WATER QUALITY MONITORING SYSTEM USING WATER QUALITY SENSOR

NUR SYAFIQAH NABILAH BINTI CHE MOHAMAD

Bachelor of Computer Science (Computer System & Networking) With Honor

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	NUR SYAFIQAH NABILAH BINTI CHE MOHAMAD
Date of Birth	:	30/11/2000
Title	:	Water Quality Monitoring System Using Water Quality Sensor
Academic Session	:	Semester 2 2023/2024

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang Al-Sultan Abdullah reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang Al-Sultan Abdullah
2. The Library of Universiti Malaysia Pahang Al-Sultan Abdullah has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

001130-03-0254

New IC/Passport Number
Date: 3 July 2024

(Supervisor's Signature)

WAN ISNI SOFIAH BINTI WAN DIN

Name of Supervisor
Date: 31/07/2024

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Universiti Malaysia Pahang Al-Sultan Abdullah,
Lebuh Persiaran Tun Khalil Yaakob,
26300, Gambang, Kuantan, Pahang.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

____
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, *Universiti Malaysia Pahang Al-Sultan Abdullah* with its copy attached to the thesis.

UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Science.

____

(Supervisor's Signature)

Full Name : WAN ISNI SOFIAH BINTI WAN DIN

Position : SENIOR LECTURER

Date : 31/07/2024

____

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang Al-Sultan Abdullah or any other institutions.

(Student's Signature)

Full Name : NUR SYAFIQAH NABILAH BINTI CHE MOHAMAD

ID Number : CA20094

Date : 30 October 2023

WATER QUALITY MONITORING SYSTEM USING WATER QUALITY SENSOR

NUR SYAFIQAH NABILAH BINTI CHE MOHAMAD

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Computer System and Networking) with honor

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

30 OCTOBER 2023

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to Ts. Dr. Wan Isni Sofiah bt Wan Din, my supervisor for my final year project, whose unwavering guidance, support, and readiness to address my inquiries have been invaluable. Ts. Dr. Wan Isni Sofiah bt Wan Din's expertise and dedication have significantly contributed to overcoming the challenges encountered during this endeavor. Her consistent monitoring of my progress and invaluable advice on crafting high-quality reports have been pivotal to the success achieved. My sincere appreciation goes to UMP for providing the essential facilities and support necessary for the completion of my project. Access to journal services, facilitated by EZPROXY UMP, has been instrumental in conducting comprehensive research, utilizing resources such as the Library Explorer, ScienceDirect website, and IEEE Explorer. The resources provided by UMP have significantly enhanced the quality of my work.

I am deeply indebted to my family for their unwavering support and cooperation throughout the challenging phases of my project. Their encouragement, understanding, and sacrifices have been instrumental in overcoming obstacles and persevering through this endeavor.I am immensely grateful to my friends and colleagues for their invaluable suggestions, assistance, and warmth throughout this project. Their unwavering input, motivation, and support have served as a constant source of strength and inspiration. To all those mentioned above and to anyone whose support and contribution may not have been explicitly stated, I express my deepest gratitude for being an integral part of this journey. Thank you all for your unwavering support and encouragement.

ABSTRAK

Projek ini bertujuan untuk mencipta sistem pemantauan kualiti air yang mesra pengguna menggunakan sensor yang murah. Peranti ini akan memantau secara berterusan faktor kritikal dalam badan air, termasuk pH, kepekatan oksigen terlarut, kekeruhan dan suhu. Matlamatnya adalah untuk membolehkan pengecaman pantas kejadian pencemaran air melalui pengumpulan data masa nyata dan analisis asas, membolehkan tindakan segera untuk melindungi keselamatan air dan kesihatan alam sekitar. Sistem yang dicadangkan menekankan kesederhanaan dan kebolehaksesan, menjadikannya sesuai untuk pelbagai pengguna , termasuk agensi alam sekitar, penyelidik dan penduduk tempatan. Ia akan mempunyai antara muka yang mudah untuk operasi yang mudah dan visualisasi data, membolehkan pihak berkepentingan mengesan perubahan kualiti air dengan cekap. Inisiatif ini bertujuan untuk mendemokrasikan pemantauan kualiti air dengan memanfaatkan penderia kos rendah dan teknologi yang mudah didapati, menggalakkan orang ramai untuk mengambil bahagian secara aktif dalam pengawasan alam sekitar. Kos rendah dan kemudahan penggunaan sistem menjadikannya alat yang berkesan untuk meningkatkan kesedaran orang ramai tentang kebimbangan kualiti air dan menggalakkan tindakan kerjasama untuk pengurusan dan pemuliharaan sumber air. Secara keseluruhannya, inisiatif ini membantu memajukan kaedah pemantauan kualiti air dengan menawarkan penyelesaian praktikal dan kos efektif yang boleh dilaksanakan dalam pelbagai konteks. Pelaksanaannya bertujuan untuk menggalakkan amalan pengurusan air yang mampan di samping melindungi kesihatan ekosistem akuatik untuk generasi akan datang.

ABSTRACT

This project aims to create a user-friendly water quality monitoring system using inexpensive sensors. The device will continually monitor critical factors in water bodies, including pH, dissolved oxygen concentration, turbidity, and temperature. Its goal is to enable fast identification of water contamination occurrences by real-time data collecting and basic analysis, allowing for prompt action to protect water safety and environmental health. The suggested system emphasizes simplicity and accessibility, making it appropriate for a wide range of users, including environmental agencies, researchers, and local populations. It will have a simple interface for easy operation and data visualization, allowing stakeholders to efficiently track water quality changes. This initiative aims to democratize water quality monitoring by leveraging low-cost sensors and easily available technologies, encouraging people to actively participate in environmental stewardship. The system's low cost and ease of use make it an effective tool for raising public awareness of water quality concerns and encouraging collaborative actions for water resource management and conservation. Overall, this initiative helps to advance water quality monitoring methods by offering a practical and cost-effective solution that can be implemented in a variety of contexts. Its implementation seeks to encourage sustainable water management practices while also protecting the health of aquatic ecosystems for future generations.

.

TABLE OF CONTENT

LIST OF TABLES

No table of figures entries found.

LIST OF FIGURES

No table of figures entries found.

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

SBPWM	Simple Boost Pulse Width Modulation
ZSI	Z source inverter

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The integration of Internet of Things (IoT) technology in water quality monitoring has ushered in a new era of environmental awareness and resource management. This technological synergy promises to revolutionize society's understanding of water quality by enabling real-time, remote and accurate data collection through water quality sensors. In this context, I try to explore the importance of IoT-based water quality monitoring, examine its application domain and shed light on related issues. Water pollution has been a serious concern in recent years. Considering water is crucial to humans, constantly water monitoring will be necessary in dealing with this issue. The aim of the project is to create an effective IoT system for monitoring water quality in real time employing physiochemical sensors(Sung et al., 2021)

Water quality monitoring is a critical aspect of environmental science and resource management. It involves continuous assessment of physical, chemical and biological parameters in water bodies to measure their suitability for various purposes, such as drinking, irrigation, industrial use and ecosystem health. Traditionally, water quality assessment has relied on periodic manual sampling and laboratory analysis. However, this approach has limitations in terms of frequency, real-time response and coverage. IoT technology has bridged this gap by allowing continuous data collection through water quality sensors used in various aquatic environments, such as rivers, lakes, reservoirs and even wastewater treatment facilities.

The importance of IoT-based water quality monitoring cannot be disputed. Clean and safe water is a basic human need, and its availability has profound implications for public health, agriculture, industry and ecosystems. With the increasing challenges posed by climate change, pollution and population growth, the need for efficient real-time monitoring has become paramount. IoT-based water quality monitoring ensures timely detection of pollution, enabling rapid response and mitigation measures.

Despite the promise of IoT-based water quality monitoring, several issues and challenges persist. Some of these include sensor calibration and maintenance, data accuracy and reliability, data privacy and security, and the cost of implementing and maintaining an IoT network. Sensor calibration is essential to ensure accurate measurements, and periodic maintenance is essential to prevent sensor drift and failure. The accuracy and reliability of data is important for decision making, as incorrect data can lead to incorrect conclusions and actions. Data privacy and security concerns arise when sensitive environmental data is transmitted and stored digitally, potentially vulnerable to breach or tampering. Additionally, the cost of deploying and maintaining IoT networks, especially in remote or resource-limited areas, can be a limiting factor in the widespread adoption of this technology.

As an example of real-time water quality monitoring of water quality using temperature, pH, turbidity, conductivity, and dissolved oxygen sensors proposed by (Vijayakumar & Ramya, 2015)They use raspberry pi as controller, and they conducted an experiments on the internet using cloud computing to make it suitable for environment and ecosystem monitoring.

(Manoj et al., 2022)for water quality monitoring systems for fish ponds using IoT and physiochemical sensors. To verify and monitor the quality of water in a large area, in this project presents a sensor node measurement device mainly focused on monitoring quality of water. Their data collected into PMS database and displayed in real-time using a wireless sensor network.

1.2 Problem statement

Currently, the implementation of water quality monitoring systems in Malaysian rivers is inadequate in effectively addressing the escalating pollution from industrial and agricultural activities, as highlighted in recent research studies (Razman et al., 2023)The release of untreated effluents and the use of agricultural pesticides continue to impair river water quality, endangering ecosystems and human settlements. Therefore, there exists a compelling need for a more comprehensive and focused strategy in water quality monitoring. As a result, the lack of a comprehensive and effective Water Quality Monitoring System adapted to the characteristics of Malaysian rivers makes it difficult to correctly detect and quantify the effects of industrial and agricultural pollutants. This study seeks to close this essential knowledge and implementation gap by developing, deploying, and verifying an innovative monitoring system based on water quality sensors. By doing so, the study hopes to give a practical solution for successfully detecting and mitigating the presence of contaminants in river water, assuring its safety and acceptability for conversion into drinkable water. To provide a solution to this essential gap, this research project suggests implementing a Water Quality Monitoring System with an Arduino Uno microcontroller, TDS sensor, and other required components. The chosen technology will enable real-time monitoring and data collecting, with results visible on an LED display and accessible via the Blynk app. By taking this novel method, the study hopes to give a practical answer for detecting and mitigating the presence of contaminants in river water, assuring its safety and acceptability for conversion into drinkable water.

1.3 Objective

I. To detect water quality levels in water sources, distinguishing between clean and polluted sources, using a TDS meter connected to an ESP32 device.

II. To develop and implement a water quality monitoring system based on IoT technology, specifically utilizing TDS measurements for assessing water quality.

III. To evaluate and validate the accuracy of the developed IoT-based water quality monitoring system in differentiating between clean and polluted water sources based on TDS measurements.

1.4 Scope

The project aims to develop a comprehensive water quality monitoring system utilizing an ESP32 microcontroller integrated with Total Dissolved Solids (TDS) and temperature sensors. The system's primary objectives include designing a robust monitoring mechanism capable of accurately measuring TDS levels and temperature variations in water bodies. By deploying the system across 3-5 diverse locations, the project seeks to assess the variability in water quality across different environmental conditions. The system architecture will involve the utilization of ESP32's capabilities alongside TDS and temperature sensors, ensuring seamless data collection and transmission. The methodology will encompass the setup process for each sensor, including calibration procedures to ensure data accuracy. System testing will involve validating sensor readings against established water quality measurement techniques, ensuring reliability and accuracy under various environmental conditions. Deployment will entail selecting distinct locations for system installation, considering factors such as sensor placement and initial data collection for calibration. Data analysis will involve compiling and comparing data from different locations, identifying trends and anomalies in water quality variations. Results and discussion will focus on presenting findings, discussing implications, and evaluating the system's performance and limitations. The conclusion will summarize key findings, offer recommendations for further improvements, and assess the feasibility and effectiveness of the proposed water quality monitoring system.

1.5 Thesis Organization

The thesis is divided into five chapters, the first three of which deal on PSM 1. Chapter 1 discusses the topic and presents the project, emphasizing its contribution to resolving the stated issue. The chapter then describes the project aims before concluding with a summary of the project's scope and general thesis organization.

Chapter 2 digs into the literature study, which includes an overview of the current system's evolution. It highlights other projects, methodologies, and technology relevant to the current project and provides an overview of possible approaches. The chapter also discusses various learning methodologies.

In Chapter 3, the thesis describes the methodology used for project development. This chapter describes the programming or technology used to complete the project. It also includes information about the project design, which provides insights into the process and technical components of the execution.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The rising awareness of environmental sustainability and the need for safe and suitable for consumption, rivers and lakes has increased the need for effective water quality monitoring dramatically in recent years. The incorporation of Internet of Things (IoT) technology into water monitoring systems, namely those that use Total Dissolved Solids (TDS) sensors, has emerged as a potential approach.

Water quality monitoring provided by IoT delivers real-time data, improves accessibility, and enables rapid decision-making for water resource management. In this study, I am look at three noteworthy works that use TDS sensors to contribute to the field of IoT-based water quality monitoring. The primary use is to evaluate the relevance of existing systems and suggested methodologies to the goal of providing accurate and actionable water quality information.

2.2 Existing system/works

2.2.1 System A: IoT-based Water Quality Monitoring

This system was proposed by Sung W, Fadlillah F, and Hsiao S (Sung et al., 2021) that represents a significant advancement in environmental monitoring. Through the integration of Internet of Things (IoT) technologies, this system offers real-time and remote monitoring capabilities, providing a comprehensive and accurate assessment of water quality parameters. The utlization of sensors and data transmission mechanisms enables efficient data collection, processing and dissemination.

The study's findings emphasise the system's dependability in providing fast and exact water quality data, which contributes to proactive environmental management. The IoT framework's scalability and agility make it suitable for a wide range of environments, from industrial applications to natural water bodies.

Furthermore, including advanced analytics and visualisation capabilities improves the system's usability, allowing stakeholders to better comprehend and act on acquired data. The provided system serves as a baseline for future improvements in the field of water quality monitoring, highlighting the relevance of IoT in tackling modern environmental concerns.

To summarise, Sung W, Fadillah F, and Hsiao S's IoT-based Water Quality Monitoring system offers a reliable and effective approach for monitoring and regulating water quality. Its technical breakthroughs provide the groundwork for future innovation, providing a more sustainable and data-driven approach to environmental management.

Figure 1 depicts the system architecture employed in this investigation. The physical layer connects sensors to the network layer, which includes the Arduino UNO microcontroller and WiFi module. Sensors monitor the surroundings and collect sensory data. Data are securely stored in the cloud. The Wi-Fi module allows all devices to connect and access the internet. The network layer includes a microcontroller that enables all devices to function as one system. This microcontroller comprises of both hardware and software. Sensing data are transmitted to the Arduino UNO microcontroller for processing. ThingSpeak, accessible by personal computer or mobile phone, displays all sensor readings.

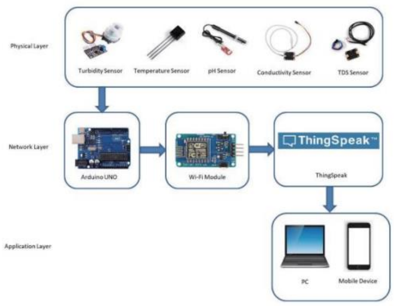


Figure 1: system architechure

2.2.2 System B: Aquarium Water Quality Monitor with TDS Sensor & ESP32

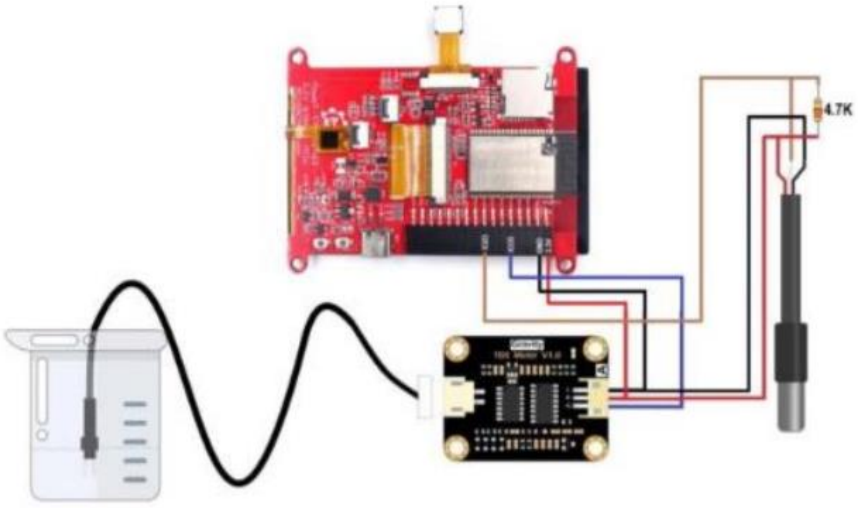


Figure 2: Water Quality Monitor with TDS Sensor & ESP32

A Total Dissolved Solids (TDS) sensor, an ESP32 microcontroller, and a built-in WiFi module are among the essential components used in the Aquarium Water Quality Monitor. The TDS sensor measures dissolved solids in water and outputs an analogue signal that corresponds to TDS levels. The ESP32 is the central processing unit, in charge of data collecting, processing, and transmission. The ESP32, when combined with a WiFi module, creates a wireless link for data transmission. The power supply, which is commonly provided by a USB port, delivers consistent power for continuous monitoring. Breadboards and jumper wires make component connections and prototyping easier. The circuit connects the TDS sensor to the ESP32 via the VCC, GND, and Signal pins. The built-in WiFi module of the ESP32 enables network communication. In operation, the ESP32 takes analogue data from the TDS sensor, transforms it to a TDS value, and sends it over WiFi to a cloud-based platform or server. Power management, online monitoring platforms, optional alarm systems, and watertight enclosures for protection are all important concerns. A sample Arduino code receives TDS readings and displays them on an OLED screen, laying the groundwork for further customization. Datasheets for components contain comprehensive specifications for pin connections and programming intricacies.

2.2.3 System C: IoT Based TDS Meter using ESP8266 for Water Quality Monitoring

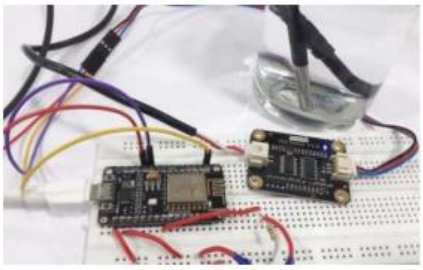


Figure 3: TDS Meter using ESP8266

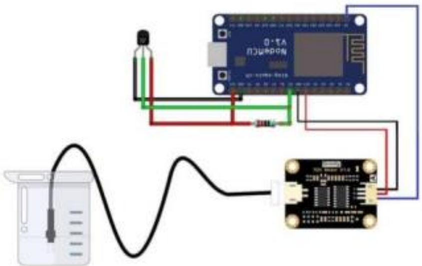


Figure 4: IoT Based TDS Meter using ESP8266 for Water Quality Monitoring

IoT-based TDS Meter using ESP8266 for Water Quality Monitoring involves integrating a TDS sensor and an ESP8266 module to enable real-time monitoring and data transmission. The TDS sensor measures the concentration of Total Dissolved Solids in water, providing an analog output based on dissolved ion concentration. The ESP8266, a low-cost, WiFi-enabled microcontroller, facilitates connectivity and data transmission.

The circuit connections include linking the TDS sensor to the ESP8266, and the ESP8266 to a local WiFi network for internet access. The operation entails the ESP8266 reading analog data from the TDS sensor, representing dissolved solids concentration, and transmitting it to a cloudbased platform or server through WiFi for remote monitoring. Additional considerations encompass power management, integration with online monitoring platforms, optional alert systems, and the potential use of waterproof enclosures.

2.3 Analysis/ Comparison of Existing system

2.3.1 Analysis of comparison on existing system.

Existing system	Method/Technology	Advantages	Disadvantages
IoT-based Water Quality Monitoring	The system employs IoT technologies for real-time water quality monitoring using sensors and an Arduino UNO microcontroller connected to a Wi-Fi module. Data is processed and securely stored in the cloud.	The system offers advantages such as timely data on water quality changes, remote accessibility, scalability for diverse environments, and enhanced usability through advanced analytics. It serves as an innovation baseline for future improvements.	The water monitoring system has issues: it needs a good internet connection and storing data in the cloud might not be super secure. Setting it up is expensive because of the special equipment, and you need to know tech stuff for maintenance.
Aquarium Water Quality Monitors	Integrates TDS sensors into ESP32-based systems for real-time monitoring.	combines real-time monitoring with potential remote access, leveraging the capabilities of the ESP32 microcontroller and WiFi connectivity.	The increased complexity may pose challenges in terms of setup and maintenance, potentially affecting reliability.
IoT-Based TDS meters using ESP8266	Employs ESP8266 microcontrollers for IoT connectivity in TDS monitoring systems.	This system strikes a balance between simplicity and wireless capabilities, offering a comprehensive solution for real-time monitoring and remote access.	While achieving a balance, it is essential to consider potential limitations in terms of power consumption, data transmission reliability, and overall system complexity.

Table 1: Analysis of comparison on existing system

Aspect	System A: IoT-based Water Quality Monitoring	System B: Aquarium Water Quality Monitor	System C: IoT Based TDS Meter using ESP8266
Features	Real-time & remote monitoring using IoT technologies, advanced analytics, and visualization.	TDS sensor, ESP32 microcontroller, WiFi module, USB power supply, breadboards, jumper wires.	TDS sensor, ESP8266 module, real-time monitoring, data transmission.
Strengths	Dependable, scalable, and agile IoT framework; Advanced analytics enhance usability.	Measures TDS levels with precision, wireless data transmission via WiFi, customizable with Arduino code.	Real-time monitoring of TDS levels, data transmission using WiFi, integration with IoT for remote monitoring
Weaknesses	Connectivity dependency, potential security concerns with cloud storage.	Requires technical expertise, potential power management and waterproofing concerns.	Potential for connectivity issues, limited functionality without WiFi, may need additional features.
Advantages	Comprehensive water quality assessment, suitability for various environments.	Precise measurement of TDS, wireless connectivity, customization with Arduino code.	Real-time monitoring, data transmission, integration with IoT for remote accessibility.
Disadvantages	Connectivity interruptions, potential security issues, high implementation costs.	Technical expertise needed, potential power and waterproofing concerns.	Connectivity issues, limited functionality without WiFi, may need additional features.

Table 2: Analysis of comparison on existing system

2.4 Summary

In summary, System A, suggested by Sung W, Fadlillah F, and Hsiao S, offers a significant development in environmental monitoring due to its IoT-based Water Quality Monitoring. It focuses on real-time and remote capabilities, scalability, and the incorporation of sophisticated analytics. System B is an aquarium water quality monitor that uses a TDS sensor and an ESP32 microprocessor to measure dissolved solids. It has a wireless data connectivity, a USB power supply, and a focus on component connections and prototyping. System C is an IoT-based TDS metre that uses the ESP8266 for real-time monitoring and data transfer of Total Dissolved Solids in water. It features a TDS sensor, an ESP8266 module, and access to a local WiFi network. Each system demonstrates distinct approaches, with System A emphasizing environmental monitoring, System B tailored for aquariums, and System C focusing on TDS measurement in water quality monitoring.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will go over the methods used in this project to achieve the proposed project's aims, which are to build a water monitoring system using water quality sensors and the Blynk app. The agile methodology has been chosen as the project development technique, which encompasses the specifications, planning, creation, testing, implementation, and revision stages. Aside from that, it will teach you how to create an initial version and which equipment and programmes to use.

3.2 Project Management Framework/Methodology

Figure 5: Agile methodology

Based on the study of the surface, the agile method was chosen as the best strategy for this project. In compared to other coding structures, the agile model is the most effective method for arranging projects, framework, and improvement. This is due partially to the agile the model's adaptability in regarding time and budget, as well as its ability to rapidly establish and demonstrate features. To reduce hazards, the water quality monitoring system may be modified and improved during the Agile model's implementation phase. Figure 5 depicts the agile phase.

3.2.1 Plan/requirement phase

Throughout the requirement stage, the proposed project for a water monitoring quality system employing water quality sensors seeks to enhance technology by replacing Raspberry Pi with an Arduino Uno that uses gadgets as well as identify water quality levels using TDS and turbid sensors. Aside from that, the system will send warnings to the Blynk apps. This project uses two types of notifications since the Blynk software need internet access to convey warnings; if the user does not have online access, the message may be shown on the user's LCD. With the development of the programme, it will be easy for the administrator to determine the quality of the river. As a result, the admin will get to monitor water quality more flexibly and efficiently compared to the previous technology test project.

3.2.2 Design phase

During the design phase, this project will start by understanding the flow chart, diagrams for use cases, and context diagrams for the system's connections to users. The flow graphic shows how water quality monitoring works. The use-case diagram depicts how users interact with the system. This system is utilised by people in charge of managing and regulating the water inspection system. Users, also known as actors, may carry out tasks such as monitoring water quality, informing staff of the detection of certain metrics, and so on. The context diagram was created to show how information moves from the Arduino Uno to the system, along with data movement from the structure to the user or the opposite way around. Following that, a project prototype and interface design will be created, using the storyboard technique to portray navigation in the water quality monitoring system's interface. Prototype design will begin with planning and go via sketching to project production. During this stage, the optimum building approach will be selected.

3.2.3 Development phase

Development of the system will be focused throughout the process of building it. The proposed system would consist of stages like evolution. Writing code for a water quality monitoring system is one technique to put the proposed system into effect. The Arduino Uno will serve as the project's basic hardware platform, as well as the programming for how it runs, while Blynk will be used to develop the water quality monitoring system. This project supports three coding languages: C, C++, and in phyton. In addition, any system updates or improvements will be implemented during this time. The process of development and programming are among the endeavors' that will be carried out.

3.2.4 Testing phase

In the testing phase of the water quality monitoring system, users evaluate the system by testing water samples from lakes, rain, and tanks. The sensors monitor dissolved solids concentration and water pH, triggering notifications to the admin's smartphone and the SSD1306 LCD monitor upon detection. Upon receiving notifications, the admin assesses the water quality for aquarium use and human consumption, comparing results across the three water sources. If issues are identified, the system undergoes corrections in the development phase. After resolving problems, the admin retests the system to ensure functionality across all components, finalizing the comprehensive testing of the entire system.

3.2.5 Deployment phase

When an equipment has gone through testing and is ready to be prepared for end users, it enters the deployment phase. The user may utilise the device, and the first version will be tested during this development phase.

3.2.6 Review phase

The review phase enables the user to return from the stage of requirements to the deploying phase, outlining every process and error that will happen when the project is turned on. At the planning phase, replacing the Raspberry Pi device with an ESP 32 will cause minor changes to the project. If there are any problems when executing the project, the user can adjust throughout the design and development stages. Water quality monitoring warnings are given to consumers via the Blynk App. The notification delivery system will be implemented during the trial period.

3.3 Project Requirement

The user requirements will outline the requirements that the user has for the system. This project will also learn how the system's process interacts with users. This system includes an application that displays water quality data to the user. In this case, too, depending on the user's answer after using the system, it can enhance the user's needs.

3.3.1 Functinal requirement

- Admin should be able to get the notification of water quality data.
- Admin should able to monitor quality of water
- Obtaining info from a server to ensure water quality meets project criteria and recommendations..
- The data request can be generated instantly by the programme.
- Both administrators and users may access water quality information.
- The apps must deliver on-demand services.
- The application itself must not have any interface or compilation difficulties.

3.3.2 Non-functional requirement

- The programme requires an intuitive user experience for effective and fast monitoring and regulation of water quality data.

- The system should be user-friendly, ensuring that all users, including the administrator, can use it easily.

- The proposed solution should seamlessly interact with the Blynk application system for remote monitoring and control.

- During registration of an account, users must generate secure passwords with at least 8 characters and a combination of alphabets, numbers, and symbols.

- For security reasons, only the administrator should be able to read water quality data

3.3.3 Limitations

- The ESP32 or any other components powering the water quality monitoring system require a continuous power supply for uninterrupted functionality.

- The project requires an secure internet connection for alert messages and facilitate remote surveillance.

- The application is designed for admin-only access, limiting viewing and management capabilities to authorized personnel.

- Interruptions to internet and electricity can impact the accuracy of real-time data collection.

- The system's testing phase is limited to specific types of water, such as river which may not fully represent all possible water sources and conditions. The system's accuracy and applicability may be influenced by the characteristics of these specific parameters types.

3.3.4 Constraints

- This project development incurs significant costs that must align with the allocated budget of water quality prototype.
- The project depends on the reliability of water quality sensors, emphasizing the need for careful handling and usage to maintain accurate measurement results.

3.4 Project Overview

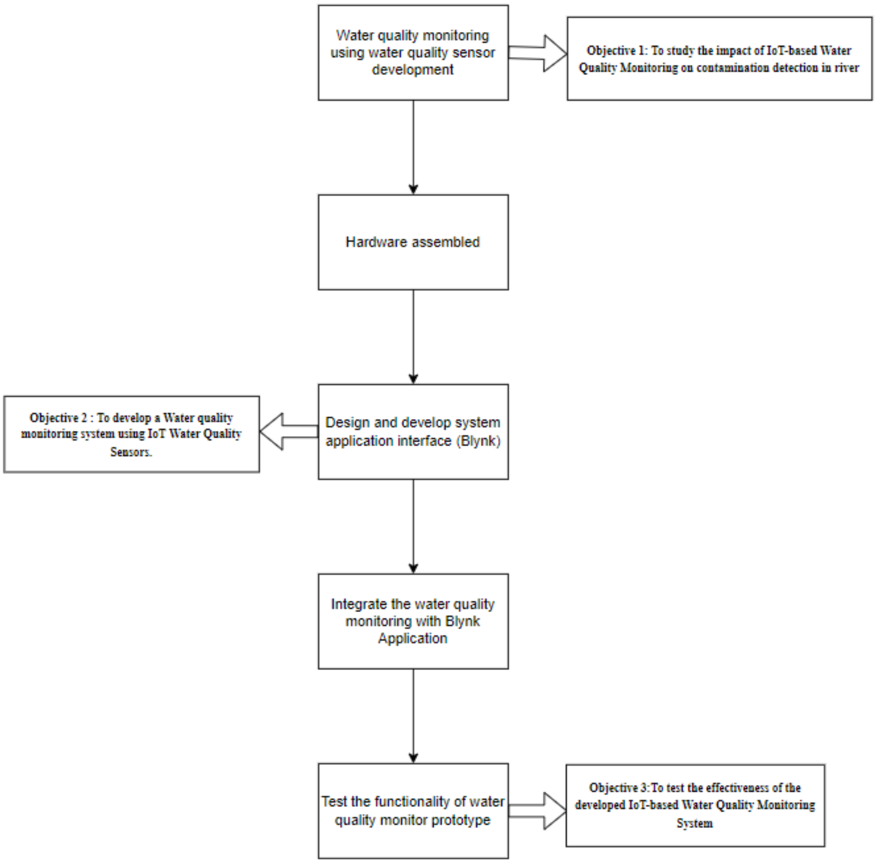


Figure 6: project overview

Figure 6 show the project overview for developing a water quality monitoring system. The first step is to assemble the hardware part. The assembling required electrical wiring works. The second step is to design the mobile application interface. Before designing the interface, a presentation model is produced where the model represents wireframe of the interface. In this step, third-party application, Blynk is used to design and develop the mobile application. The interface of mobile application designed is simple and user-friendly to smoothen the user experience. The third step is integration of Water quality monitoring system developed. Both hardware and software are required to work successfully in this step. The final phase is testing and evaluating the features of the water evaluation prototype. Before data is sent to monitor, a connection must be established between proposed prototype and mobile application. Next, data are sent to water quality monitor via mobile application to test the functionality. If data successfully displayed on monitor, the prototype is ready to be deployed.

3.5 Propose Design

The suggested design is crucial since it identifies the essential hardware and software to help define the whole project's structure. Flowcharts, use case diagrams, and context diagrams are every sort of diagrams. This system design is critical in that it serves as a road map for acquiring a better grasp of the overall project.

3.5.1 Flowchart

Figure 7 displays the water quality monitoring system's actual flow system. The processes of the water quality monitoring system are also highlighted, from turning on the AC power source to the worker checker receiving messages. A water-quality monitor system offers an inventive solution that will help to maintain a safe and healthy water environment. Figure 8 displays a flowchart for the login and sign-up operations. Only the admin user may add the new worker checker as a new user. When an admin user correctly registers their information, the worker checker may access water quality data. Figure 9 demonstrates how data is sent across the system.

TDS Level Chart for drinking water

TDS in Water (measured in PPM)	Suitability for Drinking Water
50-150	Excellent for drinking
150-250	Good
250-300	Fair
300-500	Poor (not good for drinking)
Above 1200	Unacceptable

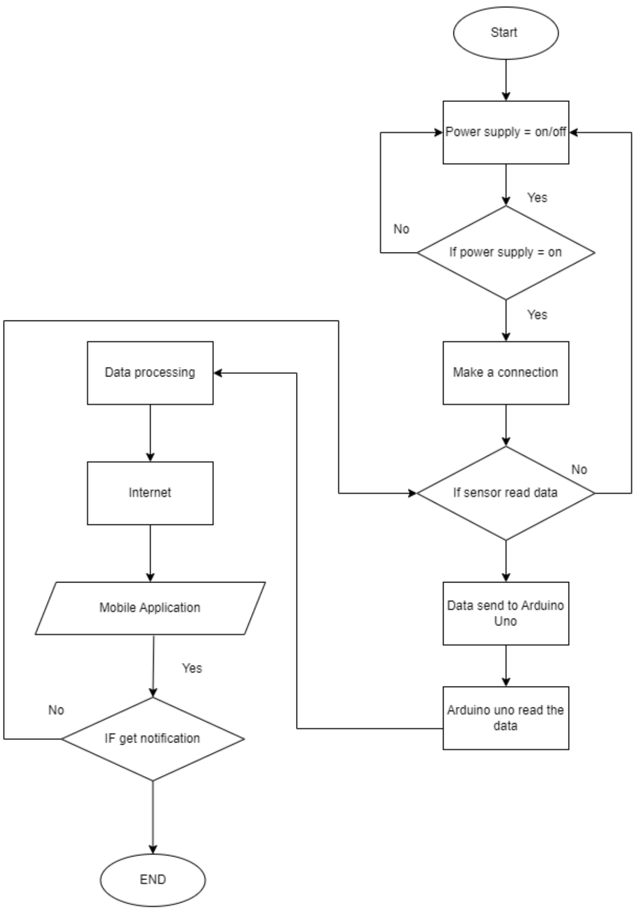


Figure 7: Flowchart of water quality monitoring system

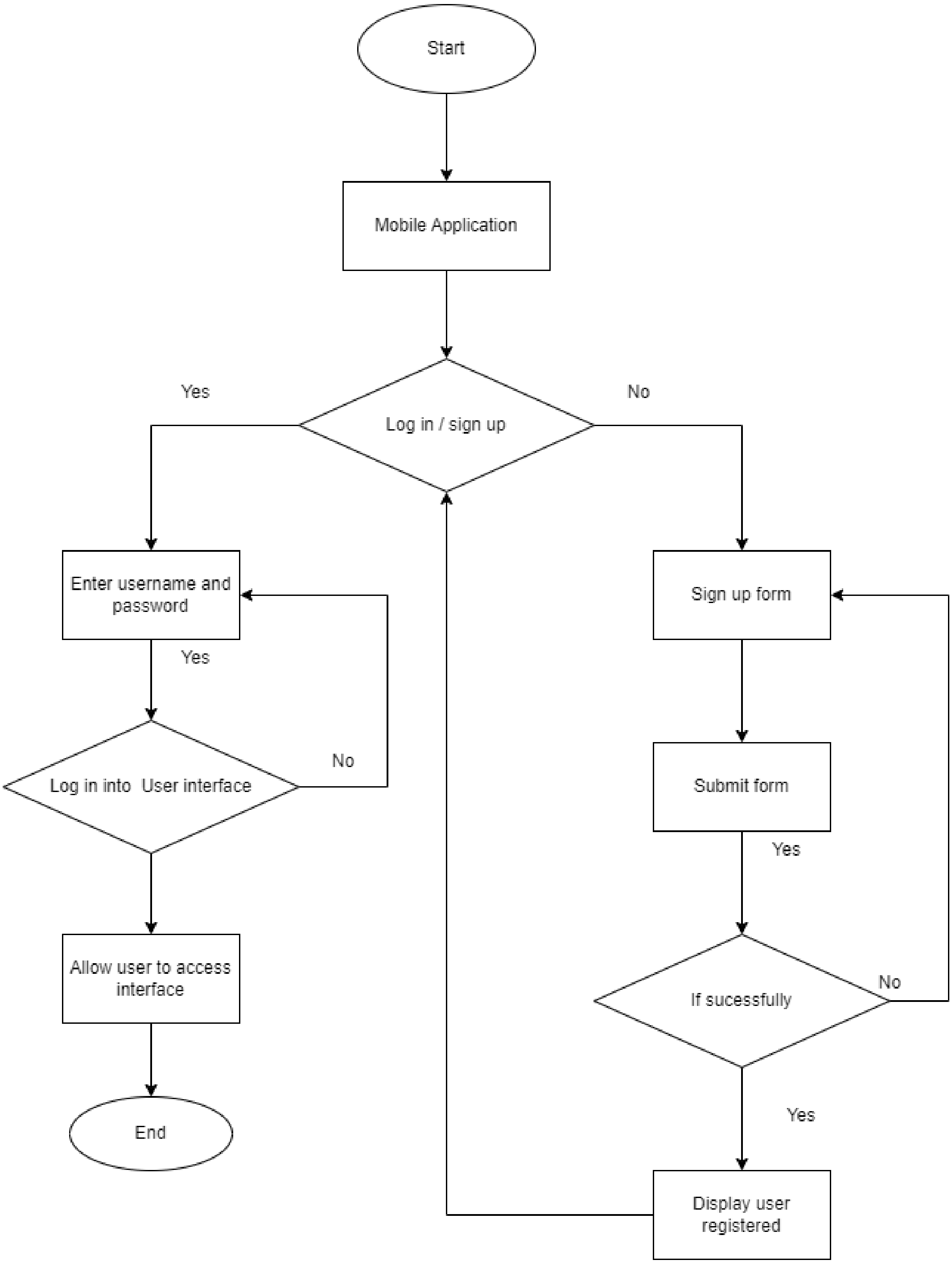


Figure 8: Flowchart of the water quality monitoring login system

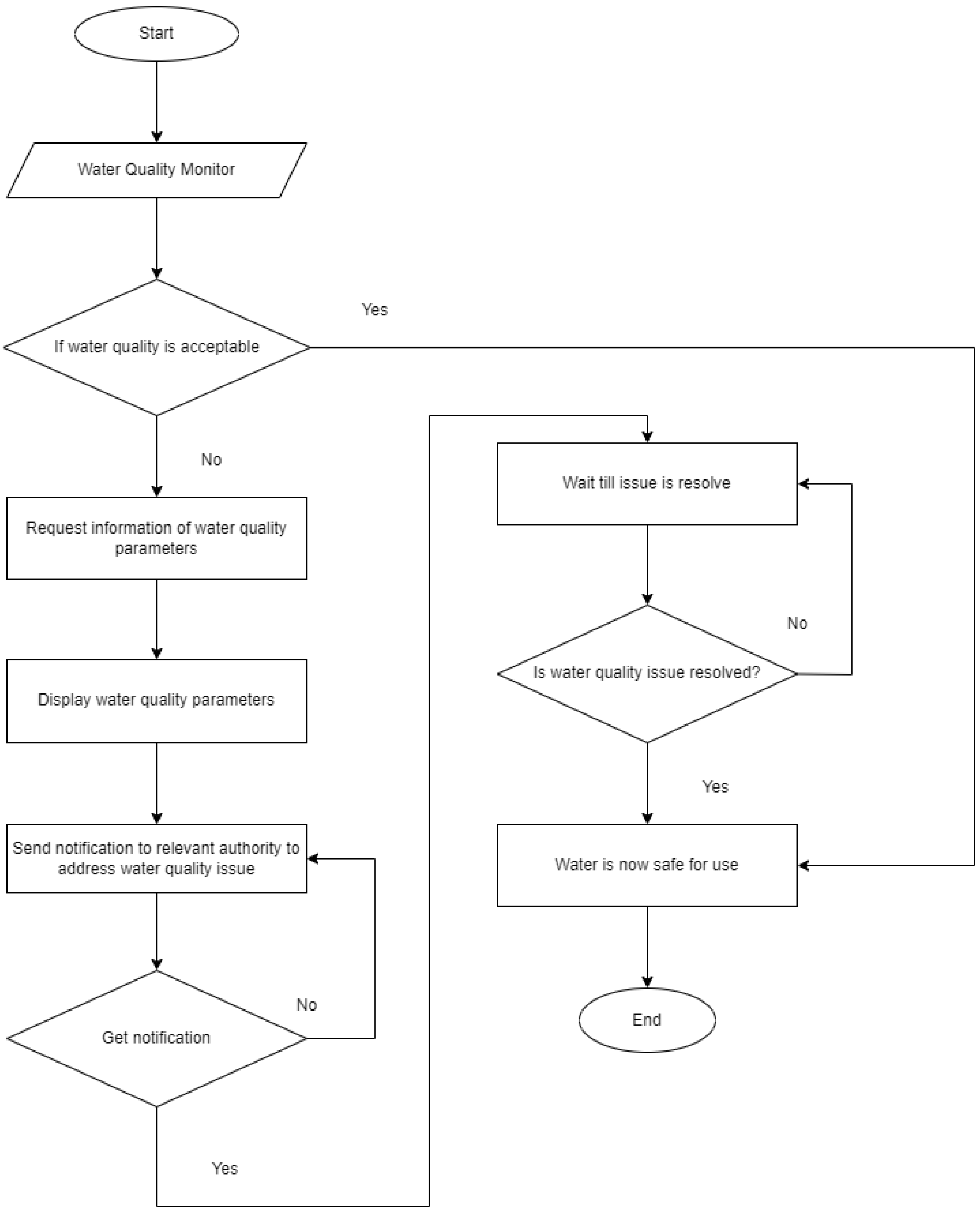


Figure 9: Flowchart of the system application

3.5.2 Use case diagram

A use case diagram is a form of behavioural diagram that is constructed and generated as part of the use case research process. Its objective is to offer a visual depiction of how a system works in terms of actors, objectives regardless of and any relationships between use cases. Figure 10 demonstrates how the system interacts with the actors and the use case. The use case outlines the activities that the actors may take. The administrator is in charge of system updates, water quality monitoring, user data management, notification management, and, if necessary, water quality management. Meanwhile, the worker checker may use the system, get notifications, and monitor water quality.

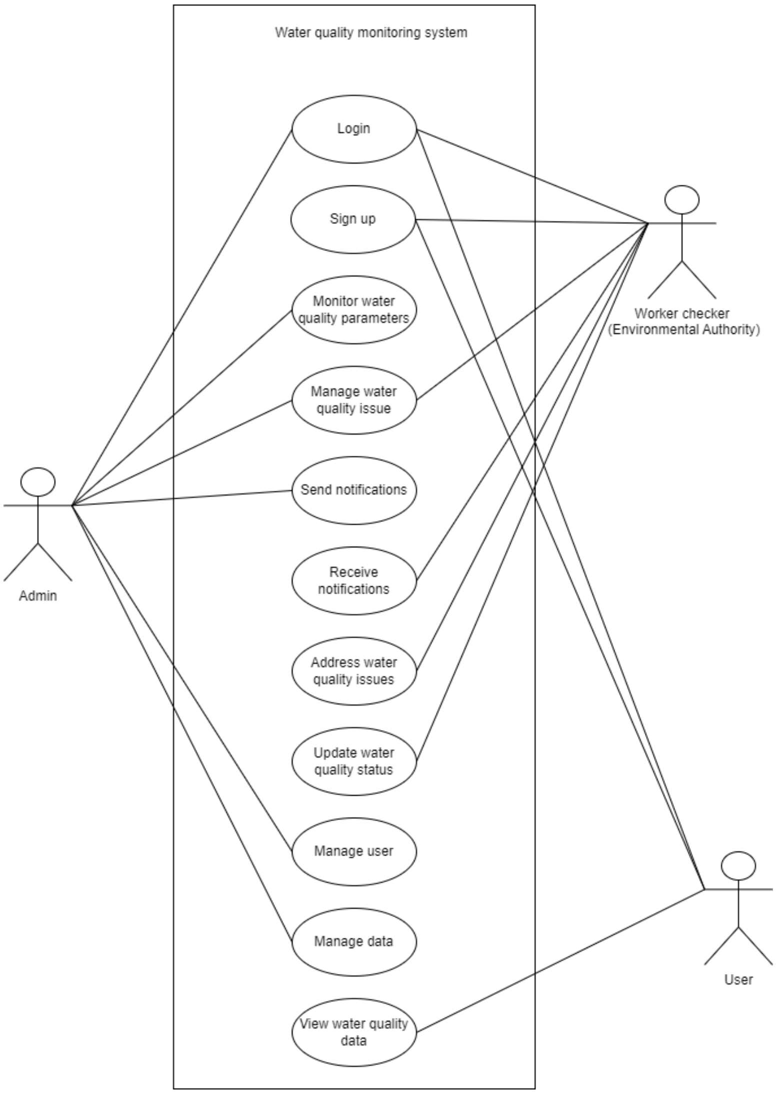


Figure 10: use case diagram

3.5.3 Context diagram

The context diagram depicts the system's context and restrictions, as well as the interaction between the system and external entities, through information flows. Figure 11 shows an overview structure for the idea for the system. External entities include users and the quality of the water. The arrows reflect the steps of a proposed system.

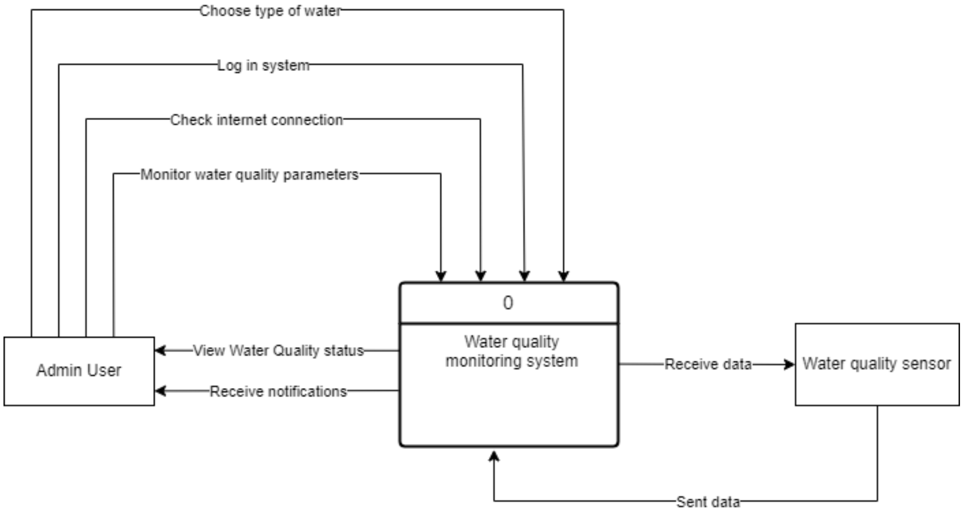


Figure 11: context diagram

3.5.4 Activity diagram

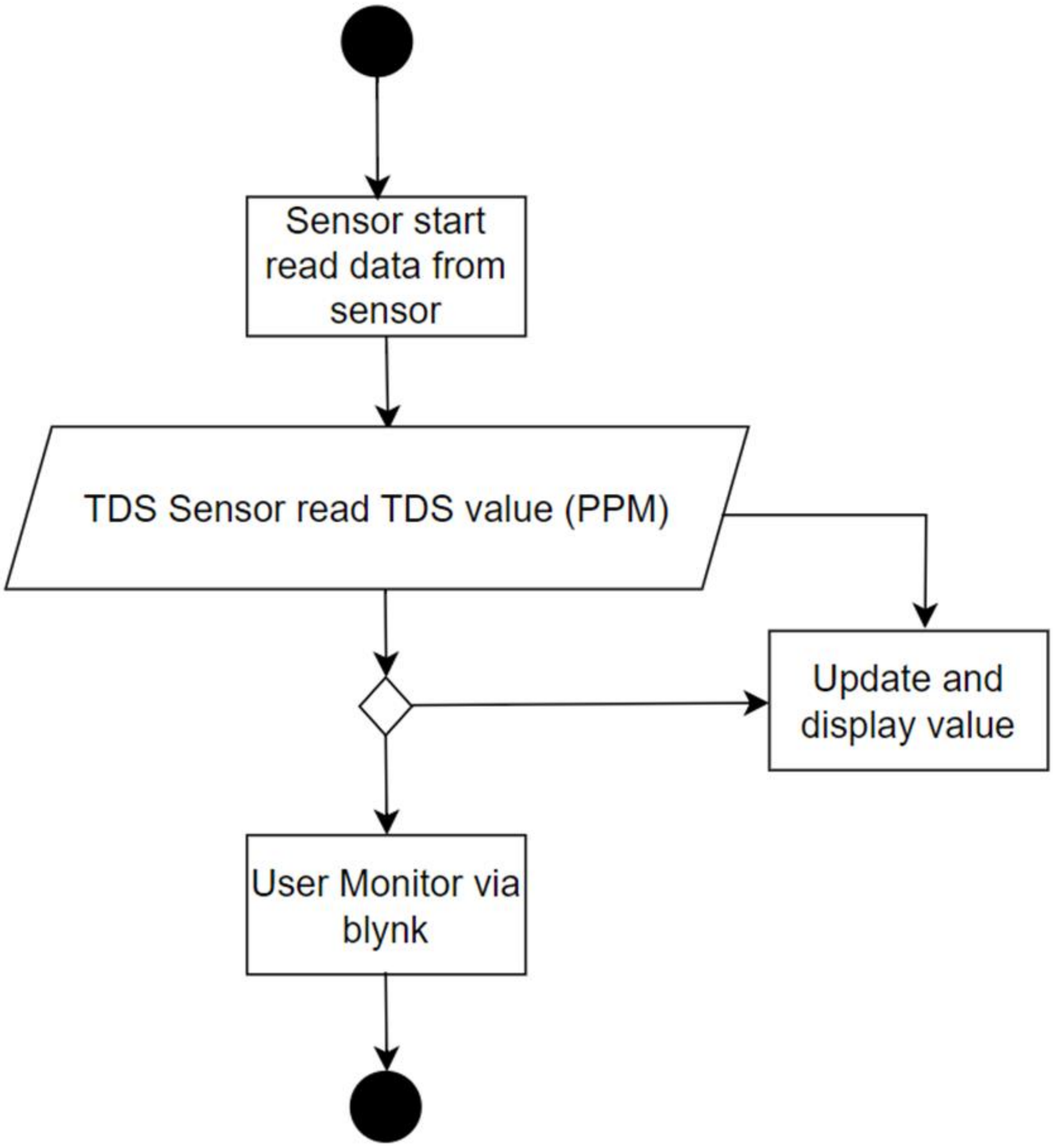


Figure 13: Hardware Activity diagram

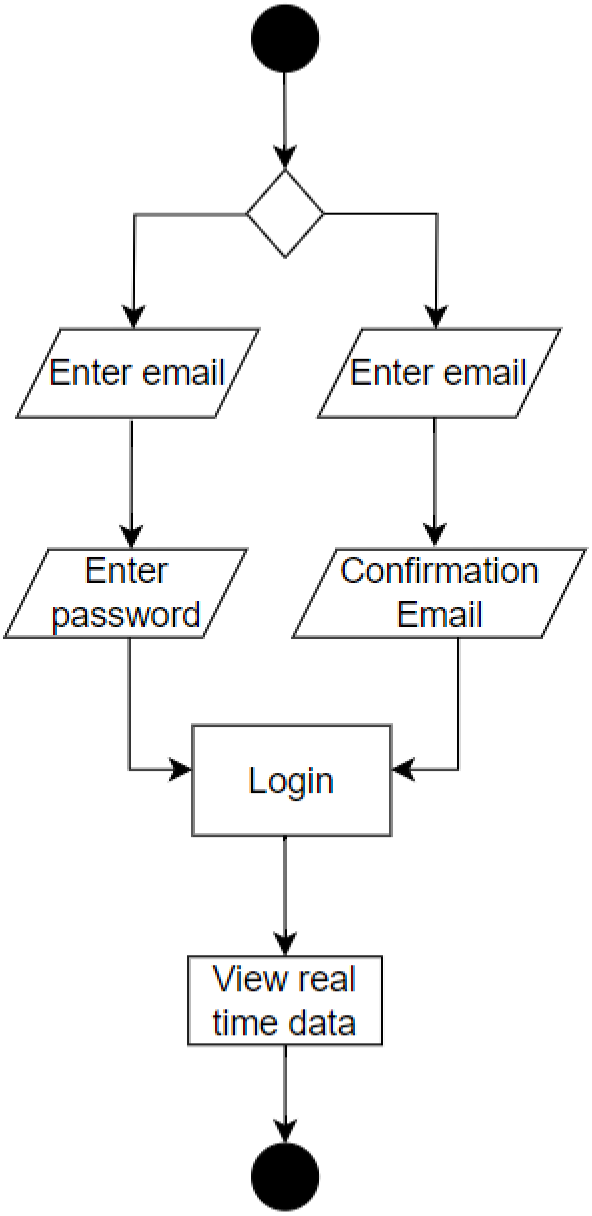
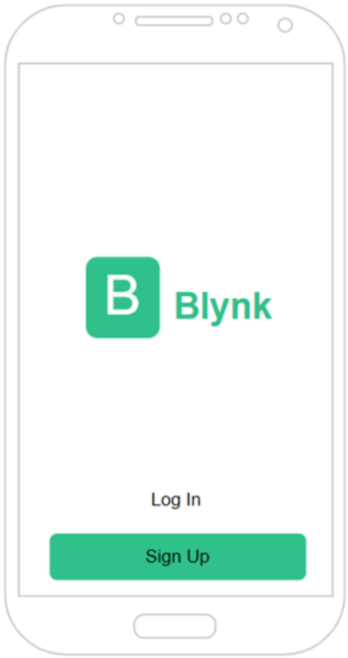


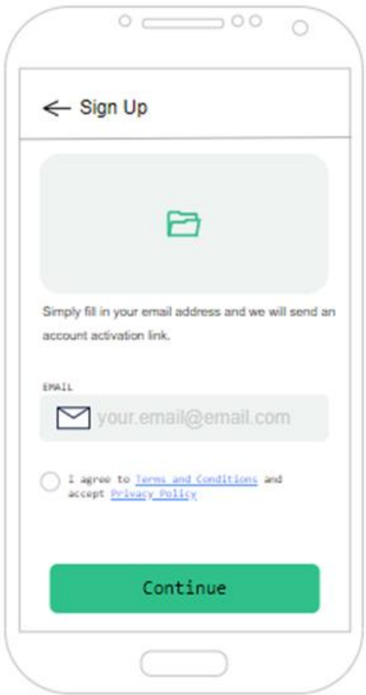
Figure 14: The system Activity diagram

3.5.5 Water Quality Monitoring System application storyboard



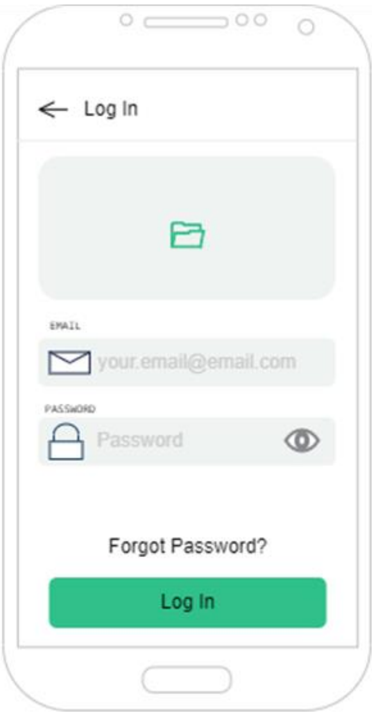
The user needs to enter the application Blynk and log in.

Figure 15: UI for log in/ sign up water quality monitoring app



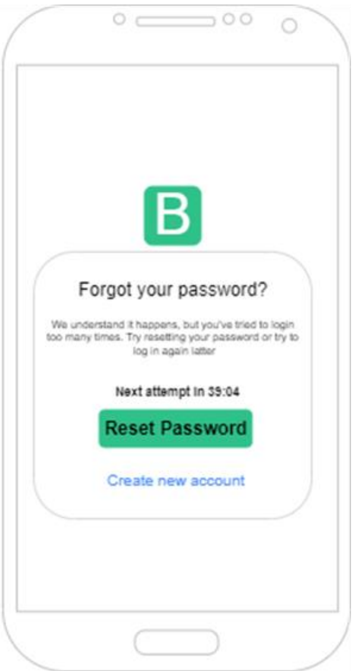
Only the admin user can register new users' details. The administrator will enter their email address and click the "agree to terms and conditions" button. Following that, click "Continue".

Figure 16: UI for sign up registration water quality monitoring app



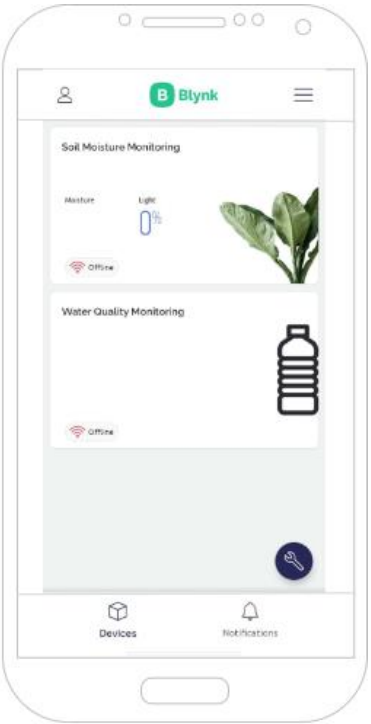
Before logging in, the user must provide both his or her username and password.

Figure 17: UI for Log in into water quality monitoring app



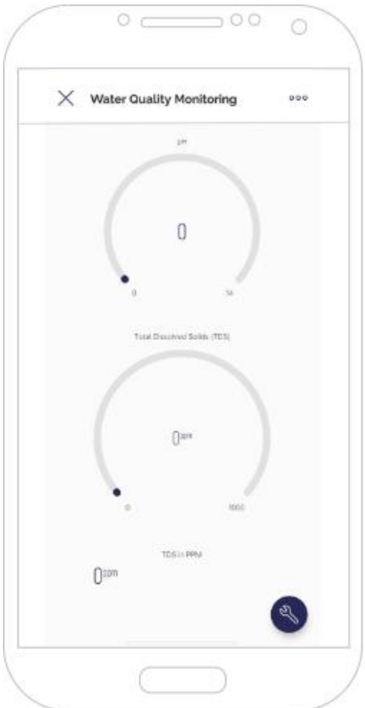
If the user enters the incorrect email address and password, the page "Forget your password?" is shown. The user can change their password or establish a new account.

Figure 18: UI for forgot password in water quality monitoring app



This is the application system's page. It will display the lists of monitors that are already registered in this development. Then, the user must select Water Quality Monitoring to monitor any sort of water.

Figure 19: UI for choose device to connect water quality monitoring app



After clicking the Water Quality Monitoring button, you will be sent to this page. Users must check water quality and obtain up-to-date information on the water. The output measurements will display here.

Figure 20: UI for main monitoring water quality

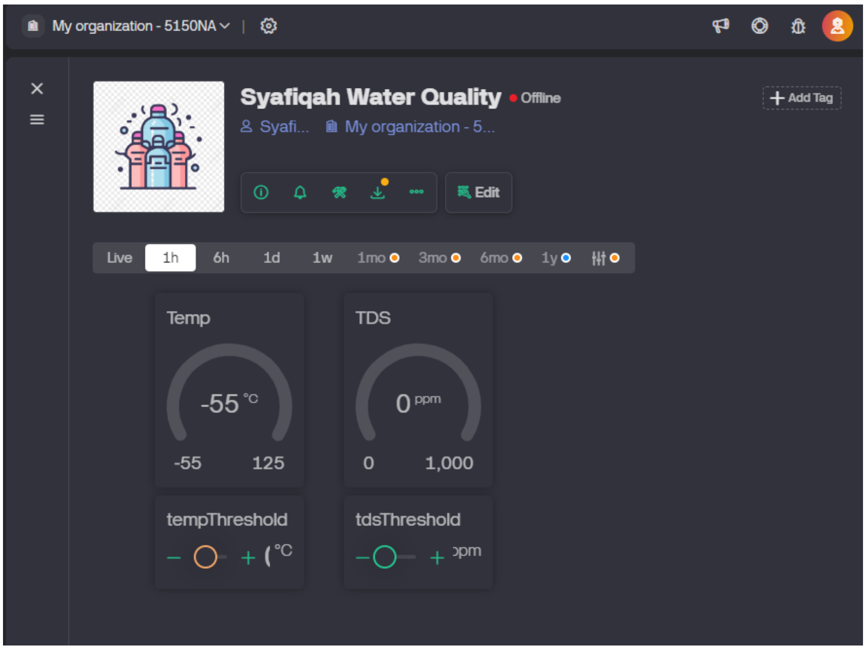


Figure 21: UI for main monitoring water quality in web

3.6 Data Design

As a visual representation of the data model, an entity-relationship diagram (ERD) lists the objects or entities in a system and shows how they are related to one another. This tool, which is commonly used in database design, accurately illustrates the database structure. The Water Quality system ERD system is depict in Figure 22.

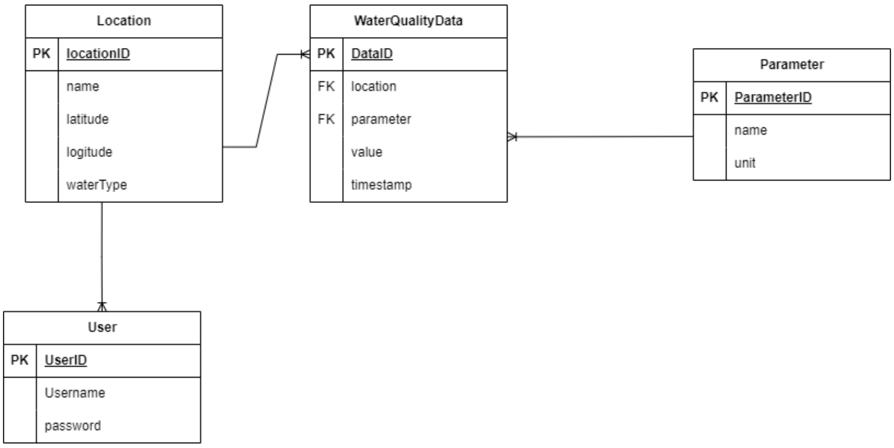


Figure 22: ERD

3.7 Proof of Initial Concept

3.7.1 Water Quality Monitoring system physical design project

Figure 23 displays the actual structure idea for a system to monitor water quality based on the Internet of Things (IoT). For this project, all you need is a container (water), an Arduino Uno, a TDS sensor, an adapter, and jumper wires. A jumper wire will link the TDS sensor to the Arduino Uno. After the setup and configuration processes are completed successfully, the installed TDS sensor will monitor the water quality. The data collected by the TDS sensor will be relayed to the Arduino Uno. The Arduino Uno will immediately transmit the received data to the admin user's cell phone.

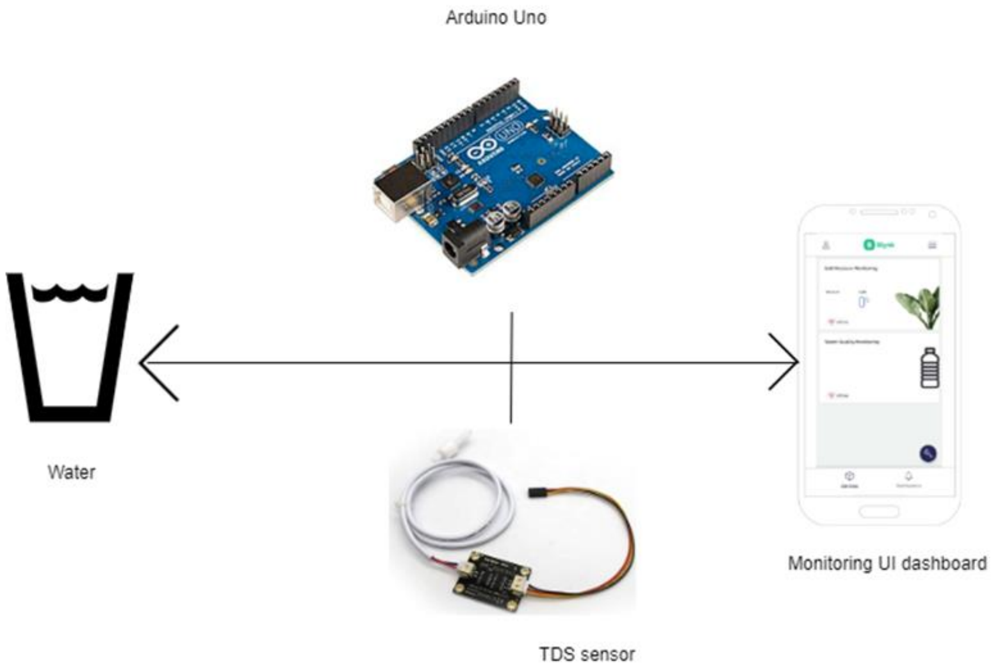


Figure 23: physical design of water quality monitoring

3.7.2 Hardware and software specification

This section gives a quick overview of the software and hardware utilised in this project, as well as the pricing and specs. The most important needs for this project are hardware and software. The project hardware can be an Arduino Uno, and the output device can be a TDS sensor for measuring quality. However, hardware cannot work without software. This project necessitates the development of hardware and software. Blynk is the required software that I use.

3.7.3 Hardware requirement

3.7.3.1 NodeMCU ESP 32

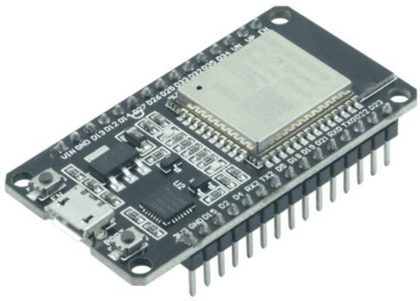


Figure 24: NodeMcu ESP32

The NodeMCU ESP32 is a microcontroller board that may be used to assess water quality by attaching many sensors to it. In this example, we'll utilise a TDS (Total Dissolved Solids) sensor to determine the concentration of dissolved solids in water. The ESP32 board has an analog-to-digital converter (ADC) that can read the voltage output of the TDS sensor. We'll apply a median filtering method to acquire a more consistent result from the sensor.

3.7.3.2 Arduino Uno

Figure 25: Arduino Uno

The Arduino Uno is a flexible microcontroller board that may be integrated into a Water Quality Monitoring System (WQMS) with ease. The Arduino Uno acts as the core processing unit in such a system, linking and handling different sensors that detect crucial factors of water quality. Sensors like as pH, turbidity, and conductivity sensors may be attached to the Arduino Uno to capture real-time data on the acidity, clarity, and conductivity of the water.

This data is processed by the Arduino Uno, which may be configured to send messages or activate water treatment systems depending on predetermined criteria. Furthermore, the Arduino Uno can help with data logging, which allows for the preservation and study of past water quality patterns. The Arduino Uno is an ideal choice for the core component of a Water Quality Monitoring System due to its small size, low cost, and ease of programming, enabling efficient and accurate monitoring of water conditions for a variety of applications ranging from environmental conservation to public health management.

3.7.3.3 TDS Sensor

Figure 26: TDS Sensor

A Total Dissolved Solids (TDS) sensor is critical in measuring the overall quality of water in the context of a Water Quality Monitoring System. The concentration of dissolved substances in water, including salts, minerals, and other organic stuff, is referred to as TDS. The TDS sensor detects water's electrical conductivity, which is related to the concentration of dissolved particles. Because water quality is frequently determined by the presence of certain ions and chemicals, the TDS sensor gives important information about the purity of the water.

The TDS sensor in a Water Quality Monitoring System aids in identifying variations in water composition, allowing the identification of possible pollutants or changes in water quality over time. The system may warn users to deviations from established water quality requirements by continually monitoring TDS ranges, allowing for rapid intervention and maintenance to assure the delivery of safe and clean water.

3.7.3.4 Waterproof DS18b20 temperature sensor

Figure 27: DS18b20 temperature

The Waterproof DS18B20 Temperature Sensor is a popular digital temperature sensor that finds usage in a variety of applications, including IoT projects, weather stations, and industrial automation. Its waterproof construction allows it to endure water and moisture without losing function, making it perfect for situations where the sensor must be submerged in water or subjected to high humidity. Maxim Integrated (previously Dallas Semiconductor) manufactures the DS18B20, a 1-wire device that requires just one wire for communication, making it simple to incorporate into designs. The DS18B20 temperature sensor has a range of -55°C to 125°C (-67°F to 257°F) with an accuracy of ±0.5°C (±0.9°F) between -10°C to 85°C (14°F to 185°F), making it a dependable and accurate choice for many applications.

3.7.4 Software requirement

The software used to develop this project is crucial for ensuring that the water quality monitoring system works and that the input and output are appropriately set. Table 3 lists the software that has been utilised.

Software	Function/Specification
C++ Programming	C++ is a general-purpose programming language. It has low-level memory manipulation capabilities, as well as imperative, object-oriented, and generic programming features.
Phyton	Python is a computer language that is widely used in scientific computing, data analysis, and web development. Python is another object-oriented programming language that allows us to define and operate with objects. It also contains a large standard library with a wide range of jobs and features, making it a strong and adaptable language.
Blynk	Blynk is an IoT platform that allows users to operate and monitor their gadgets remotely from their smartphone. It offers a suite of tools and services that make it simple for users to create linked applications for a variety of devices. Blynk also includes a mobile app for creating a visual interface for IoT projects, as well as a server component that manages communication between devices and the app. Blynk can operate a variety of devices, including sensors and others.

Table 3: software requirement

3.8 Testing/Validation

The testing phase is the most crucial. After the project has been built, testing is essential to ensure that every feature works as planned. Additionally, when completing functional and non-functional testing, ensure that the system meets the goal and goals. During this phase, it is critical to check for errors or defects in the system. If there are any more, the process will loop back to the initial step and take corrective action. All system components involved in testing are tested. After the problem has been resolved, the project is available for users to utilize.

- Testing the functionality of Water Quality Monitoring system using Arduino Uno.
- Testing the functionality of Arduino Uno with the TDS sensor.
- Monitoring to see if the TDS sensor is functioning properly.
- Testing the mobile application functionality before it is used in the Water Quality Monitoring system.
- Testing to ensure the mobile application will be able to detect the water quality parameters in the reservoir, whether it is optimal, suboptimal, or poor.
- The testing target is tested.

3.9 Potential Use of Proposed Solution

A prototype is a pre-production sample, model, or version of a product that was developed to test a theory or technique or to act as a model for replication or learning. It is a term used in a variety of domains, including semantics, design, electronics, and software development. A prototype is generally used by system analysts and users to evaluate a new design in order to improve precision. The Water Quality Monitoring System using Arduino Uno and TDS sensor has been applied through prototype design, and the prototype design has been produced to observe the outcome of this project.

3.10 Gantt Chart

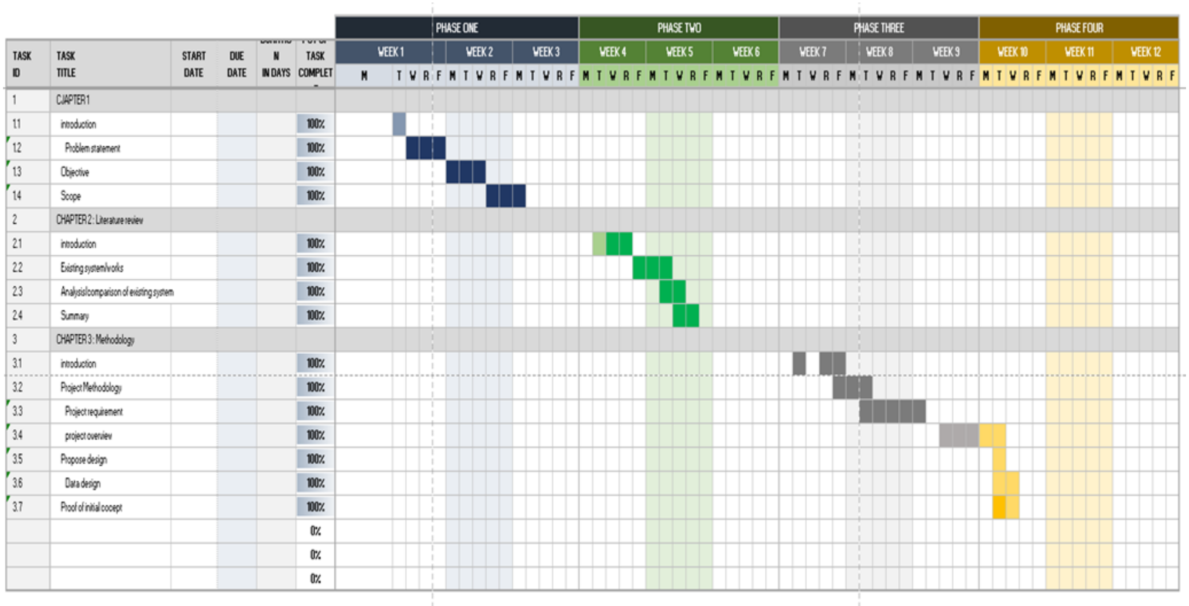


Figure 28: Gantt Chart

3.11 Summary

The project overview is vividly visualized in this chapter to demonstrate the evolution of the suggested prototype. All objectives are met, and the test process is depicted using a flowchart. As a result, the proposed flowchart will be utilized to create a prototype of a Water Quality Monitoring system. Furthermore, the hardware and software indicated will be employed in the development of the prototype. All recommended plans, such as a system flowchart, a list of hardware and software utilized, and a software development life cycle, will be followed in order to construct the prototype, which will be built using the Agile methodology. The proposed project, the Water Quality Monitoring system, will be carried out in accordance with the stated plan. The prototype of will be the expected product of effective preparation in this chapter.

CHAPTER 4
IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 Introduction

This chapter outlines the whole process of constructing a water quality monitoring system employing TDS and temperature sensors, as well as the tests and data acquired during the project system implementation process. The preceding chapter addressed agile methodology, which is employed in the process sequence of system implementation and testing. This chapter will offer a full overview of the system implementation processes, outcomes, and testing methodologies used to guarantee project objectives are achieved. Throughout the project's execution, programming languages such as C for the Arduino IDE are utilized in the hardware configuration. This chapter will also discuss project restrictions. The outcomes and conversations acquired through this project method aid in future planning and direction.

4.1.1 Development Environment

The system was built with the Arduino IDE and the Blynk programme. The Arduino IDE is a piece of software that allows you to write code, upload it, and observe sensor data via the serial monitor. The Arduino programming language was used to write the build code for interfacing with the NodeMCU ESP32 microcontroller with associated sensors, and sensor data was tracked using the serial monitor. Meanwhile, the Blynk Application was utilised to develop a smartphone app that monitors Total Dissolved Solids (TDS) levels in water and may be used to indicate water quality through alarm messages and user login. The NodeMcu ESP32 microcontroller board with WIFI was used to build a water quality monitoring system using TDS and temperature sensors. Arduino is a free, open-source electronics platform that includes basic hardware and software. It has a text editor, an output area, a toolbar, and upload/compile functionality. Its primary programming language is C, and it employs the GNU C compiler.

The Arduino IDE's build code connects with the NodeMcu ESP32 microcontroller and attached sensors. The completed code may be uploaded to the NodeMcu ESP32 microcontroller, defining its hardware functionalities.

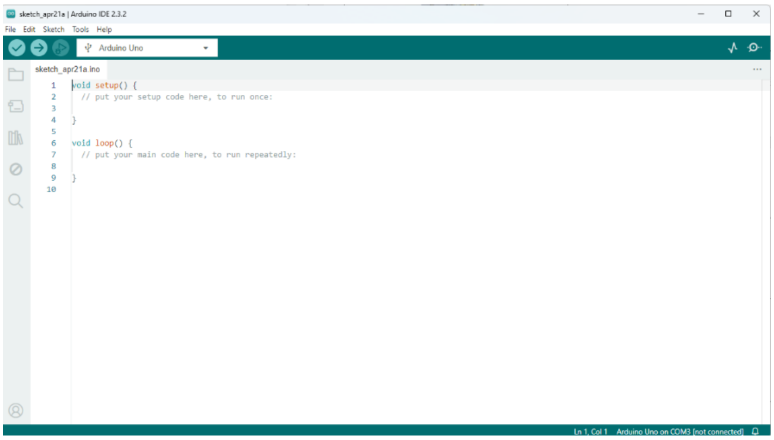


Figure 29: Arduino IDE

4.2 Mobile Application System

4.2.1 Login Interface

Figure 30 depicts the login interface's role, which is to confirm the user before giving access to the water monitoring system. The user must input the correct username and password that the developer supplied to them. Furthermore, if the user's login information does not match, they will be refused access to the system.

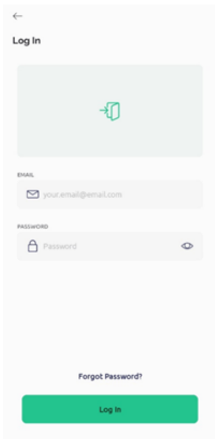


Figure 30: login interface

4.2.2 Home interface

If the user successfully logs into the application, it will redirect the user to the home interface in Figure 31, where the user can access application functions such as indicating how many parts per million of soluble solids are dissolved in one litre of water and receiving alert notifications if the TDS value (PPM) is higher, the more soluble solids are dissolved in water but the water is less clean. The entire system's role is to attain its aim. Unless the user presses the logout button, the system will stay logged in.

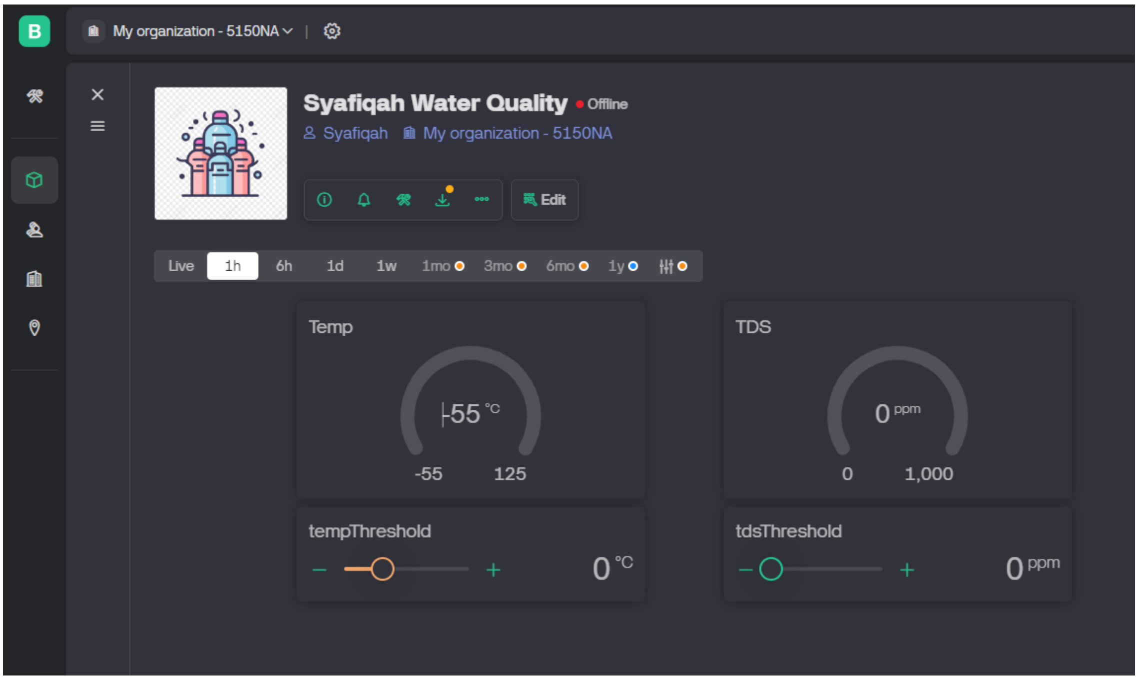
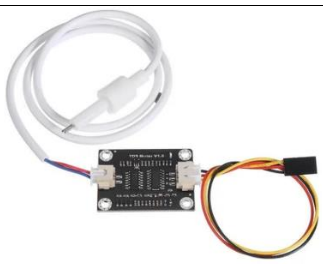


Figure 31: web UI of Water quality monitoring

4.3 Development Tools

Table 4 shows the tool that are used to develop the water quality monitoring system

	LCD 1602 with i2c module
	ds18b20 temperature sensor
	ESP32 30 PINS
	Total Dissolved Solids (TDS) sensor

	DC female connecter
	Jumper wires: male female
	crocodile clip wires

Table 4: list of components

4.4 Hardware Implementation

The microcontroller is the primary piece of hardware in this system. It functions as a controller on an integrated circuit, doing a specific duty. This system employs the Nodemcu esp32 microcontroller board, which has 30 digital input/output pins. The features enable the connection of two sensors: a TDS sensor, a temperature sensor, and a GSM module. The sensors include temperature and temperature differential sensors. This microcontroller features a Wi-Fi module that communicates sensor data to the Blynk server.

Sensors are connected to the Nodemcu esp32 microcontroller board using hardware such as a shield board, breadboard, and male to male and male to female jumper wires. The hardware must be installed first. The sensors and components are then wired to a breadboard to communicate with the microcontroller. The code was sent to the Nodemcu esp32 via USB. Debugging was done using a computer connection and a serial monitor.

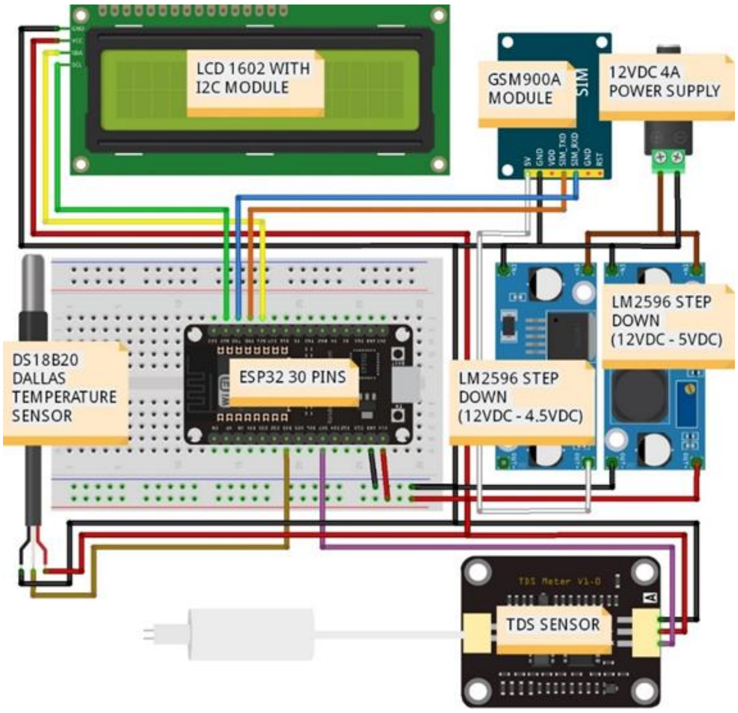
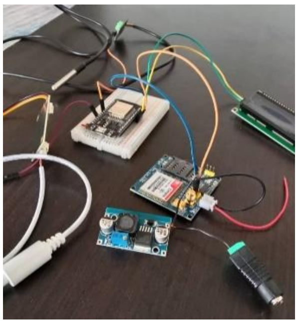
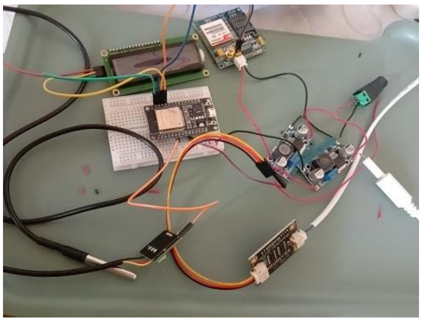
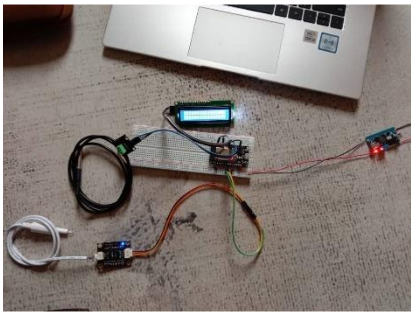
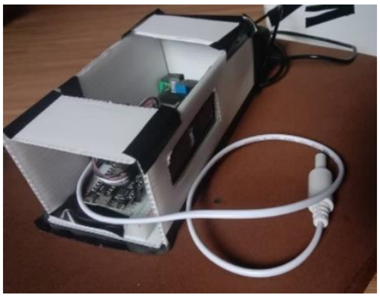


Figure 32: circuit diagram water quality monitoring

Figure 33: Output measure of TDS (Total Disolve Solid) sensor.

4.5 Evolution of development

	1st phase
	2nd phase
	3rd phase
	Final phase

4.6 Prototype of Water Quality Monitoring System

Figure 34 Prototype of Water Quality

4.7 Blynk Application

The Blynk programme was used to preserve the user's information from the mobile app, such as their login and password. The Blynk programme also stored sensor reading data on the NodeMCU ESP32 microcontroller. The communication between the Blynk application and the NodeMCU ESP32 was created via an internet connection and the appropriate setup parameters. The real-time readings from the TDS and temperature sensors were stored in Blynk.

4.8 Blynk Connection

Data retrieval from the Blynk server was necessary for function login and data presentation on the interface. As a consequence, before beginning data retrieval, connect to the Blynk server. The Blynk connection is written in C. The code in the box below represents the Blynk connect function. The Blynk function accepts the token's address, SSID, and password. The Blynk.begin() method validates the database connection.

```
#define BLYNK_TEMPLATE_ID "TMPL6xybMcdhg"
#define BLYNK_TEMPLATE_NAME "Water Quality Monitoring"
#define BLYNK_AUTH_TOKEN "KYtxzTgH4nJqahfhx68z9sddOauazRbj"
#define BLYNK_PRINT Serial

// User WiFi Credentials
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "syafiqah";
char pass[] = "cmkr9999 ";

// 9. Main Function and Setup
void mainFunction() {
    // Your main function code here
}

#define handleAll_INTERVAL 1000L
BlynkTimer HandleAll;

void setup()
{
  Serial.begin(115200);
  sensors.begin();

  WiFi.begin(ssid, pass);
  Blynk.begin(auth, ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }

  pinMode(pPin.tds, INPUT);

  Blynk.virtualWrite(Temperature, 0);
  Blynk.virtualWrite(TDS_Level, 0);
  Blynk.virtualWrite(tempThreshold, 0);
```

```
  Blynk.virtualWrite(tdsThreshold, 0);

  HandleAll.setInterval(handleAll_INTERVAL, mainFunction);

  lcd.begin();
  lcd.backlight();
}

void loop()
{
  Blynk.run();
  HandleAll.run();
}
```

4.9 Testing and Discussion

In this section, we will look at the testing and discussion of the water quality monitoring system. This includes both hardware and software testing to guarantee that the sensors and mobile apps work as planned to meet the system's goals. The sensors monitor numerous water quality characteristics, such as pH levels and pollution, which has a substantial influence on the system's performance. One important discovery is the monitoring system's price, which makes it more accessible to a larger range of users. The system's potential effect is enhanced by its affordability, which allows it to be implemented in a variety of contexts where water quality monitoring is required. Furthermore, the system offers users with rapid and trustworthy data output, allowing for prompt decision-making in water quality management. The testing step entails using the hardware prototype, mobile application, and internet connection to validate the system's performance. Several scenarios will be run to determine the system's accuracy and reliability in detecting and reporting water quality metrics. Additionally, the system will go through User Acceptance Testing to determine its general usability and efficacy. By extensively evaluating the water quality monitoring system, we want to assure its dependability and applicability for real-world applications, resulting in better water quality management methods.

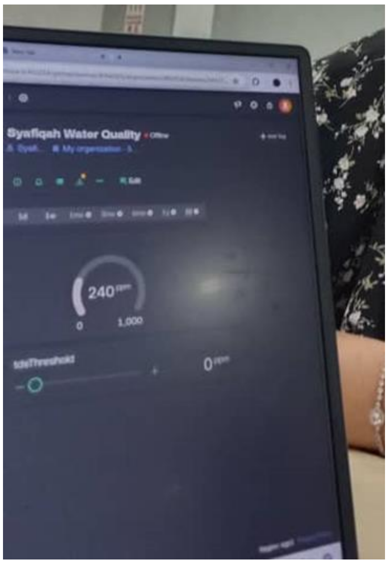


Figure 35 : Alert on Water Quality Monitoring System.

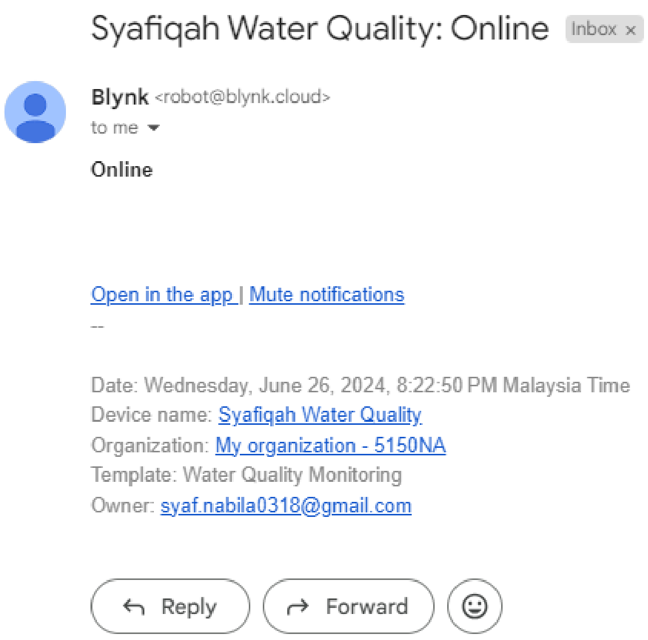


Figure 36: Online notification on Email.

4.9.1 Result on various River

NO	River	Type River	Parts per million & celcius reading
1		a river in the forest that passes through various types of tree roots	TDS value: n/a PPM value: n/a
2		River water that used to be a place where people threw garbage	TDS value: n/a PPM value: n/a
3		The river water on the hill that passes through the rocks and is the preferred place to bathe	TDS value: n/a PPM value: n/a

4.10 Conclusion

The water quality monitoring system was implemented by first constructing the essential components on a breadboard, which included an ESP32 microcontroller. Following that, we began developing mobile applications. After configuring the hardware and mobile application using minimal code, a connection to the Blynk server was created easily. Jumper cables simplified the connecting of sensors to the ESP32, eliminating the need for soldering. The consolidated sensor code was then uploaded to the ESP32 using the Arduino IDE. Concurrently, the mobile application was set up and launched. The setup with the Blynk server went well, making it easier to link the ESP32 to the mobile application. Testing protocols were carefully followed to ensure that the TDS sensor and temperature sensor worked properly. The ESP32 was configured to send data on TDS levels and temperature to a mobile application for real-time monitoring. This system seeks to monitor water quality in 3-5 various river areas, providing useful information on regional variances.

CHAPTER 5

CONCLUSION

5.1 Introduction

Water quality monitoring is an essential public health and environmental problem across the world. Keeping clean water is essential for human health, environmental balance, and sustainable development. Poor water quality can have serious consequences for human health and ecosystems. To solve this, a simple yet efficient Water Quality Monitoring System is presented for real-time monitoring of water quality at various sites along a river. The purpose of this project is to build a dependable water quality monitoring system utilising straightforward and cost-effective sensors. The system measures crucial water quality parameters using an ESP32 microcontroller, a TDS measure sensor, and a DS18B20 temperature sensor. The TDS measure sensor monitors total dissolved solids (TDS) in water and returns results in parts per million (ppm), indicating water cleanliness and possible contaminant levels.

The Water Quality Monitoring System is designed to be deployed at 3-5 different locations along a river. The data is wirelessly transmitted to a central database using the ESP32 microcontroller, which reads the data from the sensors. Users may then view this data on an LCD panel or through the Blynk application, which enables them to keep an eye on the water quality in real time. This system is implemented by configuring the ESP32 microcontroller to communicate with the TDS measure and DS18B20 sensors, gathering data, and wirelessly transmitting it to the database. Users can view the gathered data on platforms provided by the Blynk application and LCD screen, giving them the ability to make knowledgeable decisions regarding the quality of the water. The goal of the suggested Water Quality Monitoring System is to offer a scalable and reasonably priced option for ongoing water quality monitoring. The system provides a useful method for real-time water quality monitoring, protecting public health and water resources by utilising the ESP32 microcontroller and simple sensors.

5.2 Constraints

There are several constraints exist during the system implementation process. The constraints include cost, time, hardware and technical constraints.

5.2.1 Hardware Constraints

Numerous hardware issues arose during the Water Quality Monitoring System's deployment. Because of compatibility problems and calibration requirements, integrating the TDS metre and DS18B20 temperature sensor with the ESP32 microcontroller was a challenging operation. Furthermore, there were times when interference and environmental factors caused the sensors to give erroneous readings, which called for recalibration and sensor replacement. There were power supply variations that caused instability and sporadic resets in the ESP32 microcontroller. By using safety circuitry and voltage regulators, this problem was lessened. Voltage imbalances during the programming process might have caused the ESP32 microcontroller to short circuit, but this could have been avoided by making sure that power management was done correctly. Moreover, when there were problems with sensor data transfer, the system would periodically show "SENSOR ERROR" on serial monitor at Arduino ide.

Despite these difficulties, all hardware problems were found and fixed in time for the project's submission, guaranteeing the system's dependable operation. However, due to restrictions on sensor accuracy and ambient unpredictability, the system still has difficulty reliably detecting contamination levels. The bulk of these hardware limitations were successfully resolved, enabling the system to operate as planned, monitoring water quality in real time and setting the stage for further advancements. Most of these hardware limitations were successfully resolved, enabling the system to operate as planned, monitoring water quality in real time and setting the stage for further advancements. However, the actual findings have been uneven, underlining the need for additional improvement to ensure accurate contamination detection.

5.2.2 Time Constraints

The Water Quality Monitoring System implementation is time-constrained because it must solve issues with hardware integration, sensor calibration, and ambient conditions that affect sensor accuracy in a certain amount of time. The system's interoperability, stability, and dependability need careful planning and execution of the work. To avoid project milestone delays, enough time must be set out for addressing hardware problems, such as power supply variations and sensor mistakes. Furthermore, more time may be needed to fine-tune the system for precise contamination detection. Despite these difficulties, completing projects on schedule is crucial. To guarantee that the system functions as planned and lays the groundwork for future improvements, effective time management and proactive problem-solving are required.

5.2.3 Cost Constraints

The student was facing an excessive amount for the hardware needed for this system, and the cost of finishing the system had gone up since various modules and microcontrollers needed to be replaced on a regular basis.

5.2.4 Technical Constraints

To deploy this system, new ideas such as the Internet of Things and mobile apps have to be learned. Due to a lack of expertise, the load of researching and learning the programming language for Arduino, Blynk Application, and configuring the NodeMCU ESP32 with the sensors has grown. There were many errors from the beginning until the system was completed in code implementation, but the system was completed after the arduous of study and learning about the idea and implementation process of the Water quality monitoring system employing water quality sensors.

5.3 Future Work

To enhance the IoT-based water quality monitoring system, additional sensors for parameters such as pH, dissolved oxygen, temperature, and turbidity may be added to provide a more thorough understanding of water quality. Improving data transmission and processing capabilities, as well as providing user-friendly interfaces for real-time monitoring and decision-making, can increase usability and accessibility. Engaging local communities and stakeholders through partnerships and outreach campaigns may promote collective responsibility and action to solve water quality issues. Furthermore, researching novel sensor technologies and data analytics approaches, such as sophisticated machine learning algorithms, can improve the accuracy and early detection of water pollution occurrences. Emphasising community involvement tactics such as citizen science activities and educational programmes encourages locals to actively participate in monitoring efforts while advocating for sustainable water management practices. By focusing on continual development and cooperation, the system grows into a powerful instrument for protecting water resources and fostering environmental stewardship on a larger scale.

5.4 Summary

To put it simply, the water quality prototype falls short of the objectives and must be modified and improved before it can be deployed in the river area. This device might serve as an option for obtaining clean water in the river area. This device may also send out alerts and warnings regarding contaminated water. Although this initiative was successful, there is still room for improvement in the future as the world changes. The Internet of Things (IOT) has limitless promise in the future, as numerous businesses incorporate the technology into their products.

REFERENCES

Manoj, M., Kumar, V. D., Arif, M., Bulai, E. R., Bulai, P., & Geman, O. (2022). State of the Art Techniques for Water Quality Monitoring Systems for Fish Ponds Using IoT and Underwater Sensors: A Review. In *Sensors* (Vol. 22, Issue 6). https://doi.org/10.3390/s22062088

Razman, N. A., Wan Ismail, W. Z., Abd Razak, M. H., Ismail, I., & Jamaludin, J. (2023). Design and analysis of water quality monitoring and filtration system for different types of water in Malaysia. *International Journal of Environmental Science and Technology*, *20*(4). https://doi.org/10.1007/s13762-022-04192-x

Sung, W. T., Fadillah, F. N., & Hsiao, S. J. (2021). IoT-based Water Quality Monitoring. *Sensors and Materials*, *33*(8). https://doi.org/10.18494/SAM.2021.3342

Vijayakumar, N., & Ramya, R. (2015). The real time monitoring of water quality in IoT environment. *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*. https://doi.org/10.1109/ICIIECS.2015.7193080

"Grove - TDS Sensor | Seeed Studio Wiki," Jan. 06, 2023. https://wiki.seeedstudio.com/Grove-TDS-Sensor/

S. Santos and S. Santos, "ESP32 Pinout Reference: Which GPIO pins should you use? | Random Nerd Tutorials," *Random Nerd Tutorials*, Nov. 08, 2022. https://randomnerdtutorials.com/esp32-pinout-reference-gpios/

U. Ahmed, R. Mumtaz, H. Anwar, S. Mumtaz, and A. M. Qamar, "Water quality monitoring: from conventional to emerging technologies," *Water Supply*, vol. 20, no. 1, pp. 28–45, Oct. 2019, doi: 10.2166/ws.2019.144.

APPENDICES

Appendix A: Full Code

```
#define BLYNK_TEMPLATE_ID "TMPL6xybMcdhg"

#define BLYNK_TEMPLATE_NAME "Water Quality Monitoring"

#define BLYNK_AUTH_TOKEN "KYtxzTgH4nJqahfhx68z9sddOauazRbj"

#define BLYNK_PRINT Serial



#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>



#include <OneWire.h>

#include <DallasTemperature.h>



#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "syafiqah";
char pass[] = "cmkr9999";


struct PhysicalPins
{
  int tds;
};
PhysicalPins pPin = {
    27,
};


#define ONE_WIRE_BUS 5
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
```

```
#define Temperature V0

#define TDS_Level V1

#define tempThreshold V2

#define tdsThreshold V3



int temperatureThresholdVpinVal = 0;

int tdsThresholdVpinVal = 0;



struct VirtualData

{

  bool isSend;

};

VirtualData tempNotify = {false};

VirtualData tdsNotify = {false};



BLYNK_WRITE(tempThreshold)

{
```

```
  temperatureThresholdVpinVal = param.asInt();

  Serial.println("Trigger temperature : " + String(temperatureThresholdVpinVal));

}



BLYNK_WRITE(tdsThreshold)

{

  tdsThresholdVpinVal = param.asInt();

  Serial.println("Trigger TDS : " + String(tdsThresholdVpinVal));

}



const int numSensorReading = 2;



float smoothingFloatValue(float sensorValue, float *readingSensor, int
&indexSensor)

{

  float totalReadingSensor = 0.0, smoothedValue = 0.0;

  readingSensor[indexSensor] = sensorValue;

  indexSensor = (indexSensor + 1) % numSensorReading;
```

```
    for (int i = 0; i < numSensorReading; i++)
    {
      totalReadingSensor += readingSensor[i];
    }

    smoothedValue = totalReadingSensor / numSensorReading;
    return smoothedValue;
  }

  int smoothingSensorValue(int sensorValue, int *readingSensor, int
&indexSensor)
  {
    int totalReadingSensor = 0, smoothedValue = 0;
    readingSensor[indexSensor] = sensorValue;
    indexSensor = (indexSensor + 1) % numSensorReading;
    for (int i = 0; i < numSensorReading; i++)
    {
```

```
    totalReadingSensor += readingSensor[i];

   }

   smoothedValue = totalReadingSensor / numSensorReading;

   return smoothedValue;

  }



  float smoothTemperature(float temperatureValue, float readingSensor[], int
&indexSensor)

  {

   float temperatureSmooth = 0.00;

   temperatureSmooth = smoothingFloatValue(temperatureValue, readingSensor,
indexSensor);

   Serial.println("Temperature Smooth: " + String(temperatureSmooth));

   Serial.println();

   return temperatureSmooth;

  }



  #define mostTDS 100
```

```
#define leastTDS 850



int readPPMAnalog(int sensorPin, int dataForMinPercentage, int dataForMaxPercentage, int ppmMin, int ppmMax, int readingSensor[], int &indexSensor)

{

  int sensorValue = 0, sensorSmooth = 0, ppmValue = 0;



  sensorValue = analogRead(sensorPin);

  sensorSmooth = smoothingSensorValue(sensorValue, readingSensor, indexSensor);



  ppmValue = map(sensorSmooth, dataForMinPercentage, dataForMaxPercentage, ppmMin, ppmMax);



  Serial.println("TDS Value  : " + String(sensorValue));

  Serial.println("TDS Smooth : " + String(sensorSmooth));

  Serial.println("TDS PPM    : " + String(ppmValue));

  Serial.println();
```

```
  return ppmValue;
}


void clearRow(int row)
{
 lcd.setCursor(0, row);
 lcd.print("                ");
}


void printRow(int column, int row, String messages)
{
 clearRow(row);
 lcd.setCursor(column, row);
 lcd.print(messages);
}


void printRowData(int column, int row, String messages, float sensorValue)
```

```
{
  clearRow(row);
  lcd.setCursor(column, row);
  lcd.print(messages + String(sensorValue));
}


float readingSensor1[numSensorReading];
int indexSensor1 = 0;


int readingSensor2[numSensorReading], indexSensor2 = 0;


struct EventData
{
  float value;
  float decimal;
  float smooth;
  int counter;
```

```
  bool isTrigger;

};

EventData tempDallas = {0.0, 0.00, 0.00, 0, false};

EventData indeedTemp = {0.0, 0.00, 0.00, 0, false};

EventData tds = {0, 0.00, 0.00, 0, false};

EventData indeedTDS = {0, 0.00, 0.00, 0, false};



int debounceThreshold = 3;



void clampingCounter(int &counter)

{

  if (counter < debounceThreshold)

  {

    counter++;

  }

}
```

}

```
void mainFunction() {

    Serial.println("// ---------------------------------------------------------------- //");

    Serial.println("   TEMPERATURE READING IN CELSIUS UNIT");

    Serial.println("// ---------------------------------------------------------------- //");

    sensors.requestTemperatures();

    float tempCelsius = sensors.getTempCByIndex(0);

    float tempSmooth = smoothTemperature(tempCelsius, readingSensor1,
indexSensor1);

    bool tempTrigger = (tempSmooth >= temperatureThresholdVpinVal);

    Blynk.virtualWrite(Temperature, tempSmooth);

    printRowData(0, 0, "Temp: ", tempSmooth);



    if (tempTrigger) {

        clampingCounter(tempDallas.counter);



        if (tempDallas.counter >= debounceThreshold) {

            indeedTemp.isTrigger = true;

        }
```

```
    } else {

        indeedTemp.isTrigger = false;

        tempNotify.isSend = false;

        tempDallas.counter = 0;

    }



    Serial.println("   TDS SENSOR READING IN PPM UNIT");

    Serial.println("// ------------------------------------------------------------------ //");

    int tdsValue = readPPMAnalog(pPin.tds, leastTDS, mostTDS, 0, 1000,
readingSensor2, indexSensor2);

    bool tdsTrigger = (tdsValue >= tdsThresholdVpinVal);

    Blynk.virtualWrite(TDS_Level, tdsValue);

    printRowData(0, 1, "TDS: ", tdsValue);



    if (tdsTrigger) {

        clampingCounter(tds.counter);



        if (tds.counter >= debounceThreshold) {
```

```
            indeedTDS.isTrigger = true;

        }

    } else {

        indeedTDS.isTrigger = false

;

        tdsNotify.isSend = false;

        tds.counter = 0;

    }

}



#define handleAll_INTERVAL 1000L

BlynkTimer HandleAll;



void setup()

{

  Serial.begin(115200);

  sensors.begin();
```

```
WiFi.begin(ssid, pass);

Blynk.begin(auth, ssid, pass);

while (WiFi.status() != WL_CONNECTED)

{

  delay(500);

  Serial.print(".");

}



pinMode(pPin.tds, INPUT);



Blynk.virtualWrite(Temperature, 0);

Blynk.virtualWrite(TDS_Level, 0);

Blynk.virtualWrite(tempThreshold, 0);

Blynk.virtualWrite(tdsThreshold, 0);



HandleAll.setInterval(handleAll_INTERVAL, mainFunction);
```

```
  lcd.begin();
  lcd.backlight();
}

void loop ()
{
  Blynk.run();
  HandleAll.run();
}
```

Appendix B: River