**BERA: BILAR EMERGENCY RESPONSE APPLICATION USING DYNAMIC CLUSTERING PROTOCOL**

A Thesis

Presented to the Faculty of the

College of Technology and Allied Sciences

BOHOL ISLAND STATE UNIVERSITY

Zamora Bilar Bohol

In Partial Fulfilment

Of the Requirements for the Degree

In Bachelor of Science in Computer Science

Armando C. Jumawid Jr.

Ronelyn G. Perocho

Mark Henry Perez

May 2024

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **TABLE OF CONTENTS** |  |
|  |  |  | **Page** |
| **TITLE PAGE** | | ..…………………………………………………… |  |
| **TABLE OF CONTENTS** | | …………………………………………………….. |  |
| **LIST OF TABLES** | | …………………………………………………….. |  |
| **LIST OF FIGURES** | | …………………………………………………….. |  |
| **LIST OF PREVIEWS** | | …………………………………………………….. |  |
| **ABSTRACT** |  | …………………………………………………….. |  |
| **Chapter 1** | **THE PROBLEM AND ITS SCOPE** | |  |
|  | Rationale ……………………………………………………. | |  |
|  | Literature Background …..………………………………... | |  |
|  | **THE PROBLEM** | |  |
|  | Statement of the Problem …………………………………… | |  |
|  | General Objective …………………………………..……... | |  |
|  | Scope and Delimitations.………………………………….. | |  |
|  | Significance of the Study ………………………………….. | |  |

|  |  |  |
| --- | --- | --- |
|  | **RESEARCH METHODOLOGY** |  |
|  | Development Framework ………………………………….… |  |
|  | Block Diagram…………………………………………………. |  |
|  | Development Model and Approaches………………………. |  |
|  | Purpose & Requirements ……………………………………. |  |
|  | Process Specification ………………………………………... |  |
|  | Software Development Tools ...…………………………….. |  |
|  | Environment and Participants………………………………. |  |
|  | Data Collection ………………………………………………. |  |
|  | Definition of Terms …………………………………………... |  |
| **2** | **PRESENTATION OF FINDINGS, ANALYSIS, AND INTERPRETATION OF DATA** | |
|  | Existing Operations and Processes………………………... |  |
|  | Needs of the Existing Operation.…………………………… |  |
|  | Use Case Diagram…………………………………………… |  |
|  | Use Case Narrative……………..……………………………. |  |
|  | Database Design…………………..…………………………. |  |
|  | Class Diagram………………………………………………… |  |
|  | Domain Model Specification…...……………………………. |  |
|  | Information Model Specification….…………………………. |  |
|  | Service Specification…..…………………………………….. |  |
|  | IoT Level Specification………………………………………. |  |
|  | Operational View Specification …………………………….. |  |
|  | Device Component Specification…………………………… |  |
|  | Application Development……………………………………. |  |
|  | Data Schema………………………..................................... |  |
|  | Database Structure………………………………………….. |  |
|  | Program Hierarchy………………………………................. |  |
|  | Technical Requirements.……………………………………. |  |
|  | Software Used …………..…………………………………... |  |
|  | Functional Requirement…………………………………….. |  |
|  | Non-Functional Requirement ……………………………… |  |
|  | Test Case…………………………………………………….. |  |
|  | Business Intelligence……………………………………….. |  |
|  | Screen Layout……………………………………………….. |  |
|  | Testing and Evaluation……………………………………… |  |
|  | Hosting Implementation…………………………………….. |  |
|  | Hardware Evaluation………………………………………... |  |
|  | Physical Layout……………………………………………… |  |
|  | Screen Layout……………………………………………….. |  |
|  | Testing and Evaluation……………………………………… |  |
|  | System Usability…………………………………………….. |  |
| **3** | **SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS** |
|  | Summary of Findings……………………………………….. |  |
|  | Conclusions………………………………………………….. |  |
|  | Recommendations………………………………………….. |  |
|  | **REFERENCES……………………………………………….** |  |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF TABLES** |  |
| **Table** |  | **Page** |
| 1 | Summary of Respondents in the System Usability….………... |  |
| 2 | Interpretative Guide of the System Usability………………….. |  |
| 3 | SQLyog Database………………………………………………... |  |
| 4 | Hardware Interface………………………………………………. |  |
| 5 | Hardware Cost…………………………………………………… |  |
| 6 | System Usability Assessment Result …………………………. |  |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Figure** |  | **Page** |
| 1 | Conceptual Diagram of the Proposed System…………………. |  |
| 2 | Block Diagram……………………………………………………... |  |
| 3 | IoT Design Methodology – Steps………………………………... |  |
| 4 | Process Specification……………………………………….......... |  |
| 5 | Use Case Diagram of IoT – Monitoring System………………... |  |
| 6 | Class Diagram……………………………………………….......... |  |
| 7 | Domain Model Specification……………………………………… |  |
| 8 | Information Model Specification…………………………………. |  |
| 9 | Service Specification……………………………………………… |  |
| 10 | Service Specification ……………………………………………... |  |
| 11 | IoT Level Specification………………….………………………… |  |
| 12 | Functional View Specification……………………………………. |  |
| 13 | Operational View Specification………………………………...... |  |
| 14 | Device and Components Integration……………………………. |  |
| 15 | Dashboard…………………………………………………………. |  |
| 16 | Temperature Analysis…………………………………………….. |  |
| 17 | Humidity Analysis…………………………………………………. |  |
| 18 | Home Screen……………………………………………………… |  |
| 19 | Reports Screen……………………………………………………. |  |
| 20 | Data Schema ……………………………………………………… |  |
| 21 | Program Hierarchy of Microcontroller…………………………… |  |

|  |  |
| --- | --- |
| 22 | Program Hierarchy of Mobile Application……………………… |
| 23 | Arduino Uno Microcontroller……………………… |
| 24 | Raspberry Pi……………………… |
| 25 | Temperature and Humidity Sensors……………………… |
| 26 | Hardware Architecture Design of Microcontroller……………… |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF PREVIEWS** |  |
| **Preview** |  | **Page** |
| 1 | Automated Microcontroller………………………. |  |
| 2 | Mobile Application Home screen………………………………… |  |
| 3 | Mobile Application Sidebar Menu……………………………… |  |
| 4 | Mobile Application Monitor……………………………………... |  |
| 5 | Mobile Application Temperature Analysis….. |  |
| 6 | Mobile Application Humidity Analysis………………………... |  |
| 7 | Mobile Application Reports……………………………… |  |

# ABSTRACT

Lettuce, a seemingly simple vegetable, offers a complex perspective on the global food system. Its diverse varieties showcase the range of crops feeding the world, while its nutritional value highlights the link between food and a healthy population. However, current monitoring methods in agriculture, like visual inspection used by BSVF, lack precision and hinder informed decision-making. This points to a critical challenge, the need for timely and accurate data to optimize crop production. Researchers have developed the LetsuApp system to address this challenge. LetsuApp is an integrated temperature and humidity sensor system built on the principles of a Smart Internet of Things (IoT) Monitoring System. This system simplifies the monitoring of crucial factors including temperature, and humidity of lettuce. Its ultimate goal is to improve crop yields and promote the use of data-driven monitoring technology in agriculture. LetsuApp collects environmental data through sensors, meticulously processes it using a microcontroller and Raspberry Pi, and performs analysis with Python scripts. The system stores data, performs machine learning predictions, and transmits real-time updates to a cloud platform. This combination of hardware, software, and cloud platforms allows for efficient data collection, analysis, and ultimately, improved decision-making. User feedback on LetsuApp's effectiveness and usability was overwhelmingly positive, with an average score of 4.35 (on a 5-point scale). This strong user endorsement highlights the potential benefits of integrating into BSVF's operations. Based on these compelling results, integrating the developed system into BSVF's operations is strongly recommended.

**Keywords:** Internet of Things, Machine Learning, Lettuce Monitoring Application, Smart Farming

**Chapter I**

## THE PROBLEM AND ITS SCOPE

### **Rationale**

Emergencies are inherently unpredictable, occurring anywhere and anytime, necessitating rapid decision-making. This unpredictability can disrupt the usual chain of command, potentially leading to lapses in judgment and significant losses. The general public is not exempt from a various security threats, both internal and external. These threats can cause physical harm and even death. Among the causes of death are heart disease, traffic accidents, and death caused by criminal activities.

Building upon the insights gained from the global response to the COVID-19 pandemic, as highlighted by Erkhembayar et al. (2020), our research aims to further enhance emergency response mechanisms. The proactive measures taken by countries, such as Mongolia, underscore the importance of efficient emergency response frameworks, facilitated by legal structures like the State Emergency Committee and the Disaster Protection Law. Despite these frameworks, challenges persist in promptly reporting emergencies, especially in remote areas where victims may encounter difficulties in using traditional communication methods like phone calls or SMS.

Drawing from established research, such as the work of Khalemsky & Schwartz (2017) & McNab et al. (2009), which highlight the transformative impact of mobile emergency response applications, our study seeks to expand upon these insights. Khalemsky & Schwartz, (2017) demonstrated the substantial reduction in response times achieved through mobile emergency applications, resulting in faster medical assistance and improved patient outcomes. McNab et al. (2009) provided valuable insights into the design principles of these applications, emphasizing the importance of achieving optimal performance. In the context of our research, these studies underscore the potential of advanced technology in enhancing emergency services and response effectiveness

To address the challenge of improving emergency response times in remote areas, the implementation of this research can provide a centralized platform for rapid emergency calls, precise GPS location information, direct communication with local emergency services, and emergency requestor identification. This comprehensive approach enhances the region's emergency response capabilities and better protects its residents.

As technology advances, it provides alternatives, and solutions begin to emerge. One suggested technology is a personal emergency notification mobile application (Hong et al., 2017). Technologies like Bilar Emergency Response Application (BERA) are crucial in enhancing response times and effectiveness. Through this application, the public can effortlessly dispatch emergency messages to family, friends, or relevant institutions. The application aids in facilitating evacuation by providing information about the sender's location and the nature of the threat encountered. This application is anticipated to make a practical contribution to the community by offering a swift communication method during crucial situations. It is also expected to support existing emergency assistance services by providing valuable assistance to authorized agencies.

### **Literature Background**

According to Republic Act No. 10121, also known as the "Philippine Disaster Risk Reduction and Management Act of 2010," this legislation provides a robust legal and policy framework for disaster risk reduction and management in the Philippines. By adhering to the provisions of this act, the application aims to contribute to enhancing the country's disaster preparedness, response, and recovery efforts. It underscores the commitment to promoting resilience and reducing the adverse impact of disasters on communities and individuals, as envisioned in RA 10121.

Republic Act No. 10121, enacted on May 27, 2010, serves as the legal foundation for the Philippine Disaster Risk Reduction and Management System. The law outlines the country's commitment to addressing vulnerabilities, enhancing institutional capacities, and building community resilience to disasters and climate change impacts. It emphasizes a holistic and proactive approach, incorporating international principles, and aims to integrate disaster risk reduction into various aspects of development, governance, and peace processes. The Act underscores the importance of gender-responsive, community-centered, and environmentally sustainable disaster risk reduction and management practices.

The Senior Technology Exploration, Learning, and Acceptance (STELA) model, as proposed by Tsai et al. (2019) in their longitudinal randomized controlled trial published in Educational Gerontology (45(12), 728-743), aims to investigate the intricate dynamics of technology exploration and learning. As individuals age, the process of adopting new technologies becomes increasingly complex.

As we enter the era of globalization, technology usage optimization should be put into practice and continue to grow in the area of personal and public security. This is important because as we advance in the globalization timeline, the risks to our security in every area of our lives are at risk.

The Women's Design Service in London, UK, developed the "Making Safer Places" procedure in Bristol, Wolverhampton, London, and Manchester. The procedure uses a tool called the "fear-o-meter" to investigate the things that make women afraid (Whitzman, Legacy et al. 2013). The women conducted a survey of their domestic neighborhood, estate, and playground using highly participatory equipment in order to determine the aspects of the physical environment that need to be improved. For instance, rerouting roads, establishing fences, and identifying and moving vegetation.

Both personal and public safety are issues that could be addressed by further improving the current environment in our society. We cannot dispute that neglected infrastructure exists in both our urban and rural areas, and that these areas need to be improved in order to stop serving as magnets for criminal activity.

In Steurer's (2018) study on worldwide crime, homicide rates and burglary/housebreaking rates were analyzed across 198 countries, utilizing data from Knoema. The examination unveiled a diverse range of crime rates, prompting the application of logarithmic transformation to attain a more normalized distribution.

As they say, "Prevention is better than cure," so law enforcement should seriously think about using technology to deter crime. Given that crime prevention could be improved in an efficient and effective manner, this could help law enforcement officials in their duties.

Four stages of disaster management are typically recognized in the field: mitigation, preparedness, response, and recovery (Zlatanova et al., 1998). Preparation is mostly focused on routine planning within the emergency services and law enforcement (e.g., for emergency situations such as police, ambulance, fire). Every phase is interrelated and vital, but from the perspective of saving lives, the response and recovery phases are frequently regarded as the most important.

As a result, it is imperative that law enforcement respond to calls from crime victims as soon as they are received. In order to lessen the aftereffects of a crime towards the victim or victims who made the call, moments like these should be handled carefully and quickly.

The idea of an event timeline is one that is frequently applied in emergency response. This outlines the incidents, emergency calls, reactions, and other actions in chronological order. that take place throughout an event Sene (2008). Timelines are useful for post-event response assessment and can be made instantly available to help other responders comprehend the circumstances.

Clustering algorithms serve as essential tools in data analysis, enabling the automatic grouping of similar data points and unveiling patterns within large datasets. By categorizing data into distinct clusters, these algorithms simplify the interpretation of complex information, aiding researchers and practitioners in extracting meaningful insights. Their versatility extends to various domains, contributing to market segmentation, image analysis, biological data interpretation, and more, making clustering algorithms indispensable for effective data exploration and decision-making.

According to Xu and Wunsch's influential research on clustering algorithms, cluster analysis emerges as a crucial tool for comprehending unlabeled data, with their exploration of hierarchical structures and group formation emphasizing specific goals while acknowledging ongoing efforts to address associated challenges (Xu & Wunsch, 2005). In another study by Xu, D., & Tian, Y. (April 2015), an overview of commonly used clustering algorithms is presented, introducing their basic ideas, specifying sources, and analyzing the advantages and disadvantages of 19 selected categories, aiming to offer readers a systematic and clear understanding of this important data analysis method. A study by Na, S., Xumin, L., & Yong, G. (April 2010) explores the significance of clustering analysis in data mining, emphasizing the direct impact of clustering algorithms on results. It specifically discusses the drawbacks of the standard k-means algorithm, proposing an enhanced version that optimizes efficiency by utilizing a simplified data structure to store information across iterations, ultimately improving both the speed and accuracy of clustering.

Regarding our law enforcement's emergency responses, there is one thing we can all agree upon. In an emergency, emergency response should always occur as quickly as possible to protect victims of crime from additional harm from the already-occurring crime.

The following articles provide valuable insights and resources for creating and enhancing emergency response systems and applications, offering essential guidance for developers seeking to improve disaster preparedness and response capabilities.

Designing Mobile Applications for Emergency Response: Citizens Acting as Human Sensors, the authors conducted an investigation of emergency notification (EN) mobile applications, aiming to analyze their characteristics and practical usefulness. They used the Design Science Research (DSR) approach and identified that while generic social applications are commonly used for large-scale crises, specific EN applications are more effective for small-scale events. They also found that users prefer multimedia features over text and forms in such applications, suggesting the potential for improved usability and adoption in emergency situations.

The review of emergency response in disasters: present and future perspectives study conducts a systematic analysis of 3,678 publications (1970–2019) from the Web of Science to investigate the emerging field of emergency response research in disasters. The analysis employs bibliometric and social network analysis methods, revealing key research topics such as emergency response, simulation, optimization, emergency medicine, and education. The paper also identifies four primary research themes and highlights two research hotspots ("optimization" and "demand"), providing valuable insights and directions for future research in the field of emergency response.

Disaster management and emerging technologies: a performance-based perspective paper employs a systematic literature review (SLR) and VOSviewer software to analyze the impact of emerging technologies (ETs) on disaster management (DM) processes. It highlights the complexity and varying terminology in the DM field, emphasizing the importance of clarifying phases and roles. The study identifies key ETs, such as simulation, robotics, IoT, and social media, and their associations with different DM phases, emphasizing the potential of simulation for preparedness, robotics and IoT for response, and social media for performance measurement, management, and accountability. Additionally, the paper suggests future research directions and practical implications for enhancing DM performance using ETs.

User-Centered Design enhances the research by providing a fundamental framework for developing an effective emergency response application, such as "BERA." Examination of technology-driven strategies in "BERA" seeks to reveal the profound impact of user-centered design principles on its overall effectiveness. This approach ensures the app is meticulously tailored to meet user needs, especially in high-stress emergency scenarios.

The integration of a clustering protocol algorithm within the "BERA" emergency response application plays a vital role in enhancing situational awareness and response prioritization. By systematically identifying incident clusters and directing resources in a data-driven manner, the application not only reduces response times but also contributes to overall user safety.

There are numerous related applications that are running in different organizations and institutions. Among these significant studies where:

Citizen – This App provides COVID contact tracing, real-time safety alerts, and 24/7 assistance at your fingertips. In the midst of the pandemic, it helps track potential COVID exposures and offers key features like free at-home testing, crime alerts, police activity updates, breaking news videos, and safety alerts for loved ones. Citizen app is a powerful tool for personal safety and COVID-19 awareness, puts vital information and resources in your hands, ensuring you stay informed and prepared in challenging times.

Emergency + – This national app utilizes the GPS built into smartphones to display users' location coordinates. In the event of an emergency, when users dial Triple Zero (000), this app enables them to convey their precise location to the emergency call-taker. Additionally, the app provides information on other national numbers such as Crime Stoppers, Health Direct, and the National Relay Service. Furthermore, it includes built-in accessibility features that audibly describe on-screen content, allowing callers to use the app even without visual interaction. Vision Australia conducted a review of the app to ensure compliance with accessibility requirements.

EchoSOS – EchoSOS was created to improve first contact and communication between people requesting help and the emergency services. EchoSOS is constantly being developed to face the challenges in rescue and health care and to improve communication and information exchange. EchoSOS was developed with the primary goal of enhancing the initial contact and communication between individuals seeking assistance and emergency services, and it continues to evolve to meet the ever-evolving challenges in rescue and healthcare, prioritizing the enhancement of communication and information exchange for more effective emergency responses.

SirenGPS – Dialing 911 from a mobile phone doesn't bring instant aid, because dispatchers need some location info to find you. SirenGPS (Android, iOS) puts them at the touch of one big red button. If your community subscribes to Siren 911, nearby first responders will receive your location and profile (emergency contacts, medical history, allergies and current medications, which you put into the app), improving your chance of being rescued in time.

## THE PROBLEM

### **Statement of the Problem**

This study aimed to design and develop an application called BERA for Bilar Search and Rescue Unit (BISARU) in BISU to improve communication, coordination, and response times during emergencies.

Specifically, it seeks to answer the following questions:

1. What are the current processes for receiving emergency calls and tracking the location of emergency during response operations?

2. What features were essential in the development of the BERA?

3. How to design the application with the modules?

1. Emergency Notification
2. User Authentication and Role Management
3. Location Tracking
4. Clustering Protocol
5. Reports
6. What is the level of application acceptability regarding the usability of the emergency response application, as perceived by the target users?

### **Objectives of the Study**

The main objective of the study is to design an application called Bilar Emergency Response Application (BERA).

Specifically,

1. To develop an Emergency Response Application for BISARU in Bilar, Bohol.
2. To test and evaluate the system using System Usability Survey.
3. To implement and deploy the application called BERA in Bilar Bohol.

**Scope and Delimitations**

The proposed application focused on the following modules:

* **Emergency Notification –** This module's scope involves defining how emergency alerts are received, processed, and communicated within the application. It may include features for reporting different types of emergencies and the level of detail provided in alerts.
* **User Authentication and Role Management –** ensure secure access to the emergency response application, allowing users to register, log in, and recover passwords while maintaining security measures. Additionally, role management enables administrators to assign and modify user roles, ensuring that authorized individuals have appropriate access levels to perform their designated tasks within the application.
* **Location Tracking –** The scope of this module encompasses how the application tracks and displays the real-time location of emergency responders and individuals in distress. It may specify the technology used for location tracking and the level of accuracy required.
* **Clustering Protocol -** Integrating a clustering algorithm within the Emergency Response Coordination module optimizes resource allocation, enhances situational awareness, and streamlines response prioritization. By systematically identifying incident clusters and directing resources in a data-driven manner, the application not only reduces response times but also contributes to overall user safety during emergency scenarios.
* **Reports** **–** This module's scope includes the generation of reports and analytics related to response times, resource utilization, and overall system performance. It defines the types of reports and the key performance indicators to be tracked.

The study may predominantly pertain to Bilar Bohol, Philippines, and might not be directly applicable to regions with markedly distinct geographic attributes, infrastructure, or emergency response requirements. The system will exclusively be available on platforms like Android mobile devices with provisions for offline functionality in areas with restricted connectivity. The application will solely perform functions such as real-time incident reporting, location tracking, resource allocation, communication, mapping, user management, reporting, notification, and data storage, all aimed at augmenting emergency response capabilities within the confines of the municipality of Bilar.

### **Significance of the Study**

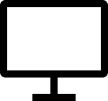
This project is to bring positive implications to the society in the issue of requesting for help during critical or emergency situations. In particular, the following entities may benefit from this study:

* Affected Individuals and Communities
* Researchers
* Future Researcher

## RESEARCH METHODOLOGY

### **Development Framework**

Figure 1 below represents the conceptual diagram of the study that represents the principle of input-process-output. Inputs are the incident reports of emergency requestor collected by the administrator. The process involves incident reporting, location tracking, user authentication, clustering and resource allocation. The output includes reports detailing incidents and response times.



Output

(Reports)

Location Tracking

Incident Reporting

User Authentication

Clustering Algorithm

Firebase Server

Emergency Requestor

Emergency Responder

Administrator

Response

Figure 1. Conceptual Diagram of the BERA Application

Figure 1: Conceptual Diagram of the Proposed System.

**Block Diagram**

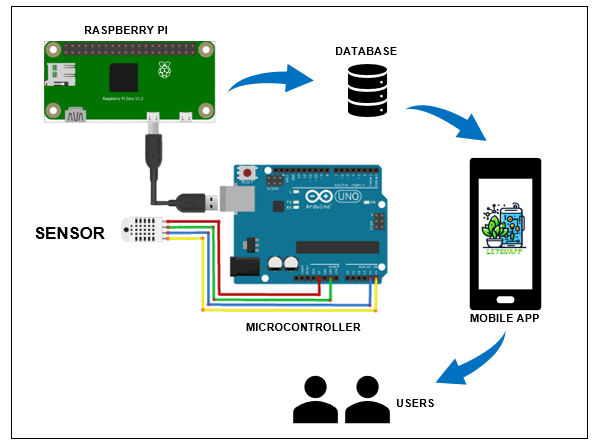
In Figure 2 various sensors are depicted, each tasked with measuring distinct conditions and parameters. Once the sensors detect data, the microcontroller takes charge of processing this information. After processing, the microcontroller establishes an automatic connection with the Raspberry Pi. Through this connection, the processed data is transmitted to the Raspberry Pi, which acts as an intermediary computing platform. Upon reaching the Raspberry Pi, the data is stored on a database server. This server is designed to handle data storage efficiently, ensuring that the information is securely stored and readily accessible for subsequent use. The database server is seamlessly integrated with a mobile application, enabling automatic display of the stored data in a detailed format. Users can then access this information on their mobile devices, gaining insights and real-time updates on the conditions and parameters monitored by the sensors.

Figure 2: Block Diagram

### **Development Models**

IoT-Based Design Methodology has a wide range of choices available for each component. IoT systems designed with IoT-Based design methodology will reduce design time, testing time, maintenance time, complexity, and better interoperability (Oppos et.al, 2022). The steps involved in designing an IoT system can be summarized as shown in Figure 3 below.



Figure 3. IoT Design Methodology – Steps

### **Purpose & Requirements**

This study aims to develop a System for Monitoring Lettuce that enables the automatic monitoring conditions and their display on mobile applications. The system operates in auto mode, continuously measuring various parameters such as soil moisture content and soil temperature. This auto-mode behavior ensures that the system remains functional and provides real-time updates.

In terms of application deployment, the system is designed to be deployed locally on the device itself. This ensures that the monitoring capabilities are readily available and accessible without relying on external servers or internet connectivity.

It's important to note that the system lacks user authentication capability. This means that there is no mechanism in place to authenticate or verify the identity of the users accessing the system. Therefore, the system operates without requiring users to provide credentials or log in to access the monitoring data. While this lack of authentication may provide convenience in terms of accessibility.

**Process Specification**

Figure 4 shows the process specification in the hardware components of the Internet of Things system for monitoring lettuce, in which the microcontroller is in auto mode in detecting and measuring conditions that include humidity and temperature using a temperature sensor and humidity sensor.

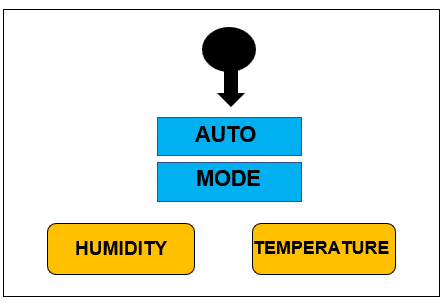


Figure 4. Process Specification

### **Software Development Tools**

For the development of the system, the resources, that the researchers will use are the following:

**Arduino IDE (v.1.8.19).** Uses, it primarily as a simplified version of C++ for writing code. Arduino programming, however, often involves the use of standard C++ features along with specific libraries provided by the Arduino platform to interact with the hardware easily.

**Python (v.3.12.0).** The researcher uses this Ideal for rapid development of the server and anomaly detection.

**PyCharm (v.2023.3.3).** The researcher uses this IDE because it is versatile and extensible, supporting multiple languages and equipped with various extensions for IoT development.

**Kodular App Inventor.** The researcher uses this application to make mobile applications that allow to creation of mobile apps for Android devices without needing programming experience. It can be used to create apps that can interact with a microcontroller board via Bluetooth, allowing you to control and monitor the microcontroller remotely.

**C++.** The researcher uses this language that executes and allows the building of a scalable embedded system for microcontroller device management.

**Django.** Serves as the backend framework in creating the web API to seamlessly integrate algorithms into the mobile application. It facilitates the communication between the mobile app and the supervised and unsupervised machine learning model, enabling real-time.

**MySQL.** The researcher uses this open-source relational database management system to store, manage, and retrieve structured data, offering reliability, flexibility, and scalability for various applications.

**GitHub**. The researcher uses this control to track changes in your codebase and collaborate with a team. Platforms like GitHub or GitLab can host your repository.

**API**. The researcher uses communication protocols for web-based communication between the IoT system and other applications or servers.

**Firebase Server**. The researcher uses a cloud services platform for app development, providing real-time database and hosting services.

**Node.js.** The server-side application is built with Node.js. It enables real-time communication between Arduino-based temperature and humidity sensors and the server. Node.js was chosen for its efficiency in handling asynchronous operations, making it suitable for the IoT project's real-time requirements. Data collection, transmission, and analysis are made possible by the technology, which is critical for monitoring and managing lettuce in agricultural settings. Its event-driven architecture is well suited to the scalability and responsiveness requirements of IoT applications.

**Heroku**. The researcher integrates Heroku into the system architecture to create an API, streamlining communication between different components and enhancing the interoperability of the monitoring system.

**Environment and Participants**

The study was conducted at Riverside, Bilar, Bohol. The respondents were the External Participants, Students of BISU Bilar Campus, and IT Teachers who were the direct persons involved in gathering data and viewing the data of the conditions.

### **Data Collection**

To create an Internet of Things – System for Monitoring Lettuce, the researchers conducted a research group to get more information about the proposed system. In testing and evaluating the researchers used ISO 25010, a survey of ISO 25010 was conducted to the Seventeen (17) respondents. Device Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability factors involved Seventeen (17) respondents, which comprised three (3) external participants ten (10) selected students from Bohol Island State University- Bilar Campus, and four (4) it teachers. The distribution of these respondents is presented in Table 1.

Table 1

Summary of Respondents

|  |  |
| --- | --- |
| RESPONDENTS | NO. OF RESPONDENTS |
| External Participants | 3 |
| Students | 10 |
| IT Teachers | 4 |
| **TOTAL** | **17** |

The researchers used convenience sampling to get the target respondents which are the External Participants, Students, and IT Teachers.

This is a specific type of non-probability sampling method that relies on data collection from population members who are conveniently available to participate in the study. Convenience sampling is a type of sampling where the first available primary data source is used for the research without additional requirements.

After conducting the interview, the clients were allowed to do the actual hands-on activities with the developed system. Then, after the demonstration and hands-on activity, a questionnaire was provided to assess the level of the Device's Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability Factor. The guide for the interpretation of the results of the Device Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability Factor in the Internet of Things – System for Monitoring Lettuce using ISO-25010 is presented in Table 2.

Table 2.

Interpretation Guide of Device Factors using ISO-25010

|  |  |  |  |
| --- | --- | --- | --- |
| **Weight** | **Range** | **Description** | **Interpretation** |
| 5 | 4.20 – 5.00 | Strongly Agree | The Respondent Strongly believe and confident that the Internet of Things System for Monitoring Lettuce is very Usable. |
| 4 | 3.40 – 4.19 | Agree | The Respondent believe and confident that the Internet of Things – System for Monitoring Lettuce is Usable. |
| 3 | 2.60 – 3.39 | Neither Agree or Disagree | The Respondent are neutral in trusting the Internet of Things - System for Monitoring Lettuce is usable. |
| 2 | 1.80 – 2.59 | Disagree | The Respondent believe that the Internet of Things - System for Monitoring Lettuce is not usable. |
|  |  |  |  |
| 1 | 1.0 – 1.79 | Strongly Disagree | The Respondent are strongly confident that the Internet of Things - System for Monitoring Lettuce is not usable. |

To determine the general acceptability of the Device Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability. Factor for Internet of Things - System for Monitoring Lettuce the average weighted mean or weighted mean score computed using the following formula.

𝑊𝑀𝑆 = 1𝑓1 + 2𝑓2 + 3𝑓3 + 4𝑓4 + 5𝑓5 n

n

Where:

WMS = Weighted Mean Score

*f*1 = Frequency of respondents given a rate of 1 *f*2 = Frequency of respondents given a rate of 2 *f*3 = Frequency of respondents given a rate of 3 *f*4 = Frequency of respondents given a rate of 4 *f*5 = Frequency of respondents given a rate of 5

N=Total Number of Respondents

1, 2….5 = constant (rating to the service provided)

## OPERATIONAL DEFINITION OF TERMS

The following terms are used conceptually and operationally to better understand the study.

**Arduino IDE (v.1.8.19).** Uses, it primarily as a simplified version of C++ for writing code. Arduino programming, however, often involves the use of standard C++ features along with specific libraries provided by the Arduino platform to interact with the hardware easily.

**API.** The researcher uses communication protocols for web-based communication between the IoT system and other applications or servers.

**Microcontroller.** The researcher uses a microcontroller for testing purposes that serves as the brain of the hardware system that stocks and operates the different functions for getting the different sensors.

**C++.** The researcher uses this language that executes and allows to building of scalable and efficient server-side applications, leveraging its event-driven architecture and extensive package for diverse software solutions.

**Django.** Serves as the backend framework in creating the web API to seamlessly integrate algorithms into the mobile application. It facilitates the communication between the mobile app and the supervised and unsupervised machine learning model, enabling real-time.

**Firebase Server.** The researcher uses a cloud services platform for app development, providing real-time database and hosting services.

**GitHub.** The researcher uses this control for tracking changes in your codebase and collaborating with a team. Platforms like GitHub or GitLab can host your repository.

**Internet of Things (IoT).** The researcher uses the Internet of Things cloud platform as the database.

**Kodular App Inventor.** The researcher employs Kodular to develop a mobile application interface, providing end-users with a visually intuitive platform for monitoring and receiving real-time updates on soil temperature, humidity and status in the context of a smart IoT- monitoring system for lettuce.

**Microcontroller.** A small computer on a single integrated circuit that is designed to perform specific tasks, is often used in embedded systems.

**Monitoring System.** The researcher makes a mobile application and web-based system that serve as the output device and are capable of displaying the different sensor data.

**MySQL.** The researcher uses this open-source relational database management system to store, manage, and retrieve structured data, offering reliability, flexibility, and scalability for various applications.

**LetsuApp.** LetsuApp. Letsugas, which is an acronym for Lettuce and App for Mobile Application, is what the application stands for.

**PyCharm 2023.3.3.** The researcher uses this IDE because it is versatile and extensible, supporting multiple languages and equipped with various extensions for IoT development.

**Python 3.12.0.** The researcher uses this Ideal for rapid development, data processing, and interfacing with IoT devices.

**Raspberry Pi.** The researcher uses this for its versatility across projects, from educational coding exercises to sophisticated IoT systems, due to its expandable nature and active community support, making it an accessible and adaptable platform for a multitude of applications.

**SHT20 I2C Temperature & Humidity Sensor.** The researcher uses this device to measure the soil moisture content and the temperature level of the soil. It provides valuable data for various applications in, environmental monitoring, and research. This sensor will automatically detect or count the temperature and humidity data.

**Chapter 2**

## PRESENTATION OF FINDINGS, ANALYSIS AND INTERPRETATION OF DATA

### **Existing Operations and Processes**

In traditional lettuce farming at Bilar Stem Farm Vegetable (BSFV), temperature and humidity monitoring heavily rely on manual methods. Sir Casino L. Monil, the farm's owner, primarily uses visual inspection based on experience and observation. However, these methods have limitations. Data collection happens irregularly, leading to outdated information and hindering timely decisions. The accuracy and reliability of manually gathered data pose risks, potentially affecting lettuce crop quality and yield. Moreover, the absence of standardized industry practices worsens technological gaps in humidity and temperature monitoring. Feedback from farmers emphasizes these inefficiencies, signaling the need for an advanced system. Hence, the researchers aim to develop a smart IoT monitoring system for lettuce, providing reliable and precise information to aid BSFV, other farmers, and agricultural students in making informed decisions.

An IoT system tailored for monitoring lettuce presents an effective solution for overseeing environmental conditions at Bilar Stem Farm Vegetable (BSFV). This system utilizes diverse sensors like temperature and humidity sensors to gather data, wirelessly transmitting it to a central database or cloud-based platform. Its primary advantage lies in offering real-time and precise environmental information crucial for BSFV's owner, farmers, and agricultural students. Furthermore, the system's design prioritizes accessibility and user-friendliness, ensuring easy utilization through a mobile interface for owners, colleagues, and students to access and navigate the monitoring system seamlessly.

### **Needs of the Existing Operation**

The existing operation at Bilar Stem Farm Vegetable (BSFV) for lettuce farming faces several critical needs. Foremost among these needs is the requirement for a more reliable and accurate method of monitoring environmental conditions, particularly temperature and humidity. The current reliance on manual methods results in intermittent data collection, leading to outdated and insufficient information for timely decision-making. There is a pressing need for a system that can provide real-time and precise data regarding these vital environmental parameters. Additionally, it emphasizes the need for an automated and efficient solution. Addressing these needs is crucial to enhancing decision-making, optimizing crop yield, and streamlining operations at BSFV. Based on the researcher’s observation, they found out the following needs;

1. To develop a prototype that can monitor real-time information to the BSFV about the following monitoring environmental conditions data;
   * Temperature
   * Humidity

### **Use Case Diagram**

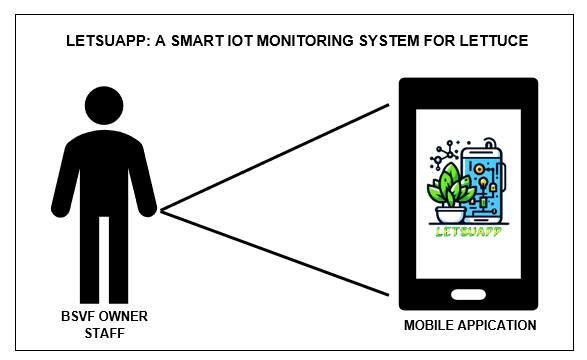
Figure 5 displays a use case diagram, as described by Krishnan (2020), illustrating how one or more individuals or entities interact with the system. A use case represents a series of steps involving actors and events, demonstrating how they engage with a system to complete a specific task. In this context, various use cases outline how the device monitors environmental conditions using different sensors. These sensors automatically transmit data to an Internet of Things platform, and the information is then showcased on a mobile application.

Figure 5. Use Case Diagram of Internet of Things Monitoring System

### **Use Case Narrative**

Use case narrative describes a use case that needs to be contextualized and depict the dialog between the user (either actors or use case) to accomplish observable goals. It needs to offer more than just a list of user-system interactions. In every use case narrative, there are pre-conditions, processes, and post-conditions. Use case 1 (Table 1) shows the Mobile Application.

Table 1. Mobile Application

|  |  |
| --- | --- |
| Use case name | Mobile Application |
| Actor | BSVF Owner and Staff |
| Precondition | BSVF Owner and Staff are already on the home screen. |
| Description | Allows the BSVF Owner and Staff to click the sidebar button to view the monitoring data analysis. |
| Typical Course of Action | |
| Actor Action | System Response |
| Step 1  Click “Sidebar Button | Step 2  Display the sidebar screen and if the actor clicks the home menu it will display the lestsuapp description. |
| Step 3  Click “Sidebar Button | Step 4  Display the sidebar screen and if the actor clicks the monitoring menu it will display the temperature, humidity, and prediction status. |
| Step 5  Click “Sidebar Button | Step 6  Display the sidebar screen and if the actor clicks the reports menu it will display the recent record of the soil moisture and soil nutrients data. |

### **Database Design**

Database design is the process of producing a data model of the database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database.

System design is defining the components, modules, interfaces, and data for a system to satisfy specified requirements of LetsuApp: A Smart IoT Monitoring System for Lettuce – Riverside, Bilar, Bohol. The researchers aim to create a new system that would be used to monitor the soil moisture and soil temperature of lettuce.

To meet the client’s needs, various improvements were made to Bilar Stem Vegetable Farm. The goal of this design was to develop and design a system that monitors the soil moisture content and soil temperature of lettuce, collaborating with the Bilar Stem Vegetable Farm at Riverside, Bilar, Bohol. This system focuses on tracking soil humidity and temperature levels. Its primary aim is to streamline the monitoring process, control watering, reduce time complexity, and offer invaluable insights.

.

### **Class Diagram**

Figure 6 shows the class diagram in the Unified Modeling Language (UML). UML forms a static structure diagram that shows the system classes, their attributes, operations (or methods), and interactions among the classes to explain the structure of a system.

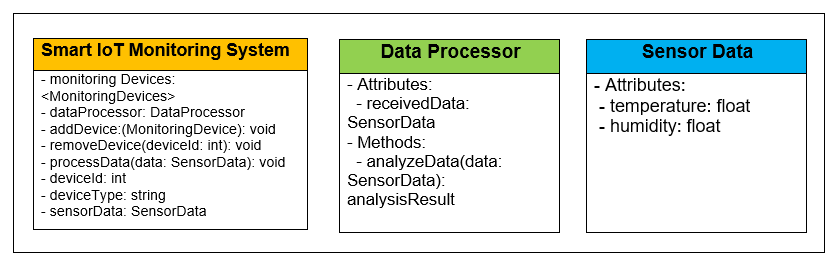


Figure 6. Class Diagram of the Study

### **Domain Model Specification**

Figure 7 shows the domain model that describes the main concepts, entities, and objects in the domain of the IoT system to be designed. The domain model defines the attributes of the objects and the relationships between objects.

The domain model specification diagram for A Smart Internet of Things Monitoring System for Lettuce described data flow begins with the collection of environmental data by sensors, which is then meticulously processed through a microcontroller before being channeled to a Raspberry Pi. The Raspberry Pi, serving as a potent processing hub, initiates Phase 1 of the operation. Here, a Python script (main.py) is executed, conducting further data manipulations and analyses. The processed data finds a home in an SQL database, ensuring organized and structured storage, and is subsequently transmitted to Heroku, a cloud platform. In this phase, the application of K-nearest neighbors (KNN) algorithms takes center stage, indicating the incorporation of machine learning for testing and predicting outcomes based on the acquired data.

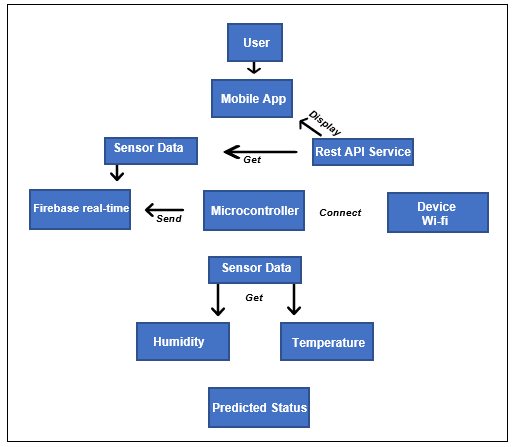
 Moving into Phase 2, another Python script establishes a link between the Raspberry Pi and Heroku for predicting the status of specific parameters. This predictive analysis is then seamlessly transmitted to Firebase, a real-time cloud platform, offering instantaneous updates. This well-orchestrated flow, combining hardware (sensors, microcontroller, Raspberry Pi), software (Python scripts), databases (SQL), and cloud platforms (Heroku, Firebase), underscores a comprehensive system aimed at efficiently gathering, processing, and analyzing data. The ultimate objective is to provide real-time insights, enable predictive monitoring, and enhance decision-making processes, particularly in domains such as agriculture or environmental monitoring.

Figure 7. Domain Model Specification

### **Information Model Specification**

Figure 8 shows the information model, which defines the structure of all the information in the IoT system. The information model specification for A Smart Internet of Thing Monitoring System for Lettuce. This step is in a field scenario that describes monitoring information is gathered from the environmental conditions, which is considered a physical entity type that is automatically detected by the sensor. The SHT20 Humidity sensor will measure the level of soil moisture content, and the DS18B20 Temperature sensor will measure level of soil temperature.

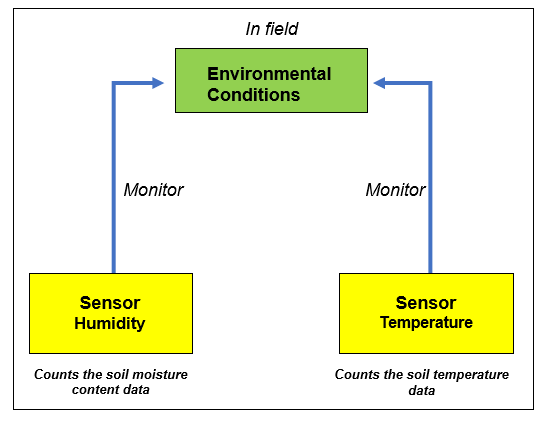


Figure 8. Information Model Specification

### **Service Specification**

The service specification, as depicted in Figure 9, serves as a comprehensive blueprint for the functionality within the IoT system. It outlines various service types, inputs, outputs, endpoints, schedules, preconditions, and effects, providing a detailed framework for the system's operation. These specifications delineate the types of services offered within the IoT ecosystem, such as data collection, processing, and dissemination. Inputs and outputs are defined, indicating the flow of information within each service, while endpoints establish the communication interfaces through which services interact. Moreover, service schedules outline the timing and frequency of service execution, adaptable to the system's operational requirements and constraints.

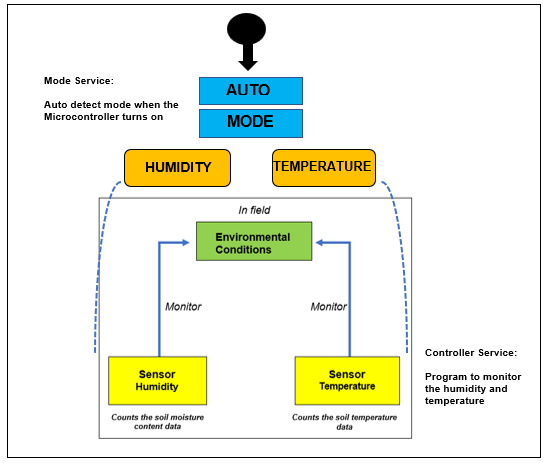


Figure 9. Service Specification

Figure 10 delineates a systematic process for gathering and disseminating sensor data. Initially, a microcontroller receives input from sensors at intervals of 5 seconds or more, contingent upon network connectivity. This microcontroller then transmits the acquired data to the Internet of Things (IoT) network. Subsequently, a REST API service interacts with the IoT platform, retrieving the sensor data. This data, considered an input for the REST API service, is then automatically showcased within a mobile application. Through this seamless integration, users gain access to real-time or archived sensor data directly on their mobile devices, facilitating efficient monitoring and analysis.

The microcontroller acts as the initial point of data acquisition, while the IoT network serves as the intermediary for transmitting and processing this data. The REST API service bridges the gap between the IoT platform and the mobile application, enabling effortless retrieval and display of sensor data.

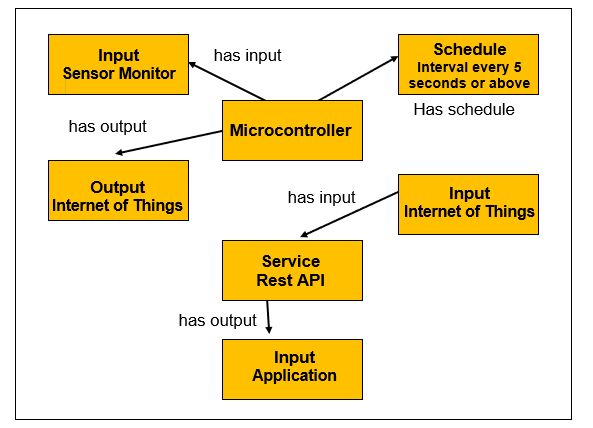


Figure 10. Service Specification

### **IoT Level Specification**

Figure 11 outlines the IoT level specification for a smart monitoring system designed specifically for lettuce. This system incorporates an Internet of Things (IoT) platform, with Firebase serving as the designated platform for data management and analysis. The IoT platform through Firebase, the platform processes and interprets the gathered data, providing valuable insights and actionable intelligence for optimizing lettuce. This specification underscores the significance of leveraging IoT technology to enhance monitoring and management processes in agricultural settings, ultimately fostering improved efficiency and productivity in lettuce production.

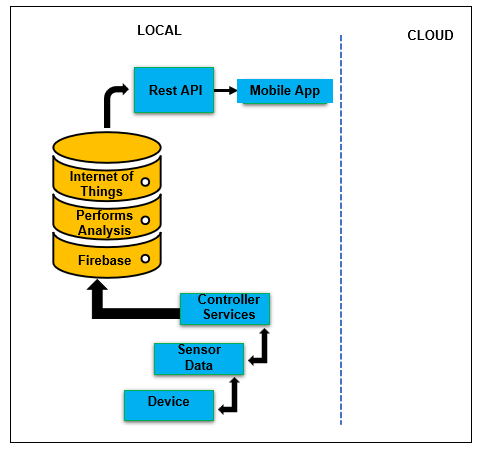


Figure 11. IoT Level Specification

### **Operational View Specification**

Figure 12 illustrates the operational view specification for a smart Internet of Things (IoT) monitoring system for lettuce, delineating the array of choices pertinent to its deployment and operation. This phase encompasses defining options ranging from service hosting to storage, device selection, and application configurations. By delineating these options, the operational view provides a comprehensive blueprint for the implementation and ongoing management of the IoT system, ensuring efficient and effective utilization of resources while meeting the requirements of weather monitoring.

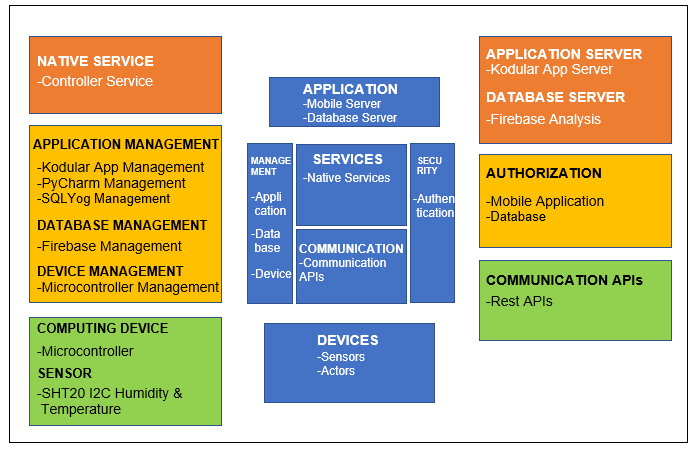


Figure 13. Operational View Specification

### **Device and Component Integration**

Figure 14 shows the device and component integration. This step defines how devices like sensors, computing devices, and other components are integrated. The interconnection of different components in the Internet of Things monitoring system. The microcontroller and automatically connected to a Wi-Fi router. To effectively integrate the described device with a microcontroller, begin by connecting its power supply and ground pins to corresponding pins on the microcontroller, ensuring compatibility with voltage requirements. Next, integrate temperature and humidity sensors by connecting their output pins to suitable input pins on the microcontroller. Once the sensors are connected, implement the necessary code to read data from them and process it accordingly. This may involve using analog or digital input pins on the microcontroller, depending on the type of sensors being utilized. Additionally, consider calibrating the sensors and adjusting the code to accurately interpret their output data for temperature and humidity readings.

With the sensors successfully integrated, establish communication between the microcontroller and the device using the I2C protocol. Connect the SDA (Serial Data Input/Output) and SCL (Serial Clock Input) pins of the device to the appropriate GPIO pins on the microcontroller configured for I2C communication. Utilize the microcontroller's I2C library or implement I2C communication protocols in your code to enable seamless data exchange between the microcontroller and the device. This allows the microcontroller to not only read data from the sensors but also interact with the device for additional functionalities such as setting configurations, retrieving status information, or controlling external components, thus enabling a comprehensive and versatile system for monitoring and control.

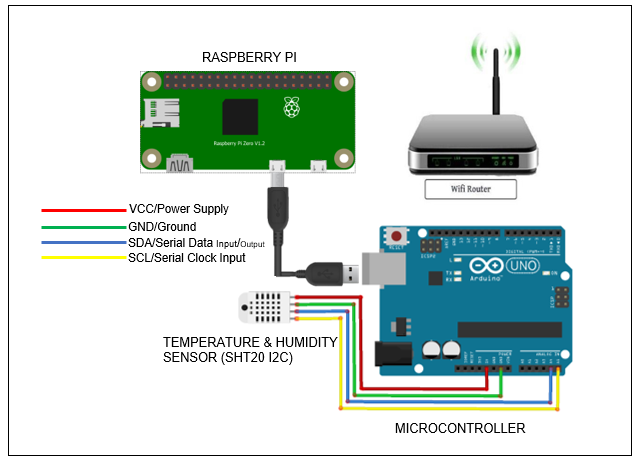


Figure 14. Device and Component Integration

### **Application Development**

Figure 15 shows the monitor of the mobile application interface in the smart IoT monitoring system for lettuce that displays data for monitoring such as temperature, humidity, and predicted status. The application also displays the sidebar menu button.

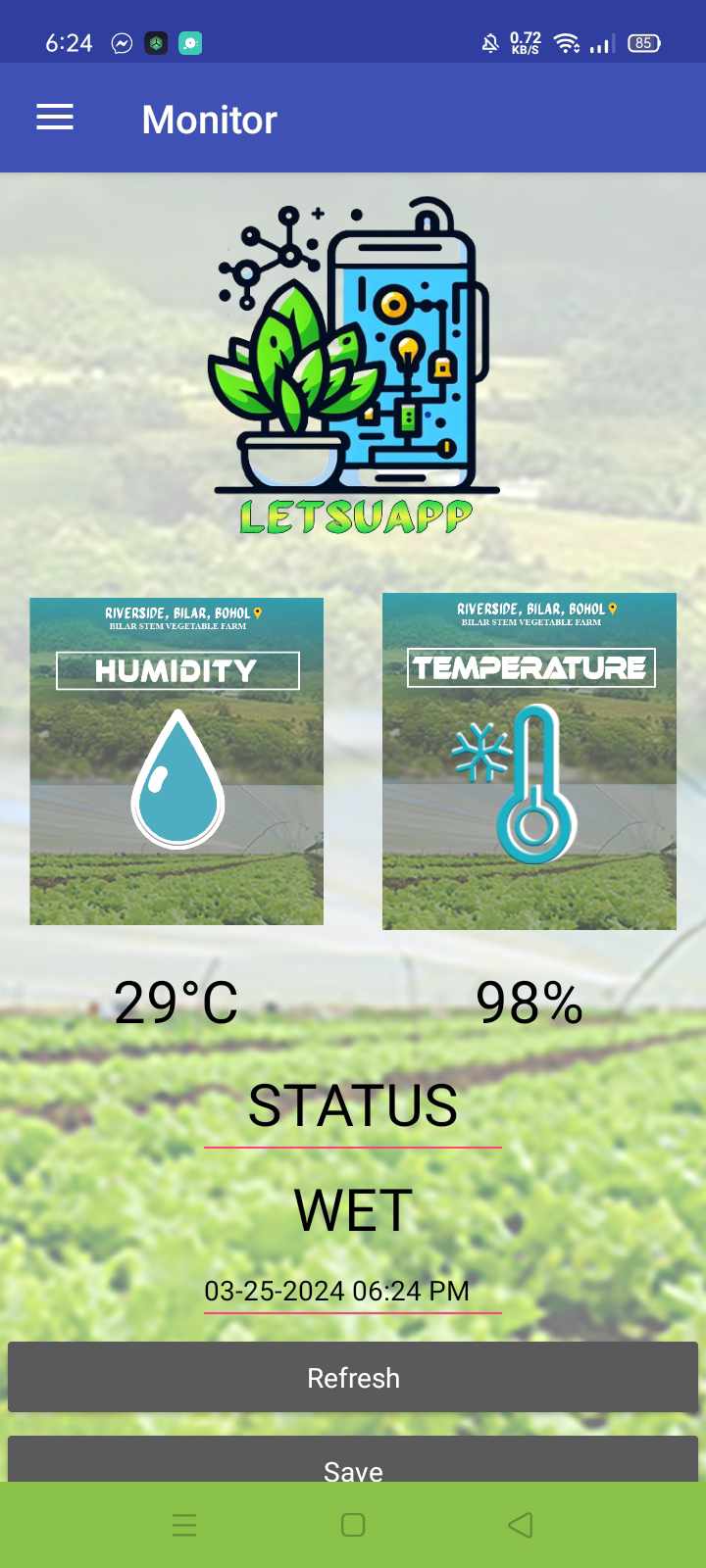
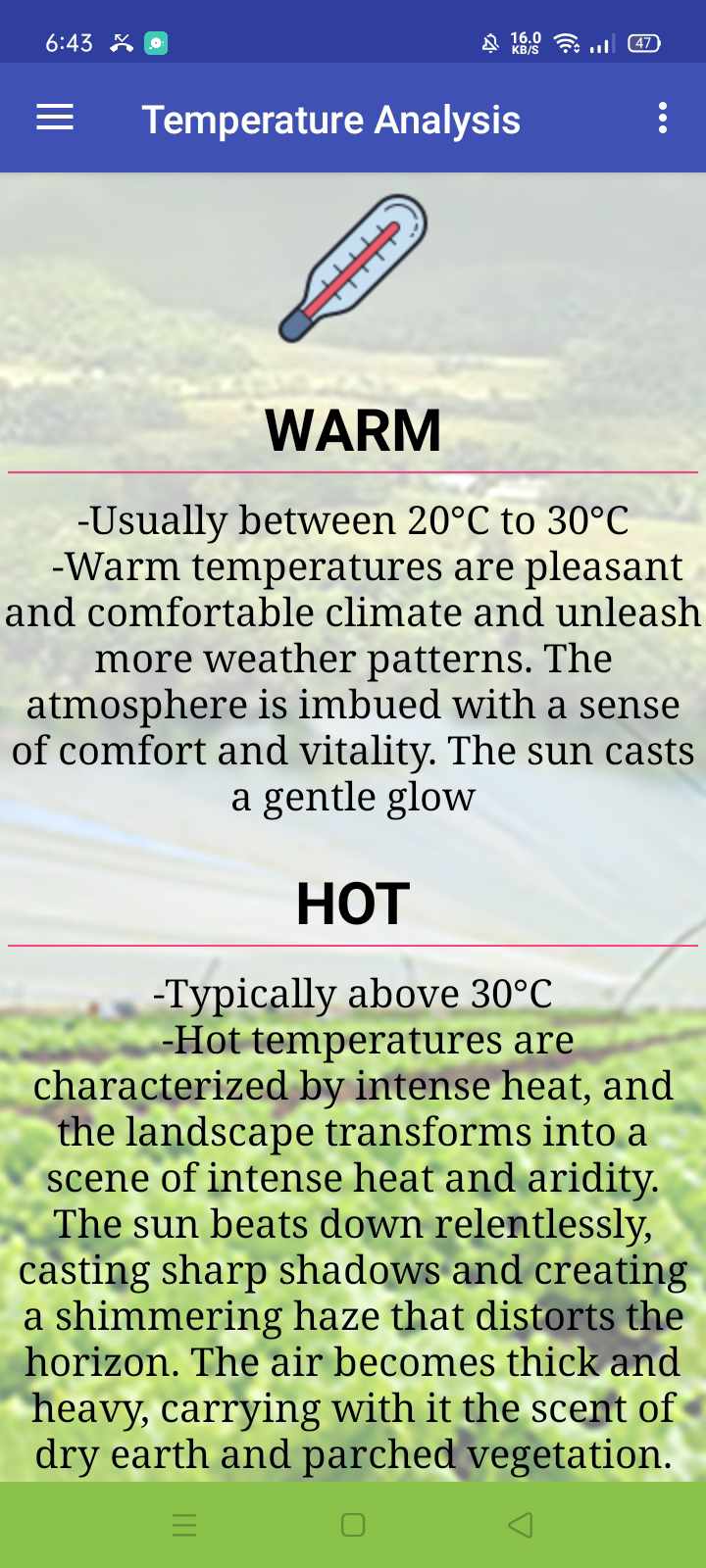


Figure 15. Dashboard

Figure 16 shows the temperature analysis of the mobile application interface in a smart IoT monitoring system for lettuce that displays the temperature data, and temperature analysis description.



.

Figure 16. Temperature Analysis

Figure 17 shows the humidity analysis of the mobile application interface in a smart IoT monitoring system for lettuce that displays the humidity data, and humidity analysis description

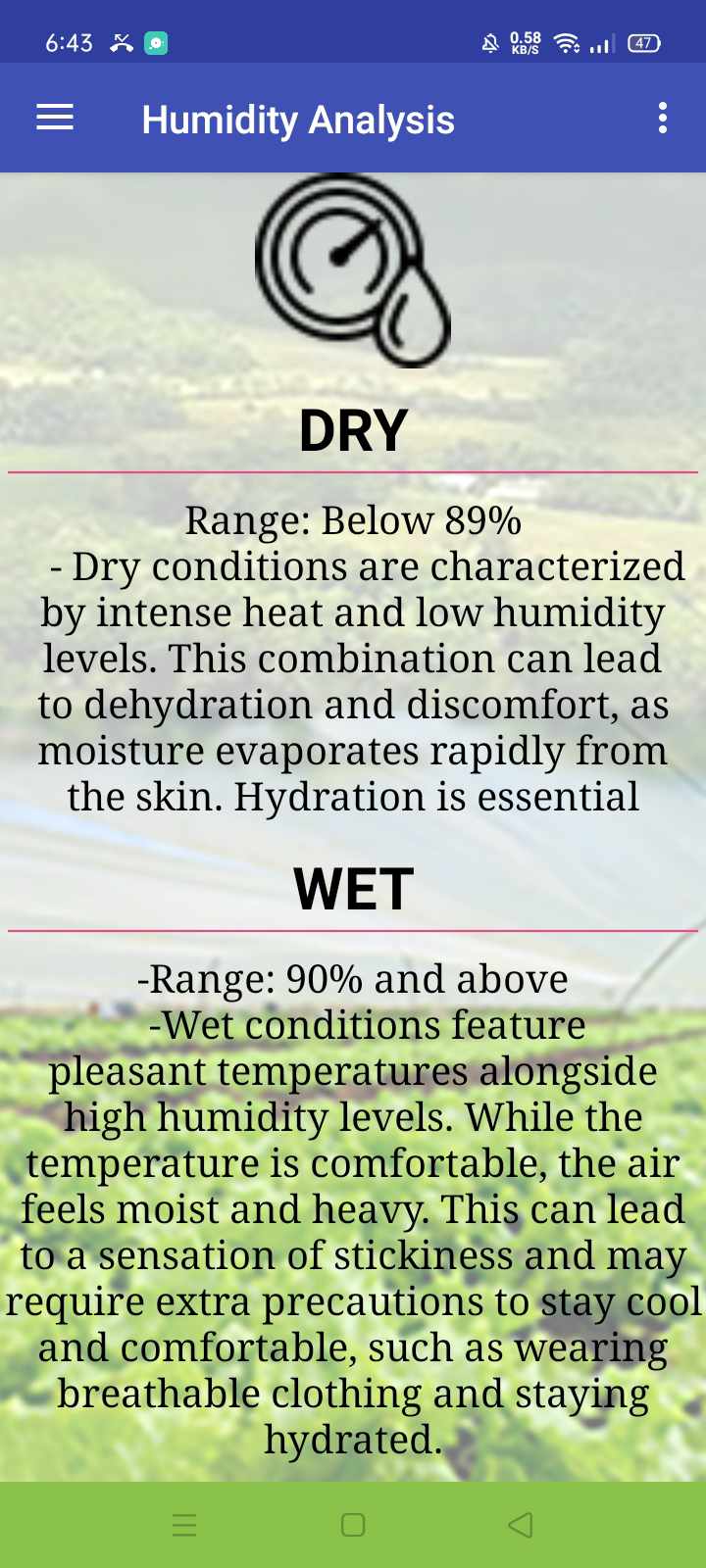
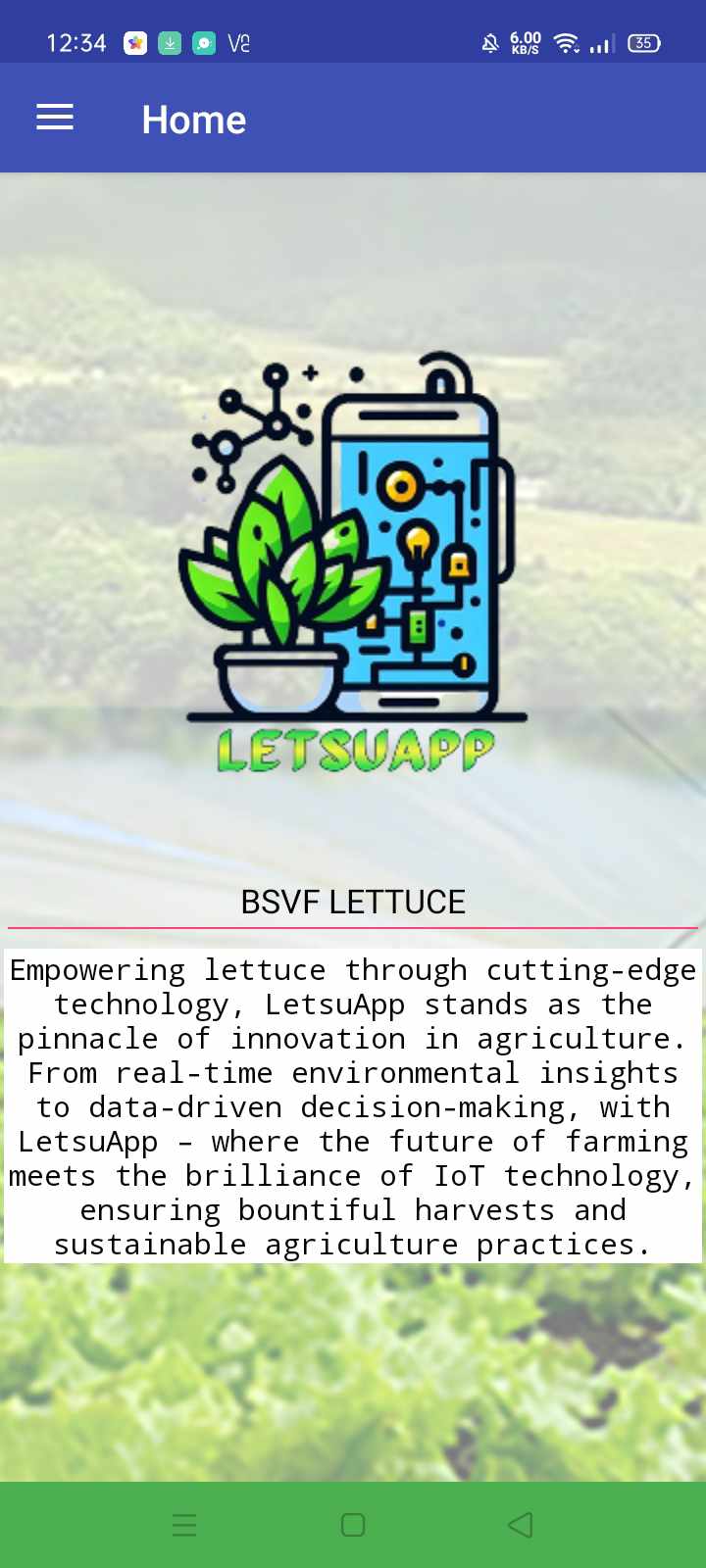


Figure 17. Temperature Analysis

Figure 18 shows the home of the mobile application interface in a smart IoT monitoring system for lettuce that displays the smart IoT monitoring system description.



.

Figure 18. Home Screen

Figure 19 shows the report's analysis of the mobile application interface in a smart IoT monitoring system for lettuce that displays the current data gathered.

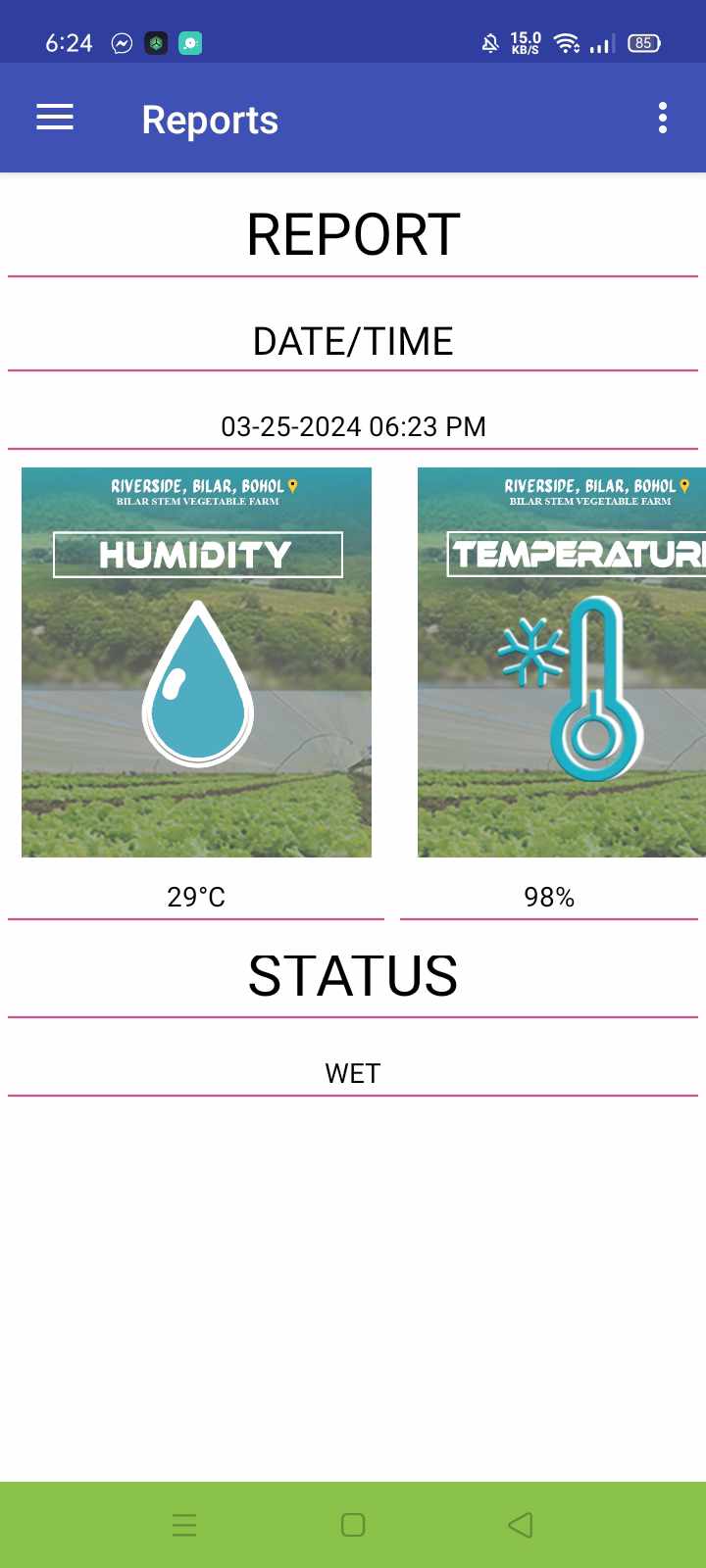


Figure 19. Reports Screen

### **Data Schema**

Figure 20 shows the Database schema is the logical representation of a database, which shows how the data is stored logically in the entire database. It contains a list of attributes and instructions that inform the database engine that how the data is organized and how the elements are related to each other. The figure below shows the tables used in the system and how they relate to each other. Allow the sharing of data within different tables.

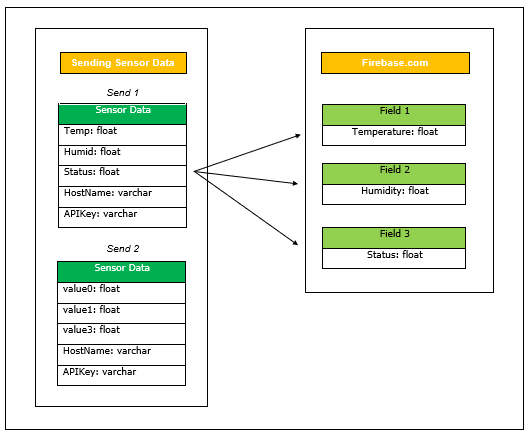


Figure 20. Data Schema of the Study

### **Database Structure**

A data structure is a way of collecting and organizing data in such a way that one can perform operations on these data in an effective way. The data structure is about rendering data elements in terms of some relationship, for better organization storage. This a structure programmed to store data so that various operations can be performed easily. The following table is the database tables used in storing the information that is inputted in the system. (Table 3) shows the Smart IoT Monitoring System Data Structure.

**Table 3**

**SQLyog (database)**

|  |  |  |
| --- | --- | --- |
| **Field No** | **Uncleaned** | **Type** |
| 1  2  3  4  5 | id  temperature  humidity  stat  dtadded | int  int  int  varchar  timestamp |

### **Program Hierarchy**

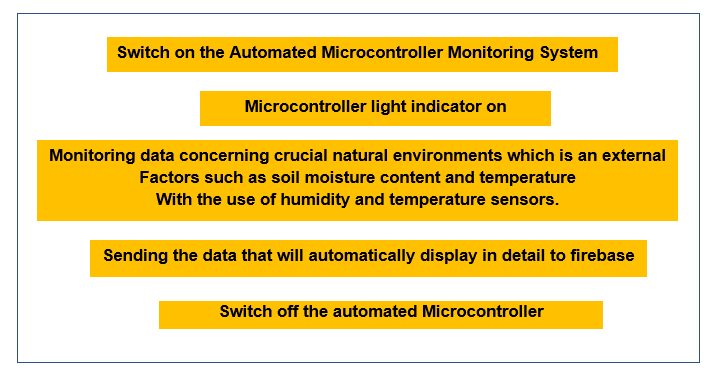
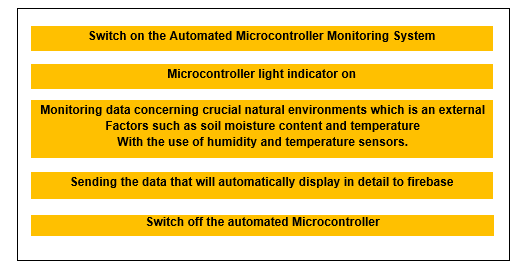
 Figure 21 shows the Program hierarchy as a chart that shows the breakdown of the proposed system to its manageable levels. As a design tool, this aids the researchers in dividing and conquering a large hardware problem that is recursively breaking down into parts that are small enough to be understood by a human brain. The figure presents the program hierarchy for the designing and development of the LetsuApp: A Smart IoT Monitoring System for Lettuce. Figure 21 shows the Program Hierarchy of the Microcontroller, and Figure 22, shows the Program Hierarchy of Mobile Application.

Figure 21. Program Hierarchy of Microcontroller

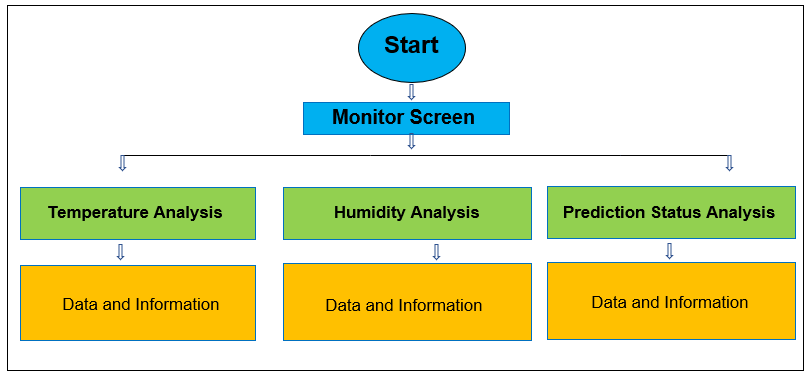


Figure 22. Program Hierarchy of Mobile Application

### **Technical Requirements**

The completion of the Internet of Things - Based has three essential requirements, namely: Hardware, Software, and Peopleware. A smart Internet of Things Monitoring System for Lettuce requirements must be present and work together in the newly assembled device.

The hardware is best described as any physical component of a computer system that contains a Microcontroller, Temperature sensor, Humidity sensor, and other electronics, which are used for processing data. Such components are classified according to their functional use within a computer system.

The software or a program can be defined as a complete set of written instructions written by the programmer, which enables the computer to obtain the solution to the problem.

The peopleware is the user who would operate the system and those who are involved in the Internet of Things - Based. Peopleware refers to the role people play in technology and the development of hardware or software.

### **Minimum Hardware Specification**

This covers the minimum hardware specification needed by the automated Microcontroller to function as intended and expected. The considerations of this specification were based on what is available in the market and what most computer package system offers or the hardware components offer.



Figure 23. Arduino Uno Microcontroller

### **Arduino Uno Microcontroller**

Figure 23 shows the Arduino Uno Microcontroller is a Micro-chip Atmega-328p microcontroller which is open-source, it consists of a physical program control board and a software which can be installed on your computer, you can program code in the computer and then upload it on the physical board. It is an IDE (Integrated Development Environment) it has resources like, source code editor, automation tools and a debugger.

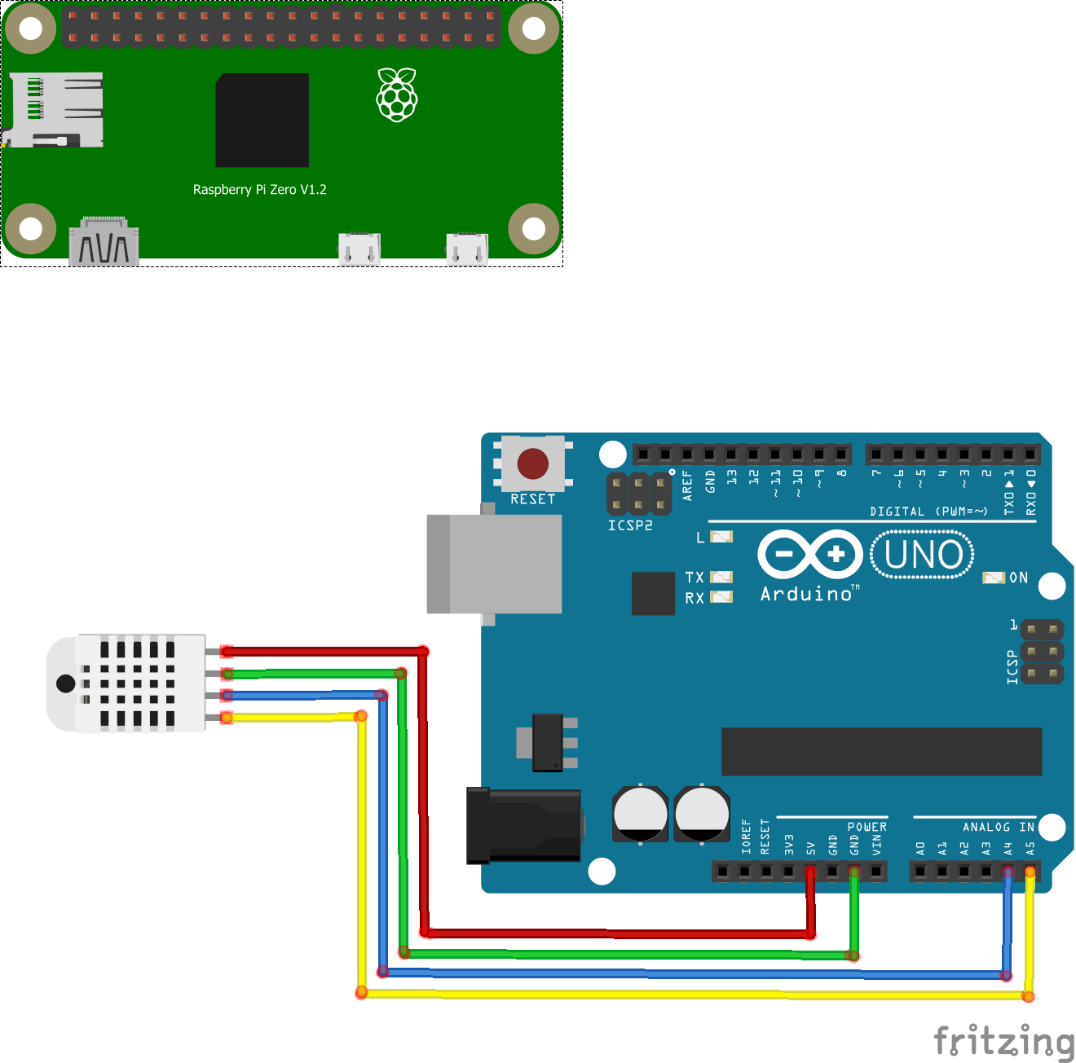


Figure 24. Raspberry Pi

### **Raspberry Pi**

Figure 24 shows the Raspberry Pi is a The Raspberry Pi can be utilized in various applications, including integrating the described device with temperature and humidity sensors. To do so, connect the power supply and ground pins of the device to the Raspberry Pi's respective pins. Then, integrate the sensors by connecting their output pins to suitable GPIO (General Purpose Input/Output) pins on the Raspberry Pi. Use Python or other programming languages to read data from the sensors and process it accordingly. Additionally, establish communication between the Raspberry Pi and the device using the I2C protocol, connecting the SDA and SCL pins to the Raspberry Pi's GPIO pins configured for I2C communication. This enables the Raspberry Pi to collect temperature and humidity data from the sensors and interact with the device for additional functionalities.



Figure 24. Temperature & Humidity Sensors

### **Temperature & Humidity Sensors (SHT20 I2C)**

Figure 24 shows the Temperature & Humidity Sensors are equipped with waterproof probes. It comes with a temperature & humidity sensor chip and the probe has gone through a dual waterproof protection test. The I2C temperature & humidity sensor adopts Sensirion techniques. It can measure the surrounding environment's temperature and relative air humidity precisely. There is also a built-in 10k Pull-up resistor and 0.1uf filter capacitor. It enables the sensor to be directly used with a microcontroller such as Arduino.

### **SOFTWARE USED**

**Arduino IDE (v.1.8.19).** Uses, it primarily as a simplified version of C++ for writing code. Arduino programming, however, often involves the use of standard C++ features along with specific libraries provided by the Arduino platform to interact with the hardware easily.

**Raspberry Pi.** The researcher uses this for its versatility across projects, from educational coding exercises to sophisticated IoT systems, due to its expandable nature and active community support, making it an accessible and adaptable platform for many applications.

**Python (v.3.12.0).** The researcher uses this Ideal for rapid development of the server and anomaly detection.

**PyCharm (v.2023.3.3).** The researcher uses this IDE because it is versatile and extensible, supporting multiple languages and equipped with various extensions for IoT development.

**Kodular App Inventor.** The researcher employs Kodular to develop a mobile application interface, providing end-users with a visually intuitive platform for monitoring and receiving real-time updates on soil temperature, humidity and status in the context of a smart IoT- monitoring system for lettuce.

**C++.** The researcher uses this language that executes and allows the building of a scalable embedded system for microcontroller device management.

**Django.** Serves as the backend framework in creating the web API to seamlessly integrate algorithms into the mobile application. It facilitates the communication between the mobile app and the supervised and unsupervised machine learning model, enabling real-time.

**MySQL.** The researcher uses this open-source relational database management system to store, manage, and retrieve structured data, offering reliability, flexibility, and scalability for various applications.

**GitHub**. The researcher uses this control to track changes in your codebase and collaborate with a team. Platforms like GitHub or GitLab can host your repository.

**Node.js.** The server-side application is built with Node.js. It enables real-time communication between Arduino-based temperature and humidity sensors and the server. Node.js was chosen for its efficiency in handling asynchronous operations, making it suitable for the IoT project's real-time requirements. Data collection, transmission, and analysis are made possible by the technology, which is critical for monitoring and managing lettuce in agricultural settings. Its event-driven architecture is well suited to the scalability and responsiveness requirements of IoT applications.

**Heroku**. The researcher integrates Heroku into the system architecture to create an API, streamlining communication between different components and enhancing the interoperability of the monitoring system.

**API**. The researcher uses communication protocols for web-based communication between the IoT system and other applications or servers.

**Firebase Server**. The researcher uses a cloud services platform for app development, providing real-time database and hosting services.

**Functional Requirement**

The developed system is composed of an Arduino Uno which can be easily programmed, erased, and reprogrammed at any instant in time. A sensor, which will detect the real-time monitoring. Table 4 shows the Hardware Interface and Architecture Design.

**Table 4**

**Hardware Interface**

|  |  |
| --- | --- |
| **Hardware Interface** | **Functions** |
| Arduino Uno Microcontroller | Use for the development of a Smart IoT Monitoring System for Lettuce.  Use for sending the accurate monitoring sensor data to the Internet of Things. |
| Temperature & Humidity Sensor SHT20 I2C | Use for detecting soil moisture content data.  Use for detecting soil temperature data. |
| Serial USB Cable | Use to connect Microcontroller to the Raspi. |
| Raspberry Pi | Use for tasks like hosting and managing. |
| Jumper Wire | Use to connect the other components to Arduino |
| USB cable | Use to connect Arduino to Arduino IDE. |
| Power Supply | Use to supply power to Arduino |

Figure 25 shows the device and component integration. This step defines how devices like sensors, computing devices, and other components are integrated. The interconnection of different components in the Internet of Things monitoring system. The microcontroller and automatically connected to a Wi-Fi router. To effectively integrate the described device with a microcontroller, begin by connecting its power supply and ground pins to corresponding pins on the microcontroller, ensuring compatibility with voltage requirements. Next, integrate temperature and humidity sensors by connecting their output pins to suitable input pins on the microcontroller. Once the sensors are connected, implement the necessary code to read data from them and process it accordingly. This may involve using analog or digital input pins on the microcontroller, depending on the type of sensors being utilized. Additionally, consider calibrating the sensors and adjusting the code to accurately interpret their output data for temperature and humidity readings.

With the sensors successfully integrated, establish communication between the microcontroller and the device using the I2C protocol. Connect the SDA (Serial Data Input/Output) and SCL (Serial Clock Input) pins of the device to the appropriate GPIO pins on the microcontroller configured for I2C communication. Utilize the microcontroller's I2C library or implement I2C communication protocols in your code to enable seamless data exchange between the microcontroller and the device. This allows the microcontroller to not only read data from the sensors but also interact with the device for additional functionalities.

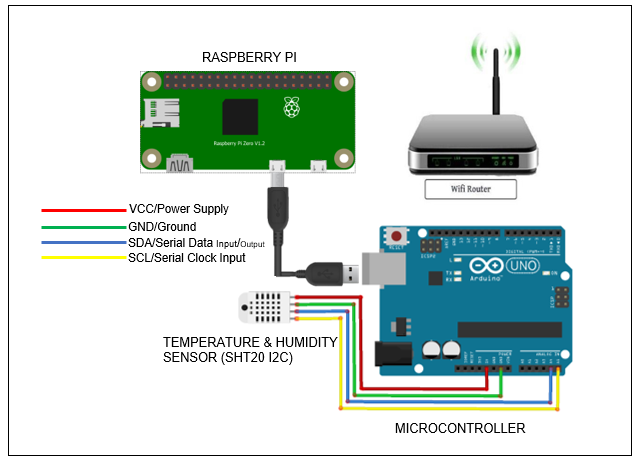


Figure 25. Hardware Architecture Design of Automated Microcontroller

### **Non-Functional Requirement**

1. Reliability

* The Internet of Things - was able to give the BSVF, Bilar, Bohol an efficient and accurate real-time monitoring status.
* The efficient and accurate reports on monitoring conditions would be based on the motion detected by the sensor.
* The Internet of Things - based relies on the soil that is detected by the sensor to send data that will automatically display in detail to the mobile application.

1. Operability

* The Bilar Stem Vegetable Farm (BSVF) in Riverside Bilar Bohol would operate the Microcontroller-based IoT.

1. Maintainability

* The researcher planned to use a Microcontroller. This is the part of the components that allows to automatically store it on the cloud platform which is responsible for sending the data that will automatically display in detail to the mobile application.

1. Availability
   * The LetsuApp: A Smart IoT Monitoring System for Lettuce is always available as long as it is connected.
2. Delivery

* A Smart IoT Monitoring System for Lettuce - would be delivered to the users in BSVF Riverside, Bilar, Bohol.

### **Test Case**

A test case is a set of conditions or variables under which a tester will determine whether an application or software system is working properly or not, a detailed procedure that fully tests an attribute or an aspect of a failure. It is also a set of input values, execution, preconditions, expected results, and executions, created for a particular objective or test condition, such as to exercise a certain program path or to verify compliance with a specific requirement.

These are the test case scenarios conducted during the acceptance testing. The text plain is to let the users use the system and follow the instructions in each test case to test the proposed system. The system should perform the expected result in each test case to be considered successful.

The following are the details of each test case:

### **Test case 1**

Module: LetsuApp: A Smart Internet of Things (IoT) Monitoring System for Lettuce

Instructions:

1. First the user must plug in the microcontroller, then the microcontroller will automatically work on which is the main to start the corresponding for monitoring and its start for detecting the sensors.

Expected Results:

1. This system comprises sensors for detecting temperature and humidity levels, which then send it to a mobile application for user access. Through this application, users can view detailed real-time readings of temperature and humidity. The application's interface is user-friendly, allowing for easy navigation and interpretation of the data. This integrated system provides users with convenient access to vital environmental information, empowering them to make informed decisions and take necessary actions as needed.

Clean-up: Plug off the microcontroller

The following are the details of each test case:

### **Test case 2**

Module: Mobile Application

Instructions:

1. On your cellphone “Click” the LetsuApp application icon.

Expected Results:

1. The home screen will display that contains the LetsuApp brief information.
2. The Sidebar button that consists of different menus including the home, monitor, temperature analysis, humidity analysis, and reports.

### **Test case 3**

Module: Sidebar/ Monitor Button/ Mobile Application

Instructions:

1. On the home screen “Click” the sidebar menu button.
2. Choose and “Click” what menu screen you want to display

Expected Results:

1. The home screen will display that contains the LetsuApp brief information.
2. The monitor, analysis, and reports of choice will be displayed.

### **Business Intelligence**

Business Intelligence (BI) encompasses an organization's capacity to gather, manage, and structure information, aiming to enhance business strategies and decision-making processes by offering advanced solutions that enhance BI capabilities and furnish precise insights. In this context, a microcontroller device is configured to oversee the soil temperature and humidity levels of lettuce crops and transmit this data automatically to an Internet of Things (IoT) platform, such as Firebase. This hardware integration facilitates BI by providing real-time monitoring data specifically related to soil temperature and humidity. The collected data is then visualized and analyzed through mobile applications, empowering users with actionable insights to optimize agricultural practices and enhance crop yield efficiency.

Moreover, the research underscores the potential for further research and development in the field of precision agriculture. This presents an avenue for businesses and academic institutions to invest in innovative solutions that refine existing technologies and develop new methodologies for crop monitoring and management. With the abundance of data generated by IoT-enabled monitoring systems, there is a growing demand for expertise in data analytics and consulting services. Businesses offering such services can assist farmers in interpreting the collected data and implementing strategies to maximize yield while minimizing resource inputs. The research not only signifies technological advancements in agriculture but also highlights the multifaceted opportunities for innovation, collaboration, and value creation across the agricultural industry.

### **Hosting Implementation**

The implementation of hosting for the collected data generated by the microcontroller constitutes a crucial aspect of the operational framework within the context of soil condition monitoring. This data, amassed through the microcontroller's sensors and processes, undergoes hosting on the BISU Bilar Campus HostGator platform. Subsequently, the stored data serves as foundational input for the development and refinement of a machine learning model, which is strategically designed to discern and analyze the prevailing status of soil conditions.

The utilization of the stored data for training the machine learning model signifies a pivotal step towards enhancing the precision and efficacy of the level of soil assessment. Through iterative processes of data analysis and model refinement, the machine learning algorithm becomes increasingly adept at extrapolating meaningful insights about level of soil temperature and humidity. Once the analysis is completed, the derived insights are transmitted to the Firebase server, thereby facilitating seamless dissemination and accessibility of the analyzed data to relevant stakeholders and decision-makers.

This hosting implementation strategy not only underscores the technical intricacies involved in data management and analysis but also underscores the pivotal role of advanced computational techniques, such as machine learning, in augmenting the capabilities of soil condition monitoring systems. Moreover, by leveraging robust hosting infrastructure and cloud-based services, such as HostGator and Firebase, the implementation ensures scalability, reliability, and accessibility of level of soil gathered data, thereby fostering informed decision-making and sustainable agricultural practices.

### **Hardware Performance Evaluation**

Table 5 shows the cost of the developed LetsuApp: A Smart Internet of Things (IoT) Monitoring System for Lettuce as a result of the development and operation of the implemented monitoring system. It determines the exact investment cost needed for the system.

### **Table 5. Hardware Cost**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Items | Specification | Qty | Unit | Price | Total |
| Microcontroller | Arduino Uno | 1 | pc | Php499 | Php499 |
| Sensor | Temperature & Humidity Sensor SHT20 I2C | 1 | pc | Php1300 | Php1300 |
| HDMI |  | 1 | pc | Php250 | Php250 |
| Case | Acrylic Case | 1 | pc | Php50 | Php50 |
| Pocket Wi-Fi | Smart Bro | 1 | pc | Php599 | Php599 |
| Physical Base |  | 1 | pc | Php250 | Php250 |
| **TOTAL COST:** | | | | | **Php2948** |

**Physical Layout**

The physical layout is one of many attributes of the device friendly user- interfaced. It should be designed in a way of the physical layout can navigate the device quickly.

Preview 1 shows the automated microcontroller and it covers the specification of the basic functionality of the prototype.

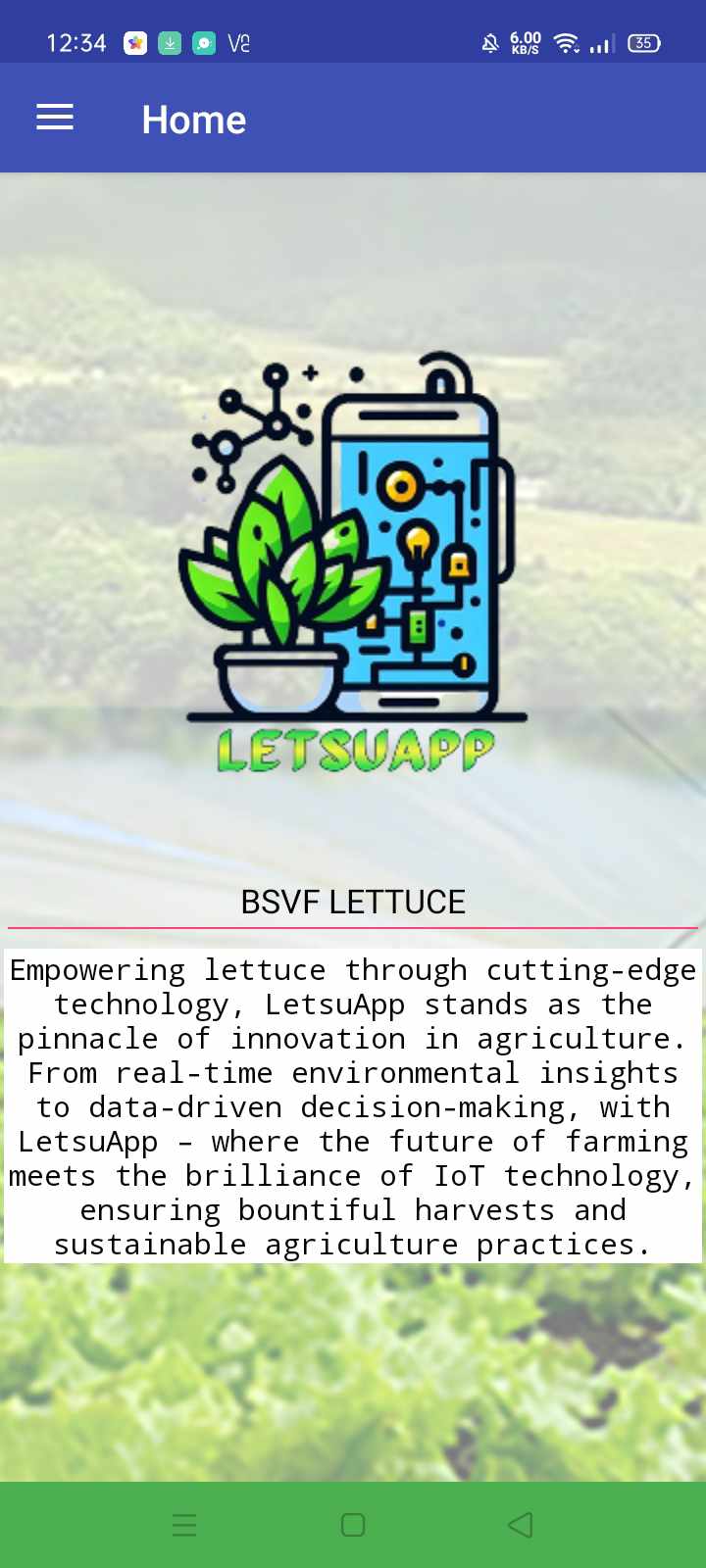


Preview 1. Automated Microcontroller

**Screen Layout**

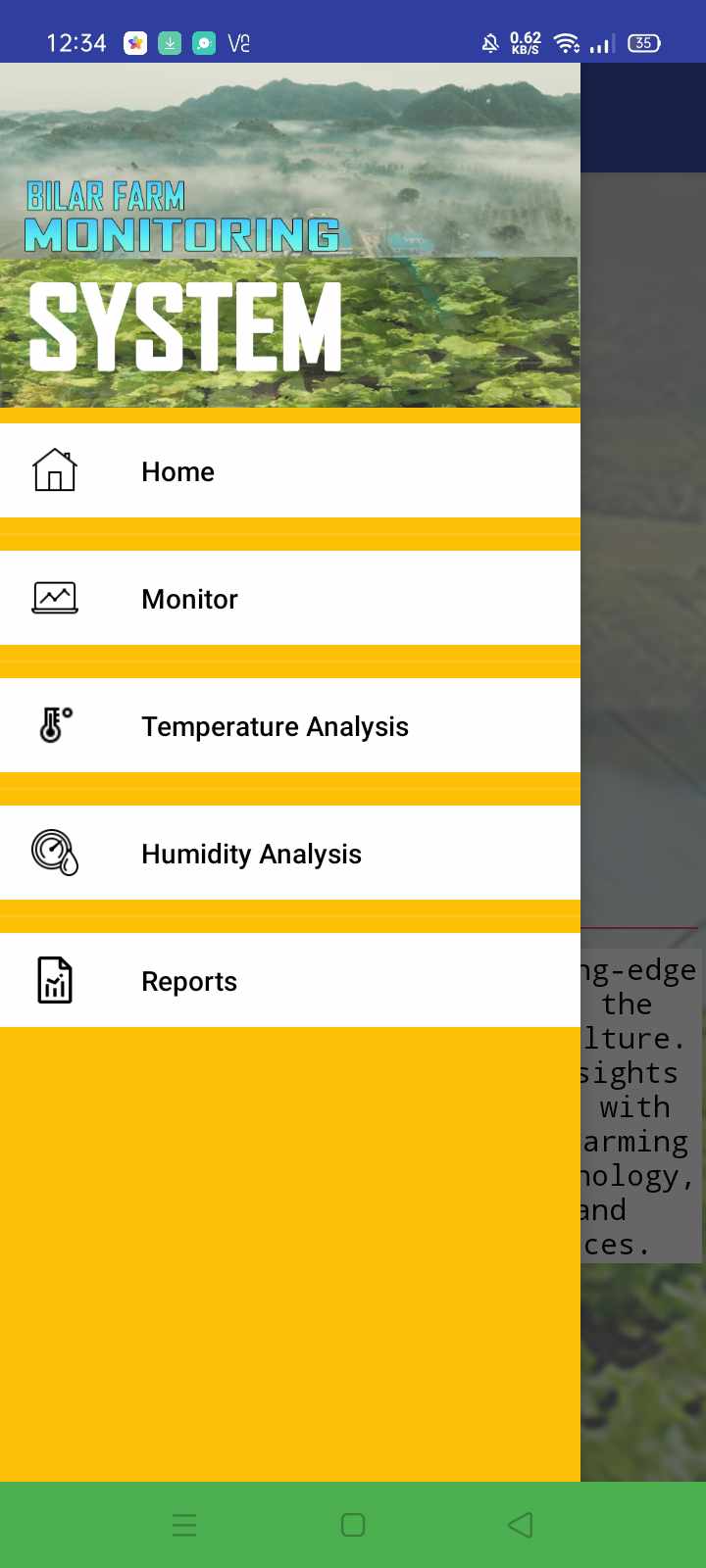
The Screen layout is one of the attributes of the system’s user-friendliness. It should be designed in such a way the browsers can navigate the system quickly and easily and it should provide a clear recognition of the task the users to perform.

Preview 2 shows the screen layout of the mobile application home screen.



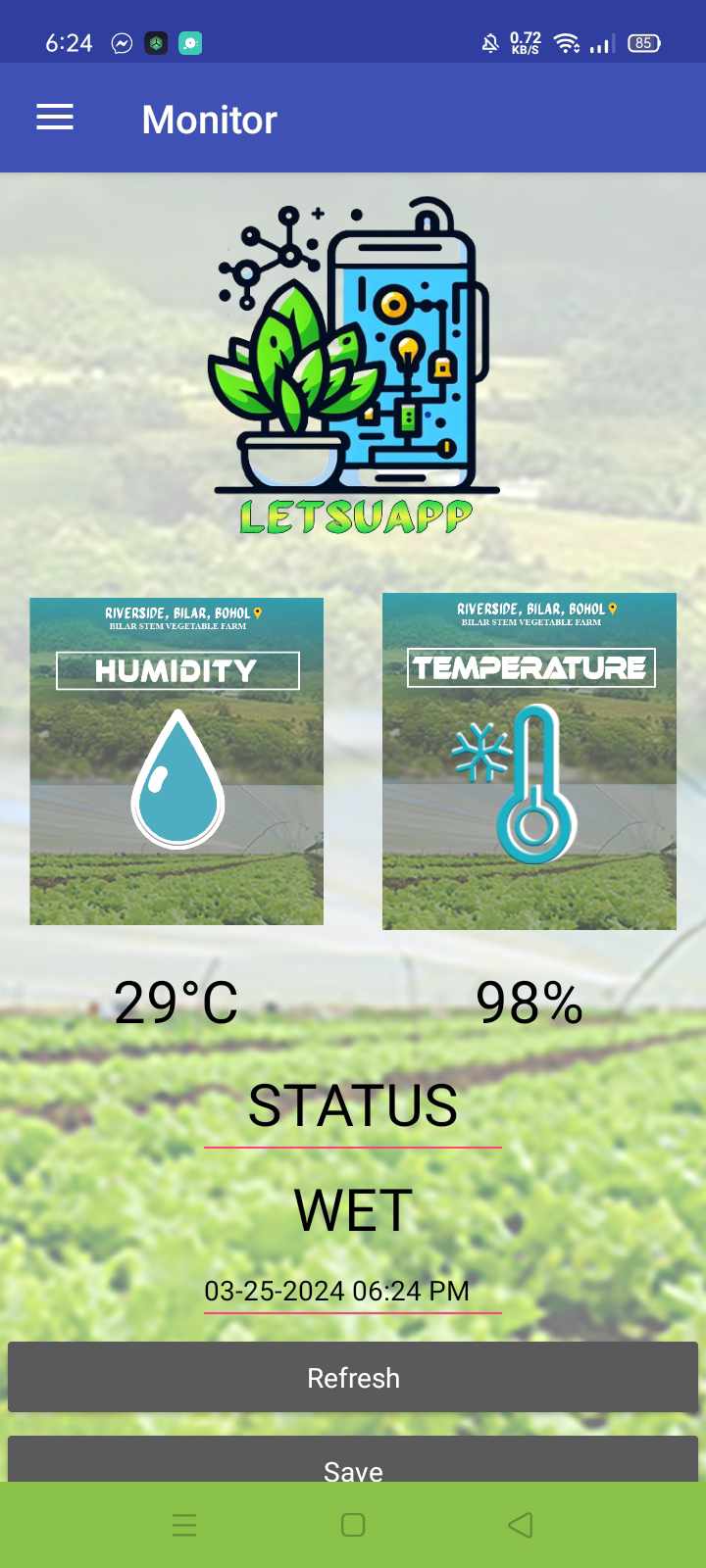
Preview 2. Mobile Application Home screen

Preview 3 shows the screen layout of the mobile app for Sidebar Menu



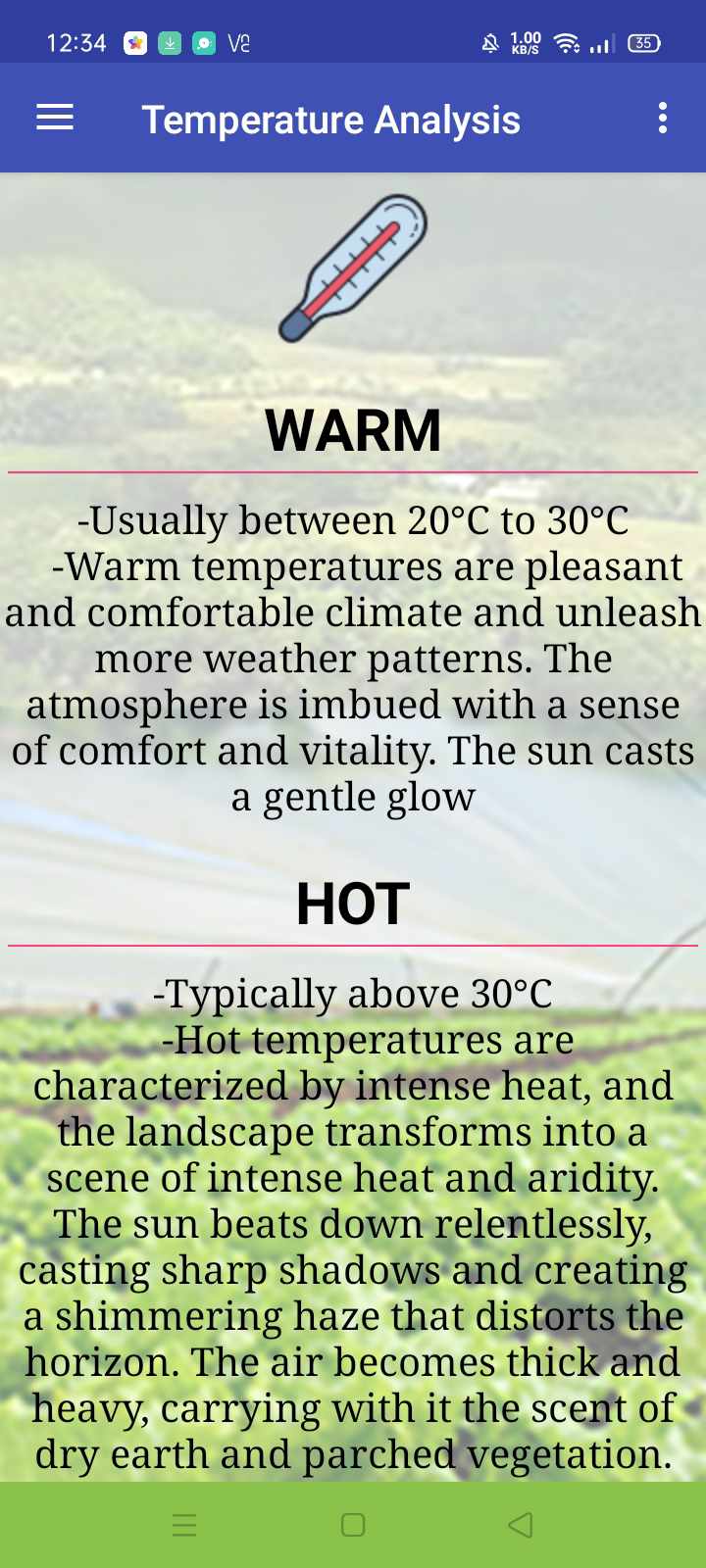
Preview 3. Mobile Application Sidebar Menu

Preview 4 shows the screen layout of the Mobile Application for Monitor



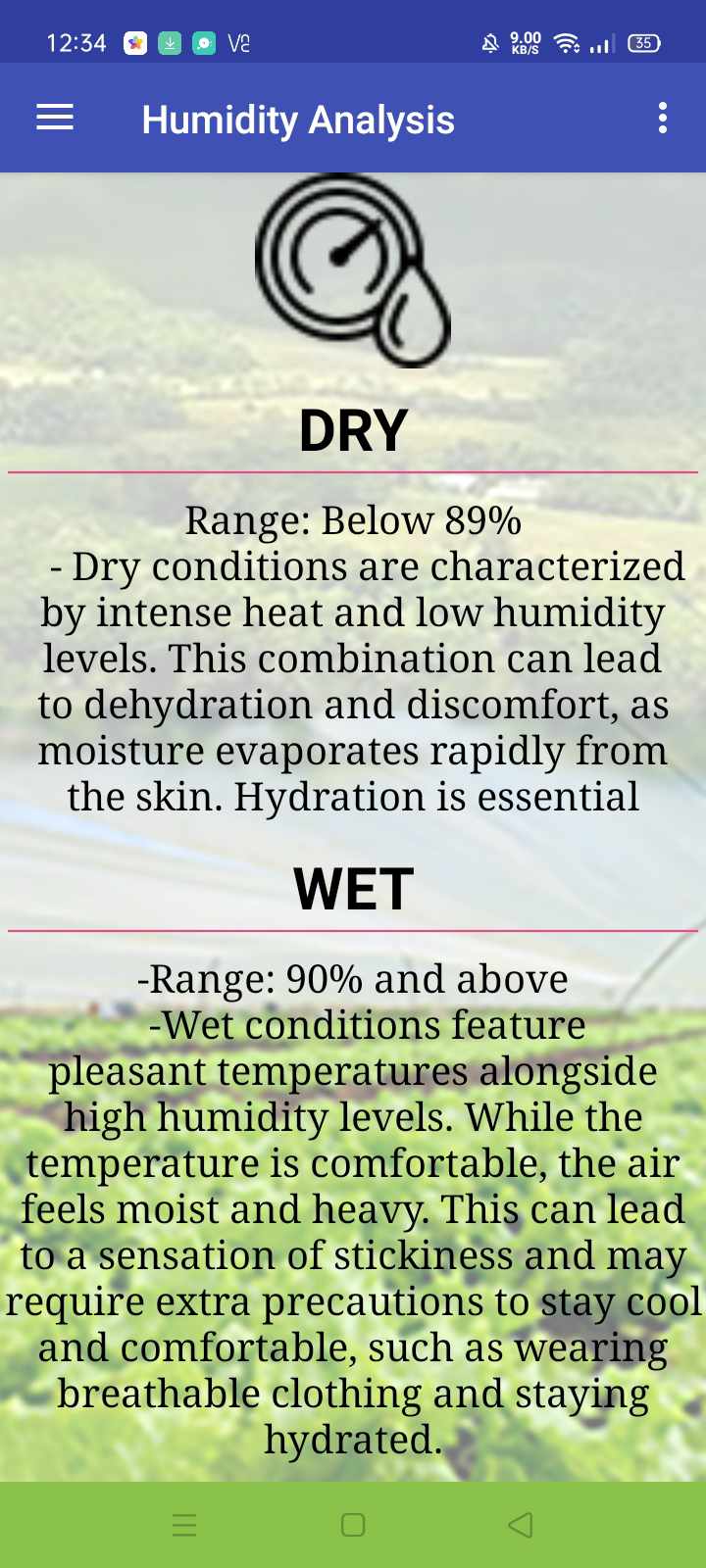
Preview 4. Mobile Application Monitor

Preview 5 shows the screen layout of the Mobile Application for Temperature Analysis



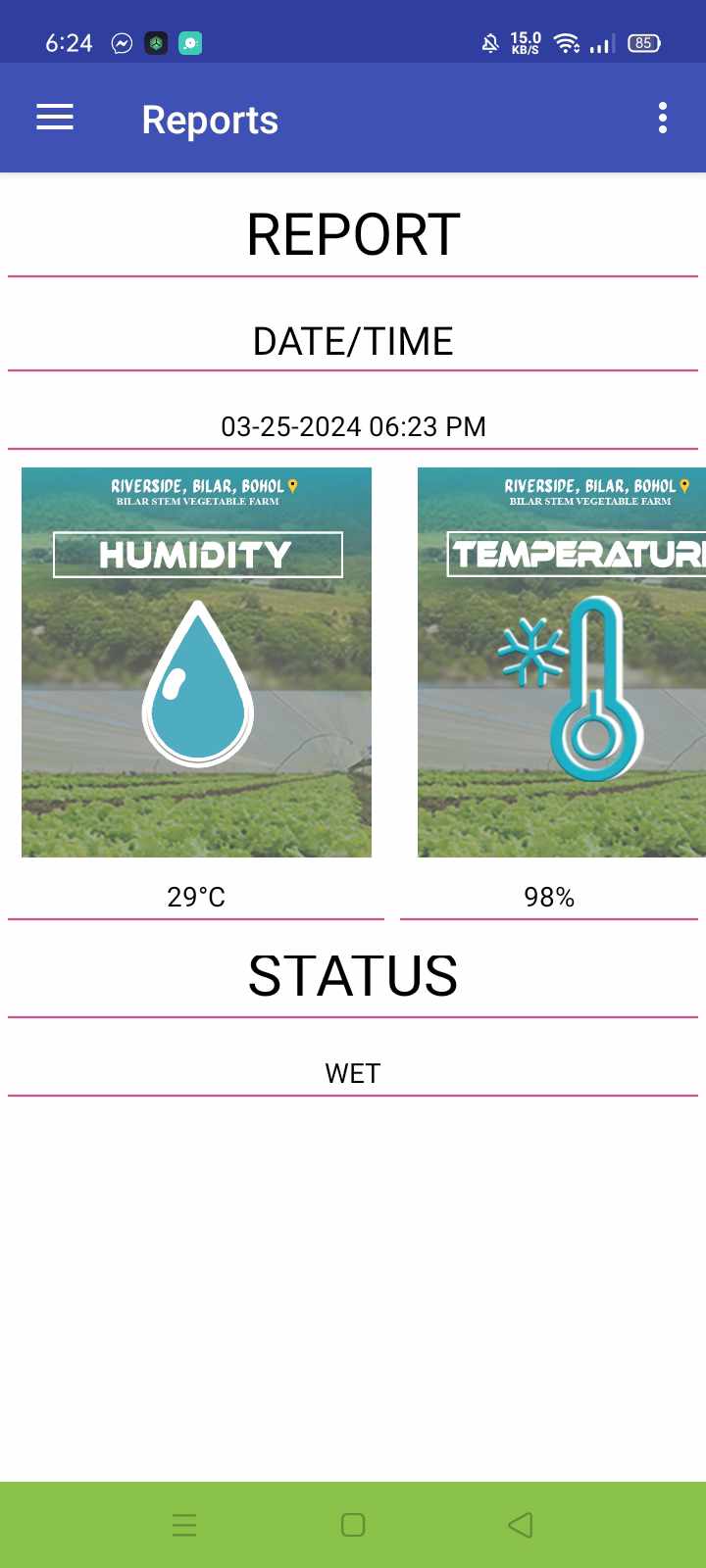
Preview 5. Mobile Application Temperature Analysis

Preview 6 shows the screen layout of the Mobile Application for Humidity Analysis



Preview 6. Mobile Application Humidity Analysis

Preview 7 shows the screen layout of the Mobile Application for Reports



Preview 4. Mobile Application Monitor

### **Testing and Evaluation**

To enhance the system's functionality and performance, rigorous testing and assessment procedures were implemented to validate its capability to deliver accurate outputs, process data efficiently, and react effectively to user inputs. The evaluation process involved comprehensive testing protocols designed to scrutinize various aspects of the system's functionality, including its ability to accurately measure the soil temperature and humidity of lettuce, as well as its responsiveness to commands and instructions from users. Through systematic testing procedures, the system's reliability and accuracy were thoroughly assessed, ensuring that it meets the stringent requirements for providing dependable and precise information essential for informed decision-making in agricultural practices.

The study focused on evaluating the system's usability, taking into account the user's perspective to assess its technical performance. Usability testing involved scenarios simulating real-world usage conditions, allowing testers to gauge the system's intuitiveness, ease of navigation, and overall user experience. By soliciting feedback from users and incorporating their insights into the assessment process, the study aimed to identify areas for improvement and refinement, ultimately enhancing the system's usability and ensuring seamless interaction between users and the technology. This user-centered approach to testing and assessment underscores the commitment to delivering a user-friendly and efficient system that meets the needs and expectations of its intended users effectively.

### **System Usability**

Table 6 shows below the based on the results of the system usability test using the questionnaire prepared by ISO 25010, the target users perceived the system to be acceptable with a general rating of "Strongly Agree". The respondents

stated that the system was suitable for recording student data and met their expectations in terms of functions and capabilities. There may be room for improvement but overall, the system performed well in the test.

Table 6

System Usability Assessment Result N=17

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria for System Usability** | | **Weighted**  **Mean** | **Rating**  **Strongly Agree** |
| **Functionality Suitability** | |  |  |
|  | Functional completeness | 4.24 | Strongly Agree |
|  | Functional correctness | 4.41 | Strongly Agree |
|  | Functional appropriateness | 4.65 | Strongly Agree |
| **Performance Efficiency** | | | |
|  | Time behavior | 4.24 | Strongly Agree |
|  | Resource utilization | 4.29 | Strongly Agree |
|  | Resource utilization | 4.12 | Agree |
| Compatibility | | | |
|  | Co-existence | 4.35 | Strongly Agree |
|  | Interoperability | 4.24 | Strongly Agree |
| **Usability** | | |  |
|  | Appropriateness Recognizability | 4.41 | Strongly Agree |
|  | Learnability | 4.59 | Strongly Agree |
|  | Operability | 4.41 | Strongly Agree |
|  | User error protection | 4.24 | Strongly Agree |
|  | User interface aesthetics | 4.65 | Strongly Agree |
|  | Accessibility | 4.41 | Strongly Agree |
| **Reliability** | | | |
|  | Maturity | 4.35 | Strongly Agree |
|  | Availability | 4.29 | Strongly Agree |
|  | Recoverability | 4.24 | Strongly Agree |
| **Security** | | | |
|  | Confidentiality | 4.41 | Strongly Agree |
|  | Integrity | 4.41 | Strongly Agree |
|  | Non-repudiation | 4.35 | Strongly Agree |
|  | Accountability | 4.83 | Strongly Agree |
|  | Authenticity | 4.12 | Agree |
| **Maintainability** | | | |
|  | Modularity | 4.59 | Strongly Agree |
|  | Reusability | 4.35 | Strongly Agree |
|  | Analyzability | 4.25 | Strongly Agree |
|  | Modifiability | 4.53 | Strongly Agree |
|  | Testability | 4.53 | Strongly Agree |
| **Maintainability** | | | |
|  | Adaptability | 4.18 | Agree |
|  | Installability | 4.29 | Strongly Agree |
|  | Replaceability | 4.65 | Strongly Agree |
| **Overall Mean** | | **4.35** | **Strongly Agree** |

## CHAPTER III

**SUMMARY OF FINDINGS, CONCLUSIONS, AND RECOMMENDATION**

This chapter deals with the summary of findings, conclusions, and recommendations based on the results of the study.

### **Summary of Findings**

Bilar Stem Vegetable Farm (BSVF) currently relies on manual monitoring methods, which, though based on experience, can lead to variations in the accuracy of soil condition assessments depending on the state of the soil. To address this issue, the researcher developed the LetsuApp: A Smart Internet of Things (IoT) Monitoring System for Lettuce, leveraging a microcontroller to detect real-time levels of temperature, and humidity, of lettuce at BSVF. This system automates data processing upon detection and seamlessly stores it on a cloud platform, ensuring efficient transmission to a mobile application. Through the utilization of the microcontroller, the data is presented comprehensively and in detail within the mobile application, providing valuable insights for informed decision-making.

Following the development and pilot testing of the system, a survey rating questionnaire was administered to target users to evaluate the system's quality. The results revealed an average weighted mean of 4.35 for hardware quality and system factor, indicating a consensus among respondents that the developed system is highly beneficial and usable. This interpretation of "Strongly Agree" signifies users' confidence in the system's capabilities and its potential to significantly enhance monitoring and management processes at BSVF, underscoring the value of incorporating IoT technologies to improve agricultural practices and optimize lettuce outcomes.

### **Conclusion**

Based on the findings of the study, the researchers have identified that the implementation of manual monitoring methods at Bilar Stem Vegetable Farm (BSVF) has demonstrated limitations, particularly in ensuring consistent and accurate soil condition assessments due to varying factors affecting environmental state. However, to overcome these challenges, the development of the LetsuApp: A Smart Internet of Things (IoT) Monitoring System for Lettuce has proven to be a significant advancement. By harnessing the capabilities of a microcontroller, this system enables real-time detection and monitoring of temperature, humidity of lettuce, while automating data processing and storage on a cloud platform. The integration of the microcontroller facilitates the efficient transmission of data to a mobile application, offering comprehensive insights for informed decision-making in lettuce production. Moreover, the positive response from target users, as evidenced by the rating obtained through the survey questionnaire, underscores the system's effectiveness and usability. This strong vote of confidence highlights the potential of IoT technologies to revolutionize agricultural practices and optimize outcomes, positioning BSVF at the forefront of innovation in lettuce cultivation.

### **Recommendation**

Based on the conclusions. It is highly suggested that the developed system should be implemented. For a successful implementation of the proposed system:

1. Regular system maintenance must be performed to ensure the system's good functionality.
2. Secure good internet connectivity to avoid lagged time and real-time in sending of data.
3. It is recommended to use high-cost hardware components and materials to ensure long-lasting system runtime.
4. The local farmers may undergo training on the microcontroller–based technology.
5. Workshop to broaden their competencies in control automation.
6. It is recommended to find other mobile application inventors that are suitable for storing the data and reports.

## REFERENCES

Anisha, M., Arsad, U. M., Dhanalakshmi, K., Anitha, S., Benisha, M., Chezhiyan,

P., & Elliot, C. J. (2021, February). Arduino-based low-cost monitoring system for small-scale farmers. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 1120-1127). IEEE.

Ashifuddin Mondal, M., & Rehena, Z. (2018, January). Iot-based intelligent agriculture field monitoring system. In 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 625-629). IEEE.

Changmai, T., Gertphol, S., & Chulak, P. (2018, January). Smart hydroponic lettuce farm using Internet of Things. In 2018 10th international conference on knowledge and smart technology (KST) (pp. 231-236). IEEE.

Erhan, L., Ndubuaku, M., Di Mauro, M., Song, W., Chen, M., Fortino, G., ... & Liotta,

A. (2021). Smart anomaly detection in sensor systems: A multi-perspective review. Information Fusion, 67, 64-79.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach

in classification. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings (pp. 986-996). Springer Berlin Heidelberg

Haji, S. H., & Sallow, A. B. (2021). IoT for smart environment monitoring based on Python: a review. Asian Journal of Research in Computer Science, 9(1), 57-70.

Hassan, S. I., Alam, M. M., Illahi, U., Al Ghamdi, M. A., Almotiri, S. H., & Su’ud, M.

(2021). A systematic review on monitoring and advanced control strategies in smart agriculture. Ieee Access, 9, 32517-32548

Maduranga, M., & Abeysekera, R. (2020). MACHINE LEARNING APPLICATIONS IN IOT BASED AGRICULTURE AND SMART FARMING: A REVIEW. International Journal of Engineering Applied Science and Technology, 04(12), 24–27.

Omar, S., Ngadi, A., & Jebur, H. H. (2013). Machine learning techniques for anomaly detection: an overview. International Journal of Computer Applications, 79(2).

Perin, Max Angelo & Cuaton, Maria & Rapirap, Apple. (2018). Automated Aerator System For Controlling Dissolved Oxygen And Monitoring Acidity And Temperature On Aquatic System. 10.13140/RG.2.2.18762.59846/1.

Perin, M. A. D., Feliscuzo, L. S., & Sta. Romana, C. L. C. (2022). EskayApp: An Eskaya-Latin Script OCR Transliteration e-Learning Android Application using Supervised Machine Learning. In Proceedings of the 5th International Conference on Digital Technology in Education (ICDTE '21) (pp. 1–7). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3488466.3488467.

Phutane, R. V. (2022). IoT: Smart Monitoring System. Journal for Research in Applied Science and Engineering Technology,(IJRASET), 2321-9653.

P. Bhadani nd V. Vashisht, "Soil Moisture, Temperature and Humidity Measurement Using Arduino," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 567-571, doi: 10.1109/CONFLUENCE.2019.8776973

Savic, M., Lukic, M., Danilovic, D., Bodroski, Z., Bajović, D., Mezei, I & Jakovetić, D. (2021). Deep learning anomaly detection for cellular IoT with applications in smart logistics. IEEE Access, 9, 59406-59419.

Sehrawat, D., & Gill, N. S. (2019, April). Smart sensors: Analysis of different types of IoT sensors. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 523-528). IEEE.

Tiwary, A., Mahato, M., Chidar, A., Chandrol, M. K., Shrivastava, M., & Tripathi, M. (2018). Internet of Things (IoT): Research, architectures and applications. International Journal on Future Revolution in Computer Science & Communication Engineering, 4(3), 23-27.