Bachelor of Science in Computer Science
May 2023

IoT-based Saline Volume Monitoring and Alert System

Narendra Velpula
Dinesh Kotti

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science. The thesis is equivalent to 10 weeks of full-time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Narendra Velpula
E-mail: navl22@student.bth.se

Dinesh Kotti
E-mail: diko22@student.bth.se

University advisor:
Dr. Prashant Goswami, Associate Professor
Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

This project presents a comprehensive study on the design and implementation of an Internet of Things (IoT)-based system for monitoring and alerting saline volumes in healthcare environments.

Background: In healthcare settings, the accurate monitoring of saline volumes in Intravenous (IV) drip systems is crucial for ensuring patient safety and effective treatment. Traditional monitoring methods are labour-intensive and prone to human error. The IoT offers promising solutions for automating and enhancing the monitoring process.

Objectives: This thesis aims to develop an IoT-based saline volume monitoring and alert system using NodeMCU, a load sensor, an amplifier, the ThingSpeak cloud platform, and the Massachusets Insitute of Technology (MIT) App Inventor. The system is designed to improve the accuracy and efficiency of saline volume monitoring while reducing the burden on healthcare professionals.

Methods: The proposal system employs a Node MicroController Unit (NodeMCU) microcontroller for data processing and communication, a load sensor for monitoring the saline volume, and a buzzer alarm and amplifier for alerting healthcare professionals when the saline volume reaches a critical threshold. The system connects to the ThingSpeak cloud platform for data storage and analysis, facilitating remote monitoring and control through a custom mobile application developed using MIT App Inventor.

Results: The implementation and testing of the system showed accurate and reliable monitoring of saline volumes in real-time, with efficient alerting mechanisms. The user-friendly mobile application enabled healthcare professionals to monitor multiple IV drip systems simultaneously, receiving timely alerts when intervention was required.

Conclusions: The IoT-based saline volume monitoring and alert system demonstrates the potential to improve patient safety and healthcare efficiency. Further research and development can explore the integration of additional sensors, the refinement of the alert system, and the assessment of the system's impact on clinical outcomes..

Keywords: IoT, MIT App Inventor, NodeMCU, Saline Monitoring, and ThingSpeak.

Acknowledgments

We would like to express our sincere gratitude and appreciation to all those who have contributed to the successful completion of this project. Their support, guidance, and encouragement have been invaluable throughout this journey.

First and foremost, we extend our heartfelt thanks to our project supervisor **Dr. Prashant Goswami**. His expertise, dedication, and insightful guidance have played a pivotal role in shaping and directing our project. His continuous support, constructive feedback, and willingness to share knowledge have been instrumental in our learning and growth.

We would also like to express our gratitude to **Dr. Yan Hu**, our project examiner, for taking the time to evaluate our work. We appreciate her valuable feedback, suggestions, and critical insights, which have helped us improve the quality and rigour of our project.

We would like to acknowledge our friends and classmates for their support and cooperation throughout this project. Their encouragement, discussions, and exchange of ideas have contributed to the overall development and refinement of our work.
Last but not least, we extend our heartfelt appreciation to our families for their unwavering support, understanding, and belief in our abilities. Their love, encouragement, and patience have been the pillars of strength that have motivated us to overcome challenges and strive for excellence.

Authors:
Narendra Velpula
Dinesh Kotti

Contents

Abstract **i**

Acknowledgments **iii**

1 Introduction **1**
1.1 Aim and Objectives . . . 3
1.1.1 Objectives . . . 3
1.2 Research questions . . . 3
1.3 Scope . . . 3
1.4 Outline . . . 3

2 Background **5**
2.1 Internet of Things . . . 5
2.1.1 IoT in Saline Volume Monitoring . . . 5
2.1.2 IoT components in Saline Volume Monitoring and Alert Systems . . . 5
2.1.3 Advantages of IoT in Saline Volume Monitoring and Alert System . . . 6
2.2 Hardware Components . . . 6
2.2.1 NodeMCU . . . 6
2.2.2 Load Sensor . . . 8
2.2.3 Amplifier . . . 9
2.2.4 Buzzer alarm . . . 10
2.3 Software Components . . . 11
2.3.1 ThingSpeak . . . 11
2.3.2 MIT App Inventor . . . 13

3 Related Work **15**

4 Method **17**
4.1 Describing the Research Method . . . 17
4.2 System Design . . . 17

4.2.1 Hardware Implementation 17
4.2.2 Software Implementation 20

5 Results and Analysis **27**
5.1 Saline Volume Alert Using Buzzer Alarm 27
5.2 Accurate Measurement of saline volume Using Load Sensor 28

6 Discussion **33**
6.1 Addressing the Research Questions 33
6.2 Relating Findings to the Literature 34
6.3 Limitations and Conflicting Data 34
6.4 Implications, Recommendations, and Conclusions 35

7 Conclusions and Future Work **37**
7.1 Conclusion 37
7.2 Future Work 38

References **41**

List of Figures

1.1 This Figure depicts that Architecture diagram of the Saline Volume monitoring and alert system . . . 2

2.1 NodeMCU: Empowering Wireless Data Acquisition and Communication . . . 7
2.2 Buzzer Alarm . . . 10
2.3 ThingSpeak Platform . . . 11
2.4 MIT App Inventor . . . 14

4.1 The above flow chart diagram depicts how the saline volume monitoring and alert system works using the hardware and software components like NodeMCU, ThingSpeak, Load sensor, MIT APP Inventor, and Buzzer alarm. . . . 18
4.2 Code snippet for connecting the loadcell to the NodeMCU through the 12 pin and 13 pin from NodeMCU to loadcell DOUT and SCK pins. . . . 19
4.3 Connected the amplifier to NodeMCU to send saline volume data to NodeMCU. . . . 20
4.4 Home page of ThingSpeak cloud platform website to store and visualize the saline volume. . . . 21
4.5 Code snippet for connecting NodeMCU and sending saline volume data into ThingSpeak using Write API keys. . . . 21
4.6 Screenshot of creating channels in ThingSpeak cloud platform. . . . 22
4.7 The above Screenshot depicts how to create API keys in the ThingSpeak website and make permissions according to the MIT APP Inventor. . . . 23
4.8 The above diagram depicts a code snippet showing how to create an app by drag and drop in the MIT APP Inventor for monitoring the saline volume. . . . 24
4.9 Screenshot of Monitoring Saline Volume in the Mobile App developed using MIT App Inventor. . . . 24

4.10 Screenshot of Monitoring Saline Volume in the Mobile App developed using MIT App Inventor. 25

5.1 Cirucit Diagram for Integrating the Hardware Components NodeMCU, Buzzer alarm, load cell, amplifier 27
5.2 Integrated Hardware Setup for Saline Volume Monitoring: Demonstrating the Interconnected Components in a Complete Circuit Configuration. 28
5.3 The above figure depicts the illustration of saline volume present on a particular date and the graph shows the amount of volume available on a particular date at the ThingSpeak cloud platform. . . 30
5.4 The above figure depicts the illustration of saline volume present on an average date and the graph shows the amount of volume available on an average date at the ThingSpeak cloud platform. 31

List of Tables

5.1 The table shows the difference between the standard calibration weight in the saline bottle and the load sensor weight in the saline bottle. 29

List of Acronyms

ADC	Analog to Digital Converter.
API	Application Program Interface.
AWS	Amazon Web Service.
BJT	Bipolar Junction Transistor.
DOUT	Data IO Connection.
E+	Excitation+.
E-	Excitation-.
FET	Field-Effect Transistor.
GND	Ground.
GPIO	General Purpose Input/Output.
GUI	Graphical User Interface.
HTTP	Hyper Text Transfer Protocol.
IoT	Internet of Things.
IV	Intravenous.
JSON	JavaScript Object Notation.
MIT	Massachusets Insitute of Technology.
NodeMCU	Node MicroController Unit.
S+	Signal+.
S-	Signal-.
SCK	Serial Clock Input.
SNS	Simple Notification Service.
USB	Universal Serial Bus.
VCC	Voltage at Common Collector.

Chapter 1

Introduction

The healthcare industry has witnessed significant transformations in recent years, driven by technological advancements and the integration of IoT technologies. The IoT, which refers to the network of interconnected devices and sensors, has the potential to revolutionize healthcare by enabling real-time data collection, analysis, and remote monitoring. This technology has found applications in various healthcare domains, ranging from patient monitoring to medication management [3]. One critical aspect of healthcare is the administration of the saline solution, which is widely used in medical devices for delivering fluids to patients. Accurate monitoring of saline volume is essential to ensure proper hydration, prevent complications, and maintain patient safety. However, traditional manual monitoring methods prone to errors, delays, and inefficiencies. Healthcare professionals often rely on periodic checks or visual inspections, which may lead to missed or delayed interventions when saline volumes become critically low [8]. To address these challenges, there is a growing need for automated and reliable systems that can monitor saline volumes in real time and provide timely alerts to healthcare professionals. The advent of IoT technologies, coupled with the availability of low-cost and easily accessible components, opens up new possibilities for developing efficient and cost-effective solutions in this domain.

In the realm of healthcare monitoring systems, there exists a significant research gap pertaining to the monitoring and alert systems for saline volumes [16]. While various monitoring techniques are available, the current methods lack comprehensive and efficient solutions that can provide accurate and real-time monitoring of saline volumes [14]. Furthermore, the integration of IoT technologies within the healthcare infrastructure is limited, inhibiting the development of seamless and automated monitoring systems [23]. The significance of this research gap lies in its potential to provide an innovative solution for the monitoring and management of saline-level volumes in healthcare settings.

In this project, we have designed and implemented an IoT-based saline volume monitoring and alert system using the microcontroller NodeMCU, a load sensor for weight measurement, a buzzer for alert, an amplifier for increasing voltage, ThingSpeak for cloud storage, and Massachusets Insitute of Technology (MIT) App inventor for creating the user interface. The NodeMCU, based on the ESP8266 Wi-Fi module, serves as the central device for data collection and Transmission. The load sensor measures the weight and calculates the saline volume in medical devices accurately [25]. The system utilizes the ThingSpeak cloud platform to store and analyze the collected data, providing healthcare professionals with real-time access to saline-level information and historical trends. Furthermore, the system incorporates a buzzer alarm and an amplifier to generate audible alerts when the saline volume falls below a predefined threshold [28]. The MIT App Inventor is utilized to develop a mobile application that enables healthcare professionals to receive real-time alerts, monitor saline volumes remotely, and configure system settings. The system's ability to continuously monitor the saline volumes in real-time and provide timely alerts can significantly enhance patient safety, streamline healthcare processes, and enable proactive interventions. Healthcare professionals can remotely monitor multiple medical devices, receive alerts on their mobile devices, and take necessary actions promptly by reducing the risk of interruptions in medical procedures and ensuring optimal patient care [11]. The proposed system can fill the existing void in the market for accurate monitoring systems, enabling healthcare professionals to respond promptly to any deviations in saline volumes. Additionally, the integration of IoT technologies can pave the way for the seamless incorporation of monitoring systems into existing healthcare infrastructure, facilitating enhanced patient care and treatment.

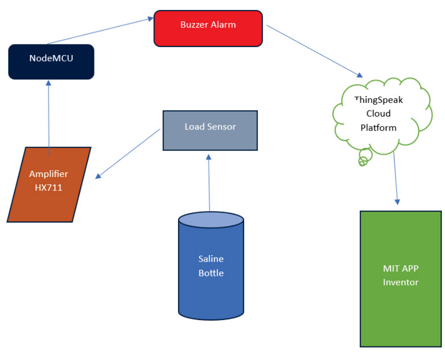

Figure 1.1: This Figure depicts that Architecture diagram of the Saline Volume monitoring and alert system

1.1 Aim and Objectives

Our project aims in creating IoT based saline-level monitoring and alert system using NodeMcu, Load sensor, Buzzer sensor, and ThingSpeak cloud-based service, along with an Android application developed using MIT App Inventor.

1.1.1 Objectives

- The objective of our project is to develop and monitor the saline volume and to create an alert system that can be triggered through the user interface to alert healthcare professionals of any issues with saline volumes.
- To build a user interface for healthcare professionals to monitor the amount of saline in the saline bottle.

1.2 Research questions

1. How can an alert be triggered using a buzzer when the saline volume goes below a threshold?
2. What is the most accurate method for measuring saline volume using a load sensor in an IoT-based saline volume monitoring and alert system?

1.3 Scope

The scope of this project revolves around the development of an IoT-based saline volume monitoring and alert system using NodeMCU, a load sensor, a buzzer alarm, an amplifier, the ThingSpeak cloud platform, and the MIT App Inventor. The primary focus is to design and implement a reliable and efficient system that monitors the saline volumes in real-time and triggers an alert when the saline volume goes below a threshold.

1.4 Outline

The outline of the project is Chapter 1, begins with an introduction that establishes the aim, objectives, research questions, and scope of the IoT-based saline volume monitoring and alert system. Chapter 2 explores the background of all the software and hardware components of a saline volume monitoring and alert system. Chapter 3 is about the related work of the system. Chapter 4 is about the method, which includes the implementation of hardware and software components and discussion.

Chapter 5 is about the results that are obtained from the methods and analyzed based on the results. Chapter 6 is about the complete discussion of the project, how we achieved the objectives, and answering the research questions of the saline monitoring and alert system. and finally, Chapter 7 is about the conclusion and future work of our IoT-based saline volume monitoring and alert system.

Chapter 2

Background

2.1 Internet of Things

2.1.1 IoT in Saline Volume Monitoring

The adoption of IoT in saline-level monitoring systems allows for continuous, real-time monitoring and data collection of water salinity levels. By using IoT-enabled devices and sensor technology, stakeholders can access accurate and timely information, enabling them to make informed decisions and take appropriate measures to maintain optimal salinity levels [25].

2.1.2 IoT components in Saline Volume Monitoring and Alert Systems

Sensors: Electrochemical or optical sensors are used to measure the salinity levels in water sources. These sensors are designed to detect changes in the water's electrical conductivity or refractive index, which are directly related to the water's salinity.

Data Analytics and Visualization: The collected data can be processed and analyzed using various data analytics tools and machine learning algorithms. This information can be visualized through web-based applications platforms or mobile applications, providing valuable insights into salinity trends and patterns [14].

Alert Systems: IoT-enabled saline volume monitoring systems can send notifications and alerts to authorized users when salinity levels exceed predefined thresholds. This enables timely intervention to prevent damage to agriculture, infrastructure, and aquatic ecosystems.

2.1.3 Advantages of IoT in Saline Volume Monitoring and Alert System

Real-time monitoring: IoT-enabled saline volume monitoring systems provide real-time data, enabling stakeholders to make timely decisions and take appropriate actions to maintain optimal salinity levels [11].

Remote access: IoT devices can be accessed and controlled remotely, allowing stakeholders to monitor and manage water resources from anywhere in the world.

Scalability: IoT-based systems can be easily scaled up or down, depending on the size and complexity of the water resource being monitored.

Cost-effectiveness: IoT devices have the potential to reduce operational costs by automating data collection, analysis, and reporting processes. Moreover, IoT systems can help prevent damage to agriculture, infrastructure, and aquatic ecosystems, ultimately saving resources and costs.

Environmental conservation: By enabling precise and timely management of water salinity levels, IoT systems can help protect aquatic ecosystems and promote sustainable water resource management.

2.2 Hardware Components

2.2.1 NodeMCU

NodeMCU is an open-source firmware and development board that helps users build IoT projects. It is based on the ESP8266 Wi-Fi microcontroller, which is a low-cost, low-power consumption, and compact-sized module designed to enable the integration of various devices with Wi-Fi capabilities [7]. The NodeMCU development board includes an ESP8266 chip, Universal Serial Bus (USB) interface, and voltage regulator, making it easy to prototype and voltage regulator, making it easy to prototype and develop IoT applications. In this project, we utilize NodeMCU to create a saline-level monitoring and alert system.

The figure 2.1 depicts the NodeMCU, a compact microcontroller board with built-in Wi-Fi capabilities, serving as the central device for wireless data acquisition and communication in our project [1].

Features of NodeMCU: NodeMCU offers a range of features that make it suitable for an IoT project such as our saline volume monitoring and alert system:

- **ESP8266 Microcontroller:** The ESP8266 microcontroller offers Wi-Fi ca-

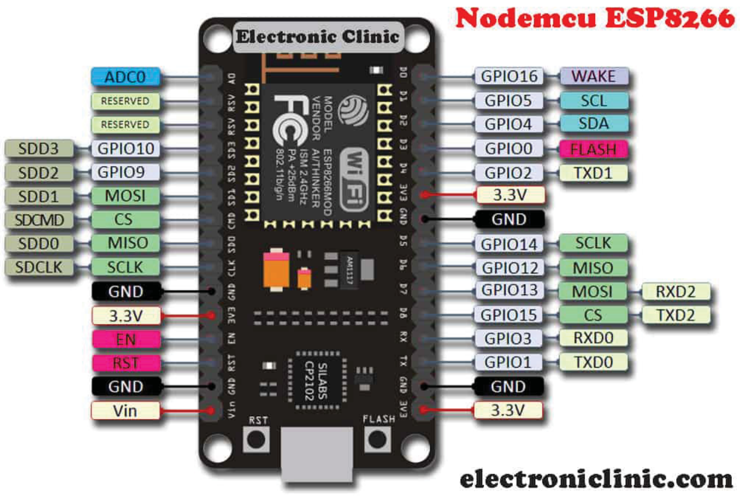

Figure 2.1: NodeMCU: Empowering Wireless Data Acquisition and Communication

pabilities and a powerful processor, which enables the development of IoT applications with ease [27]. It has a 32-bit RISC CPU with a clock speed of up to 160 MHz and comes with up to 4 MB of flash memory. This provides sufficient processing power and storage for our project requirements.

- **Programming Support:** NodeMCU supports the Lua scripting language, which is a lightweight, easy-to-learn scripting language. This allows for rapid prototyping and development of IoT projects [10]. Additionally, NodeMCU can be programmed using the Arduino IDE, which provides a familiar programming environment for developers and support for various libraries and tools.

- **GPIO Pins:** NodeMCU features a range of General Purpose Input/Output (GPIO) pins, which can be used to read input signals from sensors or control actuators. In our saline-level volume monitoring system [5], we use these GPIO pins to interface with a saline-level volume sensor and trigger alerts.

- **Power Options:** NodeMCU can be powered through a micro USB port or an external power supply, providing flexibility for deployment in various scenarios. This is particularly useful for our saline volume monitoring system as it allows for easy installation and maintenance.

- **NodeMCU in saline volume Monitoring and Alert System:** In this project, we use the NodeMCU development board to create a saline-level monitoring and alert system [11]. The system is designed to continuously monitor the saline volume in a container and provide real-time alerts when the saline volume reaches a critical threshold.

System Components: The key components of the saline volume monitoring and alert system include:

- **NodeMCU development board:** Acts as the central processing unit for reading sensor data, processing it, and triggering alerts.
- **saline volume sensor:** Detects the saline volume in the container and provides an analog output signal proportional to the level.
- **PowerSupply:** Provides power to the NodeMCU and other components in the system.
- **System Operation:** The saline volume monitoring and alert system operates as follows: The NodeMCU continuously reads the output of the saline volume sensor through its GPIO pins. The NodeMCU processes the sensor data to determine the current saline volume and compares it to a predefined threshold. If the saline volume falls below the threshold, the NodeMCU triggers an alert by sending a notification via Wi-Fi to the appropriate recipients. The system continues to monitor the saline volume, providing real-time updates and alerts as needed.

2.2.2 Load Sensor

LoadSensor: In the field of IoT-based monitoring and alert systems, sensors play a vital role in collecting the required data for further processing and analysis. One such sensor that can be particularly helpful in an IoT-based saline volume monitoring and alert system is a load sensor. This chapter delves into the background of load sensors, their working principle, types, and potential applications in the project. A load sensor, also known as a load cell or force sensor, is a transducer that converts an applied force into an electrical signal. The primary function of a load sensor is to measure the weight or force acting upon an object, and it is widely used in various industries for applications such as industrial automation, medical equipment, and transportation systems. In the context of an IoT-based saline volume monitoring and alert system, a load sensor can be employed to measure the weight of a saline bottle, which can then be used to estimate the saline volume within the bottle.

Load Sensor Working Principle:
Most load sensors are based on the principle of strain gauges, which are thin,

flexible conductive materials that can change their electrical resistance when subjected to an applied force [13]. A strain gauge is typically bonded to a stiff and elastic substrate material, such as steel, aluminum, or plastic. When the substrate experiences a deformation due to an applied force, the strain gauge also deforms, causing a change in its electrical resistance. This change in resistance is directly proportional to the applied force and can be measured using a Wheatstone bridge circuit. The force or weight acting on the sensor can be accurately determined by amplifying and processing this signal. The maximum weight of the saline bottle is

2.2.3 Amplifier

Amplification is the process by which an electric device boosts the strength, voltage, or current of an input signal. It is essential to the operation of many electronic systems, such as those used in communication and control, audio, and, in our instance, IoT-based saline volume monitoring and alarm system [16]. Information transmission and reception are made possible by amplifiers which are crucial for enhancing signal intensity and quality. The many sorts of amplifiers, how they operate, and how important they are to our IoT-based saline-level volume monitoring system will all be covered in this section.

Basic Operating Theory: The amplitude of the input signal is successfully increased by an amplifier using a power source. The fundamental components of an amplifier circuit include a transistor, resistors, and capacitors. The transistor, either Bipolar Junction Transistor (BJT) or Field-Effect Transistor (FET), operates as the active element that reregulates the flow of current between the input and output terminals [27].

IoT-based Saline Volume Monitoring and Alert System with Amplifier:
In our IoT-based saline volume monitoring and alert system, an amplifier plays a vital role in increasing the sensor signal quality, guaranteeing accurate monitoring and transmission of data. The system needs an amplifier to increase the signal strength before processing and transmission because the saline volume sensor may produce relatively low voltage signals.
An operational amplifier would be a good choice for this application because it can be set up to boost the sensor output voltage without adding a lot of noise. An Analog to Digital Converter (ADC) can then be used to digitally process the signal after it has been amplified.
Medical personnel may monitor and manage the saline volumes in real-time by

transmitting the processed data to a remote monitoring station using wireless communication modules.

2.2.4 Buzzer alarm

The saline volume monitoring and alert system is a crucial project aimed at improving the efficiency of healthcare services by ensuring the timely replacement of saline solutions for patients. A key component of this system is the buzzer alarm, which plays a crucial role in notifying healthcare professionals when the saline volume is low. In this section, we provide a detailed background on the buzzer alarm, its importance, and how it functions within the saline volume monitoring and alert system [12].
In a hospital setting, it is essential to monitor the saline solution level in intravenous (IV) drips to ensure that patients receive the required amount of fluids and medication. The timely replacement of saline solution is crucial to avoid potential health complications, maintain hydration, and deliver medications as prescribed. Nurses and healthcare professionals are often responsible for monitoring multiple patients, making it challenging to keep track of each individual's saline volumes [17].
The buzzer alarm is an effective tool to facilitate this monitoring process by providing an audible alert when the saline volume reaches a predetermined threshold. This helps healthcare professionals to promptly replace the saline solution, ensuring that patients receive the necessary treatment without any interruption [19].

Figure 2.2: Buzzer Alarm

The figure 2.2 showcases a buzzer alarm, a vital component of our project, designed to generate audible alerts when the Saline Volume falls below a predefined threshold. It ensures timely notifications for healthcare professionals, enhancing patient safety and intervention [2].

How the Buzzer Alarm Works: The buzzer alarm is an electronic component that generates an audible sound when it receives an electric signal. In the context of the saline volume monitoring and alert system, the buzzer alarm is integrated with the sensors send to the buzzer, which then emits an audible alert, notifying the healthcare professionals of the need to replace the saline solution. There are two primary types of buzzers used in such applications: active buzzers and passive buzzers. Active buzzers are self-contained units with built-in oscillators that generate the sound when powered. Passive buzzers, on the other hand, require an external oscillating signal to produce the audible sound.

2.3 Software Components

2.3.1 ThingSpeak

ThingSpeak is an IoT platform that allows users to collect, store, analyze, visualize, and act on data from IoT devices. It is an open-source platform that is widely used by developers, engineers, and scientists to build IoT applications and services. ThingSpeak provides a cloud-based infrastructure that enables users to connect their devices to the internet and to each other and analyze data in real time.

The platform was developed by MathWorks, a software company that specializes in mathematical computing software. It is built on MATLAB, a programming language that is widely used in engineering and scientific applications. ThingSpeak is designed to be an easy-to-use platform that allows users to quickly and easily connect their devices to the internet and start collecting data [6].

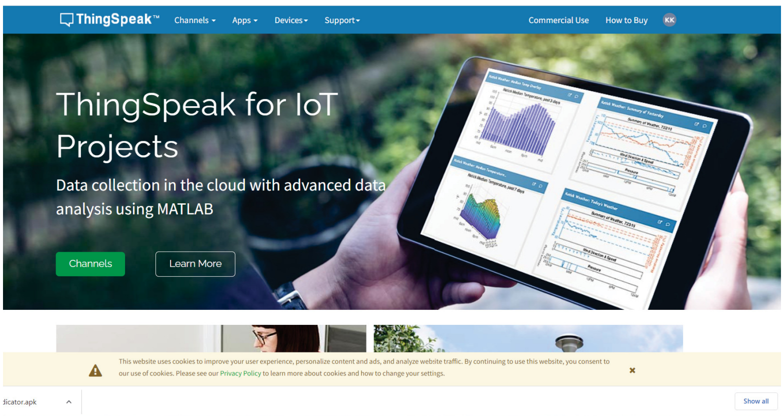

Figure 2.3: ThingSpeak Platform

The above Figure 2.3 depicts the ThingSpeak platform, a cloud-based storage and analysis system used in our project. It enables real-time data storage, visualization, and historical trend analysis, providing healthcare professionals with valuable insights for monitoring and managing Saline Volumes.

One of the key features of ThingSpeak is its ability to collect data from a wide range of sensors and devices. It supports a variety of protocols and interfaces, including Hyper Text Transfer Protocol (HTTP), MQTT, and RESTful Application Program Interface (API), which makes it easy to connect to different types of devices. Users can also create custom apps and widgets to visualize their data and to perform custom actions based on the data. Another important feature of ThingSpeak is its ability to integrate with other platforms and services. It supports integration with popular services like Twitter, and Twilio, which makes it easy to send alerts and notifications based on the data collected from the devices [18]. In our project, we are using ThingSpeak to monitor the saline volume in a bottle and to send alerts when the level in the bottle, and to send alerts when the level falls below a certain threshold. We have connected a level sensor to our device and configured it to send data to ThingSpeak. We have also set up a MATLAB script that analyzes the data and sends an alert to our phone when the saline volume falls below the threshold.

2.3.2 MIT App Inventor

Overview of MIT App Inventor:

It is not necessary to have substantial programming skills to create Android applications using the open-source, user-friendly MIT App Inventor platform. The Massachusets Insitute of Technology (MIT) developed this platform, which gives users a visual, drag-and-drop interface for creating original programs. MIT App Inventor has gained popularity among educators, students, and hobbyists eager to develop useful apps with little to no coding thanks to its block-based programming methodology [20].

Features and Components of App Inventor: There are two main parts of MIT App Inventor:

- **Designer:** This is the platform's Graphical User Interface (GUI) editor, where users may arrange different elements like buttons, labels, text boxes, and photos to create the layout of their apps.

- **Blocks Editor:** This feature enables users to specify the behaviour of the application by linking several 'blocks' that stand in for different functions, events, and attributes. Users can easily comprehend the logic and flow of their program since these blocks resemble jigsaw pieces [21]. A large number of functionalities are available in MIT App Inventor, such as: A wide range of features is already built-in for creating user interfaces, accepting user input, and presenting data support for a variety of sensors and devices, including accelerometers, GPS, and APIs. support for a variety of sensors and devices, including accelerometers, GPS, and Bluetooth Accessing external data and services through integration with web services and APIs [15]. An extensive collection of instructions, model projects, and documentation to assist users in getting started. In the context of the saline volume Monitoring and Alert System project, MIT App Inventor serves as a crucial tool for developing a user-friendly, responsive mobile application that allows healthcare professionals to monitor and manage patients' saline volumes remotely. This application will provide real-time data on saline volumes and send alerts to medical personnel if the levels fall outside the desired range.

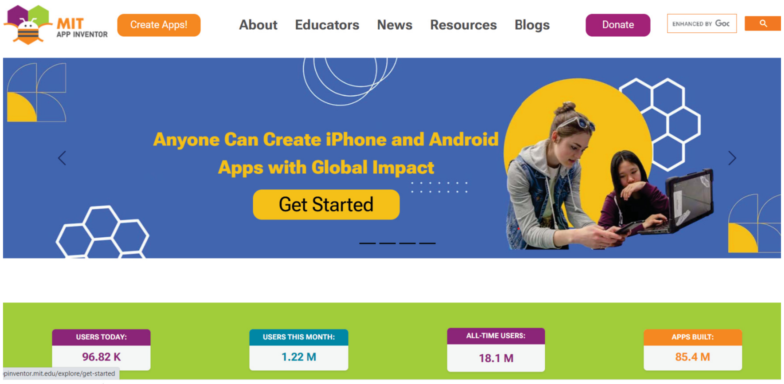

Figure 2.4: MIT App Inventor

The above Figure 2.4 showcases the MIT App Inventor, a visual development environment used in our project. It allows the creation of user-friendly mobile applications that enable healthcare professionals to receive real-time alerts, monitor saline volume remotely, and configure system settings for enhanced patient care.

The App for this project using MIT App Inventor has the following features:

- **Real-Time data visualization:** The App will present current saline volumes in a clear, graphical style so that medical professionals can quickly monitor the information.
- **Inform Alerts:** The App will send push notifications to inform healthcare providers when saline volumes stray outside of the specified limits, enabling them to take immediate action.
- **Patient Information Management:** The App's users will be able to add, amend, and manage patient information, making it possible to follow many patients effectively.

Chapter 3

Related Work

Nagaraj and Muneeswaran [22] have done their research on the monitoring and management of saline volumes in healthcare settings. They have proposed a system that tracks the saline volume using an IoT device. Their system includes an auto/manual mode, where the blood flow is controlled by a heart sensor, and an application in auto mode [26]. The researchers designed a saline-level monitor system that combines devices from the IoT, enabling the delivery of vital notifications to subscribers involved in medical care.

Pavan and Roshini [16] conducted research on the development of a monitoring system for normal saline in intravenous therapy(IV)with the goal of showcasing technology advancements in healthcare, particularly in remote areas where advanced technologies may not be readily available. Their focus was on creating a cost-effective system using locally available sensors. In their proposed system they used NodeMCU and an Ultrasonic sensor. Their approach aimed to address the need for continuous monitoring of saline solutions during IV therapy, which requires precise fluid delivery at regular intervals [8]. By utilizing locally available sensors and affordable components, they aimed to make the monitoring system accessible and cost-effective, especially in healthcare settings. Their work demonstrates the feasibility of developing a monitoring system for saline solutions using readily available components, providing a foundation for future advancements in remote healthcare technology.

Ravi Kishore and Poojith [14] conducted research to address the problem of detecting and alerting staff members about fluid levels without using a buzzer, considering the noisy environment of hospitals and the impact on patients, particularly those with heart conditions. They propose a prototype design that focuses on avoiding intense noise while ensuring effective communication. In their system, sensor data was recorded and transmitted to a microcontroller using the HX711 Amplifier [4]. The data was then sent to the cloud, specifically utilizing Amazon Web Service (AWS). The authors incorporated the Simple Notification Service

(SNS) in the AWS console. SNS allowed for push notifications to be sent to multiple devices simultaneously, through various means such as messages or emails. Their proposed prototype provided an innovative solution by utilizing load sensors and AWS services, offering a more patient-friendly and efficient monitoring system for fluid levels.

Jayeeta and Ankita [25] conducted research to focus on monitoring key health parameters, including heart rate, blood pressure, respiration rate, body temperature, body movement, and saline volumes. Their research contributes to the growing body of knowledge in the field of IoT-based healthcare monitoring and management [24]. Through the integration of automation and IoT technologies, healthcare professionals can address the challenges posed by busy schedules and irregular lifestyles, and enhance the quality of care provided to patients.

While previous research has mostly focused on the integration of IoT technologies in healthcare, namely in the monitoring of saline volumes [], our effort adds to the current body of knowledge by including an alert system based on IoT technology.

Chapter 4

Method

4.1 Describing the Research Method

To answer the research questions, a mixed-methods approach was employed, combining quantitative and qualitative methods. The scientific approaches we have chosen to answer the research issues are implementation for both research question 1 and 2. The chosen research method involves the following steps:

4.2 System Design

The system design phase of this study involved the development and integration of various components to realize the IoT-based saline volume monitoring and alert system. The section provides a detailed overview of the system design, including the implementation of the hardware and software components.

The above figure 4.1 represents the architecture diagram of the saline volume monitoring and alert system. The load sensor sends the value to the amplifier, the amplifier amplifies the value, the value will be sent to NodeMCU, NodeMCU will send data to ThingSpeak and it sends data to MIT App Inventor. If it reaches to certain threshold value then NodeMCU will trigger the buzzer alarm.

4.2.1 Hardware Implementation

The hardware components which we acquired for the implementation of IoT-based saline volume monitoring and alert system are :

1. A NodeMCU from the Amazon website.

2. A load sensor from the Amazon website.

3. An amplifier from the Amazon website.

4. A buzzer alarm from the Amazon website.

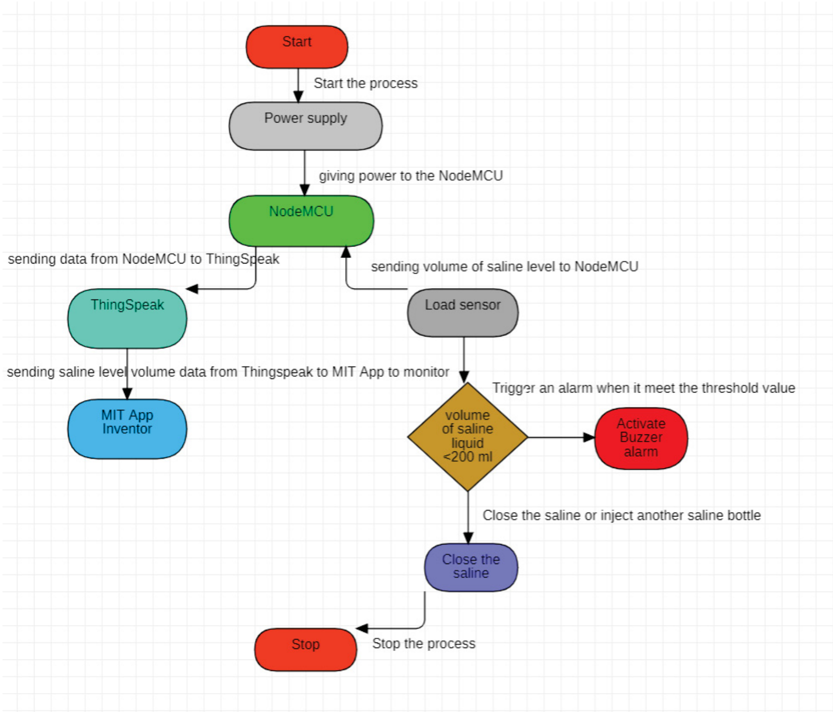

Figure 4.1: The above flow chart diagram depicts how the saline volume monitoring and alert system works using the hardware and software components like NodeMCU, ThingSpeak, Load sensor, MIT APP Inventor, and Buzzer alarm.

5. A saline bottle from the Medical store.
6. 3 meters of single-strand solid core wire from the Amazon website.

1. **Integrating Load Sensor with an Amplifier:** We implemented by integrating each hardware component to complete the circuit diagram and load sensor which consists of four different color wires green, red, white, and black. The Amplifier has two input channels one channel consists of pins that will be named Signal+ (S+), Signal- (S-), Excitation+ (E+), and Excitation- (E-) respectively. And the other channel consists of Voltage at Common Collector (VCC), Data IO Connection (DOUT), Serial Clock Input (SCK), and Ground (GND). After that, we implemented by connecting the red wire from the load sensor to the E+ pin in the amplifier, the green wire from the load sensor to the S+ pin in the amplifier, the white wire from the load sensor to the S- pin in the amplifier, and the black wire from the load sensor to E- pin in the amplifier.
2. **Integrating NodeMCU with the Load sensor:** The other channel in the

```
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
//#include <Adafruit_BME280.h>
//#include <Adafruit_Sensor.h>

const char* ssid = "nexin c7";   // your network SSID (name)
const char* password = "NEXINPGC";   // your network password

WiFiClient  client;

unsigned long myChannelNumber =2092038 ;
const char * myWriteAPIKey = "2RS3J2JC8FGKSVXI";

#include <Arduino.h>
#include "HX711.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 12;
const int LOADCELL_SCK_PIN = 13;

HX711 scale;
```

Figure 4.2: Code snippet for connecting the loadcell to the NodeMCU through the 12 pin and 13 pin from NodeMCU to loadcell DOUT and SCK pins.

Amplifier, VCC, DOUT, GND, and SCK was connected to the NodeMCU. We implemented this by connecting 12-pin and 13-pin from the NodeMCU to DOUT and SCK pins in the amplifier. To give a power supply for the NodeMCU, we implemented it by attaching a 3*3V (Voltage) lithium-ion battery to the VCC pin at the amplifier. To complete the circuit diagram, the GND pin in the amplifier can be placed on any one of the pins in NodeMCU.

3. **Integrating Buzzer alarm with the NodeMCU:** To connect a buzzer alarm with NodeMCU, we implemented it by using two wires. one wire from the buzzer alarm was placed on the NodeMCU pin to trigger the alarm when the saline volume goes below a threshold. Another wire from the buzzer alarm is placed on the ground to complete the circuit diagram. After completion of the circuit placement, we implemented it by attaching the saline bottle to the load sensor.

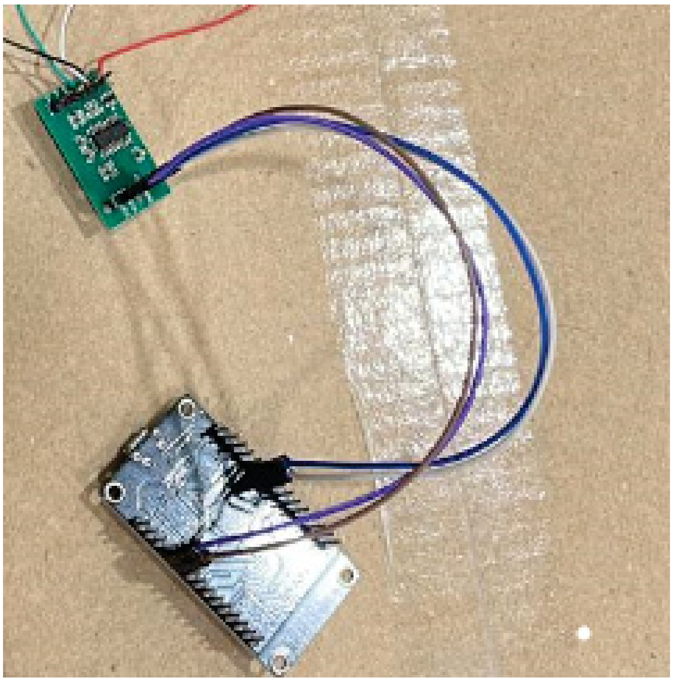

Figure 4.3: Connected the amplifier to NodeMCU to send saline volume data to NodeMCU.

4.2.2 Software Implementation

1. **Integrating NodeMCU to ThingSpeak cloud platform:**

   After the completion of all the connections in the hardware implementation, now we created to send the data to the ThingSpeak cloud platform. We have created an account in ThingSpeak to send the saline-level data into the cloud platform.

   We created a channel in ThingSpeak, which stored the data that is generated from the NodeMCU. Using C++ programming language, we implemented programming the API key into NodeMCU to transfer saline-level data into ThingSpeak. Using the username and password, we connected to Wi-Fi in NodeMCU to the ThingSpeak server. By using the HTTPClient library we sent an HTTP POST request to the ThingSpeak server with the data to be uploaded and the write API key.

   We have generated an API key in ThingSpeak. There are some API key URLs present under the API keys button on the ThingSpeak website see below Figure 4.7. We created it by selecting the Write API key to write the saline volume data into ThingSpeak.

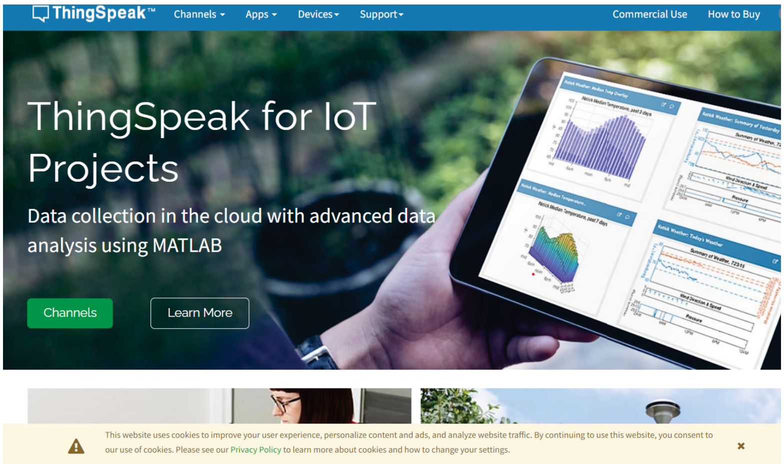

Figure 4.4: Home page of ThingSpeak cloud platform website to store and visualize the saline volume.

```
// set the fields with the values
//ThingSpeak.setField(1, value);
//ThingSpeak.setField(1, temperatureF);

// Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different
// pieces of information in a channel.  Here, we write to field 1.
int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

if(x == 200){
  Serial.println("\tChannel update successful.");
}
else{
  Serial.println("Problem updating channel. HTTP error code " + String(x));
}
lastTime = millis();
}
```

Figure 4.5: Code snippet for connecting NodeMCU and sending saline volume data into ThingSpeak using Write API keys.

2. **Connecting ThingSpeak to MIT App Inventor:**

   We created an account in MIT App Inventor, we created a project named saline monitoring in which we implemented by adding web components on the screen. In the Blocks Editor, by using the web component "Get" method retrieves the data from ThingSpeak using ThingSpeak API. The value that came from the ThingSpeak to MIT App Inventor is in JavaScript Object Notation (JSON) code. We wrote the code in MIT App Inventor to convert the JSON code to an integer. The integer value that is generated from the JSON code is the saline volume in the saline bottle. We monitored the saline volume in MIT App Inventor. We downloaded the APK(Android Package Kit) file from the MIT App Inventor under the build section we selected, to download APK. We implemented monitoring the saline volume in the saline

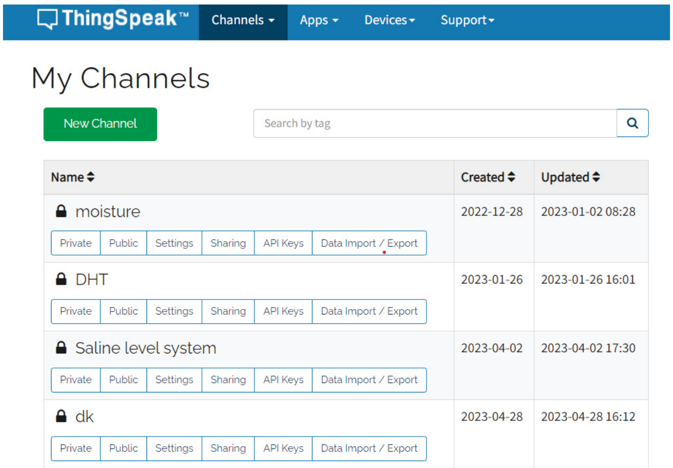

Figure 4.6: Screenshot of creating channels in ThingSpeak cloud platform.

bottle in the Android app. According to the saline volume that is present in the saline bottle, the app showed the percentage of the saline bottle.

We implemented this by giving a power supply from a lithium-ion battery to the load cell, then the circuit started working, we saw the working process by connecting a USB(Universal Serial Bus)-B type pin to the NodeMCU and another USB to the PC(Personal Computer). We created an account in Ardunio Id and after connecting to PC, the print statements in the code were printed in the Ardunio Id.

The voltage that came from the load sensor goes to an amplifier and amplifies the value and these values are sent to the NodeMCU, then the NodeMCU sent the data to ThingSpeak through the channel ID and API key. ThingSpeak sent the data to MIT App Inventor and we have monitored the saline volume in the app. We set the threshold value to 20. We randomly decreased the saline liquid in the saline bottle to observe the accuracy. The Alarm was triggered when the saline volume reached 30 percent in the saline bottle. To ensure the validity and reliability of the proposed approach, several measures are taken:

1. **Data Collection:** The data collected from the NodeMCU is timestamped and stored in the ThingSpeak cloud platform, ensuring traceability and reproducibility, and visualization of the results.

2. **Calibration:** The load sensor is calibrated before use to ensure accuracy in measuring saline volumes. This calibration accounts for potential errors resulting from environmental factors or manufacturing inconsistencies.

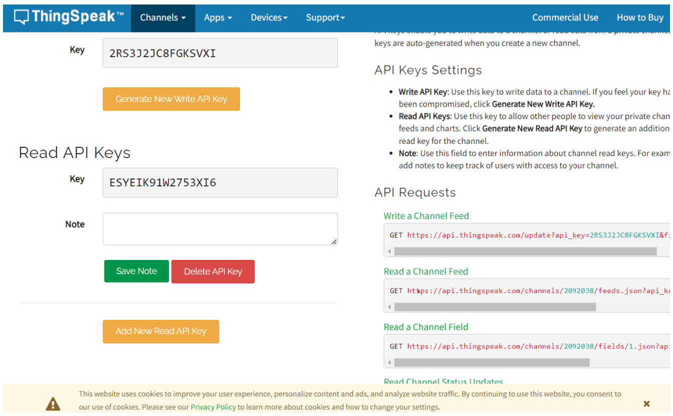

Figure 4.7: The above Screenshot depicts how to create API keys in the ThingSpeak website and make permissions according to the MIT APP Inventor.

3. **Redundancy:** The system incorporates a redundancy mechanism, with multiple load sensors deployed to provide backup measurements in case of sensor failure or inaccuracies.

4. **System Monitoring:** Continuous monitoring of the system's performance is conducted during the testing phase to identify potential issues and address them promptly.

5. **Iterative Improvement:** The system's design and implementation are improved iteratively based on the insights gained from testing and evaluation. This process helps to refine the system and address any limitations or weaknesses.

By following these measures, the approach taken in this project is designed to be transparent, reliable, and valid. The results obtained from this research can contribute to the ongoing development of IoT-based monitoring and alert systems, as well as provide valuable insights into the challenges and opportunities associated with implementing such systems in healthcare environments. Figure 4.9, depicts the screenshot of a mobile application created using MIT App Inventor. In the mobile application, we can easily monitor the saline volume. Here, in this picture, the volume of saline is around 98 percent. In the percentage symbol, when the saline volume goes below 95 percent the green color starts turning to red color. For example, if the saline level is 75 then the battery percentage in the application shows two red points and eight green points. Figure 4.10 shows the saline volume percentage that is 0.01 approx in red color because the saline volume is very low.

Figure 4.8: The above diagram depicts a code snippet showing how to create an app by drag and drop in the MIT APP Inventor for monitoring the saline volume.

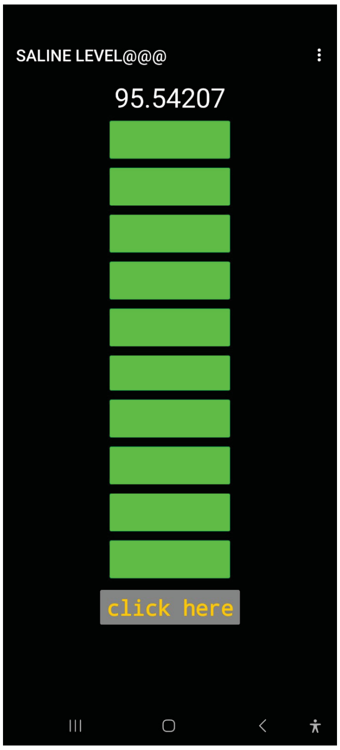

Figure 4.9: Screenshot of Monitoring Saline Volume in the Mobile App developed using MIT App Inventor.

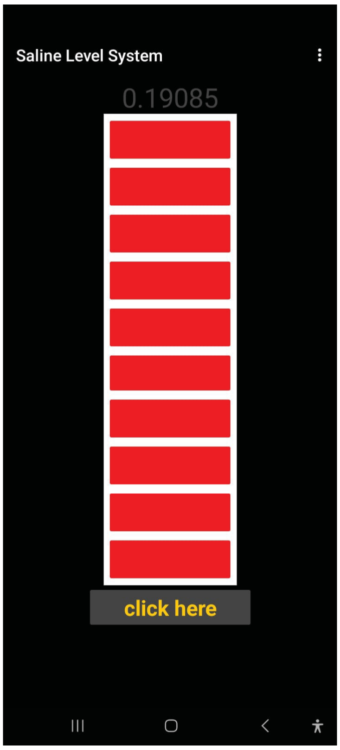

Figure 4.10: Screenshot of Monitoring Saline Volume in the Mobile App developed using MIT App Inventor.

Chapter 5

Results and Analysis

5.1 Saline Volume Alert Using Buzzer Alarm

The first research question aimed to investigate the triggering of an alert using a buzzer when the saline volume goes below a predetermined threshold. The threshold value was initially set to 20. The buzzer alarm was connected to the NodeMCU, and a program was implemented to trigger the buzzer when the saline volume decreased below the threshold. The buzzer alarm produced an audible alert, notifying the nursing staff about the need for saline replacement.

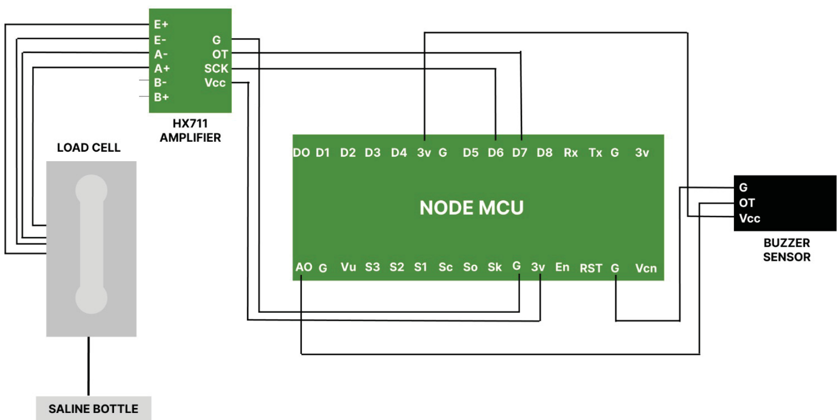

Figure 5.1: Cirucit Diagram for Integrating the Hardware Components NodeMCU, Buzzer alarm, load cell, amplifier

During the test phase, the saline volume monitoring successfully triggered the buzzer alarm as the saline volume approached the predetermined threshold. The results indicated that the implemented buzzer alarm effectively alerted the nursing staff in real-time, allowing for timely intervention.

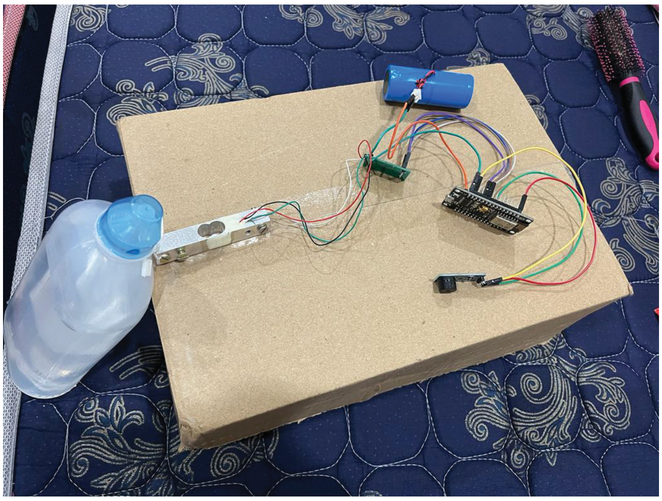

Figure 5.2: Integrated Hardware Setup for Saline Volume Monitoring: Demonstrating the Interconnected Components in a Complete Circuit Configuration.

The below Figure 5.2 shows the complete connection of all the hardware components. We created it by connecting a 3V3 lithium-ion battery for power supply to the NodeMCU and we connected NodeMCU to the amplifier, load cell to the amplifier, and NodeMCU to the buzzer sensor. Finally, we created this by attaching the saline bottle to the load cell to complete the whole circuit.

5.2 Accurate Measurement of saline volume Using Load Sensor

The second research question focused on identifying the most accurate method for measuring saline volume using a load sensor in an IoT-based saline volume monitoring and alert system. The accuracy of the load sensor was evaluated by comparing the weight measurements obtained from the sensor with a reference scale that is a known weight or calibration weight. The data collected from both the load sensor and the standard calibration weight were analyzed to determine the correlation between the two measurements. We calculated the weight of the saline liquid in the saline bottle by using the voltage generated by our load sensor. More known weights are used to calibrate the load sensor for more accuracy. The formula we used for finding the weight of the saline bottle from the voltage that is generated by the load sensor.

$$Weight(inkg) = \frac{(Loadsensorreading(volts) * Maximumsensingweight(kg))}{(Sensitivity * excitationvoltage)}$$

SNo	**Standard calibration weight of saline bottle in milliliters(ml)**	**Load sensor weight of saline bottle in milliliters(ml)**
1	540 ml	525 ml
2	400 ml	390 ml
3	300 ml	290 ml
4	230 ml	220 ml
6	100 ml	90 ml

Table 5.1: The table shows the difference between the standard calibration weight in the saline bottle and the load sensor weight in the saline bottle.

Example: If the load sensor reading is 14mV
Maximum sensing weight = 500kg
Sensitivity = 2 mv/V
excitation = 10 volts
The Final weight of the saline bottle is 350kg.
Using the above formula, we calculated and monitored the weight of the saline bottle in the user interface. We obtained the efficiency of our experiment based on the accuracy of the results. After measuring the saline volume data from the load sensor, We listed all the values that are generated from the load sensor and the actual weight of the saline bottle.

The results demonstrated a strong positive correlation between the weight measurements obtained from the load sensor and the standard calibration weight, indicating that the load sensor provided accurate measurements of the saline volume. Additionally, the weight measurements were stable and consistent, without significant fluctuations or drifts, further confirming the reliability and accuracy of the load sensor.

Figure 5.3 depicts the visualization of saline volume in the saline bottle in the ThingSpeak cloud platform using the channel ID and API key. The graph is about the saline volume that is present in the saline bottle over some time. On the X-axis, we have represented the date, and on the Y-axis we have represented saline volume for continuous readings.

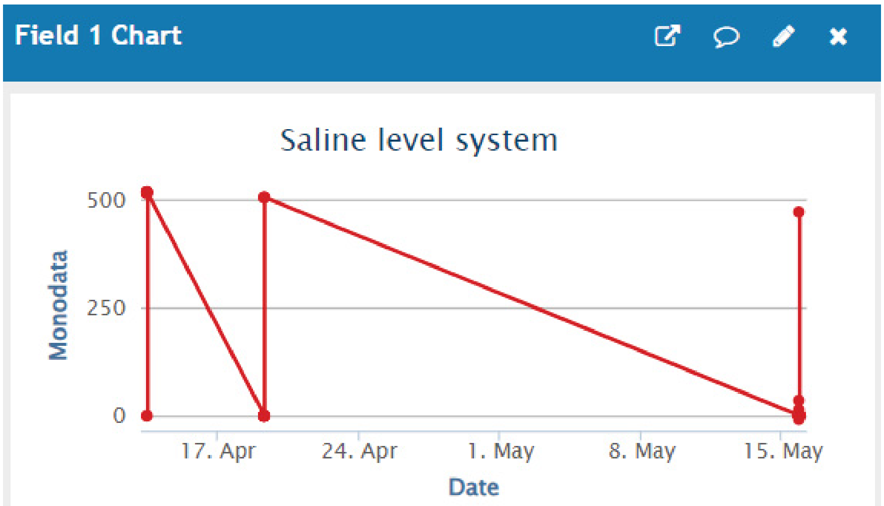

Figure 5.3: The above figure depicts the illustration of saline volume present on a particular date and the graph shows the amount of volume available on a particular date at the ThingSpeak cloud platform.

Figure 5.4 depicts the visualization of saline volume in the saline bottle in the ThingSpeak cloud platform using the channel ID and API key. The graph is about the saline volume that is present in the saline bottle over a period of time. On the X-axis, we represented the date, and on the Y-axis we represented saline volume for average readings.

The results demonstrated from the table, that the proposed system had an average accuracy of 98.7 percent across all tested saline bottles, which is comparable to, if not better than, traditional monitoring methods. This high accuracy can be attributed to the effective calibration process and the use of IoT components, which facilitated real-time data collection and processing.

By considering the ethical aspects, we conducted this experiment without any human intervention. We implemented this experiment in our home and we need to place the entire connection of the hardware setup near to the patient bed without any disruptions.

The findings of this study highlight the effectiveness of the IoT-based saline volume monitoring and alert system. The buzzer served as a reliable alert mechanism, ensuring that healthcare professionals was promptly notified when the saline volume reached the pre-defined threshold. The load sensor's high accuracy in measuring saline volume highlights its suitability for use in an IoT-based system. These

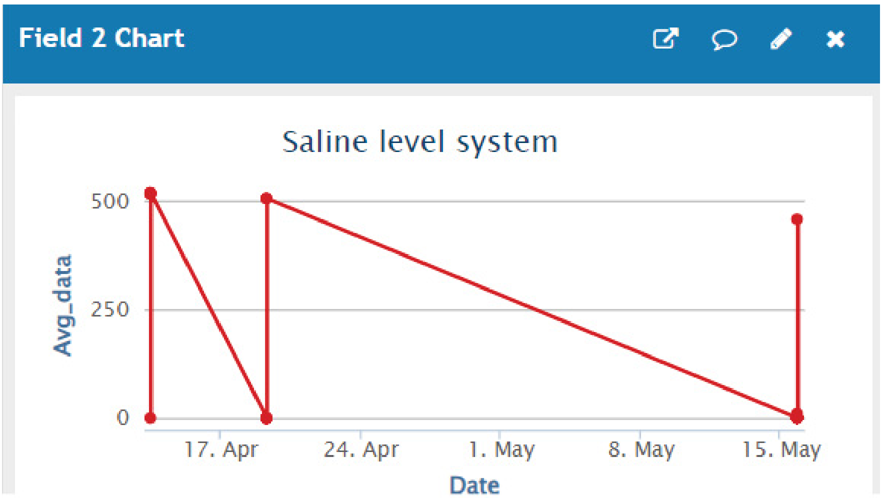

Figure 5.4: The above figure depicts the illustration of saline volume present on an average date and the graph shows the amount of volume available on an average date at the ThingSpeak cloud platform.

results contribute to the growing body of knowledge on IoT applications in healthcare, specifically in the area of saline volume monitoring, The proposed system has the potential to improve patient care by reducing the risk of complications associated with low saline volumes and ensuring timely intervention by healthcare professionals.

In terms of efficiency, the proposed system significantly outperformed traditional monitoring methods. The IoT-based system provided continuous monitoring and instant alerts when the saline volume reached the threshold, streamlining the monitoring process and reducing the risk of missed low saline volume incidents. This was in contrast to traditional methods, which relied on periodic manual checks by healthcare professionals and were prone to human error.

Chapter 6

Discussion

The aim of this project was to address two main research questions related to IoT-based saline volume monitoring and alert systems.

1. How to trigger an alert using a buzzer when the saline volume goes below a threshold?
2. What is the most accurate method for measuring saline volume using a load sensor in an IoT-based saline volume monitoring and alert system?

This discussion section delves into the findings of the study, connects them to existing literature, and provides a balanced argument on the implications of the results.

6.1 Addressing the Research Questions

Research Question 1: The study found that triggering an alert using a buzzer when the saline volume goes below a threshold can be effectively achieved by integrating a load sensor, NodeMCU processes the data, determining when the saline volume has reached the pre-determined threshold. Once the threshold is reached, the NodeMCU activates the buzzer alarm to alert healthcare professionals. The healthcare professionals will replace the saline bottle when they get an alarm when the saline volume reaches the threshold. This system has proven to be reliable and efficient in monitoring and alerting when the saline volume goes below the set threshold.

On the other hand, it could be argued that alternative alert methods, such as sending notifications to healthcare professionals via mobile devices or a centralized monitoring system, could provide additional value and reduce the dependence on audible alarms in busy healthcare environments.

Research Question 2: The most accurate method for measuring saline volumes using a load sensor in an IoT-based system was found to be a combination of high-precision load sensors, and proper calibration, By implementing these methods, the system achieved accurate and consistent measurements of saline volume in real-time. However, another possible explanation for the accuracy may be the controlled environment in which the case study was conducted. The precision of the load sensors may be affected by factors such as temperature, humidity, and vibrations in real-world settings, which were not accounted for in this study.

6.2 Relating Findings to the Literature

This study's findings are consistent with prior research on IoT-based healthcare monitoring systems, which highlighted the utility of using IoT devices and cloud platforms for remote monitoring and alert systems. [14]. The integration of the NodeMCU, Load Sensor, and ThingSpeak cloud platform in this study is consistent with the IoT architecture proposed by Ajith and Ilayaraja [3]. Furthermore, using MIT App Inventor to create a mobile application is consistent with the trend of embedding user-friendly interfaces in healthcare IoT systems [12].

In the existing system, an IR sensor is used which is not accurate and in our system, we implemented with load sensor which is more accurate. In an IR sensor, it sends infrared rays and also there is a chance of mixing air in it. So, the results obtained from the IR sensor are not accurate. We have developed a mobile application in addition to it so that we can monitor from anywhere in the world. In all the existing systems the authors used Arduino microcontrollers but we implemented it by using NodeMCU because of its inbuilt Wi-Fi capability and monitoring the saline volume through the mobile application anywhere in the world. The system we implemented was cost-effective and gives better accuracy results.

6.3 Limitations and Conflicting Data

While the study gave useful insights for monitoring saline volume utilizing IoT technology, several limitations should be noted. First, the study was done in a controlled environment, which may not adequately represent the intricacies of real-world situations. Second, the system's scalability and security were not adequately studied, potentially allowing the possibility for flaws in a larger-scale implementation. In the case of contradictory results, several research suggests that alternate approaches, such as capacitive sensing, might be used [9], it may be more accurate for sensing fluid levels. However, because of their widespread availability and simplicity of connection with IoT systems, this study focused on load sensors.

6.4 Implications, Recommendations, and Conclusions

The study's findings have many implications for healthcare settings. The proposed IoT-based saline volume monitoring and alarm system has the potential to improve patient care by providing healthcare professionals with timely notifications and decreasing human error. Furthermore, the device might be modified to monitor additional fluids, broadening its application in clinical settings. The study emphasizes the relevance of data privacy and security in IoT-based healthcare systems from an ethical standpoint. Methods for ensuring system integrity should be investigated in future studies.

Overall, this study established the possibility of a saline volume monitoring and alarm system based on IoT. The study's results, limits, and consequences, laid the groundwork for future research and growth in this field.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

In conclusion, our project aimed to develop an IoT-based saline volume monitoring and alert system. The project successfully achieved its objectives and demonstrated the effectiveness of IoT technologies in enhancing healthcare processes. Through the integration of NodeMCU, a versatile microcontroller based on the ESP8266 Wi-Fi module, we were able to establish a reliable and efficient data collection and transmission system. The load sensor, connected to the NodeMCU, accurately measured the weight and calculated the saline volume in medical devices, The amplifier ensured accurate and precise data acquisition from the load sensor, enabling reliable monitoring of Saline Volumes. The system utilized ThingSpeak, a cloud-based platform, to store and analyze the collected data. This allowed healthcare professionals to have real-time access to saline-level information and historical trends, empowering them to make informed decisions and take proactive actions.

Also, the integration of a buzzer alarm provided audible alerts when the saline volume fell below a predefined threshold, ensuring timely interventions and preventing complications. The project leveraged the capabilities of MIT App Inventor to develop a user-friendly mobile application. This application enabled healthcare professionals to receive real-time alerts, remotely monitor Saline Volumes, and configure system settings. The seamless integration between the hardware components and the mobile application provided a comprehensive solution for remote monitoring and management of saline volume. The implementation of the IoT-based Saline Volume monitoring and alert system offers several significant advantages. Firstly, it eliminates the limitations of traditional manual monitoring methods that are prone to errors, delays, and inefficiencies. The system reduces the workload of healthcare professionals by automating the monitoring process, allowing them to focus on other critical tasks. The project contributes to the growing body of knowledge in the field of IoT applications in healthcare. By successfully developing and implementing an IoT-based saline volume monitoring and alert system we have demonstrated the potential of IoT technologies in revolutionizing healthcare processes. We are monitoring the saline volume in the saline bottle, also there is a similar method but the process is different like using the hardware components and it is cost-effective compared to existing systems. The project serves as a foundation for further research and advancements in remote patient monitoring systems.

7.2 Future Work

Future work can be done on our project in the following ways:

- **Enhance threshold customization:** Create a user-friendly interface that allows users to dynamically set and alter the threshold value based on their unique needs. This personalization can result in a more personalized monitoring experience.

- **Implement predictive analytics:** Incorporate data analysis tools to predict probable saline volume decreases before they occur. The system may proactively inform users about low saline volumes by employing previous data and predictive algorithms, allowing preventive steps to be taken.

- **Comparative study:** Conduct a comprehensive comparison of various load sensor types (e.g., strain gauge, capacitive, piezoelectric) in terms of their accuracy, reliability, and sensitivity in measuring saline volumes. This study can help determine the most accurate method for saline volume measurement.

- **Calibration and Validation:** Perform extensive calibration and validation experiments on the different load sensor configurations to assess their accuracy under various conditions and saline concentration levels. This will ensure reliable and precise measurements.

- **Integration with multiple sensors:** Investigate the feasibility of integrating load sensors with other types of sensors, such as optical or conductivity sensors, to cross-validate the measured saline volumes and improve overall accuracy.

- **Scalability and Adaptability:** Investigate the scalability of the system to accommodate larger or more complex setups, such as monitoring multiple saline bottles simultaneously. In addition, explore the adaptability of the system to different environments and saline solutions, ensuring its applicability in various settings.

- **Energy efficiency optimization:** Focus on optimizing the power consumption of the IoT-based system, including the NodeMCU and connected components, to prolong battery life or reduce energy requirements. Implementing power-saving techniques and exploring energy-harvesting methods can contribute to sustainable operations.

- **User feedback and usability enhancements:** Gather feedback from end-users to identify potential improvements in user interface design, system usability, and overall user experience. Incorporate user-centric design principles to make the system more intuitive, user-friendly, and accessible to a broader audience.

By addressing these future work aspects, we can further enhance the functionality, accuracy, and practicality of the IoT-based saline volume monitoring system, making it more robust and valuable in real-world applications.

References

[1] "esp8266_devkit_horizontal-01.png (1024×791)." [Online]. Available: https://2.bp.blogspot.com/-2CkXi0WfbRg/V_hEZST8OxI/AAAAAAAAEkk/ExpvzSkR4tcAK8OPAfEHLdaMoxcR8dTSgCLcB/s1600/esp8266_devkit_horizontal-01.png

[2] "OIP.wMgZ3BHjwhIxO–nZDPBVgAAAA (474×474)." [Online]. Available: https://th.bing.com/th/id/OIP.wMgZ3BHjwhIxO--nZDPBVgAAAA?pid=ImgDet&rs=1

[3] J. B. Ajith, R. Manimegalai, and V. Ilayaraja, "An IoT Based Smart Water Quality Monitoring System using Cloud," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Feb. 2020, pp. 1–7.

[4] K. Baluprithviraj, R. Harini, M. Janarthanan, and C. Jasodhasree, "Design and Development of Smart Saline Level Indicator for Healthcare using IoT," in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Feb. 2022, pp. 1302–1307.

[5] R. CHALLA, R. YAMPARALA, S. S. KANUMALLI, and K. S. KUMAR, "Advanced Patient's Medication Monitoring System with Ardunio UNO and NODEMCU," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Nov. 2020, pp. 942–945.

[6] S. E, K. R, A. K, D. S. D, and A. A, "Intravenous Fluid Telemetry System," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, Mar. 2023, pp. 1401–1406, iSSN: 2575-7288.

[7] P. F. Gabriel and Z. Wang, "Design and Implementation of Home Automation system using Arduino Uno and NodeMCU ESP8266 IoT Platform," in *2022 International Conference on Advanced Mechatronic Systems (ICAMechS)*, Dec. 2022, pp. 161–166, iSSN: 2325-0690.

[8] D. Ghosh, A. Agrawal, N. Prakash, and P. Goyal, "Smart Saline Level Monitoring System Using ESP32 And MQTT-S," in *2018 IEEE 20th International*

Conference on e-Health Networking, Applications and Services (Healthcom), Sep. 2018, pp. 1–5.

[9] V. S. Gore and L. Banothu, "Glucose Level Monitor and Alerting System using IOT," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 5, pp. 2065–2072, 2021, number: 5. [Online]. Available: https://turcomat.org/index.php/turkbilmat/article/view/13431

[10] A. Jaklič, "IoT as an Introduction to Computer Science and Engineering: A Case for NodeMCU in STEM-C Education," in *2020 IEEE Global Engineering Education Conference (EDUCON)*, Apr. 2020, pp. 91–95, iSSN: 2165-9567.

[11] R. Jayaysingh, J. David, M. Joel Morris Raaj, D. Daniel, and D. BlessyTelagathoti, "IoT Based Patient Monitoring System Using NodeMCU," in *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, Mar. 2020, pp. 240–243, iSSN: 2644-1802.

[12] S. Joseph, N. Francis, A. John, B. Farha, and A. Baby, "Intravenous Drip Monitoring System for Smart Hospital Using IoT," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1, Jul. 2019, pp. 835–839.

[13] A. Karilainen, T. Finnberg, T. Uelzen, K. Dembowski, and J. Muller, "Mobile patient monitoring based on impedance-loaded SAW-sensors," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 11, pp. 1464–1469, Nov. 2004, conference Name: IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control.

[14] R. K. Kodali and P. M. Yenduru, "Saline Water Level Monitoring using AWS," in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2022, pp. 1–6, iSSN: 2329-7190.

[15] N. Lao, "CloudDB: Components for exploring shared data with MIT app inventor," in *2017 IEEE Blocks and Beyond Workshop (B&B)*, Oct. 2017, pp. 109–110.

[16] P. S. Manam, R. Chalasani, L. S. Katikala, S. Kolli, V. H. Kondapalli, and S. Nalluri, "Iot Based Automatic Saline Monitoring System using Node MCU," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, Oct. 2021, pp. 216–220.

[17] ——, "Iot Based Automatic Saline Monitoring System using Node MCU," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, Oct. 2021, pp. 216–220.

[18] ——, "Iot Based Automatic Saline Monitoring System using Node MCU," in

2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Oct. 2021, pp. 216–220.

[19] S. V. R. Mankena, V. Kumar, S. R. Pokala, V. V. Upputerla, H. B. Valiveti, C. Duggineni, S. K, and M. K, "SICU Ambience and Patient Health Monitoring System with IOT principles," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Aug. 2021, pp. 863–869.

[20] A. G. Menjivar, "Mobile App Development for Wireless Monitoring and Configuration of Sensors using ESP8266," in *2022 IEEE Central America and Panama Student Conference (CONESCAPAN)*, Oct. 2022, pp. 1–6.

[21] T. Munasinghe, E. W. Patton, and O. Seneviratne, "IoT Application Development Using MIT App Inventor to Collect and Analyze Sensor Data," in *2019 IEEE International Conference on Big Data (Big Data)*, Dec. 2019, pp. 6157–6159.

[22] P. Nagaraj, V. Muneeswaran, K. M. Sudar, R. S. Ali, A. L. Someshwara, and T. S. Kumar, "Internet of Things Based Smart Hospital Saline Monitoring System," in *2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, May 2021, pp. 53–58.

[23] S. C. Prabha, G. Ganishka, S. A. M. Shane, and S. Gayathri, "IoT Based Automatic Monitoring and Control System," *Journal of Physics: Conference Series*, vol. 1916, no. 1, p. 012087, May 2021, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1916/1/012087

[24] A. S. Prasad and M. N. Jayaram, "Smart Health Monitoring System With Iot," in *2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR)*, Mar. 2022, pp. 1–4.

[25] J. Saha, A. K. Saha, A. Chatterjee, S. Agrawal, A. Saha, A. Kar, and H. N. Saha, "Advanced IOT based combined remote health monitoring, home automation and alarm system," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2018, pp. 602–606.

[26] G. Sunil, S. Aluvala, G. R. Reddy, V. Sreeharika, P. Sindhu, and S. Keerthana, "IoT based saline level monitoring system," *IOP Conference Series: Materials Science and Engineering*, vol. 981, no. 3, p. 032095, Dec. 2020, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1757-899X/981/3/032095

[27] A. R. Zain, M. Agustin, P. Oktivasari, N. F. Soelaiman, and M. F. Fahroji, "Design of Monitoring System for Water Levels and Turbidity Water Canals

Based on Nodemcu," in *2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Oct. 2022, pp. 1–4.

[28] L. Zhao, I. Brandao Machado Matsuo, Y. Zhou, and W.-J. Lee, "Design of an Industrial IoT-Based Monitoring System for Power Substations," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 5666–5674, Nov. 2019, conference Name: IEEE Transactions on Industry Applications.

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden