Applied Information Technology and Computer Science Vol. 5 No. 1 (2024) 410-426

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Web-Based IoT Flood Management System

Jia Xin Chew[1], Nayef Abdulwahab Mohammed Alduais [1]*,

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2024.05.01.023
Received 24 June 2023; Accepted 18 May 2024; Available online 30 August 2024

Abstract: Floods can affect the economy, ecology, and human life. It will destroy crops, properties, public facilities, and other assets and cause the loss of lives. An IoT-based flood management system is a web-based system developed with PHP and MySQL. It allowed users to register multiple IoT nodes, providing a user interface to monitor the data. This system used numerous sensors to display real-time water levels, rainfall, temperature, and humidity. The collected data is sent to the system's monitoring website. The Visual Studio Code, Arduino IDE, and ThingSpeak used as software while Arduino IDE, Arduino WeMos D1 UNO board, water level sensor (HC-SR04), temperature humidity sensor (DHT11), and rainfall sensor are hardware used to develop the system. The IoT hardware is built to test the system. Parallel development is used in this project to ensure the system developed and functioned properly to meet the user requirements.

Keywords: Arduino, flood, monitoring website, ultrasonic sensor

1. Introduction

Natural disasters can severely impact the environment, property, wildlife, and people's health in tables and figures. A flood is a body of water that rises, swells, and overflows onto land not usually covered by water. Frequent flooding causes include intense rainfall, abnormally high tides, tsunamis, dam failure, and rising retention river levels. [1]. A flood can severely affect the economy, environment, and human lives. The immediate impacts of floods include loss of life, property damage, crop destruction, livestock loss, and chronic health condition due to waterborne illnesses [2]. Flood management needs to be prioritized to minimize the impact of the flood on the environment and citizens.

Malaysia has experienced numerous floods, with disastrous results, especially in peninsular Malaysia. Many flood monitoring systems in rural areas use traditional ways to measure water levels and predict when a flood will occur. The resident looks for some anticipated water-raising events near the river. Besides, the insufficient information on early flood monitoring systems is due to the collected data needing to be updated and completed as they are poorly managed. This issue will reduce the accuracy of the data collection.

Therefore, an IoT-based flood management system is designed with sensors and IoT technologies to manage and monitor the data to be aware of the flood event. This system aims to analyze and develop a web-based flood management system with Internet of Things (IoT) technologies.

**Corresponding author: nayef@uthm.edu.my.*

The paper is divided into six sections. The first section detailed the introduction, and the second section discussed related work while Section 3 is about methodology used in the system. Section 4 is about system analysis and design. Section 5 is the conclusion and discussion. Lastly, Section 6 discusses the conclusion.

2. Related Work

2.1 Internet of Things (IoT)

The Internet of Things (IoT) is a concept in which any connected device can be transformed into an IoT device [3]. IoT technologies aim to improve system efficiency by simplifying procedures in numerous domains [4]. The Internet of Things can gather, share, receive, and respond to information. Firstly, sensors collect and transmit data so we can respond quickly. The following procedure begins when the device receives and responds to information. An IoT gadget will gather and react to data without the involvement of humans. Wired or wireless technologies can transmit and process data. Wireless data transmission uses cloud computing to process and store data. Edge computing collects and processes data on the devices themselves [5]. IoT devices are linked to cloud computing, making data accessible anywhere, anytime.

2.2 Studying of the existing system

Three existing systems which are related to the proposed system in this project are discussed in this chapter. The three existing systems are MyPublic Infobanjir, Disaster Information Management System(myDIMS) and National Flood Forecasting and Warning System Sungai Muar (NaFFWS). The characteristics and features of the existing systems are studied to compare the existing systems and proposed systems in this project. The insufficient of the existing system is analysed to apply the improvement opportunity to the proposed system. There are a few areas for improvement in the existing system, as they needed a alert notification to notify the user about the reading. As a result, a monitoring interface is designed to make it easier for users to access the website directly.

Table 1: Comparasion within existing systems and proposed system

System / Features	MyPublic InfoBanjir	Disaster Information Management System (myDIMS)	National Flood Forecasting and Warning System Sungai Muar (NaFFWS)	Proposed system (Iot-based Flood Management System)
Register and Login	√	√	√	√
Graph analysis	√	X	X	X
Real-time data logging	√	√	√	√
Water level reading	√	X	√	√
Rainfall reading	√	X	X	√
Temperature and humidity reading	X	X	X	√

Table 1: (cont)

System / Features	MyPublic InfoBanjir	Disaster Information Management System (myDIMS)	National Flood Forecasting and Warning System Sungai Muar (NaFFWS)	Proposed system (Iot-based Flood Management System)
Point of interest	√	√	√	X
Readings interface	√	√	√	√
Maps	√	√	√	X
List of emergency contacts	√	X	√	X
Alert Notification	√	X	X	√
Forum	X	X	√	X

3. Methodology/Framework

The parallel development methodology can independently and simultaneously create scenarios from the original concept to the final line of code. The parallel development methodology makes the application development process more efficient as each development is separated [9]. The development life cycle of this system consists of five phases which are requirements analysis phase, design phase, implementation phase, testing phase and maintenance phase. The functional and non-functional requirements were identified once the tasks were completed.

3.1 System Requirements Analaysis

System analysis should specifics system requirements. Requirement analysis identifies user needs. System requirements include evaluating, reporting, and approving. To develop a successful system, functional and non-functional requirements should be identified. Functional specifications enable the system to perform as planned. Tables 2 shows the functional specifications of the proposed system.

Table 2: Functional requirements of the proposed system

No	Module	Functionality
1	IoT node registration	This module allows the registration of the IoT node characteristics, such as the channel id and API key.
2	Register and Login	This module allows the user and admin to register, log in, and log out.
3	User management	This module allows the admin to perform create, read, update, and delete actions on the users. The user can edit user profile and update it.

Table 2: (cont)

No	Module	Functionality
4	Monitoring page	This module allows the user and admin to view the node id, entry id, channel id, latitude, longitude, water level, temperature, humidity and rainfall intensity and the status of registered nodes.
5	Report	This module allows the user and admin to view and print the report of the selected nodes.
6	Notification alert	This module lets users notify the pop-up message of the node's status if the parameters reach the threshold value.

4. System Analysis and Design

The sub-section has four sections. Section 4.1 is use case diagram, section 4.2 is sequence diagram, section 4.3 showed the activity diagram, section 4.4 is wiring diagram, section 4.5 for entity relationship diagram (ERD) and section 4.6 for user interface design.

4.1 Use Case Diagram

A use case diagram is a visual representation of a user's potential interactions to show the functions of the proposed system. Two actors are involved in this system: the user and the admin. Users can log in, edit profiles, and view water level, rainfall, temperature and humidity data. A user needs to register by admin to access the monitoring system. Furthermore, the admin can access the user management page to perform CRUD actions on users. Besides, the admin can register IoT nodes by channel id and API key to add the data of the nodes in the system. A notification alert will display the node's status if they are in a caution, warning, or dangerous situation.

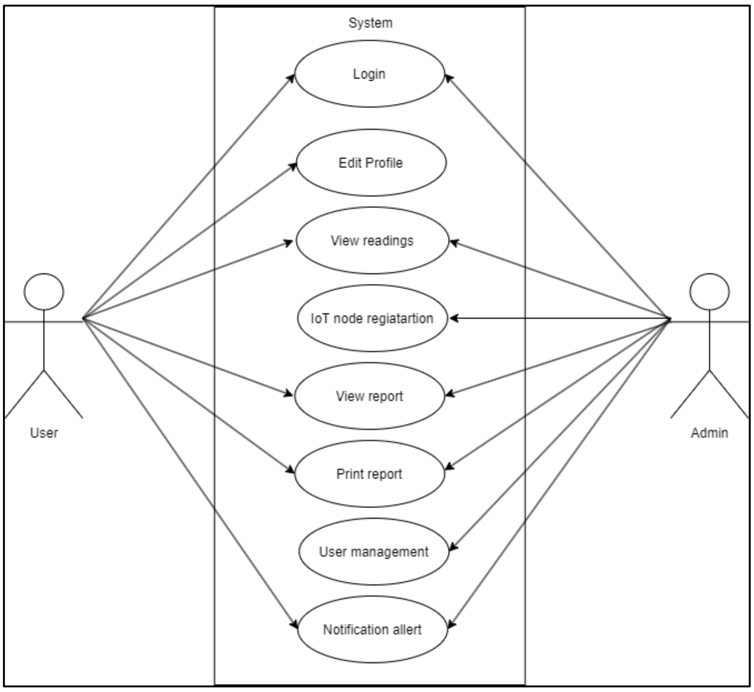


Figure 1: Use case diagram of the proposed system

4.2 Sequence Diagram

A sequence diagram is an interaction diagram that describes users' actions within the system. To access the monitoring website, a user or admin must log in with their user id and password. The login credentials are validated using the database after the user id and password are provided.

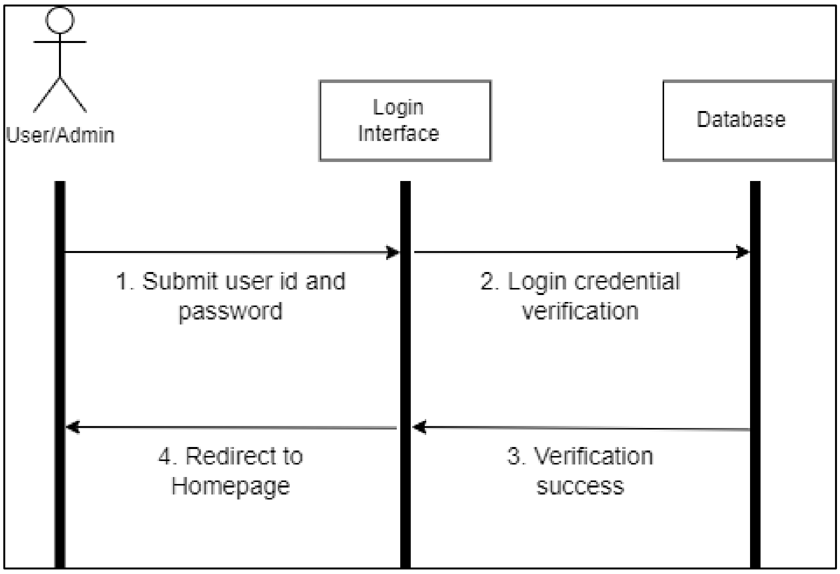


Figure 2: Sequence diagram of login interface for user and admin

The sequence diagram of the monitoring interface explains that the sensors set in the IoT node will collect water level readings, rainfall intensity, temperature, and humidity. ThingSpeak will store the collected data in a structured manner, associating each data with the corresponding IoT node. Furthermore, users and administrators can access the monitoring interface of the system. Users and admins can view the readings from each IoT node. They can see the latest data of each node.

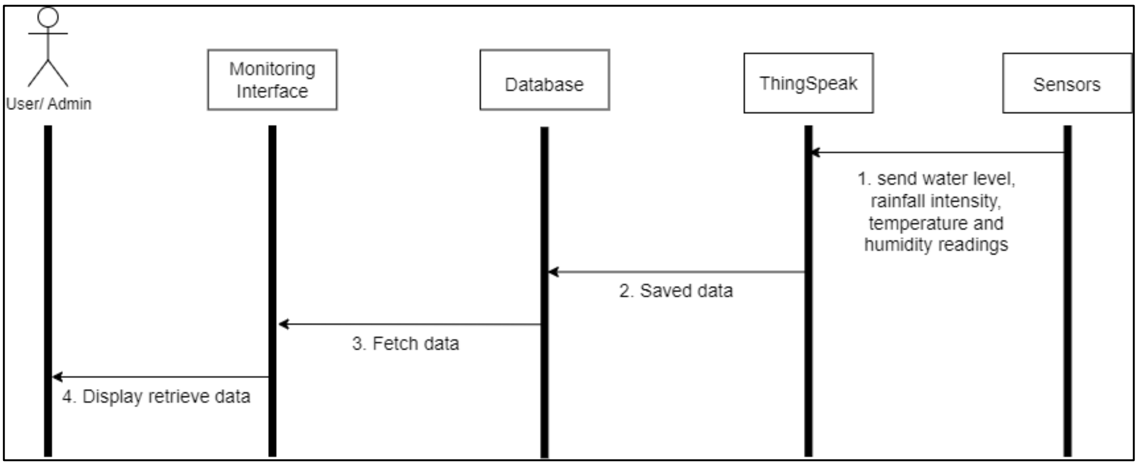


Figure 3: Sequence diagram of the monitoring interface

Firstly, the admin can register the user by inserting user details and setting the user role to enable the user to log in to the system. The system will save user records into the database to ensure easy access to the user information. Besides, the admin can disable or activate the user account to prevent the user from accessing the system. Admin can delete unused user accounts to ensure data security and manage the system effectively.

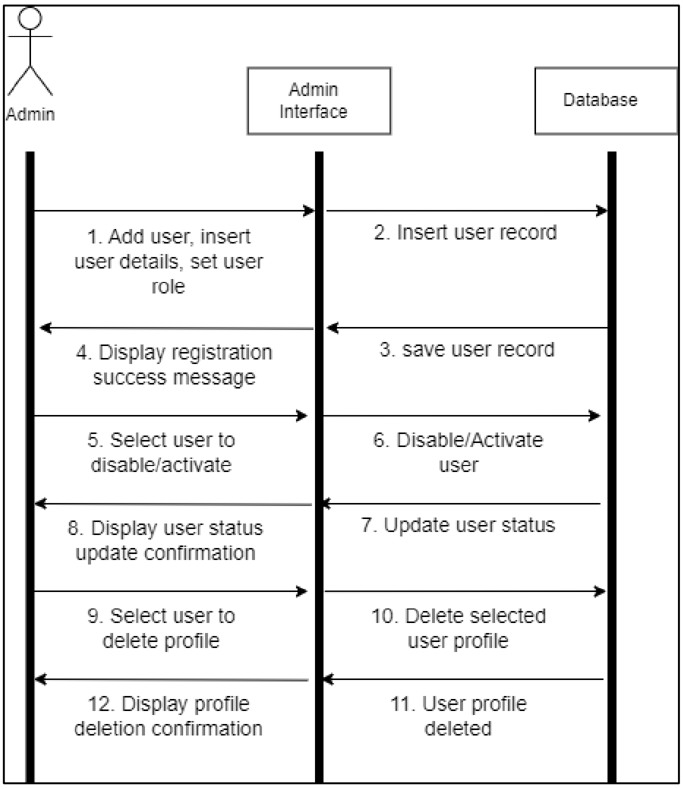


Figure 4: Sequence diagram of user management for admin

When users navigate to the report interface, they can select a report of the node from a dropdown menu. The database retrieves data from the node table and returns it to the user. The report details are then shown in a table to the user. If the user wants to print the report, they click the "Print" button, which opens the printable report in a new tab or window.

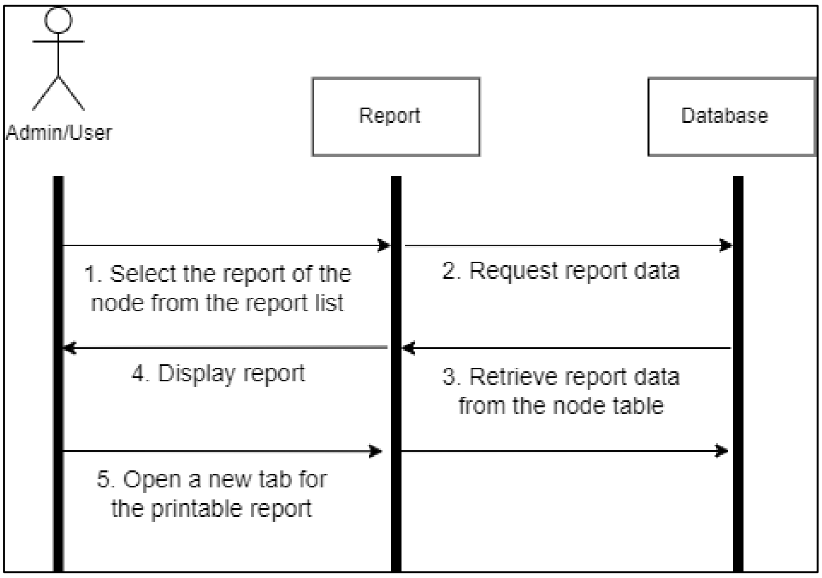


Figure 5: Sequence diagram of the monitoring interface

4.3 Activity Diagram

The activity diagram shows the actions carried out in the IoT-based flood management system. The user and admin need to log in to the system to access the monitoring website. If their login credentials are correct, they can access the website; otherwise, they will be redirected to the login page. The admin can manage users by adding, modifying, disabling, or deleting their accounts. Additionally, the admin can register IoT nodes by inserting their channel id and API key. Once logged in, users can update their profiles and change their passwords. Both admin and user can view water level, rainfall intensity, temperature, and humidity readings on the monitoring page. A notification will appear indicating the node's status if any of these readings exceed a specific limit. Users can generate and print reports by navigating to the report interface.

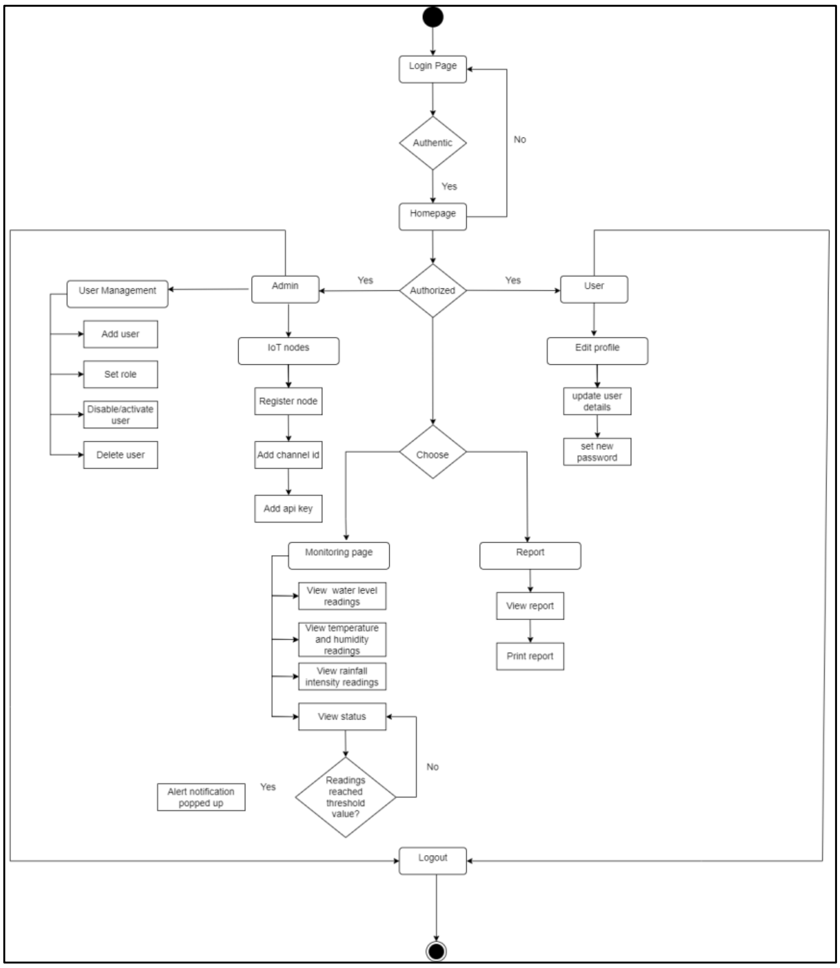


Figure 6: Activity diagram of IoT-based flood management system

4.4 Wiring Diagram

Wiring diagrams show the IoT system's physical connections. It shows the system's component arrangement to aid in development. Figure 7 shows the system's wiring diagram. The IoT hardware is built to test the system. Arduino WeMos R1 board, ultrasonic water level sensor (HC-SR04), temperature/humidity sensor (DHT11), and rainfall sensor. Coloured lines show component connections. HC-SR04 sensor is an ultrasonic distance sensor. It consists of a transmitter and receiver that works on the frequencies [6]. DHT11 sensor comes with a dedicated NTC and is very useful for measuring the surrounding temperature and humidity with a capacitive humidity sensor and thermostat [7]. A rainfall sensor is a device used for water detection to sense the rainfall. The sensor is a resistive dipole with lower resistance when water is an electrical conductor, and higher resistance when there are no raindrops on board [8].

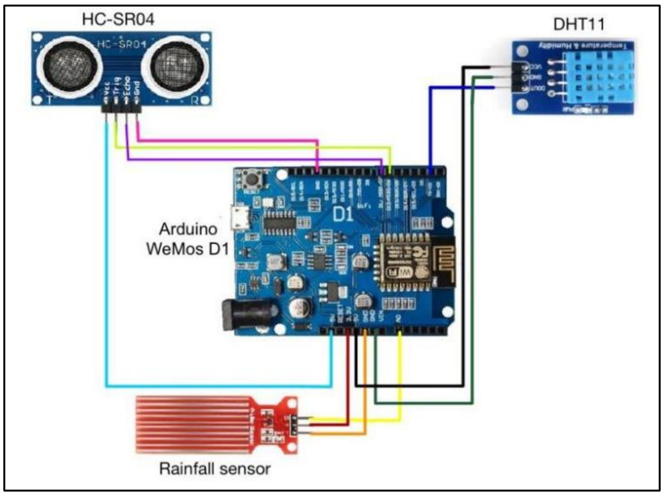


Figure 7: wiring diagram for the IoT-based flood management system

4.5 Entity Relationship Diagram (ERD)

In database design, the entity relationship diagram (ERD) is used. The three entities are user, roles, sensors, nodes, reports, and status. The primary key uniquely identifying a relational database table record while foreign key refers to a field in a table that is the primary key of another table. Each user has a unique role assigned by the admin. The nodes have sensors that are connected to them, and multiple nodes can be saved in the database. Each node has a single report associated with it, and the status of each node can change over time. Users can view and print multiple reports, and multiple users can access the same report.

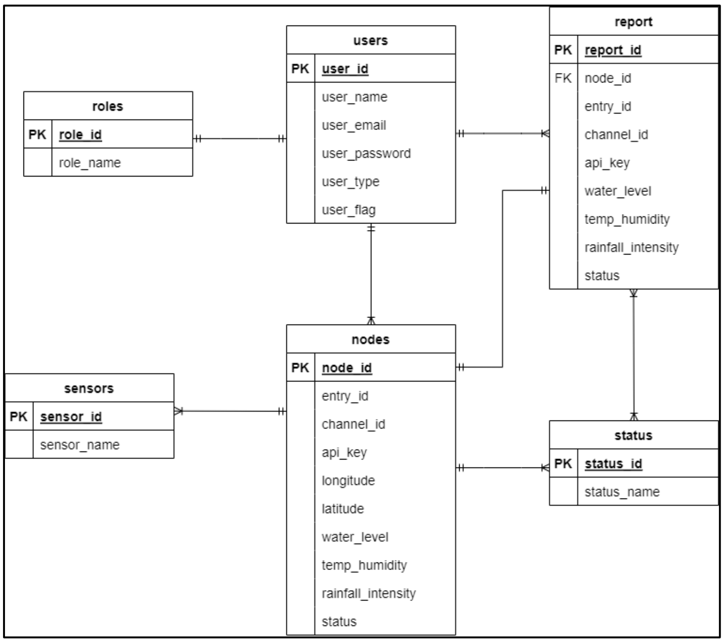
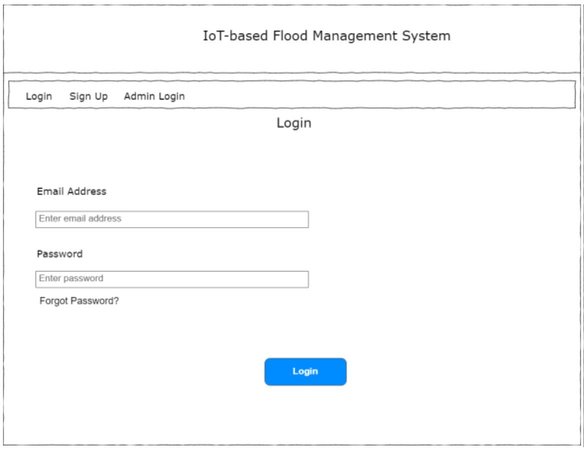


Figure 8: ERD diagram for the IoT-based flood management system

4.6 User Interface Design

User interface design is a visual arrangement that lets users interact with a website. Wireframes show the proposed system's page structure and style. Figure 9(a) shows an IoT-based flood management login page. Users and administrators need an email address and password to access the monitoring website. Users can register an account if they are new users. The sign-up form will show on this page. Figure 9(b) shows the sign-up form for the website.



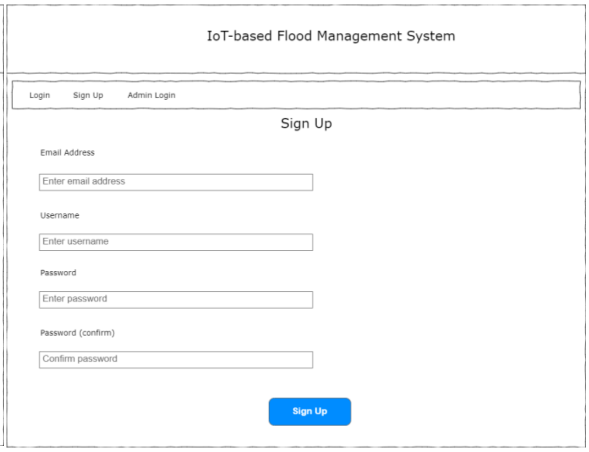
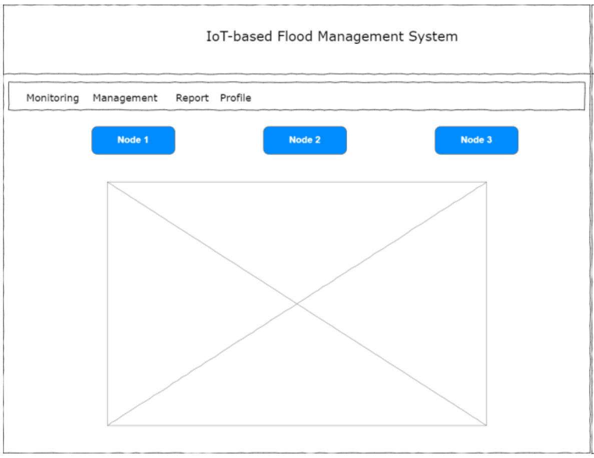


Figure 9(a): Login page **Figure 9(b): Sign-up page**

Figure 10(a) shows the monitoring webiste of the system.A monitoring website for an IoT-based flood management system allows users and administrators to view readings based on selected node. Users and administrators can view and manage the report page. Figure 10(b) shows the user interface of the report page.



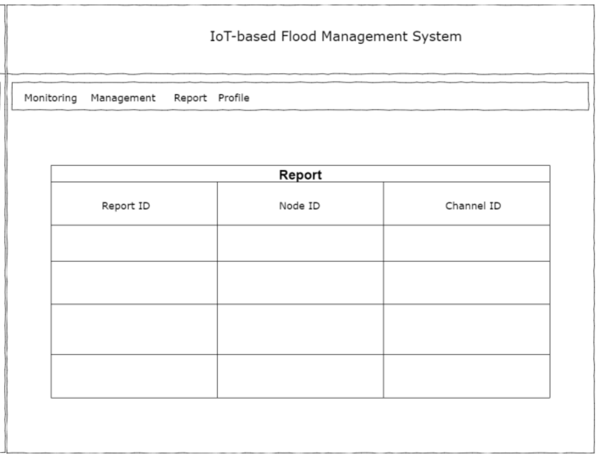
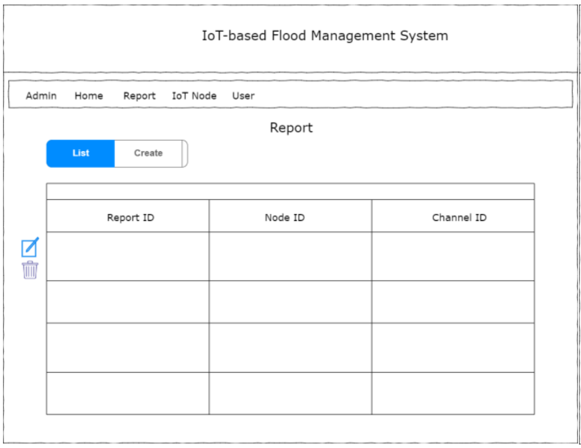


Figure 10(a): Monitoring Website **Figure 10(b): Report page**



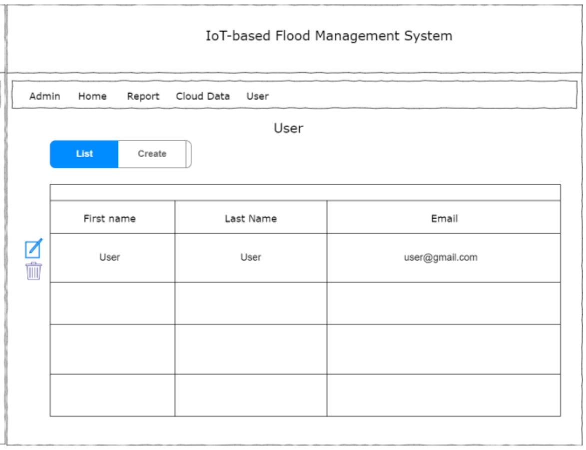


Figure 11(a): Report page **Figure 11(b): User management page**

After login to the system, the administrator can create, edit, delete, and update cloud data and user details. Figure 11(a) shows the report page of admin dashboard. Figure 11(b) shows the user management page of admin dashboard.

5. Result and Discussion

The sub section has 2 sections which are section 5.1 for test plan and section 5.2 for user interface.

5.1 Test Plan

System testing is software testing that aims to find faults in the application. It tests the system and the application to ensure that it functions properly and meets the desired goals. Below is the table which is the two categories in the test plan.

Table 3: Test category in test plan

Test category	Description
1	Test the functionality of the system will store and manipulate the data
2	Test the security of the system repulsing malicious outside attacks and assuring software application security

Table 4: Test category 1 in test plan

No	Module	Test	Expected result	Actual result
1	Login	• User can login to the system by entering username and password.	The user is logged in to the system if the user's input is valid, otherwise, the user should not be logged in.	Pass
2	User management	• Staff can view and update his/her profile. • Admin can add new users. • Admin can update user profile. • Admin can delete, activate, and disable user account.	User’s profile is shown with the staff’s personal information. Staff can insert their new information into the update form. Data is updated after clicking the update button. An error message of update failed is displayed if the staff insert the invalid data. Admin can add new user, activate, or disable user account and delete unused user account.	Pass
3	Node Registration	• Admin can insert node details. • Admin can add multiple nodes.	Admin can add channel id and API key of more than one node. The data of each node can be viewed in the monitoring page.	Pass

Table 4: (cont)

No	Module	Test	Expected result	Actual result
4	Monitoring website	• User can view the readings interface.	Users can view the water level readings, rainfall readings, and temperature and humidity readings. The user can view the location of each node by its longitude and latitude.	Pass
5	Report	• Users can view the report of the node. • User can print the report of the node	Users can view the node list in the report. The report is displayed in another tab and user can download the report of the selected node.	Pass
6	Notification Alert	• Users can view the status of each view. • An alert message will pop up if the parameters reached threshold value	Users can view the status of each node. An alert message popped up when the paramaters reached threshold value. The status is classified into 3 types: caution, warning, and danger. The alert does not pop up when the node is in normal state	Pass

Table 5: Test category 2 in test plan

No	Checklist	Actual result
1	The error message should not indicate which part of the authentication data is inappropriate. For example, an error message should not show “incorrect password” or “incorrect email.”	Pass
2	Ensure the security validation is complex. For instance, password creation requires alphabetic and numeric characters, while email input requires format validation.	Pass
3	Ensure the Password length under the password policy. For instance, a minimum of eight characters and a maximum of 64 characters.	Pass
4	Verify that each user role has the proper access credentials so that users can only use the significant functions and data for their roles and responsibilities.	Pass

5.2 User Acceptance Testing

User acceptability testing is a sort of testing used to evaluate the user experience with the system's functionality. The Likert scale ranges from 1 to 5, with 1 indicating strong dissatisfaction and 5 indicating high satisfaction. The results are shown in Figure 12.

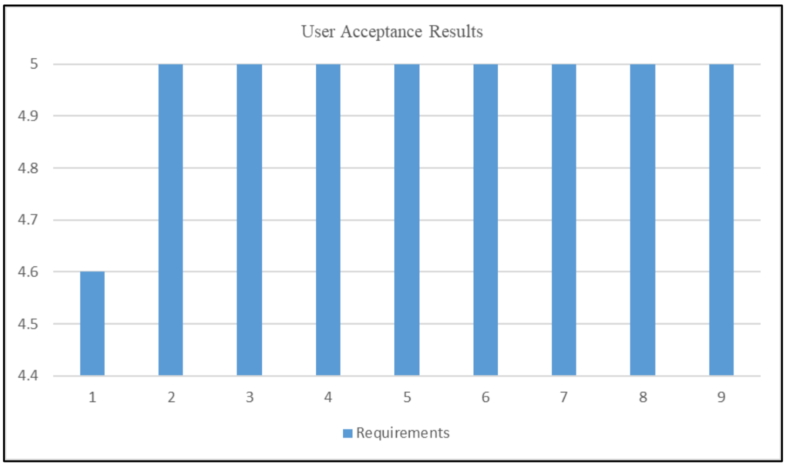


Figure 12: Result of user acceptance testing

5.3 User Interface

User Interface (UI) is the application part that a user can interact with. It controls how information is shown on the screen and how the user interacts with the application. Figures 13 to Figure 17 show the user interface of the system.

The login page for the web system application is shown in Figure 13. The user will use their username and password to log into the web-based application.

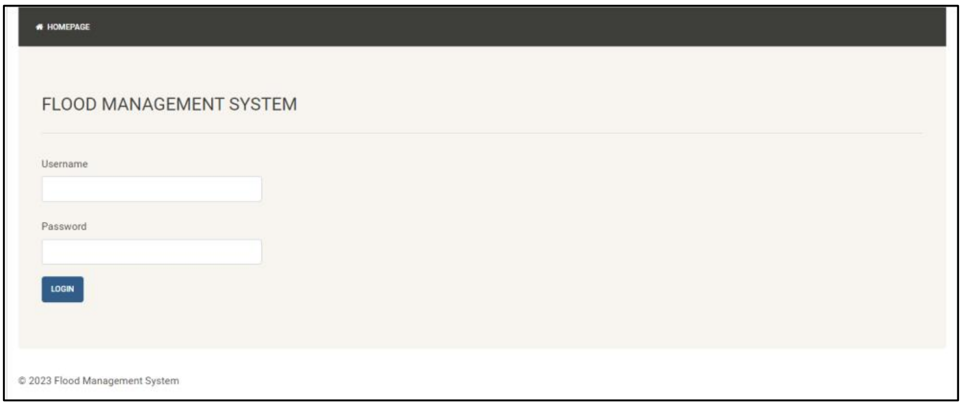


Figure 13: Login page

Figure 14 displays a dynamic dashboard page for the admin with various statistics related to nodes, while Figure 15 shows the user dashboard page.

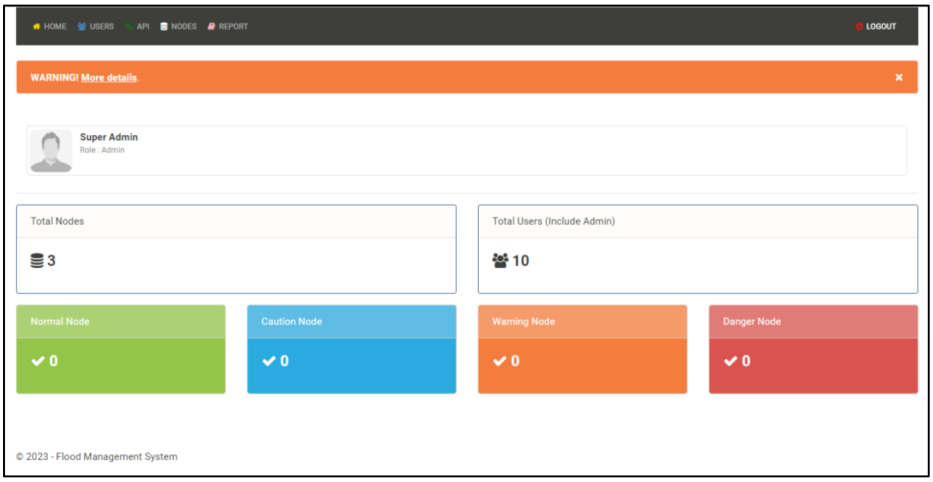


Figure 14: Admin dashboard

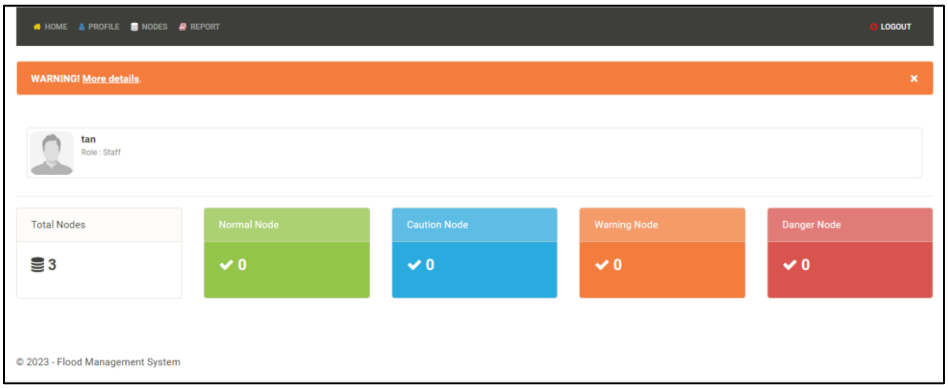


Figure 15: User dashboard

After logging into the system successfully, users can update their details on the profile page shown in Figure 16.

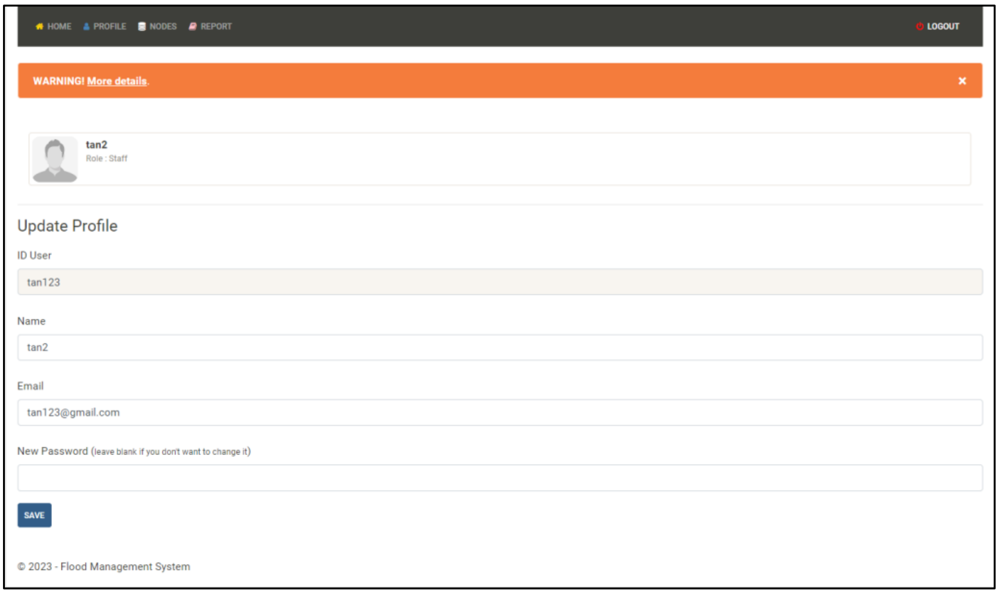


Figure 16: Profile page

Figure 17 displays the page for managing users in the web application system. Admin can perform various actions such as adding, editing, and deleting user accounts using this module.

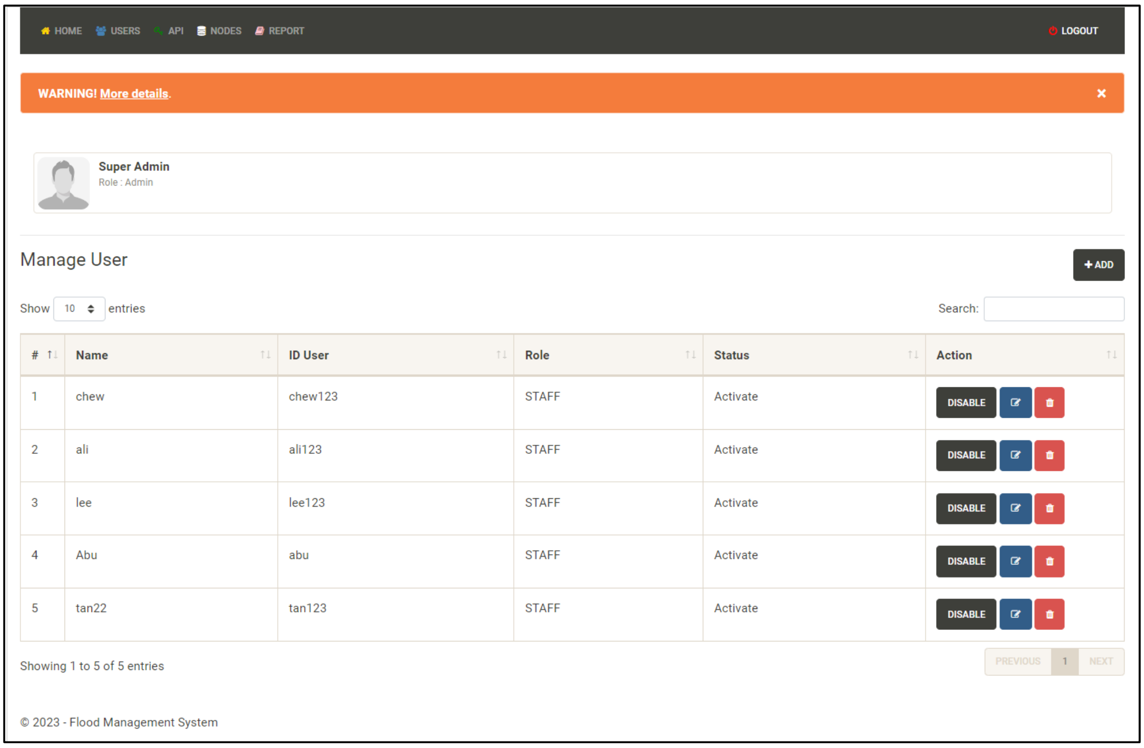


Figure 17: User management page

The API module is shown in Figure 18, which enables the admin to add API URLs. The admin can also add new nodes into the system by entering their channel id and API key. This module also allows multiple nodes to be stored in the database.

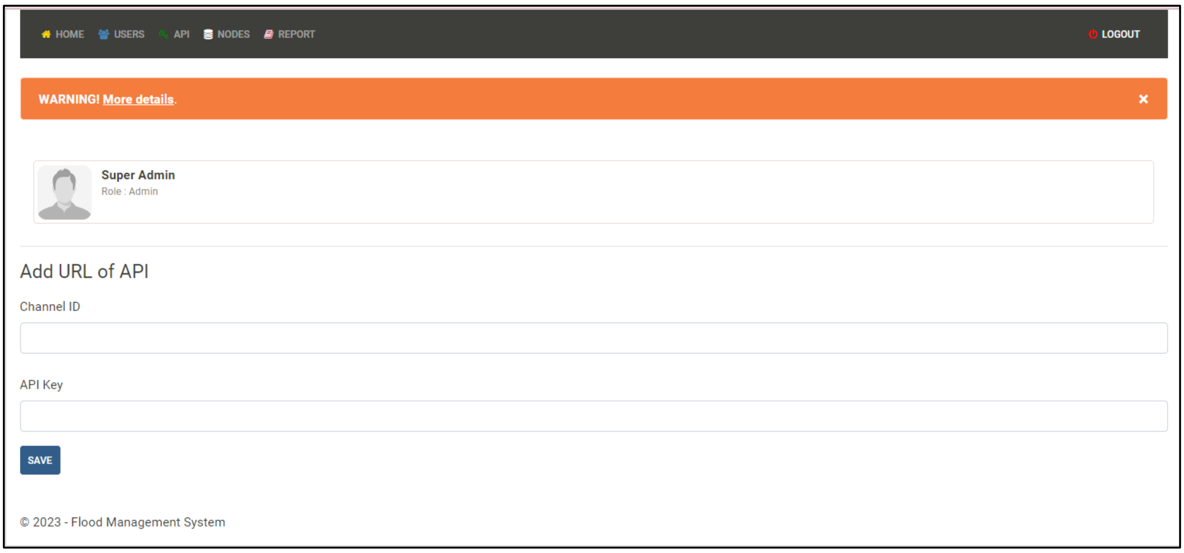


Figure 18: API page

In Figure 19, the node module allows both the administrator and user to view data from the "API" table. The users can sort the data in ascending or descending order and search for specific information by entering text or values into the search box.

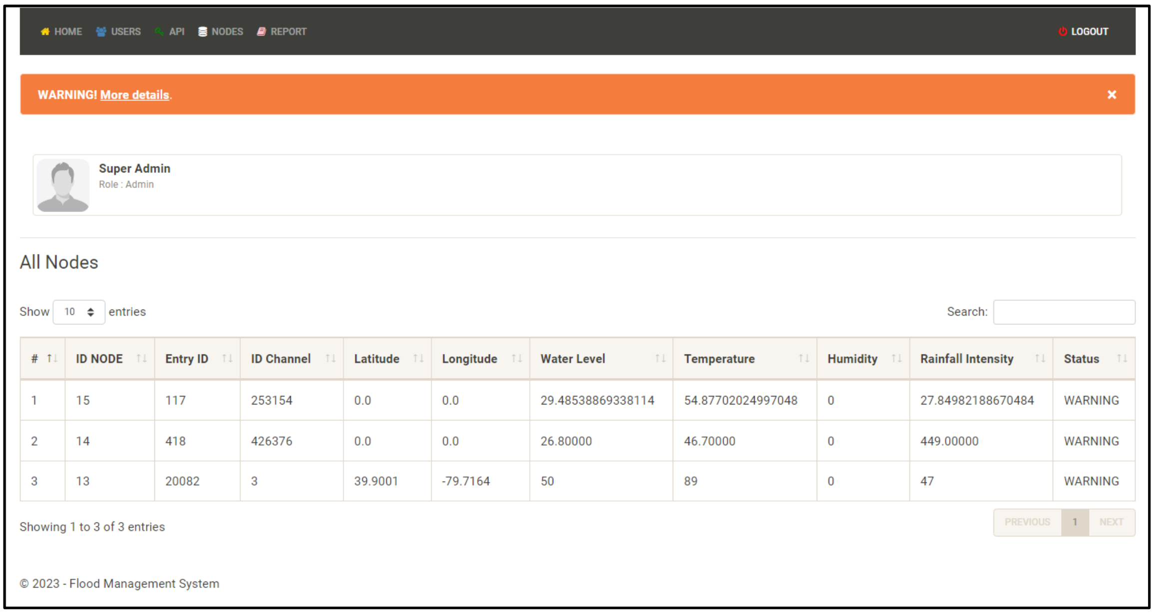


Figure 19: Nodes page

The alert message displayed in Figure 20 indicated the status of the nodes. The admin and user can be aware of the status of the node whether in caution, warning, or danger. The alert message indicated in all pages of the system; users can click on the "more details" to redirect to the nodes pages.

Figure 20: Alert message

The report page displayed in Figure 21 allowed the admin and user to view and select the report from a dropdown menu. Users can choose which node to view and print the report.

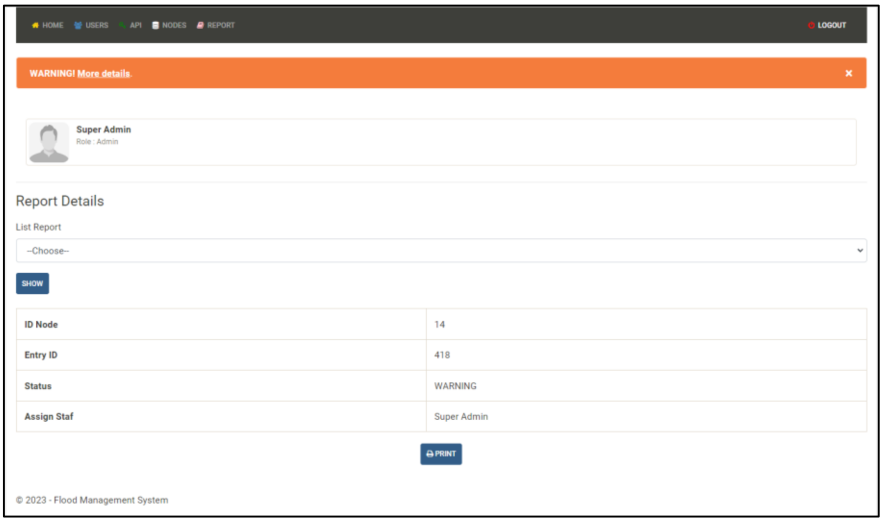


Figure 21: Report page

In Figure 22, the print report module can generate a report in a pdf file. This module allows users and admins to print the report. When users click the print button, a new window or tab is opened with the page.

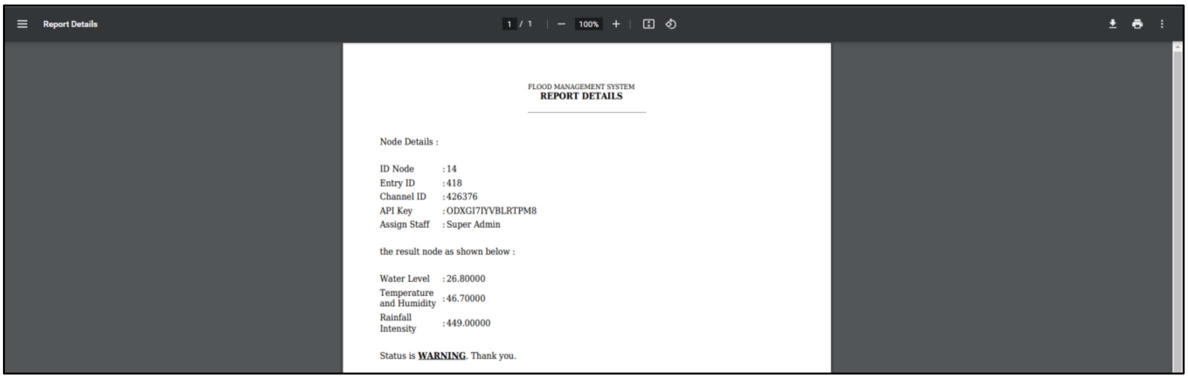


Figure 22: Print report page

6. Conclusion

In conclusion, an IoT-based flood management system was developed with IoT nodes. The goal of this system was to solve the problem of inconsistent data and unaware of the flood event. The target users of the system are staff and admin. The system was tested by test plan to ensure all modules are fully functional. Despite this, the system still needs some improvement, for example, the requirement that the interface needs to work with mobile and web browsers. Besides, users are unable to register their accounts by themselves. Hence, there are two recommendations to improve the system. Firstly, an interface for both mobile and web can be developed, while another suggestion is to add a page where people can sign up for an account. The admin should permit them to access the system before they log in to the system.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix A

Figure A shows the gantt chart of the proposed system.

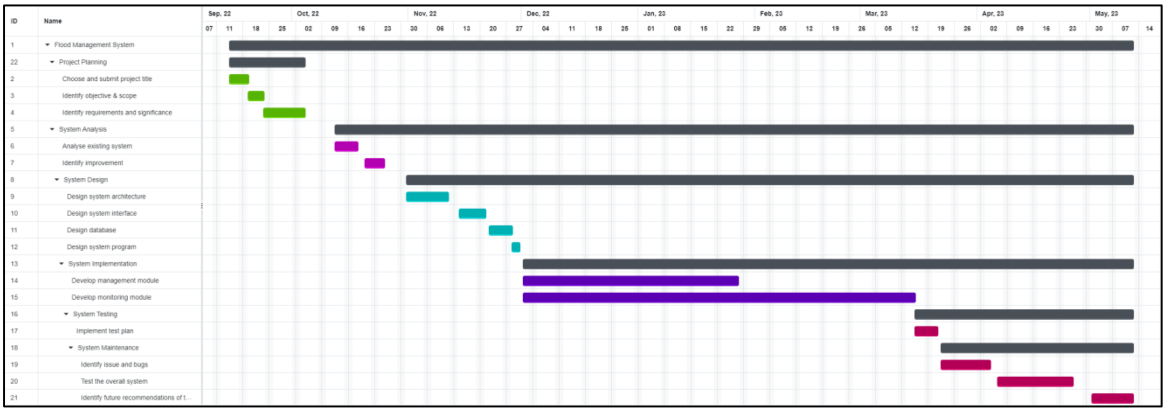


Figure A: Gantt Chart

References

[1] Alcoforado, F. “Flood control and its management”. Journal of Atmospheric & Earth Sciences, 2(1), pp. 1–13, 2018. doi: 10.24966/aes-8780/100005

[2] Bari, M. A., Alam, L., Alam, M. M., Rahman, L. F., & Pereira, J. J., “Estimation of losses and damages caused by flash floods” , 2021, Malaysia. Arabian Journal of Geosciences, 14(3). doi: 10.1007/s12517-021-06503-x

[3] Kumar, S., Tiwari, P., & Zymbler, M. “Internet of Things is a revolutionary approach for future technology enhancement”: a review. J Big Data 6, 2019, pp. 111, doi: 10.1186/s40537-019-0268-2

[4] Sandro, N., & Petar, S., Diego, L., & Luigi, P. “Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future.” Journal of Cleaner Production, 2020, pp274, doi: 10.1016/j.jclepro.2020.122877.

[5] Sahar, T., & Dogan., I. “A Review on Cloud Computing and Internet of Things”, International Journal of Computer, Electrical, Automation, Control and Information Engineering. 11, pp.462-467, 2017.

[6] Nadhir, I., Ihsan, J., & Nahla, A. “Investigating the resolution ability of the HC-SR04 ultrasonic sensor” , 2020, doi: 10.1088/1757-899X/745/1/012043

[7] Feresu, Z. “DHT11 Based Temperature and Humidity Measuring System”., J Electr Eng Electron Technol, 2022, doi: 10.4172/jeeet.1000902.

[8] Sabar., Anjani., Dewi., & Wijaya, “Water Level Detection System based on Arduino and LabVIEW for Flood Monitors using Virtual Instrumentation.” Al-Fiziya: Journal of Materials Science, Geophysics, Instrumentation and Theoretical Physics. 4, pp.29-35, 2021, doi: 10.15408/fiziya.v4i1.19808.

[9] Rosenberg, D., Boehm, B. W., Wang, B., & Qi, K. (2019). The parallel agile process: Applying parallel processing techniques to software engineering. Journal of Software, https://doi.org/10.1002/smr.2144