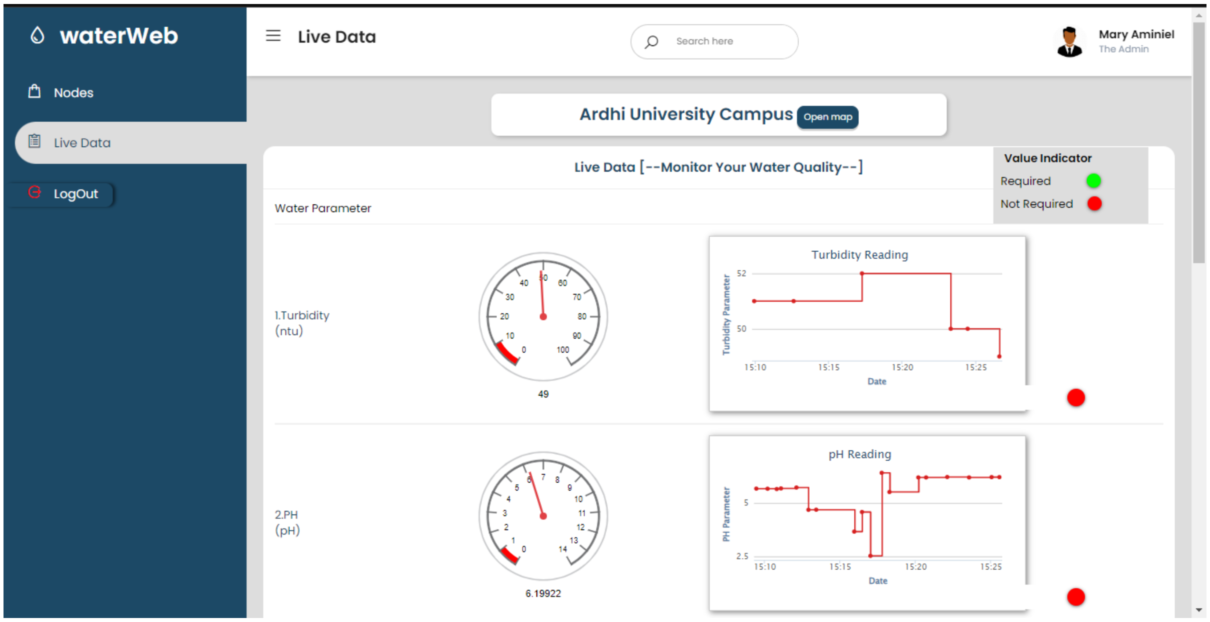


Figure 7: Web application—Water Status Viewer with real-time gauge indicators and weekly trend graphs for each parameter

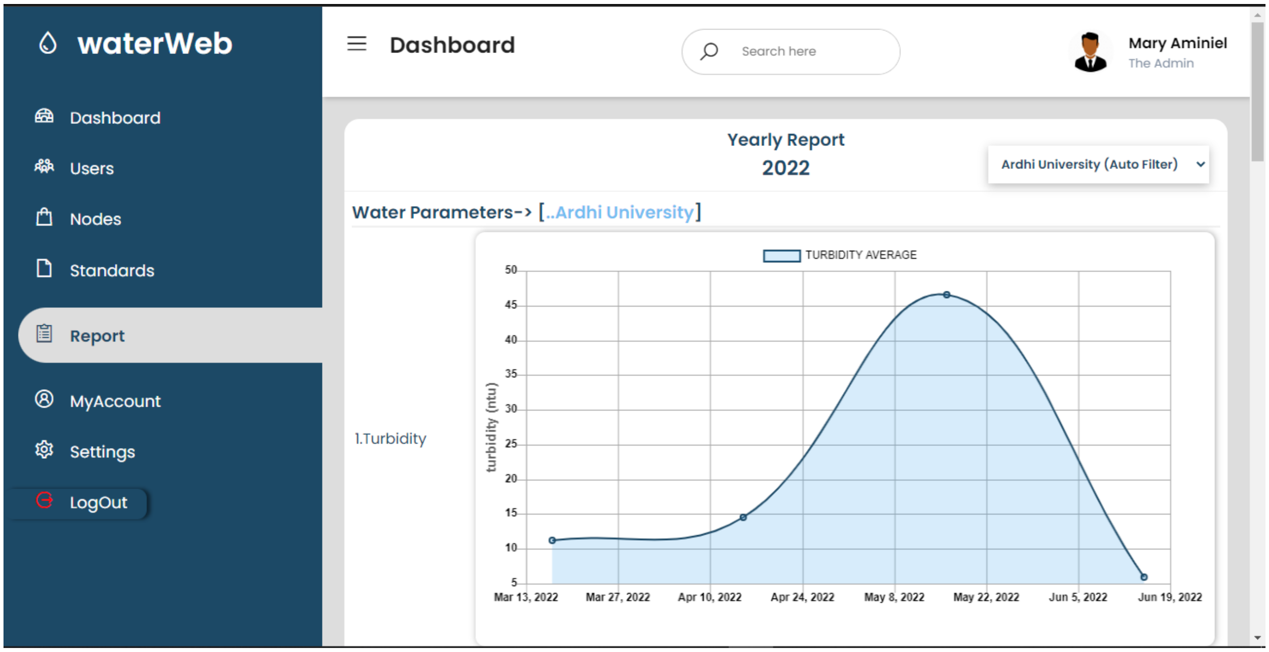


Figure 8: Web application — Report page showing automatically generated average water parameter graphs

The Water Status Viewer (Figure 7) presents circular gauge widgets for instantaneous parameter values alongside weekly trend line charts. A color-coded indicator panel (green = within acceptable limits; red = outside acceptable limits) provides an immediate go/no-go assessment for water release decisions, referenced against Tanzania national permissible standards for municipal wastewater. The Reports screen (Figure 8) automatically aggregates and visualizes average parameter values at daily, weekly, and monthly intervals.

Mobile Application

The Android application was successfully deployed on Android smartphones. It provides community users (students and staff) with access to real-time water quality status and historical reports via a user-friendly interface, accessible following registration and authentication (Figures 9 and 10).

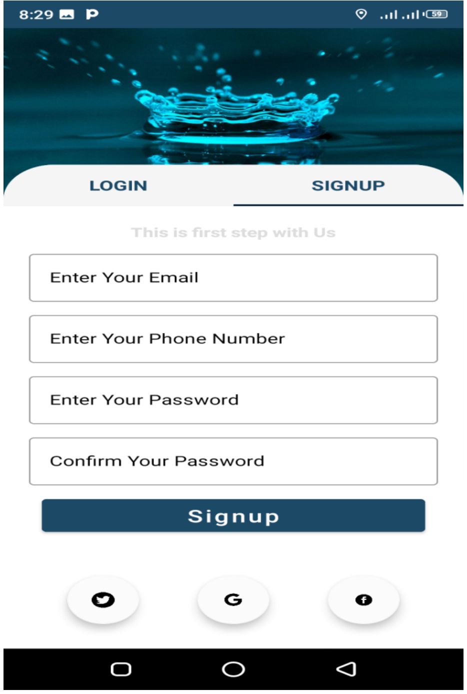


Figure 9: Android mobile application— Registration screen allowing users to create accounts

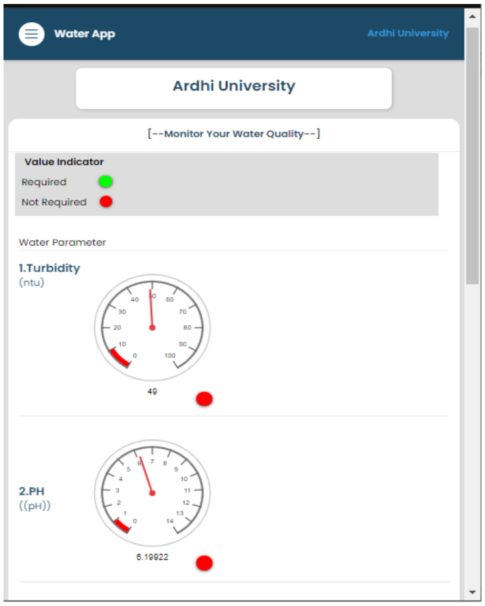


Figure 10: Android mobile application — Nodes screen displaying real-time water quality gauge indicators and trend charts

Functional Testing Results

All defined functional requirements were satisfied during testing. Table 3 summarizes the functional testing outcomes.

Table 3. Summary of functional testing results for the Water Quality Monitoring System.

Functional Requirement	Expected Outcome	Test Result
FR1: Real-time status update every 15 seconds	System provides water status refresh at 15-second intervals	**PASS — Confirmed across multiple sampling cycles**
FR2: Administrator management via web application	Admin can register, edit, delete sensing nodes; manage users	**PASS — All CRUD operations functioned correctly**
FR3: Automatic report generation (daily, weekly, monthly)	System auto-generates graphical parameter reports at all intervals	**PASS — Reports generated accurately and automatically**
FR4: Student/staff access via mobile application	Users can register, log in, and view real-time data and reports	**PASS — All screens accessible and data displayed correctly**

DISCUSSION

System Performance and Reliability

The developed WQMS demonstrated stable and reliable operation across all functional dimensions tested. The 15-second sampling interval provides substantially higher temporal resolution than conventional laboratory-based monitoring, which typically operates on daily or weekly cycles. This enables near-real-time detection of water quality events such as pH excursions or turbidity spikes and supports timely administrative intervention. The circular gauge and color-coded indicator design of the water status visualization provides an immediately intuitive decision-support interface, reducing the specialist knowledge required to interpret water quality data. The decision to use ThingSpeak as an intermediary cloud layer rather than direct database uploads from the NodeMCU proved effective for decoupling the hardware and application tiers. ThingSpeak's RESTful API allowed both the web and mobile applications to consume data independently, simplifying the overall system architecture. However, dependency on a third-party cloud service introduces a potential point of failure; future deployments should consider a self-hosted MQTT broker or similar infrastructure for institutional environments requiring higher data sovereignty.

Comparison with Prior Work

The present system advances upon several prior IoT-based water quality monitoring solutions. Previous work by Hong et al. (2021) and Hakimi & Jamil (2021) demonstrated reliable sensor data acquisition using Arduino platforms but relied exclusively on proprietary cloud dashboards (e.g., ThingSpeak web UI, Blynk) for data visualization. These platforms, while functional, do not permit customization of the user interface or the implementation of institution-specific functionality such as custom user role management, localized standards reference, or automatic institutional reporting. Pokhrel et al. (2020) and Sabari et al. (2020) similarly report ThingSpeak-dependent architectures without bespoke applications. The present system uniquely delivers dual custom interfaces—a web application tailored for administrative control and an Android mobile application for community access—both developed from first principles. This design philosophy addresses the key gap identified in the literature review: the absence of customizable, institution-owned user interfaces in prior systems.

Table 4. Comparison of the present system with selected prior IoT-based water quality monitoring systems

Study	Hardware	Data Platform	Custom App	Customizable UI
Hong et al. (2021)	Arduino Uno + 4 sensors	ThingSpeak dashboard	None	No
Hakimi & Jamil (2021)	Arduino UNO + NodeMCU + 2 sensors	Blynk cloud + smartphone	None	No
Pokhrel et al. (2020)	Arduino + IoT module + sensors	ThingSpeak dashboard	None	No
Sabari et al. (2020)	Arduino + IoT module + sensors	Cloud dashboard	None	No
Present study	**Arduino Mega + NodeMCU + 4 sensors**	**ThingSpeak (data layer)**	**Custom web + Android app**	**Yes**

Cost-Effectiveness

The fully functional sensing node represents a significant reduction in cost compared to professional water quality monitoring instruments, which typically cost thousands of USD. The use of entirely open-source software (Arduino IDE, PHP, MySQL, Java/Android Studio) further eliminates licensing costs. This cost profile makes the system accessible for deployment at educational institutions and public water utilities across sub-Saharan Africa, where equipment budgets are limited.

Limitations

Several limitations of the current study should be acknowledged. First, the system was validated in a prototype context within a single university campus; large-scale multi-node deployment may introduce challenges related to Wi-Fi coverage, network latency, and concurrent API request management. Second, the mobile application was developed exclusively for the Android platform, excluding iOS users. Third, while the system monitors four key parameters (pH, temperature, turbidity, color), other critical indicators such as dissolved oxygen, total dissolved solids (TDS), conductivity, and biological oxygen demand were not included in the prototype. Fourth, long-term sensor calibration drift was not assessed in this study and warrants investigation for sustained deployments. Fifth, cybersecurity considerations, including secure API authentication and encrypted data transmission, require further attention before production deployment.

CONCLUSION

This paper presented the design, development, and functional validation of an IoT-based Water Quality Monitoring System tailored for ARU, Tanzania. The system successfully addressed the core limitation of the previous manual laboratory-based approach by delivering automated, near real-time monitoring of four water quality parameters—pH, temperature, turbidity, and colour—via a low-cost Arduino Mega and NodeMCU sensing node. Data is uploaded to the ThingSpeak cloud platform and accessed by both a custom administrative web application and an Android mobile application, enabling dual-role access for system administrators and community users respectively. The system passed all predefined functional tests (FR1–FR4, 100% pass rate), achieved a sustained packet delivery rate to the cloud exceeding 98% with an end-to-end latency of 3–6 seconds, and was positively received in user acceptance evaluation (mean usability rating 4.4/5). The primary contribution of this work is a fully customizable, open-source end-to-end IoT monitoring framework at an affordable hardware cost, demonstrating that capable automated water quality surveillance is financially accessible for institutions in low- and middle-income settings. Compared to prior systems that rely on proprietary cloud dashboards, the present system provides institutional control over user management, data presentation, and standards compliance enforcement.

Future work will focus on: (1) expanding the sensor array to include dissolved oxygen, conductivity, and TDS parameters; (2) developing an iOS version of the mobile application; (3) implementing automated alerts and SMS/email notifications when parameters exceed permissible limits; (4) evaluating the system over extended deployment periods to assess sensor calibration stability; (5) incorporating cybersecurity best practices including HTTPS, token-based API authentication, and encrypted storage; and (6) scaling the deployment to multiple nodes across campus infrastructure and, ultimately, broader water utility networks in Tanzania.

REFERENCES

DFRobot. (2019). Analog pH sensor/meter kit (SKU: SEN0161). Atlas Scientific. https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_

Hakimi, I. M., & Jamil, Z. (2021). Development of water quality monitoring device using Arduino UNO. IOP Conference Series: Materials Science and Engineering, 1144(1), 012064. https://doi.org/10.1088/1757-899x/1144/1/012064

Hong, W. J., Shamsuddin, N., Abas, E., Apong, R. A., Masri, Z., Suhaimi, H., Gödeke, S. H., & Noh, M. N. A. (2021). Water quality monitoring with Arduino-based sensors. Environments – MDPI, 8(1), 1–15. https://doi.org/10.3390/environments8010006

Kiran, S., & Sriramoju, S. B. (2018). A study on the applications of IoT. Indian Journal of Public Health Research and Development, 9(11), 1173–1175. https://doi.org/10.5958/0976-5506.2018.01616.9

Myers, D. N. (2014). Why monitor water quality? United States Geological Survey. http://water.usgs.gov/owq/WhyMonitorWaterQuality.pdf

Pokhrel, S., Pant, A., Gautam, R., & Kshatri, D. B. (2020). Water quality monitoring system using IoT. Journal of Innovations in Engineering Education, 3(1), 155–164. https://doi.org/10.3126/jiee.v3i1.34337

Sabari, M., Aswinth, P., Karthik, T., & Bharath Kumar, C. (2020). Water quality monitoring system based on IoT. Proceedings of ICDCS 2020 – 5th International Conference on Devices, Circuits and Systems, 279–282. https://doi.org/10.1109/ICDCS48716.2020.243598

United Nations World Water Assessment Programme [UN WWAP]. (2003). Water for people, water for life: The United Nations World Water Development Report. UNESCO/Berghahn Books.

Williams, T. O., Johnston, R., & Merrey, D. J. (2015). Accessing and putting water to productive use in sub-Saharan Africa. International Water Management Institute (IWMI).

World Bank. (2016). Agriculture is the largest consumer of water in sub-Saharan Africa and a rapidly rising population is increasing food demand and water scarcity. World Bank Group.