

Web-based of Smart Water Quality Monitoring System

Tan Yee Han^{1*}, Firkhan Ali Hamid Ali¹

1 *Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: firkhan@uthm.edu.my
DOI: <https://doi.org/10.30880/aitcs.2025.06.01.009>

Article Info

Received: 13 June 2024
Accepted: 10 Jne 2025
Available online: 30 June 2025

Keywords

Water quality, Monitoring system,
Internet of Things (IoT)

Abstract

The Web-based of Smart Water Quality Monitoring System is a water quality monitoring system specially designed for Bao Lin Yuan Nursery to control the quality of water in the plantation. This water quality monitoring system based on the Internet of Things (IoT) is designed using the TS-300B turbidity sensor and pH sensor for enabling the detection of water turbidity and pH value of the water. The primary objective of this system is to assist the owner in monitoring water quality, safeguarding plants from potential harm due to water impurities. The use of this comprehensive system ensures careful management, thereby reducing the risks of poor water quality. The structured approach is used to develop the system using various IoT devices such as the TS-300B turbidity sensor, pH sensor kit, ultrasonic sensor and Node MCU ESP8266. This system also integrates with ThingSpeak as the database. The waterfall methodology was used to complete the project because it is an iterative, phased approach that emphasizes distinct stages of development. As the result, this system able to provide user-friendly interface for the owner of the nursery to manage their staff, manage the water tank registered and monitoring status of the water tank in their plantation, making it easier to control the water quality.

1. Introduction

Water quality defined by Collins Dictionary as the degree of cleanliness determining its suitability for various purposes, is crucial for agriculture, where it influences plant growth, drinking water and fish habitats.[1] Key factors such as pH value and soluble salts significantly impact water quality. Therefore, a standard benchmark for pH value and turbidity is specified and the water conditions is classified by the Interim National Water Quality Standard (INWQS) for distinct purpose. Specifically, water with a pH value between 5.0 to 7.0 is ideal for agriculture use while water with a pH value of 5.0 to 9.0 is suitable for broader agriculture purposes according to the INWQS. As an example, Malaysia's agricultural sector, pivotal to the nation's Gross Domestic Product (GDP), confronts challenges due to water scarcity and contamination, constituting approximately 7.1% of the country's GDP in 2020.[2]

The project focuses on developing a web-based monitoring system utilizing turbidity and pH sensors. It aims to provide real-time data on water quality, accessible through a user-friendly interface. This will enable farmers and agricultural professionals to make informed decisions, ensuring optimal water usage and preventing crop damage. The system's functional modules include user login, water tank monitoring, and user management, all designed to enhance efficiency and water resource conservation in the plantation and also efficiency to manage the staff in the plantation.

The objective of this system is to design a user-friendly, dynamic and responsive web-based smart water quality monitoring system integrating with turbidity sensor and pH sensor using structured approach and develop web-based smart water quality monitoring system using web-based approach. The next objective of this system is to test the usability and functionality of the developed system.

2. Related Work

In this section, it will discuss about the related work such as Internet of Things (IoT) and existing systems which are related to the proposed system. Three existing system that will be discussed which are Water Quality and Controlling using Iot, Iot-GSM Based Controlling and Monitoring System and Water Quality Monitoring and Management of Building Water Tank Using Industrial Internet of Things.

2.1 Internet of Things (IoT)

Internet of Things can be defined as the Internet allows us to connect physical things to it, accessing data from faraway sensors and controlling things from a distance. When we combine this collected data with information from the internet, like web data, it creates new and more powerful services. This is what the Internet of Things is all about. A smart object, the basic part of the Internet of Things, is basically an embedded system (a computer system within a device) that's linked to the Internet. [3]

2.1.1 pH Sensor

The pH sensor is used to detect the pH value of water in the water tank. It consists of a pair of electrodes which one is for sensing and one as a reference. These electrodes help gauge the concentration of hydrogen ions in a solution. When ions interact, they produce an electric signal. A pH meter then translates this signal into a pH value that we can read and understand. [4]

2.1.2 Turbidity Sensor

This sensor is designed to assess water quality by checking how clear or cloudy the water is. It works by using light to spot tiny particles floating in the water. By measuring how much light passes through and how much gets scattered, it figures out how many particles are there.

2.1.3 HC-SR04 Ultrasonic Sensor

This sensor is used to detect the distance between the water surface and the surface of water tank. The ultrasonic sensor is a gadget that figures out how far away something is by sending out really high-pitched sound waves and then listening for the echoes.

2.1.4 Node MCU ESP8266

The Node MCU ESP8266 is a wireless module that sends sensor data to other devices. It uses Wi-Fi to pass on the collected data to ThingSpeak where users can easily check and see the sensed data.

2.2 Comparison with the Existing Systems

Table 1 shows the comparison between the Water Quality and Controlling using IoT [5], IoT-GSM Based Controlling and Monitoring System [6], Water Quality Monitoring and Management of Building Water Tank Using Industrial Internet of Things [7] and Web-based of Smart Water Quality Monitoring system. The table shows the comparison between the systems in terms of whether they are web-based or mobile app and the function of the system whether they are wireless where the system can be accessed remotely. It also compares whether the systems have a login page, turbidity check, live monitoring and also the system alert which are stated in the form of “/” or “X” indicating if the features are available or not respectively.

Table 1 Comparison of Related Work

Features/System	Water Quality and Controlling using IoT [5]	IoT-GSM Based Controlling and Monitoring System [6]	Water Quality Monitoring and Management of Building Water Tank Using Industrial Internet of Things [7]	Web-based of Smart Water Quality Monitoring System
Web-based / Mobile app	Mobile app	Mobile app	Mobile app	Web-based
Login and Register	X	X	X	✓
Mobile Based	✓	✓	✓	✓
Turbidity check	X	X	X	✓
Live monitoring	✓	✓	✓	✓
pH level check	✓	X	✓	✓
User Management	X	X	X	✓
Water Tank Management	X	X	X	✓

Legend: ✓ =Available; X=Not Available

3. Methodology

In this section, it will discuss the waterfall methodology used in this project and the explanation about each phase in the methodology.

3.1 Waterfall Model

The waterfall model is a linear and sequential approach to software development. It progresses through distinct phases such as requirements, analysis, design, implementation, testing, deployment and maintenance. This model is suitable for projects with well-defined and stable requirements for its nature. [8] The waterfall model is chosen due to its step-by-step process and clear milestones and deadlines. The waterfall model is included in Appendix A.

The requirement phase of the Smart Water Quality Monitoring System project establishes a solid foundation by conducting an in-depth investigation into the project background to align with its objectives. This phase identifies the technical and functional requirements and determines the project schedule and budget, ensuring the resources and timeline are accurately estimated. The output includes a detailed project proposal outlining the scope and preliminary plan, along with a Gantt chart in Appendix B that visualizes the project timeline and key milestones. Following this, the analysis phase involves a comprehensive literature review and detailed analysis of both software and hardware components. This phase also benchmarks against existing solutions to identify best practices, culminating in a research report and detailed specifications for hardware and software requirements.

The design phase focuses on creating the system's architecture and detailed design, starting with the overall system architecture, flowchart and context diagram to outline component interactions. Data flow diagrams (DFD) and an Entity-Relationship Diagram (ERD) are developed for modelling the system's structure and database schema, while the user interface is designed for optimal usability. The DFD and ERD are included in Appendix C and Appendix D respectively. The deliverables include the system architecture, flowchart, context diagram, DFD, ERD and user interface design. The implementation phase translates these designs into a functional system, developing sensors, the database, and a web-based management monitoring interface. The testing phase ensures the system's functionality through a comprehensive test plan that includes automated and manual tests, as well as User Acceptance Testing (UAT). The phase concludes with code integration and refinement, resulting in a tested, refined system and a documented test plan. The last phase which is operation phase is to launch the system.

3.2 System Development Workflow

There are total of six phases from the waterfall model. As shown in Table 2, each phase has its own assignment and output that need to produce during the entire project development. Besides that, the output had been completed within the specific days that have been given.

Table 2 Software development activities and their task

Phase	Task	Output
Requirement	<ul style="list-style-type: none"> Investigate project background 	<ul style="list-style-type: none"> Project proposal
	<ul style="list-style-type: none"> Identify requirement of the project 	<ul style="list-style-type: none"> Create Gantt chart
	<ul style="list-style-type: none"> Determine the project schedule and budget for the project 	
Analysis	<ul style="list-style-type: none"> Literature review 	<ul style="list-style-type: none"> Related work research
	<ul style="list-style-type: none"> Software and hardware analysis 	<ul style="list-style-type: none"> Hardware Requirement and software requirement
Design	<ul style="list-style-type: none"> Design architecture of smart water quality monitoring system 	<ul style="list-style-type: none"> General system architecture
	<ul style="list-style-type: none"> Design database 	<ul style="list-style-type: none"> Context diagram
	<ul style="list-style-type: none"> Design user interface 	<ul style="list-style-type: none"> Data flow diagram
		<ul style="list-style-type: none"> ERD User interface of monitoring system

Table 2 (cont.)

Phase	Task	Output
Operation	<ul style="list-style-type: none"> Launch the Web-based of Smart Water Quality Monitoring System 	<ul style="list-style-type: none"> Web-based of Smart Water Quality Monitoring System
Implementation	<ul style="list-style-type: none"> Develop the sensor in Smart Water Quality Monitoring System 	<ul style="list-style-type: none"> Smart Water Quality Monitoring System
	<ul style="list-style-type: none"> Develop database 	<ul style="list-style-type: none"> Web-based
	<ul style="list-style-type: none"> Develop Monitoring system 	<ul style="list-style-type: none"> Monitoring system
	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">
Testing	<ul style="list-style-type: none"> System Test Plan 	<ul style="list-style-type: none"> Integrate and
	<ul style="list-style-type: none"> User Acceptance Testing 	<ul style="list-style-type: none"> refine the coding
	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">

3.3 System Design

This section listed out the hardware requirements and software requirements that contributed to produce the system architecture and system development

Table 3 shows the hardware requirement needed to develop Web-based Smart Water Quality Monitoring System.

Table 3 Hardware requirements for the proposed system

No	Hardware	Function
1	Arduino UNO	To read input from the sensor and turn it to output
2	pH sensor kit	To detect the pH value of the water
3	TS-300B turbidity sensor	To detect the cloudiness of the water
4	HC-SR04 ultrasonic sensor	To detect the water level in the water tank
5	NodeMCU ESP8266 WiFi module	To connect Laptop to Wi-Fi using a Universal Serial Bus (USB) cable
6	Laptop	To complete the development of system using programming language

Table 4 shows the software requirement needed to develop Web-based Smart Water Quality Monitoring System.

Table 4 *Software requirements for the proposed system*

No	Software	Function
1	Arduino IDE	To code and compile for Arduino
2	Google Firebase	To store data
3	ThingSpeak	To visualise and analyse live data stream in the cloud
4	Visual Studio Code	To develop web-based system

Table 5 illustrated functional requirements analysis explains the interaction between the system and the environment and how the system functions.

Table 5 *Functional requirements for the proposed system*

No	Module	Function
1	Login	-Users is able to login using valid username and password
2	Reset Password	-Users able to reset the password for their account
3	Water tank Monitoring	-Users can manage the water tank status in the list table with the data collected and bar chart
4	Water tank Management	- Users able to add, update the information of the water tank and delete the water tank added.
5	Staff Management	-Admin able to add staff account or delete the staff account.
6	Personal Information	-The system shows the personal information for users and users can update their profile

The system architecture will consist of several components such as HC-SR04 ultrasonic sensor, pH sensor kit, TS-300B turbidity sensor and Node MCU ESP8266 WiFi module. All the data collected will be stored to the ThingSpeak. Fig.1 shows general system architecture of the web-based smart water quality monitoring system will be included

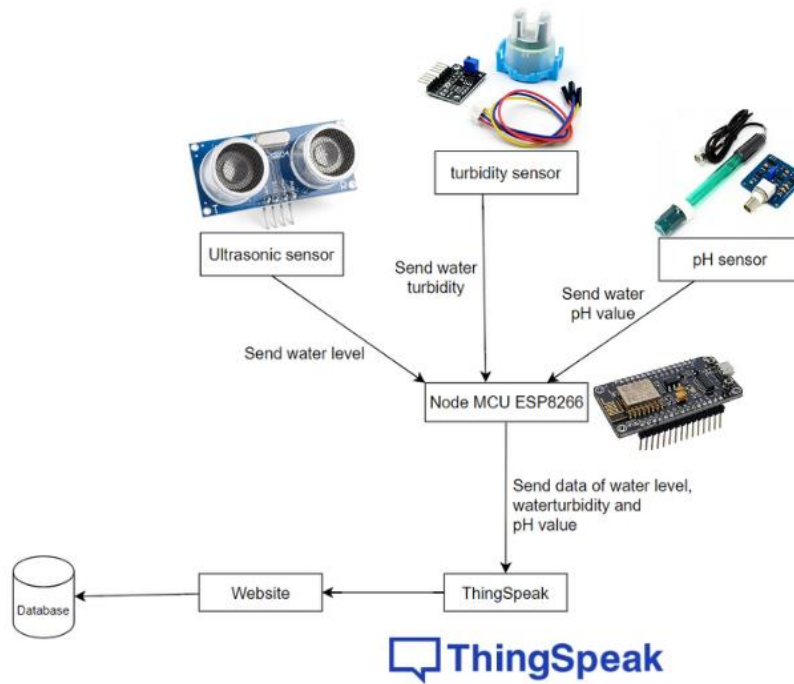


Fig.1 General System Architecture of for Web-based Smart Water Quality Monitoring System

Context diagram is a high-level visualization displaying a system as a single process interacting with external entities. There are three components in the context diagram which are external entities, process and data flow lines. The Fig. 2 below shows the context diagram of the proposed system.

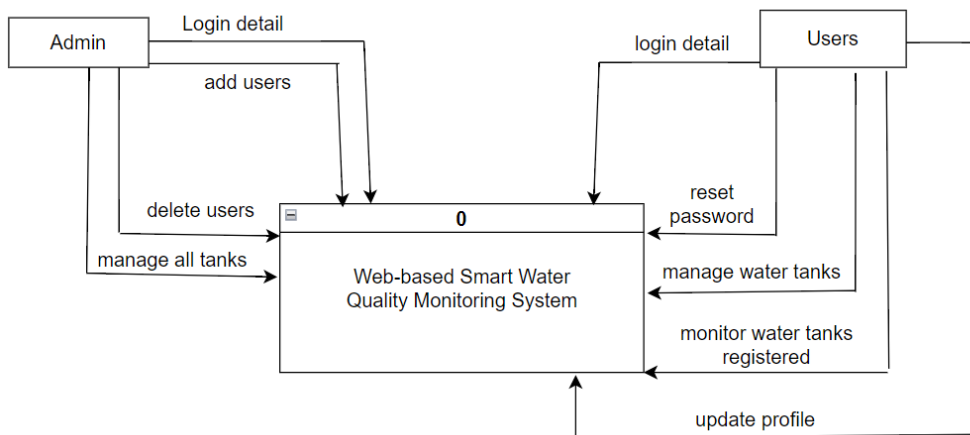


Fig.2 Context Diagram of for Web-based Smart Water Quality Monitoring System

4. Result and Discussion

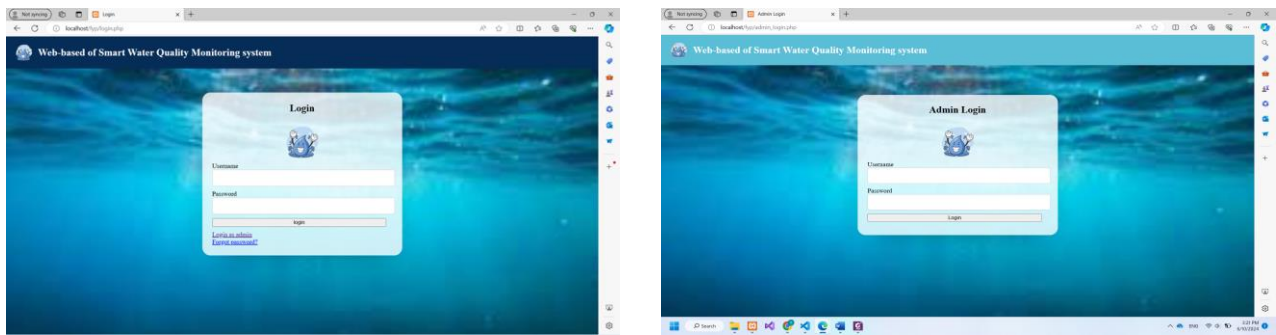
Web-based of Smart Water Quality Monitoring System is implemented using Arduino C++ Programming Language, Hypertext Preprocessor (PHP), Hypertext Markup Language (HTML), Cascading Style Sheet (CSS) and

Javascript. The database system used is MySQL which is extensively utilized for data management and storage in a variety of applications.

4.1 System Implementation

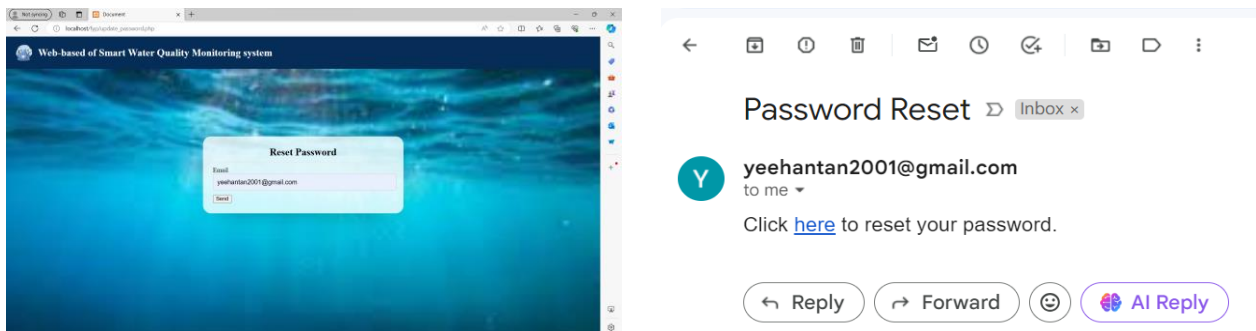
The implementation of system included Login Module, Reset Password Module, Water Tank Management Module, Water Tank Monitoring Module, Staff Management Module and Personal Information Module.

Fig.3 shows the login module of web-based Smart Water Quality Monitoring System for user site and admin panel.

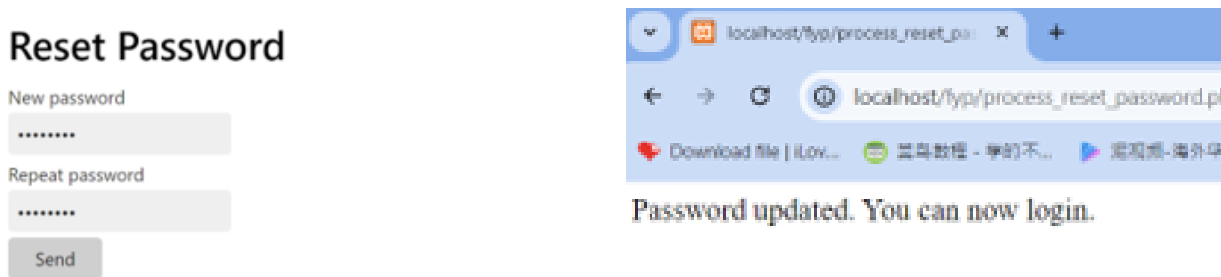


(a) (b)
Fig.3 Interface (a) Login Interface for user (b) Login Interface for admin panel

Fig.4 and Fig.5 shows the reset password process. The users are required to enter their registered email to reset password. After that, user will receive email to reset their password and click the link sent to their email to reset their password.



(c) (d)
Fig.4 Interface (c) Reset password for user (d) Email received by the user



(e) (f)
Fig.5 Interface (e) Reset password for entering new password (f) Reset password successfully

Fig.6 shows the code segment of the reset password which it initializes and configures PHPMailer to send emails using SMTP. It sets up the SMTP server details, including host, port, encryption type and enables HTML formatting for the email body. Therefore, user received an email to reset their password

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

require __DIR__ . "/vendors/vendors/autoload.php";

$mail = new PHPMailer(true);

// $mail->SMTPDebug = SMTP::DEBUG_SERVER;

$mail->isSMTP();
$mail->SMTPAuth = true;

$mail->Host = "smtp.gmail.com";
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
$mail->Port = 587;
$mail->Username = "yeehantan2001@gmail.com";
$mail->Password = "adeiatkshdmgycg";

$mail->isHtml(true);

return $mail;
```

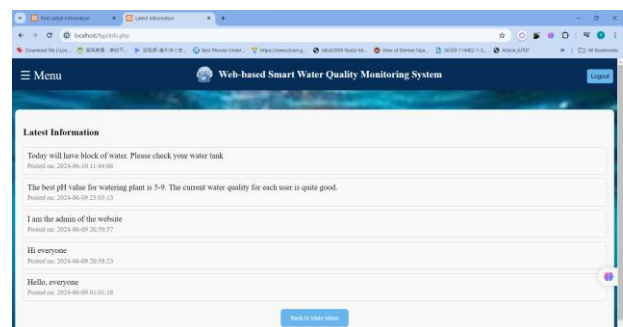
(i)

Fig.6 Code segment (i) Reset password

Fig.7 and Fig.8 shows the latest information function for admin to share their information with the staff and the code segment for posting latest information. The admin can write something and click on "Post" button and it will redirect back to dashboard in admin panel and show the posted information. For user site, the user can see the information posted by the admin.



(j)



(k)

Fig.7 Interface (c) Posting new information in admin panel (d) Latest information shown in user site

```

1 <?php
2 session_start();
3 include('./config.php');
4
5 // Check if the user is an admin
6 if (!isset($_SESSION['adminid'])) {
7     header("Location: admin_login.php");
8     exit();
9 }
10
11 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
12     $message = $con->real_escape_string($_POST['message']);
13     $query = "INSERT INTO latest_information (message) VALUES ('$message')";
14     if ($con->query($query) === TRUE) {
15         // Redirect to admin_home.php after posting
16         header("Location: admin_home.php");
17         exit();
18     } else {
19         echo "Error: " . $query . "<br> " . $con->error;
20     }
21 }
22 >?
    
```

(l)

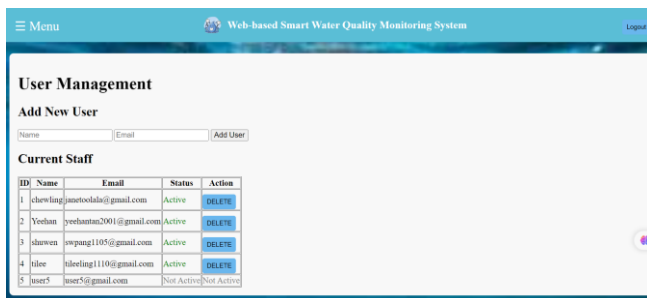
```

1 <?php
2 session_start();
3 include('./config.php');
4
5 // Fetch all the information
6 $query = "SELECT message, created_at FROM latest_information ORDER BY id DESC";
7 $result = $con->query($query);
8 $messages = $result->fetch_all(MYSQLI_ASSOC);
9 >?
    
```

(m)

Fig.8 Code segment (l) Post latest information in admin panel (f) Received information posted

Fig.9 shows the user management in the admin panel and code segment to add and delete the user. The admin required to enter the username and email and click on “Add Staff” button to create new user account with default password. After clicking add button, the new user added shown in the list table.



(n)

```

<?php
include "config.php";
$message = "";

// Check if the form is submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    // create
    if (isset($_POST['create'])) {
        // Verify that all expected keys exist
        if (isset($_POST['username'], $_POST['email'], $_POST['password'])) {
            $username = $_POST['username'];
            $email = $_POST['email'];
            $password = $_POST['password']; // Default password

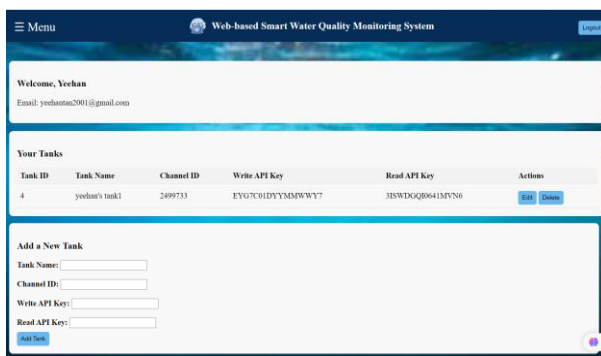
            $sql = "INSERT INTO users (username, email, password, status) VALUES (?, ?, ?, 'Active')"; // Set status as 'Active'
            $stmt = mysqli_prepare($con, $sql);
            mysqli_stmt_bind_param($stmt, "sss", $username, $email, $password);
            mysqli_stmt_execute($stmt);
            $message = "div class='success'>User added successfully</div>";
        } else {
            $message = "Please fill in all fields.";
        }
    }

    // delete
    if (isset($_POST['delete'])) {
        $id = $_POST['delete'];
        $sql = "UPDATE users SET status = 'Not Active' WHERE userid = ?"; // Set status as 'Not Active'
        $stmt = mysqli_prepare($con, $sql);
        mysqli_stmt_bind_param($stmt, "i", $id);
        mysqli_stmt_execute($stmt);
        header("Location: add_user.php"); // Redirect to avoid deletion on refresh
        exit();
    }
}
    
```

(o)

Fig.9 (n) Interface for user management in admin panel (o)Code segment to create and delete user

Fig.10 shows the Tank management in user site and admin site. The user can enter the information of the water tank such as tank name, channel id, read api key and write api key to register a water tank. The registered water tank shown in the list table with edit and delete button which enable the user to update the water tank information if there is any mistake in the old information and delete the registered water tank if no using the water tank again. In the user site, they can only see the water tank registered by themselves while for the admin, the admin can see the information of the tank registered by all the users.



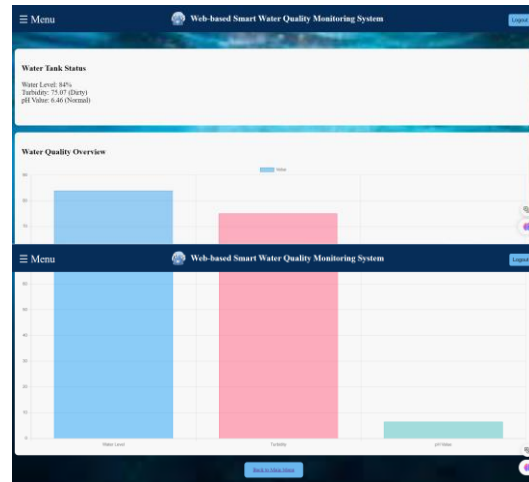
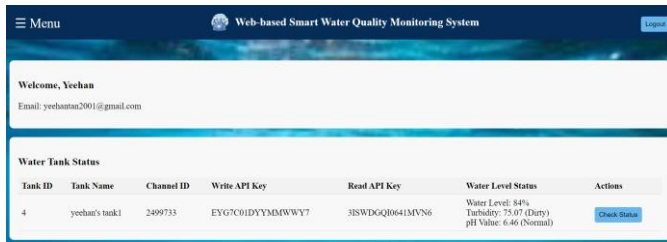
(p)



(q)

Fig.10 Interface (p) Tank management in user site (q) Tank management in admin site

Fig.11, Fig.12 and Fig.13 shows the water tank monitoring in user site and code segment to fetch data from thingspeak and turns the data collected to the bar chart. There are three categories for the water turbidity and pH value. For example, the turbidity less than 20 is stated as clear, between 20 and 50 is stated as cloudy and more than 50 is stated as dirty. The pH value is categorised to three section which is less than 5 is stated as acidic, in between 6 to 9 is stated as normal and more than 9 is stated as alkaline.



(r)

(s)

Fig.11 Interface (r) Tank monitoring in user site (q) Tank monitoring in bar chart in user site

```

5 // Check if tank_id is provided
6 if (!isset($_GET['tank_id'])) {
7     echo json_encode(["error" => "No tank selected."]);
8     exit();
9 }
10
11 $tankId = $_GET['tank_id'];
12 $userId = $_SESSION['userId'];
13
14 // Fetch tank details from the database
15 $stmt = $conn->prepare("SELECT channel_id, read_api_key FROM tanks WHERE tank_id = ? AND user_id = ?");
16 $stmt->bind_param("iI", $tankId, $userId);
17 $stmt->execute();
18 $result = $stmt->get_result();
19 $tank = $result->fetch_assoc();
20 $stmt->close();
21
22 if (!$tank) {
23     echo json_encode(["error" => "Tank not found or access denied."]);
24     exit();
25 }
26
27 $channel_id = $tank['channel_id'];
28 $read_api_key = $tank['read_api_key'];
29
30 // Fetch data from Thingspeak using $channel_id and $read_api_key
31 $thingspeak_url = "https://api.thingspeak.com/channels/$channel_id/feeds.json?api_key=$read_api_key&results=1";
32
33 // Initialize curl session
34 $ch = curl_init();
    
```

```

35 // Set curl options
36 curl_setopt($ch, CURLOPT_URL, $thingspeak_url);
37 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
38
39 // Execute curl request and get the response
40 $response = curl_exec($ch);
41
42 // Check if the request was successful
43 if ($response === false) {
44     echo json_encode(["error" => "Failed to fetch data from thingspeak: " . curl_error($ch)]);
45     curl_close($ch);
46     exit();
47 }
48
49 // Close curl session
50 curl_close($ch);
51
52 // Decode JSON response
53 $data = json_decode($response, true);
54
55 // Check if data is available
56 if (!isset($data['feeds'][0])) {
57     $feed = $data['feeds'][0];
58     $percentage = $feed['field1'];
59     $turbidityValue = $feed['field2'];
60     $pHValue = $feed['field3'];
61
62 // calculate turbidity
63 $turbidity = map($turbidityValue, 0, 750, 100, 0);
64 $turbidity = number_format($turbidity, 2); // set to two decimal points
65 $turbidityStatus = getTurbidityStatus($turbidity);
    
```

(t)

(u)

Fig.12 Code segment (t) Fetch data from Thingspeak (u) Fetch data from Thingspeak

```

66 // Initialize db connection
67 $pdo = new PDO($dsn, $username, $password);
68
69 // Query data for database
70 $stmt = $pdo->prepare("SELECT tank_id, user_id, percentage, turbidity, turbidity_status, ph_value, ph_status FROM water_quality_data WHERE tank_id = ? AND user_id = ?");
71 $stmt->bind_param("iI", $tankId, $userId);
72 $stmt->execute();
73 $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
74
75 // Fetch the data as json
76 echo json_encode([
77     "percentage" => $percentage,
78     "turbidity" => $turbidity,
79     "turbidity_status" => $turbidity_status,
80     "ph_value" => $ph_value,
81     "ph_status" => $ph_status
82 ]);
83
84 // else {
85 //     echo json_encode(["error" => "No data available."]);
86 // }
87
88 // function to map values
89 function map($value, $fromLow, $fromHigh, $toLow, $toHigh) {
90     return ($value - $fromLow) * ($toHigh - $toLow) / ($fromHigh - $fromLow) + $toLow;
91 }
92
93 // function to determine turbidity status
94 function getTurbidityStatus($turbidity) {
95     if ($turbidity < 20) {
96         return "clear";
97     } else if ($turbidity <= 50) {
98         return "cloudy";
99     } else {
100         return "dirty";
101     }
102 }
    
```

```

103 // Create bar chart
104 const ctx = document.getElementById('waterQualityBarChart').getContext('2d');
105 new Chart(ctx, {
106     type: 'bar',
107     data: {
108         labels: ['Water Level', 'Turbidity', 'pH Value'],
109         datasets: [
110             {
111                 label: 'Value',
112                 data: [
113                     data.percentage,
114                     data.turbidity,
115                     data.pHValue
116                 ],
117                 backgroundColor: [
118                     'rgba(54, 162, 235, 0.5)',
119                     'rgba(255, 99, 132, 0.5)',
120                     'rgba(75, 192, 192, 0.5)'
121                 ],
122                 borderColor: [
123                     'rgba(54, 162, 235, 1)',
124                     'rgba(255, 99, 132, 1)',
125                     'rgba(75, 192, 192, 1)'
126                 ],
127                 borderWidth: 1
128             }
129         ],
130     },
131     options: {
132         responsive: true,
133         scales: {
134             y: {
135                 beginAtZero: true
136             }
137         }
138     }
139 });
    
```

(v)

(w)

Fig.13 Code segment (v) Fetch data from Thingspeak (w) Turns data into bar chart

4.2 System Testing

The proposed system must be tested before releasing. The system must be bug-free and meet criteria and this procedure ensures system performance. The system was tested using a test plan and user acceptance testing.

4.2.1 System Testing

The test plan outlines the system’s testing strategy, expected result and actual result. In this test plan, it describes the testing process and guides testing actions. Table 6 and Table 7 shows test plan for modules in hardware and website.

Table 6 *Test plan for Smart Water Quality Monitoring Module*

Test	Task	Output
Measuring the water level of the water tank by using ultrasonic sensor	<ul style="list-style-type: none"> The system detects the water level of the water tank. 	Pass
	<ul style="list-style-type: none"> The data from the ultrasonic sensor is sent to the ThingSpeak 	Pass
Measuring the water turbidity in the water tank by using TS-300B turbidity sensor	<ul style="list-style-type: none"> The system detects the water turbidity in the water tank. 	Pass
	<ul style="list-style-type: none"> The data from the turbidity sensor is sent to the ThingSpeak 	Pass
Measuring the pH value of the water by using pH sensor	<ul style="list-style-type: none"> The system detects the pH value of the water. 	Pass
	<ul style="list-style-type: none"> The data from the pH sensor is sent to the ThingSpeak 	Pass

Table 7 Test plan for Smart Water Quality Monitoring Website

Module	Task	Output
Account Login Module	<ul style="list-style-type: none"> • The user enters username and password to login to the account of the website 	Pass
	<ul style="list-style-type: none"> • After entering the correct username and password, login successfully to the homepage of the website 	Pass
	<ul style="list-style-type: none"> • Error message when entering wrong username or password 	Pass
Reset Password Module	<ul style="list-style-type: none"> • After clicking “Forgot Password” button, user redirected page to enter their email. 	Pass
	<ul style="list-style-type: none"> • An email is sent to the user to reset their password 	Pass
	<ul style="list-style-type: none"> • Login using new password successfully 	Pass
Water Tank Management Module	<ul style="list-style-type: none"> • All registered water tank shown in admin panel 	Pass
	<ul style="list-style-type: none"> • Added water tank successfully in the staff site 	Pass
	<ul style="list-style-type: none"> • Delete Water tank successfully in the staff site with an alert message 	Pass
	<ul style="list-style-type: none"> • Update or edit water tank information in the staff site 	Pass
Water Tank Monitoring Module	<ul style="list-style-type: none"> • All registered water tank shown in admin panel 	Pass
	<ul style="list-style-type: none"> • Only registered water tank shown in the table in staff site 	Pass
	<ul style="list-style-type: none"> • Water tank status shown in written form and bar chart 	Pass
Staff Management Module	<ul style="list-style-type: none"> • All registered user shown in the list table in the admin panel 	Pass
	<ul style="list-style-type: none"> • Added a new user in the admin panel 	Pass
	<ul style="list-style-type: none"> • User created by the admin login successfully 	Pass
Personal Information Module	<ul style="list-style-type: none"> • Personal information for user shown in the staff site 	Pass
	<ul style="list-style-type: none"> • Update or edit personal information in the staff site 	Pass

4.2.2 User Acceptance Test Result

During User Acceptance Testing (UAT), ten users get to explore and test a web-based of Smart Water Quality Monitoring System. These users perform tasks like manage the water tank and provide feedback on the system’s usability. User Acceptance Testing ensures that the system meet all the requirement and provide user-friendly experience to the users. Fig. 14 below shows the user acceptance testing result after testing this system.

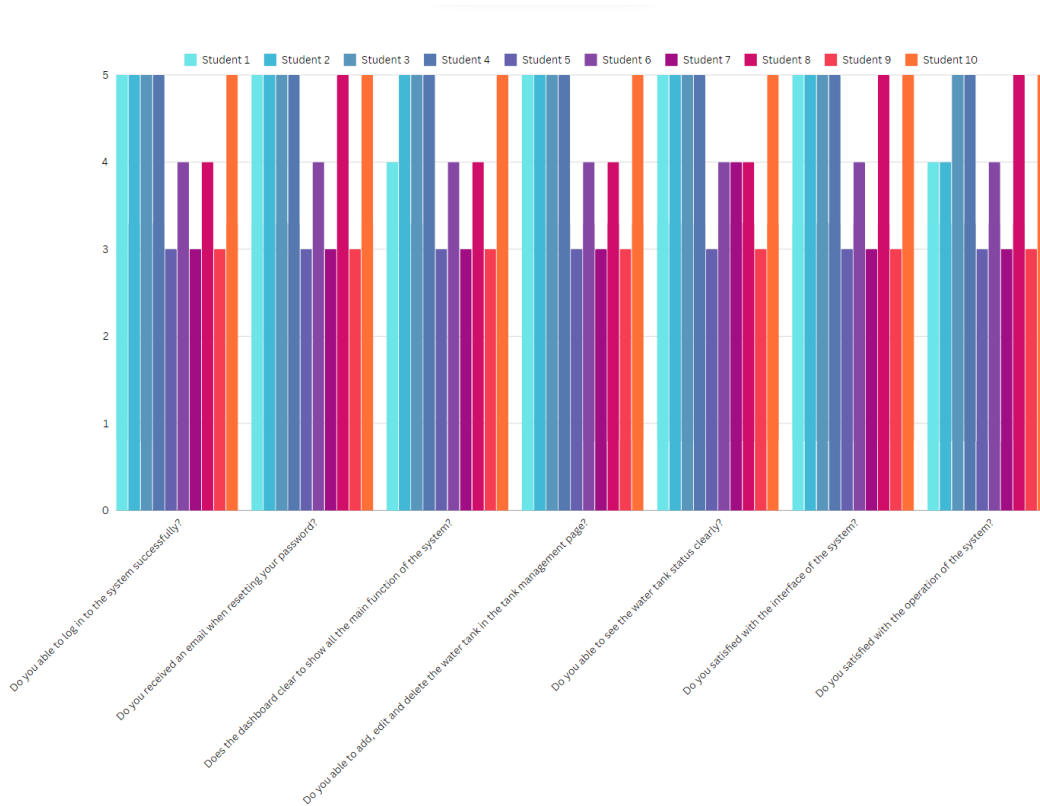


Fig. 14: User Acceptance Testing

5. Conclusion

The Web-based Smart Water Quality Monitoring System offers several advantages which convenience, time-saving, user-friendly, and flexibility. Users can easily monitor the water quality in their registered water tank to safeguard the water quality used for plant watering by using the turbidity sensor and pH sensor.

There are also limitations faced by the developed system. For example, the sensor depends on stable internet and power supply to continue working to send the data to the ThingSpeak. Besides that, the scalability concerns is also a limitation in this system because the system’s ability to scale with an increasing number of sensors and water tanks causing the system may not handle the larger deployments. Future improvements include expand sensor network and data analytics to integrate additional sensors and implement advanced data analytics to provide comprehensive water quality assessments..

Acknowledgement

I, Tan Yee Han would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

I, Tan Yee Han declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

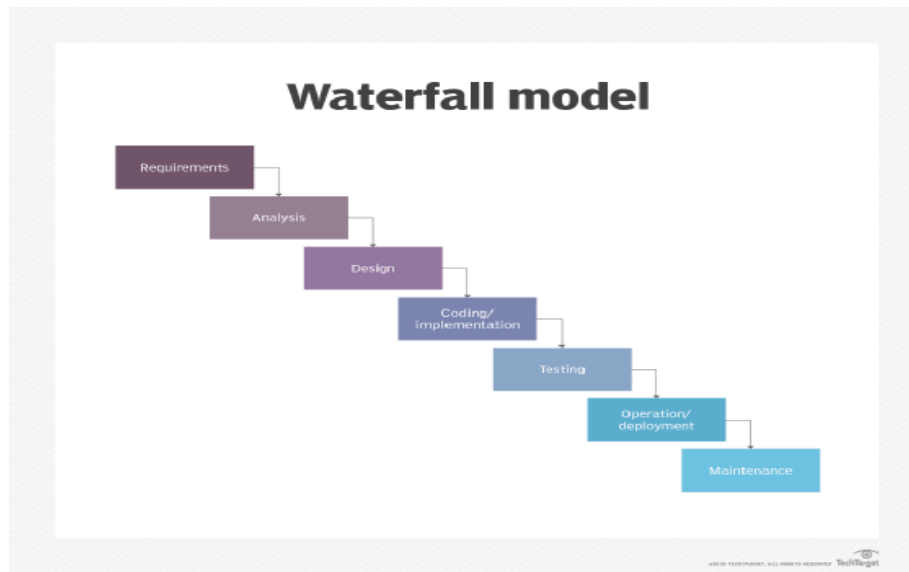
This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

I, Tan Yee Han confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

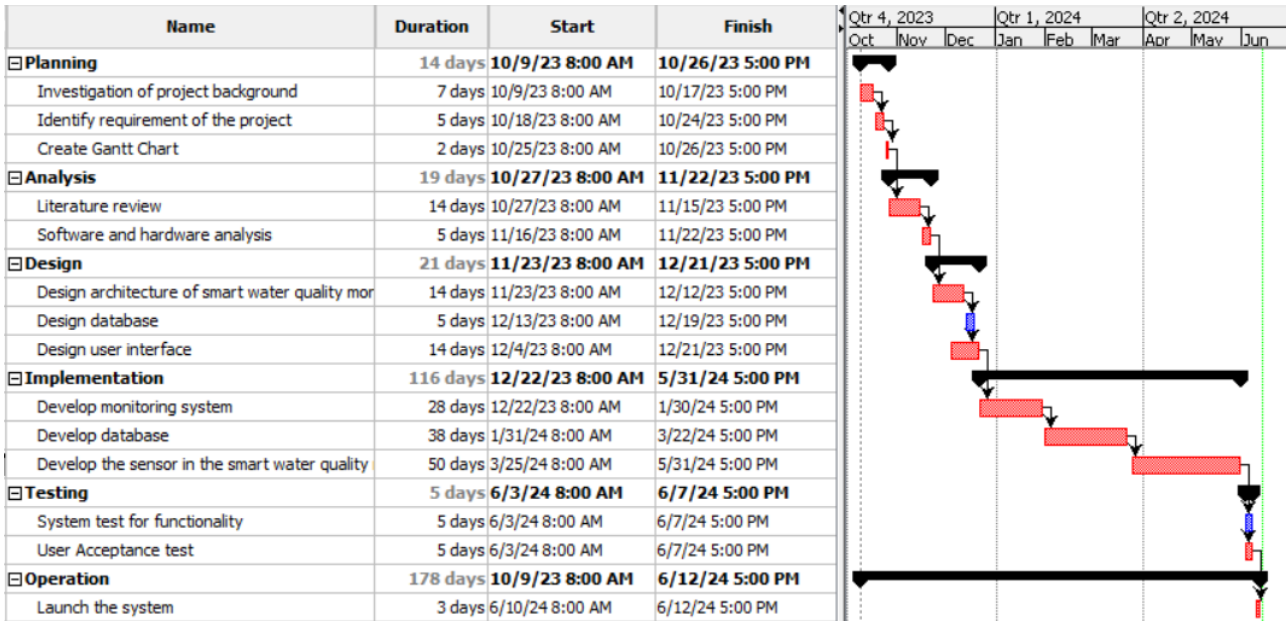
- [1] [WATER QUALITY definition and meaning | Collins English Dictionary. In: Collins Dictionaries.; 2023.](https://www.collinsdictionary.com/dictionary/english/water-quality)
<https://www.collinsdictionary.com/dictionary/english/water-quality>
- [2] [Water R. The role of water in agriculture and food production in Malaysia and SEA. Published April 5, 2023.](https://www.linkedin.com/pulse/role-water-agriculture-food-production-malaysia-sea-rivil)
<https://www.linkedin.com/pulse/role-water-agriculture-food-production-malaysia-sea-rivil>
- [3] Kopetz, H., Steiner, W. (2022). Internet of Things. In: Real-Time Systems. Springer, Cham.
https://doi.org/10.1007/978-3-031-11992-7_13
- [4] Apure, "What Is pH Probe and How It Works? - Apure," *Apure*, Sep. 01, 2022.
<https://apureinstrument.com/blogs/what-is-ph-probe-and-how-it-works/>
- [5] H. L. . Johar, S. Mohamad, S. Mohd Shah, and R. Mohd Hanifa, "Water Quality Monitoring and Controlling Using IoT", *jeva*, vol. 2, no. 1, pp. 20–25, Jul. 2021.
- [6] A. A. S. Chowdhury, Y. Arafat and M. S. Alam, "IoT-GSM Based Controlling and Monitoring System to Prevent Water Wastage, Water Leakage, and Pollution in the Water Supply," *2022 International Conference on Innovations in Science, Engineering and Technology (ICISSET)*, Chittagong, Bangladesh, 2022, pp. 567-572, doi: 10.1109/ICISSET54810.2022.9775876.
- [7] R. Singh, M. Baz, A. Gehlot, M. Rashid, M.Khurana, S. Vaseem Akram, S. S. Alshamrani and A. Saeed AlGhamdi, "Water Quality Monitoring and Management of Building Water Tank Using Industrial Internet of Things," *Sustainability*, vol. 13, no. 15, p. 8452, Jul. 2021, doi: <https://doi.org/10.3390/su13158452>.
- [8] B. Lutkevich, "What is waterfall model? - Definition from WhatIs.com," *SearchSoftwareQuality*, 2019.
<https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>

Appendix A: Waterfall Model

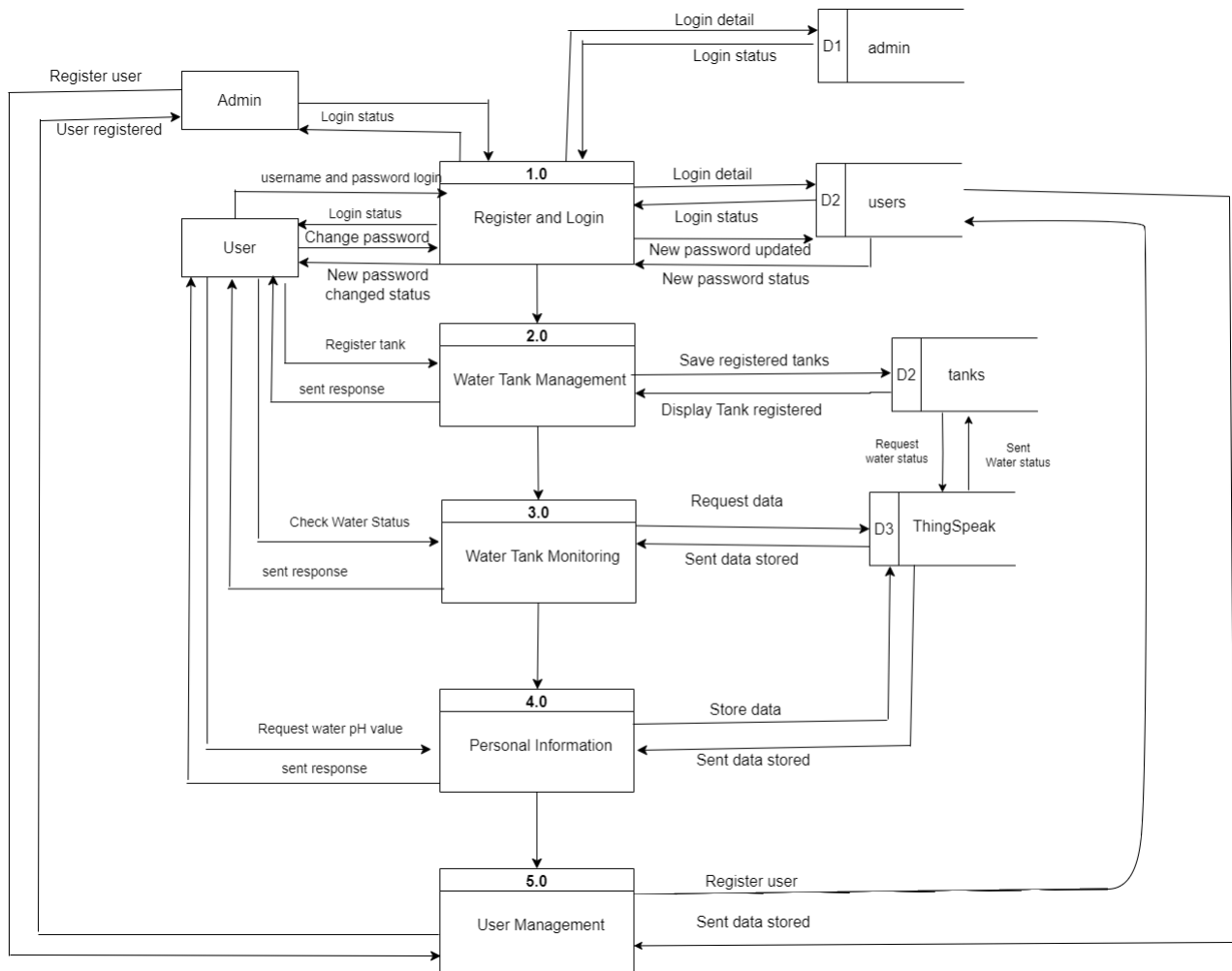


Waterfall model [8]

Appendix B: Gantt Chart

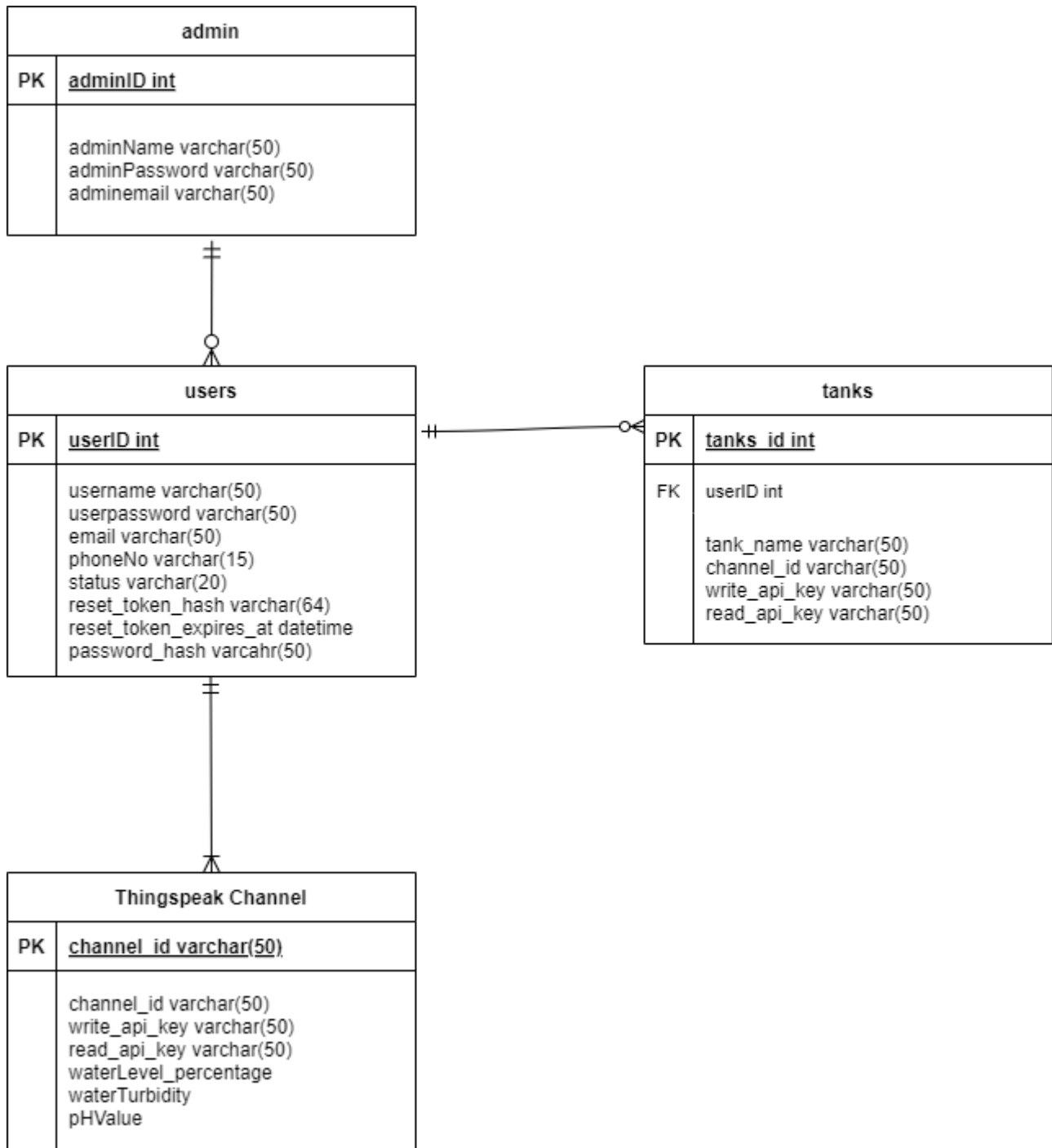


Appendix C: Data Flow Diagram Level 0



Data Flow Diagram Level 0

Appendix D: Entity-Relationship Diagram



Entity-Relationship Diagram