

Ateneo de Manila University

Archium Ateneo

Electronics, Computer, and Communications
Engineering Faculty Publications

Electronics, Computer, and Communications
Engineering Department

2021

Software and Data Visualization Platform for Groundwater Level and Quality Monitoring System

Carlos M. Oppus

Ateneo de Manila University

Maria Aileen Leah G. Guzman

Ateneo de Manila University

Maria Leonora C. Guico

Ateneo de Manila University

Jose Claro N. Monje

Ateneo de Manila University, jcmonje@ateneo.edu

Mark Glenn F. Retirado

Ateneo de Manila University

See next page for additional authors

Follow this and additional works at: <https://archium.ateneo.edu/ecce-faculty-pubs>



Part of the [Agriculture Commons](#), [Engineering Commons](#), and the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Oppus, C. M., Guzman, M. A. L. G., Guico, M. L. C., Monje, J. C. N., Retirado, M. G. F., Kwong, J. C. T., Ngo, G. C., & Domingo, A. J. (2021). Software and data visualization platform for groundwater level and quality monitoring system. *Journal of Physics: Conference Series*, 1803(1), 012007. <https://doi.org/10.1088/1742-6596/1803/1/012007>

This Article is brought to you for free and open access by the Electronics, Computer, and Communications Engineering Department at Archium Ateneo. It has been accepted for inclusion in Electronics, Computer, and Communications Engineering Faculty Publications by an authorized administrator of Archium Ateneo. For more information, please contact oadrcw.ls@ateneo.edu.

Authors

Carlos M. Oppus, Maria Aileen Leah G. Guzman, Maria Leonora C. Guico, Jose Claro N. Monje, Mark Glenn F. Retirado, John Chris T. Kwong, Genevieve C. Ngo, and Annael J. Domingo

PAPER • OPEN ACCESS

Software and data visualization platform for groundwater level and quality monitoring system

To cite this article: C Oppus *et al* 2021 *J. Phys.: Conf. Ser.* **1803** 012007

View the [article online](#) for updates and enhancements.

You may also like

- [Rational allocation of water resources based on ecological groundwater levels: a case study in Jinghui Irrigation District in China](#)
H Li, W B Zhou, Q G Dong et al.
- [Domestic well vulnerability to drought duration and unsustainable groundwater management in California's Central Valley](#)
R A Pauloo, A Escriva-Bou, H Dahlke et al.
- [Effects of irrigation on groundwater recharge under deep buried depth condition](#)
DONG Qiguang, ZHOU Weibo, ZHANG Yang et al.



The Electrochemical Society
Advancing solid state & electrochemical science & technology

241st ECS Meeting

May 29 – June 2, 2022 Vancouver • BC • Canada

Abstract submission deadline: Dec 3, 2021

Connect. Engage. Champion. Empower. Accelerate.
We move science forward



Submit your abstract



Software and data visualization platform for groundwater level and quality monitoring system

C Oppus¹, M A L Guzman², J C Monje¹, M L Guico¹, G Ngo¹,
A Domingo¹, J C Kwong¹, M G Retirado¹

¹Electronics, Computer, and Communications Department, Ateneo de Manila University, Katipunan Avenue, 1108, Philippines

²Environmental Science Department, Ateneo de Manila University, Katipunan Avenue, 1108, Philippines

Email: coppus@ateneo.edu¹

Abstract. Rapid urbanization and increasing population come with the increased extraction and use of groundwater resources. To track the effect of these activities on groundwater level and quantity, a system for real-time monitoring is devised. In this paper, we present a software system design that enables a locally-developed groundwater level and water quality monitoring hardware setup to gather water quality parameter data, send it to a cloud server, and present organized data for better visualization. The hardware setup consists of an Arduino microcontroller. Upon deployment, the hardware setup is linked to an Android application that connects to the web-based platform.

1. Introduction

The bulk of the water supply in the Philippines is groundwater [1]. Based on the water rights granted by the National Water Resources Board (NWRB), 49% of groundwater is consumed by the domestic sector, and the remaining percentage is shared by agriculture, industry, and other sectors [2]. Due to the growing population in the country, the demand for water surges, resulting in a continued increase in the extraction of groundwater [1]. With almost half the population relying on groundwater for drinking [3], it is imperative that constant monitoring of groundwater level and quality be implemented. Other near-real time monitoring systems with data visualization utilizing different water-pertinent sensors had also been developed to monitor rain, flood, rivers, and sea levels [4][5][6].

The Groundwater Management Plan (GMP) project seeks to provide accessible real-time water level and quality data from various monitoring wells located all over the Philippines. This data can guide policymakers in making informed decisions regarding groundwater use and regulation [7].

To better provide meaningful data to various stakeholders, the goal of this paper is to develop a software system that would assist the project in delivering real-time water level and quality data. This is to be done by (1) developing a program that would enable the designed hardware setup to perform data calibration, gathering, and transmission of groundwater quality and quantity parameters, (2) setting-up a cloud service that stores the data transmitted by the hardware, and (3) coming up with an avenue to visually showcase the data gathered by the deployed system [8].

2. System Design

The general framework for this paper focuses on the sensor node (sensor control box), cloud server, and data dashboard through a web-based platform. The interaction, as shown in Figure 1, follows a flow starting from the sensors to the box, then to the cloud service, and finally to the visualization component. While not integral to the hardware setup, a supplementary Android application comes



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

handy when calibrating or debugging the hardware system. Similar systems have also been developed with data visualization presented to either an existing open- or an exclusively- developed platform [9][10][11].

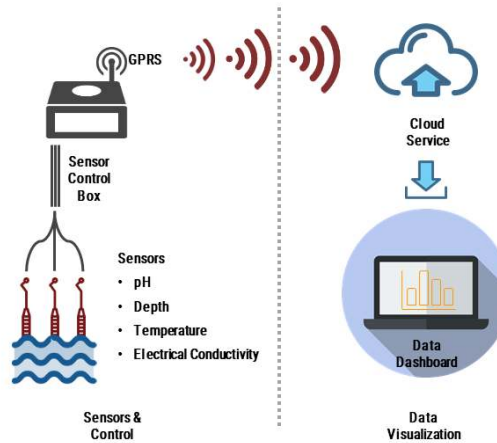


Figure 1. General Software System Design.

2.1. Sensor Node

The sensor node program is in the Arduino language. This is in line with the use of the Bluno Mega 2560 as the microcontroller dedicated to data gathering and transmission.

After calibration, the node gathers and transmits, at an hourly rate, the following parameters: pH levels, Temperature in Celsius, Electrical Conductivity (EC) in micro Siemens/centimeter (μS), Total Dissolved Solids (TDS) in parts per million (ppm), Salinity in practical salinity unit (psu), static water level (SWL) in meters below ground, and battery reading in volts.

The data transmission from the node to a cloud server is done through a GSM/GPRS module. To utilize the module, a library is compiled based on the GPRS library so that the methods relating to data transmission can be called by the main Arduino program. This library has the capability to flush the stream data, and send it over HTTPS reliably.

2.2. Android Application

The Android application is coded in Android Studio 3.4 with Gradle as the build tool. The creation of an application serves as a convenient feature to access the microcontroller for calibration and debugging.

The application is capable of sending commands to the microcontroller and receiving data through serial communication at a common baud rate. Taking into full consideration the microcontroller's specifications, the serial communication may be established using Bluetooth connection or via USB On-the-Go (OTG).

2.3. Cloud Service

Serving as the bridge between the sensor node and the web-based platform, the cloud service is where the data from the hardware setup is archived and accessed for display through REST APIs. The cloud also houses the following information about a well site where the sensor node is deployed:

- Sensor node ID
- Site name, address, and area
- Latitude and Longitude
- Well's as-built SWL
- Installation date

There are two types of tables that are kept in the cloud – a table for storing the information on a well site, and a table for storing all the data transmitted by the sensor node.

2.4. Web-based Platform

The web-based platform ensures better and useful presentation of the data for the general public. There are two ways to access the platform – privately and publicly. The private access is used mainly to track the regular and consistent data transmission and, thus, requires a user account to access. The public access caters more to the general public and displays a simplified and/or graphical data presentation.

The platform is built on PHP-Laravel, a web application framework designed to provide tools for common tasks in web projects such as authentication, routing, sessions, and caching. In fetching the APIs generated, Promise and Fetch API interface is used. Fetch allows a program to make network requests, similar to XMLHttpRequest (XHR). One of the main differences is that Fetch uses Promises, which provides a way to avoid callbacks. A Promise is an object that represents the response state (pending, fulfilled or rejected) of an asynchronous function.

3. Program Design

3.1. Sensor Node

The Arduino sketch for the sensor node can be divided into three function blocks: (1) calibration, (2) data gathering, and (3) data transmission. The calibration function runs only once upon booting while the gathering and transmission of data are executed successively in hourly intervals.

During calibration, the program is designed to be activated upon boot. In the flowchart in Figure 2, the user is first prompted to choose whether to enter the calibration mode or not. If no action is made within 20 seconds, the program continues with data gathering and transmission. The calibration process can be subdivided into five commands: calibration of the (1) pH, (2) EC, TDS, Salinity and (3) temperature, (4) read mode, and (5) manual mode.

In calibrating the pH parameter, the program will require the user to submerge the pH sensor to three (3) different pH solutions (e.g., pH 7, pH 4, and pH 10) consecutively. The program is designed to continuously read the analog voltage from the sensor. While still submerged in a corresponding pH solution, once the sensor readings have stabilized, the user can press the confirmation button to accept the calibration value. The last analog reading shown is used as the base value for the pH value of the solution.

In calibrating the EC, TDS, and Salinity, the program will require the user to record the reading from the EC sensor at a dry surface and when submerged to 1,440 μS and 12,880μS solutions. The EC sensor comes with a circuit that is able to calibrate the EC, TDS, and Salinity by following the required procedure.

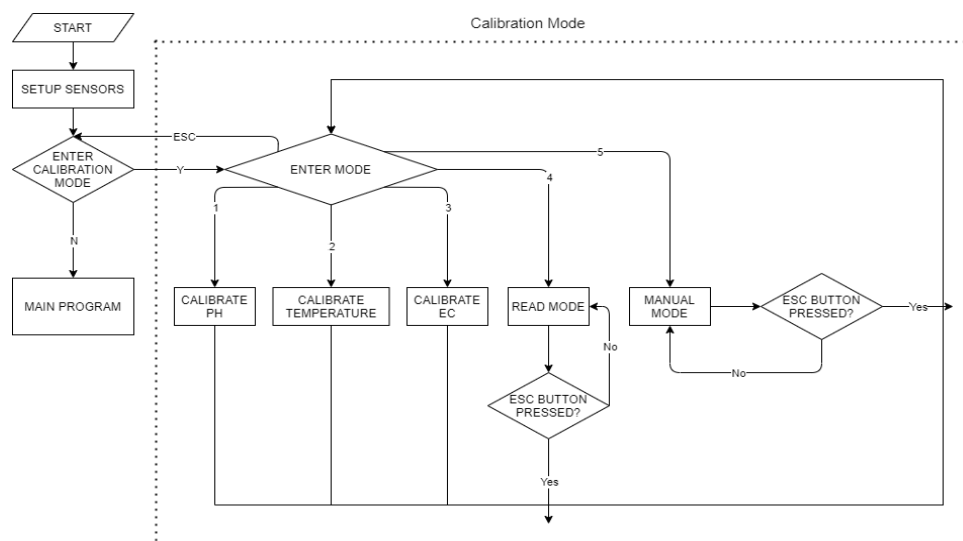


Figure 2. Sensor calibration flowchart.

In calibrating the temperature parameter, the program will require the user to submerge the temperature sensor in water at room temperature, at freezing point, and at boiling point.

During read mode, the program displays all parameter readings by the second. This ensures that the sensors are calibrated correctly before lowering the sensors into the monitoring well. This mode can also be used to check if the values returned by the sensors are stable.

The manual mode is used for parameters that require user input values such as the sensor node ID used to distinguish itself from the other sensor nodes in the cloud server. SWL calibration is also done in this mode by inputting the analog value of the depth sensor at zero depth (not submerged) and the estimated depth to groundwater to which the sensor will be submerged to.

For the pH, EC, and temperature sensors, nine successive data points are collected. These are then used to set the median for each of the sensor values. The SWL and battery voltage readings are read through the ADC of the Bluno Mega and are recorded as outright values.

Data transmission follows after all the needed parameters are read. It starts with the software-based activation of the GSM/GPRS module followed by a short delay to ensure that the module managed to latch on a base station. Before transmitting, a local copy is stored on a SD card in the sensor node. Once saved, the module is then configured in consideration for the network used and standards of communication. To ensure data transmission reliability, the system takes into account the response of the cloud server before proceeding. Should the transmission to the cloud server fail, the system tries again and stops after a predetermined number of attempts.

Upon completion of both data gathering and transmission, the program sets the GSM/GPRS module to go into sleep mode for an hour and then activates it again to gather and transmit a new set of data points, as presented in Figure 3.

3.2. Android Application

The Android application consists of a single activity consisting of buttons, a TextView widget, and an EditText widget as shown in the screenshot in Figure 4. The buttons are used to send commands to the sensor node, e.g., calibration options. The TextView serves as the display area for the serial information sent by the microcontroller. The EditText serves as the avenue for user input commands especially during manual calibration mode.

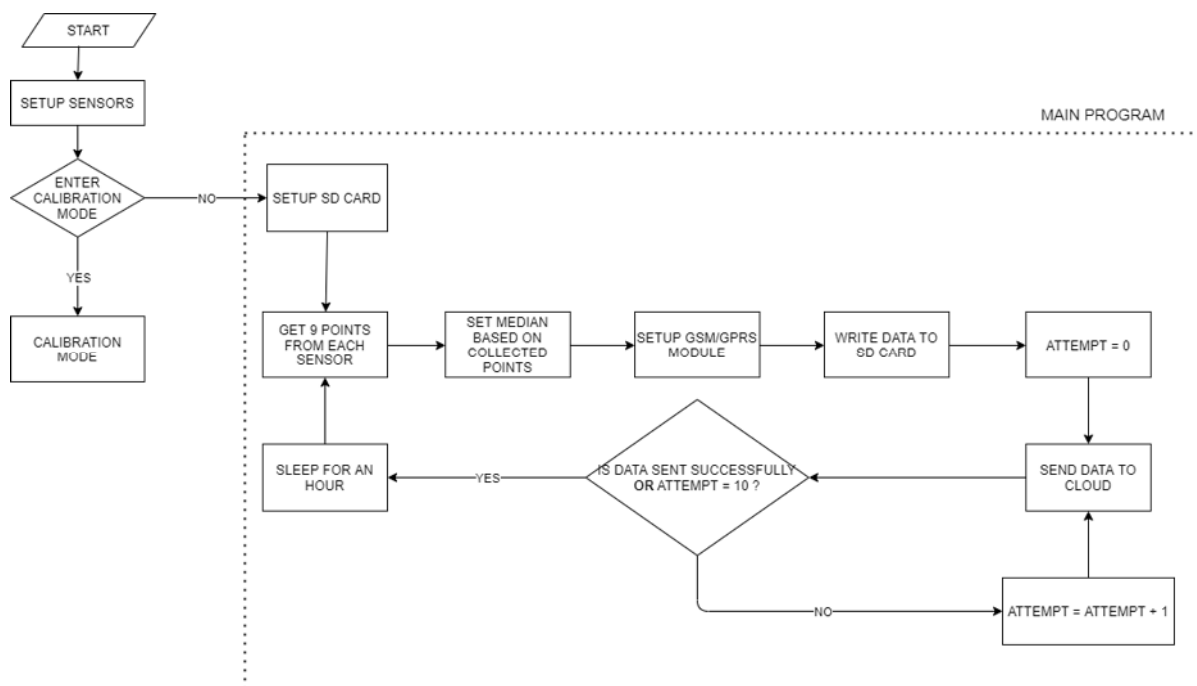


Figure 3. Data gathering and transmission flowchart.



Figure 4. Screenshot from the Android application.

To operate the application, the user must allow the request for permission to use the device's Bluetooth API or USB host API, depending on the connection made.

3.3. Cloud Service

Two REST APIs are generated – an API for accessing the well site information, and an API for accessing the data transmitted by the sensor node.

For the API accessing the well site information, four HTTP requests are created, namely, read, insert, update, and delete. This API accesses the information table about the sites where the sensor nodes are located.

Read is a GET request that returns the information on all the well sites in JSON format.

Insert is a POST request that appends a new row in the spreadsheet and requires the necessary information about the site as parameters. Out of all the parameters, the sensor node ID is strictly required while all the others may be left blank and be filled later on using the update request.

Update is a PATCH request that lets the user modify any information related to the stated sensor node ID. All information, except the sensor node ID, can be modified through this request. This requires the chosen sensor node ID to initially exist in the records before any modifications on the information.

Delete is a DELETE request that erases all the information related to the sensor node ID specified. This cannot be undone once invoked and does not affect the other API.

For the API accessing the data transmitted by the sensor node, three HTTP requests are created, namely, addWell, addItem, and getItem. This API accesses the tables containing the data transmitted by the sensor nodes.

The addWell is a POST request that lets the user create a table for a specific sensor node. This request requires the sensor node ID as a parameter. This request is called together with the insert request of the well site information API.

The addItem is another POST request that is invoked during the data transmission part of the sensor node program. It asks for the sensor node ID and the data parameter values as input. Once successfully called, a row will be appended to the specified table.

The getItem is a GET request that returns a JSON format of all the data recorded in a table. To retrieve the data, the ID of the desired sensor node is needed.

3.4. Web-based Platform

The private platform has three pages that presents the data to the user, namely, Summary Page, View Site Page, and Add Well Site Page.

The Summary Page shown in Figure 5 displays a data table containing all the information about the sites. There are also cards that show information such as, the total number of sensor nodes deployed,

total number of areas/provinces the sensor node is deployed, and the current date and time. This Summary Page allows the user to modify any information about the site and to delete a site from the records.

On the View Site Page, the user can view and download all the data points transmitted by the sensor node in graphical and tabular format. In the screenshot of the page in Figure 6, the information about the selected site is displayed. To view all those details, the user has to first select which sensor node to view. This page allows the user to modify any information about the site selected.

On the Add Well Site Page, the user is presented with a form in which they have to fill in the necessary information about a sensor node. A screenshot of the page is displayed on Figure 7.

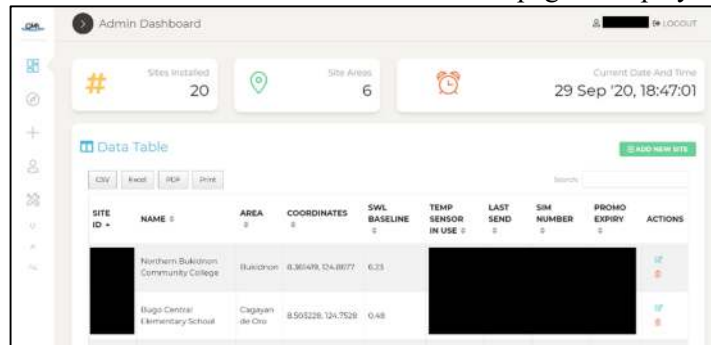
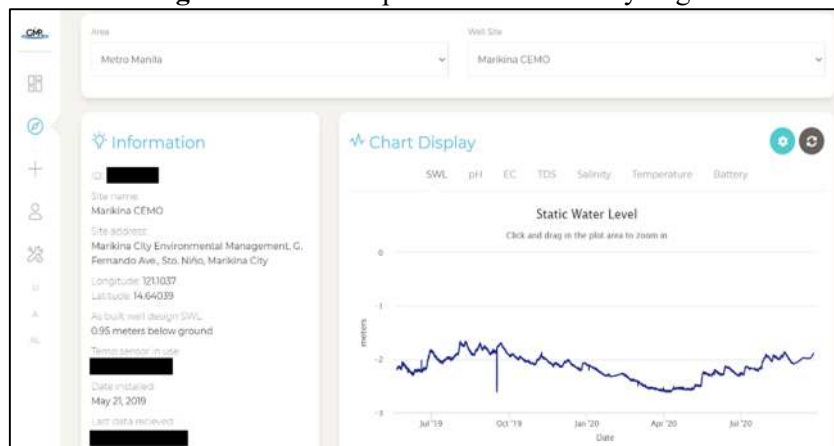


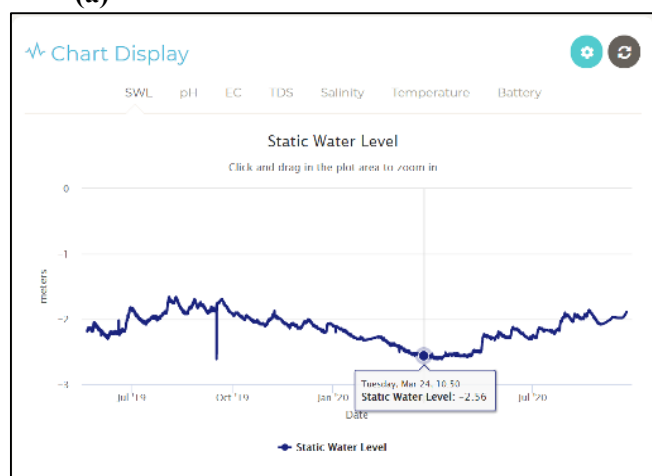
Figure 5. Screen capture of the Summary Page.



(a)



(b)



(c)

Figure 6. View Site Page: (a) full screen at first look; (b) left side; (c) right side.

Figure 7. Screen capture of the form from Add Well Site Page.

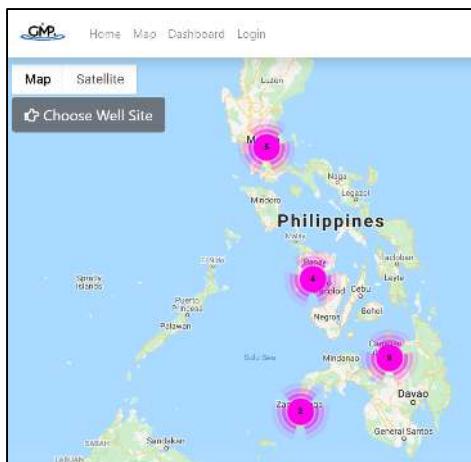


Figure 8. Screen capture of the Philippine map from Site Location Page.

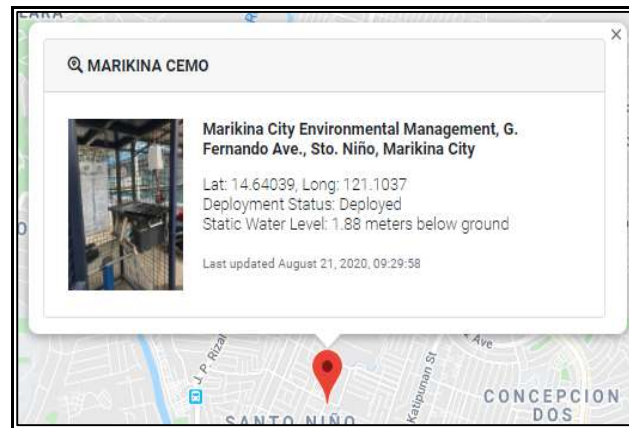


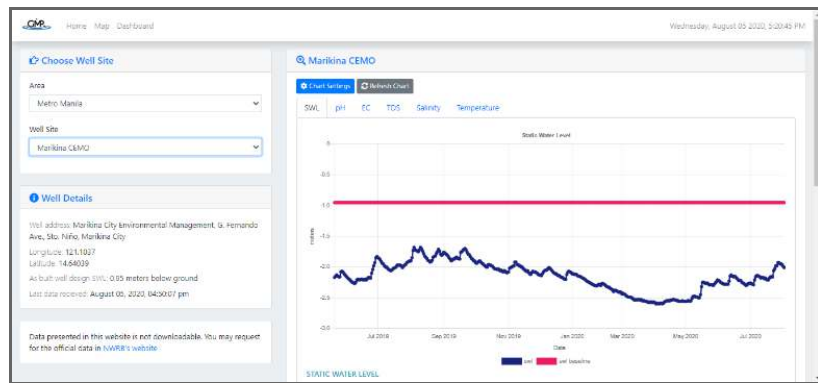
Figure 9. Screen capture of the Site Location Page with Selected Site.

The public platform, on the other hand, has two pages that showcase the data to the user, namely, the Site Location Page and the Dashboard Page.

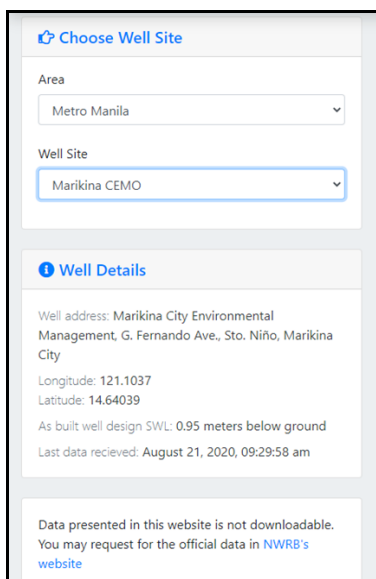
On the Site Location Page are the locations of the sensor nodes. Google Maps API is used in providing the maps and the markers for the ground well sites. A screenshot of the page is shown on Figure 8. On this page, the user can see more site information by clicking on the marker itself or by selecting the site name through the dropdown button on the upper left of the screen. An example of what the user can view upon selecting a site is shown on Figure 9. The information window contains the following details:

- The ground well site name and location
- The latest SWL reading
- The timestamp when the sensor node last sent a data
- The picture of the site itself

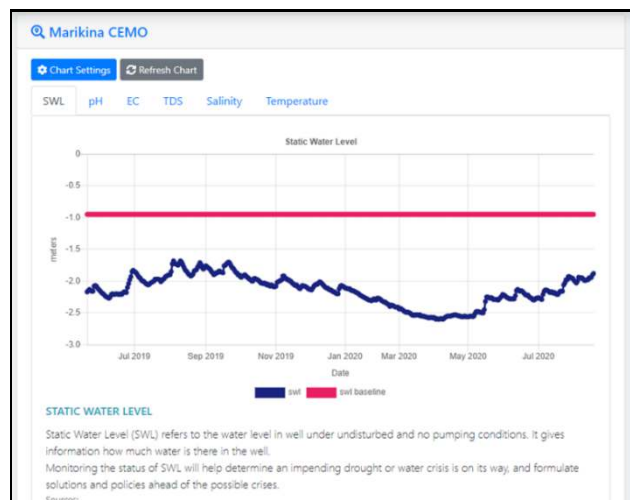
On the Dashboard Page, the user can view a graphical representation and a table of the data received from the sensor node. Figure 10 shows a screenshot of the page. The user selects which site to access through a dropdown option. Once a site is selected, the user can navigate through the tabs to select which parameter to view. Each tab shows a short description of the parameter aside from its graphical representation of the data. By default, the daily average reading is displayed, but the user has the option to view the weekly and monthly average of the data transmitted by the sensor node.



(a)



(b)



(c)

Figure 10. Dashboard Page: (a) full screen at first look; (b) left side; (c) right side

4. Results and Discussion

There are currently 20 sensor nodes deployed across 6 areas in the Philippines each continually transmitting data at an hourly rate. Before submerging the sensors, the calibrated values are cross-checked with tools that can measure the parameters needed. The Android application makes the calibration and testing process easier to carry out since it eliminated the need to open the casing just to connect the microcontroller physically to a laptop. This fully utilizes the Bluetooth feature of the microcontroller.

The cloud service employed for this project synchronizes with the Philippine time zone. This is crucial in recording the data timestamp from the deployed set-ups. The use of the cloud services provides convenience by eliminating the task of maintaining a physical server.

The web-based platform can be accessed at <https://admuwater.com>. Its overall design is simple and easy to navigate through. It is also mobile responsive. However, it is strongly recommended to view the graphs on devices with relatively large screens. The graphical view is especially useful in observing the trend of a groundwater well’s historical activity. One such instance is when noting the difference in the water level of a well site between summer and rainy seasons. In that instance, as seen in the graph on Figure 10 (c), the water level in one of the well sites is lower from March to May (Philippine summer season), but starts rising in July (which usually marks the beginning of the Philippine rainy season).

5. Conclusion

A software system that can assist the GMP project in delivering real-time water level and quality data to the public has been developed. The software system consists of a sensor node program, an Android application, a cloud service, and a web-based platform.

The sensor node can perform data calibration, gathering, and transmission. Calibration is prompted to the user at initialization and is optional while the data gathering and transmission are repeated hourly. An Android application is developed to provide ease of use in accessing the sensor node.

The cloud service houses all the information about the well sites and the data sent from the sensor nodes deployed. The web-based platform presents the data to a user in graphical and tabular form. In this age where technology continues to advance at a fast rate, it is recommended to regularly keep track of the system for hardware and software updates especially with the tools and libraries utilized.

6. References

- [1] Racadio, C. T., Castañeda, S. S., Cariño, F. A., & Mendoza, N. D. 2020. Isotopic Data for Inferring Groundwater Dynamics in Cagayan De Oro City, Philippines. *Philippine Journal of Science*, 149(1), 188-199
- [2] World Bank Country Office -Manila, PH, Philippines environment monitor 2003 : water quality, World Bank, v, 42 p
- [3] C. Boyd 2018 "Water supply and demand and the drivers of change," in *Water Policy in the Philippines: Issues, Initiatives, and Prospects*, 1st ed., ser. Global Issues in Water Policy 8, R. A. H. e. Agnes C. Rola, Juan M. Pulhin, Ed. Springer International Publishing ch. 2, pp. 18, 27.
- [4] Lalithadevi, B., Yadav, A., Pandey, A., & Adhikari, M. 2019 IoT Based WSN Ground Water Monitoring System with Cloud-Based Monitoring as a Service (Maas) and Prediction using Machine Learning. *International Journal of Innovative Technology and Exploring Engineering Regular Issue*, 9(1), 816-821. doi:10.35940/ijitee.a4316.119119
- [5] Zhou, Q., Chen, C., Zhang, G., Chen, H., Chen, D., Yan, Y., . . . Zhou, R. 2018 Real-Time Management of Groundwater Resources Based on Wireless Sensors Networks. *Journal of Sensor and Actuator Networks*, 7(1), 4. doi:10.3390/jsan7010004
- [6] Chowdury, Mohammad & Emran, Talha & Ghosh, Subhasish & Pathak, Abhijit & Alam, Mohd & Absar, Nurul & Andersson, Karl & Hossain, Mohammad 2019 IoT Based Real-time River Water Quality Monitoring System. *Procedia Computer Science*. 155. 161-168. 10.1016/j.procs.2019.08.025.
- [7] Water Resources Assessment for Prioritized Critical Areas, National Water Resources Board (NWRB), *Assessment of water resources in Metro Manila*
- [8] National Water Resource Board (NWRB) 2018 Annual Report, National Water Resource Board,
- [9] Pasika, S., & Gandla, S. T. 2020 Smart water quality monitoring system with cost-effective using IoT. *Heliyon*, 6(7). doi: 10.1016/j.heliyon.2020. e04096
- [10] Paul B. Bokingito, Orven E. Llantos 2017 Design and Implementation of Real-Time Mobile-based Water Temperature Monitoring System, *Procedia Computer Science*, Volume 124 Pages 698-705, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2017.12.207>.
- [11] A. E. Retamar et al., 2017 "HydroMet: Deployment of a large scale nationwide hydrometeorological sensor network for flood warning and monitoring," 2017 IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, pp. 1-10, doi: 10.1109/GHTC.2017.8239323.

Acknowledgments

This research is a project implemented by the Ateneo de Manila University (AdMU), funded by the Department of Science and Technology (DOST) - Philippine Council for Industry, Energy and Emerging Technology Research and Development (PCIEERD), National Water Resources Board (NWRB), and partially funded by the University Research Council (URC) of AdMU and the Ateneo Research Institute of Science and Engineering (ARISE).