TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
1338 Arlegui St., Quiapo, Manila

A Design Project Submitted to the Department of Computer Engineering
In Partial Fulfillment of the Requirements for the Course CPE 413 - CpE Design 2
for the degree of Bachelor of Science in Computer Engineering

dotBean: A Design of an Automated Temperature-Controlled Storage with Threshold-based Algorithm and Real-time Moisture Monitoring via Mobile App for Green Coffee Beans in La Trinidad, Benguet

Submitted by:
Gagabi, Rome Angelo A.
Mariscotes, Reinier D.
Root, Guillen Minerva A.
Rotoni, Carlo R.
Agcaoili, Raven D.

May 2024

APPROVAL SHEET

This design project entitled "**dotBean: A Design of an Automated Temperature-Controlled Storage with Threshold-based Algorithm and Real-time Moisture Monitoring via Mobile App for Green Coffee Beans in La Trinidad, Benguet**" designed and submitted by Rome Angelo A. Gagabi, Reinier D. Mariscotes, Guillen Minerva A. Root, Carlo R. Rotoni, and Raven D. Agcaoili, in partial fulfillment of the requirements for the course CPE413 - CPE Design Project 2 is hereby approved.

Dr. Jennalyn N. Mindoro
Design Project Adviser

Dr. Jennalyn N. Mindoro
Course Adviser

Dr. Jennifer B. Enriquez
Design Project Lead Panel

Engr. Marte Nipas
Design Project Panel

Mr. Dennis Nava
Design Project Panel

Approved by:

Dr. Jennalyn N. Mindoro
Assistant Program Chair, Computer Engineering Department

Dr. Jennifer B. Enriquez
Program Chair, Computer Engineering Department

Dr. Marianne L. Yumul, ASEAN Eng.
Dean, College of Architecture and Engineering

ACKNOWLEDGEMENT

Completing this design project was made possible through the support and contributions of several individuals and organizations, for which the developers extend sincere gratitude. The following individuals and organizations deserve special recognition for their roles in this project.

The developers wish to express profound appreciation to Dr. Jennalyn N. Mindoro, the dotBean's Project Adviser and Course Adviser, whose guidance, encouragement, and critical suggestions were invaluable throughout this project.

The developers also wish to recognize the valuable feedback, criticism, and guidance from panel members Engr. Marte Nipas and Mr. Dennis Nava, headed by Dr. Jennifer B. Enriquez, during project consultations and presentations. This significantly improved presentation skills and the overall quality of this project.

Gratitude is extended to Mr. Eli Natividad and his team at Farm to Cup Benguet for their unwavering support and allowing dotBean's field testing at their DIY Brew Bar in La Trinidad, Benguet.

The developers are also grateful to Engr. Jimbo Ian Constantino for his assistance in planning the hardware design options and material acquisition, and to Engr. Cayetano Hiwatig for his support in developing and validating the hardware design and testing.

Appreciation is due to Mr. Neilson Diñoso for his contributions to the development of the project's mobile and website applications, and to Mr. Rodelio Orate and Mr. Jomarie de la Cruz for their assistance in the final prototype fabrication and development.

Special acknowledgement is also due to Ms. Chona P. Opeña, who provided unwavering support to the proponents in all aspects of the project from its inception through its completion.

Finally, the developers extend heartfelt thanks to their family and friends for their unwavering support and understanding throughout this project.

Their combined efforts were crucial in carrying out and concluding this study. Their understanding, guidance, and provision of resources provided reliable support for this research to move forward.

PROJECT ABSTRACT

Environmental conditions such as temperature and humidity can adversely affect the quality of coffee beans, leading to moisture absorption and mold growth. This study addresses these challenges by designing a storing system, the dotBean, for arabica green coffee beans, focusing on automated temperature control and moisture content monitoring. The project explores three design options for optimal storage conditions, emphasizing performance, response time, and power consumption. Design option two, which employs a radiator fan and a mobile application for monitoring, was the most effective, with a performance time of 16 minutes and 51 seconds, a response time of 198 microseconds, and a maximum power consumption of 0.1853 kWh. The assessment of project objectives yielded results, with the temperature and humidity sensor achieving a standard deviation of 1.13 and the moisture sensor achieving a standard deviation of 0.27 The device maintained the coffee beans' moisture content within the target range of 11% to 12.5% in 99.54% of tests. Response time in relation to automated fan control by the system was successful in 100% of the testing, with an average response interval of 196.67 microseconds. The device's power consumption averaged 0.0296 kWh, further reinforcing the reliability of the dotBean project. These findings suggest that the dotBean project effectively monitors and maintains coffee bean quality under controlled conditions. The system's design provides a potential solution for enhancing the storage process, which may benefit coffee producers not only by mitigating quality loss but also by reducing associated financial impacts.

Keywords: *Coffee Farmer, Monitor, Temperature, Humidity, Moisture Content*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: PROJECT BACKGROUND

1.1 The Project

In the Philippines, coffee is not just a commodity; it is a testament to resilience, a symbol of pride, and a source of livelihood for many families. Stretching back to the start of coffee in the country, coffee is not endemic to the Philippines; the first coffee beans, Liberica beans, were brought over to the Philippines in 1740 by two Spanish Franciscan friars and planted in the cool, elevated plantations of Lipa, Batangas, laying the foundation for the country's rich coffee heritage and the subsequent growth of its coffee industry.

Coffee cultivation and consumption hold a significance that transcends mere economic value. According to Tampon, V. (2023), the Philippines is the 14th largest coffee producer globally, contributing significantly to the global coffee industry; it is expected to grow at a Compound Annual Growth Rate (CAGR) of 3.5% from 2021 to 2025. However, despite the production rate increasing by 1.62% between 2021 and 2022, consumption fell short of production by 4.07% in 2022. Between 2018 and 2020, coffee consumption increased by 2.1%, while production has declined annually by 3.4% over the past ten years (Philippine et al. 2021-2025, 2022).

Regional challenges contribute to this complex dynamic. For instance, in Benguet, coffee farmers face difficulties due to the region’s low temperatures. Motisi et al. (2019) state that changing climate conditions threaten coffee farming, including temperature shifts and precipitation variations. Furthermore, Arabica coffee cultivation is projected to decrease due to climate change, indicating potential challenges for coffee-growing areas like Benguet (Koutouleas et al., 2022). These local issues underscore the broader global challenges within the coffee industry, where regional environmental factors impact production and consumption trends.

1.1.1 Current Scenario and Human Problem

Farmers associated with the Farm to Cup Benguet are primarily concerned with low temperature, which increases relative humidity, introduces moisture, and spoils the beans. This issue results in losses due to mold growth and spoilage.

Coffee beans are sensitive to low temperatures. Rosa et al. (2011) found that storing beans at 10°C and 20°C substantially negatively impacted their vigor and seedling quality. The adverse effects were most pronounced at 20°C. The average temperature in La Trinidad, Benguet, ranges between 13°C and 25°C. Despite being ideal for cultivation, maintaining the moisture content of crops during storage remains a challenge.

After the post-harvesting production, the client places the coffee beans in an airtight bag to prevent moisture absorption. They put these bags inside a rice sack and stored them inside a room for added protection. They ensure the Beans are kept from direct sunlight and contact with cement to prevent light exposure and odor absorption (*see Figure 1-1*).

Figure 1-1. Client's Storage Room

Nonetheless, the farmers are still experiencing situations where the coffee beans are affected by moisture. When the beans absorb the moisture, molds are created, which results in deterioration and loss of freshness. The farmers employ a quality control process for their coffee beans, relying on manual monitoring techniques. These involve tag affixation containing information such as the harvest date, when the coffee beans were stored, and moisture content. Farmers also use a moisture tester to check the beans' moisture content upon opening sacks. They must air dry it if it falls outside the 11-12.5% range. Despite corrective actions, mold growth remains an enduring issue.

On the other hand, molds are not the sole indicator of bean deterioration. Farmers can assess the quality of the beans by using visual cues such as appearance and color, as well as sensory attributes like smell and taste (Yazawa, 2023). Some of the interviewed farmers, located in Barlig, Mountain Province, take a different approach to storage methods, using jute bags to store their coffee beans. However, instead of maintaining the desired 11% to 12.5% moisture content, the beans tend to reach up to 14%, leading to losses.

1.1.2 Problem Validation

To attest to the validity of the abovementioned problem, the developers conducted a survey involving various individuals with ties to the coffee industry to ensure a comprehensive understanding of the issue. Below are the survey results:

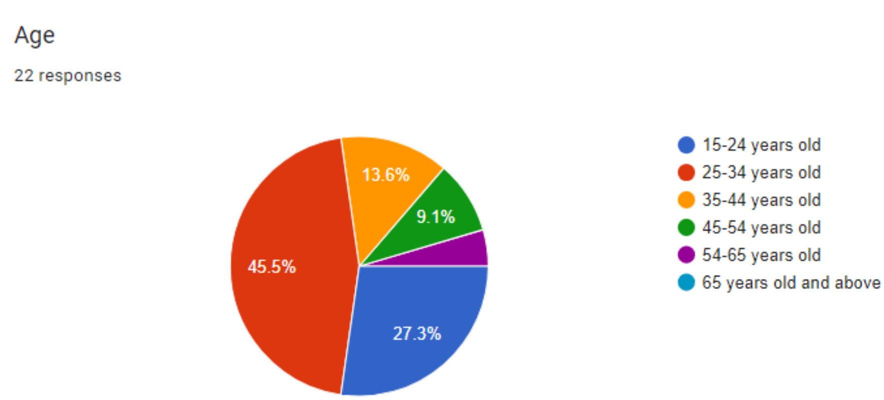


Figure 1-2. Demographic Profile of the Respondents in Terms of Their Age

Figure 1-2 presents an overview of the respondents' demographic characteristics, focusing on their age distribution. Among the 22 respondents, 10 (45.5%) fall within the age group of 25-34 years. Additionally, six respondents (27.3%) are 65 and above, while 3 (13.6%) belong to the 35-44 age bracket. Furthermore, two respondents (9.1%) fall into the 45-54 age category. Finally, one respondent (4.5%) is in the 54-65 age range.

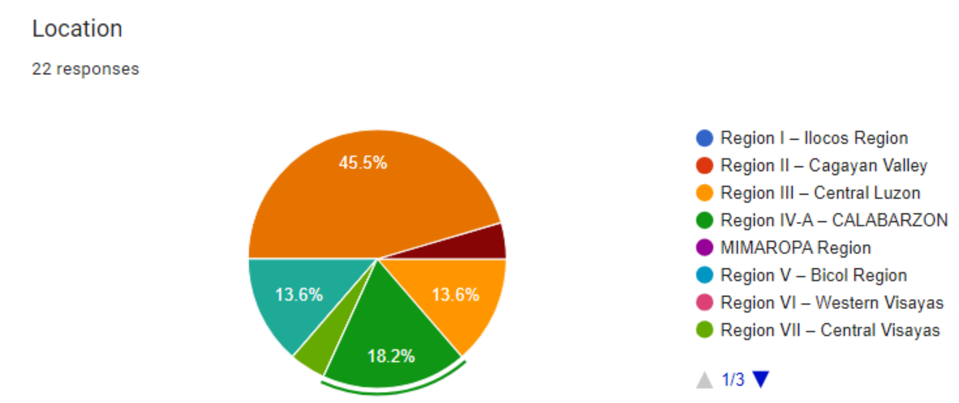


Figure 1-3. Demographic Profile of the Respondents in Terms of Their Location

Figures 1-3 illustrate the respondents' demographic profile regarding their location. 10 out of 22 respondents (45.5%) are located in NCR, while four respondents (18.2%) are located in Region IV-A – CALABARZON, while both Central Luzon and Davao Region have three respondents (13.6%). Lastly, Region VII and CAR have only one respondent (4.5%).

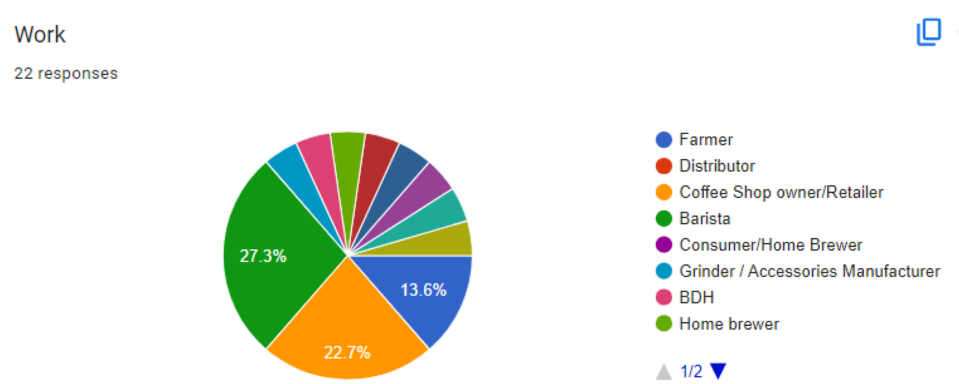


Figure 1-4. Demographic Profile of the Respondents in Terms of Their Occupation

Figures 1-4 illustrate the respondents' demographic profile regarding their occupation. Out of 22 respondents, 6 (27.3%) are Baristas, 5 (22.5%) are Coffee Shop owners, and 3 (13.6%) are Coffee Farmers. The remaining 8 (36.6%) respondents are divided into occupations such as Legal Compliance Administrator, Supervisor, Bean Trader, Distributor, and Grinder/Accessories

Manufacturer.

Figure 1-5. Survey Responses to Question #1

Figures 1-5 illustrate the respondents' answers regarding whether they store their coffee beans. Twenty-two out of 22 respondents (100%) agreed that they do.

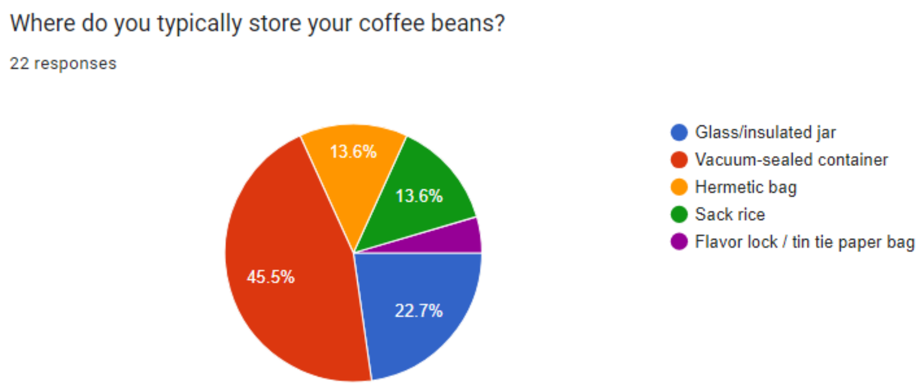


Figure 1-6. Survey Responses to Question #2

Figure 1-6 depicts the respondents' storage preferences for their coffee beans. The majority, accounting for 10 (45.5%) of respondents, opt for vacuum-sealed containers. A total of 5 respondents (22.7%) choose to store their coffee beans in glass/insulated jars. Both hermetic bags and sack rice storage methods are favored by three respondents, constituting 13.6% of the sample for each category. Lastly, only one respondent (4.5%) reported using a flavor lock or paper bag for coffee bean storage.

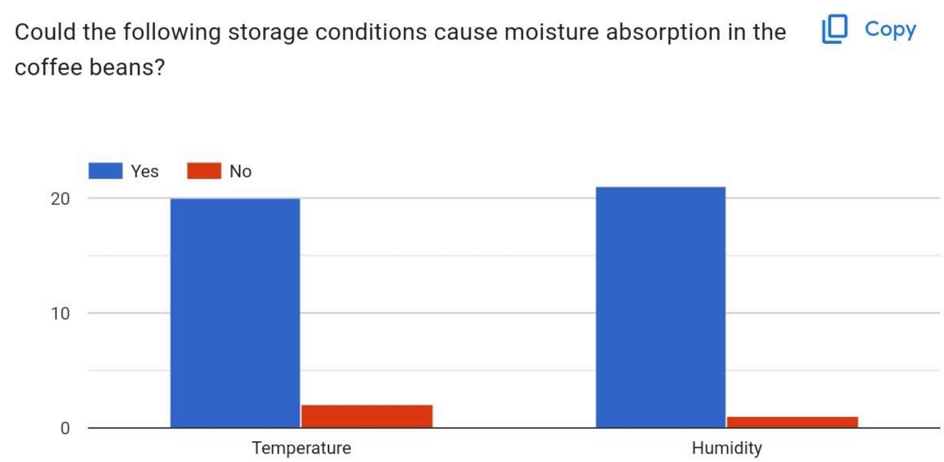


Figure 1-7. Survey Responses to Question #3

Figure 1-7 illustrates the respondents' answers regarding whether the storage condition could result in moisture absorption. Based on the results, 20 out of 22 respondents (91%) agreed on temperature, and 21 out of 22 respondents (95.5%) agreed that humidity can cause moisture.

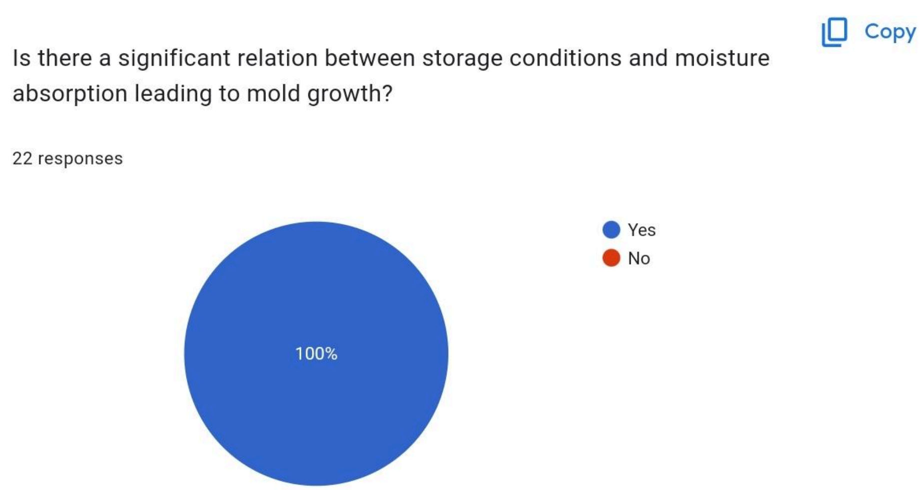


Figure 1-8. Survey Responses to Question #4

Figures 1-8 illustrate the respondents' answers regarding the significant relation between storage conditions and the freshness and quality of the coffee beans. Twenty-two respondents (100%) agreed there is a substantial relationship between the storage conditions and moisture absorption of the coffee beans leading to mold growth.

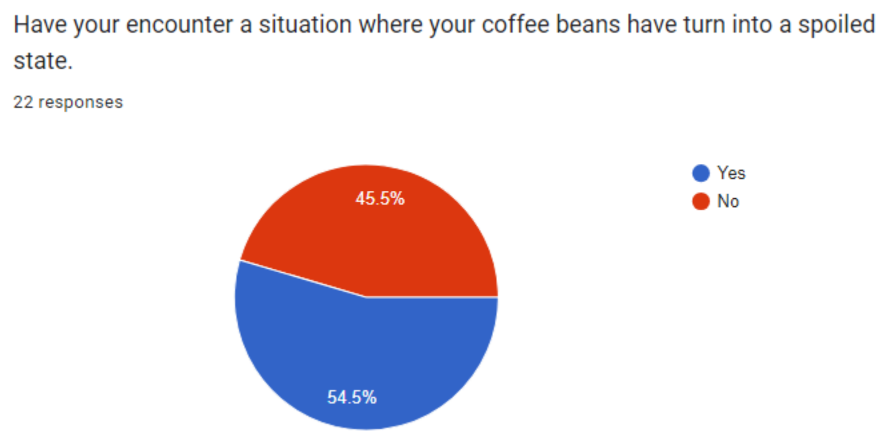


Figure 1-9. Survey Responses to Question #5

Figures 1-9 illustrate the respondents' answers in terms of their encounter with their coffee beans turning into a spoiled state. Out of the 22 respondents, 12 (54.5%) reported encountering instances where their coffee beans turned spoiled. Conversely, 11 respondents (45.5%) disagreed, indicating they had not experienced coffee bean spoilage.

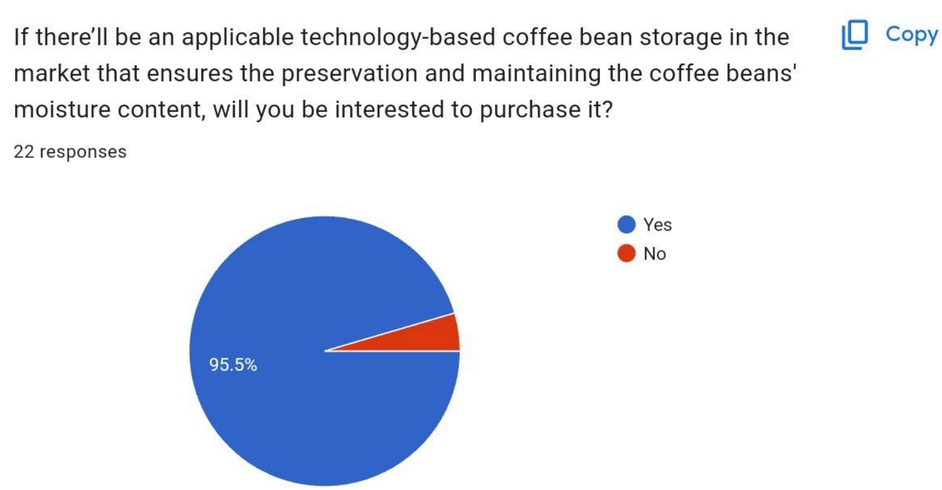


Figure 1-10. Survey Responses to Question #6

Figures 1-10 illustrate the respondents' answers that they will purchase a technology-based coffee bean storage application to ensure maintaining coffee beans' moisture content. Based on this result, 21 out of 22 respondents (95.5%) agreed to purchase the system.

1.1.3 Findings

Based on the survey, it was concluded that:

- 12 out of 22 (54.55%) respondents reported experiencing spoilage of their coffee beans despite adhering to the proper storage practices by storing beans in their designated containers/bags.
- An average of 97.62% concurred that temperature and humidity influenced the moisture content of the coffee beans during storage.
- 22 out of 22 respondents (100%) agreed there is a significant relationship between the storage conditions and the moisture content of the coffee beans.

1.1.4 Existing Solutions

The developers can better understand the client's needs by evaluating the current solutions and providing tailored recommendations for a more practical approach. Therefore, table 1-1 below summarizes the advantages and disadvantages of the client's existing solutions, which can help the developers identify areas for improvement and optimization.

Table 1-1. Client's Existing Solutions for Storing the Coffee Beans

Existing Solutions		Advantages	Disadvantages
Manual Monitoring	Tagging	● Ensure quality control for coffee beans, maintaining their condition and characteristics for top-notch product quality. ● Cost-effective.	● Entails labor costs and data inconsistency risks without experience. ● Raises the risk of human errors in data recording and monitoring. ● Time consuming.

Existing Solutions		Advantages	Disadvantages
	Moisture Tester	● Monitors coffee beans' moisture content during storage ● Reduces the risk of mold growth	● Costly, it is inaccessible to small-scale farmers. ● Regular calibration and upkeep are needed.
Hermetic Bag *(GrainPro Bag)*		● It prevents fungal growth, mold development, and insect infestation. ● Shields coffee beans from light, humidity, and temperature. ● Offers long-term savings.	● Costly, it is inaccessible to small-scale farmers. ● Compressed air can build up and damage the beans' integrity when stored for extended periods. ● Not efficiently remove existing oxygen, potentially causing coffee compounds to oxidize gradually.
Rice sack		● Easy to handle due to lightweight and practical size. ● Affordable and widely accessible. ● Offers extra protection against moisture and pests.	● It is not fully shielded against pests' temperature/humidity changes. ● Not fully controlling oxygen allows gradual penetration that could lead to oxidation and quality decline. ● Odor contamination.

Existing Solutions	Advantages	Disadvantages
Jute Bag	● Allows air circulation, reducing moisture and mold risks. ● Biodegradable and eco-friendly. ● Affordable.	● It needs to be airtight, making humidity control challenging. ● Less pest-resistant, leading to infestations. ● It needs to be more durable, leading to wear and tear and the need for more frequent replacements.

1.1.5 Design Problem

Based on the existing coffee bean storage method *(see Table 1-1)* used by the Farm to Cup Philippines Association, these methods could not effectively address the issues arising from environmental conditions' impact on storing the coffee beans. This leads to moisture absorption that causes mold growth, potentially leading to profitability losses for farmers.

1.1.5.1 Specific Problem

The following stated below is the list of specific problems gathered from the general design problem of the project that the developers opt to address based on the client's current scenario:

- Moisture absorption from low-temperature conditions causes mold growth, impacting coffee bean quality;
- External odors, inadequate airtight seals, and light exposure can contribute to mold growth on coffee beans by allowing moisture to accumulate, creating favorable conditions for mold spores to develop, and
- The client typically relies on manual monitoring, which includes a moisture tester to determine the current moisture content, but no device to determine the current temperature of their storage area.

1.2 Project Objectives

The primary objective of the dotBean project is to develop an automated temperature-controlled storage system for moisture content monitoring for arabica green coffee beans, assuring that it can still carry the usual capacity that the client stores while being carried or dragged at ease.

The developers, to achieve this objective, aim to:

- Determine the correct temperature and humidity levels by utilizing a temperature and humidity sensor to closely monitor the coffee beans' moisture content inside a closed storage device;
- Design a storage device that's capable of maintaining coffee beans' moisture content from 11% to 12.5%;
- Develop an automated fan control system based on a pre-defined threshold; and
- Develop a device that maintains an average power consumption below 1000W per hour.

1.3 The Client

To address the problem, the developers are collaborating with Mr. Eli Natividad and his associates from La Trinidad, Benguet. Mr. Eli is the program manager of the Farm to Cup Association, which supports small-scale farmers and coffee shop owners. In particular, the developers conducted the field testing on Sir Eli's farm-slash-coffee shop, "Farm to Cup Benguet," located on Pico Rd., La Trinidad, Benguet.

1.4 Scope and Limitations

The dotBean project addresses the issue of moisture absorption-related problems with green coffee beans arabica in La Trinidad, Benguet. Targeting key stakeholders who have manually operated existing storage processes are the Coffee Farmers and Coffee Shop Owners; dotbean, in collaboration with Farm to Cup Benguet, introduces an automated temperature-controlled storing system incorporating IoT application for monitoring the coffee beans' temperature and humidity inside and moisture content. Therefore, the scope of this project is as follows:

- Green coffee arabica bean is the specific type of coffee bean that will be used for actual testing;
- A small-scale farm (small-scale farm refers to having less than 5 hectares of farming land according to the Department of Agriculture) located in La Trinidad, Mountain Province, named Farm to Cup DIY Brew Bar, and their farmers and associates are the test-users for this project;
- Small-scale coffee farmers have existing traditional storing methods;
- Maintaining the ideal storage conditions within 20°C to 25.4°C & 50% RH to 70% RH; and
- Monitoring the moisture levels of the arabica, maintaining 11% to 12.5% of the coffee beans within the storing period.

Consequently, it is essential to note that this project needs to cover facilities and producers already integrating technology into their existing storage facilities. As mentioned, the problem arises from the client's farm, located in the province of Benguet. Therefore, the testing protocol and results may differ from other regions or locations. In line with this, the limitations of the dotBean project are as follows:

- Excludes coffee farmers and producers that have already adopted advanced technological integrations within their current storage infrastructures, including but not limited to those utilizing Warehouse Management Systems (WMS);
- The project location is constrained to La Trinidad, Benguet, Philippines, and its findings and solutions may not be directly applicable to coffee producers and consumers in other places;
- This limited scope of location may not fully explore the scalability and broader applicability of the proposed solution beyond this specific place, and
- The project is constrained by a limited budget of Php 50,000 for the system's production. This budget constraint may affect the extent of technological capabilities and the resources available for the project.

1.5 Project Development

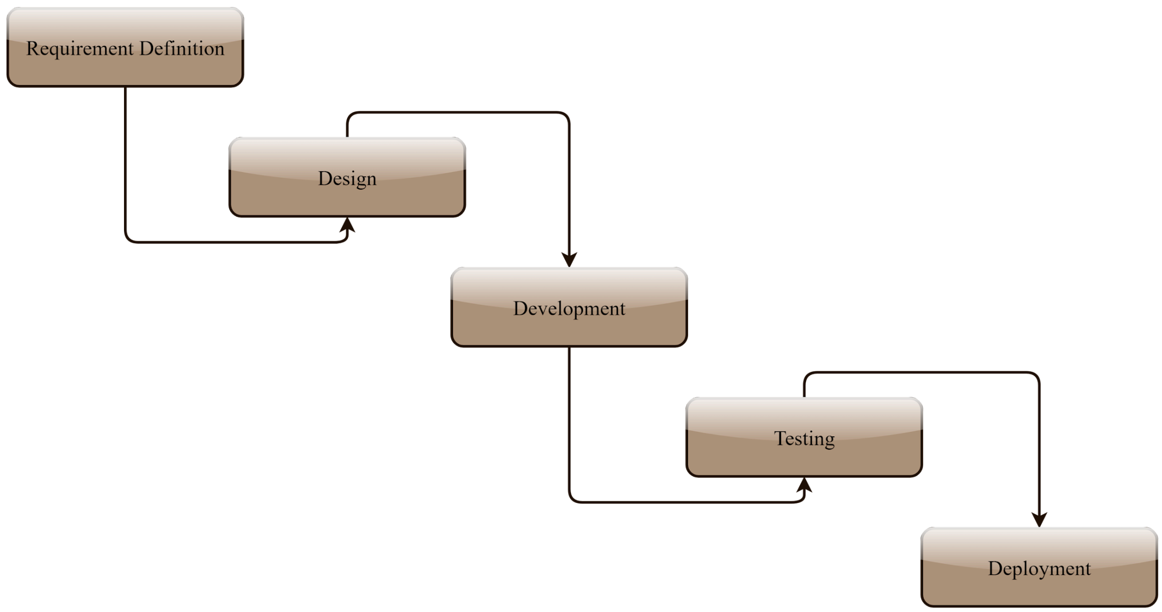


Figure 1-11. The dotBean's Project Development Waterfall Model

The developers chose a waterfall model to plan and develop the dotBean project, as depicted in Figure 1-11, from its topic selection to the system's deployment for the target clients and end-users. Below is the discussion of how the developers planned each step:

Requirement Definition: The first phase of project development involves defining requirements, including brainstorming for topic selection, identifying the problem and affected community, and validating the problem through surveys. Developers also conducted a technical issue and patent search for validation. Finally, the developers interviewed clients to identify needs, wants, objectives, and the project's scope and limitations.

Design: During this phase, the developers identified the components, software, and hardware required to develop the designs. This step also includes defining the general system's architecture, understanding how it functions based on its design process, and exploring different options to ensure compatibility with client requirements, design constraints, and design criteria for feasibility testing. To achieve this, hardware system development concluded through a simulation utilizing Solidworks to mimic the operation of each design option.

Development: In this phase, the developers choose the design that best meets the constraints and criteria and prepare it for prototyping and hardware development. The process begins with software development, including sensor calibration, parameter definition, and configuring control and monitoring systems. Hardware development follows, involving the construction of storage containers, sensor placement, actuator installation, and peripheral integration.

Testing: Once the device is constructed, the testing phase commences. During this phase, the developers introduced and demonstrated the prototype to the client during the field testing.

Deployment: After testing and necessary revisions are completed, and the system is deemed fit for deployment, the developers visit end-users and target clients for solution validation.

CHAPTER 2: DESIGN INPUTS

2.1 The Client Requirements

The initial client requirements for the proposed solution are:

- The device should accurately monitor the temperature and humidity inside the storing device and, at the same time, determine the dampness of the coffee beans to closely monitor their moisture content within a range of 11% to 12.5%, not reaching up to 14%,
- Ensures that the temperature is maintained within 20°C - 25.4°C and the humidity at 50% - 70%;
- The system should automatically command the fan to be turned on or off; and
- The device should maintain an average power consumption of below 1000W per hour.

2.2 Design Criteria and Design Constraints

Table 2-1. Design Criteria and Design Constraints

Design Criteria	Design Constraints
The dotBean should be able to adapt to the set temperature within 30 minutes when there is a variation.	Performance
The Raspberry Pi automatically commands the fan to turn on or off for less than 1 second.	Response Time
Produces low power consumption with no more than 1kW for 1 hour.	Power Consumption

2.2.1 Performance

Based on how it is perceived in this project, performance refers to a system that effectively achieves the desired temperature and humidity levels faster. It encompasses factors such as speed of reaching stability and consistency in maintaining the conditions.

In this project, the developers utilized a software application named SolidWorks, which can simulate the process it can produce in testing. The developers conducted a flow simulation for this software. This approach simulates the process to ensure the system meets the desired performance.

2.2.2 Response Time

Response Time is the time between detecting a temperature change, specifically when the temperature drops below the set threshold, and activating the fan.

The system's response time is critical in applications where timely action is required to maintain desired environmental conditions. In the context of this project, where the system should automatically turn on the fan when the temperature drops below 20°C, the response time directly affects how quickly the system reacts to changes in temperature.

According to AfterAcademy (2019), the response time can be calculated by:

$$T_{responseTime} = T_{fanActivated} - T_{readingReceived}$$

Where:

- $T_{responseTime}$ Is the Total Response Time in seconds.
- $T_{readingReceived}$ It is the time when the Raspberry Pi receives the temperature reading.
- $T_{fanActivated}$ It is the time when the fan is activated.

2.2.3 Power Consumption

Power consumption plays a role in determining how efficient a system can be. It is the energy a device consumes to operate (Harris & Harris, 2022). Developers are responsible for creating and implementing energy-efficient storage systems. This involves choosing the components that consume low power. This is crucial for reducing the system's overall power consumption, making it more efficient and environmentally friendly.

$$W = \sum_{i=1}^{n} V_i \times I_i$$	*(Equation 2)*

Where:

- W is the total wattage
- V*i* is the voltage across the *i*-th components
- I*i* is the current through the *i*-th components
- Σ is the sum of the wattage of each element from 1 to n.

The equation above assumes that each component has a unique identifier (i) and that the voltage (V*i*) and current (I*i*) for each component are known. The summation runs from (i=1) to (i=n), where n is the total number of components in the circuit (Elert, 2023).

Furthermore, the developers utilized the Meralco 2022 1H Average Rate for all Meralco residential customers in the refrigerators and freezers category to determine the benchmark, which had an average of 1.2382 kWh (Meralco, 2022).

After getting the total wattage of the device, calculate the electricity (kWh) it consumes. According to LearnMetrics (2022), the following formula can be used to calculate it:

$$P = W \times h$$	*(Equation 3)*

Where:

- P is the energy measured in watt-hours (Wh)
- W is the power in wattage
- h is the time in hours

The equation shows that the total energy consumed by a device is calculated directly. This equation can be used in appliances to calculate their energy efficiency battery life of devices and understand the electricity usage cost (Śmietańska, 2024).

Lastly, to compare the energy consumption with that of other devices and to calculate the cost of electricity usage, the power consumption is converted from watt-hours (Wh) to kilowatt-hours (kWh), as also mentioned by LearnMetrics (2022). The formula is as follows:

$$P_{kWh} = \frac{P_{Wh}}{1000}$$	*(Equation 4)*

Where:

- P_{Wh} is the power measured in watt-hour
- P_{kWh} is the power measured in kilowatt-hour

2.2.4 Public Health and Safety

Coffee beans are characterized as porous and hygroscopic. The porous nature of coffee beans refers to the presence of small holes in the cellular composition of the beans, which includes air pockets. This increases the risk of contamination, making it challenging to store coffee away from areas involved in hulling, grading, and cleaning operations to reduce exposure to contaminants like coffee husks, which can harbor Ochratoxin A (OTA), a mycotoxin linked to health risks. The hygroscopic properties, on the other hand, of coffee beans mean that they tend to absorb moisture from the surrounding environment (Kanniah, 2021). As a result, coffee beans are susceptible to moisture-related issues. Excess humidity can compromise the quality of the coffee beans. With this, addressing their porous and hygroscopic properties during storage is essential, ensuring proper management before roasting and consumption.

The choice of cover material for coffee beans is essential to prevent these issues. The likelihood of contamination increases when coffee beans are stored without proper insulation, airtight containers, and moisture control. For example, high moisture content can lead to mold growth, and consuming coffee contaminated with mold can lead to acute health issues.

In addition, using a food-safe cover material that can absorb moisture and insulate the temperature within the surroundings helps maintain the quality of the beans during storage. Proper storage conditions, including using moisture-absorbing covers, play a significant role in preserving the moisture content of the coffee beans (Moulana, 2024) while ensuring they remain safe for consumption. Furthermore, periodic maintenance and cleaning of the storage containers before and after use are necessary to ensure cleanliness. Therefore, choosing an easy-to-clean cover material is another effective way to maintain proper hygiene and prolong the quality of the green coffee beans.

2.2.5 Social Welfare

The development of technology to monitor and maintain the moisture content of green coffee beans may offer valuable benefits to farmers, particularly in communities where temperatures are low and maintaining the quality during storage is challenging. This project aims to address the issue of moisture absorption at low temperatures, which can lead to flavor degradation and market losses. Farmers can ensure their product's quality by providing the moisture content remains between 11% - 12.5%.

Furthermore, the project was developed with attention to its practical implementation in developing the project. The monitoring system should be user-friendly, ensuring that even those with limited technological knowledge can easily use it. By focusing on these factors, the system benefits from maintaining the moisture content and monitoring purposes for farming communities.

2.2.6 Economics

When developing a system, choosing suitable materials is crucial for cost-effectiveness. The developers should consider the availability and resources, especially in regions like La Trinidad Benguet, where options may be limited. Developers must balance the need for cost-effectiveness with the potential logistical difficulties of sourcing materials in such areas. This requires careful planning and adaptability to ensure the design remains practical for the target customers. Addressing the challenge of incorporating locally sourced materials is a strategic consideration that can influence the project's success in achieving the desired results within budget constraints.

2.2.7 Social

When developing a system, considering whom it will be intended for is crucial because it directly affects its functionality and effectiveness. Understanding the specific needs and challenges of the target market ensures that the system addresses the real problem they face.

In this case, coffee farmers need to maintain the moisture content of the coffee beans during storage. Hence, the system must effectively monitor the moisture levels to prevent quality degradation. With this, small-scale farmers have traditional storage processes in regions where temperatures are the key customers for this project. Considering the farmers' particular type and location, the systems may be designed specifically to their requirements.

2.2.8 Global

Several practical considerations should be considered to make the dotBean project globally viable while still being feasible to develop within the time and budget constraints. First, the device should comply with essential safety standards, such as electrical safety, to ensure the device is safe for use. Selecting modules (Wi-FI et al. module) that are already pre-certified can simplify the process of meeting FCC requirements. Developing a simplified mobile app with essential features, such as real-time monitoring and alerts, is vital. Initially, the app should be developed in English to ensure ease of use and accessibility for a broad audience.

Furthermore, power compatibility can be addressed by using power adapters that support a range of voltages (100V—240V) for essential international compatibility and selecting energy-efficient components to reduce power consumption. Furthermore, for cost-effective design, using off-the-shelf components and open-source software will help reduce costs while focusing on the core functionality of moisture monitoring and maintenance.

2.2.9 Cultural

In La Trinidad, Benguet, coffee farming is a significant part of the community’s identity, blending tradition and innovation. Understanding that farmers have established methods and practices is crucial; introducing new technology should be done to enhance, rather than disrupt, these practices. The device should be simple, providing clear benefits without significantly changing how coffee farmers work. Rather than changing the coffee farming and processing procedure, the developers should solely apply the innovation in the specific area the project addresses. The developers should avoid interfering with procedures before and after the targeted coffee bean processing stage.

Consulting with local farmers—in this case, dotBean's clients—helps to ensure the device is practical and relevant to their needs. Furthermore, the device should also be designed with the local environment and resources in mind. Using materials and components that are accessible and easy to maintain in the region can make the technology more efficient and sustainable for long-term use.

2.2.10 Environmental

In developing a prototype to monitor the moisture content of the coffee beans, several key considerations can help minimize the environmental impact. First, selecting materials that are both food-safe and environmentally friendly is essential. Opting for recyclable or biodegradable materials, such as food-grade plastics or stainless steel, can reduce waste and environmental degradation. For example, acrylic plastic is typically made from polymers (acrylonitrile-based polymers and polyvinyl chloride (PVC), which undergoes toxicity reduction to minimize any potential harm (Gilani et al., 2023). Additionally, the migration of substances from plastic materials into food is a critical consideration, with studies showing that specific migrated substances from plastic materials like polyethylene, polypropylene, polycyclohexane-1,4-di-methylene terephthalate, and polycarbonate are considered safe (Shin, 2024).

On the other hand, stainless steel is known for its use in various industries as it offers excellent corrosion resistance and mechanical properties (Moon et al., 2020). When coated with antimicrobial peptides, stainless steel's antimicrobial properties further enhance its suitability for food-related applications by combating biofilm formation (Hage et al., 2021). Moreover, when exposed to simulated food environments, stainless steel's electrochemical behavior underscores its durability and suitability for food processing applications (Sandoval-Amador et al., 2018).

Furthermore, the prototype can use minimal power to enhance energy conservation by incorporating low energy consumption and automation that activates the heating mechanisms only when necessary. Minimizing electronic waste is another important consideration. The design should use modular components that can be easily replaced if they fail, avoiding the need to dispose of the entire device. Selecting standard and quickly recyclable components can also facilitate recycling, reducing the environmental footprint.

2.3 Relevant Information

2.3.1 Storyboard

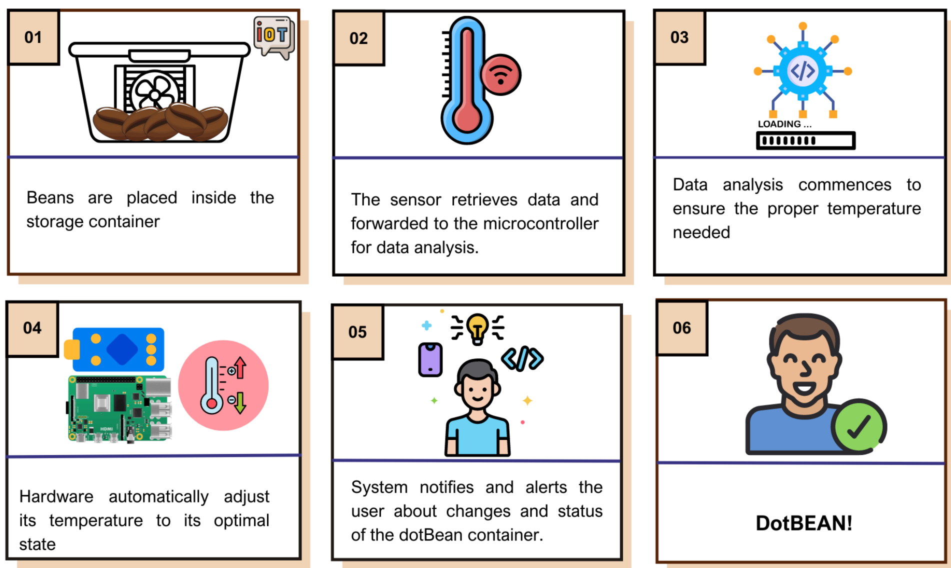


Figure 2-1. dotBean's StoryBoard

Shown above is the storyboard created to visualize the possible scenario that leads clients to the proposed system, as illustrated in Figure 2-1, which is presented to easily discuss how the dotBean operates and takes into effect in the current scenario of the client. Coffee beans are usually stored before reaching into consumers' cups, and during this stage, coffee beans are sensitive to environmental conditions. In the context of the dotBean project, after placing the beans inside, the system will be analyzed to measure the coffee beans' moisture content and the temperature conditions inside the storage. The system initializes these readings from sensors placed inside the container. When the coffee beans are not stored in the ideal storage conditions, the microcontroller will send commands to the actuators and notify the end-users about the status of the dotBean.

Compared to the current scenario of the dotBean's client, this ensures all-in-one monitoring capabilities, ensuring that the moisture content of the coffee beans is maintained, as well as other conditions that affect its quality and freshness. Unlike the traditional storing process where the clients have to store the coffee beans in multiple bags and pile them up inside their storage room, it ensures that farmers can save time from manual monitoring and limit the excess waste due to spoiled coffee beans.

2.3.2 Review of Related Literature

In the preceding pages, different studies and literature were provided, which discuss the various factors affecting the coffee beans' attributes, implementation of automation regarding controlling the temperature inside the storage and monitoring the coffee beans' moisture content, and other methods of measuring the moisture content of the coffee bean. Lastly, the purpose and efficacy of the hardware components in the subject matter were also presented.

2.3.2.1 Factors Affecting the Coffee Beans' Attributes

2.3.2.1.1 Temperature, Humidity, and Moisture

Temperature plays a pivotal role in determining coffee quality by affecting the release of free fatty acids (FFAs) through the hydrolysis of triglycerides. This process can impart an undesirable taste to the product. Studies show that storage with high temperatures can increase the speed of the growth of FFA in coffee beans. This rise is affected by the surrounding environment's temperature and humidity. (Błaszkiewicz, J., Nowakowska-Bogdan, E., Barabosz, K., et al., 2023). Controlling the humidity of the storage is essential in the storage of coffee beans; too much moisture can result in mold growth and make the taste musty, while the low humidity in the coffee bean storage affects the aroma and removes the excellent flavor of the coffee beans. The desirable humidity range for storing coffee is between 60% and 70% to prevent excessive dryness. Humidity exceeding 70% will speed up the build of molds on the beans. In comparison, levels below 50% can lead to the rapid evaporation of oils, resulting in stale-tasting coffee (Natalie, E., 2023).

According to the author Ahmed S. et al. (2021), temperature dramatically impacts the quality of coffee beans. As per Ahmed, S. coffee that grew at high temperatures has sometimes increased sensory attributes, while at other times, it decreased. For instance, warmer temperatures during seed development negatively affect the coffee's senses, which results in sour flavors. Additionally, there will be a change in coffee flavor quality due to the negative relationship of coffee with the warm temperature. Based on the study, if there is an increase in caffeine and phenolics above the limited threshold, it means there will be a buildup of bitterness due to the decrease in its sensory attributes. In contrast, increasing trigonelline improves the sensory qualities that result in good flavor and aroma.

On the other hand, the study entitled, From Plantation to Cup: Changes in Bioactive Compounds during Coffee Processing (Bastian et al. O.S., Dirpan A., Nainu F., Harapan H., Emran T.B., & Simal-Gandara J., 2021), revealed that moisture, among other factors, can significantly influence the quality of coffee beans. During maturation, the beans absorb moisture from the environment, affecting their chemical composition and flavor profile.

Coffee beans are known as hygroscopic, meaning they absorb environmental moisture. The presence of moisture can lead to the growth of mold and bacteria, which can affect the taste and aroma of the coffee. Additionally, moisture can cause the beans to become stale, which reduces their quality (Liberty et al. Company, n.d.). Coffee beans usually contain around 45-55% moisture when harvested. The next step is to air dry the beans, which can take a week or two. This process reduces the moisture content to about 10-12%, making the beans stable enough to transport but retaining enough moisture to provide excellent flavor when roasted (Brewing et al.).

Therefore, high temperatures accelerate staleness and may alter flavor, while low temperatures impact overall quality. High humidity, on the contrary, leads to mold and flavor issues, and low humidity causes beans to dry out. Lastly, there is fluctuating moisture content. Initially, 45-55% at harvest requires careful management to prevent quality issues as beans absorb moisture from the environment.

2.3.2.1.2 Light Exposure

Exposure of coffee beans to too much light can significantly affect how good the coffee tastes. Studies have found that when coffee beans get too much light, their flavor and smell can change due to the chemical reaction in the beans when light sets off (Lapoint-Tufts, 2021).

Light exposure significantly affects coffee quality, such as flavor, aroma, and freshness. Exposing the coffee beans in direct sunlight will oxidize the oil in the coffee beans, resulting in bad-quality flavor in the brewed beverage. Good packaging is essential to mitigate these issues. High-quality coffee uses UV-resistant bags to shield the beans from light. This bag is stored in a relaxed, dark environment to avoid the effect of light exposure. The impact of light exposure varies with roast levels, with lighter roasts potentially more susceptible to flavor degradation than their darker counterparts. In summary, light exposure harms the integrity of the coffee quality. To avoid this impact, careful and secure packaging and storage locations can preserve the freshness and quality of the beans.

In a study written by Schwabe, A. L., Hansen, C. J., Hyslop, R. M. & McGlaughlin, M. E. (2021) entitled “Climate Change and Coffee Quality: Systematic Review on the Effects of Environmental and Management Variation on Secondary Metabolites and Sensory Attributes of Coffea arabica and Coffea canephora”, it looks at how light affects the taste of coffee using 12 different measures. Out of the 12 studies reviewed, eight found that more light makes coffee taste less promising, while four said it improved the flavor. Interestingly, the study also found that less shade, below 45%, made coffee taste bitter and not so lovely while increasing shade from 37% to 61% improved the coffee's body. So, the amount of light and shade can change how coffee tastes.

2.3.2.1.3 External Odor

External odors can negatively impact coffee beans by being absorbed, altering their flavor profile. Beans are porous and can easily absorb surrounding smells during storage, transportation, and brewing, leading to off-notes in the coffee. Due to their hygroscopic nature, coffee beans can absorb moisture from the air, leading to oxidation and mold (Perfect et al., 2020).

When green beans absorb vast amounts of moisture, oxygen-bleachedness happens, which means the oxidation in coffee beans causes the physical look to appear lighter and affects the flavor. The coffee beans can absorb moisture from the air due to their hygroscopic nature, leading to oxidation and mold (Błaszkiewicz et al., 2023).

According to a study by Barea-Ramos, J.D., Cascos, G., Mesias, M., Lozano, J., & Martin-Vertedor, D. (2022), the volatile organic compound in the coffee beans may cause changes in their concentration and sensory strength due to the external odors that can impact them. These changes can result in a change in the coffee's aroma and flavor.

In conclusion, coffee beans' susceptibility to absorb external odors can significantly impact the delicate balance of volatile organic compounds, ultimately influencing the aroma and flavor characteristics of the brewed coffee.

2.3.2.1.4 Storage

Based on the study “Storage fungi and ochratoxin A associated with arabica coffee bean in postharvest processes in Northern Thailand" (Khewkhom et al., W.L., 2021) shows that the arabica coffee produced in Chiang Rai and Chiang Mai provinces of northern Thailand has fungal contaminations and the presence of mycotoxin ochratoxin A (OTA) was detected. This study also shows that the common fungal contamination in the Arabica coffee is Aspergillus and Penicillium. The choice of packaging material is vital to diminish fungal contamination. The use of polypropylene woven bags to store the coffee beans was proven to lower the Aspergillus contamination compared to other packaging. The findings emphasize that fungal contamination and mycotoxins like OTA in coffee products can be minimized by giving importance to postharvest practices, storage packaging, and proper drying methods.

Additionally, the study written by Knighton, M., Letter, B., McCarrick, A., Nea, S.J., & O’Dwyer, S. (2019), entitled “Carbon Monoxide Release From Whole Bean Roasted Coffee in Storage,” focused on estimating carbon monoxide (CO) emission factors from roasted coffee, particularly in non-ventilated storage spaces within the coffee industry. It was concluded that beans are typically sealed in vented bags with one-way valves that release these glasses, preventing oxygen from entering and degrading the beans. Storing roasted coffee in unventilated spaces can lead to dangerous CO concentrations, particularly in shipboard storage, raising concerns for Navy personnel safety. Some bags lost their vacuum seal, suggesting coffee delivered after packaging shortly may off-gas CO more rapidly.

Furthermore, Adam, J., Lantz, I., Mühlemann, S., Smrke, S., & Yeretzian, C. (2022) this study is about four different methods of storing whole coffee beans when once opened, namely an airtight canister. Gas chromatography-mass spectrometry assesses the storage stability to maintain aroma and freshness. The 2-methylfuran index was identified as the most suitable for evaluating coffee staling. The study found that the screw cap packaging method outperformed the others. The beans will lose their freshness faster when using a clip or closing the storage with tape. The research shows that freshness can be affected when carbon dioxide and oxygen are inside the packaging. This research indicates that consumer practices can influence the lifespan of coffee beans., complementing existing knowledge on changes during the primary shelf-life.

2.3.2.2 Implementation of Automation

2.3.2.2.1 Storage Conditions

Wati, E., Hidayanti, F., and Prasetya, A. developed a post-harvest Arabica coffee bean storage system (2020). This system ensures that the storage maintains the humidity between 60% and 70% and that the storage area remains within the range of 19°C to 27°C, as per Badan Standardisasi Nasional guidelines. The actuator in this system uses a fan, and to measure the room temperature and humidity, a DHT 11 sensor is needed, which an Arduino-Uno controls. The study was done with two experiments, one in a non-conditioned storage room and one in a non-conditioned storage room, where two bags of Toraja Arabica green coffee that contained 500 grams were stored in these rooms. Then, the Arduino-Uno controls the temperature of the two DHT 11 sensors placed on the opposite sides of the storage box walls. The results revealed that the controlled environment storage successfully maintained coffee temperature between 19°C and 27°C and humidity at 60% to 70% for approximately 20 hours, accomplishing this within less than 30 hours.

In addition, a study entitled “Comparison of chemical compounds and their influence on the taste of coffee depending on green beans storage conditions” assessed the impact of storage conditions on green coffee beans' chemical composition and taste quality to ensure ongoing excellence. Over three months, natural and washed coffees initially had similar attribute levels, with aroma and acidity prominent and aftertaste less so. Washed coffees generally saw a 5-7% decrease in quality, while natural coffees experienced smaller drops of 4-5%. After 12 months, natural coffees maintained higher quality, with scores just below 80 points. Washed coffees stored at 18°C and 20 °C in both jute and GrainPro bags recorded significantly lower scores, losing around 12% of their initial values, especially in aroma and acidity. The ten °C storage consistently yielded the highest quality, particularly for natural coffees in jute bags, which retained their quality impressively.

2.3.2.2.2 Measuring Moisture Content

In a study conducted by Zambrano, M.V., Dutta, B., Mercer, D.G., MacLean, H.L. and, Touchie, M.F. (2019), the methods of measuring moisture content (MC) in dried food products were categorized into two groups: direct and indirect.

Under the category of direct methods, the researchers evaluated Gravimetric and Chemical methods. The Gravimetric method involves weighing the sample before and after drying to determine the weight loss due to moisture. The Chemical methods include Karl Fischer titration and Coulometric methods, which calculate the amount of water in the sample based on its interaction with chemical reagents. On the other hand, the researchers evaluated several techniques in the indirect method, including Optical, Dielectric, Nuclear, and Hygrometric methods. Optical methods, such as infrared spectroscopy, infrared thermography, and hyperspectral imaging, analyze the reflectance or emissivity of the sample to determine its moisture content.

The dielectric method was used to measure the electrical properties such as electrical conductivity, microwaves, and radio frequency to determine the moisture content. The most common methods used for assessing the moisture content and monitoring the variation in capacitance on a material's dielectric properties are capacitance and resistive methods. These methods are widely used in moisture meters through a contact method; a small electrical current is passed through the sample, and the resulting resistance is correlated with the moisture content.

Measurement methods include the Nuclear method, which measures the hydrogen content and can help estimate moisture content, and the Hygrometric method, which measures humidity to determine moisture content.

Factors such as the type of sample, specific measurement requirements, and available equipment can affect the strength of this method.

Determining the final coffee product's quality and flavor is based on measuring the coffee beans' moisture content. Moisture content affects characteristics, from acidity and sweetness to aroma and mouthfeel. One of the most traditional methods of measuring the moisture content of green coffee is the oven-drying method. This process involves sample green beans baked at 105°C (220°F) in a convection oven for 24 hours, during which weight loss is recorded. Jane Merchant (2021).

However, drying coffee beans using sunlight takes longer and is less efficient (Putra, M.M.A & Herlina, A., 2020). Their study "Design of Measurement of Coffee Seed Water Content Using Load Cell Sensor On Coffee Dryer" suggested using a Load Cell sensor with a Hybrid Collector and LPG for the coffee dryer. This helps measure the weight loss of coffee beans while drying and determines the moisture content while ensuring quality beans. With the Load Cell sensor, we can accurately measure the weight loss of coffee beans while closely monitoring the drying process. This ensures the coffee beans reach the correct moisture level for quality standards.

Furthermore, most modern moisture testers can quickly measure the moisture content within seconds. It can be backed up to the computer, allowing the tracking of results without needing thermometers or charts. Another standard method is the Karl-Fischer Titration (KFT) method, considered one of the most accurate ways to measure moisture content. However, since KFT testing destroys the sample and requires significant cost, expertise, and time, it is not a very popular method. The standard across the industry is to use moisture meters that approximate the amount of moisture present in green coffee. Most of them are known as capacitance devices. Capacitance devices measure those beans' dielectric constant to calculate the water inside (Sucafina Specialty, 2023). For instance, it is like using a special magnet that can stick to a solid. This magnet can tell if the material is wet or dry. If the material is moist, the magnet can stick to it because water can attract it. If the material is dry, the magnet cannot stick because no water attracts it (Gupta, D., 2018) (*see Figure 2-3*).

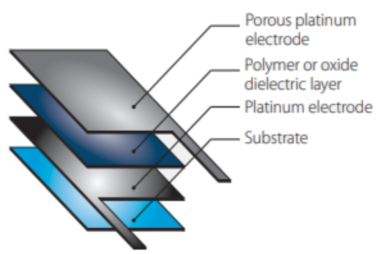


Figure 2-2. Capacitor Sensor

In a study by D A. Firmansyah et al. (2020), the researchers measured the capacitive signal with the help of a capacitive sensor made from copper plates. Then, the comparator and oscillator were turned to a voltage signal. Finally, to get the moisture content, the voltage signal was converted using a microcontroller that will be displayed on the LCD screen. The prototype's accuracy results for the coffee with water and evaporated coffee show a percentage error of 0.08% and 0.22%. This result was measured by comparing the prototype and moisture content from oven heating.

Furthermore, a system introduced by Alibayan, Bobadilla, Carnicer, M., et al. (2019) from the Technological University of the Philippines has a dual functionality for measuring moisture levels and incorporating the quality correction feature. This custom-designed software has advanced components like electrical plates, a microcontroller, and an integrated dryer to evaluate the beans' moisture based on the criteria needed. This system achieved a high accuracy rating in measuring the beans' moisture content, with a 91.67% achieving rate.

Another method is the resistive method; however, this device is not commonly used due to the resistive method having direct contact with the material. This can get dirty and rusty when in contact, making it unsuitable for coffee bean application. Despite this, a resistive sensor is sensitive and stable, providing accurate measurements of 30 to 90% moisture content (Yu et al., 2021). A resistive sensor is like using a special light that can see through solids. This light can tell if the material is wet or dry. If the material is moist, the light can see through it because water is like a clear glass. If the material is dry, the light cannot see through it because there is no water to let it through. The resistive method uses two probes to measure the resistance. When there is more water, it can conduct more electricity, which means there is less resistance. Therefore, the sensor can tell that the material is wet. Nevertheless, when there is less water, the material cannot conduct as much electricity, which means more resistance. Ergo, the sensor can tell the material is dry (Gupta, D., 2018) (*see Figure 2-3*).

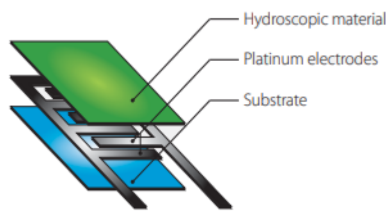


Figure 2-3. Resistive Sensor

The resistive moisture sensor offers a quick and efficient method for measuring moisture content in food samples. The humidity sensor that detects moisture by measuring changes in electrical resistance is called a humidity sensor. Additionally, it is reliable in measuring moisture because the food sample's density does not influence it. In a study by Yin, Jianhui, Cheng, Shengli, Wu, & Guojian (2015) titled “Resistive Grain Moisture Measurement System,” the system provides continuous real-time grain moisture and temperature measurements. It uploads data and generates alarm information for users. The system can measure moisture from 11% to 40% and temperature from -40°C to 180°C. The system converts resistance signals to frequency signals and uses frequency and period measurements to determine grain moisture. It uses an oscillating circuit and a single-chip microcomputer with optimized software and algorithms to measure different food grains swiftly and accurately.

2.3.2.3 Components Selection

2.3.2.3.1 Raspberry Pi 4B

Raspberry Pi has been successfully used in various projects, and many resources are available to guide developers through the process. According to (Murthy & AjaySaiKiran, 2018) (Yongsheng, Xie., Hong, Ding., Fendong, and Huang, 2019), Raspberry Pi functions as a single-board computer that collects data from sensors such as DHT22, and the collected data is transmitted to cloud platforms for further analysis and processing. Since the Raspberry Pi can be programmed to send real-time notifications and commands based on the measured values, the system can monitor the temperature and humidity in real-time. With this efficient control, preventing potential safety hazards is possible. Lastly, the Raspberry Pi is known for its low cost and versatility; it can be connected to other devices and networks. The microprocessor's versatility is ideal for embedded systems and the Internet of Things.

2.3.2.3.2 Acrylic Plastic

The unique benefits of acrylic containers, including their shatterproof nature ensuring durability, make them the optimal choice for storing coffee beans. The transparency of acrylic allows for easy monitoring of coffee bean quantity and condition without opening the container, preserving freshness. The lightweight and stackable design makes these containers convenient for handling, stacking, and optimizing space in storage rooms. Cost-effective yet durable, acrylic offers a practical solution for various storage needs. Their versatility and interchangeability cater to individual preferences, allowing customization for different storage requirements. When designated as food-safe, acrylic containers meet FDA standards, providing a secure option for storing coffee beans. However, it is essential to note that acrylic containers should avoid exposure to high heat or microwaving (Goodmann, 2022) (Walker, 2023).

2.3.2.3.3 Desiccant Gel

According to Johnson (2023), good-grade silica gel is known for its safety and moisture-absorbing properties. It is instrumental in preserving food items such as coffee beans. Moreover, silica gel desiccants effectively remove excess moisture without altering the food's composition (FDA, 2021.). Unregulated in many countries, silica gel's inert and non-toxic nature ensures it will not permeate or change the food it protects. Beyond coffee beans, it finds applications in spice drawers and packaging for items like seaweed, dried fruit, and jerky. It is also beneficial for storing potatoes, garlic, and onions, slowing the sprouting process (WebstaurantStore, n.d.). The versatility and reliability of food-grade silica gel make it a valuable solution for maintaining the freshness and quality of various stored food items.

2.3.2.3.4 Polyurethane Foam and Polyolefin Foam

Polyurethane foam, particularly PU rigid foam plastic, is a precious material for storing food like coffee beans. Its application in cold storage, refrigerated trucks, and commercial refrigerators as a thermal insulation material contributes to energy conservation and reduces carbon dioxide emissions (*PU et al. Help Keep Goods Fresh and Save Energy*, n.d.). In food processing and storage, polyurethane ensures effective temperature control, preserving food at optimal conditions both indoors and outdoors (Technology, n.d.). Beyond insulation, polyurethane's resistance to mold, ease of cleaning, and reduced energy consumption make it essential for ensuring food safety and sustainability in the industry. Innovations like

On the other hand, polyolefin foam is widely used for various applications due to its properties, which are suitable for thermal insulation and moisture management. Low-density polyethylene (LDPE) foams are particularly well suited for cold insulation, as they balance cost, insulation value, and ease of installation. Polyolefin foams are highly effective in managing moisture because they strongly resist water vapor. This quality implies that they are essential for preventing moisture accumulation in insulation used for buildings and equipment. The closed-cell structure of the material guarantees long-lasting durability by limiting water absorption (Foam Factory, Inc. & Author Foam Factory, Inc., 2024). This feature allows the insulation to remain effective even in environments with high humidity or moisture exposure.

2.3.2.3.5 Plastic Agitator

In the context of processing and combining food, agitation refers to the movement of ingredients that stimulate different effects depending on the particular procedure. According to Flow

(2023), it is vital to investigate the features and specifications of the equipment before getting too deep into the details of food processing and mixing. Different components are used in agitation operations; these components can be classified as liquid-liquid, solid-liquid, or gas-liquid. Homogenization techniques are helpful in the food manufacturing industry as they merge disparate ingredients, such as liquid milk and solid chocolate or cocoa. Specialized agitators were used to make sure that these materials blended well, and temperature monitoring was needed to maintain the consistency of the product (Dey, 2023).

2.3.2.3.6 Thermoelectric Peltier Module

The Peltier Module is employed to uphold a steady temperature within the container. This can help maintain the temperature to preserve the coffee bean flavor and quality. Lastly, the ability of the Peltier module to maintain a consistent temperature and humidity level helps extend the storage life of coffee beans, resulting in the beans remaining fresh for a longer period of time.

2.3.2.3.7 Radiator Fan

The radiator fan circulates the air inside the storage. This helps to cool down the beans and maintain an optimal temperature range. Adding a radiator fan is essential and helpful to maintain the coffee bean flavor; it maintains a consistent temperature, preventing the beans from becoming too hot.

CHAPTER 3: PROJECT DESIGN

This chapter presents the general system architecture of the dotBean project, the design process flow based on the system architecture, its use case, data flow, and entity relation diagrams, and the three design options developed for the proposed solution. Design options are carefully discussed and analyzed, including their sub-parts: system architecture, design process, software process and design, and hardware components, to support and back up each quality and proposition of design options to the proposed solution. This chapter also includes the testing and validation of design options based on the design criteria and constraints of the client's requirements, as shown below:

Table 3-1: Design Criteria and Design Constraints Based on Client Requirements

Client Requirements	Design Criteria	Design Constraints
Ensures the temperature is maintained within 20° C - 25.4° C and the humidity at 50% - 70%.	The dotBean should be able to adapt to the set temperature for 30 minutes when there is a variation.	Performance
The system should automatically command the fan to turn on or off.	The Raspberry Pi automatically commands the fan to turn on or off in less than 1 second.	Response Time
The device should swiftly manage power to stabilize temperature and humidity, balancing energy savings with operational needs.	Produces low power consumption with no more than 1kWh for 1 hour.	Power Consumption

With this, design options are carefully analyzed and examined, and their feasibility and functionality in supporting the proposed solution are assessed.

3.1 System Architecture

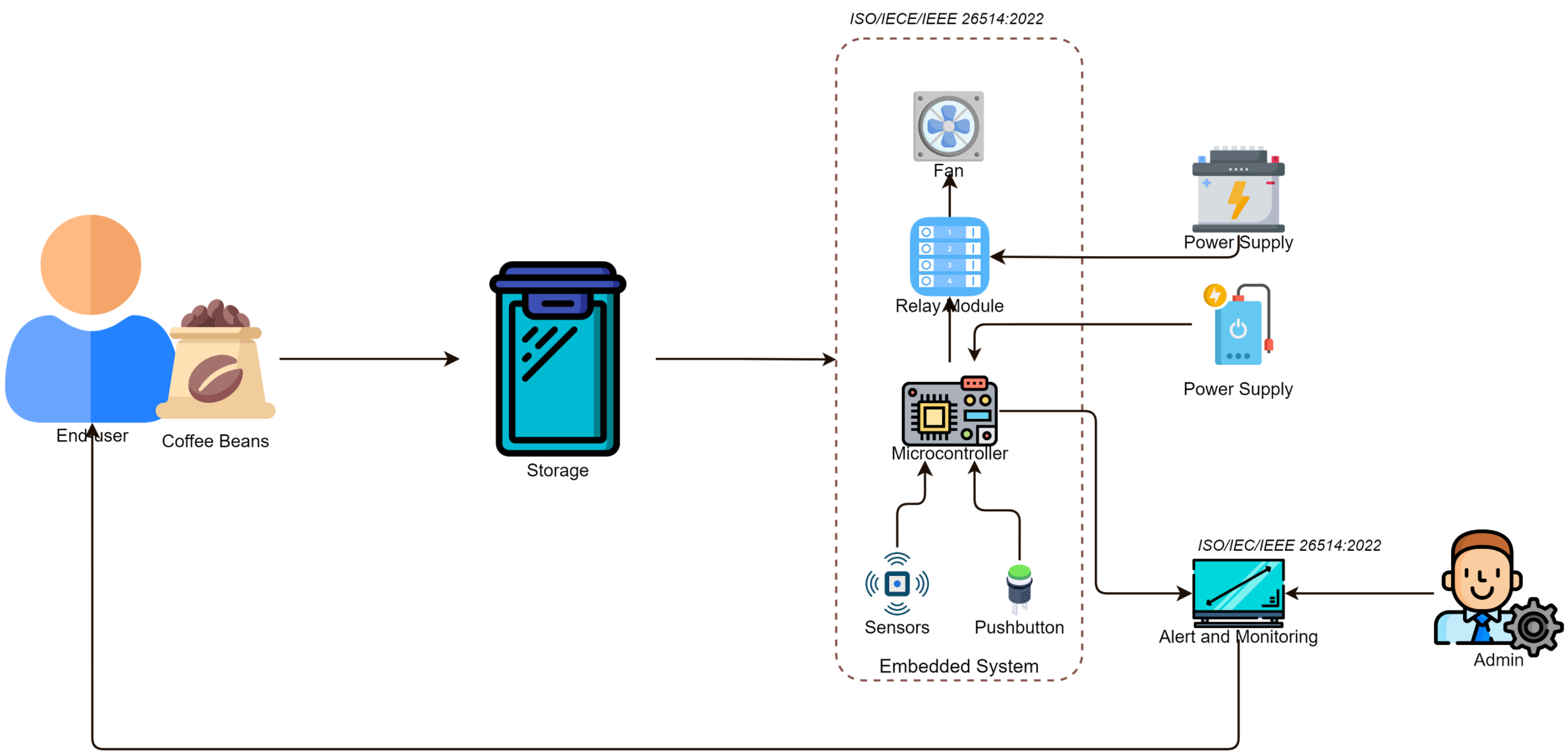


Figure 3-1. General System Architecture

System architecture is a blueprint for a system's structure, behavior, and organization. It defines the system's components, relationships, and the principles governing its design and evolution (Parker, J., 2023). The general system architecture of dotBean's project, as shown in Figure 3-1, involves monitoring and controlling the storage conditions for coffee beans. Temperature, humidity, and moisture sensors are placed within the storage container to monitor these conditions continuously. The collected data is then sent to a microcontroller for initializing. The microcontroller analyzes the sensor readings (data) to determine whether the current conditions deviate from the predefined parameters. If the data falls outside the set parameters for any monitored conditions, the microcontroller triggers an actuator to regulate the air within the container, maintaining optimal storage conditions.

To provide users access and visibility into the state of the coffee bean storage, it offers monitoring accessibility. As the dotBean operates automatically, the monitoring includes showing the changes in temperature, humidity, and moisture content conditions, as well as the status of actuators. The dotBean system manages and controls all the usability and functionality to maintain the desired environmental conditions inside the container and keep the moisture content of the beans close to 11% - 12.5%.

3.2 Design Process Flow

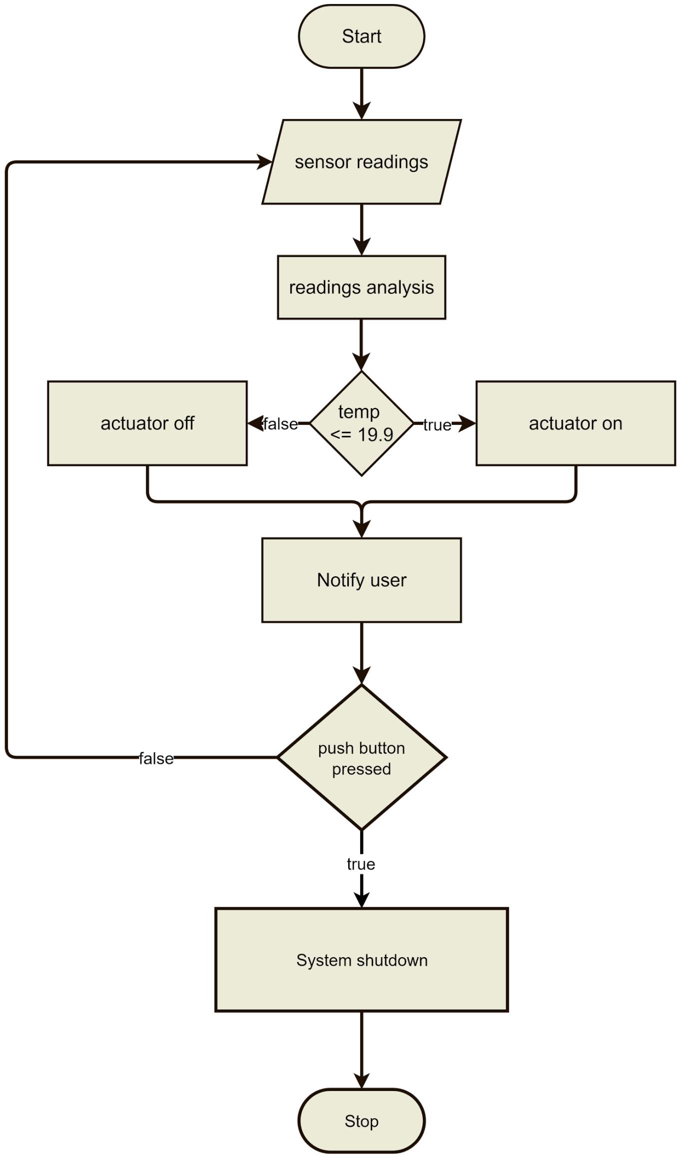


Figure 3-2. General Design Process

Design process flow is a diagrammatic representation of a sequence of activities involved in a process. It provides a visual representation that helps understand a process's steps, decisions, and dependencies. While architecture provides a high-level view of the system's structure and components, the process flow describes the specific sequences and interactions for executing a particular task.

At its core, Figure 3-2 illustrates the general design process flow of the dotBean in a simple diagram. The process begins with data from the sensor, and the microcontroller analyzes the received data. A predefined threshold was set and used as the basis for actuators to run automatically. Consequently, all controls and processing were done using a microcontroller, specifically the Raspberry Pi 4B. With this, the user can monitor and access all the changes in the storage conditions, the coffee beans' moisture content, and the status of the actuators.

3.3 Data Flow Diagram (DFD)

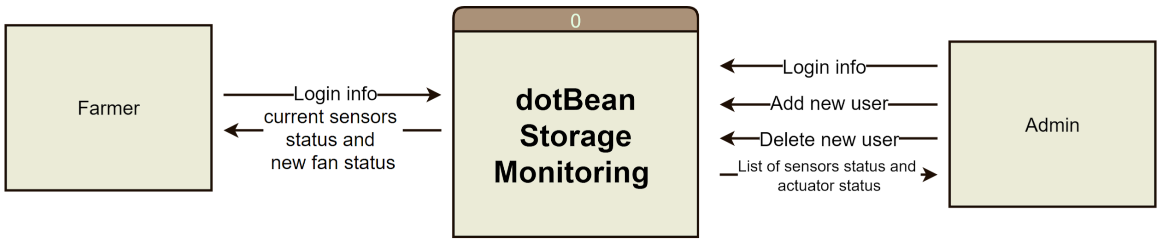


Figure 3-3. dotBean's DFD Level 0

A Data Flow Diagram (DFD) is a formal representation of a system's structure that facilitates comprehension by both programmers and users. This visual representation effectively depicts the flow of data within the system and the processes responsible for manipulating this data. DFDs play a crucial role in system design by offering a transparent visualization of the system's operations and the pathways through which data moves.

DotBean's DFD level 0 depicts the data received from the actors to the system and vice versa (see Figure 3-3), offering a general flow of how the data is exchanged between the external entities and the system. Below is the DFD level 1, which explains the processes inside the system more deeply.

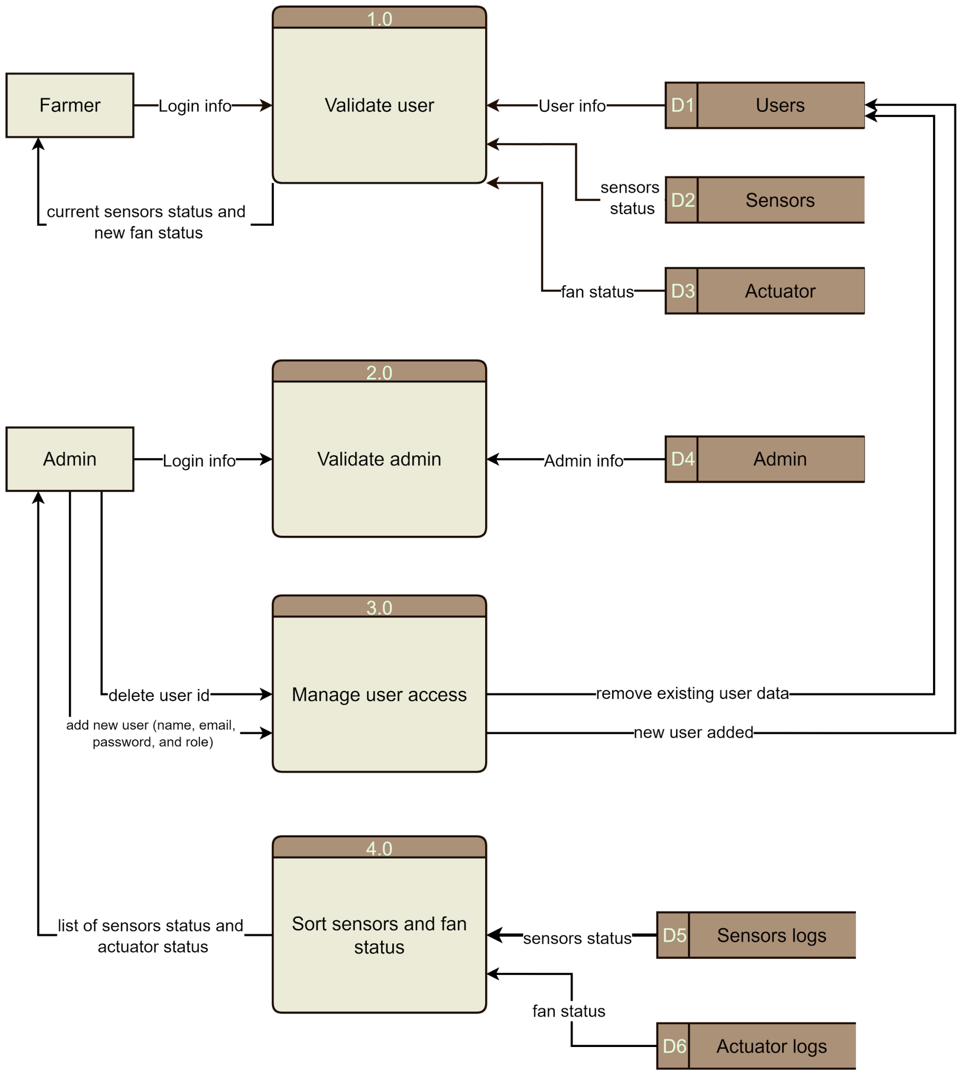


Figure 3-4. dotBean's DFD Level 1

The DFD Level 1, as presented in Figure 3-4, details how information flows between the Farmer and Admin users, the processes that handle their requests, and the associated data stores that keep critical system information.

The diagram begins with Process 1.0: Validate User, where the Farmer provides their login credentials, including their email and password, to access the system. The login information is processed through the "Validate User" function, which cross-references the provided credentials with those stored in Data Store D1: Users. If the credentials match, the system retrieves the user information and confirms the user's access to the platform. Once logged in, the Farmer can view the current status of sensors and fans. This data is fetched from Data Store D2: Sensors, which holds information on environmental metrics such as temperature, humidity, and moisture content, and Data Store D3: Actuator, which stores the status of the fan. If there are updates to the fan status, the new information is sent back to the Actuator data store to keep the records current.

Parallel to this, Process 2.0: Validate Admin shows the actions for an Admin logging into the system. Like the Farmer login, the Admin provides login credentials verified against Data Store D4: Admin. The admin can manage the system's user data and view relevant status information if the credentials are authenticated.

The Admin can use Process 3.0: Manage User Access to add or remove users upon successful authentication. This process involves two key actions: adding a new user or deleting an existing one. To add a new user, the Admin inputs details such as the user's name, email, password, and role. This information is then stored in Data Store D1: User, and the newly added user gains access to the application. Conversely, if the Admin decides to delete a user, they specify the user ID to remove, and the system removes the relevant information from the data store, effectively denying future access to that user.

The Admin can monitor environmental data and fan statuses through Process 4.0: Sort Sensors and Fan Status. In this process, the Admin can view the list of sensor statuses, which the system retrieves from Data Store D5: Sensors Logs. Additionally, the Admin can view information regarding the fan status, which is fetched from Data Store D6: Actuator Logs in Process 4.0. This gives the Admin a comprehensive view of the average temperature, humidity, calibrated moisture content, and fan status.

3.4 Entity Relationship Diagram (ERD)

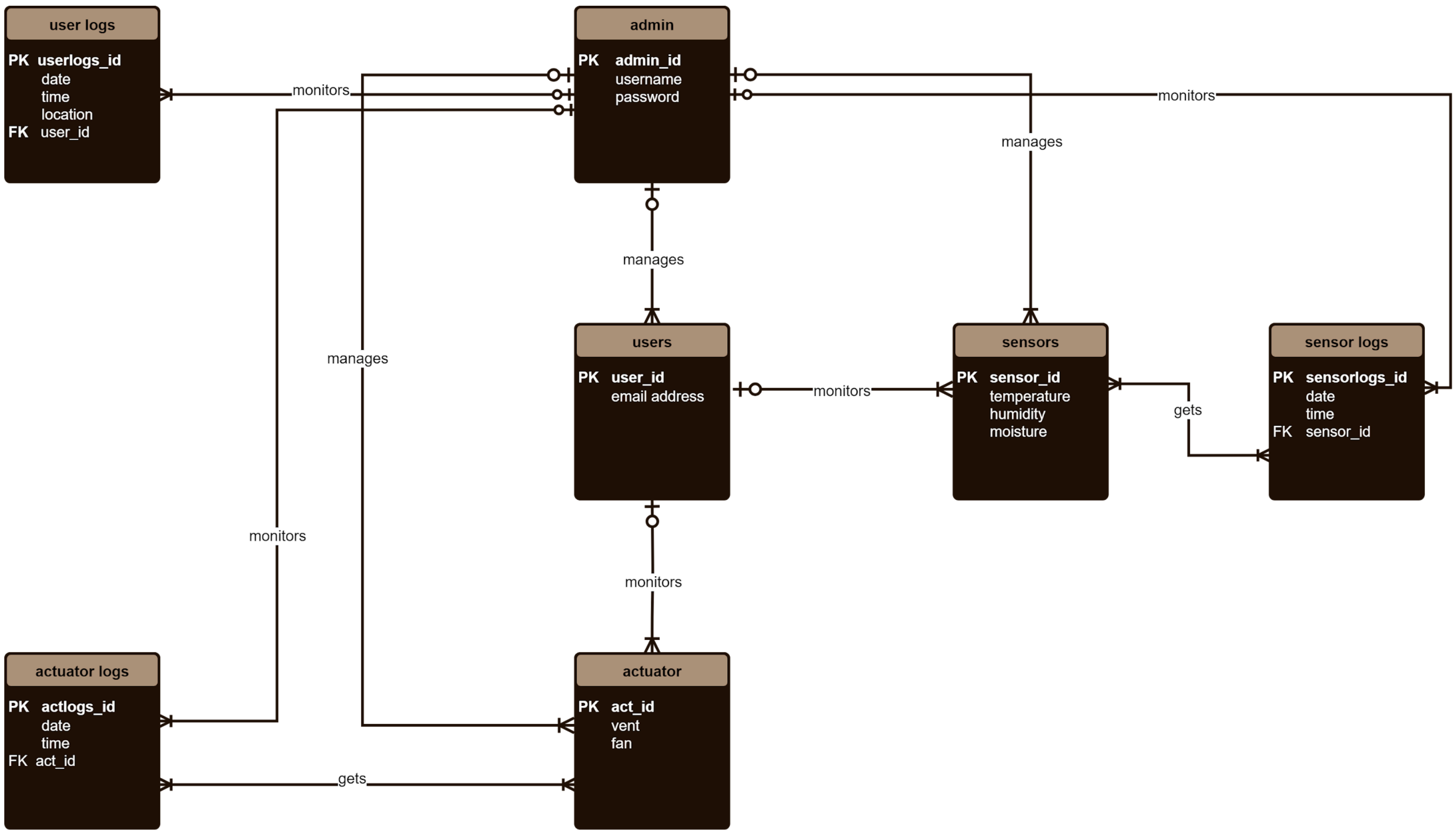


Figure 3-5. dotBean's Entity Relationship Diagram

Figure 3-5 illustrates the ERD of the dotBean project. The "user logs" database will serve as records of user interactions and activities within the system. Each user log entry is associated with userlogs_id; the userlogs_id serves as a primary key in the user entity, and a foreign key in the user logs entity links each log entry to the corresponding user.

The "admin" database functions as a centralized storehouse for data about administrative duties and settings within a system. The admin_id is the primary key in this database to ensure the administrator has a distinct identifier. This admin database demonstrates a one-to-many relationship with the "user logs" database: date, time, and location, the "sensors" database: temperature, humidity, and moisture, and the "actuator" database. The “users” database is a foundational component, storing essential information about users within the system. Each user is uniquely identified by a user_id in this database and their associated details. This user database establishes a one-to-many relationship with the “sensors” and “actuator” databases.

This relational model ensures a cohesive structure where user information is linked to various environmental readings and actuator statuses, facilitating a comprehensive understanding of user interactions within the system.

3.5 Moisture Sensor Calibration

To do this, the developers conducted 50 tests to identify the sensorValue from the sensor, which is the corresponding moisture content, when the beans were measured using the moisture tester.

Table 3-2. Agratronix Result and Capacitive Sensor Value

Agratronix Result	Capacitive Sensor Value
11.6	486
11.5	483
11.2	484
11.2	484
11	483
11.3	485

Agratronix Result	Capacitive Sensor Value
11.2	485
11	485
11.1	485
11.1	487
10.9	484
10.8	485
10.8	490
10.8	487
11	486
11	484
11	488
11.1	488
11	487
11	487
11.2	487
11.2	486
10.9	487
10.8	485
10.8	487
11.7	489
11.1	487
11	484
10.9	487
11	487
11.4	489
11.3	488
11.1	486

Agratronix Result	Capacitive Sensor Value
11.1	484
11.1	487
11.1	486
11.3	487
11.3	485
10.9	484
11	488
11.1	490
11.2	491
11.1	490
10.8	487
10.9	489
11.2	490
11.1	492
10.9	489
11	489
11.2	490

Once the sensors were stabilized at the sensorValue of 510 (for the capacitive) and 658 (for the resistive) when not in contact with the beans (calibratedMoistureContent = 0), using the 50 data from the conducted testing, find the slope (m) and intercept (b) of the sensor.

Slope refers to the rate at which the dependent variable (y) changes concerning the independent variable (x), representing the steepness/incline of the line. The intercept of the sensorValue (x) is the point on the y-axis where the line crosses when x equals 0, meaning it is the value of y when the sensorValue is multiplied by the slope and added to the intercept.

Once the slope and intercept were determined, the developers conducted another 50 tests of the sensors separately. Below figure is the formula used to determine the moisture content of the beans using the sensor utilizing the linear regression equation with an absolute value:

```
// Calibrate the sensor value to match the Agratronix measurements
  float calibratedMoistureContent = abs(slope * sensorValue +
intercept) //y = mx+b; linear regression code
```

Figure 3-6. Calibrated Moisture Sensor Formula Snippet Code

Table 3-3. Referenced Moisture Sensing Device vs. Capacitive Soil Moisture Sensor

Test #	Referenced Moisture Sensing Device	Capacitive Soil Moisture Sensor
1	10.9	10.81
2	10.9	11.3
3	11.4	11.3
4	11.1	11.3
5	11.1	11.79
6	11.3	10.81
7	11.4	10.81
8	11.9	11.79
9	13.2	12.77
10	13.18	12.77
11	11.7	11.79
12	11.7	11.3
13	11.8	11.79
14	12.3	12.28
15	12.7	12.77
16	12.8	11.79
17	13.4	13.27
18	12.5	11.79
19	12.1	11.79
20	12.2	11.79
21	11.7	11.79
22	11.7	12.28
23	11.6	11.3

Test #	Referenced Moisture Sensing Device	Capacitive Soil Moisture Sensor
24	11.5	12.28
25	11.5	11.79
26	11.5	11.79
27	11.8	12.28
28	11.8	11.3
29	11.7	11.79
30	11.7	11.79
31	11.6	11.79
32	11.1	10.81
33	11.4	11.79
34	11.6	11.3
35	11.6	11.3
36	11.6	11.3
37	11.6	11.3
38	11.4	11.3
39	11.3	11.79
40	11.5	11.3
41	11.3	11.3
42	11.4	11.79
43	11.5	11.3
44	11.2	11.3
45	11.1	11.3
46	11.3	11.79
47	11.5	11.79
48	11.6	11.3
49	11.6	11.79
50	11.6	11.3

Given the data above, their mean, variance, and standard deviation can be seen below:

```
refmoisture_device - Mean: 11.7 Variance: 0.31 Standard Deviation: 0.56
capmoisture_device - Mean: 11.66 Variance: 0.27 Standard Deviation: 0.52
```

Figure 3-7. AgraTronix Moisture Tester and Capacitive Moisture Sensor Mean, Variance, and Standard Deviation

The results for the moisture data from both devices provide insights into the consistency and variability of the measurements, as depicted in Figure 3-7. The mean moisture level recorded by the `refmoisture_device` is approximately 11.70, while the `capmoisture_device` records a slightly lower average of 11.66. This indicates that, on average, the moisture levels are very close between the two devices, though not precisely the same. Regarding variability, the variance for `refmoisture_device` is 0.3082, slightly higher than the 0.2677 variances of `capmoisture_device`. This suggests that the moisture readings from `refmoisture_device` are more dispersed around the mean than from `capmoisture_device`.

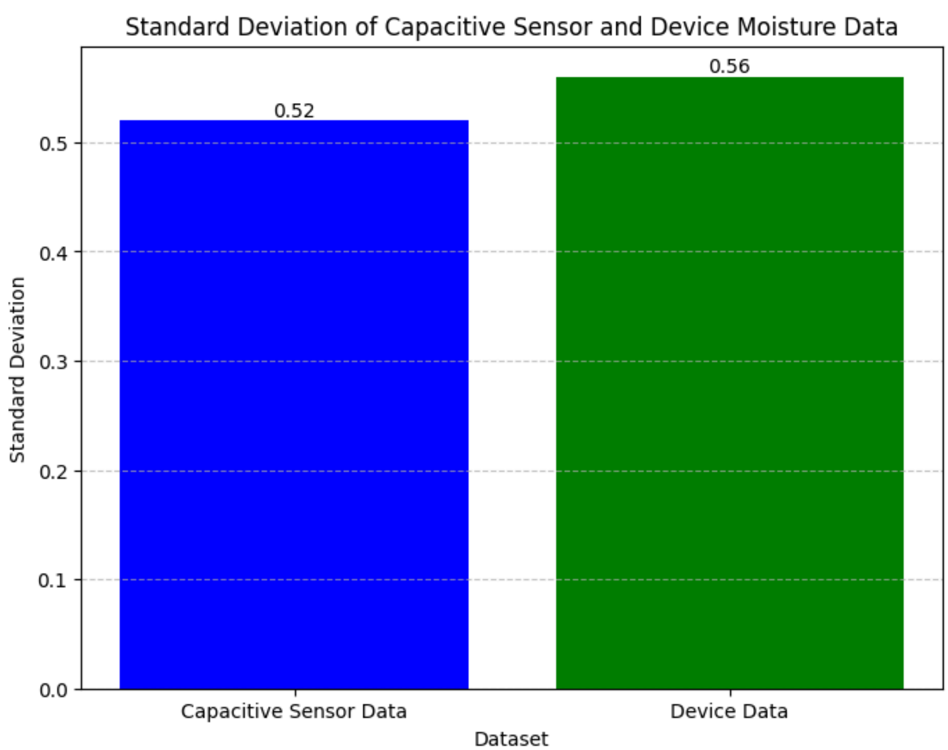


Figure 3-8. Standard Deviation of Capacitive Sensor and Device Moisture Data

The standard deviation further reinforces this, with `refmoisture_device` having a standard deviation of 0.5552 and `capmoisture_device` having 0.5174, as depicted in Figure 5-8. These values indicate that most moisture readings from both devices typically deviate from their respective means by about 0.55 and 0.52 units, respectively. While both devices provide fairly consistent readings, `refmoisture_device` shows slightly more variation in its measurements than `capmoisture_device`.

Table 3-4. Referenced Moisture Sensing Device vs. Resistive Soil Moisture Sensor

Test #	Referenced Moisture Sensing Device	Resistive Soil Moisture Sensor
1	11.2	10.78
2	11.1	10.8
3	11.5	11.3
4	11.4	11
5	11.7	11.2
6	11.6	11.2
7	10.9	11
8	11.7	11.4
9	11.3	11.9
10	12.5	12.3
11	12.3	12.14
12	11.1	11.3
13	11.3	11.5
14	11.2	11.6
15	11	11.5
16	10.9	11.3
17	10.7	11.5
18	11.3	11.1
19	11.5	11.4
20	11.6	11.27
21	11.5	11.9

Test #	Referenced Moisture Sensing Device	Resistive Soil Moisture Sensor
22	11.5	12.3
23	11.3	11.8
24	11.8	11.1
25	11.9	12.4
26	11.9	11.5
27	11.7	12.3
28	12.7	11.5
29	12.3	11.8
30	11.8	12
31	11.5	12
32	11.5	12.1
33	11.8	12.1
34	11.9	12.1
35	12.3	12.5
36	11.4	10.93
37	11.5	10.8
38	11.5	10.9
39	11.4	10.95
40	11.5	11
41	11.6	11
42	11.3	10.8
43	11.7	10.9
44	11.5	11
45	11.5	11
46	11.6	11.1
47	11.4	11
48	11.3	10.8
49	11.5	10.9

Test #	Referenced Moisture Sensing Device	Resistive Soil Moisture Sensor
50	11.5	10.87

Given the data above, their mean, variance, and standard deviation can be seen below:

```
refmoisture_device - Mean: 11.55 Variance: 0.15 Standard Deviation: 0.39
resistive_sensor - Mean: 11.42 Variance: 0.26 Standard Deviation: 0.51
```

Figure 3-9. AgraTronix Moisture Tester and Resistive Moisture Sensor Mean, Variance, and Standard Deviation

On average, the reference device reads slightly higher moisture levels (11.55) than the resistive sensor (11.42). This means the reference device tends to measure a slightly wetter environment than the resistive sensor.

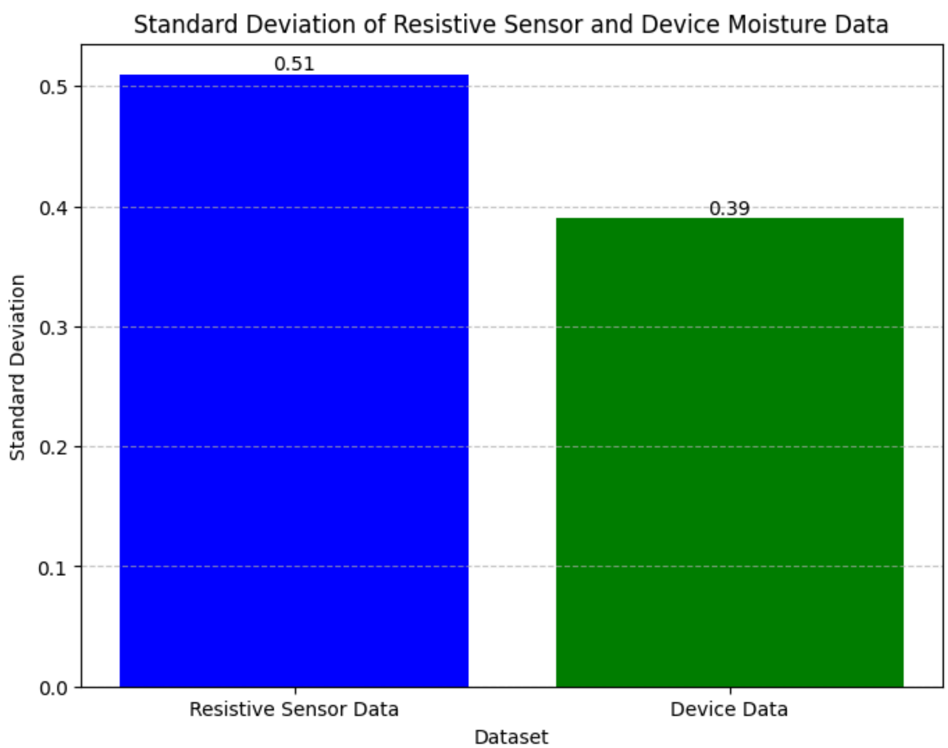


Figure 3-10. Standard Deviation of Resistive Sensor and Device Moisture Data

Regarding how much the readings vary, the reference device fluctuates less, with a standard deviation of 0.39, which indicates that its readings are relatively consistent. In contrast, the resistive sensor shows more variability, with a standard deviation 0.51. This suggests that the resistive sensor’s readings are more spread out and less consistent than the reference device's. These differences mean that while both devices measure moisture, the resistive sensor has more variability in its results (*see Figure 3-10*).

Based on several mathematical computations, the resistive sensor and the reference device exhibit a mean of 11.55, a variance of 0.15, and a standard deviation of 0.39, suggesting relatively consistent measurements with less variability. In contrast, the resistive sensor has a slightly lower mean of 11.42, with more significant variability (variance of 0.26 and a standard deviation of 0.51), reflecting less consistency in its readings.

On the other hand, the capacitive moisture device shows a mean of 11.66, a variance of 0.27, and a standard deviation of 0.52, which are somewhat higher than those of the reference device but show more alignment with the reference device compared to the resistive sensor.

The capacitive moisture device is the better choice, providing a more consistent and reliable measurement than the resistive sensor. This suggests that the capacitive sensor is more closely aligned with the reference device’s readings, making it a preferable option for applications requiring accuracy and consistency.

3.6 Design Options

3.6.1 Summary of Design Options (per Constraint)

The developers designed three design options to address the project’s objectives by assessing their efficacy by validating design constraints. Each design constraint has different components that can produce different results and yield different implications. Ultimately, the goal of the three proposed designs is to address the following problems: where the design option must be able to adapt to changes in temperature when it drops, leading to performance constraints, where the automated fan-control system must be reliably fast enough to turn on or off when the system detects new readings, and lastly, must consume low power. These options will help the developers decide which design suits the project objectives.

Table 3-5. Summary of Design Options (per Constraint)

Constraints	Design Option 1	Design Option 2	Design Option 3
Performance	● 316 Food-grade Stainless Metal Steel (4m x 6m) ● 6xPolyurethane Foam (1.2m x 1m x 6mm) ● Acrylic Plastic (1.2m x 1m) ● 3xThermoelectric TEC-12706 Peltier Cooling System Heatsink Kit	● Food-Grade Stainless Steel (4m x 6m) ● Acrylic Plastic (1.2m x 1m) ● 3xPolyolefin Foam (1.2m x 1m x 12mm) ● Radiator Fan	● 3xFood-Grade Stainless Steel (1.2m x 1m) ● Food-Grade Stainless Steel (1.2m x 0.4m) ● 2xThermoelectric TEC-12706 Peltier Cooling System Heatsink Kit
Response Time	● Raspberry Pi 4B 8GB ● 4xThermoelectric TEC-12706 Peltier Cooling System Heatsink Kit	● Raspberry Pi 4B 8GB ● Radiator Fan	● Raspberry Pi 4B 8GB ● 2xThermoelectric TEC-12706 Peltier Cooling System Heatsink Kit
Power Consumption	● Raspberry Pi 4B 8GB ● Arduino Uno R3 ● Resistive Soil Moisture Sensor ● 3xDHT22 Module ● Grove Red LED ● 3xThermoelectric TEC-12706 Peltier Cooling System Heatsink Kit ● 2x4-channel 5V Relay Module	● Raspberry Pi 4B 8GB ● Arduino Uno R3 ● Capacitive Soil Moisture Sensor v2.0 ● 3xDHT22 Module ● Radiator Fan ● 2-channel 5V Relay Module	● Raspberry Pi 4B 8GB ● Arduino Uno R3 ● Capacitive Soil Moisture Sensor v2.0 ● 3xDHT22 Module ● 2xThermoelectric TEC-12706 Peltier Cooling System Heatsink Kit ● 2-channel 5V Relay Module

Table 3-5 compares the materials and components used by each design. They are all tailored to different project constraints, including performance, response time, and power consumption. The materials and components selected for each design option were specifically chosen to meet the system requirements, allowing the developer to assess tradeoffs and their advantages based on the components used.

3.6.2 Design Options Breakdown

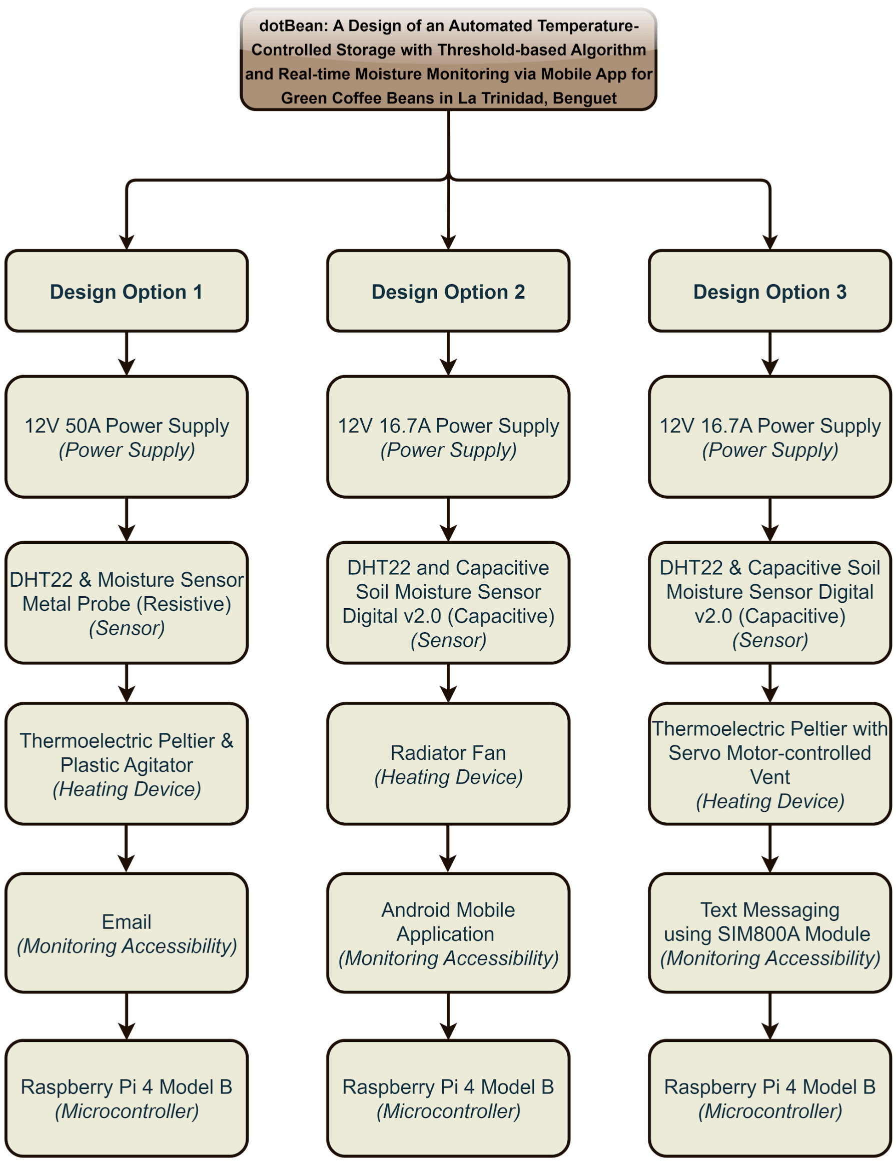


Figure 3-11. Design Options Breakdown

The project objectives and the client requirements mainly influenced the selection of design constraints and hardware components. Therefore, the design options breakdown of the dotBean, as depicted in Figure 3-11, focuses on different sensor components and actuators that will serve as a basis for various outputs and results during testing and validation of design constraints. With this strategy, the developers can assess the feasibility and usability of the system in terms of real-time monitoring and energy efficiency.

3.6.3 Design Option 1

Design option 1 of dotBean's main features involves utilizing a Plastic Agitator and Thermoelectric pellet for temperature—and humidity-related activities and a dielectric method for measuring the moisture content of the coffee beans, specifically a resistive moisture sensor. The user will be notified through their email address/es for monitoring purposes. Below is the discussion of how this design option works and functions under dotBean's objectives and client requirements.

3.6.3.1 System Architecture

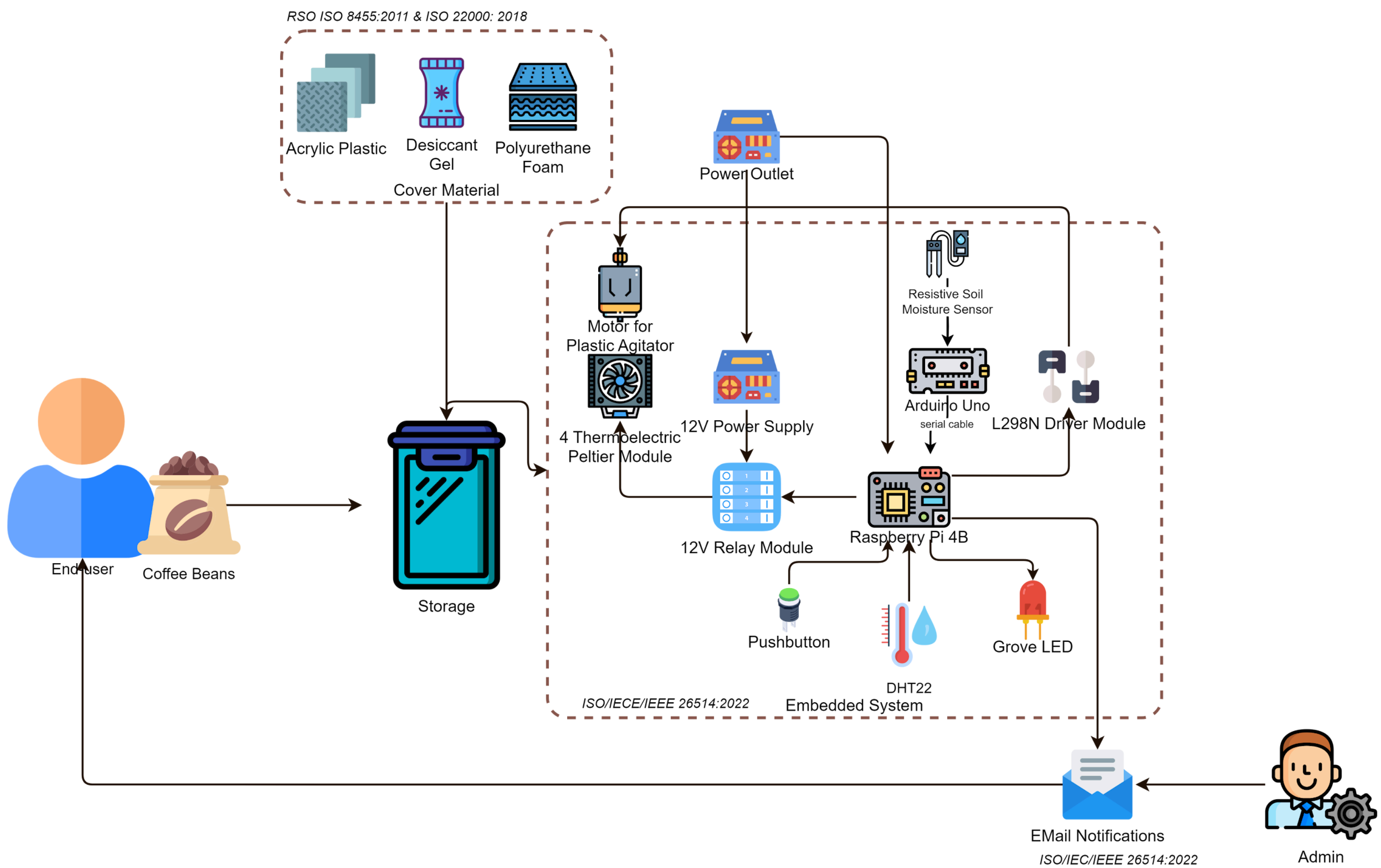


Figure 3-12. Design Option 1 System Architecture

Similarly, with the dotBean general system architecture, the Design Option 1 system architecture explicitly illustrates how the dotBean operates in this design option. Design Option 1's cover materials include acrylic plastic, foam, and desiccant gel. The following cover materials are effective moisture absorbents and help prevent the absorption of excessively low or excessively high temperatures from outside. This design option's actuators mainly feature a Thermoelectric Peltier Module and a Plastic Agitator to distribute the air within the container equally and maintain the desired conditions. Integrated sensors are DHT22 for temperature and humidity monitoring and a Moisture Sensor with a Metal Probe for measuring the moisture content. A Moisture Sensor with a Metal Probe is a resistive type of sensor where there is direct contact between its probe and the coffee beans.

Consequently, the readings from two sensors are transmitted in the Raspberry Pi 4B for data analysis. This microcontroller also controls the actuators that regulate and maintain the desired conditions inside the container. Lastly, this design option offers an accessibility option through email notifications. All of the actions and commands within the system are operated by the Raspberry Pi 4B, a versatile and all-around microcontroller known for its reliability and efficiency.

3.6.3.2 Design Process Flow

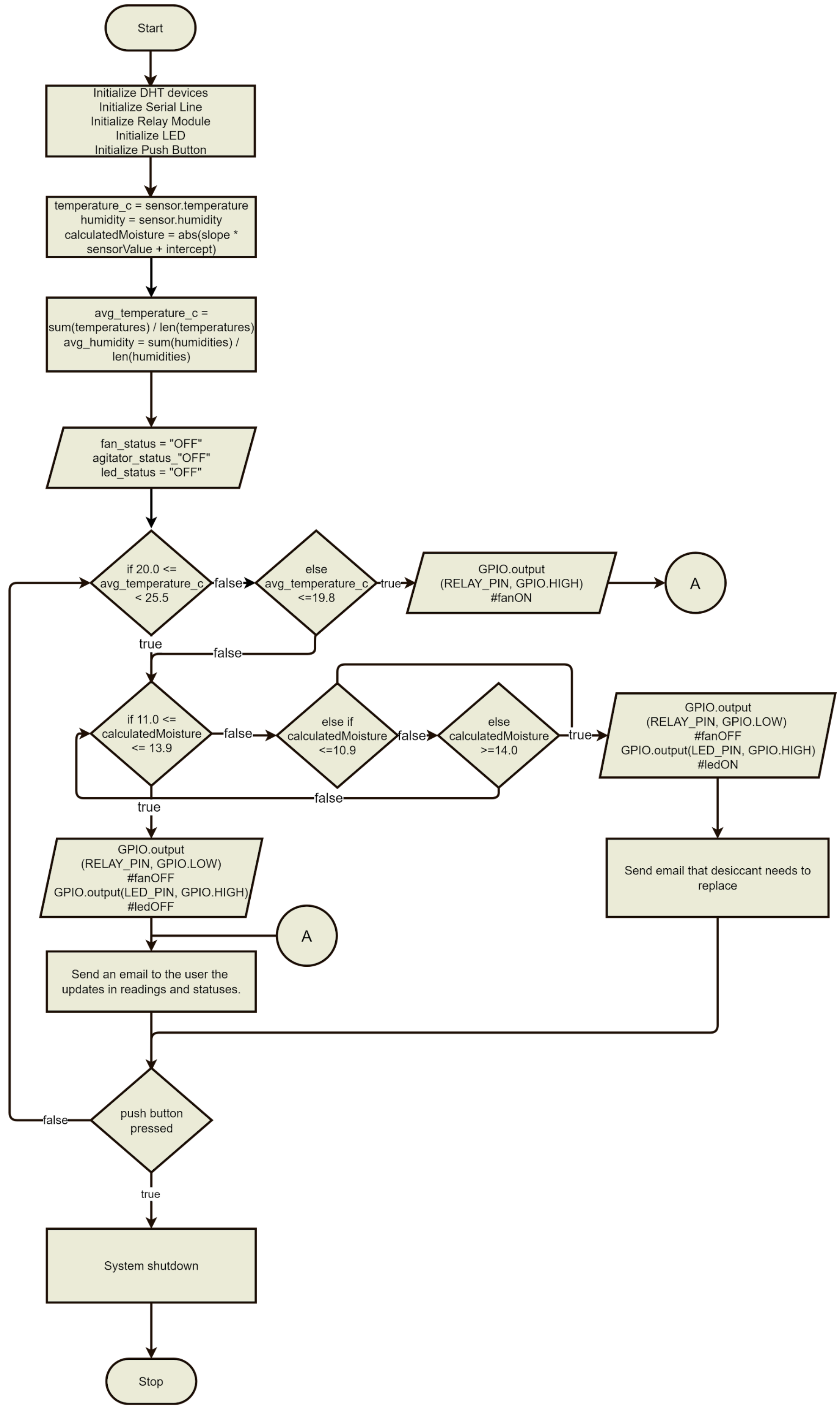


Figure 3-13. Design Option 1 Design Process

The design process of Design Option 1, as illustrated in Figure 3-13, involves retrieving data from sensors to Raspberry Pi and the data analysis process. This design option utilzed a metal probe moisture sensor to monitor the moisture content of the coffee beans. A DHT22 sensor is also deployed inside the container to monitor the temperature and humidity, balancing the right amount of air inside and achieving the desired environmental conditions and the coffee beans' moisture content. The Raspberry Pi will send another command to turn the actuators off if it is between the pre-set parameters for temperature and humidity levels. These readings and status will be sent to the client via EMail.

However, when the moisture content exceeds the recommended range of 11% - 12.5%, the desiccant gel has reached its saturation point even if other environmental parameters, such as temperature and relative humidity, are maintained within the specified limits. It can no longer effectively absorb more moisture. A desiccant gel is a drying compound that absorbs moisture from the surrounding environment. Its effectiveness is crucial for maintaining the desired moisture levels within a given space or container. Notifying the client promptly about replacing the desiccant gel is essential to prevent potential damage to the coffee beans that require controlled moisture levels.

3.6.3.3 Software Design

Design Option 1 incorporates accessibility features through an integrated EMail messaging notification. This system delivers real-time data from sensors and the current statuses of actuators, all processed by the Raspberry Pi. In line with dotBean's commitment to full automation, the EMail notifications convey the latest data and statuses, conforming to predetermined parameters. The following sub-parts outline various software applications relevant to the study. Lastly, the software development, including its flow within the software process, is illustrated in a figure and discussed accordingly.

3.6.3.3.1 Software Technologies

SolidWorks for Prototype Layouting and Simulation

SolidWorks is known for its computer-aided design (CAD) and engineering (CAE) prowess. Offering a parametric modeling approach, the software empowers users to create 3D models by defining parameters and constraints, facilitating easy modifications. The developers utilized this software to lay out the prototype of each design option and conduct a simulation to attest and validate the design option's capability to suffice the design constraints and design criteria. Overall, SolidWorks, with its parametric modeling, assembly features, simulation capabilities, and integration support, is a valuable tool in designing and developing computer engineering projects, providing a comprehensive solution for creating electronic systems with precision and efficiency.

Fritzing for Circuit and Schematic Diagram

Fritzing distinguishes itself by providing a user-friendly interface specifically tailored to accommodate individuals new to electronics and beginners in the field. Including drag-and-drop functionality, coupled with visual representations of components, streamlines the process of circuit creation, offering a simplified and intuitive design experience. With its breadboard view feature, the developers easily experiment with circuit designs in a simulated environment before transitioning to a tangible prototype. This functionality is instrumental in identifying potential issues and errors during the design process.

Geany as Raspberry Pi IDE for Programming

Geany establishes itself as a robust code editor for Raspberry Pi, given its pre-installation in Raspberry Pi OS and its adeptness in facilitating Python or C/C++ coding. A built-in terminal enables developers to compile and execute scripts seamlessly within the Geany environment. Geany IDE is utilized for sensor and actuator configuration, and EMail configuration is used to allow users to monitor changes in environmental conditions from sensor readings and actuator status.

3.6.3.3.2 Software Development

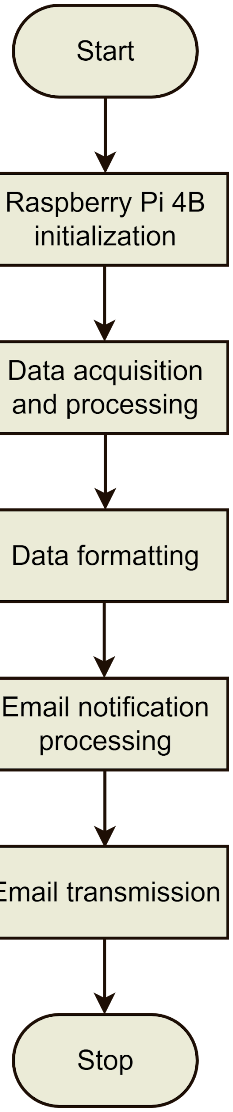


Figure 3-14. Design Option 1 Software Process

Upon initialization, the Raspberry Pi configures its software modules, specifically those tasked with interfacing with sensors and actuators. As operations commence, the Raspberry Pi's software interfaces with sensors and actuators, retrieving pertinent data such as temperature metrics or patterns. The Raspberry Pi undertakes processing where it ensures data accuracy and optimizes it for display purposes. Moreover, during post-processing, the data undergoes formatting that is applicable for transferring the data from the Raspberry Pi to a channel that caters to EMail notification processing. Using established communication protocols, the Raspberry Pi transmits the precise text derivatives to clients through their given Email Address/es.

In this design option, the developers utilized sending EMail messages to address all the variables and values present in the container, including the temperature and humidity readings from the DHT22, the moisture content level from the 50kg load sensor, the status of both the plastic agitator and thermoelectric Peltier module, and, lastly, the replacement of the desiccant gel.

3.6.3.4 Hardware Design

The primary elements comprising Design Option 1 include the Plastic Agitator and Thermoelectric Peltier Module, which are dedicated to regulating temperature- and humidity-related activities. The Plastic Agitator's function involves the rotation of coffee beans, ensuring an even air distribution within the container. Simultaneously, the Peltier Module produces air to maintain the specified temperature range. Three strategically positioned DHT22 sensors are employed to monitor temperature and humidity levels across the container accurately. These sensors offer precise measurements, covering the entire spectrum from the bottom to the top of the container.

Consequently, developers have adopted the gravimetric method to assess the moisture content of the coffee beans in this design option. This approach incorporates at least two 50 kg load sensors to measure the weight of the coffee beans. Lastly, the developers employed email notifications as the primary communication between the user and the system to allow the clients to monitor the updates and statuses of the storage container seamlessly. Below is an overview of the components selected by the developers for this specific design option and its prototype design:

3.6.3.4.1 Prototype Layout

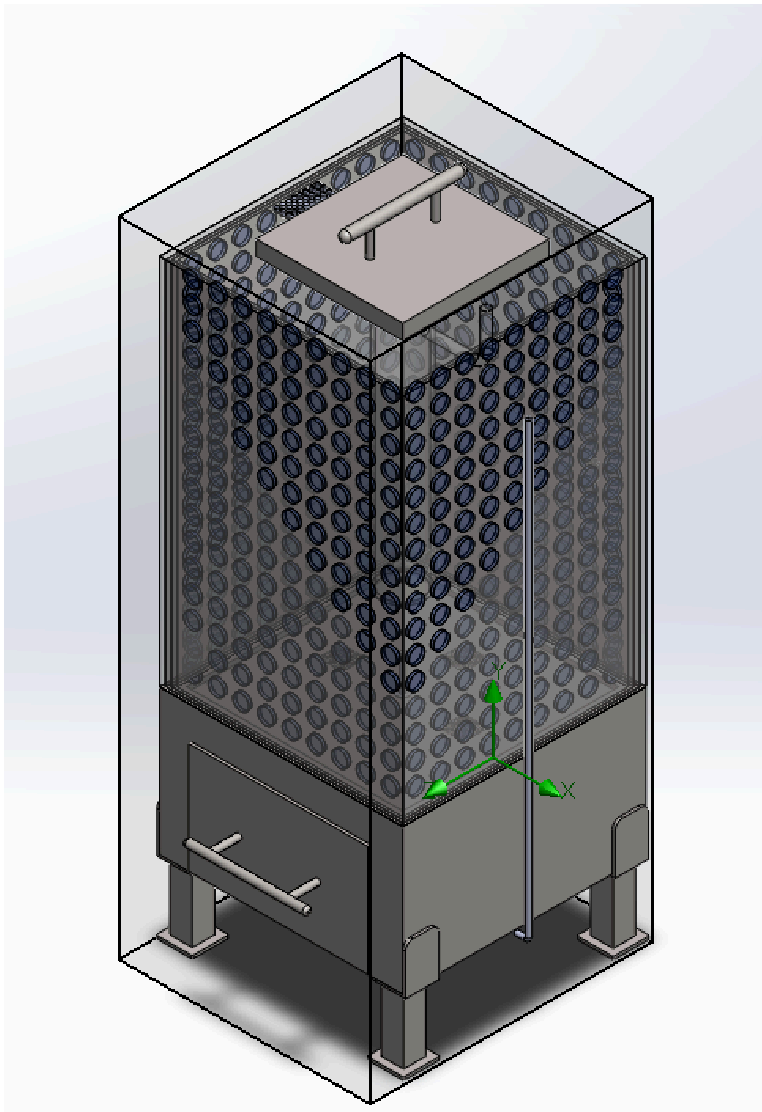


Figure 3-15. Design Option 1 3D View

A prototype layout plays a vital role in the product development lifecycle, providing a tangible representation of design concepts that function as both a visual and functional model that is reliable for conceptualizing and evaluating the proposed product or system. The preceding figure represents the isometric view of Design Option 1, whose purpose relies solely on its capability to store coffee beans, monitor their moisture content, and control the temperature inside. It serves as the primary storage container for the target clients.

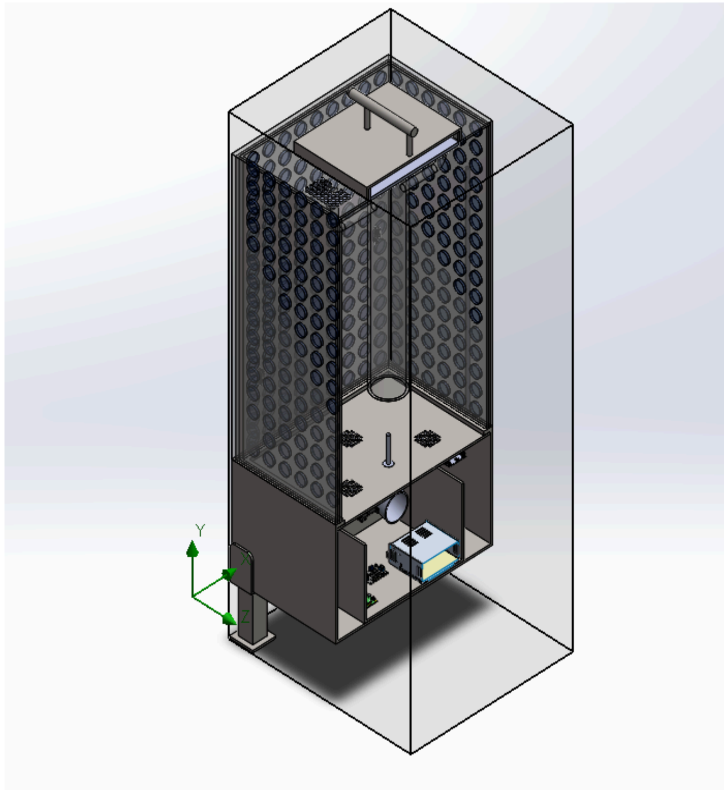


Figure 3-16. Design Option 1 Section View

Design option 1 consists of different types of hardware components classified depending on their types, such as cover materials including the inner and outer layer, the microcontroller and sensors, the heating and cooling regulator devices, and the other components that will power the device (*see Figure 3-16*). Acrylic plastic is used as the primary cover material for the container for storing coffee beans due to its cost-effectiveness and durability. It is also designated as a food-safe acrylic container that meets FDA standards and provides a secure option for storing coffee beans. The polyurethane foam helps the device to be insulated in terms of energy conservation, and it also achieves air tightness in cold rooms while maintaining the device's temperature. The desiccant gel pack acts as a moisture-absorbing material to remove the excess moisture without altering the food's composition. The agitator will act as a regulator for distributing the coffee beans equally in the container. The thermoelectric Peltier fans will act as a heating device and turn on when necessary, making the temperature reach at least 20 when it is below. A 12V power supply will power the components.

3.6.3.4.2 Schematic Diagram

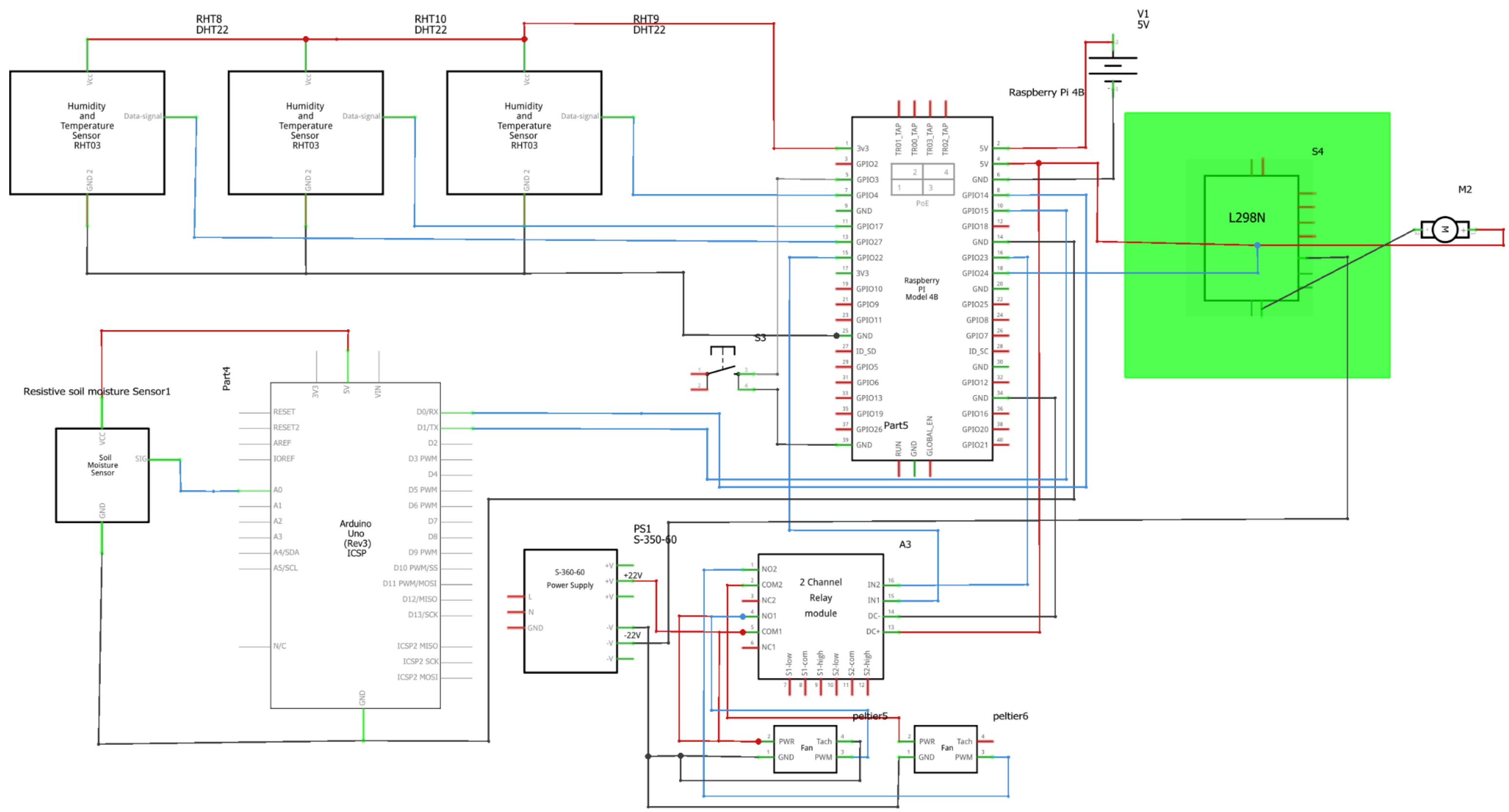


Figure 3-17. Design Option 1 Schematic Diagram

A schematic diagram outlines the functionality and connectivity among different electrical components in a two-dimensional circuit (Yogendrappa, 2021). In this way, the developers can plan the necessary components required to make the system functional and complete their connection. This schematic diagram of design option 1, as illustrated in Figure 3-17, shows the different components connected either with one another or in a microcontroller; included in this figure are the following components: Raspberry Pi 4B, DHT22 sensor, Resistive Moisture Sensor and Arduino Uno that act as middlemen between the moisture sensor and Raspberry pi, a DC motor that rotates the plastic agitator, Thermoelectric Peltier Module, 12V Power Supply Unit, a momentary pushbutton, and a 2-channel 12V relay module.

Below are the complete hardware components used in this design option and a discussion of their specifications and descriptions to help you understand each component's functionalities.

3.6.3.4.3 Hardware Components

Microcontroller and Sensor Components

- Raspberry Pi 4B 8GB
- Resistive Soil Moisture Sensor with Metal Probe
- DHT22 Module
- Grove Red LED

Cover Materials (outer and inner)

- Acrylic Plastic
- 316 Food-grade Stainless Metal Steel
- Desiccant Gel
- Polyurethane Foam

Heating Regulator Devices

- Plastic Agitator
- Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit

Other Components

- 12V Switching Power Supply Unit (50A & 16.7A)
- Sandisk Extreme 64GB SD Card
- L298N Driver Module
- Momentary Push Button Switch
- Raspberry Pi Charger Cable
- Extension Cord

3.6.3.4.4 Functional Specifications

Raspberry Pi 4B 8GB

The Raspberry Pi 4B is a single-board computer featuring a 40-pin GPIO (General Purpose Input/Output) header. This header is utilized to connect the Raspberry Pi to external devices. The GPIO pins can be set as input or output and are used for various applications. They can also generate output from PWM (Pulse et al.). The board supports SPI (Serial et al.), I2C (Inter-Integrated Circuit), and UART (Universal et al.) serial communication protocols.

Table 3-6. Raspberry Pi 4B GPIO Configuration

Pin Type	GPIO Pins
GPIO Pins	GPIO2, GPIO3, GPIO4, GPIO5, GPIO6, GPIO7, GPIO8, GPIO9, GPIO10, GPIO11, GPIO12, GPIO13, GPIO14, GPIO 15, GPIO 16, GPIO 17, GPIO 18, GPIO 19, GPIO 20, GPIO 21, GPIO 22, GPIO 23, GPIO 24, GPIO 25, GPIO 26, GPIO 27
Power Pins	3.3V, 5V, GND
PWM pins	GPIO12, GPIO13, GPIO18, GPIO19
SPI Pins	SPI0: GPIO9 (MISO), GPIO10 (MOSI), GPIO11 (SCLK), GPIO8 (CE0), GPIO7 (CE1) SPI1: GPIO19 (MISO), GPIO20 (MOSI), GPIO21 (SCLK), GPIO18 (CE0), GPIO17 (CE1), GPIO16 (CE2)
I2C Pins	Data: (GPIO2), Clock: (GPIO3) EEPROM Data: (GPIO0), EEPROM Clock: (GPIO1)
UART Pins	TX: (GPIO14) RX: (GPIO15)
Other Pins	ID_SD (I2C ID EEPROM) ID_SC (I2C ID EEPROM)

GPIO pins are used for general-purpose input/output operations. This allows for the control of various digital devices, such as LEDs, buttons, and sensors. These pins also can read data from these devices. On the contrary, PWM pins are designed to generate pulse width modulation signals, which control the amount of power supplied to devices like motors or LEDs by adjusting the signal's duty cycle. Moreover, SPI pins are synchronous serial communication interfaces used for short-distance communication, primarily in embedded systems. It enables full-duplex communication between a controller device and one or more secondary devices.

Additionally, other pins are known as I2C pins, a communication protocol used for short-distance, intra-board communication. It allows multiple low-speed devices to communicate with each other using only two wires. Furthermore, UART, TX, and RX pins are used for Universal Asynchronous Receiver/Transmitter communication. UART is a hardware device or software implementation that translates data between parallel and serial forms. It is used for serial communication between devices. Lastly, other pins include ID_SD, which supports SD card communication, while ID_SC pins support SPI communication. SD cards are a type of flash memory card used for storing digital data, while SPI is a synchronous serial communication interface. See the figure below to visualize the different GPIO pins discussed above:

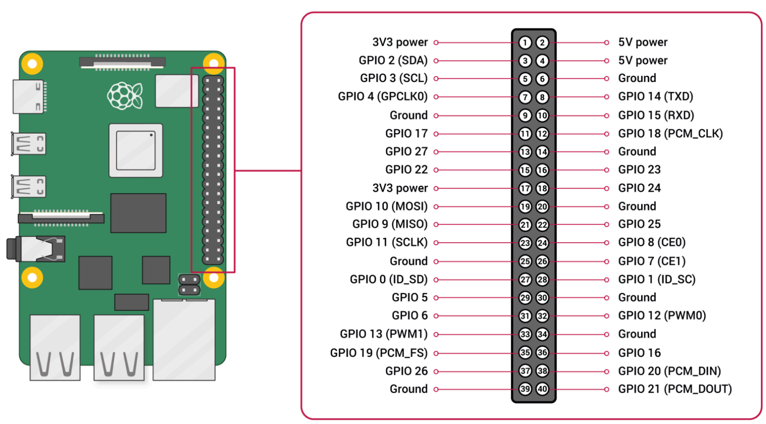


Figure 3-18: Raspberry Pi 4B 40-GPIO Pins

The Raspberry Pi 4 Model B is a high-performance single-board computer introduced by the Raspberry Pi Foundation in June 2019. It is powered by a 64-bit Broadcom 2711 Cortex A72 processor at 1.5GHz. Table 3-7 provides a detailed overview of the Raspberry Pi 4 Model B's capabilities. It includes information about the processor, memory, connectivity options, and power requirements, among other details.

Table 3-7. Raspberry Pi 4B Specification

Raspberry Pi 4 Computer Model B Specification	
Specification Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	1GB, 2GB, 4GB, or 8GB LPDDR4 (depending on model) with on-die ECC
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports two × USB 2.0 ports.
GPIO	Standard 40-pin GPIO header (fully backward-compatible with previous boards)
Video & sound	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); and OpenGL ES, 3.0 graphics
SD card support	Micro SD card slot for loading operating system and data storage
Input power	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Operating temperature 0–50ºC

Raspberry Pi 4 Model B, a high-performance single-board computer, is powered by a 64-bit Broadcom BCM2711 Cortex-A72 processor that operates at 1.5GHz. It comes in three memory configurations: 1GB, 2GB, 4GB, or 8GB LPDDR4, with on-die ECC, and supports a wide range of tasks from simple to complex applications. For connectivity, it supports 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE, and Gigabit Ethernet. It also has two USB 3.0 ports and two USB 2.0 ports, which can connect various peripheral devices. The Raspberry Pi 4 also features a standard 40-pin GPIO header, which is fully backward-compatible with previous boards. This allows for a wide range of input/output operations, making it a versatile tool for many projects.

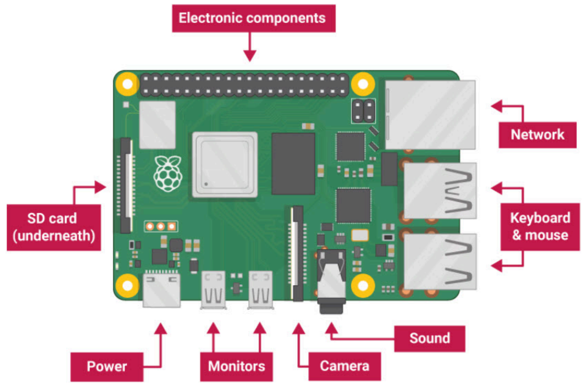


Figure 3-19: Raspberry Pi 4B General Parts

Furthermore, as shown in Figure 3-19, the Raspberry Pi 4B supports 2 × micro HDMI ports (up to 4Kp60 supported), a 2-lane MIPI DSI display port, a 2-lane MIPI CSI camera port, and a 4-pole stereo audio and composite video port. The Raspberry Pi 4 supports H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode), and OpenGL ES 3.0 graphics. It also has a micro SD card slot for loading the operating system and data storage. The Raspberry Pi 4 can be powered via a 5V DC USB-C connector or a 5V DC GPIO header, providing a minimum of 3A. It also supports Power over Ethernet (PoE) capability, although this requires a separate PoE HAT.

Soil Moisture Sensor Detector Module Humidity Test Sensor with Metal Probe

Since it is mainly designed to measure soil moisture levels, the developers calibrated this sensor to measure resistance in coffee beans. This module comprises various components, including a moisture sensor, resistors, capacitors, a potentiometer, and a comparator IC (LM393), all integrated into a single circuit.

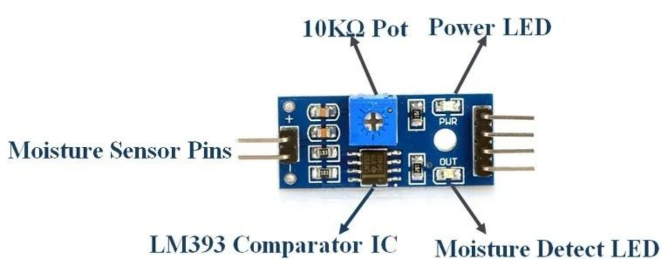


Figure 3-20: Soil Moisture Sensor Detector Module and Parts

The module's core lies the LM393 Comparator IC, the central processing unit. By connecting pin 2 of the LM393 to a preset (a 10kΩ potentiometer) and pin 3 to the moisture sensor, the comparator can compare the preset threshold voltage with the voltage detected from the moisture sensor. This comparison process is crucial for determining whether the soil is dry, moist, or somewhere between, enabling the module to generate a digital signal indicating the soil's moisture level (*see Figure 3-20*).

Figure 3-21: Soil Moisture Sensor Detector Module with Metal Probe

The moisture sensor features two probes, as shown in Figure 3-21. These probes are vital for passing current through the soil, enabling the sensor to measure resistance and reduce moisture content. The onboard preset allows users to fine-tune the sensitivity of the digital output, enhancing the module's adaptability to different soil types and moisture levels.

Table 3-8. Soil Moisture Sensor Detector Module Humidity Test Sensor with Metal Probe Specifications

Soil Moisture Sensor Detector Module Humidity Test Sensor with Metal Probe Specifications	
Operating Voltage	3.3V to 5V DC
Operating Current	15mA
Output Digital	-0V to 5V, Adjustable trigger level from preset
Output Analog	-0V to 5V based on infrared radiation from fire flame falling on the sensor
LED	Output and power indication

As shown in Table 3-8, the operational specifications are as follows: the module operates within a voltage range of 3.3V to 5V DC, ensuring compatibility with various electronic systems while consuming a current of 15mA, striking a balance between efficient fire detection and power usage. It provides digital and analog outputs, with digital signals ranging from 0V to 5V and customizable trigger levels and analog signals varying from 0V to 5V based on infrared radiation from fire flames interacting with the sensor. Additionally, LED indicators are integrated to display output and power status, enhancing usability and aiding in troubleshooting processes.

DHT22

The DHT22 is a handy, cost-effective device that measures humidity and temperature. It is a digital sensor with a built-in analog-to-digital converter, making connecting with the Raspberry Pi easier.

Table 3-9. DHT22 Specification

DHT22 Specification	
Operating Voltage	3.5V to 5.5V
Operating current	0.3mA (measuring) 60uA (standby)
Output	Serial Data
Temperature Range	-40°C to 80°C
Humidity Range	0% to 100%
Resolution	Temperature and Humidity both are 16-bit
Accuracy	±0.5°C and ±1%

Table 3-9 illustrates the specifications for the DHT22 module; its operating voltage is between 3.5V and 5.5V, and the operating current is 0.3mA during measurement and 60uA in standby mode. This sensor outputs both temperature and humidity data through a serial connection. The temperature range is from -40°C to 80°C, and the humidity range is from 0% to 100%. Moreover, both temperature and humidity readings are 16-bit, providing high resolution, and the accuracy readings of the sensor are ±0.5°C for temperature and ±1% for humidity.

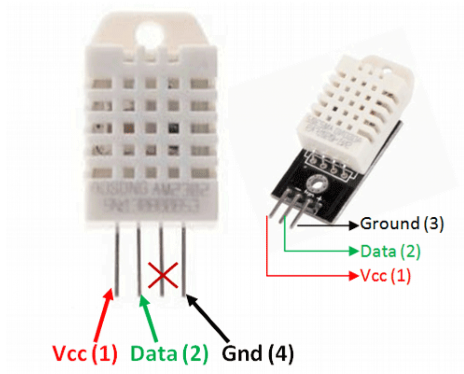


Figure 3-22. DHT22 Pin Configurations

The Vcc pin is the power supply, which requires a voltage between 3.5V and 5.5V. The Data pin is the output pin that transmits temperature and humidity data in a serial format. Lastly, the Ground pin, as its name suggests, is connected to the circuit's ground. This pin configuration is fundamental for the DHT22 module to function correctly.

Grove Red LED

Figure 3-23. Grove Red LED

The Grove Red LED, as depicted in Figure 3-23, is a component that can notify the user when the desiccant gel needs to be replaced. It is similar to the Grove LED module, which houses an LED light source and also includes a potentiometer on board to manage the power requirements of the LED. It is compatible with both 3.3V and 5V systems, and a potentiometer can adjust its brightness.

Acrylic Plastic

Figure 3-24. Acrylic Plastic

As shown in Figure 3-24, Acrylic Plastic or Plexiglas is a material for projects due to its optical clarity and impact resistance. It is lightweight, chemically resistant, and easy to machine, making it suitable for various components. In this design, it will be used as the primary cover material and to house sensors that measure the parameters.

316 Food-grade Stainless Metal Steel

316 Food-grade stainless steel is known for being resistant to corrosion and for its resistance to chlorides and acids, making it ideal for use in environments where salt exposure is expected. This device can also withstand extreme cold and hot temperatures, making it ideal for storage.

Desiccant Gel

Figure 3-25. Desiccant Gel

Desiccant Gel, shown in Figure 3-25, is a hygroscopic substance used to induce or sustain a state of dryness in its vicinity. It is the opposite of a humectant. Commonly encountered pre-packaged desiccants are solids that absorb water. They are used to maintain dryness in various environments, whether preserving food products or preventing moisture damage to electronics or medications. They absorb the humidity and moisture in the air to maintain a moisture-free environment, which is essential to prevent damage, corrosion, bacteria, and mold growth.

Polyurethane Foam

Figure 3-26. Polyurethane Foam

Polyurethane foam, as depicted in Figure 3-26, is a versatile material used in a wide range of applications due to its unique properties. It is an important component in major appliances that consumers use every day. The most common use of polyurethane in major appliances is rigid foams for refrigerator and freezer thermal insulation systems.

Plastic Agitator

Figure 3-27. Plastic Agitator

A plastic agitator can be designed to generate sufficient agitation force to disrupt the liquid's surface and promote efficient heat transfer. The motor drive of the agitator should be capable of controlling the agitation speed, which is crucial for maintaining the desired temperature and humidity levels. The impeller shape should be optimized for the liquid's physical properties and the agitator's intended use. The material of the agitator should be durable and food-safe, ensuring it can withstand the rigors of the storage process without posing any safety risks (*see Figure 3-27*).

Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit

Figure 3-28. Thermoelectric TEC-12706 Peltier

As illustrated in Figure 3-28, a Thermoelectric Peltier module is a thermal control device that can both warm and cool surfaces by passing electric current through it. This is achieved through the Peltier effect, where electrons move in one element and positive holes move in the other, allowing one side of the substrate to absorb heat and the other to radiate heat. The module's structure consists of two types of semiconductor elements arranged in tandem sandwiched between copper substrates. Using the Seebeck effect, it can also be used as a thermoelectric power generation module.

Table 3-10. Thermoelectric Peltier Design Specification

Thermoelectric Peltier Design Specification	
Material	Metal
Voltage	12V
Electricity Current	5.8A
Power	50-60W

Thermoelectric Peltier Design Specification	
Weight	350g
Fan Size	92 x 92 x 255 mm

Table 3-9 encompasses the thermoelectric Peltier design specifications, including using metal for its heat conduction properties, operating at 12V with a current draw of 5.8A. The system consumes 50-60W of power and weighs approximately 350g. It incorporates a 92 x 92 x 25 mm fan, essential for optimizing airflow and enhancing heat dissipation.

12V Switching Power Supply Unit (50A)

Figure 3-29. 12V Switching Power Supply

A 12V power supply unit is an electronic device with a regulated 12V DC output that converts electrical power from a higher voltage. This aims to provide a maximum current of 50A to the Peltier fans and agitator.

Sandisk Extreme 64gb SD Card

Figure 3-30. Sandisk Extreme 64gb SD Card

The Sandisk Extreme 64GB SD Card, as shown in Figure 3-30, is a storage solution for Raspberry Pi. It can store the operating system, applications, and data. This SD Card has a storage capacity of 64GB, a transfer rate of 100MB/s, and a read speed of up to 90MB/s. It is UHS-I compliant, meaning it can support high-speed data transfer. The card uses a microSD card form factor, compatible with the Raspberry Pi's microSD slot.

L298N Driver Module

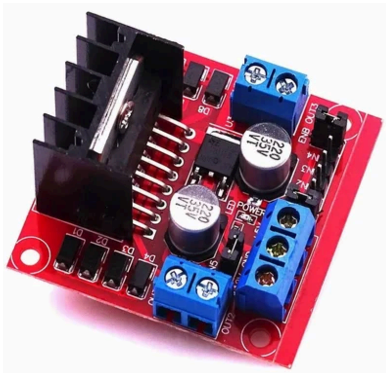


Figure 3-31. L298N Module

The L298N Driver Module is a dual bidirectional motor driver that will independently control the operation of two DC motors. This module is based on the L298N Motor Driver IC, which contains two standard H-bridges capable of driving a pair of DC motors. It has a supply range of 5V to 35V and is capable of 2A continuous current per channel, making it suitable for most DC motors. This module will be used with the Raspberry Pi 4B to control agitators. It will allow for precise control over the speed and direction of the motor (*see Figure 3-31*).

4-Channel 5V Relay Module

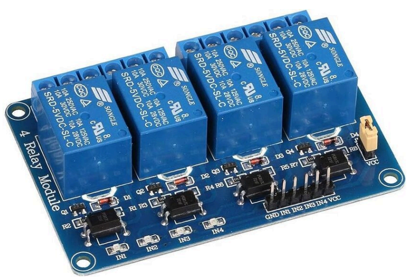


Figure 3-32. 4-Channel 5V Relay Module

The 4-channel 5V relay module is a compact electronic component that allows low-voltage microcontrollers like Arduino or Raspberry Pi to control higher-voltage devices.

Table 3-11. Pin Configuration of 2-Channel 5V Relay Module

Pin		Description
Power Pins	**VCC (or JD-VCC)**	Connected to the 5V pin of the microcontroller power supply.
	GND	Connected to the ground (GND) of the power supply.
Signal Pins	**IN1**	Receives the signal control to relay 1 module.
	IN2	Receives the signal control to relay 2 modules.
	IN3	Receives the signal control to relay 3 modules.
	IN4	Receives the signal to relay 4 modules.
Output Pins	**NO (Normally Open)**	This pin is not connected to COM when the relay is not activated (off-state).
	NC (Normally Closed)	This pin is connected to COM when the relay is not activated (off-state).
Jumper Settings	**Low-Level Trigger (LL)**	The relay is triggered when the input signal is low (0V).
	Low-Level Trigger (LL)	The relay is triggered when the input signal is high (5V or 3.3V, depending on the microcontroller).

Table 3-11 illustrates the pin configuration of a 5V relay. The power pins VCC and GND connect to the microcontroller to provide power and ground, respectively. Signal pins IN1 and IN2 receive control signals from the microcontroller to activate or deactivate each relay independently. Output pins NO and NC dictate the connection state with COM, providing flexibility in circuit design based on whether the relay is activated. Finally, jumper settings allow users to choose between low-level or high-level triggering based on the microcontroller's output signal characteristics.

Momentary Push Button Switch PBS-110

Figure 3-33. Momentary Push Button Switch PBS-110

The Momentary Push Button Switch PBS-110, as depicted in Figure 3-33, momentarily closes the normally open contact to the common contact. This action allows for brief circuit activation, commonly used in reset mechanisms or as manual control inputs in electronic devices and consumer electronics. It is used to open and close the system.

Raspberry Pi Charger Cable

Figure 3-34. Raspberry Pi Charger Cable

The Okdo Raspberry Pi Power Supply, which comes with a USB Type-C connector, 1.5m cable, and EU plug type, is a reliable choice for powering the Raspberry Pi board. It is designed to support different Raspberry Pi models and delivers sufficient power for stable operation.

Extension Cord

Figure 3-35. Extension Cord

As demonstrated in Figure 3-35, the extension cord is designed to extend the reach of power sources to devices that require placement beyond the proximity of standard outlets. In this project, the primary function of the extension cord is to provide an extended pathway for Raspberry Pi and 2 power supplies in a single outlet to access power from a distant location within the building or location. This capability is particularly beneficial in environments where mobility and reconfiguration of electrical setups are essential, enhancing operational efficiency and convenience.

3.6.3.4.5 Bill of Materials

Due to the cost constraints of the developers, below is the summary bill of materials for Design Option 1. Every item that the developers should consider, from its cover materials and fans to be used up to the sensors and other components, is listed below to ensure that this design option is within the cost constraints of the project.

Table 3-12. Design Option 1 Bill of Materials

Item	Qty.	Price/pc. (₱)	Total Price (₱)
Raspberry Pi 4B 8GB	1	5907	5907
Resistive Soil Moisture Sensor	1	75	75
DHT22 Module	3	200	600

Item	Qty.	Price/pc. (₱)	Total Price (₱)
Grove Red LED	1	49	49
Acrylic Plastic (1.2m x 1m)	1	15000	15000
316 Food-grade Stainless Metal Steel (4m x 6m)	1	3500	3500
Desiccant Gel	4	200	800
Polyurethane Foam (1.2m x 1m x 6mm)	6	700	4200
Plastic Agitator	1	4000	4000
Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit	4	248	992
12V 50A Switching Power Supply Unit	1	1450	1450
Sandisk Extreme 64GB SD Card	1	250	250
L298N Driver Module	1	74.75	74.75
4-channel 5V Relay Module	1	159	159
Momentary Push Button Switch	1	65	65
Raspberry Pi Charger Cable	1	300	300
Extension Cord	1	250	250
Total (₱)			37056.75

The table above is the estimated cost of Design Option 1 based on the required quantity for each component, and the probability of encountering difficulties or defects resulting from the purchase of new materials is beyond the developers' control from the production stage to its implementation.

3.6.3.5 Testing, Validation, and Report

During the testing phase of the three existing Design Options, the developers have decided to minimize the production and development costs, utilize SolidWorks for the testing and validating of the first design constraints, the Performance, and apply a mathematical computation and evaluation for the third design constraints, the Power Consumption, guided and supervised by two individual Engineers from Mechanical Engineering Department and Electronics and Communications Engineering Department respectively. On the other hand, during the testing and validation phase for the second design constraint, the response time was done in actual testing as it cannot be simulated virtually.

3.6.3.5.1 Performance

During the testing and validation of Design Option 1's performance, the developers utilized SolidWorks to compute and identify how efficiently the design could run to achieve the desired temperature range.

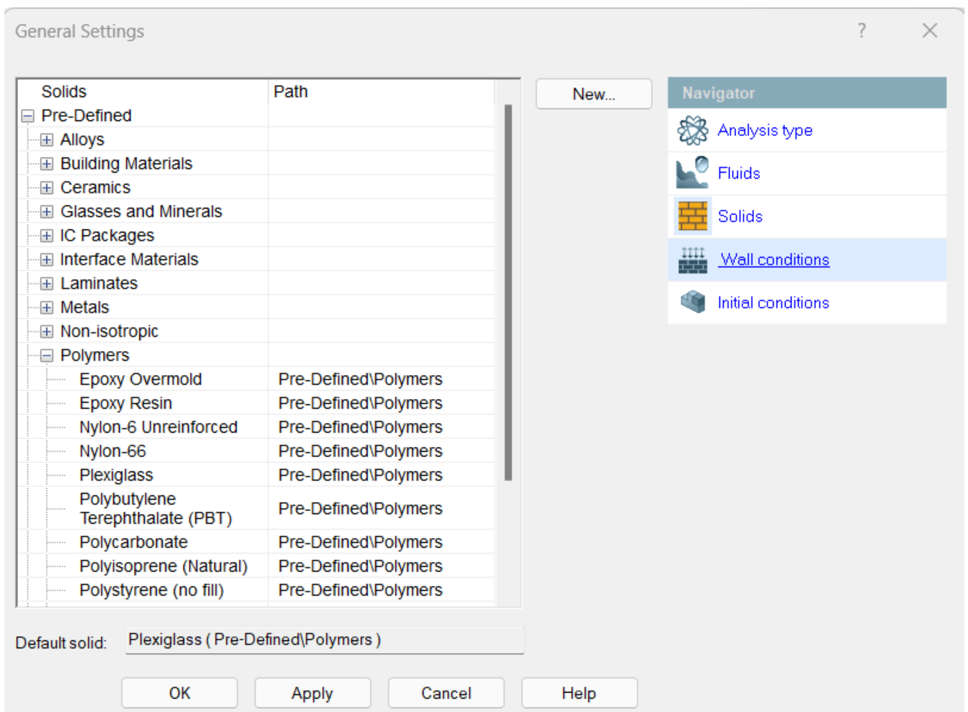


Figure 3-36. Design Option 1 Wall Conditions

Before conducting the flow simulation, determining the wall conditions was necessary. This can be done by going to "General Settings" and selecting "Wall Conditions", as displayed in Figure 3-36. Wall conditions determine how fluid interacts with the solid boundaries in the simulation. It also helps to represent real-world scenarios accurately. Different wall conditions have an impact on the flow behavior. In this case, under the Polymers, Plexiglass was chosen since Design Option 1 has a cover material of acrylic plastic.

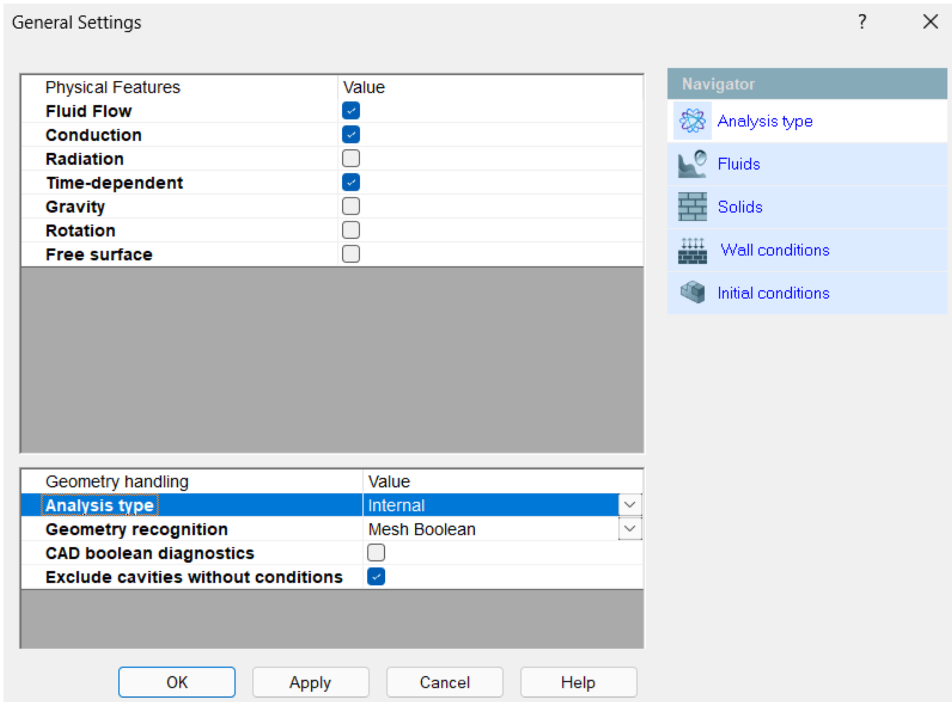


Figure 3-37. Design Option 1 Analysis Type

The second thing to consider is the Analysis Type, as presented in Figure 3-37. Analysis type determines how the simulation models the interaction between the fluid domain and the surrounding environment. In Design Option 1, the internal analysis type focuses on the enclosed fluid space. Internal analysis allows for more precise modeling of fluid flow within enclosures. This flexibility allows for accurate simulation of real-world scenarios.

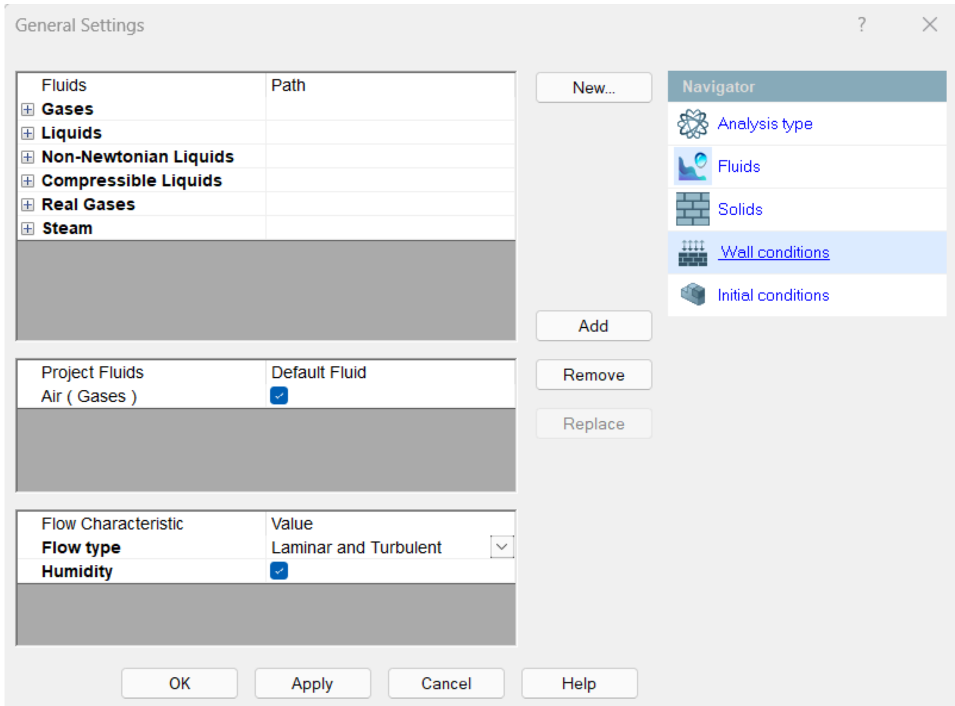


Figure 3-38. Design Option 1 Fluids

In addition to this, still under the "General Settings", the next thing to determine is the "Fluids". Figure 3-38 shows Design Option 1 Fluids where Air (Gases), Flow type, and Humidity. Air, in SolidWorks, is typically treated as an ideal gas. The properties of air can be set based on various factors such as temperature, pressure, and humidity. Flow type, on the other hand, determines how the fluid behaves during simulation. In this design option, it is "Laminar and Turbulent". Laminar refers to smooth, orderly flow with minimal turbulence. While Turbulent is chaotic, irregular flow with significant mixing between layers. Lastly, Humidity. Humidity also affects the thermodynamic properties of air. Where if the air is dry, there is no water vapor present; when air is saturated, it is the maximum amount of water vapor at a given temperature, and when partially saturated, it varies between dry and saturated states.

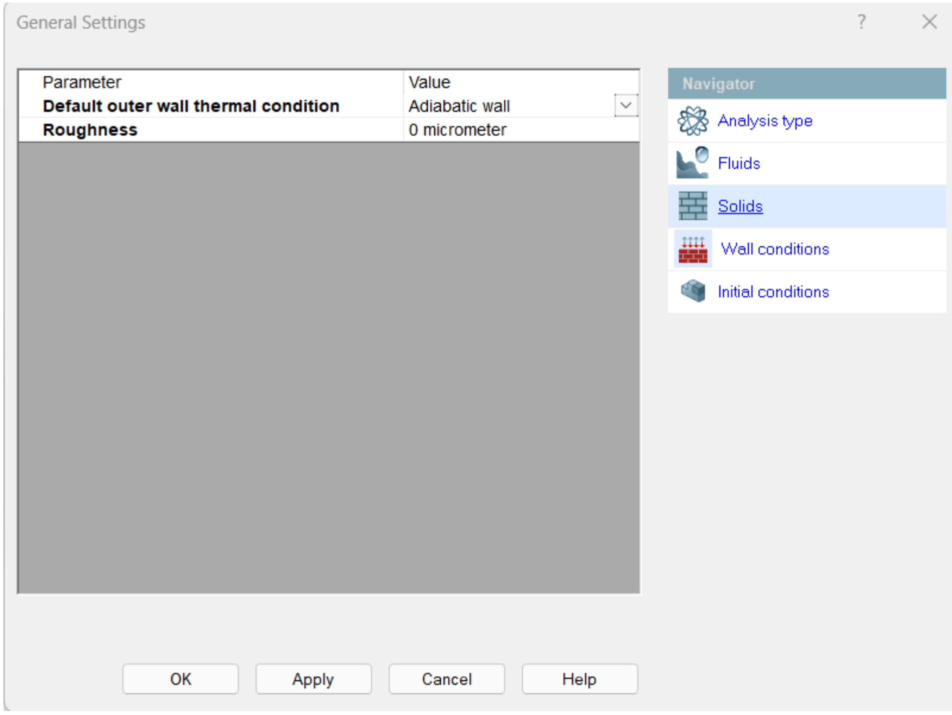


Figure 3-39. Design Option 1 Solids

Lastly, under the "General Settings" was the "Solids." In this case, the outer wall thermal condition is set to the default, Adiabatic wall, and the roughness is set to 0 micrometer (*see Figure 3-39*).

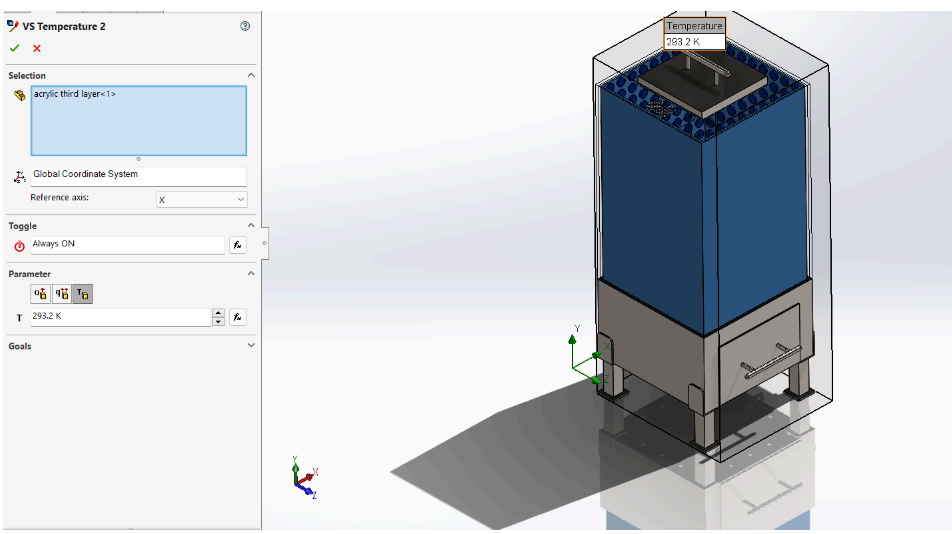


Figure 3-40. Design Option 1 Volume Source Temperature

The "Volume Source Temperature" specifies the temperature of a heat source in a computational fluid dynamics (CFD) simulation. It allows defining a region of the model where heat is added to the fluid flow. As depicted in Figure 3-40, the volume source temperature of Design Option 1 is inside and within the overall cover material.

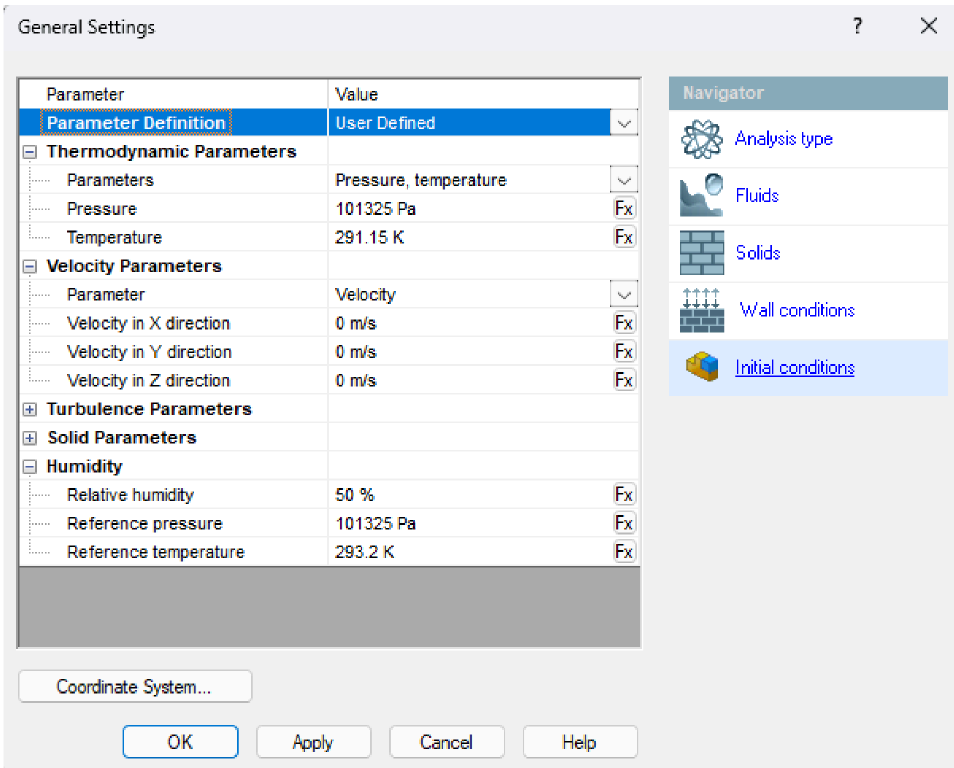


Figure 3-41. Design Option 1 Initial Conditions

Figure 3-41 shows how to determine the initial conditions for this design option. The parameters were set "Pressure, Temperature." Pressure has an initial value of 101325 Pa, and Temperature is 291.15 K. Under "Humidity," relative humidity is 50%, Reference pressure is 101325 Pa, and Reference temperature is 293.2 K.

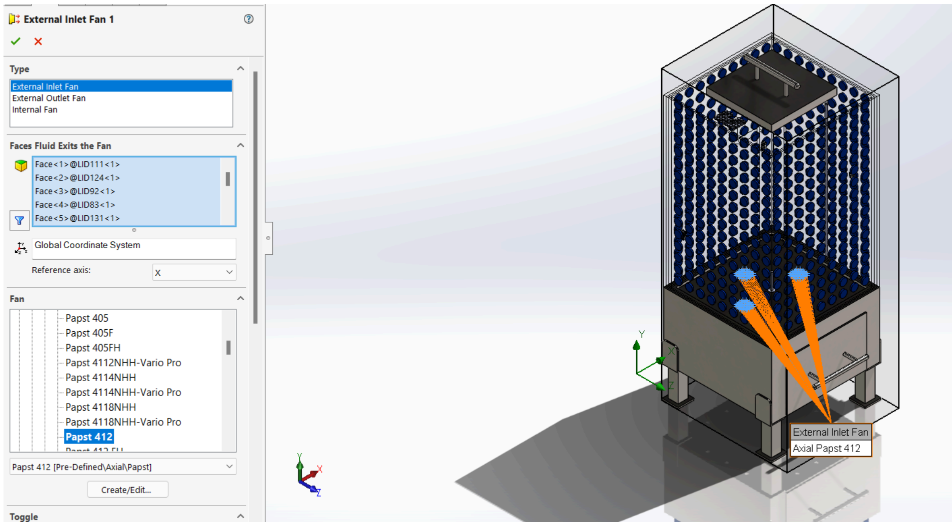


Figure 3-42. Design Option 1 Type of Fan

Additionally, the Type of Fan was also considered. This was one of the factors determining how fast the design would meet the criteria, which was set in the Initial Condition. Design Option 1 has an External Inlet Fan of "Papst 412," the fan type in SolidWorks for Thermoelectric Peltier (*see Figure 3-42*).

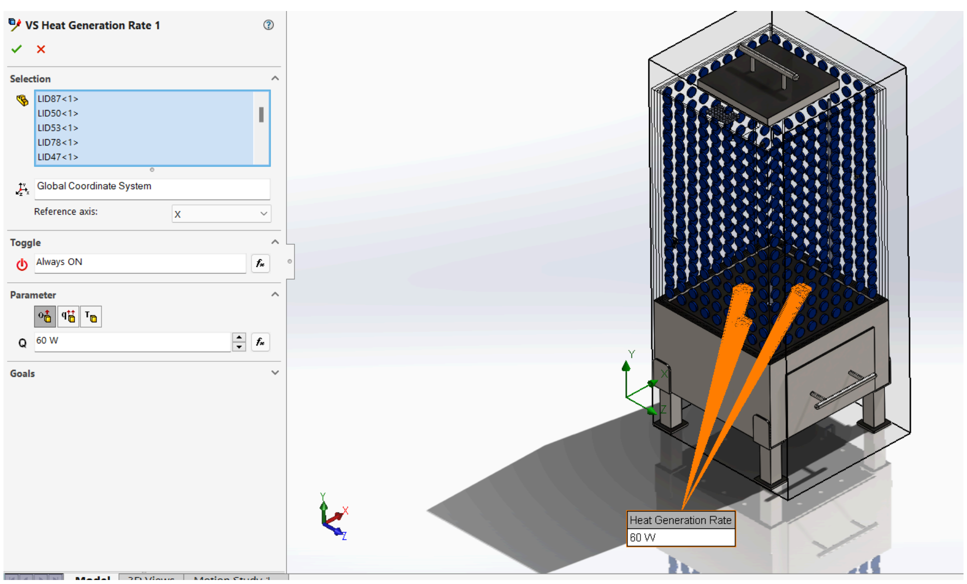


Figure 3-43: Design Option 1 Heat Generation Rate

Once the type of fan is set, it is now required to set its heat generation rate, as shown in Figure 3-43. Heat Generation Rate refers to the amount of heat energy produced per unit volume over a given period of time, which is usually expressed in units like Watts per cubic meter (W/m^3). In this Design Option, the fan's heat generation rate was 60W, the maximum power of a Thermoelectric Peltier.

Once completed, the flow simulation can start. During this simulation, it is important to observe closely to see if the design is doable and meets the requirements for maintaining the temperature and humidity inside.

Once done, the results can be seen and saved in an Excel file. In this simulation, the developers had to conduct several minor and major adjustments for this option to achieve the goal.

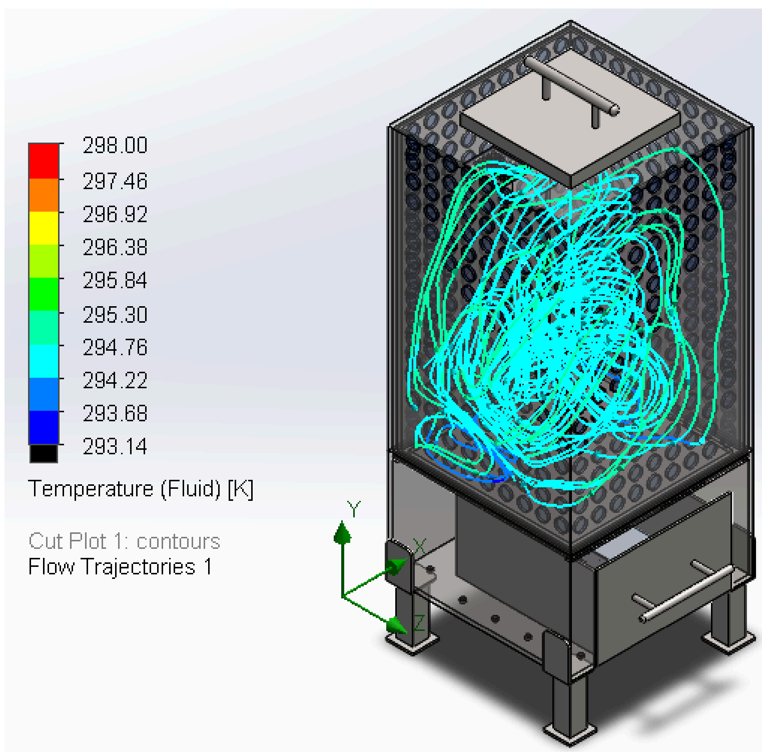


Figure 3-44. Design Option 1 Flow Trajectories Result

Flow Trajectories in Solidworks visualize and analyze fluid flow patterns within a simulated environment (Borchert, 2022). These trajectories allow developers to create animations that demonstrate how fluids move through a system or around objects over time.

As illustrated in Figure 3-44, is the flow trajectories of the Design Option 1. The colored streamlines, varying from deep blue to bright cyan, show how the fluid moves and mixes inside the storage. The colors correspond to different temperature levels, ranging from 293.14 K to 298 K, as indicated by the color scale on the left. This range, represented by a gradient from blue, a cooler temperature, to read, warmer temperature, highlights the thermal variations within the fluid.

The intricate, looping paths of the streamlines suggest turbulent flow, where the fluid continuously circulates, creating a pattern of motion that allows for efficient heat distribution throughout the space.

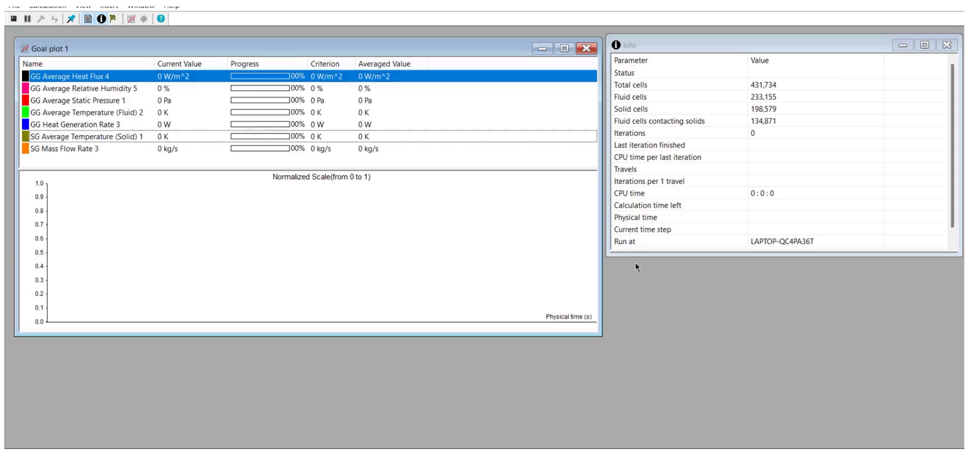


Figure 3-45. Design Option 1 Flow Simulation Start

Figure 3-45 shows the initial setup screen of a computational fluid dynamics (CFD) simulation, which is likely used to analyze airflow and temperature distribution for the dotBean Design Option 1. The left panel provides details on various monitored parameters, such as "Average Relative Humidity," "Average Temperature," and "Mass Flow Rate," among others. Each parameter has columns showing its "Current Value," "Progress," "Criterion," and "Averaged Value," all of which are currently set to 0%, indicating the simulation has just begun or is yet to run.

A list of "Parameter" and "Value" pairs on the right panel provides further details about the simulation settings. These include the number of cells being used for the simulation ("Total cells: 471,734"), the number of solid cells, fluid cells, and wall cells, and other specifics such as "Solver iterations finished," which is currently set to 0, suggesting that the solver has not yet performed any iterations. The "Run" section lists the name of the simulation machine, indicating that this is the starting point for the computational analysis.

By initializing the simulation with these parameters, the software is prepared to calculate the complex interactions of air movement, temperature distribution, and humidity levels, which are critical for optimizing the design to maintain the desired environment for storing green coffee beans. The goal is to ensure the storage conditions are controlled and consistent, preserving the beans' quality over time.

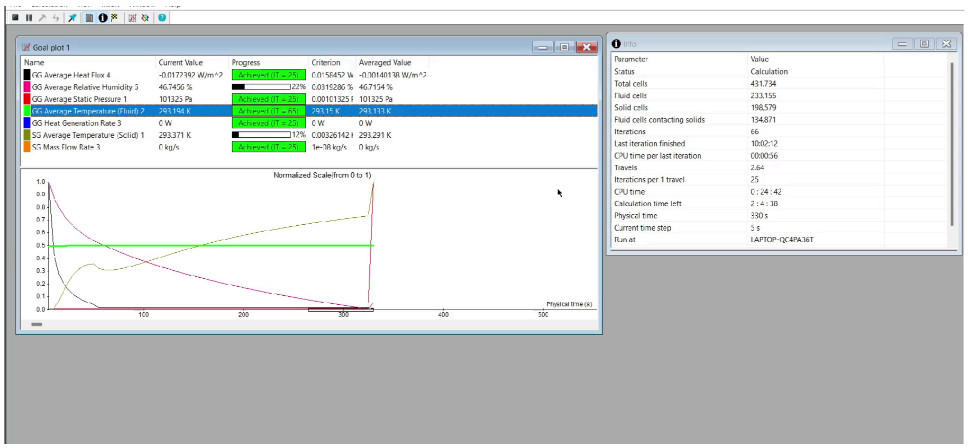


Figure 3-46. Design Option 1 Flow Simulation on Progress

Figure 3-46 presents the ongoing CFD simulation of Design Option 1, indicating that the simulation was actively calculating the airflow and thermal behavior within the storage.

In the left panel, the parameters being monitored, such as "Average Heat Flux," "Average Relative Humidity," "Average Temperature," and "Heat Generation Rate," are shown to have non-zero values. This means the simulation is now actively running and generating results. The "Progress" column shows green bars, signifying that the simulation is advancing toward its target criteria. For example, the "Average Heat Flux" has achieved 100% progress, reaching a value of approximately 0.03767879 W/m². The "Average Temperature" values, currently around 295.92 K, reflect the temperature distribution the system is calculating over time.

The plot below the parameters shows graphs representing various metrics against normalized steps or iterations of the simulation. These curves illustrate how parameters such as temperature, humidity, or heat flux change over time or iterations, revealing trends in how the system stabilizes or reaches equilibrium.

The right panel provides additional status updates, such as the number of iterations completed and the current timestep. It also shows information like the total number of cells, solid, fluid, and wall cells, as well as calculation times, which are crucial for understanding the computational effort involved.

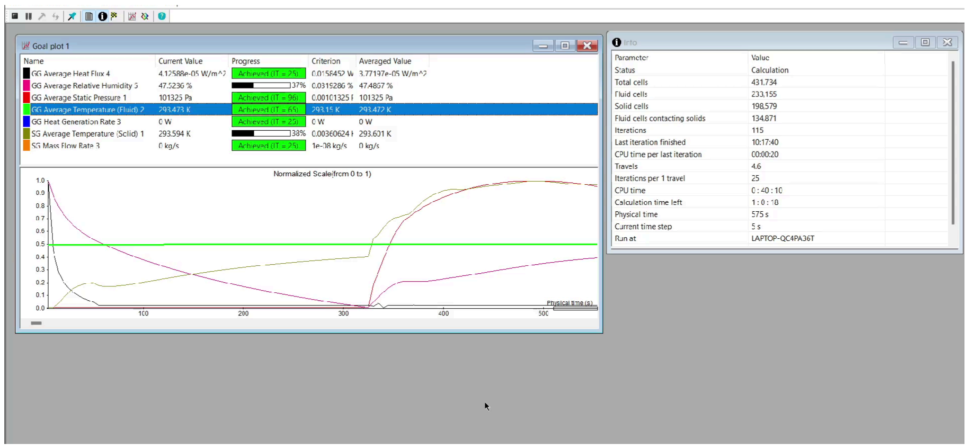


Figure 3-47. Design Option 1 Flow Simulation Finished

The CFD of Design Option 1, as shown in Figure 5-47, has now successfully finished, as evidenced by several critical indicators. The progress bars for parameters Heat Flux, Static Pressure, Temperature Fluid, Heat Generation Rate, and Mass Flow Rate have reached 100%, confirming that the simulation has met the predefined criteria. The values for these parameters have stabilized, showing no further fluctuations, which indicates that the system has achieved thermal equilibrium. Additionally, the graphs depicting changes over time reveal a convergence toward steady-state conditions, and the simulation status on the right panel shows that all necessary iterations have been completed.

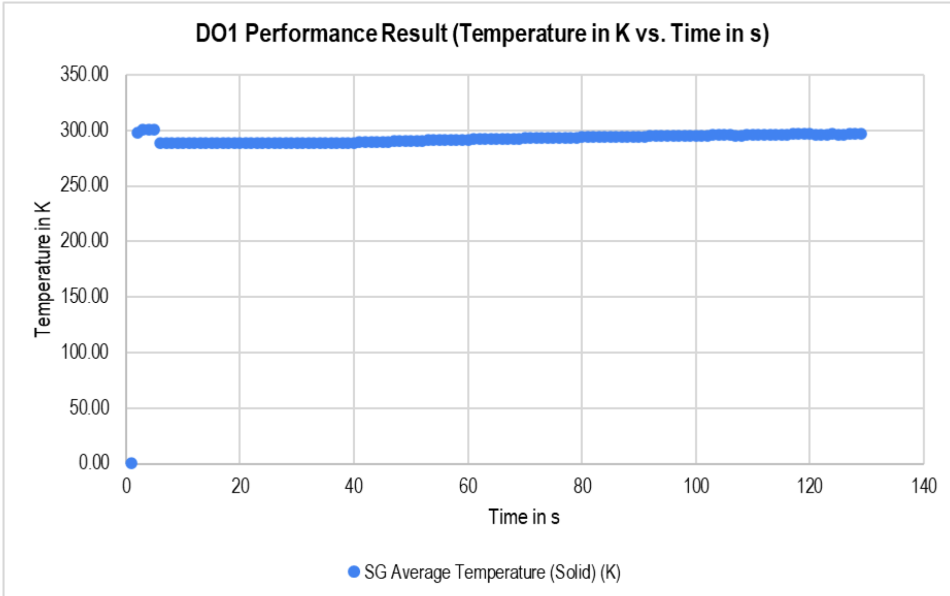


Figure 3-48. Design Option 1 Performance Result (Temperature in K vs. Time in s)

Figure 3-48 illustrates the result of a flow simulation of design option 1 in SolidWorks. Over time, the system's temperature increased from 288.15K to 293.15K; changes occurred during the simulation. Overall, it took 21 minutes and 5 seconds to complete the simulation, reaching 20°C from 15°C.

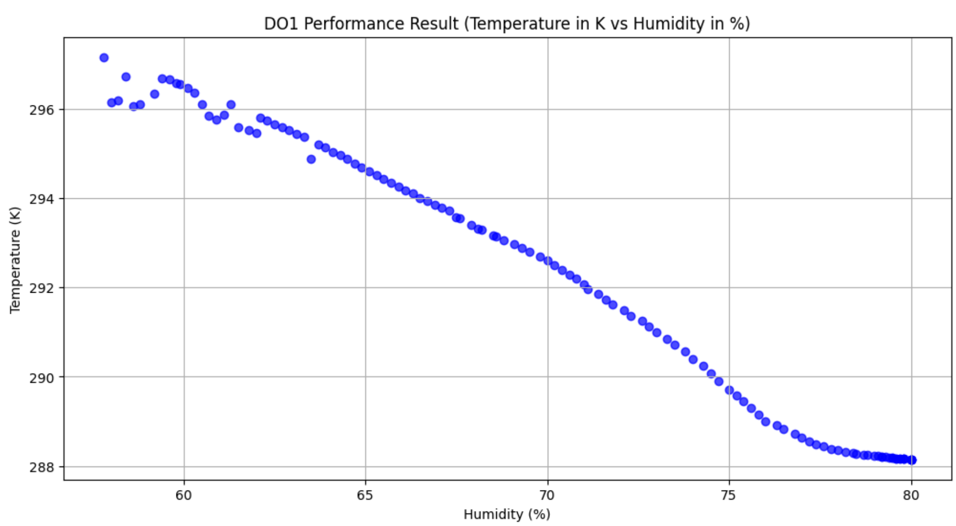


Figure 3-49. Design Option 1 Performance Result (Temperature in K vs. Humidity in %)

Furthermore, the figure above, on the other hand, shows that as the temperature changes, the humidity also changes. It indicates that as the temperature increases, the humidity decreases, reaching the optimal range between 50% and 70%, from around 80% to just below 60%.

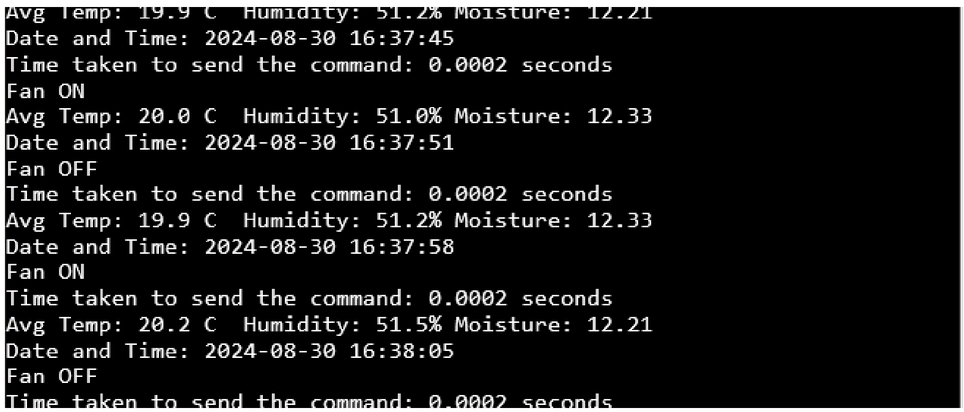
3.6.3.5.2 Response Time

The fan's response time to commands from the Raspberry Pi was evaluated to ensure it turned on or off as needed. With the system operating automatically, verifying its reliability without requiring user intervention was crucial. Measuring these response times confirmed that the fan adjusts swiftly to temperature changes, maintaining the desired conditions efficiently.

```
# Record the start time
start_time = time.time()
fan_status = "OFF" if GPIO.input(RELAY_PIN) == GPIO.LOW else "ON"

# Fan control logic
if avg_temperature_c < 20:  # Condition for turning the fan on
print("Fan ON")
GPIO.output((RELAY_PIN, RELAY_PIN1), GPIO.HIGH)  # Turn the fan on
Else:  # Default case where the fan should be off
print("Fan OFF")
GPIO.output((RELAY_PIN, RELAY_PIN1), GPIO.LOW)  # Turn the fan off

 # Record the end time
 end_time = time.time()
# Print the averaged values to the console
current_datetime = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
message = (f"Avg Temp: {avg_temperature_c:.1f} C  Humidity:
{avg_humidity:.1f}% Moisture: {line}\nDate and Time: {current_datetime}")
print(message)

# Calculate the duration in seconds with millisecond precision
duration = end_time - start_time
print(f"Time taken to send the command: {duration:.4f} seconds")
```

Figure 3-50. Response Time Command in Raspberry Pi

Figure 3-50 is designed to control a fan based on temperature readings and to log relevant information about its operation. Initially, it records the start time to measure how long the variable operates. It then checks the current status of the fan using a GPIO pin, setting a variable to indicate whether the fan is “ON” or “OFF”. The core of the code controls the fan: if the temperature is below 20, the fan is turned On, and it is Off. Lastly, the code records the end time, calculates the duration of the fan control operation with millisecond precision, and prints the average temperature, humidity, moisture levels, and the current date and time, along with the duration taken to execute the fan control logic.



```
Avg Temp: 19.9 C  Humidity: 51.2% Moisture: 12.21
Date and Time: 2024-08-30 16:37:45
Time taken to send the command: 0.0002 seconds
Fan ON
Avg Temp: 20.0 C  Humidity: 51.0% Moisture: 12.33
Date and Time: 2024-08-30 16:37:51
Fan OFF
Time taken to send the command: 0.0002 seconds
Avg Temp: 19.9 C  Humidity: 51.2% Moisture: 12.33
Date and Time: 2024-08-30 16:37:58
Fan ON
Time taken to send the command: 0.0002 seconds
Avg Temp: 20.2 C  Humidity: 51.5% Moisture: 12.21
Date and Time: 2024-08-30 16:38:05
Fan OFF
Time taken to send the command: 0.0002 seconds
```

Figure 3-51. Design Option 1 Raspberry Pi Sample Terminal Output Display

The system readings and fan status were displayed during the testing in the Raspberry Pi terminal. On the other hand, as shown in Figure 3-51, it took 0.0002 seconds for the Raspberry Pi to command the fan to turn off and turn on. The fan turned on at 19.9°C, turned off at 20°C, then turned on again at 19.9°C

Table 3-13. Design Option 1 Response Time Testing Results in microseconds

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry pi Received the Reading in microseconds (µs)*	*Time the Fan Activated in microseconds (µs)*	*Response Time in microseconds (µs)*
1	19.2	On	100	300	200
2	19.7	On	100	300	200
3	20.1	Off	200	300	100
4	20.2	Off	100	300	200
5	20.2	Off	100	300	200
6	20.2	Off	200	400	200

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry pi Received the Reading in microseconds (μs)*	*Time the Fan Activated in microseconds (μs)*	*Response Time in microseconds (μs)*
7	20.2	Off	200	500	300
8	20.2	Off	100	400	300
9	19.9	On	100	400	300
10	20	Off	200	300	100
11	20	Off	100	200	100
12	20	Off	100	300	200
13	20	Off	100	300	200
14	19.7	On	200	300	100
15	20.2	Off	100	300	200
16	20.2	Off	200	400	200
17	20.2	Off	200	500	300
18	20.2	Off	200	500	300
19	20.2	Off	200	400	200
20	19.9	On	400	500	100
21	20	Off	100	300	200
22	20	Off	100	300	200
23	20	Off	100	300	200
24	20	Off	200	300	100
25	20	Off	100	200	100
26	19.9	On	100	300	200
27	20	Off	100	400	300
28	19.9	On	100	300	200
29	20.2	Off	100	300	200
30	20.1	Off	200	500	300
31	20.1	Off	100	300	200
32	20.1	Off	200	400	200
33	19.8	On	200	400	200
34	20.1	Off	200	400	200
35	20.1	Off	200	300	100

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry pi Received the Reading in microseconds (µs)*	*Time the Fan Activated in microseconds (µs)*	*Response Time in microseconds (µs)*
36	20.1	Off	100	300	200
37	20.2	Off	100	300	200
38	20.2	Off	100	200	100
39	19.9	On	300	500	200
40	20	Off	100	300	200
41	20	Off	100	300	200
42	20	Off	200	300	100
43	20	Off	300	500	200
44	20	Off	200	400	200
45	19.9	On	200	400	200
46	20	Off	200	400	200
47	20	Off	100	300	200
48	20	Off	100	300	200
49	20	Off	100	300	200
50	20.1	Off	100	300	200
Average Response Time					**194**

Table 3-13 shows the response time testing result of design option 1. The tests were done by getting the time the Raspberry Pi receives the reading and the time the fan is activated in microseconds. The developers did 50 tests to get the average response time, and based on the response time testing results, it had an average of 194 microseconds.

3.6.3.5.3 Power Consumption

For a more straightforward approach, finding how much a design consumes power was done using a mathematical computation. To achieve this, the developers identified each component's Voltage (V) and Current (I) and multiplied their values to find the components' Power (P) in Watts (W). Each component's dataset, referenced from different online marketplaces and articles, was used to complete this computation.

According to (Makerlab Electronics, n.d.-c), Raspberry Pi 4B 8GB has an operating voltage of 5V at 3A. Arduino Uno R3, on the other hand, has a voltage of 5V at 0.02A (Arduino Tips, Tricks, and Techniques, 2024). Additionally, the Resistive Soil Moisture Sensor has a voltage of 5V and a current of 0.02A (“Soil Moisture Sensor Module,” 2020). Moreover, the DHT22 Module has a voltage and current of 3.3V and 0.0025 mA, respectively (LME Editorial Staff, 2022). Grove Red LED, as mentioned in Exploring Arduino (2021), has a voltage of 3.3V at 0.0015A current, while the Thermoelectric Peltier has a 12V voltage and a current of 5.8 (Makerlab Electronics, n.d.-d). On the other hand, a Switching Power Supply Unit has a voltage of 12V and 50A (Makerlab Electronics, n.d.-e). Furthermore, the L298N Driver module for the agitator has a voltage of 5V at a current of 0.003A (L298N Motor Driver Module - Motor Drivers - Motor and Wheels, n.d.). Lastly, a 4-channel 12V-5V relay module has a voltage of 5V and a current of 10A (Makerlab Electronics, n.d.-a), while a Momentary Push Button, as stated by Smoot (2022), has a voltage of 5V and a 1A current.

Table 3-14. Design Option 1 Power Consumption in Wattage (W) and Kilowatt-hour (kWh) Result

COMPONENTS	Qty.	V	I	P (in W)	kWh (1-hr)
Idle State					
Raspberry Pi 4B 8GB	1	5	3	15	0.0150
Arduino Uno R3	1	5	0.02	0.1000	0.0001
Resistive Soil Moisture Sensor	1	5	0.03	0.1500	0.0002
DHT22 Module	3	3.3	0.0015	0.0149	0.0000149
Grove Red LED	1	5	0.001	0.0050	0.0000050
Active State					
Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit	3	12	5.8	208.80	0.2088
L298N Driver Module	1	5	0.003	0.0150	0.000015
4-channel 5V Relay Module	2	5	10	100	0.1000

Each item has its value in volts and amperes. These values are used to calculate the power in watts in each item, as shown in Table 3-14. The Raspberry Pi consumes 0.015 kWh, the Arduino uno consumes 0.0001 kWh, the resistive soil moisture consumes 0.0002 kWh, DHT22 Module with the quantity of 3 consumes 0.0000149 kWh., grove red LED consumes 0.0000050 kWh, the thermoelectric Peltier with the quantity of 3 has a maximum wattage of 0.2088 kWh, L298N driver module consumes 0.000015 kWh, and lastly 4-channel 5V Relay Module consumes 0.100 kWh.

The developers have excluded the computation of the momentary push button because it does not consume significant power, with the electrical current involved being minimal—usually in the microampere range—since it is only used to detect state changes. The power supply is also excluded, as it does not add any extra load; it supplies power to the connected devices. This means the power consumed by the power supply from the outlet is simply the total of the power used by all connected devices, plus the minimal inefficiency caused by heat, LED, and other factors inherent in the power supply.

Table 3-15. Design Option 1 Power Consumption Result

Power Consumption	**State**	
	Idle (at use)	**Active**
Total Power (W)	15.2699	324.0849
Total Kilowatt-hour (kWh) (1 hr)	0.0153	0.3241

Table 3-15 summarizes the power consumption for the device Design Option 1, categorized into idle and active. In the idle state, the device consumes 15.2699 watts of power, reflecting the energy it requires while not actively performing tasks but remaining on standby. During active use, the power consumption increases to 324.0849 watts, demonstrating the additional energy needed to perform tasks. Power consumption over one hour is measured in kilowatt-hours (kWh), with the device using 0.0153 kWh in idle mode and 0.3241 kWh in active mode.

3.6.3.5.4 Public Health and Safety

The first design option utilizes acrylic plastic as the outer and inner layers, sandwiching a layer of polyurethane foam and adding desiccant gel for moisture absorption. Acrylic plastic is renowned for its durability, transparency, and resistance to weathering, making it a reliable choice for food storage applications. When manufactured to meet specific standards, such as those established by the FDA, acrylic plastic is considered food-safe and does not leach harmful substances into stored products (Kawasaki, 2019). Polyurethane foam is an effective thermal insulator due to its low thermal conductivity, making it suitable for maintaining stable temperatures within the storage environment, which is critical for preserving the quality of coffee beans (Smith et al., 2022). Additionally, desiccant gels, typically silica gel, are highly effective in absorbing environmental moisture. This is crucial for preventing mold growth and maintaining low humidity around the hygroscopic coffee beans (Chen et al., 2021).

3.6.3.5.5 Social Welfare

For Design Option 1, the developers create a storage device that accurately monitors the temperature and humidity inside the device, ensuring that temperature is maintained within 20° C - 25.4° C and the humidity at 50% - 70%. The moisture content of the green coffee beans should also be within the specific range of 11% to 12.5%, not exceeding 14%. The flow simulation suggests that it can take 21 minutes and 5 seconds to complete the simulation, reaching 20°C from 15°C. For the response time, the developers determined the average response time of this design option by calculating how long the system reacts when there are changes in temperature reading. After conducting 50 tests, the concluded average response time was 194 microseconds. For the power consumption, based on the calculated power for the idle (at use) and active state, the result for power consumption has a total Kilowatt-hour (kWh) of 0.0153 for the idle state and 0.3241 for the active state.

3.6.3.5.6 Economics

The developers conducted a thorough analysis to address the economic considerations of developing a system for monitoring and maintaining the moisture content of green coffee beans.

They prepared a comprehensive Bill of Materials (BOM) within a budget constraint of 50,000 PHP for production and development (*see Table 3-12*). This process involved carefully selecting components readily available in the Philippines, ensuring cost-effectiveness, and minimizing logistical challenges associated with sourcing materials in regions like La Trinidad Benguet. By focusing on locally available components such as sensors, microcontrollers, power supplies, and enclosure materials, the developers ensured that all parts could be acquired from local suppliers, thus avoiding additional import costs and potential delays.

This approach supports the local economy while ensuring the system is practical for users. The developers addressed affordability and material procurement challenges by staying within budget and sourcing components locally, ensuring the project meets its financial constraints.

3.6.3.5.7 Social

For the customer's selection, the developers designed a storage device that can be used by small-scale coffee farmers and coffee shop owners who currently use traditional storing processes in all regions experiencing lower temperatures. Coffee farmers have issues preventing the detrimental effects of excess moisture absorption on the green coffee beans that typically occur during low temperatures.

3.6.3.5.8 Global

For Design Option 1, the developers focused on creating a coffee bean storage system that is globally viable while adhering to time and budget constraints. The design incorporates a Raspberry Pi with Wi-Fi capabilities to notify clients via email, leveraging Raspberry Pi's pre-certification by the FCC to meet fundamental safety standards (Gibson, 2024). Email notifications were chosen for their simplicity and accessibility, as they require minimal technical expertise from users. The developers ensured that all components were locally sourced and replaceable in case of defects. Most components were purchased from Makerlab Electronics, which has a physical store in Manila and an online presence, allowing nationwide shipping.

Additionally, the system was designed with power consumption constraints in mind. Testing and validation revealed that the device is energy-efficient and consumes minimal power, which aligns with using energy-efficient components to reduce power consumption. This approach ensures compliance with global standards and makes the device practical and accessible to a broad audience.

3.6.3.5.9 Cultural

For Design Option 1, the developers ensured that the device integrates seamlessly into the existing coffee farming processes by focusing solely on the storage stage. The design aims to enhance the quality of stored coffee beans without altering pre-harvest or post-harvest procedures, respecting traditional farming practices. By providing a simple and user-friendly solution, this option ensures that farmers can quickly adopt the technology without needing to change their established methods. Using locally sourced materials and components makes maintenance straightforward, aligning the device with the local environment and resources and ensuring it is practical and sustainable for long-term use.

3.6.3.5.10 Environmental

Acrylic plastic is utilized for its properties as a food-safe and recyclable material. Acrylic plastic is known for undergoing processes that reduce toxicity, making it a suitable choice for food-related applications (Gilani et al., 2023). The design is optimized for low power consumption, as demonstrated in testing and validation (*see Table 3-14 and Table 3-15*), ensuring minimal energy usage during operation. Additionally, the system is designed with easily replaceable components, meaning that if one part becomes defective, it can be replaced without needing to fix or replace the entire device. This modularity helps minimize electronic waste and extends the device's lifespan.

3.6.3.6 Design Option 1 Constraint and Results Summary

After testing to validate the efficacy of the first design option, the table below summarizes the design constraints and their results, along with their remarks.

Table 3-16. Design Option 1 Constraints and Results Summary

Constraints	Results	Remarks
Performance	21 minutes and 05 seconds	Design option 1's total time to reach the minimum threshold value at 20°C was 21 minutes and 05 seconds, well below the criteria of within 30 minutes.
Response Time	194 microseconds	Design option 1's total average response time from the 50 testings was 194 microseconds or 0.000194 seconds, which achieved the criteria of less than 1 second.
Power Consumption	0.0153 kWh at idle state 0.3241 kWh at active state	Design option 1's power consumption when in the idle (at use) state and active state achieved the criteria of not consuming more than 1 kWh.

As depicted in Table 3-16, design option 1 achieved all the constraints with their perspective criteria. The performance of design option 1 suggested that the system can reach the optimal temperature when it drops below the threshold of 20°C. Moreover, design option 1 achieved a response time of 194 microseconds from the system reading the temperature to the system sending the command to the actuators, offering a reliable automated fan control system. Lastly, the first design option also achieved a low power consumption below 1kWh, achieving 0.0153 kWh when idle (at use state) and 0.3241 kWh when in active state.

Overall, design option 1 provides an efficient solution that meets all specified constraints, including performance, response time, and power consumption. Its ability to quickly reach the target temperature, respond rapidly to system changes, and maintain low energy usage makes it a reliable and energy-efficient choice for automated fan control in temperature-regulated environments.

3.6.4 Design Option 2

The design option 2 of the dotBean's main features consist of utilizing a Radiator Fan for temperature-and-humidity-related activities and an application of a dielectric method for measuring the moisture content of the coffee beans, which consists of a capacitive sensing mechanism through a sensor. To monitor and notify the user about the changes and status within the system during its operational time, the developers in this design option provide a mobile application with a database where all the sensor readings, actuator status, and other data will be stored. Below is a discussion of how this design option works and functions under dotBean's objectives and client requirements.

3.6.4.1 System Architecture

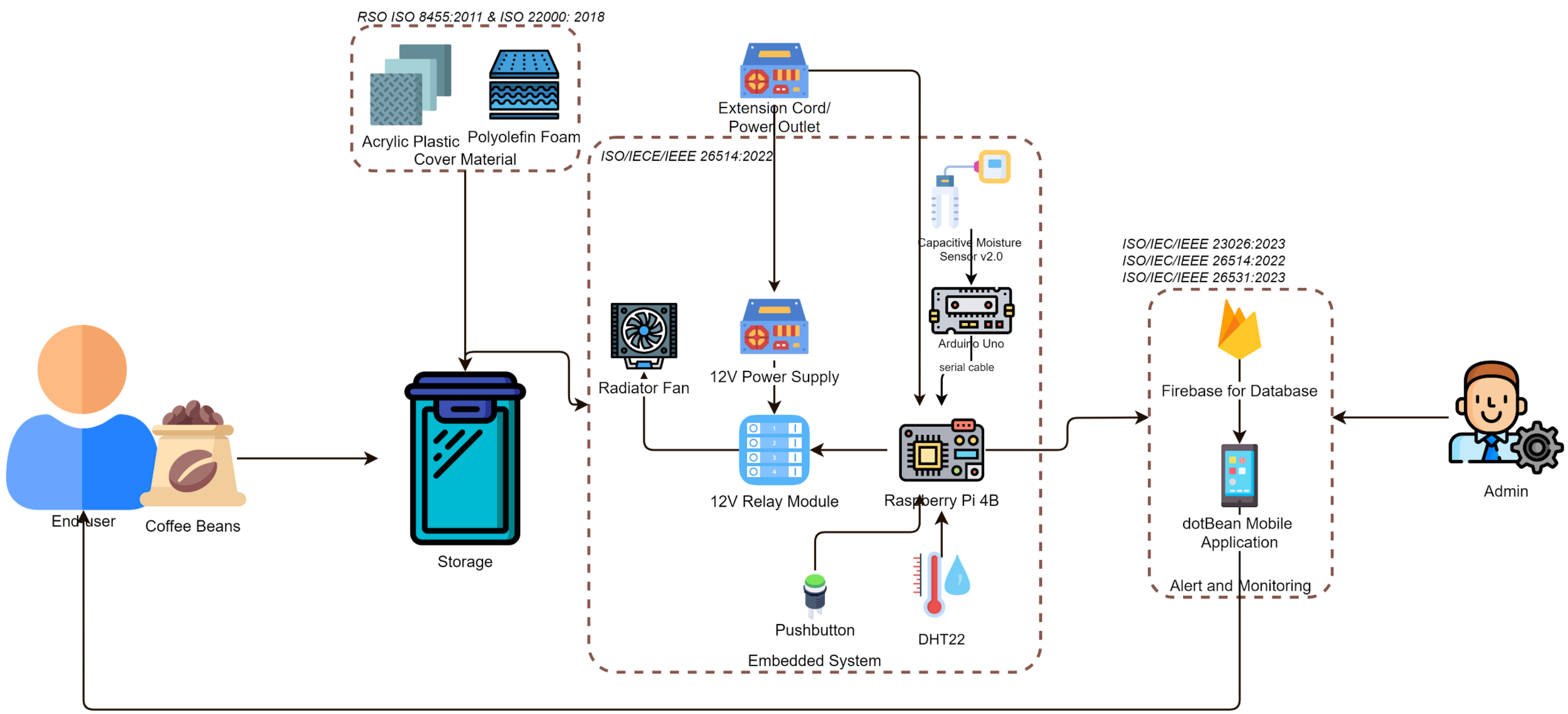


Figure 3-52. Design Option 2 System Architecture

Design Option 2 has the same components as the prior design option but only differs in the actuator and the moisture sensor. Design Option 2's cover materials also consist of acrylic plastic and foam. The following cover materials are effective moisture absorbents and help prevent the absorption of excessively low or excessively high temperatures from the outside. Actuator of this design option mainly feature a Radiator Fan dedicated to heating. Integrated sensors are DHT22 for temperature and humidity monitoring and a calibrated Capacitive soil moisture sensor v2.0. As its name suggests, the moisture sensor utilized in this design option is a capacitive type, which acts like a simple comparator to detect if there's water content in an object.

Consequently, these readings from two sensors are transmitted in the Raspberry Pi 4B for data analysis. This microcontroller also controls the actuator. Lastly, design option 2 offers a mobile application for monitoring and accessing the dotBean system.

As the dotBean mainly operates automatically using Raspberry Pi 4B, the data sent from the microcontroller to the mobile application are stored in the database. Using the Visual Studio Code IDE platform, these data are transformed to be visually presented in the mobile application.

3.6.4.2 Design Process Flow

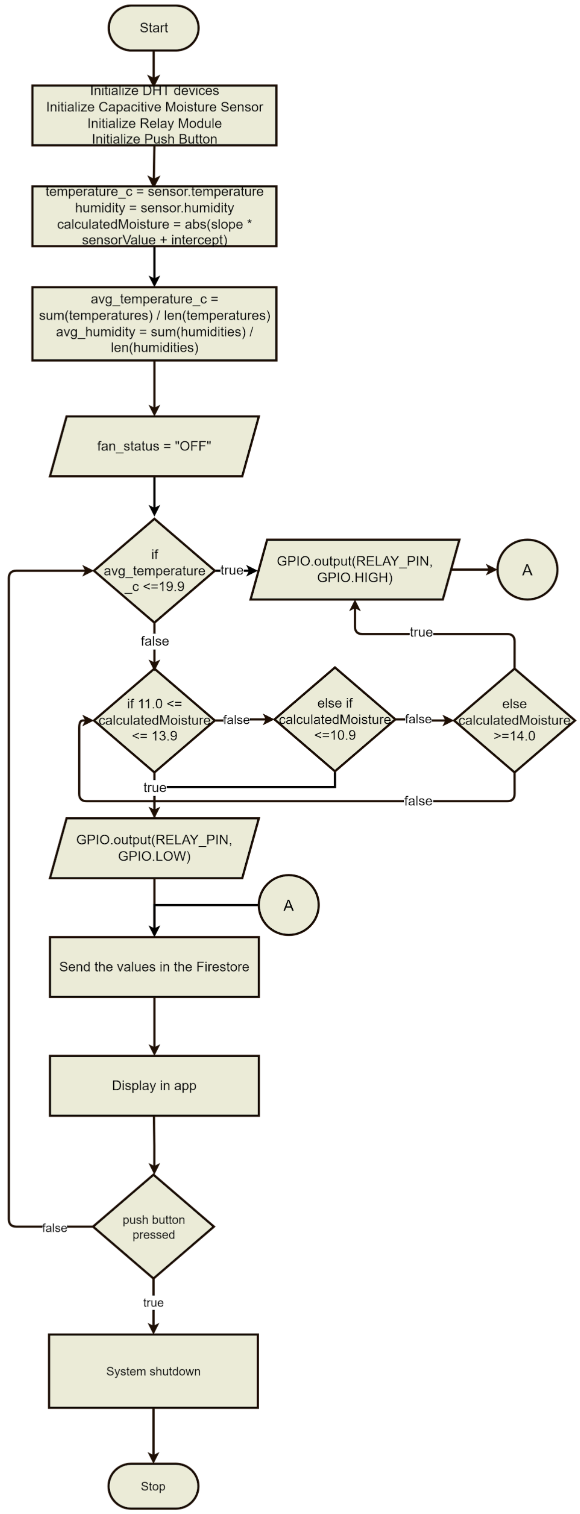


Figure 3-53. Design Option 2 Design Process

Having similar features to Design Option 1, the design process of Design Option 2, as illustrated in Figure 3-53, also involves retrieving data from sensors to Raspberry Pi and data analysis. This design option employs a capacitive method using the Capacitive Soil Moisture Sensor to measure the moisture content of the coffee beans. The moisture content is determined by when it's exposed to the coffee beans; it provides an electrical output that indicates the bean's moisture content. The value from the moisture sensor is converted into a percentage (%) using a specific computation.

Three DHT22 sensors are positioned inside the container to monitor temperature and humidity. This ensures a balanced environment, maintaining the right conditions for both moisture content and coffee quality. These readings and the system's status are then displayed on a mobile application.

3.6.4.3 Software Design

Unlike previous design options, Design Option 2's accessibility features a mobile application to allow end-users to monitor their dotBean. In this way, even if the clients are away from the storage facility, they can easily access and monitor the changes and conditions inside the dotBean in real time. This also includes real-time data from sensors and the current statuses of the actuators, which the Raspberry Pi processes. As previously mentioned, dotBean entirely operates automatically; the mobile application primarily presents the most recent data and statuses, aligning with predetermined parameters, although it includes a push notification feature for real-time monitoring. At times, the mobile application is not open on their phone. In the succeeding discussion, different software applications are listed and explained, where they are applied in the study, and the developers utilized it. Lastly, the figure illustrates and discusses software development and its process flow.

3.6.4.3.1 Software Technologies

SolidWorks for Prototype Layouting and Simulation

SolidWorks is known for its computer-aided design (CAD) and engineering (CAE) prowess. Offering a parametric modeling approach, the software empowers users to create 3D models by defining parameters and constraints, facilitating easy modifications. The developers utilized this software to lay out the prototype of each design option and conduct a simulation to attest and validate the design option’s capability to suffice the design constraints and design criteria. Overall, SolidWorks, with its parametric modeling, assembly features, simulation capabilities, and integration support, is a valuable tool in designing and developing computer engineering projects, providing a comprehensive solution for creating electronic systems with precision and efficiency.

Fritzing for Circuit and Schematic Diagram

Fritzing distinguishes itself by providing a user-friendly interface specifically tailored to accommodate individuals new to electronics and beginners in the field. Including drag-and-drop functionality, coupled with visual representations of components, streamlines the process of circuit creation, offering a simplified and intuitive design experience. With its breadboard view feature, the developers easily experiment with circuit designs in a simulated environment before transitioning to a tangible prototype.

Geany as Raspberry Pi IDE for Programming

Geany establishes itself as a robust code editor for Raspberry Pi, given its pre-installation in Raspberry Pi OS and its adeptness in facilitating Python or C/C++ coding. The presence of a built-in terminal enables developers to seamlessly compile and execute scripts directly within the Geany environment. Geany IDE is utilized for configuring sensors and actuators.

Android Studio for Developing the Mobile Application UI

Android Studio stands out as a comprehensive IDE for developing the UI of Android mobile applications. It offers visual design tools, resource management, code assistance, and integration with design guidelines to streamline the UI development process. Its features contribute to creating visually appealing, responsive, and accessible user interfaces for Android apps.

Visual Studio (VS) Code for Developing the Mobile Application UX and Connection with Raspberry Pi

VS Code provides a comprehensive set of tools for developing the mobile application's UX and establishing a connection with the Raspberry Pi. Developers use this software for its extensive language support, debugging capabilities, and integrated Git control.

Node.js for Mobile Application UX and Connection with Firebase Database

Node.js excels in real-time communication, making it suitable for applications requiring instantaneous updates. Its event-driven, non-blocking I/O model contributes to scalable and high-performing applications, especially in mobile app development, where rapid response times are crucial, particularly with numerous concurrent connections. Using JavaScript throughout the entire stack, employing Node.js on the server side and Firebase on the client side, unifies the language and data structure, streamlining both server and client-side scripting processes.

React Native for Mobile Application Front-End Rendering

React Native played a significant role in the project, providing a platform for the front-end rendering of our mobile application. The developers utilized this software to fetch data from the Firebase Realtime Database and render it in the mobile application. This involves creating a reference to the path where the data was written in the Firebase Database, reading the value at that path, and then using this data to render the components.

Firebase for Mobile and Website Application Database

The developers utilized the Firebase database to store and retrieve data in the mobile application. This involves writing data to the database using the setValue method and reading data from the database using the getValue method.

NPM for JavaScript Libraries

NPM offers a platform for managing JavaScript libraries. The developers utilized this for its ability to manage and share reusable code, which allows it to install, update, and manage dependencies for the project. This includes JavaScript libraries and frameworks that the developers can use to enhance their mobile application's functionality and user experience.

3.6.4.3.2 Software Development

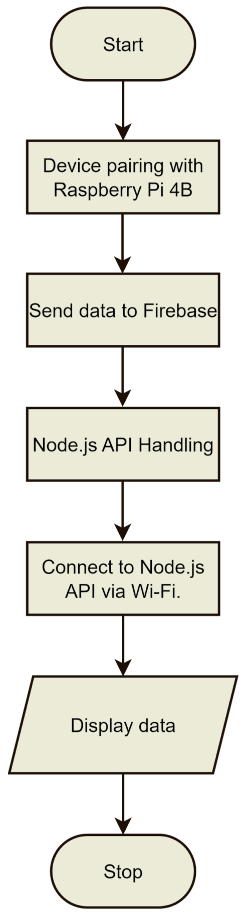


Figure 3-54. Design Option 2 Software Process

To begin the process, the Raspberry Pi establishes a connection to a Wi-Fi network, ensuring a stable wireless communication channel for subsequent operations. Following this, the sensors attached to the Raspberry Pi, namely the DHT22 and Capacitive Moisture Sensor v2.0, are initialized along with the necessary GPIO configurations. Once set up, the Raspberry Pi continuously retrieves data from these sensors, capturing both temperature and humidity readings from the DHT22 and moisture levels from the sensor. Based on the collected data, the system then determines the device's operational status. This entails making decisions regarding the radiator fan's functionality and the positioning of the vent. Once the statuses are specified, the Raspberry Pi uses its Wi-Fi connectivity to transmit this data to a Firebase database. This involves both uploading the raw sensor data and updating the operational status of the fan. Concurrently, the API fetches the latest data from Firebase and stands ready to handle requests from a mobile application. When a user interacts with the mobile application, it establishes a connection to the Node.js API via Wi-Fi. Subsequently, the mobile app fetches and displays the most recent sensor readings and device statuses to the user.

To establish a connection between the Firebase and the Mobile Application, export the following code from the dotBean database in Firebase.

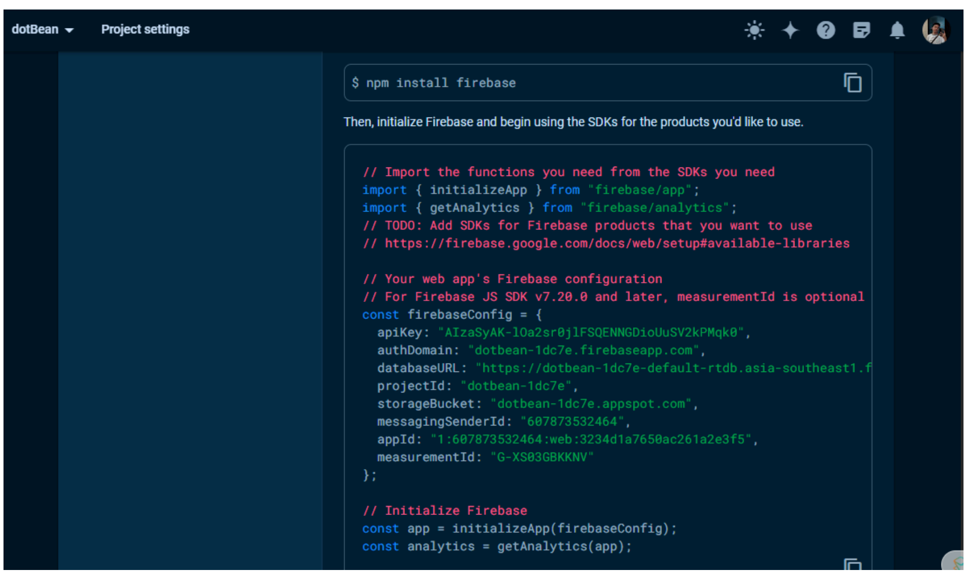


Figure 3-55. Firebase Configuration

Use the file, as shown in Figure 3-55, to create a Firebase configuration in a typescript file. Make sure that the filename exported from the Firebase is the same as the filename indicated in the command.

```
import { initializeApp } from "firebase/app";
import { getFirestore } from "@firebase/firestorm";
import { initializeAuth, getReactNativePersistence, getAuth } from
'firebase/auth';
import ReactNativeAsyncStorage from
'@react-native-async-storage/async-storage';

const firebaseConfig = {
  apiKey: "AIzaSyAK-lOa2sr0jlFSQENNGDioUuSV2kPMqk0",
  authDomain: "dot-bean-1dc7e.firebaseapp.com",
  database URL:
"https://dotbean-1dc7e-default-rtdb.asia-southeast1.firebasedatabase.app",
  projected: "dot bean-1dc7e",
  storageBucket: "dotbean-1dc7e.appspot.com",
  messagingSenderId: "607873532464",
  appId: "1:607873532464:web:3234d1a7650ac261a2e3f5",
  measurement: "G-XS03GBKKNV"
};
```

Figure 3-56. Firebase Configuration in dotBean User Mobile Application Code

Once initialized (*see Figure 3-56*), import the typescript in the homepage directory of the application as shown in Figure 3-57:

```
import React, { use effect, useState } from 'react';
import { View, Text, Image, TouchableOpacity, TextInput, Button } from
'react-native';
Import LinearGradient from 'react-native-linear-gradient';
import { snapshot } from 'firebase/firestore';
import { useNavigation } from '@react-navigation/native';
import Toast from 'react-native-toast-message';
import styles from '../../components/styles';
import { FireBase_DB } from '../../FireBaseConf';
import { doc } from 'firebase/firestore';
import { Shadow } from 'react-native-shadow-2';

const Homepage = () => {
 const [devicemoisture, setdevicemoisture] = useState(0);
 const [devicehumidity, setdevicehumidity] = useState(0);
 const [devicetemp, setdevicetemp] = useState(0);

 const navigation = useNavigation();

 useEffect(() => {
    const docRef = doc(FireBase_DB, 'sensors', 'dotbeancontainer1');
```

Figure 3-57. FireBaseConf.ts Import in homepage.js Snippet Code

On the other hand, establishing a connection between the Firebase and the site has no difference with the Mobile Application. To conduct this, import the functions from the SDKs found on the Firebase. Established a connection by inputting its APIKey, authDomain, projectId, storageBucket, messagingSenderId, appId, and measurementId (*see Figure 3-58*).

```
import { initializeApp } from "firebase/app";
import { initializeAuth, getAuth } from 'firebase/auth';
import { getFirestore } from "@firebase/firestore";

const firebaseConfig = {
  apiKey: "AIzaSyAK-lOa2sr0jlFSQENNGDioUuSV2kPMqk0",
  authDomain: "dot-bean-1dc7e.firebaseapp.com",
  projected: "dot-bean-1dc7e",
  storageBucket: "dotbean-1dc7e.appspot.com",
  messagingSenderId: "607873532464",
  appId: "1:607873532464:web:3234d1a7650ac261a2e3f5",
  measurement: "G-XS03GBKKNV"
};

// Initialize Firebase

export const FireBase_APP = initializeApp(firebaseConfig);
export const FireBase_AUTH = initializeAuth(FireBase_APP, {
});
export const FireBase_DB = getFirestore(FireBase_APP)
export const SignUp_AUTH = getAuth(FireBase_APP)
```

Figure 3-58. Firebase Configuration in dotBean Admin Website Application Code

Once done, import them, too, on the index page of the dotBean Admin Site, as shown in Figure 3-59:

```
import React, { useEffect, useState } from 'react';
import { collection, getDocs, doc, deleteDoc } from 'firebase/firestore';
import { FireBase_DB } from "../../../firebaseConf";
import Link from "next/link";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { Sidebar } from './sidebar';

export function AdminPage() {
  const [admins, setUsers] = useState([]);
  const fetchUsers= async () => {
    try {
      const querySnapshot = await getDocs(collection(FireBase_DB, 'admin'));
      const adminList = querySnapshot.docs.map(doc => ({ ...doc.data(), id:
doc.id }));
      setUsers(adminList);
    } catch (e) {
      console.error("Error getting documents: ", e);
    }
  };
```

Figure 3-59. fireBaseConf.ts Import

Below is user interface (UI) of the dotBean mobile (client-side) and website application (admin-side):

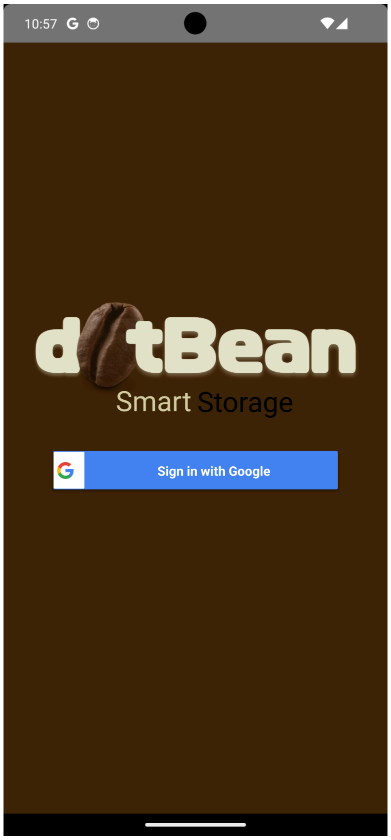


Figure 3-60. dotBean Mobile Application - Client Side: Login Page

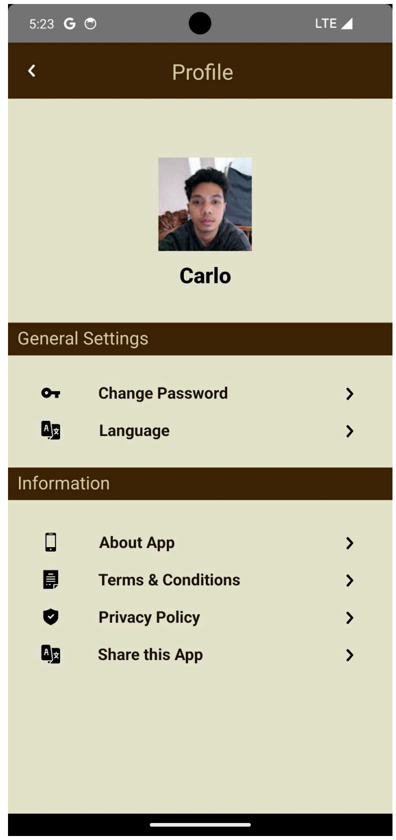


Figure 3-61. dotBean Mobile Application - Client Side: Application Settings Page

The login site ensures that only authorized users can access the data. Users must log in using their existing GMail account for more seamless signing in, as Figure 3-60 shows. Once logged in, the user can access general settings and has the option to change their password or preferred language. Additionally, in Figure 3-61, the user can view information about the app, terms and conditions, and privacy policy and share the application.

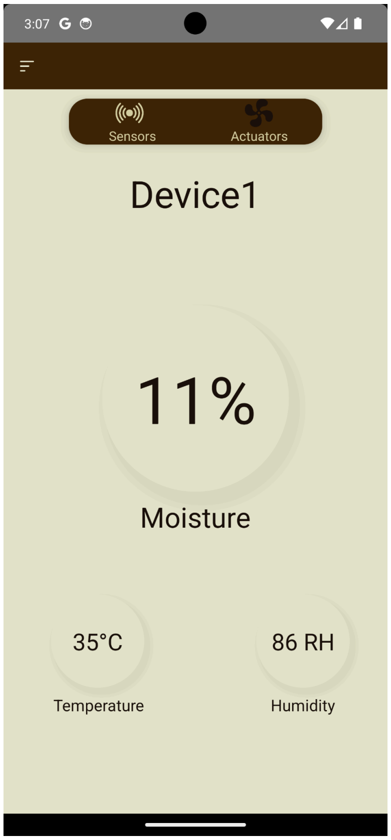


Figure 3-62. dotBean Mobile Application - Client Side: Sensors Tab

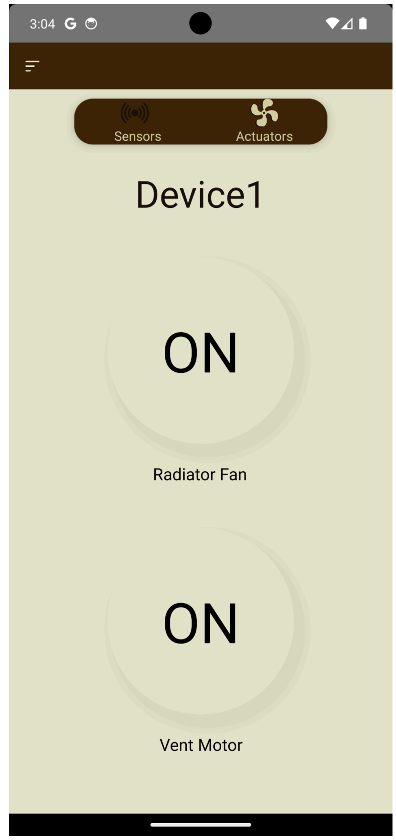


Figure 3-63. dotBean Mobile Application - Client Side: Dashboard - Actuators Tab

The dashboard shows the sensors and actuators; under the sensors tab, the user can view the percentage of moisture content, temperature, and humidity (*see Figure 3-62*). In the actuator tab, on the other hand, as depicted in Figure 3-63, the user can control the radiator fan and the vent motor.

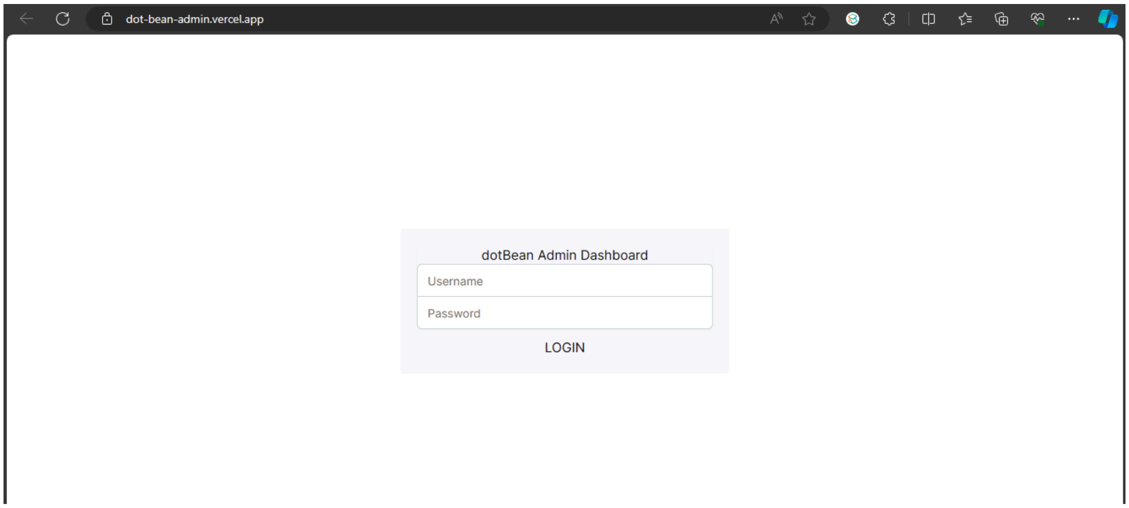


Figure 3-64. dotBean Website Application - Admin Side: Login

Figure 3-64 above shows the dotBean login page for admins. The admins will have to log in using their own username and password.

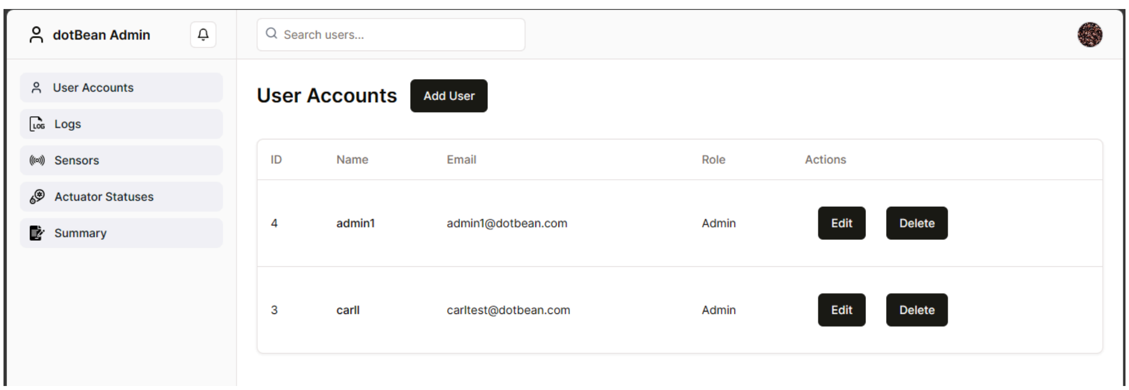


Figure 3-65. dotBean Website Application - Admin Side: Dashboard User Accounts Control Page

Figure 3-65 above shows the admin dashboard. The admin can view the users who logged in to the application, add, edit, and delete users, monitor the logs, sensors, and actuator statuses, and view the summary.

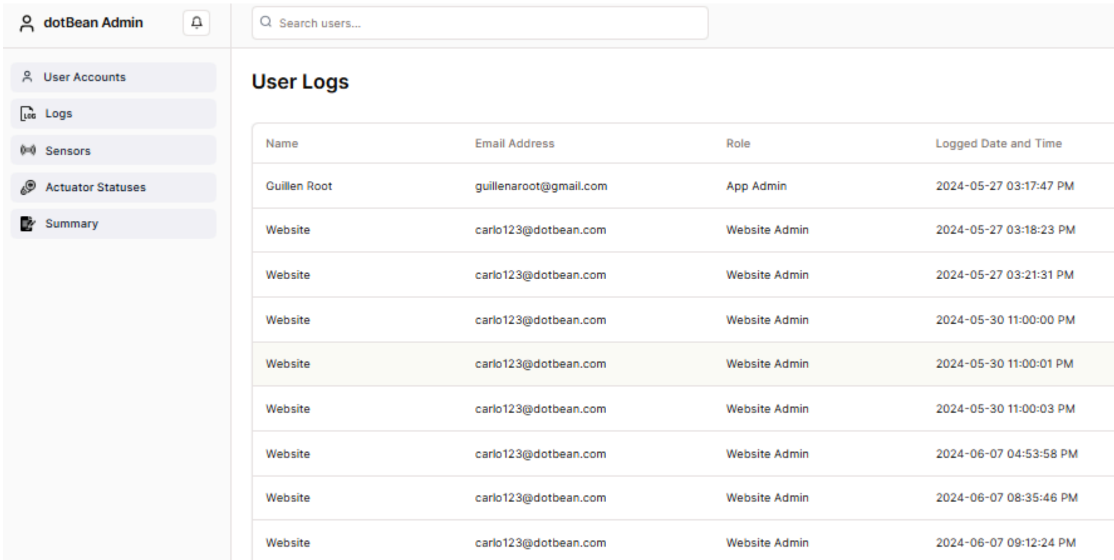


Figure 3-66. dotBean Website Application - Admin Side: Dashboard User Logs Page

Once logged in, the admin can access the users' accounts. The admin can monitor and see how many users logged into the app with their email addresses and also show the date and time of login (*see Figure 3-66*).

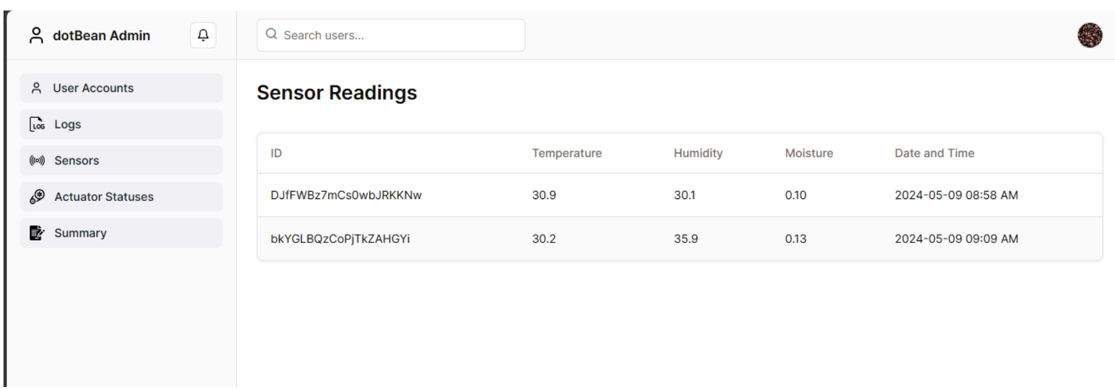


Figure 3-67. dotBean Website Application - Admin Side: Dashboard Sensors Page

The admin will have access to the dashboard page to see the sensor readings, such as temperature, humidity, and moisture, as shown in Figure 3-67. Each reading has its unique ID and shows the date and time it was recorded.

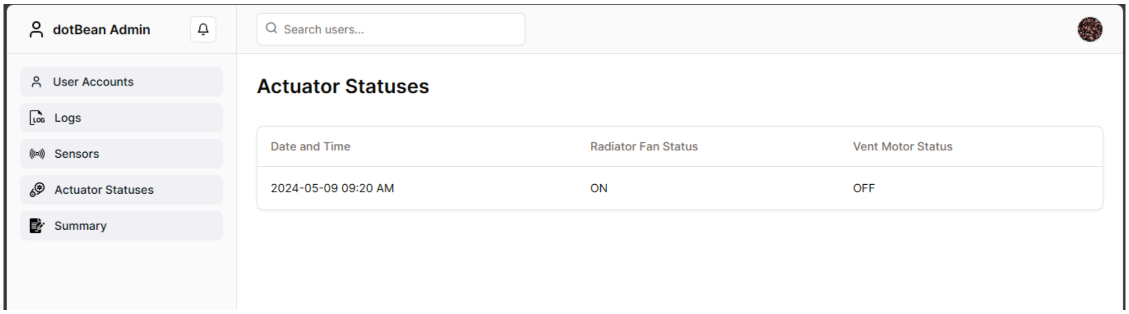


Figure 3-68. dotBean Website Application - Admin Side: Dashboard Actuators Page

In the actuators dashboard, the admin can check the radiator fan and vent motor status to see if they are on and off, as illustrated in Figure 3-68.

Figure 3-69. dotBean Website Application - Admin Side: Dashboard Summary Page

The summary page summarizes all sensors' and actuators' data received from the microprocessor. As suggested in Figure 3-69, the data are summarized in a line chart for the sensors' logs and a bar chart for the actuators' logs. The logs were summarized daily.

3.6.4.4 Hardware Design

The primary element comprising Design Option 2 included the Radiator Fan and Peltier Modules dedicated to regulating temperature and humidity-related activities. The Radiator Fan produces air to maintain the specified temperature range. Three DHT22 sensors are positioned to monitor temperature and humidity levels across the container accurately. These sensors offer precise measurements, covering the entire spectrum from the bottom to the top of the container.

Consequently, to assess the moisture content of the coffee beans in this design option, developers have adopted the indirect method using the dielectric measurement of the coffee beans utilizing a capacitive moisture sensor that measures changes in capacitance caused by variations in the moisture content of the beans. The sensor detects the dielectric properties of the beans, which are directly related to their moisture levels, providing an accurate and non-destructive means of monitoring the coffee's moisture content. Lastly, with the WiFi built-in capability of the Raspberry Pi, a mobile application is deployed in this option to allow the end-users to navigate and monitor the container in real time. Below is an overview of the components selected by the developers for this specific design option and its prototype design:

3.6.4.4.1 Prototype Layout

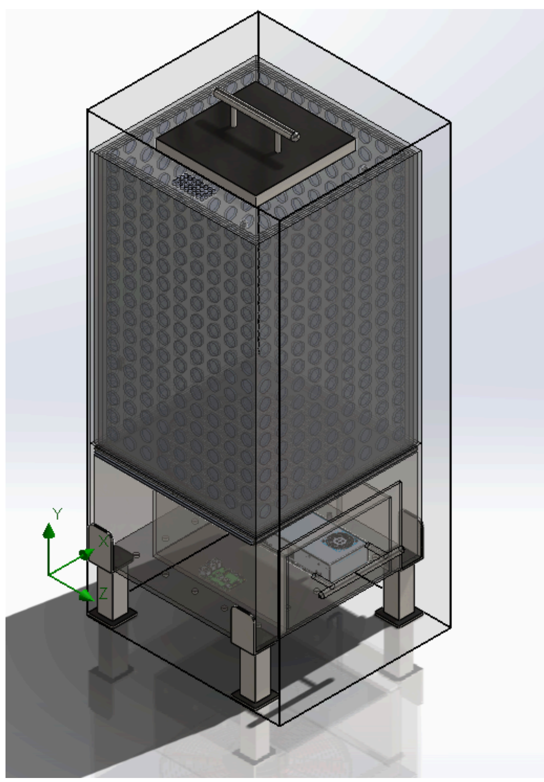


Figure 3-70. Design Option 2 3D View

A prototype layout plays a vital role in the product development lifecycle, providing a tangible representation of design concepts that function as both a visual and functional model that is reliable for conceptualizing and evaluating the proposed product or system. The preceding figure represents the sectional view of Design Option 2, whose purpose relies solely on its capability to store coffee beans, monitor their moisture content, and control the temperature inside. It serves as the primary storage container for the target clients.

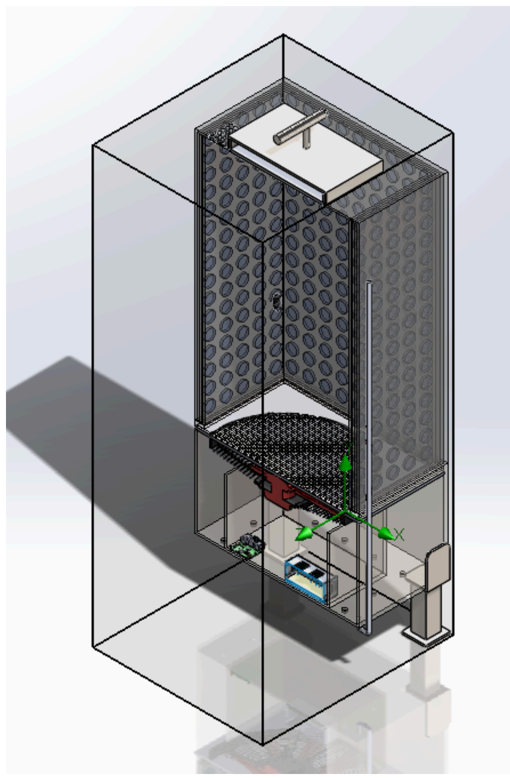


Figure 3-71. Design Option 2 Section View

Design option two has some similarities with design option one. Some components were changed, and other elements will remain, such as the material for storing the coffee beans, which was still acrylic plastic layered with polyolefin foam to maintain the insulation inside the device and remove excess moisture. For the sensors, three DHT22 sensors are placed at different points from bottom to top inside the container to detect the parameters needed at every point. The moisture sensor is intact at the top of the container.

3.6.4.4.2 Schematic Diagram

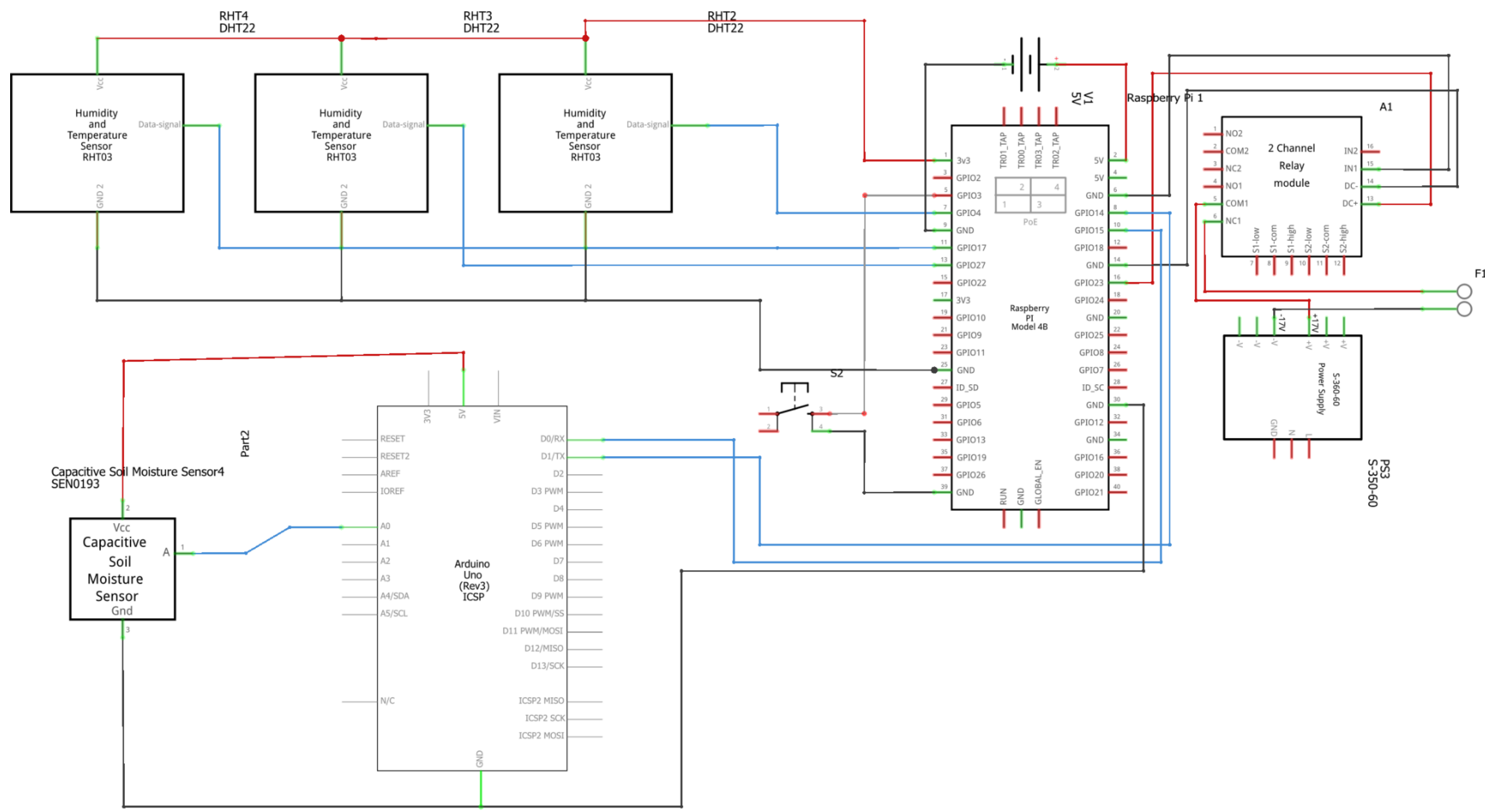


Figure 3-72. Design Option 2 Schematic Diagram

A schematic diagram outlines the functionality and connectivity among different electrical components in a two-dimensional circuit (Yogendrappa, 2021). In this way, the developers can plan for the necessary elements required to make the system functional and complete their connection. Similar to design option 1, option two also utilizes Raspberry Pi 4B as the main microprocessor and DHT22 sensor for monitoring temperature and humidity. However, the difference between the prior design option and design option 2 is the fan and moisture sensor used, as shown in Figure 3-72.

Since the moisture sensor is analog and the Raspberry Pi requires other components to make it work, this sensor is connected to the Arduino, which, through serial communication, retrieves the data from the moisture sensor.

To understand each component's functionalities, the complete hardware components used in this design option and a discussion of their specifications and descriptions are listed below.

3.6.4.4.3 Hardware Components

Microcontroller and Sensor Components

- Raspberry Pi 4B 8GB
- Capacitive Soil Moisture Sensor v2.0
- DHT22

Cover Material (outer and inner)

- Acrylic Plastic
- 304 Food-grade Stainless Metal Steel
- Polyolefin Foam

Heating Regulator Device

- 12V Radiator Fan

Other Components

- 12V Switching Power Supply Unit
- 64GB SD Card
- 2-channel 5V Relay Module

- Momentary Push Button Switch PBS-110
- Raspberry Pi Charger Cable
- Extension Cord

3.6.4.4.4 Functional Specifications

Raspberry Pi 4B 8GB

The Raspberry Pi 4B is a single-board computer featuring a 40-pin GPIO (General Purpose Input/Output) header. This header is utilized to connect the Raspberry Pi to external devices. The GPIO pins can be set as input or output and are used for various applications. They are also capable of generating PWM (Pulse Width Modulation) output. The board supports SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), and UART (Universal Asynchronous Receiver Transmitter) serial communication protocols (*see Table 3-17*).

Table 3-17. Raspberry Pi 4B GPIO Configuration

Pin Type	GPIO Pins
GPIO Pins	GPIO2, GPIO3, GPIO4, GPIO5, GPIO6, GPIO7, GPIO8, GPIO9, GPIO10, GPIO11, GPIO12, GPIO13, GPIO14, GPIO 15, GPIO 16, GPIO 17, GPIO 18, GPIO 19, GPIO 20, GPIO 21, GPIO 22, GPIO 23, GPIO 24, GPIO 25, GPIO 26, GPIO 27
Power Pins	3.3V, 5V, GND
PWM pins	GPIO12, GPIO13, GPIO18, GPIO19
SPI Pins	SPI0: GPIO9 (MISO), GPIO10 (MOSI), GPIO11 (SCLK), GPIO8 (CE0), GPIO7 (CE1) SPI1: GPIO19 (MISO), GPIO20 (MOSI), GPIO21 (SCLK), GPIO18 (CE0), GPIO17 (CE1), GPIO16 (CE2)
I2C Pins	Data: (GPIO2), Clock: (GPIO3) EEPROM Data: (GPIO0), EEPROM Clock: (GPIO1)
UART Pins	TX: (GPIO14) RX: (GPIO15)
Other Pins	ID_SD (I2C ID EEPROM) ID_SC (I2C ID EEPROM)

GPIO pins are used for general-purpose input/output operations. This allows for the control of various digital devices, such as LEDs, buttons, and sensors. These pins also can read data from these devices. On the contrary, PWM pins are designed to generate pulse width modulation signals, which control the amount of power supplied to devices like motors or LEDs by adjusting the signal's duty cycle. Moreover, SPI pins are synchronous serial communication interfaces used for short-distance communication, primarily in embedded systems. It enables full-duplex communication between a controller device and one or more secondary devices.

Additionally, other pins are known as I2C pins, a communication protocol used for short-distance, intra-board communication. It allows multiple low-speed devices to communicate with each other using only two wires. Furthermore, UART, TX, and RX pins are used for Universal Asynchronous Receiver/Transmitter communication. UART is a hardware device or software implementation that translates data between parallel and serial forms. It is used for serial communication between devices. Lastly, other pins include ID_SD, which supports SD card communication, while ID_SC pins support SPI communication. SD cards are a type of flash memory card used for storing digital data, while SPI is a synchronous serial communication interface. See the figure below to visualize the different GPIO pins discussed above:

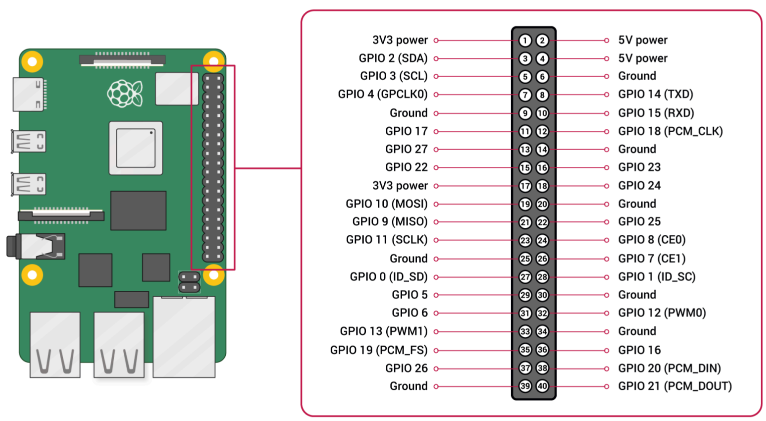


Figure 3-73. Raspberry Pi 4B 40-GPIO Pins

The Raspberry Pi 4 Model B is a high-performance single-board computer introduced by the Raspberry Pi Foundation in June 2019. It is powered by a 64-bit Broadcom 2711, Cortex A72 processor operating at 1.5GHz. Table 3-18 provides a detailed overview of the capabilities of the Raspberry Pi 4 Model B. It includes information about the processor, memory, connectivity options, and power requirements, among other details.

Table 3-18. Raspberry Pi 4B Specification

Raspberry Pi 4 Computer Model B Specification	
Specification Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	1GB, 2GB, 4GB, or 8GB LPDDR4 (depending on model) with on-die ECC
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports two × USB 2.0 ports.
GPIO	Standard 40-pin GPIO header (fully backward-compatible with previous boards)
Video & sound	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); and OpenGL ES, 3.0 graphics
SD card support	Micro SD card slot for loading operating system and data storage
Input power	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Operating temperature 0–50ºC

Raspberry Pi 4 Model B, a high-performance single-board computer, is powered by a 64-bit Broadcom BCM2711 Cortex-A72 processor that operates at 1.5GHz. It comes in three memory configurations: 1GB, 2GB, 4GB, or 8GB LPDDR4, with on-die ECC, and supports a wide range of tasks from simple to complex applications. For connectivity, it supports 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE, and Gigabit Ethernet. It also has two USB 3.0 ports and two USB 2.0 ports, which can connect various peripheral devices. The Raspberry Pi 4 also features a standard 40-pin GPIO header, which is fully backward-compatible with previous boards. This allows for a wide range of input/output operations, making it a versatile tool for many projects.

Furthermore, the Raspberry Pi 4B supports 2 × micro HDMI ports (up to 4Kp60 supported), a 2-lane MIPI DSI display port, a 2-lane MIPI CSI camera port, and a 4-pole stereo audio and composite video port. This makes it capable of handling a wide range of multimedia tasks. The Raspberry Pi 4 supports H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode), and

OpenGL ES 3.0 graphics. It also has a micro SD card slot for loading the operating system and data storage. The Raspberry Pi 4 can be powered via a 5V DC USB-C connector or a 5V DC GPIO header, providing a minimum of 3A. It also supports Power over Ethernet (PoE) capability, although this requires a separate PoE HAT. The Raspberry Pi 4 has an operating temperature range of 0–50ºC, making it suitable for various environments.

Capacitive Soil Moisture Sensor Digital v2.0

Similarly to the previously proposed moisture sensor, the developers calibrated the sensor for use in this project to evaluate the capacitance of the coffee beans when there's interaction. The sensor operates under voltages between 3.3 and 5.V, making it compatible with all the major microcontrollers available. The sensor module has an onboard voltage regulator, ensuring the sensor can work under these voltages.

It works by measuring the changes in capacitance caused by the changes in the dielectric. It does not measure soil moisture directly (as pure water does not conduct electricity well). Instead, it measures the ions dissolved in the moisture. The capacitance of the sensor is measured with the help of a 555-timer-based circuit that creates a voltage directly proportional to the capacitor inserted in the soil. This voltage is then measured by a Digital Converter, which produces a value that represents a percentage of soil moisture.

Figure 3-74. Capacitive Soil Moisture Sensor

The sensor has three pins: VCC, GND, and AOUT. The VCC pin powers the sensor and should be connected to the microcontroller's power supply or the power source you are using. The GND pin completes the circuit and should be connected to the ground of the microcontroller or power source. The AOUT pin outputs the analog signal that represents the soil moisture level. This pin should be connected to an analog input pin of the microcontroller.

DHT22 Module

The DHT22 is a handy, cost-effective device that measures humidity and temperature. It's a digital sensor with a built-in analog-to-digital converter, making connecting with the Raspberry Pi easier.

Table 3-19. DHT22 Specification

DHT22 Specification	
Operating Voltage	3.5V to 5.5V
Operating current	0.3mA (measuring) 60uA (standby)
Output	Serial Data
Temperature Range	-40°C to 80°C
Humidity Range	0% to 100%
Resolution	Temperature and Humidity both are 16-bit
Accuracy	±0.5°C and ±1%

Table 3-19 illustrates the specifications for the DHT22 module; its operating voltage is between 3.5V and 5.5V, and the operating current is 0.3mA during measurement and 60uA in standby mode. This sensor outputs both temperature and humidity data through a serial connection. The temperature range is from -40°C to 80°C, and the humidity range is from 0% to 100%. Moreover, both temperature and humidity readings are 16-bit, providing high resolution, and the accuracy readings of the sensor are ±0.5°C for temperature and ±1% for humidity.

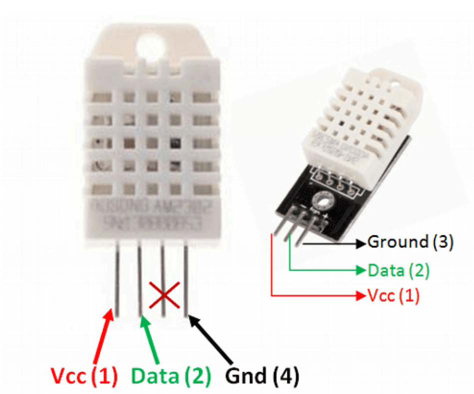


Figure 3-75. DHT22 Pin COnfigurations

The Vcc pin is the power supply, which requires a voltage between 3.5V and 5.5V. The Data pin is the output pin that transmits temperature and humidity data in a serial format. Lastly, the Ground pin, as its name suggests, is connected to the circuit's ground. This pin configuration is fundamental for the DHT22 module to function correctly.

Acrylic Plastic

Figure 3-76. Acrylic Plastic

Acrylic Plastic, or Plexiglas, is a versatile project material due to its optical clarity and impact resistance. It's lightweight, chemically resistant, and easy to machine, making it suitable for various components. In this design, it will be used as the primary cover material and to house sensors that measure the parameters.

316 Food-grade Stainless Metal Steel

316 Food-grade stainless steel is known for being resistant to corrosion and for its resistance to chlorides and acids, making it ideal for use in environments where salt exposure is expected. In this design, it will be used as a storage compartment for the components.

Polyolefin Foam

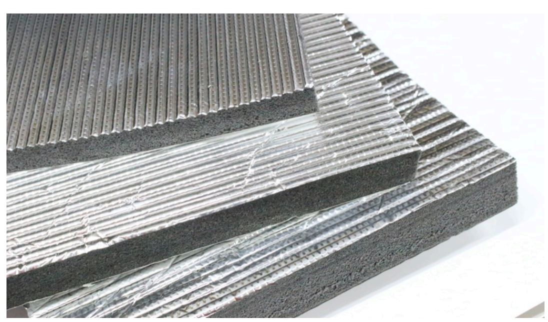


Figure 3-77. Polyolefin Foam

Polyolefin foam is highly valued in various industries for its versatility in insulation and moisture management. Although lightweight, it provides excellent thermal insulation, making it ideal for enhancing energy efficiency in buildings, from homes to commercial spaces like refrigeration units and HVAC systems. These foams effectively minimize heat transfer, reducing energy consumption and maintaining stable indoor temperatures.

Its ability to resist water vapor transmission also makes it invaluable for protecting sensitive goods such as electronics and pharmaceuticals during shipping and storage. It’s even used in waste management to absorb odors and moisture in bins, offering an eco-friendly solution. However, it's essential to handle it responsibly due to environmental concerns like proper disposal and the potential need for treatments like flame retardants in specific applications.

Radiator Fan

Figure 3-78. 12V Radiator Fan

As shown in Figure 3-78, radar fans are crucial in these designs, as they help maintain optimal temperature and humidity levels. The electrical specifications of the radiator fan include an operating voltage within the 12V ±5% V range. The fan's steady-state operation current draw should not exceed 1.5 A. During start-up, the fan's current spike can increase to 2.2A for no more than 1.0 sec. The fan should provide a tachometer output signal of an open-collector or open-drain type with two pulses per revolution. The motherboard will have a pull-up to 12V, with a maximum of 12.6V. The PWM control input signal has a target frequency of 25 kHz, with an acceptable operational range of 21 kHz to 28 kHz. The maximum voltage for logic low is 0.8 V, and the absolute maximum current sourced is five mA (short circuit current). The absolute maximum voltage level is 5.25 V (open circuit voltage).

12V Switching Power Supply Unit (16.7A)

Figure 3-79. 12V Switching Power Supply

A 12V power supply unit is an electronic device with a regulated 12V DC output that converts electrical power from a higher voltage. Its purpose is to provide a maximum current of 16.7A to the radiator and exhaust fan.

64GB SD Card

Figure 3-80. Sandisk Extreme 64gb SD Card

The Sandisk Extreme 64GB SD Card, shown above, is a storage solution for Raspberry Pi. It can store the operating system, applications, and data. This SD Card has a storage capacity of 64GB, a transfer rate of 100MB/s, and a read speed of up to 90MB/s. It's UHS-I compliant, meaning it can support high-speed data transfer. The card uses a microSD card form factor, compatible with the Raspberry Pi's microSD slot.

2-Channel 5V Relay Module

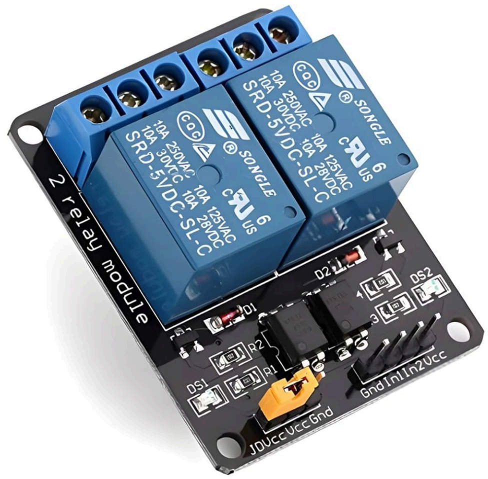


Figure 3-81. 2-Channel 5V Relay Module

The 2-channel 5V relay module is a compact electronic component that allows low-voltage microcontrollers like Arduino or Raspberry Pi to control higher-voltage devices (*see Figure 3-81*).

Table 3-20. Pin Configuration of 2-Channel 5V Relay Module

Pin		Description
Power Pins	**VCC (or JD-VCC)**	Connected to the 5V pin of the microcontroller power supply.
	GND	Connected to the ground (GND) of the power supply.
Signal Pins	**IN1**	Receives the signal control to relay 1 module
	IN2	Receives the signal control to relay 2 module
Output Pins	**NO (Normally Open)**	This pin is not connected to COM when the relay is not activated (off-state).
	NC (Normally Closed)	This pin is connected to COM when the relay is not activated (off-state).
Jumper Settings	**Low-Level Trigger (LL)**	The relay is triggered when the input signal is low (0V).
	Low-Level Trigger (LL)	The relay is triggered when the input signal is high (5V or 3.3V, depending on the microcontroller).

Table 3-20 illustrates the pin configuration of a 2-Channel 5V relay. The power pins VCC and GND connect to the microcontroller to provide power and ground, respectively. Signal pins IN1 and IN2 receive control signals from the microcontroller to activate or deactivate each relay independently. Output pins NO and NC dictate the connection state with COM, providing flexibility in circuit design based on whether the relay is activated. Finally, jumper settings allow users to choose between low-level or high-level triggering based on the microcontroller's output signal characteristics.

Momentary Push Button Switch PBS-110

Figure 3-82. Momentary Push Button Switch PBS-110

The Momentary Push Button Switch PBS-110, as depicted in Figure 3-82, momentarily closes the normally open contact to the common contact. This action allows for brief circuit activation, commonly used in reset mechanisms or as manual control inputs in electronic devices and consumer electronics. It is used to open and close the system.

Raspberry Pi Charger Cable

Figure 3-83. Raspberry Pi Charger Cable

The Okdo Raspberry Pi Power Supply, which comes with a USB Type-C connector, 1.5m cable, and EU plug type, is a reliable choice for powering the Raspberry Pi board. It is designed to support different Raspberry Pi models and delivers sufficient power for stable operation.

Extension Cord

Figure 3-84. Extension Cord

Extension cord is designed to extend the reach of power sources to devices that require placement beyond the proximity of standard outlets. In this project, the primary function of the extension cord is to provide an extended pathway for Raspberry Pi and a power supply in a single outlet to access power from a distant location within the building or area. This capability is particularly beneficial in environments where mobility and reconfiguration of electrical setups are essential, enhancing operational efficiency and convenience.

3.6.4.4.5 Bill of Materials

Due to the developers' cost constraints, below is the summary bill of materials for Design Option 2. Every item that the developers should consider, from the cover materials and fans to be used to the sensors, is listed below to ensure that this design option is within the project's cost constraints.

Table 3-21. Design Option 2 Bill of Materials

Item	Qty.	Price/pc. (₱)	Total Price (₱)
Raspberry Pi 4B 8GB	1	5907	5907
Capacitive Soil Moisture Sensor v2.0	1	75	75
DHT22 Module	4	200	800
Acrylic Plastic (1.2m x 1m)	1	15000	15000
Food-Grade Stainless Steel (4m x 6m)	1	3500	3500
Polyolefin Foam (1.2m x 1m x 12mm)	3	650	1950
Radiator Fan	1	700	700
Exhaust Fan	2	100	200
12V 16.7A Switching Power Supply Unit	1	875	875
Sandisk Extreme 64GB SD Card	1	250	250
2-channel 5V Relay Module	4	79	316
Momentary Push Button Switch	1	65	65
Raspberry Pi Charger Cable	1	300	300
Extension Cord	1	250	250
Total (₱)			30188

The table above is the estimated cost of Design Option 2 based on the required quantity for each component, and the probability of encountering difficulties or defects resulting from the purchase of new materials is beyond the developers' control from the production stage to its implementation.

3.6.4.5 Testing, Validation, and Report

As mentioned in 3.6.2.5, the developers utilized SolidWorks during this phase to test and confirm the first design constraint. They used a theoretical mathematical formula to assess the power consumption. Subsequently, the developers conducted the response time testing and validation in real life because it’s easier to accomplish than simulating it virtually.

3.6.4.5.1 Performance

Using SolidWorks to simulate virtually its run time to achieve the desired parameter for the device, the developers, before conducting the flow simulation, set desired parameters first, in this case, the type of material to be used, what kind of testing to be undertaken, whether if the testing is for the outside or within the inside of the casing, and the initial temperature of the casing.

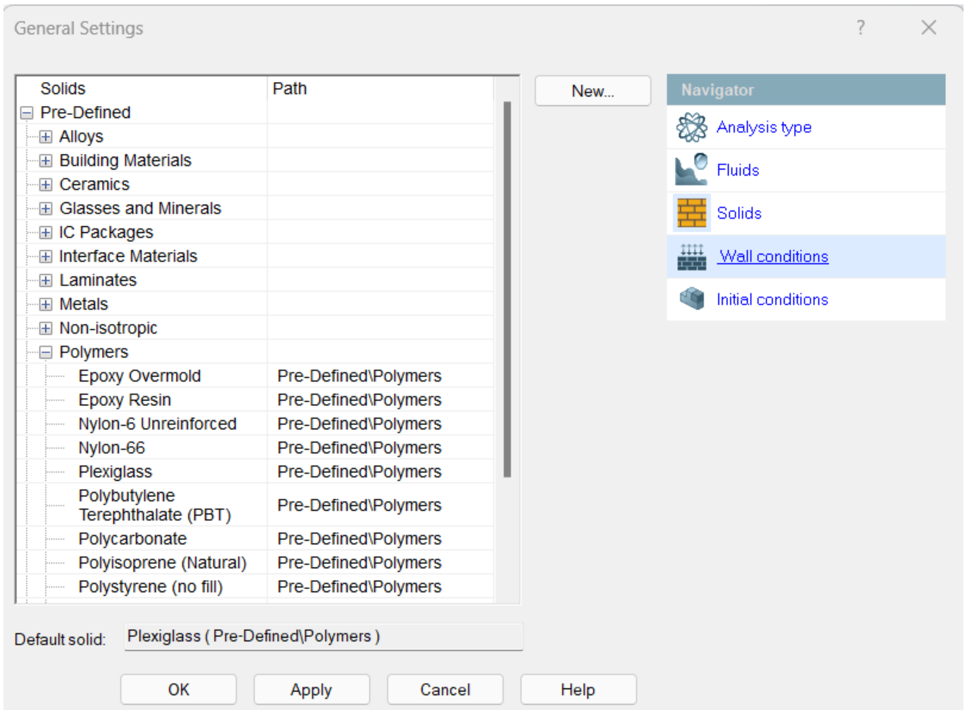


Figure 3-85. Design Option 2 Wall Conditions

In a similar procedure during the Flow Simulation of the previous Design Option, the developers determined the Wall Conditions first. Wall conditions determine how fluid interacts with the solid boundaries in the simulation. They also help accurately represent real-world scenarios. Different wall conditions impact the flow behavior. In this case, under the Polymers, Plexiglass was chosen since Design Option 1 has a cover material of acrylic plastic (*see Figure 3-85*).

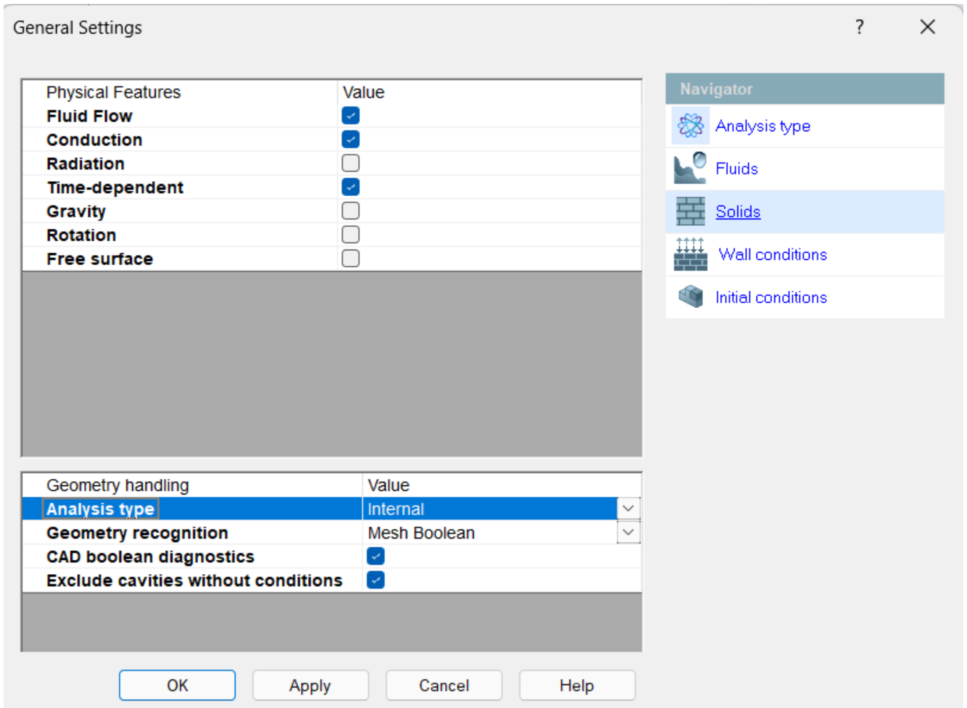


Figure 3-86. Design Option 2 Analysis Type

Next was the Analysis Type, as presented in Figure 3-86. Analysis type determines how the simulation models the interaction between the fluid domain and the surrounding environment. Design Option 2's analysis type is internal, like Design Option 1, which focuses on the enclosed fluid space. Internal analysis allows for more precise modeling of fluid flow within enclosures. This flexibility allows for accurate simulation of real-world scenarios.

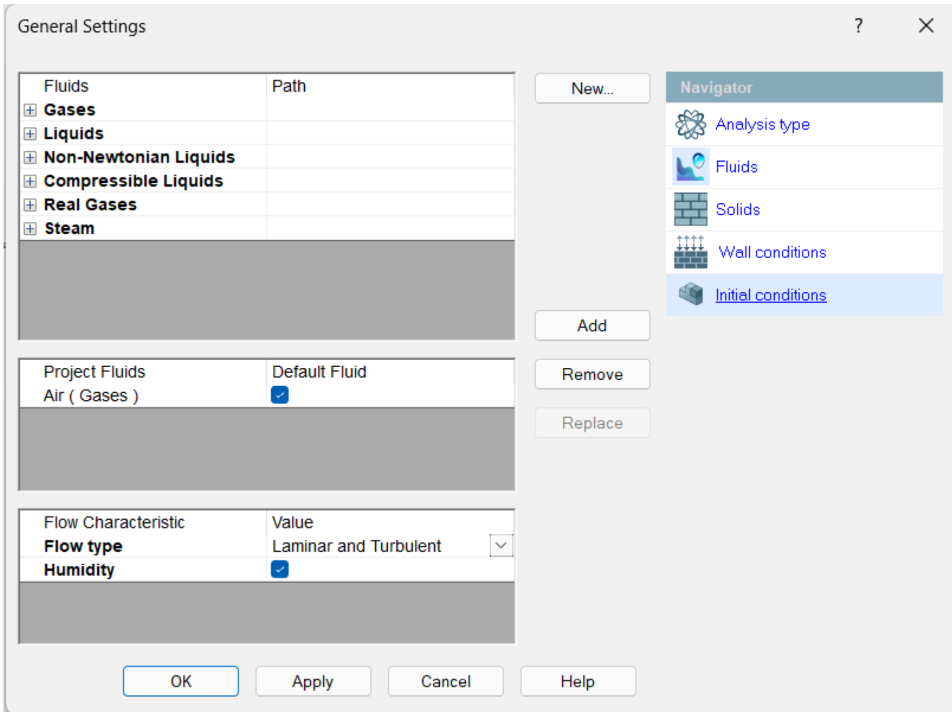


Figure 3-87. Design Option 2 Fluids

In addition to this, still under the “General Settings”, the next thing to determine is the “Fluids”. It is shown in Figure 3-87, Design Option 2 Fluids where Air (Gases), Flow type, and Humidity. Air, in SolidWorks, is typically treated as an ideal gas. The properties of air can be set based on various factors such as temperature, pressure, and humidity. Flow type, on the other hand, determines how the fluid behaves during simulation. In this design option, it’s “Laminar and Turbulent”. Laminar refers to smooth, orderly flow with minimal turbulence. While Turbulent is chaotic, irregular flow with significant mixing between layers. Lastly, Humidity also affects the thermodynamic properties of air. Where if the air is dry, there’s no water vapor present; when air is saturated, it’s the maximum amount of water vapor at a given temperature, and when partially saturated, it varies between dry and saturated states.

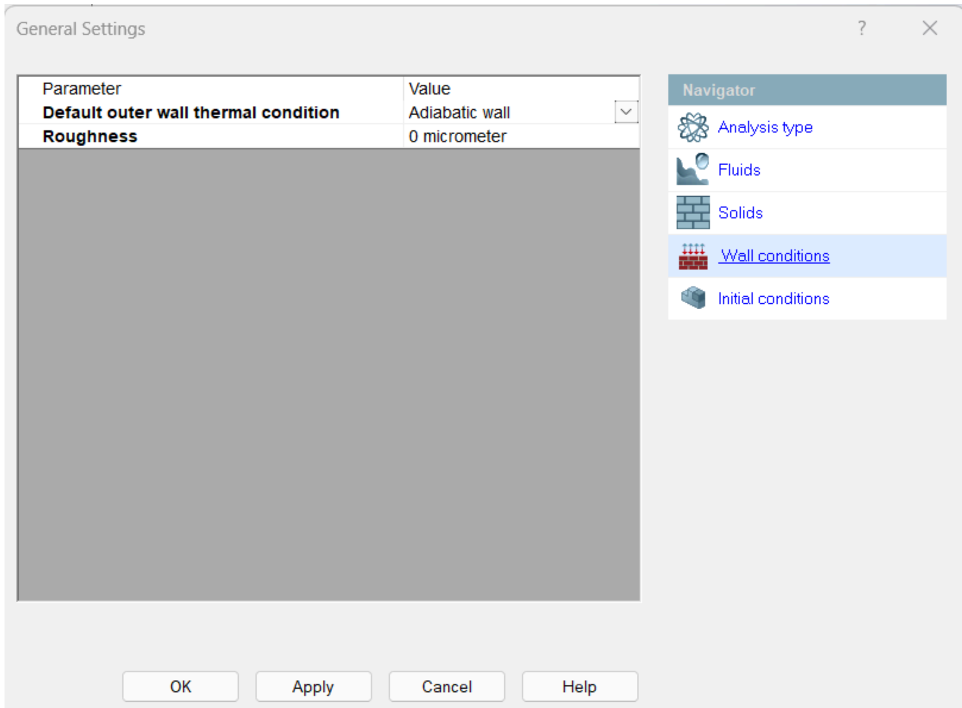


Figure 3-88. Design Option 2 Solids

The last under the "General Settings" was the "Solids," as shown in Figure 3-88. This is the same as Design Option 1, where the outer wall thermal conditions are set to an adiabatic wall, and the roughness is at 0 micrometers.

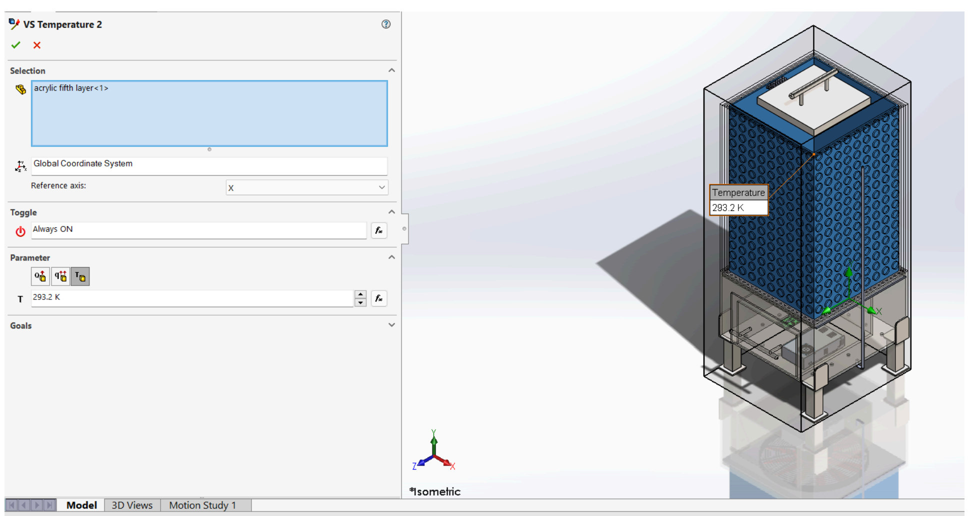


Figure 3-89. Design Option 2 Volume Source Temperature

Moreover, the “Volume Source Temperature” is used to specify the temperature of a heat source in a computational fluid dynamics (CFD) simulation, as displayed in Figure 3-89. It allows defining a region of the model where heat is added to the fluid flow. The volume source temperature of Design Option 2 is inside and within the overall cover material.

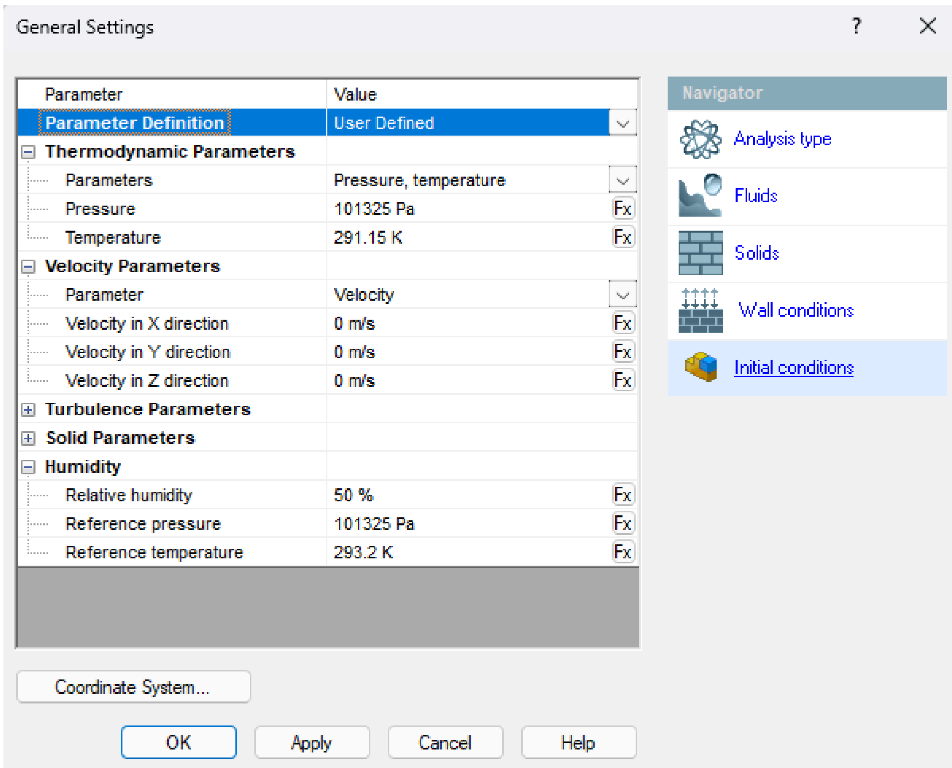


Figure 3-90. Design Option 2 Initial Conditions

Figure 3-90 shows the initial conditions for Design Option 2. The parameters were set “Pressure, Temperature.” Pressure has an initial value of 101325 Pa, and Temperature is 291.15 K. Under “Humidity,” relative humidity is 50%, Reference pressure is 101325 Pa, and Reference temperature is 293.2 K.

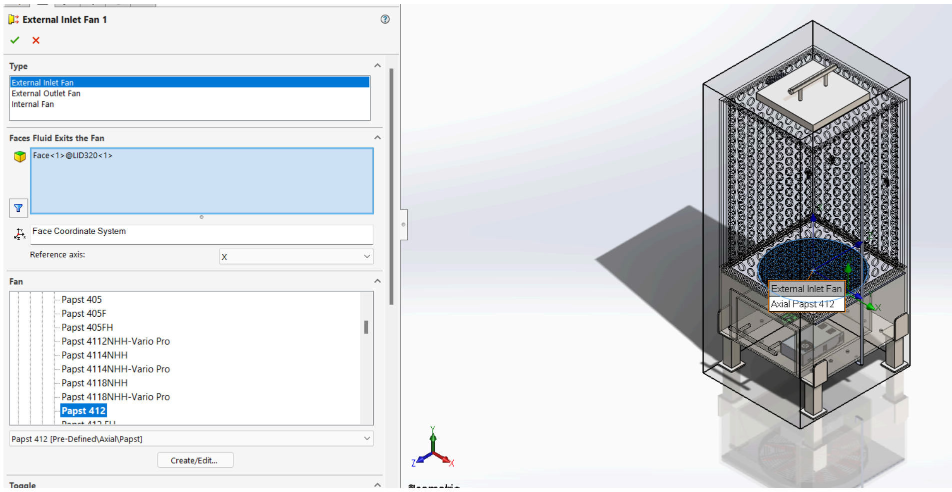


Figure 3-91. Design Option 2 Type of Fan

Moreover, the Type of Fan was also considered, as presented in Figure 3-91. This was one of the factors determining how fast the design would meet the criteria set in the Initial Condition. Design Option 2 has a customized External Inlet Fan specifically calculated for a Radiator Fan.

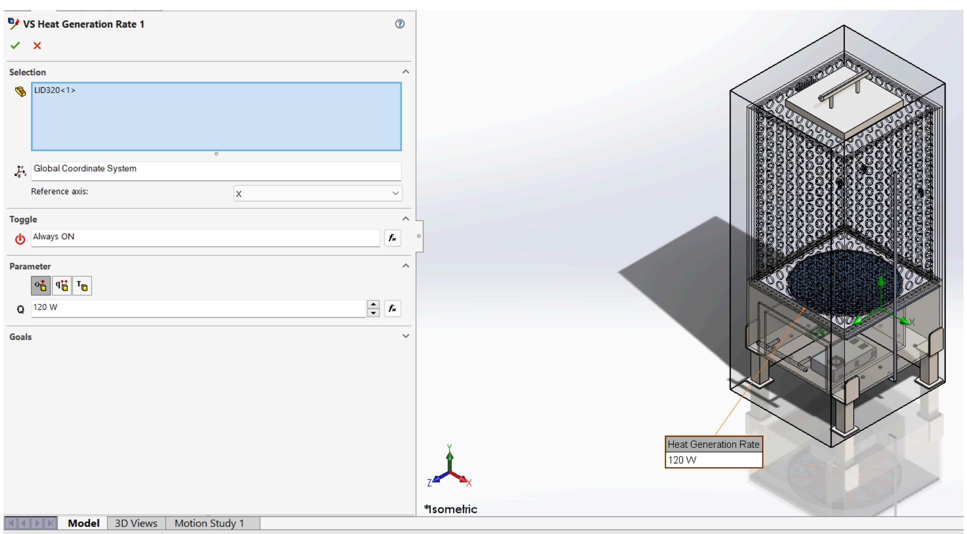


Figure 3-92. Design Option 2 Heat Generation Rate

Once the type of fan is set, its heat generation rate must be set. Heat Generation Rate refers to the amount of heat energy produced per unit volume over a given period of time, which is usually expressed in units like Watts per cubic meter (W/m3). In this design option, the fan's heat generation rate was 120W, the maximum power of a Radiator Fan, as suggested in Figure 3-92.

During the flow simulation, the developers checked the figure to see if changes occurred to ensure the design was feasible. After the simulation, the results can be seen and collected in Excel. Below is the result accumulated for this design option:

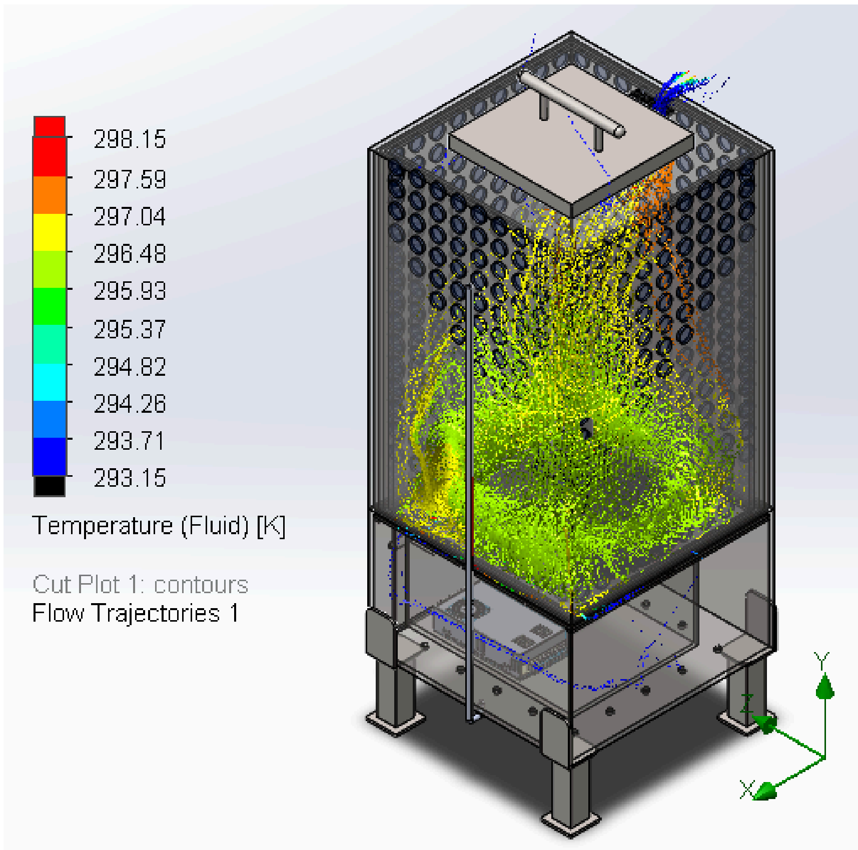


Figure 3-93. Design Option 2 Flow Trajectories Result

Figure 3-93 displays a 3D CFD simulation of fluid flow within a storage. The color gradient from red to green indicates temperature variation in Kelvin, with temperature ranging from 298.15 K at the top to 293.15 K at the bottom. In this design option two, the air inside the storage was mainly yellow-green with some touches of orange, indicating that the temperature inside was 296.48 K to 297.59.

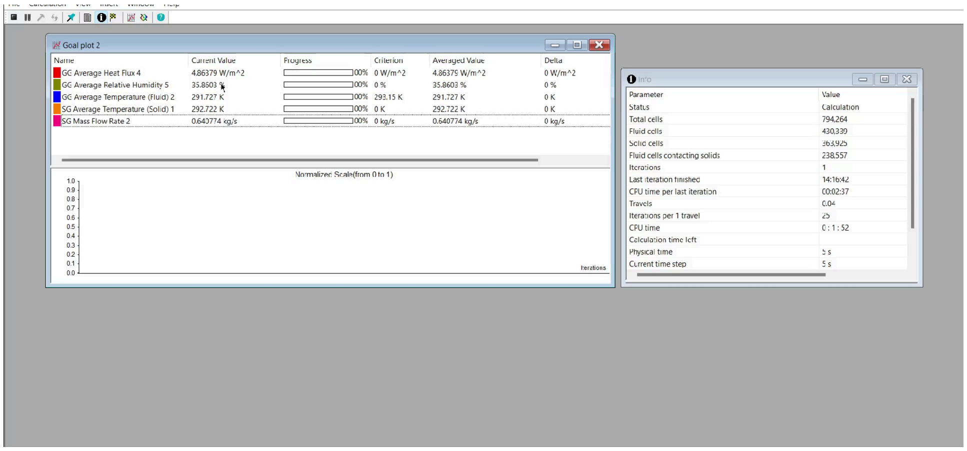


Figure 3-94. Design Option 2 Flow Simulation Start

Figure 3-94 displays the initial setup screen of a computational fluid dynamics (CFD) simulation, likely intended to study airflow and temperature distribution for the dotBean Design Option 2. The left panel shows details of various monitored parameters, including "Average Relative Humidity," "Average Temperature," and "Mass Flow Rate." Each parameter is accompanied by columns for "Current Value," "Progress," "Criterion," and "Averaged Value," all currently at 0%, indicating that the simulation has just started or is yet to begin.

The right panel lists "Parameter" and "Value" pairs, providing more specific information about the simulation settings. These details include the total number of cells used in the simulation, the count of solid, fluid, and wall cells, and other data such as "Solver iterations finished," currently set to 0, suggesting no iterations have been performed yet. The "Run" section shows the name of the computer executing the simulation, marking the starting point for the computational analysis.

By configuring the simulation with these parameters, the software can compute the complex interactions of airflow, temperature, and humidity, which is essential for optimizing the storage units designed to maintain ideal conditions for preserving green coffee beans. The primary objective is to ensure the storage environment remains controlled and stable, safeguarding the beans' quality over time.

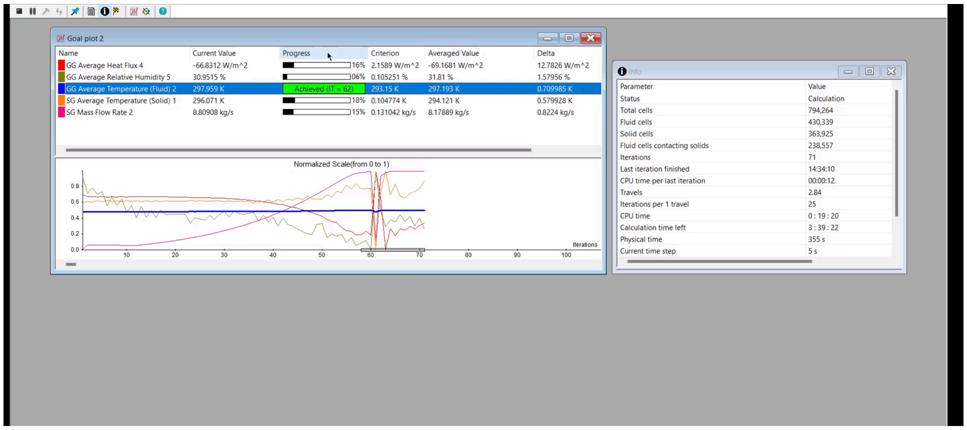


Figure 3-95. Design Option 2 Flow Simulation on Progress

Figure 3-95 illustrates the Design Option 2 Flow Simulation currently in progress. At the top, the Dashboard displays various parameters, such as "Average Particle Fluid Velocity," "Current Slice," and "Design Option 2 Average Concentration," each accompanied by progress bars and numerical values to provide an overview of the simulation's current status. Below this, a graph shows multiple lines representing different data sets over time, specifically labeled "Normalized Solute/Ca++," which helps track the changes in solute concentration.

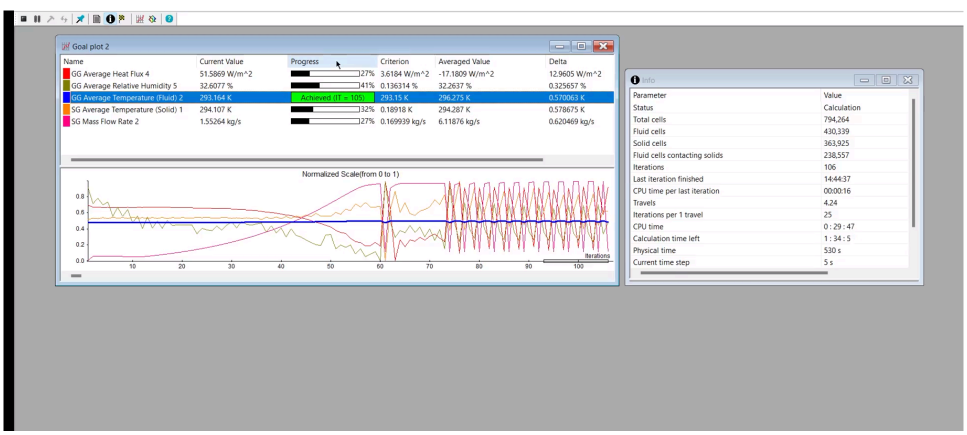


Figure 3-96. Design Option 2 Flow Simulation Finished

The graph displayed in Figure 3-96 depicts multiple lines representing the different parameters over iterations or time. Each of the four lines, distinguished by purple, red, blue, and orange colors, is labeled "Normalized Calculation," followed by a number and percentage. The lines exhibit fluctuations, reflecting changes in the monitored variables.

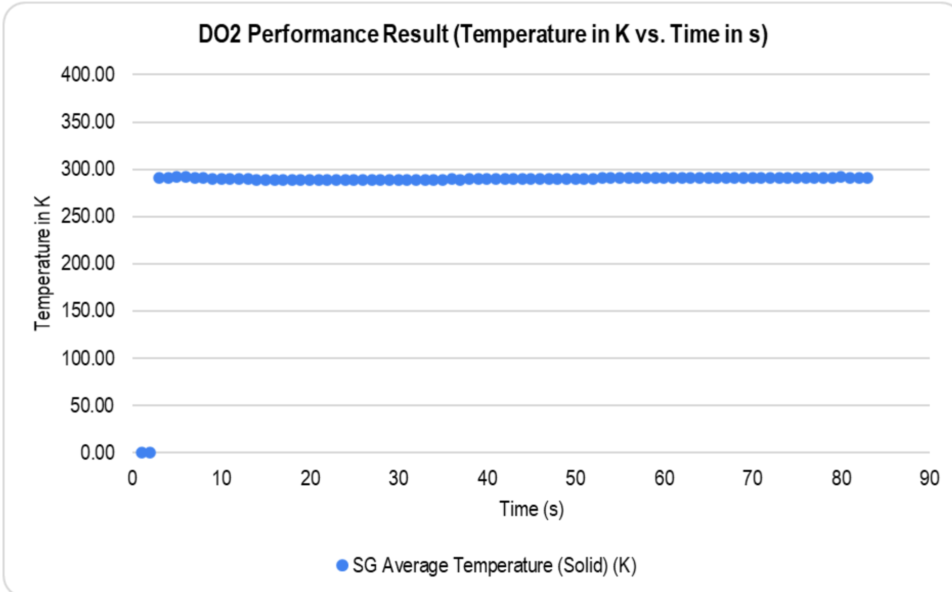


Figure 3-97. Design Option 2 Performance Result (Temperature in K vs. Time in seconds)

Figure 3-97 illustrates the result of a flow simulation of design option 1 in SolidWorks. Changes occurred during the simulation, and the temperature increased from 288.15 K to 293.15 K. It suggests that it took 16 minutes and 56 seconds to complete the simulation, reaching 20°C from 15°C.

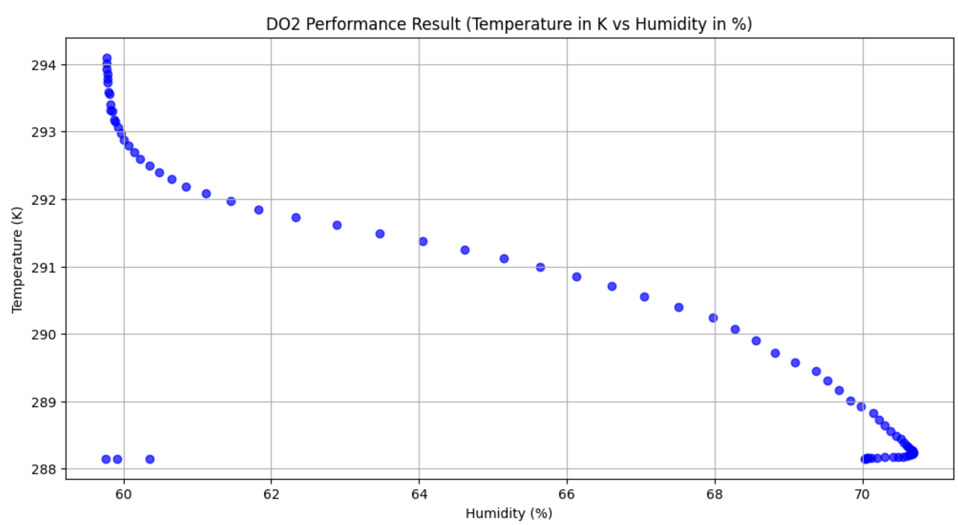


Figure 3-98. Design Option 2 Performance Result (Temperature in K & Humidity in %)

Figure 3-98, on the other hand, depicts the output of Temperature in relation to Humidity. It shows that when the temperature changes, the humidity also changes, indicating that as the temperature increases, the humidity decreases. From just above 70%, where the temperature was 288.15 K or 15°C, it achieved a humidity of below 60% when it reached 293.15 K or 20°C.

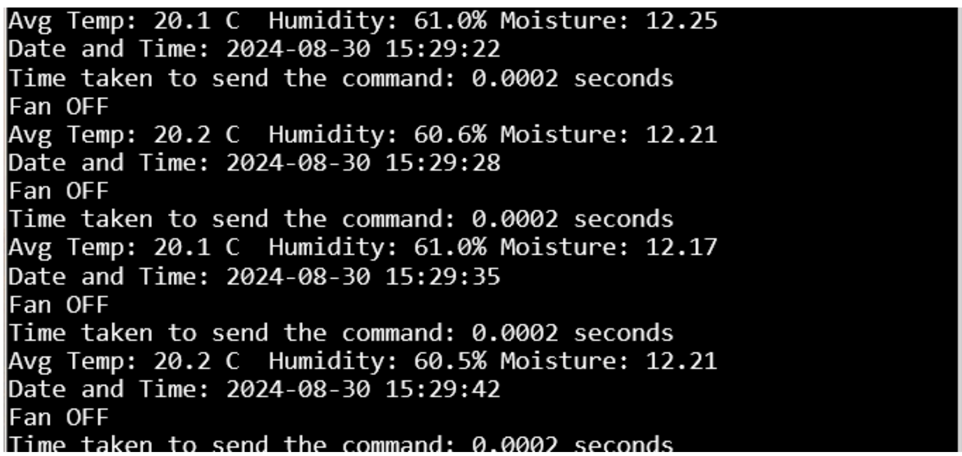
3.6.4.5.2 Response Time

The developers conducted this test to determine how quickly the fan reacted to commands from the Raspberry Pi to turn on or off. Since the system runs automatically, verifying that it operates correctly and reliably without user input is vital. Measuring the fan's response time helped ensure that the system adjusts promptly to temperature changes, maintaining the desired environment efficiently and effectively.

```
# Record the start time
           start_time = time.time()
           fan_status = "OFF" if GPIO.input(RELAY_PIN) == GPIO.LOW else "ON"
            # Fan control logic
            if avg_temperature_c < 20:  # Condition for turning the fan on
                print("Fan ONN")
                GPIO.output(RELAY_PIN, GPIO.HIGH)  # Turn the fan on
            else:  # Default case where the fan should be off
                print("Fan OFF")
                GPIO.output(RELAY_PIN, GPIO.LOW)  # Turn the fan off

# Record the end time
            end_time = time.time()

       # Print the averaged values to the console
            current_datetime = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            message = (f"Avg Temp: {avg_temperature_c:.1f} C  Humidity:
{avg_humidity:.1f}% Moisture: {line}\nDate and Time: {current_datetime}")
            print(message)

            # Calculate the duration in seconds with millisecond precision
            duration = end_time - start_time
            print(f"Time taken to send the command: {duration:.4f} seconds")
```

Figure 3-99. Design Option 2 Response Time Command in Raspberry Pi

The code shown in Figure 3-99 manages the operation of a fan based on temperature readings and logs various details. It begins by recording the current time to measure how long the fan control operations take. It then checks the current state of the fan via a GPIO pin and stores its status. The fan control logic is straightforward: if the average temperature is below 20°C, the fan is turned on; otherwise, it is turned off. The code prints the fan's status and updates the GPIO pin accordingly. After executing the fan control logic, it records the end time, calculates the duration of the operation with millisecond precision, and prints the average temperature, humidity, moisture levels, and the current date and time. Finally, it outputs the duration to execute the fan control command, providing a complete overview of the operation's performance and environmental conditions.



```
Avg Temp: 20.1 C  Humidity: 61.0% Moisture: 12.25
Date and Time: 2024-08-30 15:29:22
Time taken to send the command: 0.0002 seconds
Fan OFF
Avg Temp: 20.2 C  Humidity: 60.6% Moisture: 12.21
Date and Time: 2024-08-30 15:29:28
Fan OFF
Time taken to send the command: 0.0002 seconds
Avg Temp: 20.1 C  Humidity: 61.0% Moisture: 12.17
Date and Time: 2024-08-30 15:29:35
Fan OFF
Time taken to send the command: 0.0002 seconds
Avg Temp: 20.2 C  Humidity: 60.5% Moisture: 12.21
Date and Time: 2024-08-30 15:29:42
Fan OFF
Time taken to send the command: 0.0002 seconds
```

Figure 3-100. Design Option 2 Raspberry Pi Sample Terminal Output Display

The system readings and fan status were displayed during the Raspberry Pi terminal prompt testing. For instance, as shown in Figure 3-98, it took 0.0002 seconds for the Raspberry Pi to command the fan to stay turned off when the temperature is 20.2°C or 20.1°C. The fan will remain turned off since the temperature is between 20°C - 25.4°C

Table 3-22. Design Option 2 Response Time Test Results

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry Pi received the Reading in microseconds (µs)*	*Time the Fan Activated in microseconds (µs)*	*Response Time in microseconds (µs)*
1	19.1	On	100	300	200
2	20	Off	100	300	200
3	20.1	Off	200	400	200

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry Pi received the Reading in microseconds (µs)*	*Time the Fan Activated in microseconds (µs)*	*Response Time in microseconds (µs)*
4	20.1	Off	100	300	200
5	20	Off	100	300	200
6	20	Off	100	300	200
7	20	Off	200	400	200
8	20	Off	100	300	200
9	19.9	On	100	300	200
10	20.2	Off	100	200	100
11	20.2	Off	200	400	200
12	20.2	Off	200	400	200
13	20.2	Off	100	300	200
14	20.1	Off	100	300	200
15	19.7	On	100	300	200
16	20	Off	100	300	200
17	20	Off	100	300	200
18	20	Off	100	300	200
19	20.2	Off	200	300	100
20	20.2	Off	200	300	100
21	20.2	Off	100	200	100
22	20.2	Off	100	200	100
23	20.2	Off	200	500	300
24	20.1	Off	100	300	200
25	20.2	Off	100	200	100
26	20.1	Off	200	400	200
27	20.2	Off	100	400	300
28	20.1	Off	100	300	200

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry Pi received the Reading in microseconds (μs)*	*Time the Fan Activated in microseconds (μs)*	*Response Time in microseconds (μs)*
29	20.1	Off	100	200	100
30	20.1	Off	100	300	200
31	20.1	Off	100	300	200
32	20.1	Off	200	400	200
33	20.1	Off	100	200	100
34	20.1	Off	200	400	200
35	20	Off	100	500	400
36	20	Off	100	400	300
37	20.1	Off	100	400	300
38	20.1	Off	100	300	200
39	20.2	Off	100	300	200
40	20.2	Off	100	300	200
41	20.1	Off	100	300	200
42	20.2	Off	100	200	100
43	20.2	Off	100	300	200
44	20.1	Off	200	400	200
45	20.2	Off	200	400	200
46	20.1	Off	200	500	300
47	20	Off	100	400	300
48	20	Off	100	300	200
49	20.1	Off	100	300	200
50	20.1	Off	300	500	200
Average Response					**198**

Table 3-22 shows the response time testing result of design option 2. The tests were done by getting the time the Raspberry Pi received the reading and the time the fan reacted. The developers did 50 tests to get the average response time, and based on the response time testing results, it has an average of 198 microseconds.

3.6.4.5.3 Power Consumption

A mathematical computation determines the amount of power a design consumes. To accomplish this, the developers first identified the Voltage (V) and Current (I) of each component and then multiplied their values to determine the Power (P) in Watts (W) of the component.

Each component's dataset was used as a reference from different online marketplaces and articles to complete this computation. According to (Makerlab Electronics, n.d.-c), Raspberry Pi 4B 8GB has an operating voltage of 5V at 3A. Arduino Uno R3, on the other hand, has a voltage of 5V at 0.02A (Arduino Tips, Tricks, and Techniques, 2024). Additionally, according to Edmallon (2024), a Capacitive Soil Moisture Sensor has a voltage of 5V and a current of 0.03A. Moreover, the DHT22 Module has a voltage and current of 3.3V and 0.0025 mA, respectively (LME Editorial Staff, 2022), while the Radiator Fan has 12V voltage and a current of 10A (Powering a Radiator Fan With 12V DC Power Supply, n.d.). On the other hand, a Switching Power Supply Unit has a voltage of 12V and 16.7A (Makerlab Electronics, n.d.-a). Lastly, a 2-channel 12V-5V relay module has a voltage of 5V and a current of 10A (Makerlab Electronics, n.d.-a), while a Momentary Push Button, as stated by Smoot (2022), has a voltage of 5V and a 1A current.

The process of solving each P is illustrated in Table 3-23.

Table 3-23. Design Option 2 Power Consumption in Wattage (W) and Kilowatt-hour (kWh) Result

COMPONENTS	Qty.	V	I	P (in W)	kWh
Idle State					
Raspberry Pi 4B 8GB	1	5	3	15	0.015
Arduino Uno R3	1	5	0.02	0.1	0.0001
Capacitive Soil Moisture Sensor v2.0	1	5	0.03	0.15	0.00015

COMPONENTS	Qty.	V	I	P (in W)	kWh
DHT22 Module	3	3.3	0.0025	0.02475	0.000025
Active State					
Radiator Fan	1	12	10	120	0.12
2-channel 5V Relay Module	1	5	10	50	0.05

Table 3-23 shows the result of the power consumption in wattage; the radiator fan is used for heating in this design. The table shows the items and the quantity used in the design option 2. Each item has its value in volts and amperes. These values are used to calculate the power in watts in each item. The Raspberry Pi contains 0.015 kWh, Arduino uno consumes 0.0001 kWh, the capacitive soil moisture consumes 0.00015 kWh, the DHT22 Module with the quantity of 3 consumes 0.000025 kWh, the Radiator fan consumes 0.12 kWh, and the 2-channel 5V relay module consumes 0.05 kWh.

The developers have excluded the computation of the momentary push button because it doesn't consume significant power, with the electrical current involved being minimal—usually in the microampere range—since it is only used to detect state changes. The power supply is also excluded, as it doesn't add any extra load; instead, it supplies power to the connected devices. This means the power consumed by the power supply from the outlet is simply the total of the power used by all connected devices, plus the minimal inefficiency caused by heat, LED, and other factors inherent in the power supply.

Table 3-24. Design Option 2 Power and Energy Consumption Result

Power and Energy Consumption	State	
	Idle (at use)	Active
Total Power (W)	15.27475	185.27475
Total Kilowatt-hour (kWh) (1 hr)	0.0153	0.1853

Table 3-24 summarizes the power consumption for the device Design Option 2, categorized into two states: idle and active. In the idle state, the device consumes 15.27475 watts of power, reflecting the energy it requires while not actively performing tasks but remaining on standby. During active use, the power consumption increases to 185.27475 watts, demonstrating the additional energy needed to perform tasks. Power consumption over one hour is measured in kilowatt-hours (kWh), with the device using 0.0153 kWh in idle mode and 0.1853 kWh in active mode.

3.6.4.5.4 Public Health and Safety

Design option 2 incorporates acrylic plastic with polyolefin foam for moisture absorption and thermal insulation. Polyolefin foam is known for being lightweight, durable, and moisture-resistant, making it an excellent choice for food packaging and storage applications where temperature and humidity control are necessary. Studies indicate that polyolefin foams provide effective thermal insulation and moisture resistance, which is crucial for maintaining ideal storage conditions for sensitive products like coffee beans (Li et al., 2020). Combining acrylic plastic with polyolefin foam offers both structural integrity and efficient insulation, which helps maintain stable environmental conditions necessary for storing hygroscopic and porous coffee beans (Garcia et al., 2023).

3.6.4.5.5 Social Welfare

For Design Option 2, the developers create a storage device that accurately monitors the temperature and humidity inside the device, ensuring that temperature is maintained within 20° C - 25.4° C and the humidity at 50% - 70%. The moisture content of the green coffee beans should also be within the specific range of 11% to 12.5%, not exceeding 14%. The conducted flow simulation suggests that it can take 16 minutes and 56 seconds to complete the simulation, reaching 20°C from 15°C. For the response time, the developers identified how long the Raspberry Pi commands the fan to turn on or off whenever there’s a new temperature reading, leading to 50 tests resulting in an average response time of 198 microseconds. For the power consumption, based on the calculated power for the idle (in use) and active state, the result for power consumption has a total Kilowatt-hour (kWh) of 0.0153 for the idle state and 0.1853 for the active state.

3.6.4.5.6 Economics

The developers adopted the same approach as in the first design option by creating a Bill of Materials (BOM) with a budget of 50,000 PHP (*see Table 3-21*). The developers selected components available in the Philippines to keep costs down and minimize sourcing issues, particularly in La Trinidad, Benguet. The developers avoided additional import costs and delays by using locally sourced materials such as acrylic plastic, polyolefin foam, sensors, microcontrollers, and power supplies. This method effectively managed material procurement and cost challenges, keeping the project within its financial constraints.

3.6.4.5.7 Social

For the customer's selection, the developers designed a storage device that can be used by all small-scale coffee farmers and coffee shop owners who currently use traditional storing processes in all regions experiencing lower temperatures. Coffee farmers have issues preventing the detrimental effects of excess moisture absorption for the green coffee beans that typically occur during low temperatures.

3.6.4.5.8 Global

For Design Option 2, the developers utilized Raspberry Pi's Wi-Fi capabilities to notify users through a mobile application. According to Gibson (2024), the Raspberry Pi is FCC-certified, which meets the Federal Communications Commission's standards for controlling electromagnetic interference, ensuring the device is safe and reliable. The mobile app was designed simply, providing users with all necessary data for easy access. To streamline the user experience and avoid additional technical complexities, the developers opted for users to log in with their existing Gmail accounts instead of creating new accounts for the app. This approach enhances accessibility and reduces barriers to use.

Like Design Option 1, all components were sourced locally, primarily from Makerlab Electronics, which offers a physical store in Manila and online shopping with nationwide shipping. Power consumption was carefully monitored during testing and validation, ensuring that this design option also maintains low energy usage, aligning with the project's efficiency and budget constraint goal.

3.6.4.5.9 Cultural

Design Option 2 maintains the same cultural sensitivity by concentrating solely on improving the storage process, thus not interfering with traditional coffee farming and processing practices before and after this stage. The device is designed to be intuitive and straightforward, emphasizing ease of use so that farmers can quickly see its benefits without needing extensive routine changes. By consulting with local farmers and incorporating their feedback, the developers ensured that the device addresses the specific needs and challenges of coffee storage, making it relevant and beneficial to the community. The choice of materials and components is guided by local availability and ease of maintenance, supporting the integration of the device into the local setting.

3.6.4.5.10 Environmental

Design Option 2 employs a similar approach to Design Option 1 by utilizing acrylic plastic, ensuring the material is both food-safe and environmentally friendly. Like Design Option 1, this design is characterized by low power consumption, as shown in testing and validation (*see Table 3-23 and Table 3-24)*, contributing to energy conservation. The modular design allows for individual component replacement, reducing waste and simplifying maintenance, ensuring that only defective parts must be replaced instead of the entire device.

3.6.4.6 Design Option 2 Constraints and Results Summary

Following the testing to assess the effectiveness of this design option, the table below summarizes the design constraints, their corresponding results, and relevant remarks.

Table 3-25. Design Option 2 Constraints and Results Summary

Constraints	Results	Remarks
Performance	16 minutes and 56 seconds	Design option 2's total time to reach the minimum threshold value at 20°C was 16 minutes and 56 seconds, well below the criteria of within 30 minutes.
Response Time	198 microseconds	Design option 2's total average response time from the 50 testings was 198 microseconds or 0.000198 seconds, which achieved the criteria of less than 1 second.
Power Consumption	0.0153 kWh at idle state 0.1853 kWh at active state	Design option 2's power consumption when in the idle (at use) state and active state achieved the criteria of not consuming more than 1 kWh.

Table 3-25 shows that design option two successfully met all the specified constraints. The system performed well, quickly reaching the optimal temperature when it dropped below the 20°C threshold. Although its response time was 198 microseconds—4 microseconds slower than the first design option—it still demonstrated a reliable automated fan control system in terms of energy efficiency; design option 2 consumed only 0.0153 kWh in its idle state and 0.1853 kWh in its active state, which is lower than the power consumption of the first design option.

Overall, design option 2 offers a practical solution, meeting all performance, response time, and power consumption criteria. Its ability to rapidly achieve the target temperature, respond quickly to system changes, and consume minimal power makes it a dependable and energy-efficient choice for automated fan control in temperature-sensitive environments.

3.6.5 Design Option 3

The design option 3 of the dotBean's main features consists of utilizing a Thermoelectric Peltier Module and a vent controlled by a servo motor for temperature-and-humidity-related activities and dielectric method for measuring the moisture content of the coffee beans using capacitive sensing method, like Design Option 2. A GSM module is attached to the system to allow users to receive a notification and monitor the changes and status of the device through a short messaging service (SMS). This addresses the potential issue of internet connectivity in the area. Below is a discussion of how this design option works and functions under dotBean's objectives and client requirements.

3.6.5.1 System Architecture

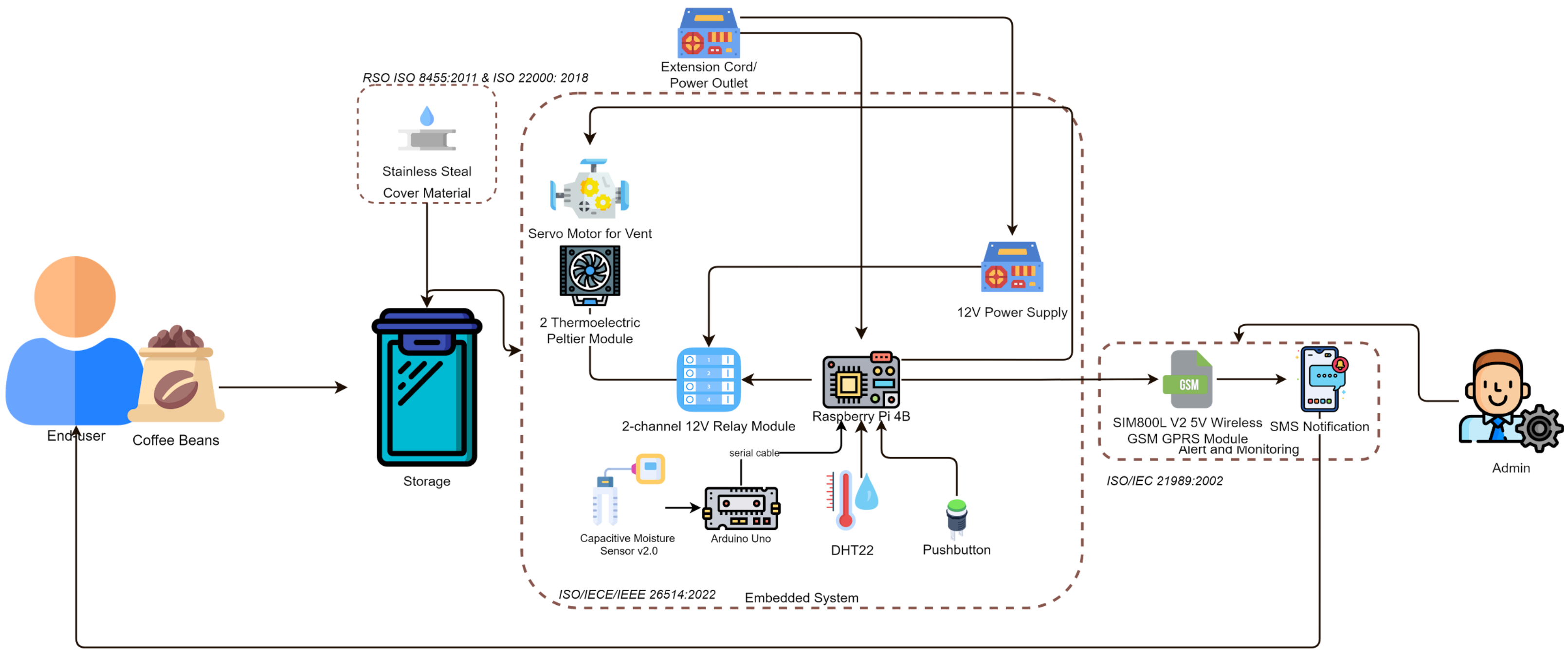


Figure 3-101. Design Option 3 System Architecture

Unlike the two previous design options, design option 3 has a different cover material, moisture sensor measurement used, accessibility options, and a GSM module for SMS notification, allowing users to notify and alert users about changes inside the dotBean container. It used food-grade stainless steel as cover material. This design option's actuators mainly feature a Thermoelectric Peltier Module to maintain the desired conditions and a Vent to let the hot air out of the container. Integrated sensors are DHT22 for temperature and humidity monitoring and Capacitive Soil Moisture v2.0 for measuring the moisture content. Capacitive moisture sensor v2.0 works by measuring the capacitance between two electrodes. Consequently, these readings from two sensors are transmitted in the Raspberry Pi 4B for data analysis. This microcontroller also controls the actuators automatically to turn on or off depending on the parameters.

3.6.5.2 Design Process Flow

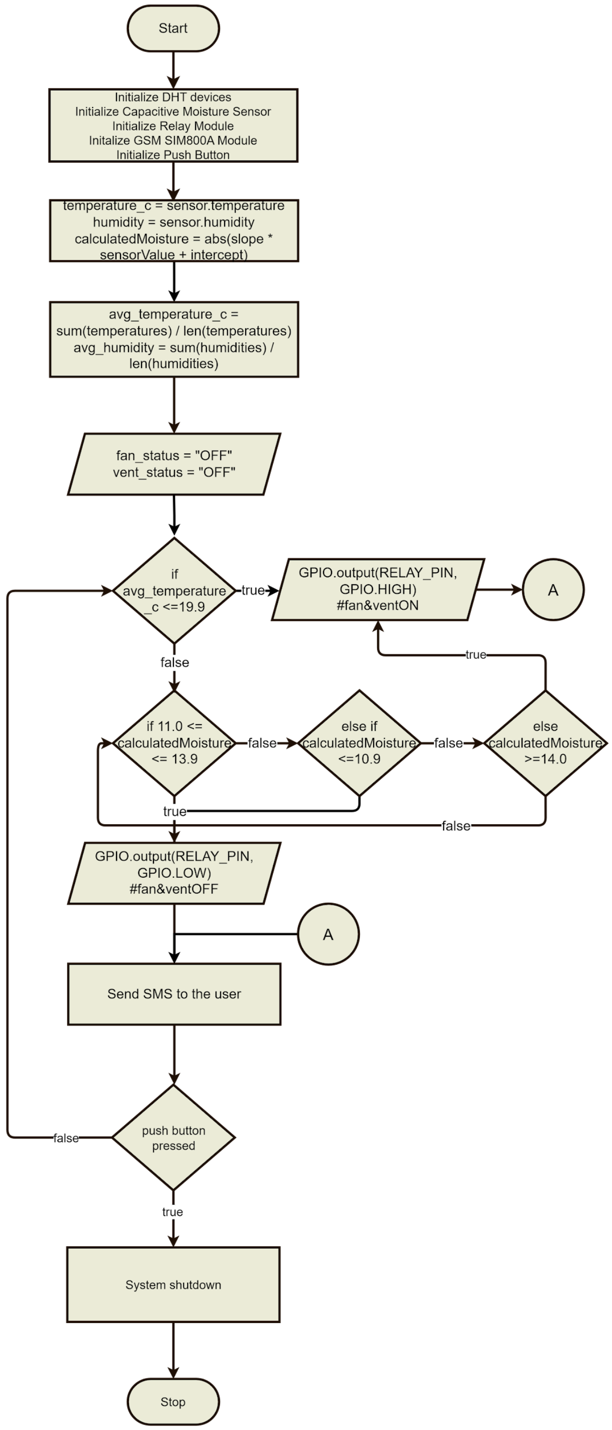


Figure 3-102. Design Option 3 Design Process

Design option three design process flow, as shown in Figure 3-102, involves retrieving data from sensors to Raspberry Pi and data analysis. This design option employs a capacitive method using the Capacitive Soil Moisture v2.0 sensor to measure the moisture content of the coffee beans. The capacitive moisture sensor works by measuring the capacitance between two electrodes. The sensor measures how quickly a capacitor charges through a capacitance. The charging rate is affected by the moisture content in the coffee beans. By measuring this effect, the sensor can indirectly measure the moisture in the coffee beans. The value from the sensor is converted into percentage (%) value using a specific computation.

Three DHT22 sensors are positioned inside the container to monitor temperature and humidity. This ensures a balanced environment, maintaining the right conditions for both moisture content and coffee quality. These readings and the system's status are then notified through SMS. As mentioned above, an SMS feature is integrated into this design option to offer a more straightforward design integration that allows users to interact with the system using telecommunication. All conditions and statuses inside the system will be sent to the users via SMS.

3.6.5.3 Software Design

Design Option 3 takes a different approach to the system's accessibility option, utilizing a GSM module that connects the user and the system through text messaging. Equipped with SIM800L V2 5V Wireless GSM GPRS Module, the system sends an automated message to the client's phone number. In that way, the system can still effectively communicate with the user even if the data connection is not at its best. This design offers a distinct advantage in areas with challenging internet connectivity. By leveraging the GSM module, the system ensures that critical notifications or alerts are promptly delivered to the client via text messages. Similar to other design options, this option also includes real-time data from sensors and the current statuses of the actuators, which the Raspberry Pi processes. As previously mentioned, dotBean entirely operates automatically; the SMS feature of the dotBean notifies the user sequentially about the system changes. The succeeding discussion lists and explains different software applications, including where they are applied in the study and where the developers utilized them. Lastly, the figure illustrates and discusses software development and its process flow.

3.6.5.3.2 Software Technologies

SolidWorks for Prototype Layouting and Simulation

SolidWorks is known for its computer-aided design (CAD) and engineering (CAE) prowess. Offering a parametric modeling approach, the software empowers users to create 3D models by defining parameters and constraints, facilitating easy modifications. The developers utilized this software to lay out the prototype of each design option and conduct a simulation to attest and validate the design option's capability to suffice the design constraints and design criteria. Overall, SolidWorks, with its parametric modeling, assembly features, simulation capabilities, and integration support, is a valuable tool in designing and developing computer engineering projects, providing a comprehensive solution for creating electronic systems with precision and efficiency.

Fritzing for Circuit and Schematic Diagram

Fritzing distinguishes itself by providing a user-friendly interface specifically tailored to accommodate individuals new to electronics and beginners in the field. Including drag-and-drop functionality, coupled with visual representations of components, streamlines the process of circuit creation, offering a simplified and intuitive design experience. With its breadboard view feature, the developers easily experiment with circuit designs in a simulated environment before transitioning to a tangible prototype. This functionality is instrumental in identifying potential issues and errors during the design process.

Geany as Raspberry Pi IDE for Programming

Geany establishes itself as a robust code editor for Raspberry Pi, given its pre-installation in Raspberry Pi OS and its adeptness in facilitating Python or C/C++ coding. The presence of a built-in terminal enables developers to seamlessly compile and execute scripts directly within the Geany environment. Geany IDE is utilized for configuring sensors and actuators and SMS configuration with the GSM module and establishing a connection with its respective API.

3.6.5.3.3 Software Development

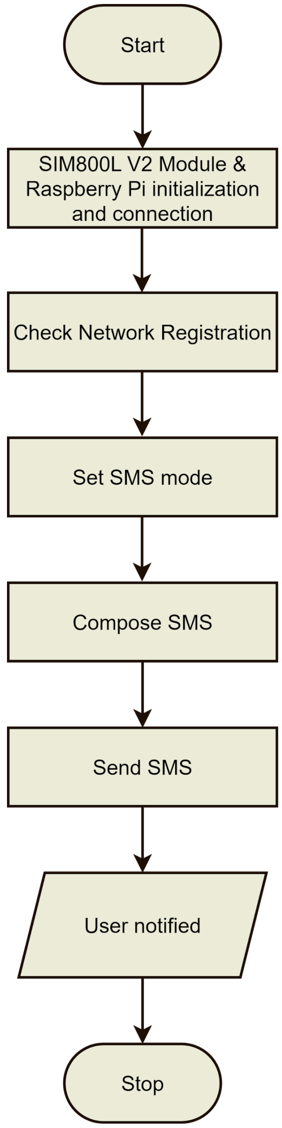


Figure 3-103. Design Option 3 Software Process

To initiate the process, the Raspberry Pi, powered on and connected to the internet, reads data from DHT22 and Capacitive Soil Moisture Sensor v2.0. Simultaneously, it checks the status of the Thermoelectric Peltier Module and Vent, determining whether it's ON or OFF and OPEN or CLOSE. A connection is established between the Raspberry Pi and the GSM module, often facilitated through UART communication or specific libraries. With the data, the Raspberry Pi constructs an SMS message detailing the sensor readings and the actuator's status. This content is then relayed to the GSM module for transmission.

Upon successfully sending the SMS, the GSM module initiates the transmission process. Once the message reaches its destination, the recipient's GSM device receives and displays the content. Lastly, once the SMS content is transmitted to the GSM module and sent, it undergoes the mobile network's processing. Once the SMS content is transmitted to the GSM module and sent, it undergoes the mobile network's processing. The recipient's mobile device receives the SMS message. After transmission, a confirmation response from the GSM module, signaling successful delivery or any encountered errors, is monitored and logged.

3.6.5.4 Hardware Design

The primary element comprising Design Option 3 included the Thermoelectric Peltier Module with a vent below to regulate temperature and humidity-related activities. The Thermoelectric Peltier Module produces air to maintain the specified temperature range. On the other hand, the Vent, controlled by a servo motor, lets the hot air inside the container come out. A minimum of three DHT22 sensors are employed point to point to monitor temperature and humidity levels across the container accurately. These sensors offer precise measurements, covering the entire spectrum from the bottom to the top of the container.

Consequently, to assess the moisture content of the coffee beans in this design option, developers have adopted the indirect method using the dielectric measurement of the coffee beans utilizing the capacitive method. This approach incorporates Capacitive Soil Moisture Sensor Digital v2.0, a capacitive moisture sensor for soil. Lastly, to allow end-users to monitor the container even if they’re not in the actual storage facility or site, a GSM module is employed in the system to send users SMS notifications about the changes and status of the container. Below is an overview of the components selected by the developers for this specific design option and its prototype design:

3.6.5.4.1 Prototype Layout

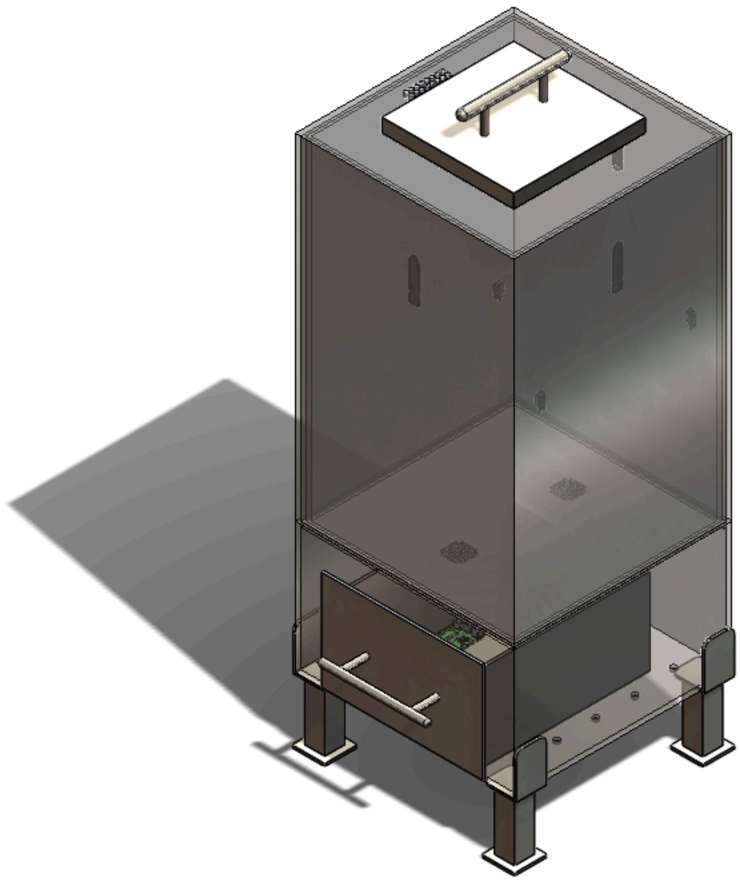


Figure 3-104. Design Option 3 3DView

A prototype layout plays an essential role in the product development lifecycle, providing a tangible representation of design concepts that function as both a visual and functional model that is reliable for conceptualizing and evaluating the proposed product or system. The above figure illustrates the 3D view of Design Option 3, where its purpose relies solely on its capability to store coffee beans, monitor their moisture content, and control the temperature inside. It serves as the primary storage container for the target clients.

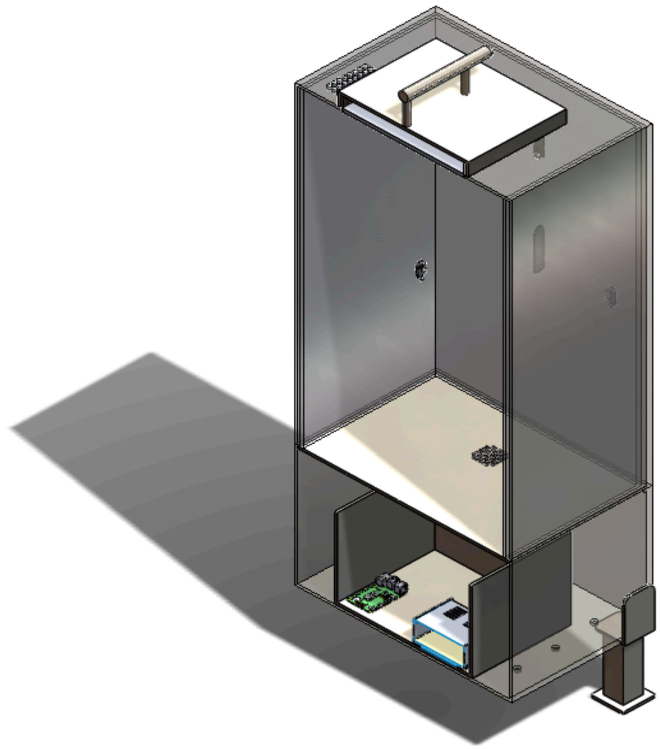


Figure 3-105. Design Option 3 Section View

The design option three devices are made of food-grade stainless steel material, making them airtight and resistant to corrosion and withstand chemicals. Installed inside the container are three DHT22 sensors and a Capacitive Soil Moisture sensors to monitor the temperature and humidity inside the container and maintain the moisture content of the coffee beans. The device will be supported by a 12V power supply mounted in the compartment together with the Raspberry Pi for the automation of the device, a Wireless GSM GPRS Module to send a notification via SMS to the end-user client..

3.6.5.4.2 Schematic Diagram

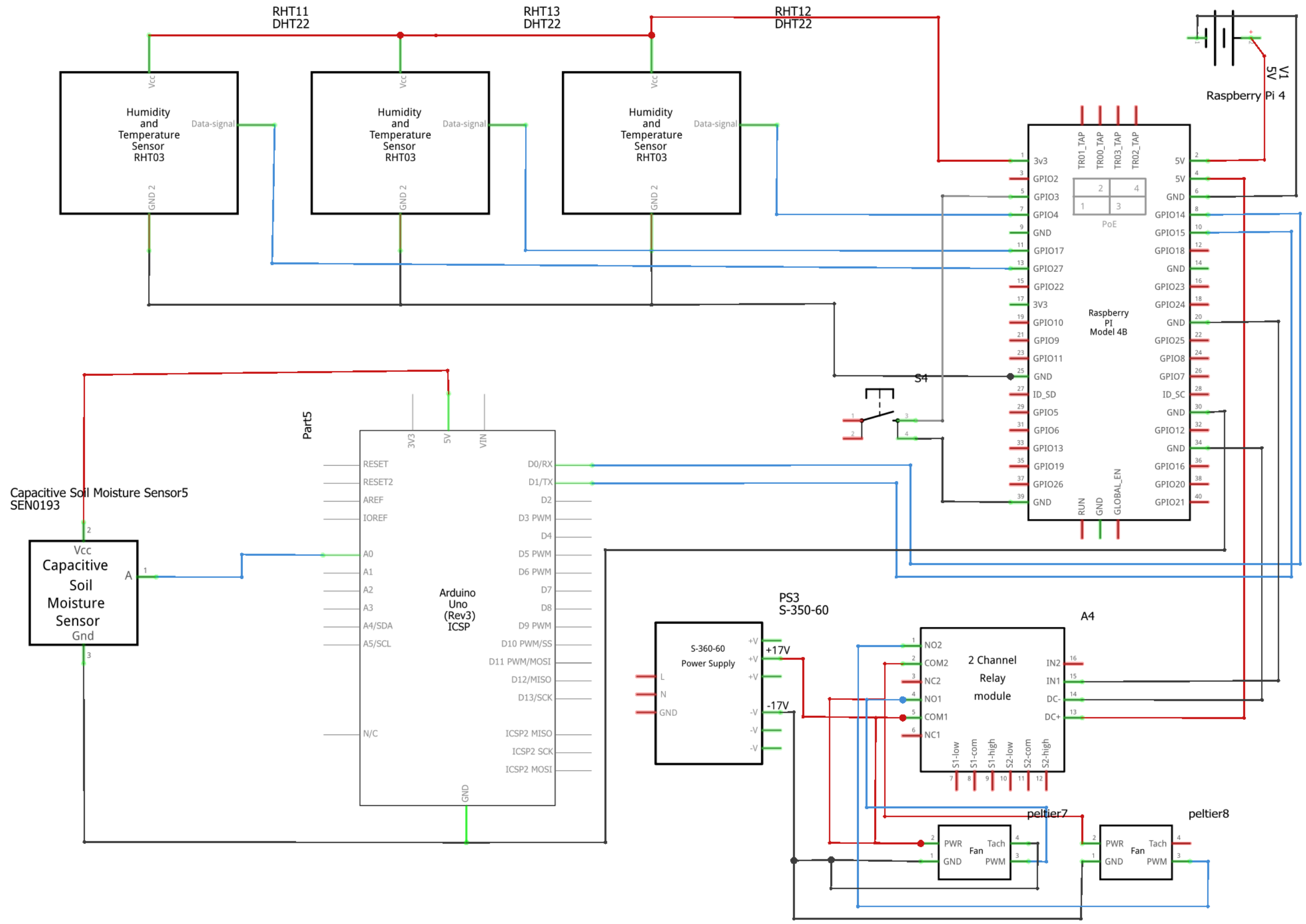


Figure 3-106. Design Option 3 Schematic Diagram

A schematic diagram outlines the functionality and connectivity among different electrical components in a two-dimensional circuit (Yogendrappa, 2021). In this way, the developers can plan the necessary components required to make the system functional and complete their connection. Similar to prior design options discussed above, design option three utilizes Raspberry Pi 4B as the main microprocessor and DHT22 sensor for monitoring temperature and humidity. However, the difference between design option two and this design option is the fan, as shown in Figure 3-106. Additionally, instead of EMail or in-app notification, the GSM module is integrated into this design option for accessibility. To understand each component's functionalities, the complete hardware components used in this design option are listed below, and the complete discussion of their specifications and descriptions.

3.6.5.4.3 Hardware Components

Microcontroller and Sensor Components

- Raspberry Pi 4B 8GB
- Capacitive Soil Moisture Sensor Digital v2.0
- DHT22

Cover Material (outer and inner)

- 316 Food-Grade Stainless Steel

Heating Regulator Devices

- Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit

Other Components

- SIM800A Module
- 12V 16.7A Switching Power Supply Unit
- Sandisk Extreme 64 GB SD Card
- Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996
- Extension Cord
- Raspberry Pi Charger Cable

3.6.5.4.4 Functional Specifications

Raspberry Pi 4B 8GB

The Raspberry Pi 4B is a robust single-board computer featuring a 40-pin GPIO (General Purpose Input/Output) header. This header is utilized to connect the Raspberry Pi to external devices. The GPIO pins can be set as input or output and are used for various applications. They are also capable of generating PWM (Pulse Width Modulation) output. The board supports SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), and UART (Universal Asynchronous Receiver Transmitter) serial communication protocols, as shown in Table 3-26 below:

Table 3-26. Raspberry Pi 4B GPIO Configuration

Pin Type	GPIO Pins
GPIO Pins	GPIO2, GPIO3, GPIO4, GPIO5, GPIO6, GPIO7, GPIO8, GPIO9, GPIO10, GPIO11, GPIO12, GPIO13, GPIO14, GPIO 15, GPIO 16, GPIO 17, GPIO 18, GPIO 19, GPIO 20, GPIO 21, GPIO 22, GPIO 23, GPIO 24, GPIO 25, GPIO 26, GPIO 27
Power Pins	3.3V, 5V, GND
PWM pins	GPIO12, GPIO13, GPIO18, GPIO19
SPI Pins	SPI0: GPIO9 (MISO), GPIO10 (MOSI), GPIO11 (SCLK), GPIO8 (CE0), GPIO7 (CE1) SPI1: GPIO19 (MISO), GPIO20 (MOSI), GPIO21 (SCLK), GPIO18 (CE0), GPIO17 (CE1), GPIO16 (CE2)
I2C Pins	Data: (GPIO2), Clock: (GPIO3) EEPROM Data: (GPIO0), EEPROM Clock: (GPIO1)
UART Pins	TX: (GPIO14) RX: (GPIO15)
Other Pins	ID_SD (I2C ID EEPROM) ID_SC (I2C ID EEPROM)

GPIO pins are used for general-purpose input/output operations. This allows for the control of various digital devices, such as LEDs, buttons, and sensors. These pins also can read data from these devices. On the contrary, PWM pins are designed to generate pulse width modulation signals, which control the amount of power supplied to devices like motors or LEDs by adjusting the signal's duty cycle. Moreover, SPI pins are synchronous serial communication interfaces used for short-distance communication, primarily in embedded systems. It enables full-duplex communication between a controller device and one or more secondary devices.

Additionally, other pins are known as I2C pins, a communication protocol used for short-distance, intra-board communication. It allows multiple low-speed devices to communicate with each other using only two wires. Furthermore, UART, TX, and RX pins are used for Universal Asynchronous Receiver/Transmitter communication. UART is a hardware device or software implementation that translates data between parallel and serial forms. It is used for serial communication between devices. Lastly, other pins include ID_SD, which supports SD card communication, while ID_SC pins support SPI communication. See the figure below to visualize the different GPIO pins discussed above:

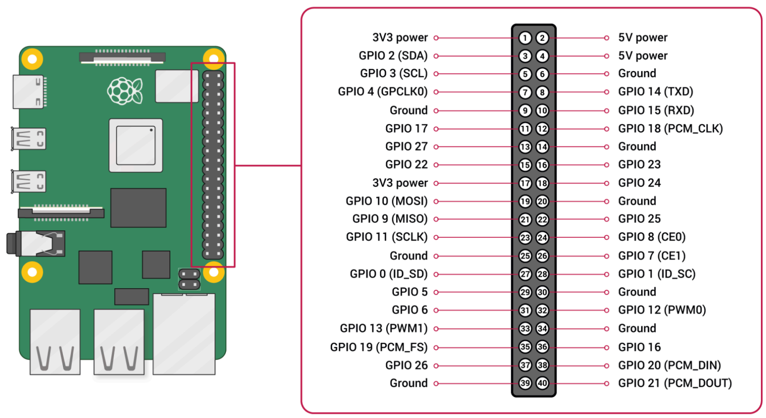


Figure 3-107. Raspberry Pi 4B 40-GPIO Pins

The Raspberry Pi 4 Model B is a high-performance single-board computer introduced by the Raspberry Pi Foundation in June 2019. This model is powered by a 64-bit Broadcom 2711 Cortex A72 processor that operates at a speed of 1.5GHz. This processor is designed to consume 20% less power and deliver 90% greater performance than its predecessor. Table 3-27 provides a detailed overview of the capabilities of the Raspberry Pi 4 Model B. It includes information about the processor, memory, connectivity options, and power requirements, among other details.

Table 3-27. Raspberry Pi 4B Specification

Raspberry Pi 4 Computer Model B Specification	
Specification Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	1GB, 2GB, 4GB, or 8GB LPDDR4 (depending on model) with on-die ECC
Connectivity	2.4 GHz and 5.0 GHz IEEE

Raspberry Pi 4 Computer Model B Specification	
	802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports two × USB 2.0 ports.
GPIO	Standard 40-pin GPIO header (fully backward-compatible with previous boards)
Video & sound	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); and OpenGL ES, 3.0 graphics
SD card support	Micro SD card slot for loading operating system and data storage
Input power	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Operating temperature 0–50ºC

Table 3-27 illustrates the specification of Raspberry Pi 4 Model B, a high-performance single-board computer powered by a 64-bit Broadcom BCM2711 Cortex-A72 processor that operates at 1.5GHz. It comes in three memory configurations: 1GB, 2GB, 4GB, or 8GB LPDDR4, with on-die ECC, and supports a wide range of tasks from simple to complex applications. For connectivity, it supports 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE, and Gigabit Ethernet. It also has two USB 3.0 ports and two USB 2.0 ports, which can connect various peripheral devices. The Raspberry Pi 4 also features a standard 40-pin GPIO header.

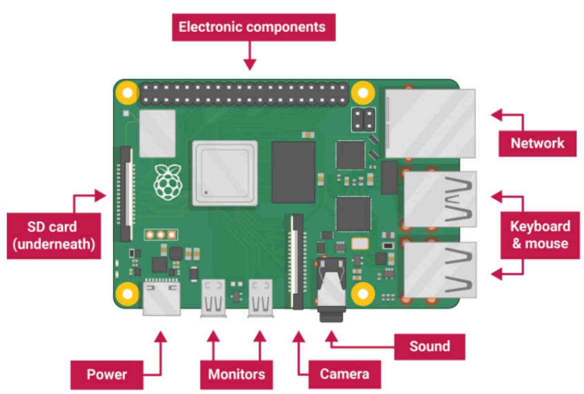


Figure 3-108. Raspberry Pi 4B General Parts

Furthermore, as shown in Figure 3-108, the Raspberry Pi 4B supports 2 × micro HDMI ports (up to 4Kp60 supported), a 2-lane MIPI DSI display port, a 2-lane MIPI CSI camera port, and a 4-pole stereo audio and composite video port. This makes it capable of handling a wide range of multimedia tasks. The Raspberry Pi 4 supports H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode), and OpenGL ES 3.0 graphics. It also has a micro SD card slot for loading the operating system and data storage. The Raspberry Pi 4 can be powered via a 5V DC USB-C connector or a 5V DC GPIO header, providing a minimum of 3A. It also supports Power over Ethernet (PoE) capability, although this requires a separate PoE HAT. The Raspberry Pi 4 has an operating temperature range of 0–50ºC, making it suitable for various environments.

Capacitive Soil Moisture Sensor Digital v2.0

Similarly to the previously proposed moisture sensor, the developers calibrated the sensor for use in this project to evaluate the capacitance of the coffee beans when there’s interaction. The sensor operates under voltages between 3.3 and 5.V, making it compatible with all the major microcontrollers available. The sensor module has an onboard voltage regulator, ensuring the sensor can work under these voltages.

The sensor functions by identifying shifts in capacitance that occur due to changes in the material surrounding it. Instead of gauging moisture directly, it tracks the ions present in the moisture, since pure water itself does not conduct electricity well. The sensor uses a 555-timer circuit to monitor these capacitance changes and produce a corresponding voltage output. This output is then converted into a digital value, which is expressed as a percentage to represent the moisture level detected.

Figure 3-109. Capacitive Soil Moisture Sensor

The sensor has three pins: VCC, GND, and AOUT. The VCC pin powers the sensor and should be connected to the microcontroller's power supply or the power source you are using. The GND pin completes the circuit and should be connected to the ground of the microcontroller or power source. The AOUT pin outputs the analog signal that represents the soil moisture level. This pin should be connected to an analog input pin of the microcontroller.

DHT22

The DHT22 is a handy, cost-effective device that measures humidity and temperature. It's a digital sensor with a built-in analog-to-digital converter, making connecting with the Raspberry Pi easier. However, its sampling rate is relatively slow, updating only once every two seconds. The single data pin can connect directly to the Raspberry Pi's GPIO, making it a convenient choice for projects that require monitoring environmental conditions.

Table 3-28. DHT22 Specification

DHT22 Specification	
Operating Voltage	3.5V to 5.5V
Operating current	0.3mA (measuring) 60uA (standby)
Output	Serial Data
Temperature Range	-40°C to 80°C
Humidity Range	0% to 100%
Resolution	Temperature and Humidity both are 16-bit
Accuracy	±0.5°C and ±1%

Table 3-28 illustrates the specifications for the DHT22 module; its operating voltage is between 3.5V and 5.5V, and the operating current is 0.3mA during measurement and 60uA in standby mode. This sensor outputs both temperature and humidity data through a serial connection. The temperature range is from -40°C to 80°C, and the humidity range is from 0% to 100%. Moreover, both temperature and humidity readings are 16-bit, providing high resolution, and the accuracy readings of the sensor are ±0.5°C for temperature and ±1% for humidity.

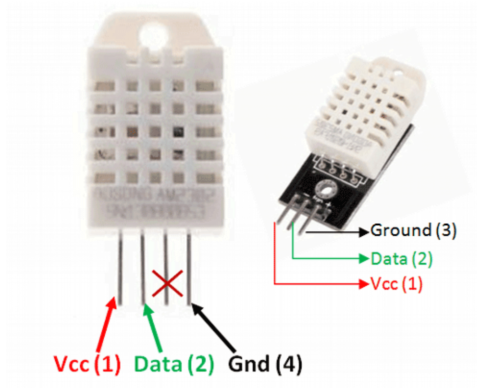


Figure 3-110. DHT22 Pin Configuration

The Vcc pin is the power supply, which requires a voltage between 3.5V and 5.5V. The Data pin is the output pin that transmits temperature and humidity data in a serial format. Lastly, the Ground pin, as its name suggests, is connected to the circuit's ground. This pin configuration is fundamental for the DHT22 module to function correctly.

Food-Grade Stainless Steel

Food-grade stainless steel is widely used in the food and beverage manufacturing and processing industries. It is known for its corrosion resistance, heat resistance, and ability to tolerate moisture and certain chemicals, making it an excellent choice for various food processing applications.

Table 3-29. Food-Grade Stainless Steel Specification

Specification	Value
Series	300 and 400.
304 Grade	Commonly used for dairy, beer, and sanitation.
316 Grade	Used for commercial food production due to better corrosion resistance.
430 Grade	More cost-effective, used in applications involving extended contact with mildly acidic compounds.

Table 3-29 illustrates the specification of Food-Grade Stainless Steel; it is composed of multiple grades known for their corrosion resistance, heat resistance, and ability to tolerate moisture and certain chemicals, making them an excellent choice for various food processing applications. The 300 and 400 series are the two leading stainless steel used in the food industry. Next, the 304 grade is commonly used for dairy, beer, and sanitation due to its ability to withstand corrosion from several oxidizing acids. The 316 grade is used for commercial food production due to its superior corrosion resistance. Lastly, the 430 grade, on the other hand, is more cost-effective and is used in applications involving extended contact with mildly acidic compounds.

Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit

Figure 3-111. Thermoelectric TEC-12706 Peltier

As illustrated in Figure 3-111, a Thermoelectric Peltier module is a thermal control device that can warm and cool surfaces by passing electric current through it. This is achieved through the Peltier effect, where electrons move in one element and positive holes move in the other, allowing one side of the substrate to absorb heat and the other to radiate heat. The module's structure consists of two types of semiconductor elements arranged in tandem sandwiched between copper substrates. Using the Seebeck effect, it can also be used as a thermoelectric power generation module.

Table 3-30. Thermoelectric Peltier Design Specification

Thermoelectric Peltier Design Specification	
Material	Metal
Voltage	12V

Thermoelectric Peltier Design Specification	
Electricity Current	5.8A
Power	50-60W
Weight	350g
Fan Size	92 x 92 x 255 mm

Table 3-30 encompasses the thermoelectric Peltier design specifications, including using metal for its heat conduction properties, operating at 12V with a current draw of 5.8A. The system consumes 50-60W of power and weighs approximately 350g. It incorporates a 92 x 92 x 25 mm fan, essential for optimizing airflow and enhancing heat dissipation. These specifications are fundamental in designing a reliable and efficient thermoelectric cooling system, ensuring effective thermal management across various applications.

12V 16.7A Power Supply

Figure 3-112. 12V Switching Power Supply

A 12V power supply unit is an electronic device with a regulated 12V DC output that converts electrical power from a higher voltage. Its purpose is to provide a maximum current of 16.7A to the Peltier fans and servo motor.

Sandisk Extreme 64 GB SD Card

Figure 3-113. Sandisk Extreme 64gb SD Card

The Sandisk Extreme 64GB SD Card, as shown in Figure 3-113, is a storage solution for Raspberry Pi. It can store the operating system, applications, and data. This SD Card has a storage capacity of 64GB, a transfer rate of 100MB/s, and a read speed of up to 90MB/s. It's UHS-I compliant, meaning it can support high-speed data transfer. The card uses a microSD card form factor, which is compatible with the Raspberry Pi's microSD slot.

SIM800L V2 5V Wireless GSM GPRS Module

The SIM800L V2 5V Wireless GSM GPRS Module is a compact GSM module that can be used in various IoT projects. It supports GSM features such as voice calls, SMS, and GPRS features. The module operates at a voltage level 5V for both VCC and TTL serial levels, enabling direct connection to Arduino or other minimum systems with a 5V voltage level (Ramteke, S., n.d.). The SIM800L module can send SMS messages, make phone calls, and connect to the internet via GPRS. It supports quad-band 850/900/1800/1900 MHz, transmitting voice calls, SMS messages, and low-power data.

Table 3-31. SIM800L V2 5V Wireless GSM GPRS Module Pin Configuration

Pin	Description
VCC	The external Supply Voltage input for SIM800L
GND	External Ground for SIM800L

Pin	Description
VDD	MicroController Supply voltage input for SIM800L
RST	Reset pin for SIM800L
RDX	Serial communication (Receiver Pin)
TXD	Serial communication (Transfer Pin)

The SIM800L GSM module has seven pins (*see Figure 3-114*) used to interface with other components and control its operation. The VCC pin serves as its power supply, necessitating a specific voltage range between 3.4V and 4.4V. Incorrect voltage applications, like using a 5V supply, can lead to module damage, while 3.3V won't allow it to function. A lithium battery or a buck converter with a current capacity of 2A is advised for powering this module. The GND pin is its ground connection, typically linked to the microcontroller's ground pin. The RST pin functions as the reset mechanism; to initiate a hard reset, this pin must be pulled low for 100ms. For serial communication, the module uses the RXD pin to receive data and the TXD pin to transmit data.

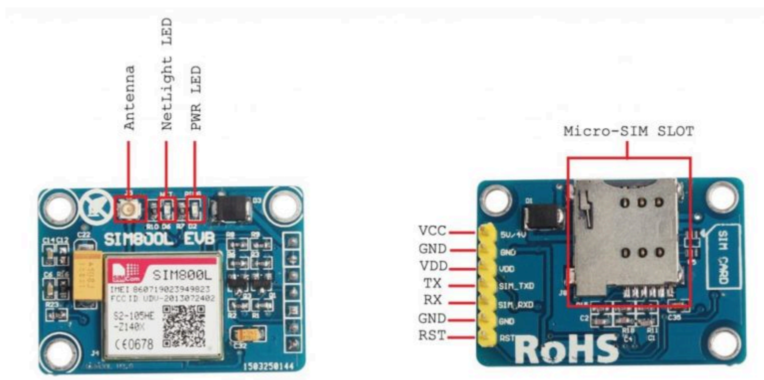


Figure 3-114. SIM800L V2 5V Wireless GSM GPRS Module Pin Configuration

The SIM800L module has an IPX antenna and allows switching between a PCB glue stick antenna and suction cups. It supports four-frequency communications, allowing data connectivity worldwide (Ramteke, S., n.d.).

Figure 3-115. SIM800L V2 5V Wireless GSM GPRS

An external antenna (*see Figure 3-115*) is required to connect to the network. The module typically includes a helical antenna that can be attached via soldering. Additionally, the board features a U.FL connector, allowing the antenna to be positioned away from the main board if desired (Admin, 2022).

Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996

Figure 3-116. Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996

The MG996R servo motor, shown in Figure 3-116, is designed to provide reliable and precise motion control in robotics and automation applications. Its high torque and quick response make it ideal for tasks requiring solid and accurate movements, such as controlling robotic arms, steering mechanisms, and other automated systems. In this project, the MG996R servo motor controls the vent door, enabling it to open and close.

Table 3-32. Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996 Specifications

Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996 Specifications	
Weight	55g
Dimension	40.7×19.7×42.9mm
Stall Torque	9.4kg/cm (4.8v) 11kg/cm (6.0v)
Operating Speed	0.19sec/60 degree (4.8v); 0.15sec/60 degree (6.0v)
Operating Voltage	4.8V - 6V
Gear Type	Plastic Gear
Temperature range	0- 55 deg
Dead band width	1us
Wire length	32cm

The Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R weighs 55 grams and measures 40.7×19.7×42.9mm. It has a stall torque of 9.4 kg/cm at 4.8V and 11 kg/cm at 6.0V. The operating speed is 0.19 seconds per 60 degrees at 4.8V and 0.15 seconds per 60 degrees at 6.0V. This servo motor operates within a voltage range of 4.8V to 6V and uses plastic gears. It functions effectively within a 0 to 55 degrees Celsius temperature range and has a dead bandwidth of 1 microsecond. The wire length is 32 centimeters (*see Table 3-32*).

2-Channel 5V Relay Module

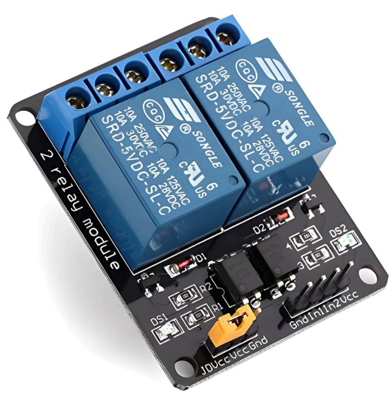


Figure 3-117. 2-Channel 5V Relay Module

The 2-channel 5V relay module is a compact electronic component that allows low-voltage microcontrollers like Arduino or Raspberry Pi to control higher-voltage devices.

Table 3-33. Pin Configuration of 2-Channel 5V Relay Module

<table>
<tr><th colspan="2">Pin</th><th>Description</th></tr>
<tr><td rowspan="2">Power Pins</td><td>VCC (or JD-VCC)</td><td>Connected to the 5V pin of the microcontroller power supply.</td></tr>
<tr><td>GND</td><td>Connected to the ground (GND) of the power supply.</td></tr>
<tr><td rowspan="2">Signal Pins</td><td>IN1</td><td>Receives the signal control to relay 1 module</td></tr>
<tr><td>IN2</td><td>Receives the signal control to relay 2 module</td></tr>
<tr><td rowspan="2">Output Pins</td><td>NO (Normally Open)</td><td>This pin is not connected to COM when the relay is not activated (off-state).</td></tr>
<tr><td>NC (Normally Closed)</td><td>This pin is connected to COM when the relay is not activated (off-state).</td></tr>
<tr><td rowspan="2">Jumper Settings</td><td>Low-Level Trigger (LL)</td><td>The relay is triggered when the input signal is low (0V).</td></tr>
<tr><td>Low-Level Trigger (LL)</td><td>The relay is triggered when the input signal is high (5V or 3.3V, depending on the microcontroller).</td></tr>
</table>

Table 3-33 illustrates the pin configuration of a 2-Channel 5V relay. The power pins VCC and GND connect to the microcontroller to provide power and ground, respectively. Signal pins IN1 and IN2 receive control signals from the microcontroller to activate or deactivate each relay independently. Output pins NO and NC dictate the connection state with COM, providing flexibility in circuit design based on whether the relay is activated. Finally, jumper settings allow users to choose between low-level or high-level triggering based on the microcontroller's output signal characteristics.

Momentary Push Button Switch PBS-110

Figure 3-118. Momentary Push Button Switch PBS-110

The Momentary Push Button Switch PBS-110, shown in Figure 3-116, temporarily connects the normally open contact to the common contact. This action allows for brief circuit activation, commonly used in reset mechanisms or as manual control inputs in electronic devices and consumer electronics. It is used to open and close the system.

Raspberry Pi Charger Cable

Figure 3-119. Raspberry Pi Charger Cable

Figure 3-117 illustrates the Raspberry Pi Charger Cable. The Okdo Raspberry Pi Power Supply includes a 1.5m USB Type-C cable and EU plug, providing a dependable option for powering Raspberry Pi boards. It is designed to support various Raspberry Pi models and ensures sufficient power for stable operation.

Extension Cord

Figure 3-120. Extension Cord

The extension cord shown in Figure 3-118 extends power sources to devices placed beyond the range of standard outlets. This project allows a Raspberry Pi and two power supplies to access power from a distant location within the building or area. This feature is handy in environments where flexibility and reconfiguration of electrical setups are necessary, improving operational efficiency and convenience.

3.6.5.4.5 Bill of Materials

Due to the developers' cost constraints, below is the summary bill of materials for Design Option 3. Every item that the developers should consider, from the cover materials and fans to be used to the sensors, is listed below to ensure that this design option is within the project's cost constraints.

Table 3-34. Design Option 3 Bill of Materials

Item	Qty.	Price/pc. (₱)	Total Price (₱)
Raspberry Pi 4B 8GB	1	5907	5907
Capacitive Soil Moisture Sensor v2.0	1	75	75
DHT22 Module	3	200	600
Food-Grade Stainless Steel (1.2m x 1m)	3	5000	15000

Item	Qty.	Price/pc. (₱)	Total Price (₱)
Food-Grade Stainless Steel (1.2m x 0.4m)	1	1000	1000
Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit	2	248	496
SIM800A Module	1	355	355
12V 16.7A Switching Power Supply Unit	2	875	1750
Sandisk Extreme 64 GB SD Card	1	250	250
Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996	1	170	170
2-channel 5V Relay Module	2	79	158
Momentary Push Button Switch	1	65	65
Raspberry Pi Charger Cable	1	300	300
Extension Cord	1	250	250
Total (₱)			26376

The table above shows the estimated cost of Design Option 3 based on the required quantity for each component, and the probability of encountering difficulties or defects resulting from the purchase of new materials is beyond the developers' control from the production stage to its implementation.

3.6.5.5 Testing, Validation, and Report

Similarly to the previous two design options, testing, validation, and report, the developers decided to test and confirm the Performance constraint in SolidWorks and measure Power Consumption using theoretical mathematical formulas. They did this because real-life response time testing and validation are easier than virtual simulation.

3.6.5.5.1 Performance

To achieve the desired parameters for the device, developers first set the desired parameters in SolidWorks, just like in previous design options testing. In this case, the type of material to be used, the kind of testing to be conducted, whether the testing is for the outside or inside of the casing, and the initial temperature. Below is the discussion of each procedure:

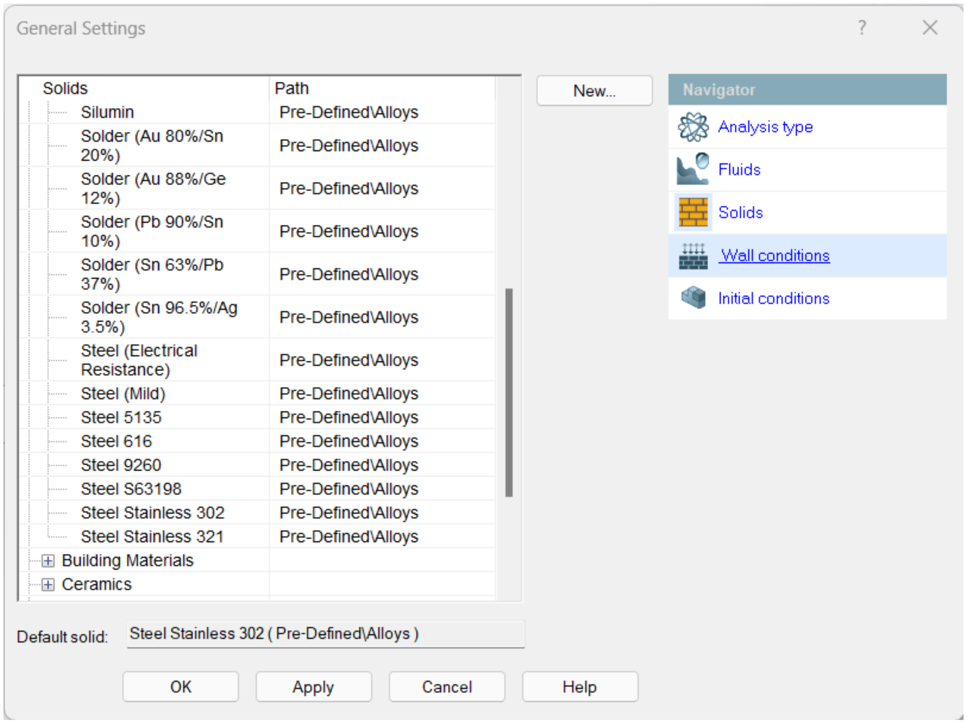


Figure 3-121. Design Option 3 Wall Conditions

Before conducting the flow simulation, determining the wall conditions was necessary. This can be done by going to “General Settings” and selecting “Wall Conditions”, as displayed in Figure 3-121. Wall conditions determine how fluid interacts with the solid boundaries in the simulation. It also helps to represent real-world scenarios accurately accurately. Different wall conditions have an impact on the flow behavior. Under the Glass and Minerals, Stainless Steel 30)was chosen in this case since Design Option 3 is fully stainless steel in cover.

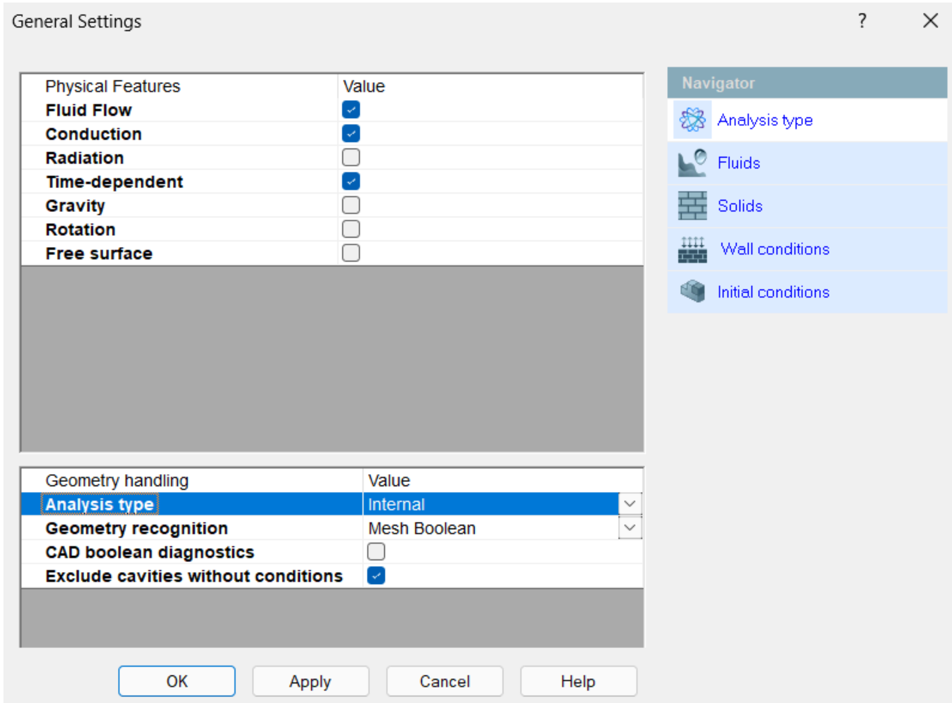


Figure 3-122. Design Option 3 Analysis Type

The second thing to consider is the Analysis Type, as presented in Figure 3-122. Analysis type determines how the simulation models the interaction between the fluid domain and the surrounding environment. In Design Option 3, the internal analysis type focuses on the enclosed fluid space. Internal analysis allows for more precise modeling of fluid flow within enclosures. This flexibility allows for accurate simulation of real-world scenarios.

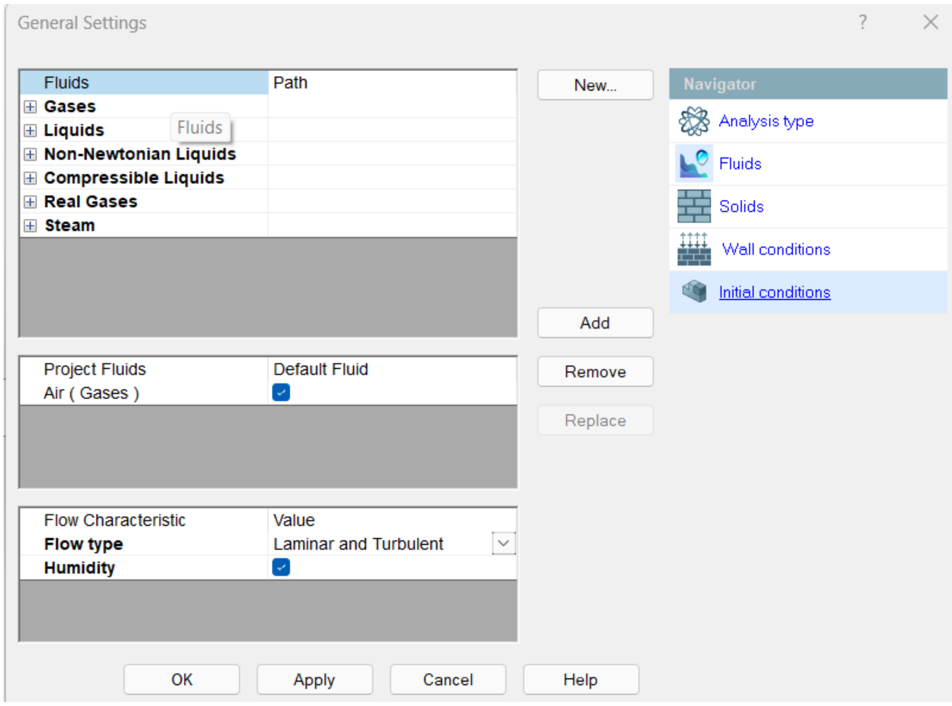


Figure 3-123. Design Option 3 Fluids

Design Optio 3 “Fluids” is similar to the two previously discussed design options where Air (Gases) was set to true, Flow Type was set to Laminar and Turbulent, and Humidity was set to true (*see Figure 3-123*).

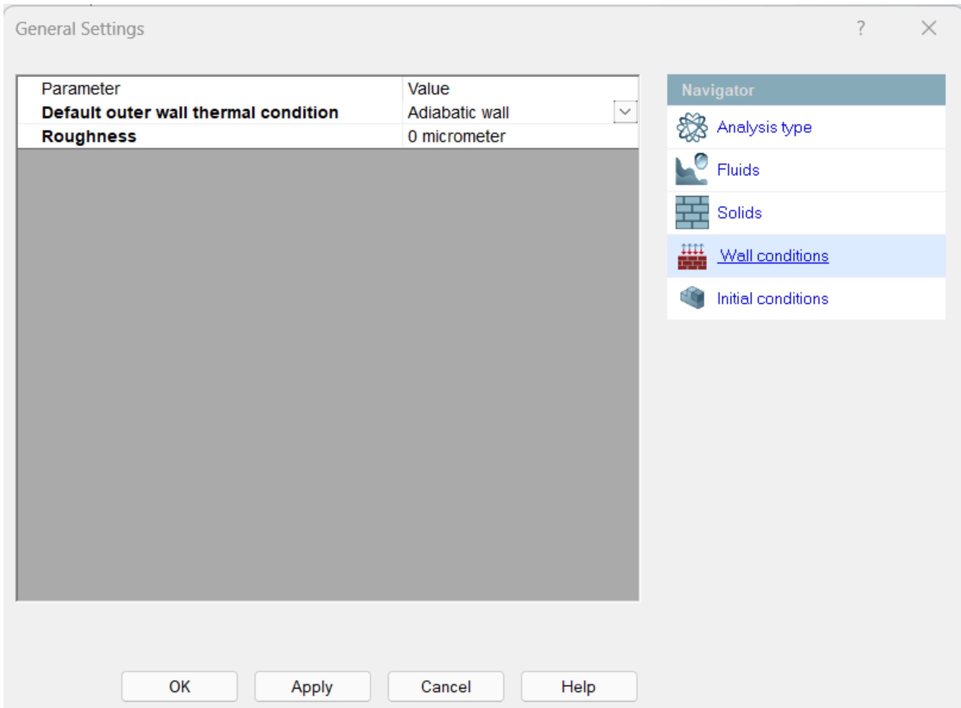


Figure 3-124. Design Option 3 Solids

Figure 3-124 presented the Design Option 3 Solids conditions, with the outer wall thermal condition set to Adiabatic Wall and the roughness set at 0 micrometer.

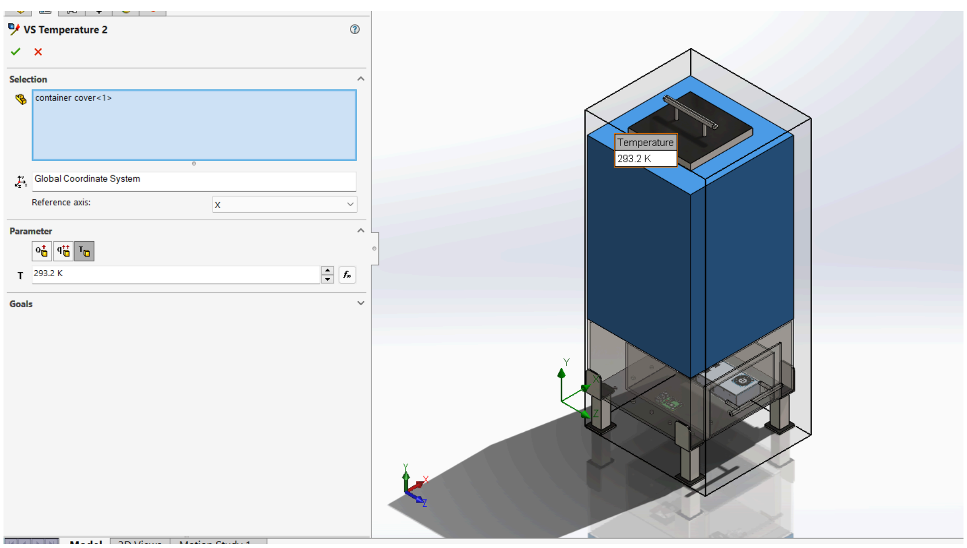


Figure 3-125. Design Option 3 Volume Source Temperature

The "Volume Source Temperature" is used to specify the temperature of a heat source in a computational fluid dynamics (CFD) simulation. It allows defining a region of the model where heat is added to the fluid flow. As shown in Figure 3-125, the volume source temperature of Design Option 3 is inside and within the overall cover material.

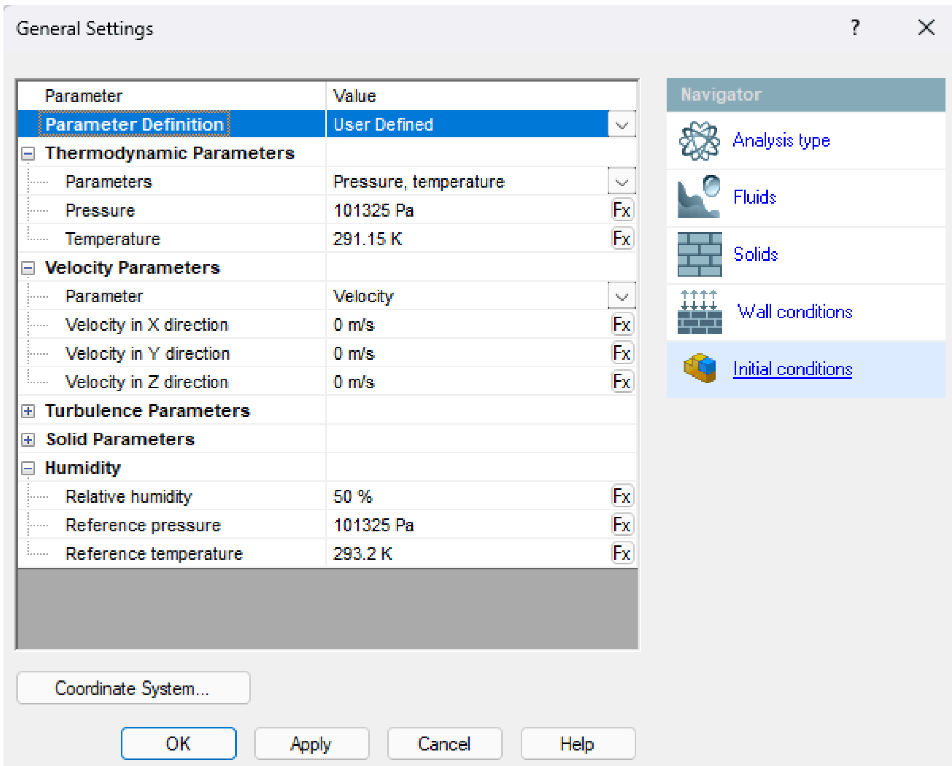


Figure 3-126. Design Option 3 Initial Conditions

Figure 3-126 shows the initial conditions for this design option. The parameters were set to "Pressure, Temperature." Pressure has an initial value of 101325 Pa, and Temperature is 291.15 K. Under "Humidity," relative humidity is 50%, Reference pressure is 101325 Pa, and Reference temperature is 293.2 K.

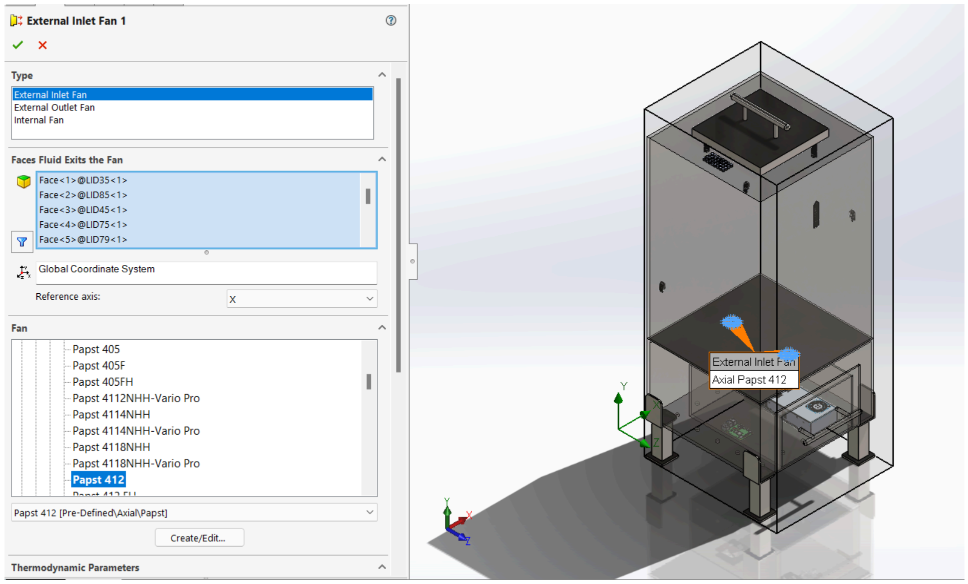


Figure 3-127. Design Option 3 Type of Fan

Moving forward, the Type of Fan was also determined. Along with the wall conditions, the type of fan impacts how long the flow simulation could last. Design Option 3, as displayed in Figure 3-127, has an External Inlet Fan of "Papst 412," similar to Design Option 1.

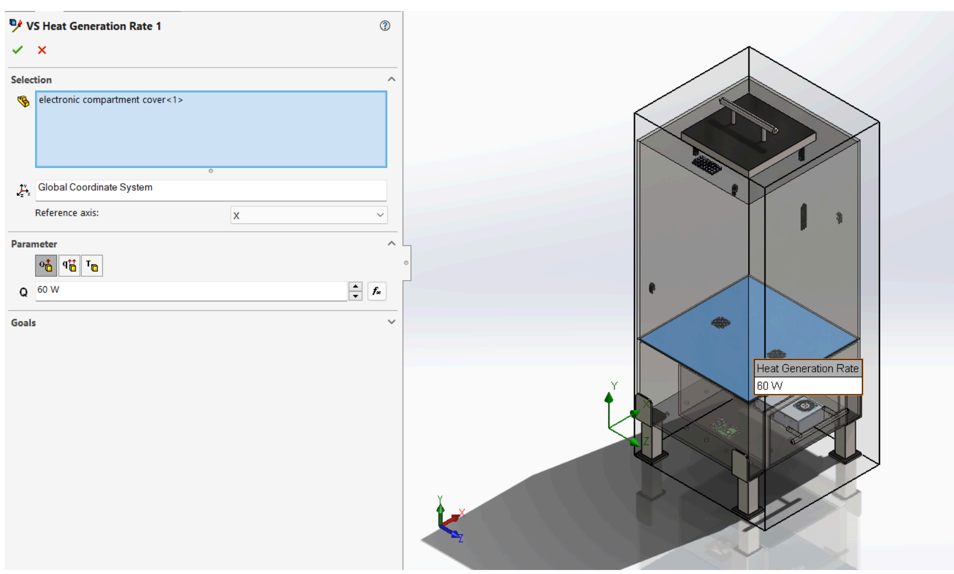


Figure 3-128. Design Option 3 Heat Generation Rate

Once the type of fan is set, it's required to set its heat generation rate, as depicted in Figure 3-128. Heat Generation Rate refers to the amount of heat energy produced per unit volume over a given period of time, which is usually expressed in units like Watts per cubic meter (W/m3).

In this Design Option, the fan's heat generation rate was 60W, the maximum power of a Thermoelectric Peltier.

Once the parameters required for the intended simulation are set and identified, the flow simulation can begin. It is crucial to monitor the design to determine whether it is feasible and can effectively sustain the temperature and humidity levels during the flow simulation.

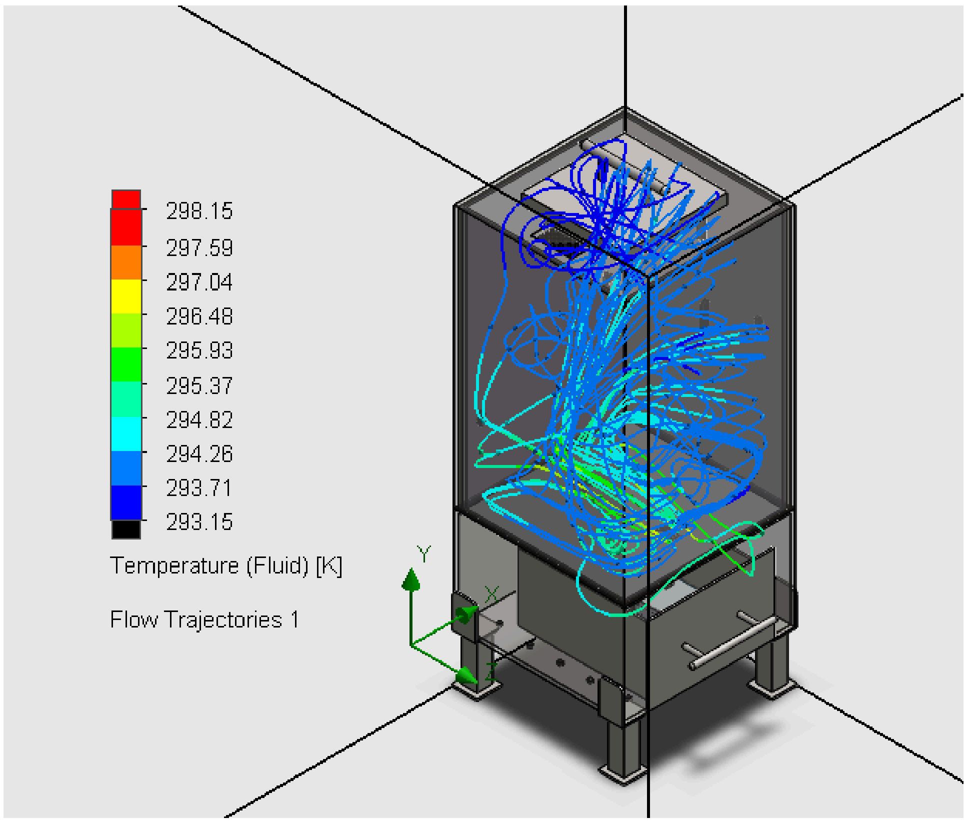


Figure 3-129. Design Option 3 Flow Trajectories Result

Figure 3-129 illustrates the Flow Trajectories of Design Option 3. The color gradient, shifting from red to green, represents temperature changes within the fluid, ranging from 297.59 K to 293.15 K, providing insight into how temperature impacts fluid flow. The blue lines depict the fluid's flow trajectories, demonstrating the movement patterns within the enclosure, which can be crucial for optimizing engineering designs. Additionally, the figure offers valuable thermal analysis, which is essential for designing efficient cooling systems or understanding heat transfer in various applications.

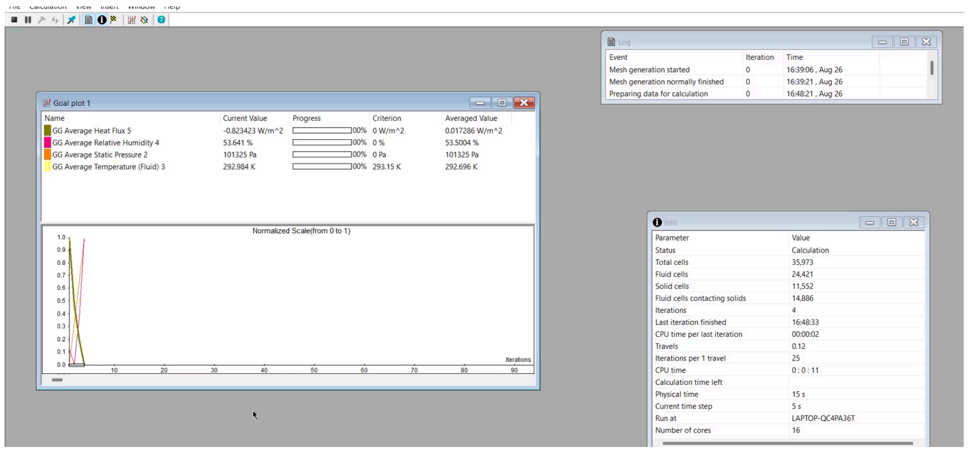


Figure 3-130. Design Option 3 Flow Simulation Start

Figure 3-130 displays the initial setup screen for a computational fluid dynamics (CFD) simulation, likely used to evaluate airflow and temperature distribution for the dotBean Design Option 3. The left panel outlines various monitored parameters, including "Average Relative Humidity," "Average Temperature," and "Mass Flow Rate," among others. Each parameter is accompanied by columns indicating its "Current Value," "Progress," "Criterion," and "Averaged Value," all of which are currently at 0%, signifying that the simulation has just started or has not yet begun.

A list of "Parameter" and "Value" pairs on the right panel offers additional insights into the simulation's configuration. This includes the number of cells being utilized ("Total cells: 471,734"), the breakdown of solid, fluid, and wall cells, and other specifics such as "Solver iterations finished," which is set to 0, indicating that no iterations have been completed yet. The "Run" section shows the name of the computer executing the simulation, marking this as the beginning of the computational analysis.

This setup represents the initial phase of the flow simulation, which is crucial for understanding the airflow dynamics within the temperature-controlled storage unit. By configuring the simulation with these parameters, the software can calculate the intricate interactions of air movement, temperature distribution, and humidity levels, which are essential for optimizing the design to maintain the ideal environment for storing green coffee beans. The primary objective is to ensure that the storage conditions remain consistent and controlled, preserving the quality of the beans over time.

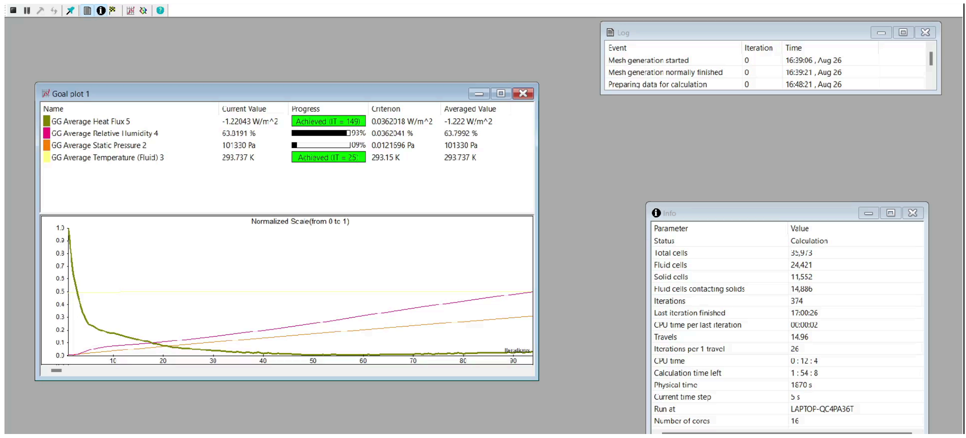


Figure 3-131. Design Option 3 Flow Simulation on Progress

In the Design Option 3 Flow Simulation currently in progress, as shown in Figure 3-131, the interface is divided into two software windows with distinct functions. The left window features a list of scenarios detailing 'Run Name,' 'Status,' 'Progress,' 'Elapsed Time,' and 'Remaining Time.' Below this list is a graph with multiple lines labeled "Normalized Scaphoid 1 to 3," which aids in visualizing the simulation's progress and results over time. The right window displays numerical data under the "Iteration" and "Residuals" tabs, showing parameters such as 'Time Step,' 'Max Upstream Peclet Number,' and various residuals for the equations. This panel provides detailed numerical data for assessing the simulation's accuracy and performance. Together, these elements facilitate real-time monitoring and analysis of fluid dynamics simulations, offering valuable insights into the ongoing process.

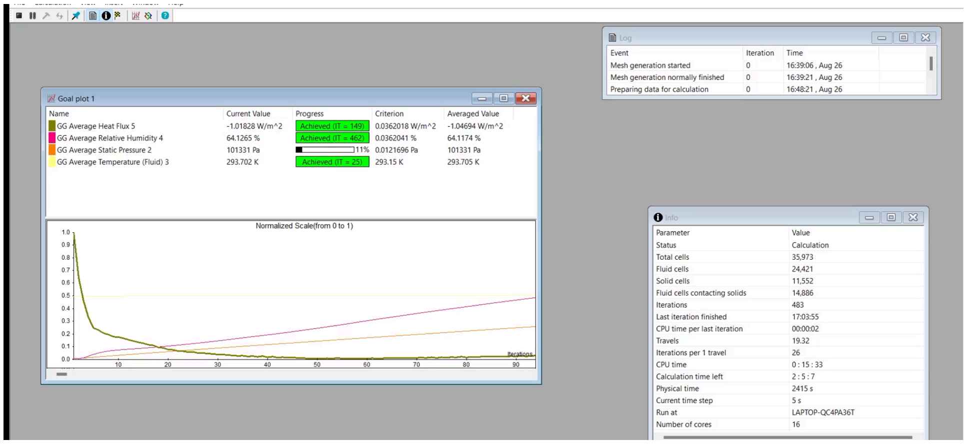


Figure 3-132. Design Option 3 Flow Simulation Finished

Lastly, the interface presents several critical elements in the completed Design Option 3 Flow Simulation, as Figure 3-132 presents. On the left side, tabs for ‘Project,’ ‘Prepare,’ and ‘Results’ are available, with the ‘Results’ tab active. This tab offers options like ‘Goal Plot,’ ‘Surface Plot,’ and ‘Flow Trajectories,’ while below these options, numerical summaries provide details such as an Average Temperature of 23.5°C, a Maximum Velocity of 0.003 m/s and an Average Static Pressure of -343 Pa. The central part of the display features a graph with multiple colored lines that represent different variables over time or calculation steps, labeled “Numerical Calculation (s).” Additionally, two smaller windows on the proper side showcase settings and parameters, including a Time Step of 1e-003 s, Physical Time of 1e+000 s, and Iteration Steps of 1000.

The figure below shows the summary result of the conducted flow simulation for this design option.

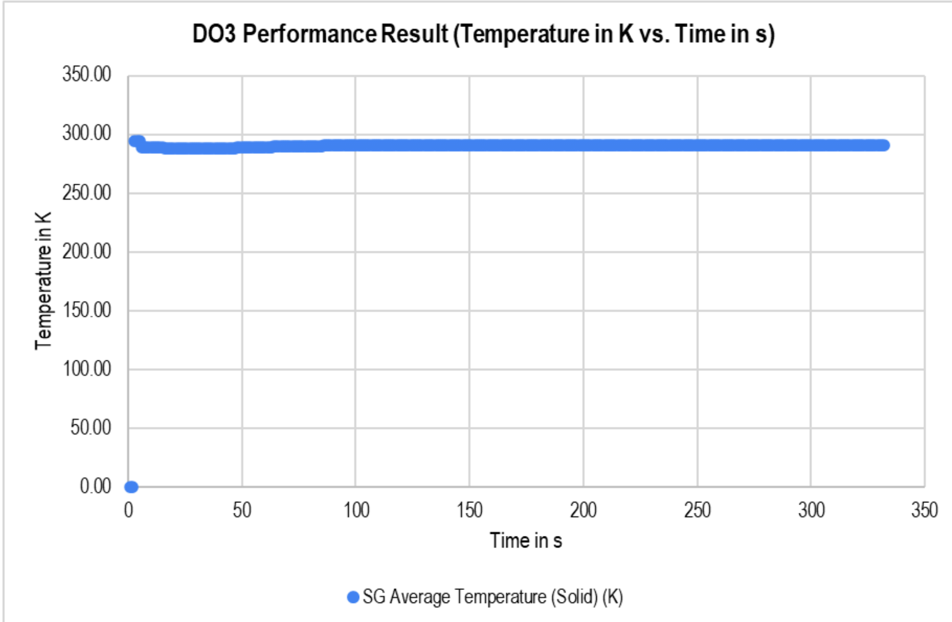


Figure 3-133. Design Option 3 Performance Result (Temperature in K vs. Time in s)

Figure 3-133 illustrates the result of the flow simulation conducted for design option 1 in SolidWorks. During the simulation process, the system starts at 288.15 K or 15°C with minimal changes in temperature value increased and reached 293.15 K or 20°C, suggesting that it takes 16 minutes and 12 seconds to complete the simulation reaching 20°C from 15°C.

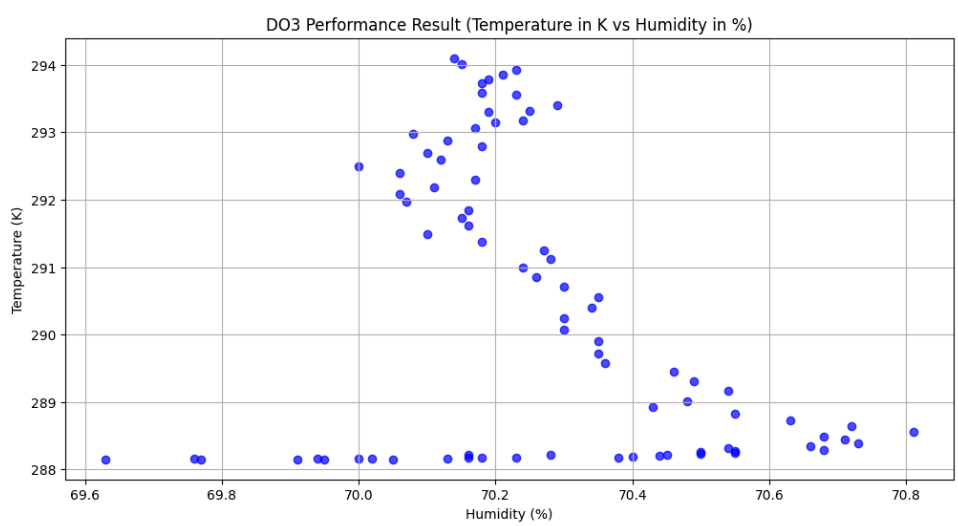


Figure 3-134. Design Option 3 Performance Result (Temperature in K vs Humidity in %)

Figure 3-134 presents the Temperature graph in Kelvin (K) vs. Humidity in percentage (%). The graph indicates that when the temperature changes, the humidity also changes. From 288.15 K, where the humidity levels varied, it maintained the required humidity level of not exceeding 70% when it reached 293.15 K.

3.6.5.5.2 Response Time

In this testing, the developers evaluated the fan's response time to turning on or off when instructed by the Raspberry Pi. Since the system is automated, it must work reliably and correctly, even without direct configuration or user contact.

```
# Record the start time
start_time = time.time()
fan_status = "OFF" if GPIO.input(RELAY_PIN) == GPIO.LOW else "ON"
# Fan control logic
if avg_temperature_c < 20:  # Condition for turning the fan on
print("Fan ONN")
GPIO.output((RELAY_PIN, RELAY_PIN1), GPIO.HIGH)  # Turn the fan on
else:  # Default case where the fan should be off
print("Fan OFF")
GPIO.output((RELAY_PIN, RELAY_PIN1), GPIO.LOW)  # Turn the fan off
 # Record the end time
 end_time = time.time()
# Print the averaged values to the console
current_datetime = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
message = (f"Avg Temp: {avg_temperature_c:.1f} C  Humidity: {avg_humidity:.1f}%
Moisture: {line}\nDate and Time: {current_datetime}")
print(message)
# Calculate the duration in seconds with millisecond precision
duration = end_time - start_time
print(f"Time taken to send the command: {duration:.4f} seconds")
```

Figure 3-135. Design Option 3 Response Time Command in Raspberry Pi

The code snippet, presented in Figure 3-135, is designed to manage a fan based on temperature readings and log relevant operational details. It begins by recording the start time to measure the duration of the fan control operation. It then checks the current state of the fan using a GPIO pin to determine whether the fan is "ON" or "OFF." The core functionality involves controlling the fan based on the average temperature. If the temperature is below 20°C, the fan is turned on by setting the GPIO pin to `GPIO.HIGH`, and the console prints "Fan ONN" (with a minor typo). If the temperature is 20°C or higher, the fan is turned off by setting the GPIO pin to `GPIO.LOW`, and the console prints "Fan OFF." After executing the fan control logic, the code records the end time and calculates the duration of the process. Finally, it prints the average temperature, humidity, moisture levels, current date and time, and the time taken to execute the fan control command, providing a comprehensive view of the fan's operation and environmental conditions.

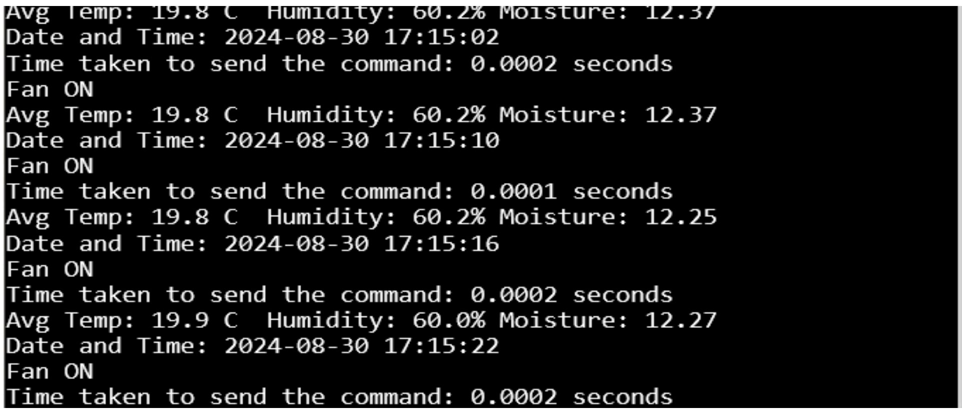


Figure 3-136. Design Option 3 Raspberry Pi Sample Terminal Output Display

During the testing, the system readings and fan status were displayed in the Raspberry Pi's terminal. On the other hand, as shown in Figure 3-136, it took 0.0002 and 0.0001 seconds for the Raspberry Pi to command the fan to stay turned on since the temperature was below 19.9°C. The fan will stay on its state since the temperature reading is still below 20°C.

Table 3-35. Design Option 3 Response Time Results in microseconds

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry Pi received the Reading in microseconds (μs)*	*Time the Fan Activated in microseconds (μs)*	*Response Time in microseconds (μs)*
1	19.4	On	100	300	200
2	20	Off	200	400	200
3	20	Off	100	300	200
4	20	Off	100	300	200
5	19.9	On	200	400	200
6	20.1	Off	100	300	200
7	20.1	Off	100	300	200

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry Pi received the Reading in microseconds (µs)*	*Time the Fan Activated in microseconds (µs)*	*Response Time in microseconds (µs)*
8	20.1	Off	100	300	200
9	20.1	Off	100	300	200
10	20	Off	100	200	100
11	19.8	On	300	600	300
12	20	Off	300	500	200
13	20	Off	100	300	200
14	20	Off	100	300	200
15	19.9	On	200	400	200
16	20	Off	100	300	200
17	20	Off	200	400	200
18	20	Off	100	300	200
19	20.1	Off	100	300	200
20	20	Off	200	400	200
21	19.9	On	300	500	200
22	20	Off	100	300	200
23	20.1	Off	100	200	100
24	20	Off	100	200	100
25	20.1	Off	200	400	200
26	20	Off	200	400	200
27	20	Off	200	400	200
28	20.1	Off	200	300	100
29	19.8	On	300	500	200
30	20	Off	100	300	200
31	20	Off	100	300	200
32	20	Off	300	500	200
33	20	Off	200	300	100
34	20	Off	100	300	200
35	19.9	On	100	300	200
36	20.1	Off	200	400	200

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry Pi received the Reading in microseconds (µs)*	*Time the Fan Activated in microseconds (µs)*	*Response Time in microseconds (µs)*
37	20.1	Off	100	300	200
38	20	Off	100	300	200
39	20	Off	200	400	200
40	20	Off	200	400	200
41	19.8	On	300	500	200
42	20.1	Off	100	300	200
43	20.2	Off	100	300	200
44	20.2	Off	100	300	200
45	20.2	Off	100	300	200
46	20.1	Off	100	300	200
47	20	Off	100	500	400
48	20.1	Off	100	300	200
49	20.1	Off	100	300	200
50	20.2	Off	100	300	200
Average Response Time					**196**

Table 3-35 shows the response time testing result of design option 3. The tests were done by getting the time the Raspberry Pi receives the reading and the time the fan is activated in microseconds. The developers did 50 tests to get the average response time, and based on the response time testing results, it has an average of 196 microseconds.

3.6.5.5.3 Power Consumption

To achieve this, the developers first identified the Voltage (V) and Current (I) of each component and subsequently multiplied their values to determine the component's Power (P) in Watts (W). Datasheets of each component were used, as referenced from different online marketplaces and articles, to complete this computation.

According to (Makerlab Electronics, n.d.-c), Raspberry Pi 4B 8GB has an operating voltage of 5V at 3A. Arduino Uno R3, on the other hand, has a voltage of 5V at 0.02A (Arduino Tips, Tricks, and Techniques, 2024). Additionally, the Resistive Soil Moisture Sensor has a voltage of 5V and a current of 0.02A ("Soil Moisture Sensor Module," 2020). Moreover, the DHT22 Module has a voltage and current of 3.3V and 0.0025 mA, respectively (LME Editorial Staff, 2022). The Thermoelectric Peltier has a 12V voltage and a current of 5.8 (Makerlab Electronics, n.d.-d). The GSM module has a voltage of 3.3V at a current of 2A, according to Future Electronics Egypt (n.d.) On the other hand, a Switching Power Supply Unit has a voltage of 12V and 16.7A (Makerlab Electronics, n.d.-e). Furthermore, Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996 has a voltage of 5V and a current of 0.0025A (Makerlab Electronics, n.d.-e). Lastly, a 2-channel 12V-5V relay module has a voltage of 5V and a current of 10A (Makerlab Electronics, n.d.-a), while a Momentary Push Button, as stated by Smoot (2022), has a voltage of 5V and a 1A current.

Table 3-36 illustrates the procedure of determining each P:

Table 3-36. Design Option 3 Power Consumption in Wattage (W) and Kilowatt-hour (kWh) Result

ITEMS	Qty.	V	I	P (in W)	kWh (1-hr)
Idle State					
Raspberry Pi 4B 8GB	1	5	3	15	0.015
Arduino Uno R3	1	5	0.02	0.1	0.0001
Capacitive Soil Moisture Sensor v2.0	1	5	0.03	0.1500	0.00015
DHT22 Module	3	5	0.005	0.0750	0.000075
SIM800A Module	1	3.3	2	6.6000	0.0066
Tower Pro Digital Robot Servo Motor (180 Rotation) – MG996R MG996	1	5	0.0025	0.0125	0.000013
Active State					
Thermoelectric TEC-12706 Peltier Cooling System Heatsink Kit	2	12	5.8	139.2000	0.1392

ITEMS	Qty.	V	I	P (in W)	kWh (1-hr)
2-channel 5V Relay Module	1	5	10	50.0000	0.05

Table 3-36 shows the result of the design option 3's power consumption in wattage. Design option 3 utilizes a thermoelectric Peltier and vent to monitor the temperature-and-humidity-related activities. The table above shows all the items used in the design, and each item has its value in volts and ampere, which was used to calculate the value of power in watts. The Raspberry Pi consumes 0.015 kWh, the Arduino uno consumes 0.0001 kWh, the capacitive soil moisture with the quantity of 1 contains 0.00015 kWh, the DHT22 Module with the quantity of 3 consumes 0.000075 kWh, and the GSM module contains 0.0066 kWh. Furthermore, the Thermoelectric TEC-12706 Peltier contains 0.1392 kWh, the Tower Pro Servo Motor contains 0.000013 kWh, and the 2-channel 5V Relay Module contains 0.05 kWh.

The developers have excluded the computation of the momentary push button because it doesn't consume significant power, with the electrical current involved being minimal—usually in the microampere range—since it is only used to detect state changes. The power supply is also excluded, as it doesn't add any extra load; instead, it supplies power to the connected devices. This means the power consumed by the power supply from the outlet is simply the total of the power used by all connected devices, plus the minimal inefficiency caused by heat and other factors inherent in the power supply.

Table 3-37. Design Option 3 Power and Energy Consumption Result

Power and Energy Consumption	State	
	Idle (at use)	Active
Total Power (W)	21.9375	211.1375
Total Kilowatt-hour (kWh) (1 hr)	0.0219	0.2111

Table 3-37 summarizes the power consumption for the device Design Option 3, categorized into two states: idle and active. In the idle state, the device consumes 21.9375 watts of power, reflecting the energy it requires while not actively performing tasks but remaining on standby. During active use, the power consumption increases to 211.1375 watts, demonstrating the additional energy needed to perform tasks. Power consumption over one hour is measured in kilowatt-hours (kWh), with the device using 0.0219 kWh in idle mode and 0.2111 kWh in active mode.

3.6.5.5.4 Public Health and Safety

The third design option utilizes 314 food-grade stainless steel as the cover material, providing a highly durable and hygienic solution for storing coffee beans. Stainless steel is renowned for its excellent corrosion resistance, non-reactivity, and ease of cleaning, making it widely used in food processing and storage industries. Its non-porous surface prevents contamination and facilitates easy cleaning, ensuring the stored coffee beans remain contaminant-free (Thomas et al., 2018). While stainless steel provides moderate thermal insulation compared to foams, its primary advantages lie in its superior hygiene and durability, which are crucial for the long-term storage of food products like coffee beans (Johnson & Lee, 2017). This makes 314 food-grade stainless steel suitable for environments where hygiene and durability are prioritized over insulation.

3.6.5.5.5 Social Welfare

For Design Option 3, the developers create a storage device that accurately monitors the temperature and humidity inside the device, ensuring that temperature is maintained within 20° C - 25.4° C and the humidity at 50% - 70%. The moisture content of the green coffee beans should also be within the specific range of 11% to 12.5%, not exceeding 14%. The conducted flow simulation suggests that it can take 16 minutes and 12 seconds to complete the simulation, reaching 20°C from 15°C. For the response time, the developers assessed how long the system took before the fan turned on or off. The testing concluded 50 tests and got an average response time of 196 microseconds. Lastly, for the power consumption, based on the calculated power for the idle (in use) and active state, the result for power consumption has a total of 0.0219 kWh in the idle state and 0.2111 kWh in the active state

3.6.5.5.6 Economics

For Design Option 3, the developers employed a similar approach by preparing a Bill of Materials (BOM) with a budget of 50,000 PHP, as shown in Table 3-34. The developers focused on selecting components readily available in the Philippines to ensure cost-effectiveness and minimize logistical challenges. The developers avoided additional import costs and potential delays by using locally sourced materials. This process enabled the developers to manage material procurement and cost challenges effectively, ensuring the project remained within its financial constraints.

3.6.5.5.7 Social

For the customer's selection, the developers designed a storage device that can be used by all small-scale coffee farmers and coffee shop owners who currently use traditional storing processes in all regions experiencing lower temperatures. Coffee farmers are having issues preventing the detrimental effects of excess moisture absorption for the green coffee beans that typically occur during low temperatures.

3.6.5.5.8 Global

For Design Option 3, the developers implemented a GSM module for notifications through SMS. The GSM module is FCC-certified, meaning it complies with the Federal Communications Commission's standards for electromagnetic interference, ensuring it is safe and reliable. SMS was chosen as the notification method because it offers a straightforward approach that requires only a cellular signal and does not necessitate user logins, simplifying user notification. Like Design Options 1 and 2, all components for this design were sourced locally, primarily from Makerlab Electronics, which provides both a physical store in Manila and an online shopping platform with nationwide shipping. The power consumption of this design was thoroughly tested and validated, confirming that the device operates with low energy usage, consistent with the project’s emphasis on efficiency and cost-effectiveness.

3.6.5.5.9 Cultural

In Design Option 3, the developers focused on using stainless steel for its durability and suitability for food storage, ensuring that the device enhances the storage process without impacting the traditional pre- and post-storage practices. This design option is tailored to fit seamlessly into the farmers' existing workflow, improving storage conditions while preserving the integrity of established methods. By engaging with local farmers and understanding their needs, the developers ensured that the device was practical, culturally sensitive, and aligned with the community's values. Using locally accessible materials and components further ensures the device is easy to maintain and supports sustainable, long-term use in the region.

3.6.5.5.10 Environmental

Stainless steel is used in this design option for its excellent corrosion resistance, mechanical properties, and suitability for food-related applications (Moon et al., 2020). Stainless steel's durability and potential antimicrobial properties make it a robust choice for ensuring the quality and safety of coffee beans. The power consumption for this design is also kept low, as suggested in Table 3-36 and Table 3-37. Similar to the other designs, this option features a modular approach, allowing for procuring and replacing individual components as needed, thereby minimizing environmental impact and facilitating repairs.

3.6.5.6 Constraints and Results Summary

Following the testing to assess the effectiveness of this design option, the table below summarizes the design constraints, their corresponding results, and relevant remarks.

Table 3-38. Design Option 3 Constraints and Results Summary

Constraints	Results	Remarks
Performance	16 minutes and 12 seconds	Design option 3's total time to reach the minimum threshold value at 20°C was 16 minutes and 12 seconds, well below the criteria of within 30 minutes.
Response Time	196 microseconds	Design option 3's total average response

Constraints	Results	Remarks
		time from the 50 testings was 194 microseconds or 0.000196 seconds, which achieved the criteria of less than 1 second.
Power Consumption	0.0219 kWh at idle state 0.2111 kWh at active state	Design option 3's power consumption when in the idle (at use) state and active state achieved the criteria of not consuming more than 1 kWh.

Design option three effectively met all the required criteria during testing. The system demonstrated strong performance by swiftly reaching the optimal temperature when it dropped below 20°C. Its response time of 196 microseconds was 2 microseconds faster than design option 2, although still slightly slower than the first option. Despite this, it maintained reliable control of the automated fan system. Furthermore, design option 3 showed good energy efficiency, consuming only 0.0219 kWh in idle mode and 0.2111 kWh in active mode, which are lower than the energy consumption of the first design option.

In summary, design option 3 is a robust solution that satisfies all performance, response time, and power consumption constraints. Its capacity to rapidly achieve the desired temperature, quickly adapt to system changes, and maintain low energy consumption makes it a dependable and energy-efficient choice for environments that require precise temperature control.

3.7 Summary of All Design Options Results

Upon completing the testing of the design options, the developers gathered and analyzed the results. The following table presents the summarized data:

Table 3-39. Design Options Summary Testing Results

Design Option	Constraints	Result	Remarks
Design Option 1	Performance	21 minutes and 05 seconds	Design option 1's total time to reach the minimum threshold value at 20°C was 21 minutes and 05 seconds, well below the criteria of within 30 minutes.

Design Option	Constraints	Result	Remarks
Design Option 2		16 minutes and 56 seconds	Design option 2's total time to reach the minimum threshold value at 20°C was 16 minutes and 56 seconds, well below the criteria of within 30 minutes.
Design Option 3		16 minutes and 12 seconds	Design option 3's total time to reach the minimum threshold value at 20°C was 16 minutes and 12 seconds, well below the criteria of within 30 minutes.
Design Option 1	Response Time	194 microseconds	Design option 1's total average response time from the 50 testings was 194 microseconds or 0.000194 seconds, which achieved the criteria of less than 1 second.
Design Option 2		198 microseconds	Design option 2's total average response time from the 50 testings was 198 microseconds or 0.000198 seconds, which achieved the criteria of less than 1 second.
Design Option 3		196 microseconds	Design option 3's total average response time from the 50 testings was 196 microseconds or 0.000196 seconds, which achieved the criteria of less than 1 second.
Design Option 1	Power Consumption	0.3241 kWh	Design option 1's power consumption when in the idle (at use) state and active state achieved the criteria of not consuming more than 1 kWh.
Design Option 2		0.1853 kWh	Design option 2's power consumption when in the idle (at use) state and active state achieved the criteria of not consuming more than 1 kWh.
Design Option 3		0.2111 kWh	Design option 3's power

Design Option	Constraints	Result	Remarks
			consumption at idle (at use) state and the active state achieved the criteria of not consuming power more than 1 kWh.

Table 3-39 shows all of the results done per design options. Design Option 1 has a performance value of 21.05 min., an average response time of 194μs, and an average power consumption of 0.3241 kWh. Design Option 2 has a performance value of 16.56 min., an average response time of 198μs, and an average power consumption of 0.1853 kWh. Lastly, Design Option 3 has a performance value of 16.12 min, an average response time of 196μs, and an average power consumption of 0.2111 kWh.

3.7.1 Performance Result Summary

The performance output of each design option was determined using the Flow Simulation of the SolidWorks application. The developers utilized this application as it gives the possible outcome of a mechanical design without needing it to be developed physically. The developers chose to test the design options' performance constraints as this testing offers cost-effectiveness. Below are the results of the conducted simulations:

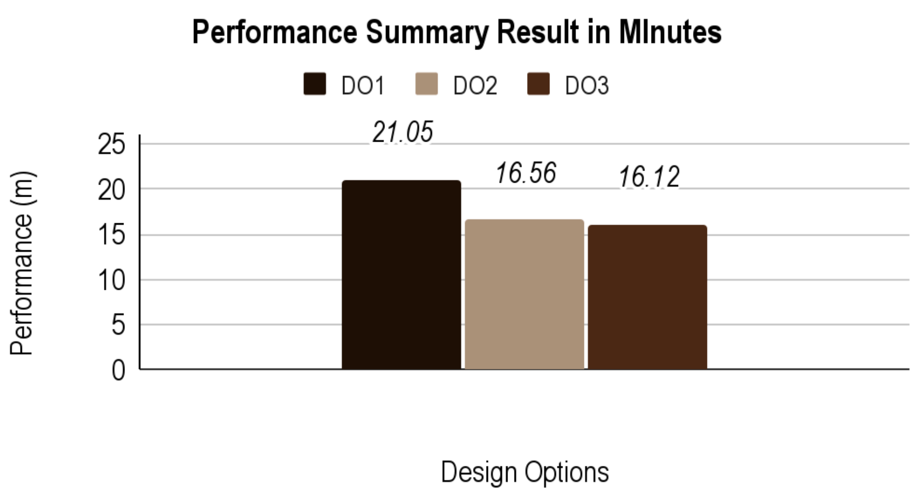


Figure 3-137. Performance Summary Result in Minutes

Figure 3-137 shows the result of the conducted testing in Minutes (m). As suggested, Design Option 3 takes 16 minutes and 12 seconds to reach the set temperature, at 20°C, from 18°C. Design Option 2, on the other hand, took 16 minutes and 56 seconds to reach this temperature from the same initial condition at 18°C. Lastly, Design Option 1 takes the longest time to achieve the optimal condition at 20°C from the given initial condition, as stated above, at 21 minutes and 5 seconds.

3.7.2 Response Time Result Summary

During this testing, the developers conducted at least 50 tests to attest to which of the three options reliably acts fast when the system commands the fan to turn on or off once new sensor readings occur. The below figure is the summary result of each option:

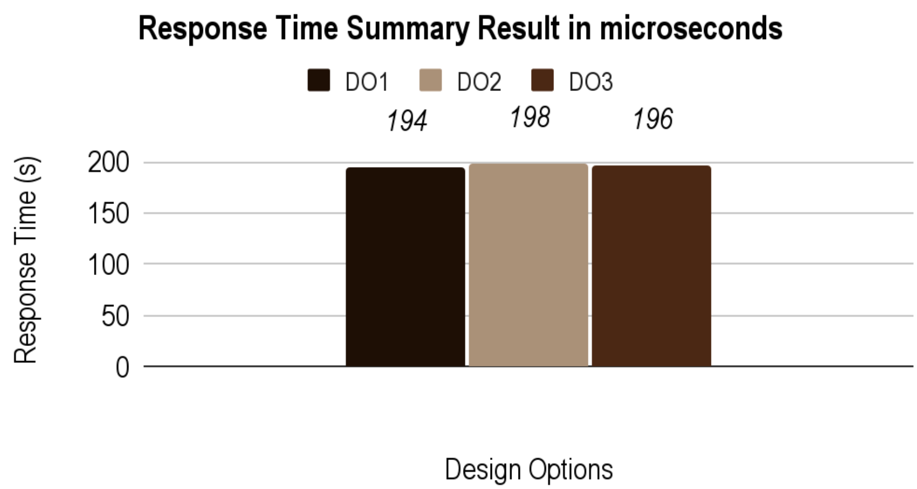


Figure 3-138. Response Time Summary Result in Microseconds

Figure 3-138 depicts the performance of the accessibility options based on response time. In the response time, the lower the value, the faster the design option. Design option 2 has the highest response time of 198µs. Design option 3, on the other hand, has a 196µs response time. Lastly, Design Option 1 has the lowest response time of 194µs, the fastest among the three options.

3.7.3 Power Consumption Result Summary

In this testing, each component of the three design options is computed to find its power consumption in Wattage (W) and converted to kiloWatt-hour (kWh). The below figure summarizes the Power Consumption of the three design options based on mathematical computation:

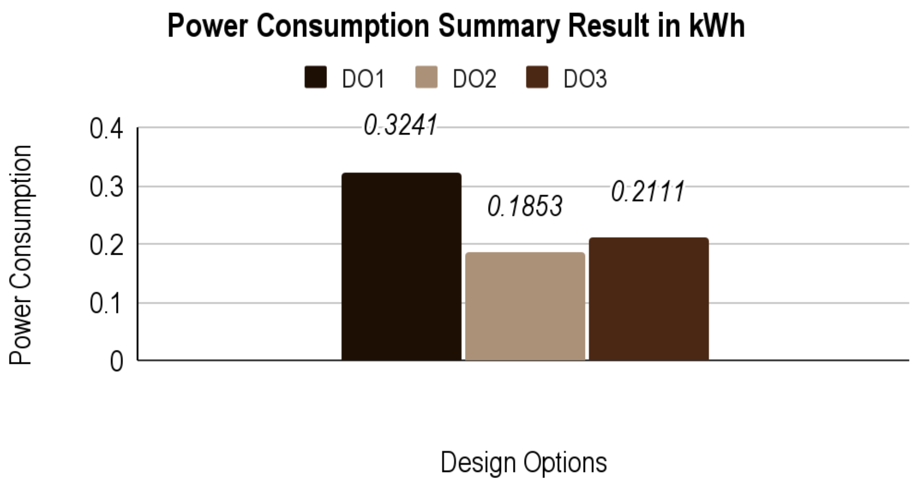


Figure 3-139. Power Consumption Summary Result in kWh

Figure 3-139 shows the different design options for power consumption. Ideally, the lower the value, the more efficient the design option is. In this case, Design Option 1 has the highest power consumption, 0.3241 kWh, followed by Design Option 3 at 0.2111 kWh and lastly Design Option 2 at 0.1853 kWh. This result shows that Design Option 2 produces lower power consumption.

CHAPTER 4: CONSTRAINTS, STANDARDS, AND TRADE-OFFS

4.1 Design Constraints

Three design constraints were established as a developing guide to assist the developers in creating the project. Below, the design constraints are explicitly discussed to understand their purpose and responsibility in this project.

4.1.1 Performance

The definition of performance constraint in designing a storage device that requires accurate monitoring and control of the temperature and humidity inside to maintain the moisture content of the coffee beans is the ability of the device to quickly adjust to the optimal condition once a deviation is detected. This involves ensuring that the storage conditions are maintained within a specific range.

4.1.2 Response Time

In the context of dotBean's project, response time is defined as the duration for the Raspberry Pi to activate the fan when there are new temperature readings. One of the client requirements is that the fan automatically reacts, on or off, when there are changes in new readings. It's essential to recognize if the system's response time is reliably fast since it's automation in terms of controlling the temperature when it drops below 20 through fan activation

4.1.3 Power Consumption

This project considers power consumption due to the client's requirement that the system operate at low power. With this, as much as possible, the developers utilized components that operate with low voltage and low current that can be controlled and directly connected with the Raspberry Pi. Additionally, all the components must be

4.2 Importance Weighting of Constraints

In various optimization and decision-making contexts, constraints often must be weighted based on their importance. This approach prioritizes certain constraints over others based on their perceived importance. This technique is instrumental in complex problems where multiple constraints compete for resources or have varying degrees of impact on the overall outcome.

Table 4-1. Constraints Importance Weighting

Constraints	Weighting Factor
Performance	0.50
Response Time	0.30
Power Consumption	0.20

The dotBean project has three design constraints: Performance, Response Time, and Power Consumption. As shown in Table 4-1, Performance has the most significant weighting factor at 0.50 value, followed by Response Time at 0.30 value, and at the lowest, 0.20 value, is the Power Consumption. The assignment of each weighting factor was influenced by the client's requirements and the design problem formulated, as discussed in Chapter 1. Since the dotBean mainly targets the low temperature that affects the moisture content of the coffee beans when stored, the system must be able to quickly adapt to these changes and reach the optimal set parameters faster.

4.3 Trade-offs Analysis

To conduct a trade-off analysis, Equation 6 below is used:

$$RankingScore \ = \ |(1 - \frac{MV - RV}{HV}) \times 5| \qquad \textit{(Equation 6)}$$

Where:

- MV is the obtained value from measurements.
- RV is the target or desired value that is being compared against.
- HV is the highest value from the data results from the testing.

The developers utilized this equation because every constraint requires a lower value to validate its effectiveness.

4.3.1 Performance Constraint

When the testing for Performance, as a Design Constraint, was conducted, the developers utilized SolidWorks to obtain its value time in seconds (s). The lower the time in seconds (s), the faster its performance achieves the required conditions. As a result, Design Option 2, consisting of a Radiator Fan, is identified as more efficient when reaching the set parameters faster than TEC-12706 alone proposed in the two other Design options. Therefore, this analysis uses the acquired value for the Design Option 2 testing results as the referenced value (RV).

$$Design\ Option\ 1\ Ranking\ Score\ =\ |(1-\frac{21.05-16.12}{21.05})\times 5|\ =\ 3.8920$$

$$Design\ Option\ 2\ Ranking\ Score\ =\ |(1-\frac{16.56-16.12}{21.05})\times 5|\ =\ 4.8955$$

$$Design\ Option\ 3\ Ranking\ Score\ =\ |(1-\frac{16.12-16.12}{21.05})\times 5|\ =\ 5.0000$$

Table 4-2. Performance Constraint Ranking Score

Design Option	Value (mv)	Ranking Score
Design Option 1	21.05	3.8290
Design Option 2	16.56	4.8955
Design Option 3	16.12	**5.0000**

The ranking of the Performance constraint for each of the available design options is shown in Table 42. Design Option 1 had the lowest ranking score of 3.8920, while Design Option 2 had a ranking score 4.8955. On the other hand, Design Option 3 prevailed in performance constraints with a ranking score of 5.0000.

4.3.2 Response Time Constraint

In this analysis, Response Time is obtained by finding the difference of time in seconds between the data received by the Raspberry Pi and the time in seconds the Raspberry Pi forwarded the data through Email, Firebase (Mobile Application), and SMS to notify the users. The testing result shows that Design Option 2 has a mobile application (using Firebase DB) to inform users about the dotBean storage changes. It shows more reliable time as it offers more real-time notifications than other accessibility options. Hence, this analysis uses the value of the mobile app as the RV.

$$Design\ Option\ 1\ Ranking\ Score\ =\ |(1-\frac{194-194}{198})\times 5|\ =5.000$$

$$Design\ Option\ 2\ Ranking\ Score\ =\ |(1-\frac{198-194}{198})\times 5|\ =\ 4.8990$$

$$Design\ Option\ 3\ Ranking\ Score\ =\ |(1-\frac{196-194}{198})\times 5|\ =\ 4.9495$$

Table 4-3. Response Time Constraint Ranking Score

Design Option	Value (mv)	Ranking Score
Design Option 1	194	**5.0000**
Design Option 2	198	4.8990
Design Option 3	196	4.9495

Table 4-3 presents the Response Time constraint ranking for all the existing design options. Design Option 2 had the lowest ranking score of 4.8990, and Design Option 3 had a ranking score of 4.9495. Lastly, based on the data presented, Design Option 1 won in Response Time constraint with a ranking score 5.0000.

4.3.3 Power Consumption Constraint

To validate and attest to the efficiency of each design option regarding how much power they consumed, the developers consulted an ECE Professor to identify how the power consumption can be computed for each design option. During the testing and validation for the Power Consumption constraint, the developers identified every component's Voltage (V) and Current (I) and multiplied these two values to find their Wattage (P). Once the P of all components is calculated, the developers summed this up, multiplied it to 1 hr, and divided it by 1000 to convert it from W to kW, resulting in kWh, as shown in the table below (*see Table 4-4*). In this constraint, Design Option 3 won with the lowest consumed power at 1 hr.

$$Design\ Option\ 1\ Ranking\ Score\ =\ |(1-\frac{0.3241-0.1853}{0.3241})\times 5|\ =2.8587$$

$$Design\ Option\ 2\ Ranking\ Score\ =\ |(1-\frac{0.1853-0.1853}{0.3241})\times 5|\ =\ 5.0000$$

$$Design\ Option\ 3\ Ranking\ Score\ =\ |(1-\frac{0.2111-0.1853}{0.3241})\times 5|\ =\ 4.6020$$

Table 4-4. Power Consumption Constraint Ranking Score

Design Option	Value (mv)	Ranking Score
Design Option 1	0.3241	2.8587
Design Option 2	0.1853	**5.0000**
Design Option 3	0.2111	4.6020

The ranking of the Power Consumption constraint for each of the available design options is shown in Table 4-4. The Design Option 1 got the lowest ranking score of 2.8587. On the other hand, the Design Option 3 has a ranking score of 4.6020. Lastly, Design Option 2 prevailed in the Power Consumption constraint with a ranking score 5.0000.

4.3.4 Overall Ranking

Now, all the ranking scores for each design option on every design constraint are calculated and then used as a reference to find the final ranking score based on the weighting factor of each constraint. In this analysis, Performance has the highest weighting factor at 50% or 0.50 value, the Response time at 0.30 value, and the Power Consumption at 0.20 value. The Ranking Score values obtained previously are multiplied by its intended Weighting Factor (i). All results from this calculation are surmised for every Design Option to find their Overall Ranking Score, as shown in the equations below:

$$Overall\ Ranking = (Ranking\ Score\ *\ i\ ) + (Ranking\ Score\ *\ i) + (Ranking\ Score\ *\ i) \quad \textit{(Equation 7)}$$

Where:

- $Ranking\ Score$ It is the value obtained from the previous computation.
- i Is the Weighting Factor.

$$Design\ Option\ 1\ Overall\ Ranking\ =\ (1.9145\ *\ 0.5)\ +\ (1.5000\ *\ 0.3)$$

$$+ (0.5717 * 0.20) = 3.9862$$

$$Design\ Option\ 2\ Overall\ Ranking = (2.4477 * 0.5) + (1.4697 * 0.3) + (1.0000 * 0.20) = 4.9174$$

$$Design\ Option\ 3\ Overall\ Ranking = (2.5000 * 0.5) + (1.4848 * 0.3) + (0.9204 * 0.20) = 4.9052$$

Table 4-5. Overall Design Constraints Ranking Score

Constraints	Weighting Factor (i)	Ranking Score		
		Design Option 1	Design Option 2	Design Option 3
Performance	0.5	1.9145	2.4477	2.5000
Response Time	0.3	1.5000	1.4697	1.4848
Power Consumption	0.2	0.5717	1.0000	0.9204
Overall Ranking		3.9862	**4.9174**	4.9052

Table 4-5 summarizes each design option's overall ranking of design constraints. In the analysis, the priority criteria of importance are the performance with 50%. The response time is the next priority, with an importance factor of 30%, and the least priority is power consumption, with a value of 20%. Using the formula above, Design Option 2 got the highest ranking of 4.9174, Design Option 3 got a ranking score of 4.9052, and the lowest was Design Option 1, which got a ranking score of 3.9862.

4.4 Sensitivity Analysis

Sensitivity Analysis explores the potential impacts of uncertainty on the outcomes of a model or decision-making process. This analytical approach is beneficial when the relationship between inputs and outputs is complex or not entirely understood, allowing for a deeper exploration of the model's behavior under different assumptions (Kenton, W., 2023). Furthermore, sensitivity analysis is associated with uncertainty analysis, aiming to identify which sources of uncertainty significantly impact the study's conclusions.

The developers conducted three sensitivity analyses for each design option. They modified the weighting factor values in the first and second analyses. In the first analysis, response time had the highest weighting factor, while performance had the lowest. Conversely, in the subsequent analysis, the constraints with the highest weighting factor were power consumption, while response time had the lowest.

In the third analysis, developers assigned each design constraint an equal weighting factor value. This allowed the developers to discern the optimal design option under varied circumstances. The succeeding pages present the sentiment analyses conducted by the developers.

4.4.1 Sensitivity Analysis 1 - Response Time-Centric Analysis

As the title suggests, Response Time has the highest weighting factor in this sensitivity analysis, at 0.50, followed by Power Consumption, at 0.30, and Performance, at 0.20 (*see Table 4-6*).

Table 4-6. Importance Weighting of Constraints for Response Time-centric Analysis

Constraints	Weighting Factor
Performance	0.20
Response Time	0.50
Power Consumption	0.30

To evaluate each Design Options ranking, the developers multiply each Weighting Factor (i) of each Constraints with the Design Options 1-3 Ranking Score as solved above (*see Table 4-2, Table 4-3, & Table 4-4*). Furthermore, the Response Time-Centric Analysis Overall Ranking was taken by summating all the constraints by each Design Options Ranking column.

Table 4-7. Sensitivity Analysis 1 Overall Final Ranking of Design Options 1-3

Constraints	Weighting Factor (i)	Ranking Score		
		Design Option 1	Design Option 2	Design Option 3
Performance	0.2	0.7658	0.9791	1.0000
Response Time	0.5	2.5000	2.4495	2.4747
Power	0.3	0.8576	1.5000	1.3806

Constraints	Weighting Factor (i)	Ranking Score		
		Design Option 1	Design Option 2	Design Option 3
Consumption				
Response Time-Centric Analysis Overall Ranking		4.1234	**4.9286**	4.8553

As tabulated in Table 4-7, Design Option 2 ranking has the overall highest value at 4.9286, followed by Design Option 3 at 4.8553, and at the lowest rank, Design Option 1 at 4.1234 value.

4.4.2 Sensitivity Analysis 2 - Power Consumption-Centric Analysis

In this second sensitivity analysis, Power Consumption has the highest weighting factor, 0.50; Performance is second, 0.30; and Response Time has the lowest value, 0.20 (*see Table 4-8*).

Table 4-8. Importance Weighting of Constraints for Power Consumption-centric Analysis

Constraints	Weighting Factor
Performance	0.30
Response Time	0.20
Power Consumption	0.50

Similar to Sensitivity Analysis 1, to get the Overall Ranking Score of Power Consumption-Centric Analysis, all products of Design Options 1-3 Ranking Scores and Weighting Factor (i) summed up together, as shown in Table 4-9.

Table 4-9. Sensitivity Analysis 2 Overall Final Ranking of Design Options 1-3

Constraints	Weighting Factor (i)	Ranking Score		
		Design Option 1	Design Option 2	Design Option 3
Performance	0.3	1.1487	1.4686	1.5000
Response Time	0.2	1.0000	0.9798	0.9899
Power Consumption	0.5	1.4293	2.5000	2.3010
Power Consumption-Centric Analysis Overall Ranking		3.5780	**4.9484**	4.7909

In power consumption-centric analysis, Design Option 2 accumulated the highest overall ranking at 4.9484. On the other hand, the second ranking is Design Option 3 at 4.7909 value; lastly, Design Option 1 has the lowest overall ranking, again at 3.5780 value.

4.4.3 Sensitivity Analysis 3 - Equal Weighting Analysis

In this final sensitivity analysis, equal weighting has been applied to each design criterion and constraint with a weighting factor value of 0.33, as shown in Table 4-10. This approach accommodates clients' preferences, who may desire uniform importance across all design elements.

Table 4-10. Importance Weighting of Constraints for Equal Weighting Analysis

Constraints	Weighting Factor
Performance	0.33
Response Time	0.33
Power Consumption	0.33

It corresponds to the above two Sensitivity Analyses, where the calculation of the Overall Ranking Score for Equal Weighting Analysis is the total of each Design Option's Ranking Score multiplied by its corresponding Weighting Factors and add it all together, as presented in Table 4-11.

Table 4-11. Sensitivity Analysis 3 Overall Final Ranking of Design Options 1-3

Constraints	Weighting Factor (i)	Ranking Score		
		Design Option 1	Design Option 2	Design Option 3
Performance	0.33	1.2636	1.6155	1.6500
Response Time	0.33	1.6500	1.6167	1.6333
Power Consumption	0.33	0.9434	1.6500	1.5187
Equal Weighting Analysis Overall Ranking		3.8569	**4.8822**	4.8020

Table 4-11 displays the overall ranking scores for each design in the third sensitivity analysis. Despite employing equal weighting factor values, Design Option 2 retains the highest overall ranking score of 4.8822. Design Option 3 follows with a value of 4.8020, while Design Option 1 trails behind at 3.8569. There are notable discrepancies in the scores. It's evident that

Design Option 2 consistently performs well across all sensitivity analyses compared to the other design options.

4.4.4 Summary of All Analyses

After conducting Trade-off Analyses and Sensitivity Analyses for gathered data from the Testing and Validation of Design Options in Chapter 3, the developers have finally summarized the accumulated results. They can rank each design option. This process has enabled the developers to break down the complexities of the data into a clear structure, making it easier to prioritize options based on their pros and cons.

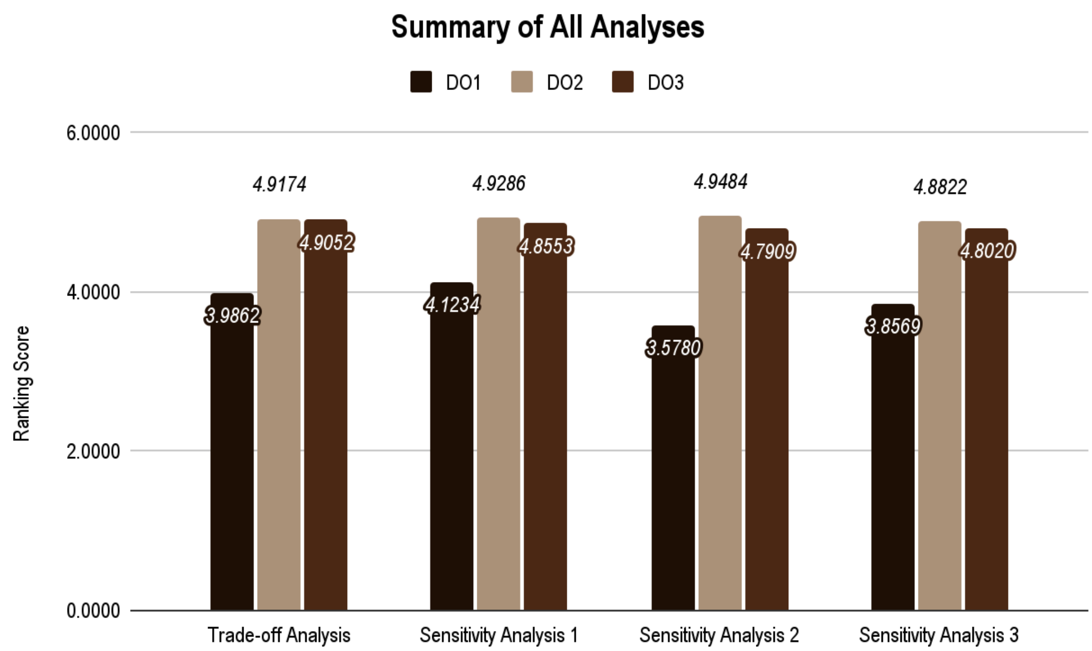


Figure 4-1. Overall Ranking Summary of Design Options 1-3

Figure 4-1 shows each design option's ranking of design constraints. It also shows which design possibilities are most advantageous given the various constraints. Furthermore, as seen in the table, design option 2 has the advantage over others. The top left, the Trade-off Analysis, shows that Design Option 2 has the highest value, amounting to 4.9174, followed by Design Option 3 at 4.9052 and Design Option 1 at 3.9862.

Beside the Trade-off Analysis result is the Sensitivity Analysis 1, where the Response Time has the highest weighting factor and the Performance has the lowest. It shows that Design Option 2 is still the highest, with a value of 4.9286; second is Design Option 3, with a value of 4.8553, and lastly, Design Option 1, with a value of 4.1234.

From the Second Sensitivity Analysis, Design Option 2 maintains its highest position at 4.9484. Similarly, Design Option 3 is still at the second highest with a value of 4.7909 and falling short at 3.5780, which is Design Option 1. Lastly, the Third Sensitivity Analysis shows how the Three Design Options maintain their ranking despite having an equal weighting factor at 33% or 0.33. Design Option 2 led the Sensitivity Analysis 3 at a value of 4.8822, followed by Design Option 3 at 4.8020, and lastly, falling short at a value of 3.8569, Design Option 1.

4.5 Design Standards

Design standards are guiding principles or criteria implemented in a design project to ensure consistency, quality, and usability. These guidelines are established to promote best practices and provide researchers with a structured framework for their work.

RS ISO 8455:2011 Green Coffee—Guidelines for Storage and Transport: The International Organization for Standardization (ISO) 8455:2011 strives to maintain the quality of green coffee beans from harvest to destination. It encompasses temperature regulation, moisture levels, packaging, and handling protocols to prevent deterioration during storage and transportation.

ISO 22000:2018 Food Safety Management Systems - Requirements for Any Organization in the Food Chain: As per the International Organization for Standardization, ISO 22000:2018 aims to ensure the safety of food products throughout the food chain. It includes comprehensive guidelines on hazard analysis, good manufacturing practices (GMP), compliance with food safety regulations, supplier verification, documentation and record-keeping, training and awareness, and continuous improvement. These requirements ensure that materials such as polyolefin foam used in food storage are managed to prevent contamination and maintain food safety.

ISO/IEC/IEEE 26514:2022: This document identifies the users' informational needs and explains how the information will be presented, prepared, and distributed. The following type of information can be developed using this information, incorporating user interface, interface multimedia systems using video, animation, and sound.

ISO/IEC/IEEE 23026:2023: This document establishes comprehensive system engineering and management requirements for websites, covering their entire lifecycle from strategy to ongoing management. It applies to those utilizing web technology for disseminating Information and Communications Technology (ICT) information, including user information, systems and software engineering project plans and reports, and IT service management policies. The requirements are relevant for website owners, providers, managers, engineers, designers, developers, and operations personnel, whether internal or external. They address public and restricted access websites, focusing on usability, information retrieval, security, accessibility, and maintenance practices. However, it excludes marketing or sales websites, instructional content delivery, or graphical user interfaces for transactional applications. It does not delve into vendor specifications, network protocols, or software architecture.

ISO/IEC/IEEE 26531:2023: This document specifies the requirements for the effective development and management of information generated throughout a system and software product's life cycle. It focuses on delivering information to system and software users and managing IT and support services. The guidelines provided are independent of the specific tools, protocols, and systems used for content management and do not cover the configuration management of software assets.

CHAPTER 5: FINAL DESIGN

5.1 System Architecture

System architecture is the foundational organization of a system, detailing its components, their interconnections, and the principles guiding its design and evolution (*System Architecture - Detailed Explanation*, 2023). It serves as a blueprint for the system's operation, encompassing hardware, software, documentation, and roles. The figure below depicts the overall system architecture of the dotBean, which illustrates each component's overall connections and relationships and how they interact with one another. This representation ensures that each part of the system functions coherently, supports scalability, and adapts efficiently to future changes and enhancements.

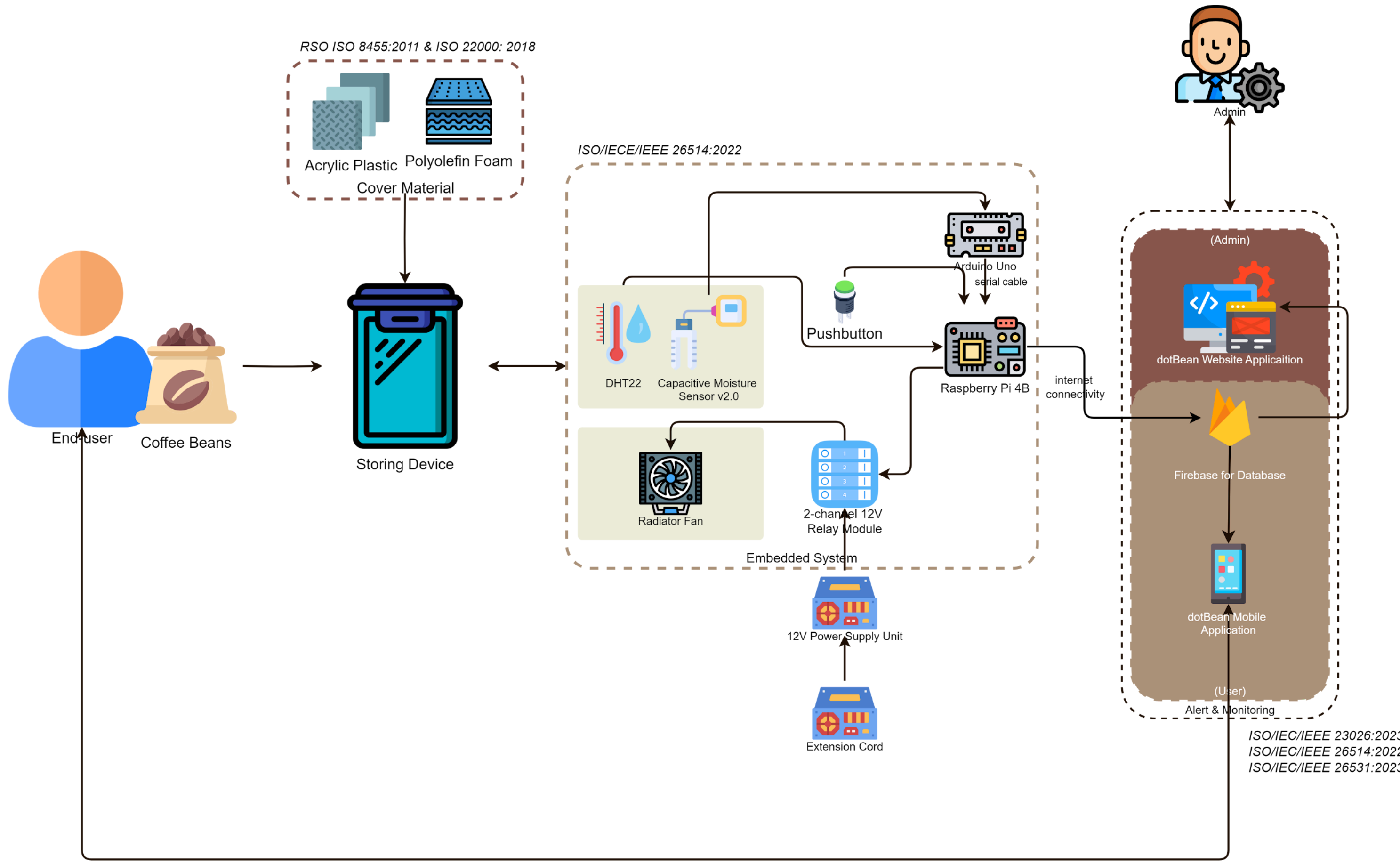


Figure 5-1. dotBean System Architecture

DotBean's clients mainly operate manually, storing their processed green coffee beans in a storage room packed in airtight bags, rice sacks, or jute sacks. This is where the project comes into play: the coffee beans may be stored directly in dotBean, which can automatically monitor the storage conditions and moisture content through a Raspberry Pi 4B. The Raspberry Pi 4B will receive the sensor data and undergo initialization to assess if the temperature is between 20 and 25 degrees Celsius. Initially, the actuators are normally closed and connected to a 5V Relay Module. The Raspberry Pi 4B will control the Relay Module to open or close the actuators based on the detected conditions, as it's directly connected to the power supply. The power supply and the Raspberry Pi 4B are connected to a dedicated extension cord, which must be plugged into a power outlet.

Once the Raspberry Pi receives data from the sensors and actuators, at intervals of 10 seconds, this data will be transferred to dotBean's Firestore DB. This database is connected to a mobile application accessible to clients and a website accessible only to administrators. Due to the sensitive nature of the data, including user account information, access to the website is restricted to individuals managing dotBean's Gmail account.

5.2 System Flowchart

The system flowchart was developed to provide a comprehensive overview of the system's operation. This flowchart visualizes how the system will address the identified problem, presenting the process in a structured and graphical format. The flowchart is essential for understanding the system's functionality by depicting the sequential flow of activities and decision points.

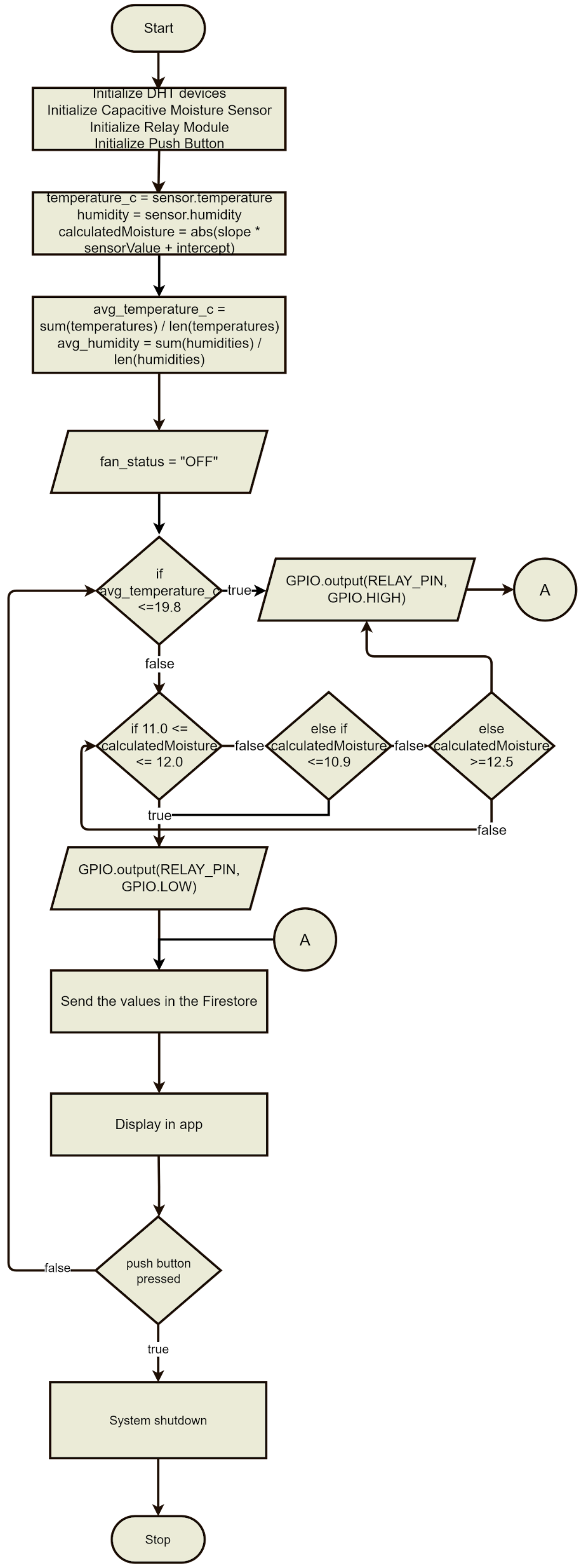


Figure 5-2. dotBean General System Flowchart

Figure 5-2 displays the device's whole operation. The system started by initializing the DHT devices, capacitive moisture sensor, and relay module. Once initialized, it will open the GUI that will display the amount of temperature, humidity, and moisture content. Then, the user will see if the average temperature is met. The fan will turn on if the temperature is less than or equal to 19.9°C. If the temperature is not below 19.9°C and the moisture content is 11% to 12.5%, the fan will remain off because the conditions inside are well maintained. Afterward, the fan's status and the sensors' output data will be sent to Firebase and reflected in the app for the user to see.

5.3 System Modules

A System Module generally denotes a distinct element or group responsible for carrying out essential operations inside a more extensive system. The architecture heavily relies on it, and the system needs to function optimally and effectively and meet all the requirements. In dotBean, the system comprises three modules: the Graphical User Interface Module, the Sensors Module, and the Temperature Control Module.

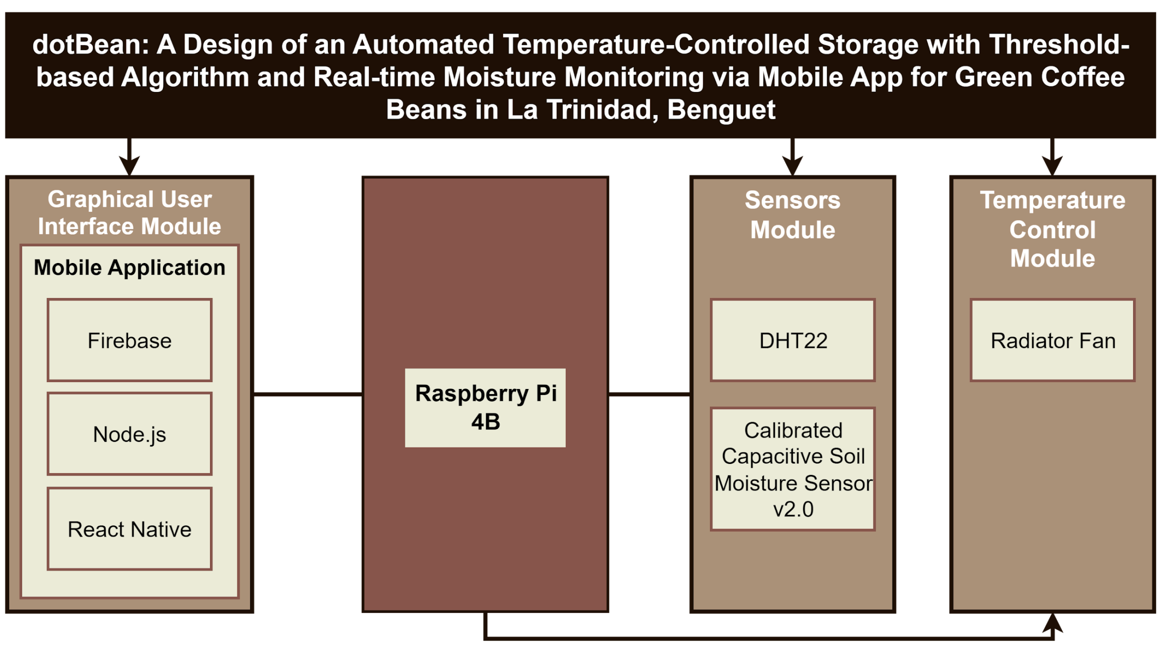


Figure 5-3. The dotBean System Modules

As proposed above, the dotBean System Modules has two subsystems connected through the microprocessor, Raspberry Pi 4B. The system mainly operates on its microprocessor, where the critical initialization and implementation of actions and commands occur. As mentioned above, the Graphical User Interface (GUI) Module is responsible for displaying the data via a mobile application, which the client can access, and via its website, which the administrator has sole access to. Firebase is the middle man between the Raspberry Pi and the Applications for a more secure and organized data flow and storage. Node.js and React Native are responsible for the applications' UI/UX. On the other side, the modules responsible for maintaining the right temperature and humidity inside the storing device and managing the moisture content of the coffee beans are the Sensors Module and Temperature-Control Module. The sensors Module is responsible for detecting environmental conditions (temperature, humidity, and moisture) that trigger the Temperature Control Module, which regulates the conditions inside the storage.

5.4 Hardware Design and Development

5.4.1 Block Diagram

A block diagram is a graphical representation used to illustrate the functional view of a system, project, or scenario. It breaks down complex systems into simpler, understandable components, showing how they interact (*What Is a Block Diagram and How to Create One | Miro*, n.d.). While System Architecture provides a detailed description for developing and managing a system, focusing on the design choices, patterns, and models, the Block Diagram, on the other hand, is the high-level, simplified view of the system's structure and functionality. The below figure illustrates the dotBean block diagram:

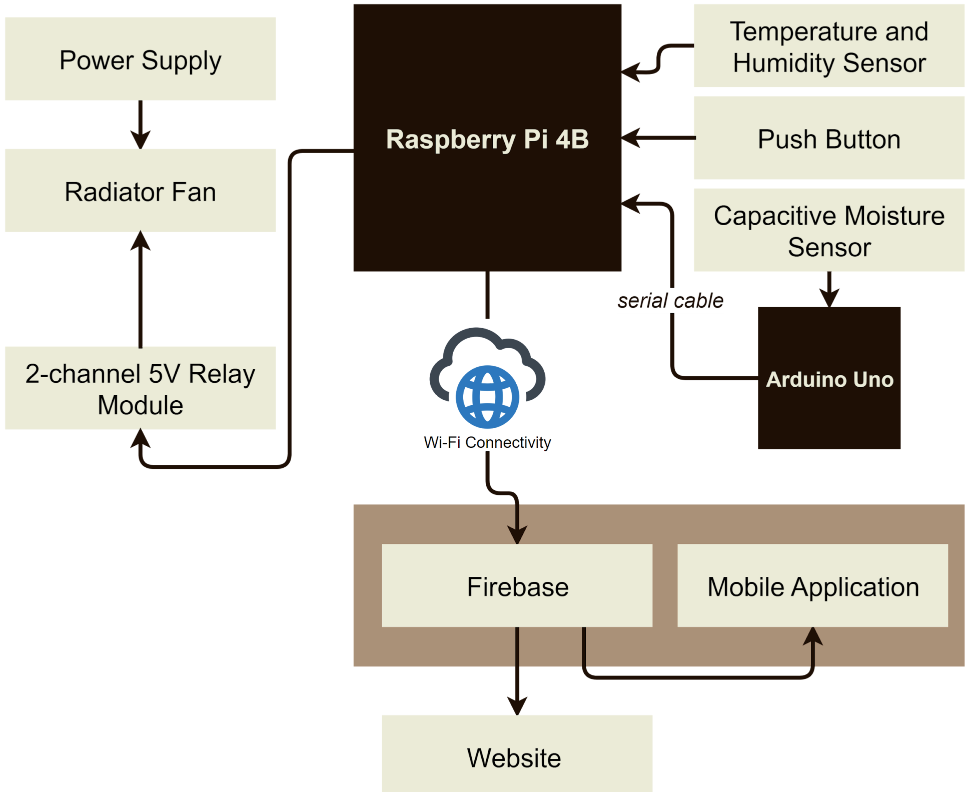


Figure 5-4. dotBean's Block Diagram

In the dotBean system, the Raspberry Pi 4B is the central processing unit. Two sensors, the DHT22, and Moisture Sensor are connected to it and positioned within the storage unit. Additionally, a momentary push button serves as the power control for the Raspberry Pi. To manage the 12V fan, which exceeds the Raspberry Pi's operational voltage of 5V, a relay module interfaces with the Raspberry Pi, enabling control over the fan's operation. A 12V power supply directly powers both the radiator and exhaust fans. The Raspberry Pi and the power supply are connected to a single extension cord, facilitating their connection to a standard power outlet.

The Raspberry Pi requires internet connectivity and Firebase configuration for mobile application integration. Firebase is an intermediary between dotBean's user application and its administrative site.

In conclusion, integrating the Raspberry Pi 4B sensors, relay module, and Firebase within the dotBean system enables efficient monitoring and control capabilities, seamlessly bridging the gap between user interaction and system administration.

5.4.2 Schematic Diagram

The primary purpose of a schematic diagram is to provide a visual blueprint that aids in the design, analysis, and maintenance of systems (Barrozo, n.d.). For instance, in electronics, a schematic diagram can depict the arrangement of circuits without delving into the physical layout of components. This diagram helps developers focus on the system's functionality and operation rather than getting impeded by physical constraints.

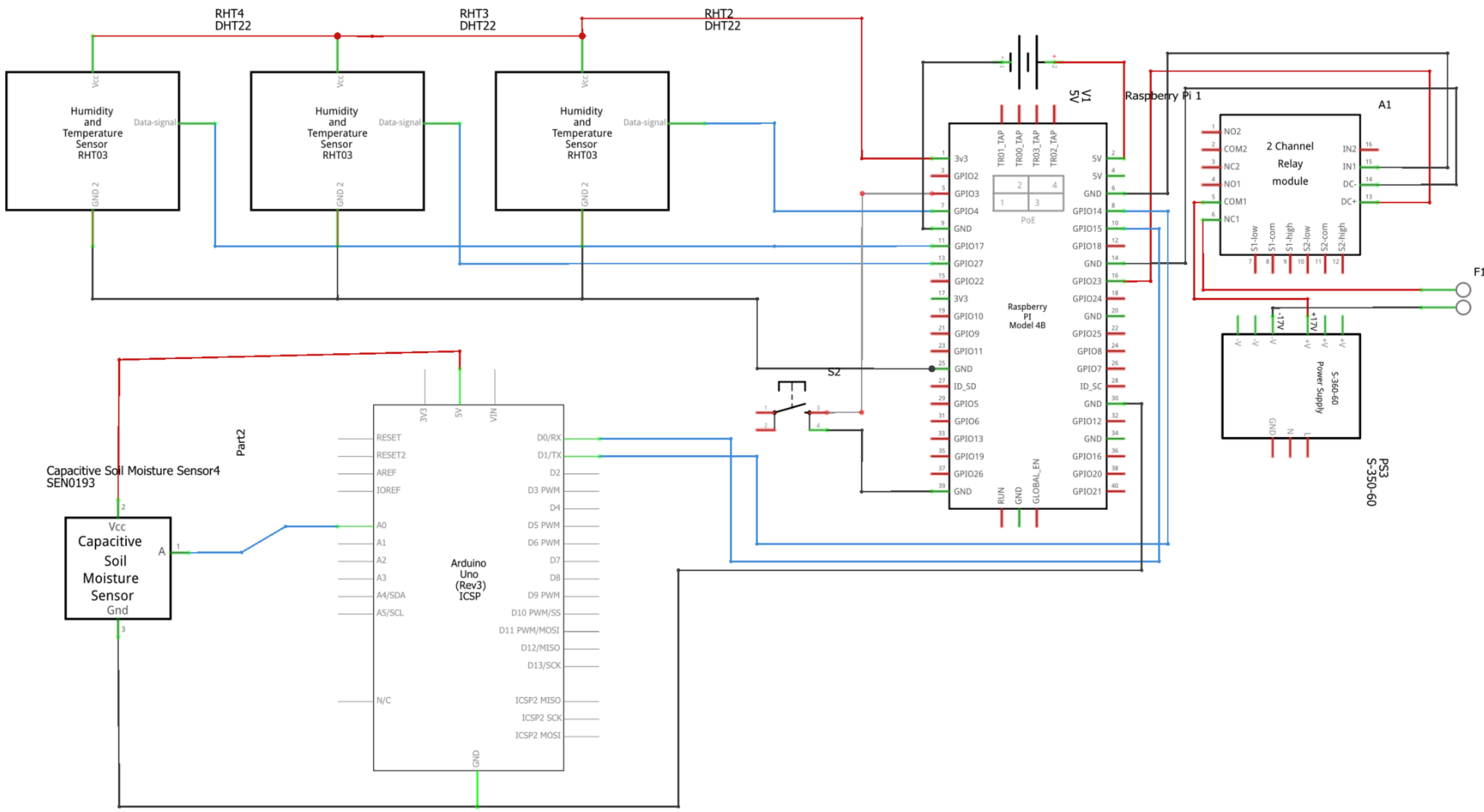


Figure 5-5. dotBean Schematic Diagram

The prototype integrates various components with microcontrollers to ensure efficient data acquisition and control. The analog moisture sensor is connected to the Arduino. This sensor's data is related to the Raspberry Pi through serial communication, with the Arduino linked to the Raspberry Pi's USB port. Three DHT22 sensors are utilized, with their data pins connected to the Raspberry Pi's GPIO pins 4, 17, and 27 (physical pins 7, 11, and 13). Their ground pins are attached to the Raspberry Pi's ground, and the VCC pins are connected to a 3.3V power source.

The setup also includes a 2-channel 5V relay module, where the VCC is connected to GPIO pin 23 (physical pin 16), IN1 to ground, and the module's ground to the Raspberry Pi's ground. A jumper wire connects RY-VCC to the 5V and ground to the ground. For the power supply unit (PSU) and radiator fan, the normally closed relay terminal connects to the radiator fan's VCC, and the common terminal connects to the PSU's V+. The radiator fan's ground is linked to the PSU's common terminal. The soil moisture sensor connects to the Arduino, with its VCC to 5V, its A0 to the analog input A0, and its ground-to-ground. Data from the Arduino is sent to the Raspberry Pi 4B through serial communication. Lastly, a push button is connected to GPIO3 (physical pin 5) and ground (physical pin 39). The Raspberry Pi and the PSU are directly connected to the extension cord.

5.4.3 Parametric Design

Figure 5-6. dotBean 3D Front View

The dotBean prototype primarily combines stainless steel and acrylic for its cover, as shown in Figure 5-6. The upper section, which serves as the main storage area for the coffee beans, is constructed from acrylic plastic and a layer of polyolefin foam sandwiched between two acrylic sheets. A wire tube is connected from the upper storage to the compartment housing the components, positioned on the right side of the device. Two removable locks on either side facilitate the easy removal of the upper storage section from the compartment when adjustments to the components are necessary.

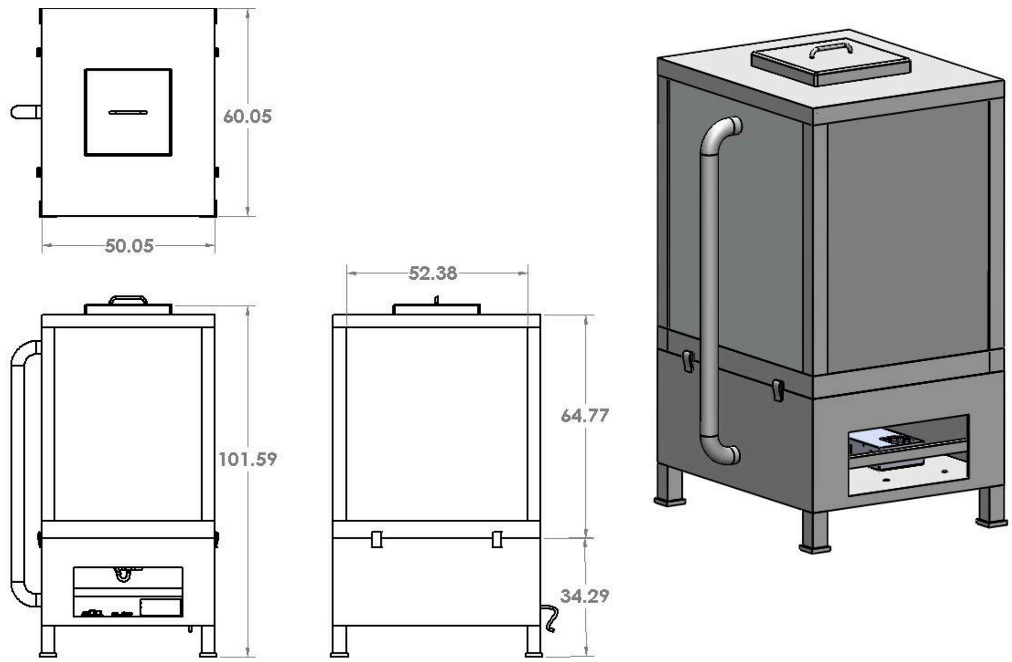


Figure 5-7. dotBean Measurement

Figure 5-7 presents the measurement, in cm, of the final design of the dotBean. Overall, it has a total height of 101.59 cm, 64.77 cm for the upper storage compartment, while the lower storage compartment has a height of 34.29 cm, and the storage handle has a height of 2.23cm. The design has a length of 50.05 cm and a width of 60.05 cm. On the other hand, the tube wire stretches from the upper storage up to the lower compartment, where the wires of every sensor are placed. It has a length of 90cm and a width of 3cm.

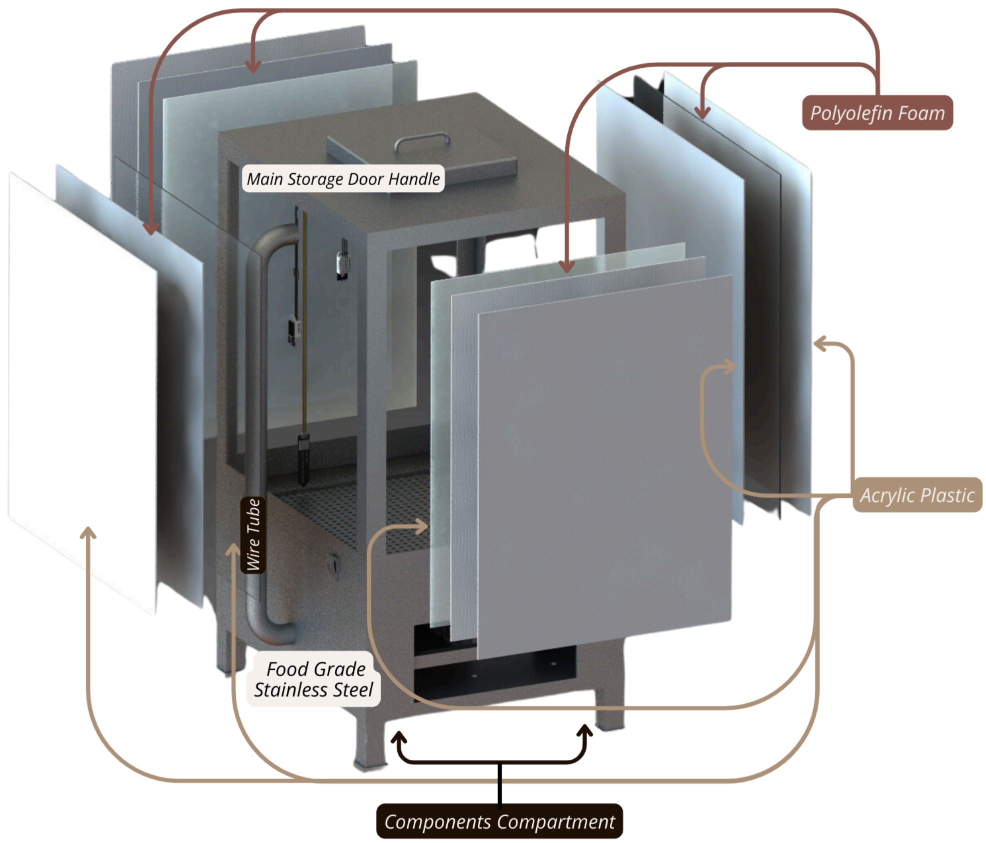


Figure 5-8. dotBean Exploded View with Labels

Figure 5-8 illustrates that each side of the prototype comprises a layer of 3 mm acrylic plastic, 12 mm polyolefin foam, and another 3 mm acrylic plastic layer.

According to Wang et al. (2019), acrylic plastic is known for its impact resistance and is frequently used in food storage to provide a durable barrier for storing food items. Moreover, Shrinithivihahshini (2024) noted that plastic containers, including those used for food storage, can release chemicals into food. Hence, it's essential to utilize safe materials like acrylic plastic that do not pose risks of chemical migration into food. On the other hand, polyolefin, in this design, acts as moisture absorption and insulation. Polyolefin foams are known for their thermal, sound, and vibration insulation properties (Bogdan et al., 2021). According to He et al. (2019), research has shown that polyolefin materials have moisture absorption characteristics that can directly impact the thermal performance of instructors. Its ability to absorb moisture can influence its insulation properties, making it a suitable choice for applications where moisture management and insulation are essential. These cover materials are combined with stainless steel to ensure compact and secure storage. The device's main handle is also made of food-grade stainless steel.

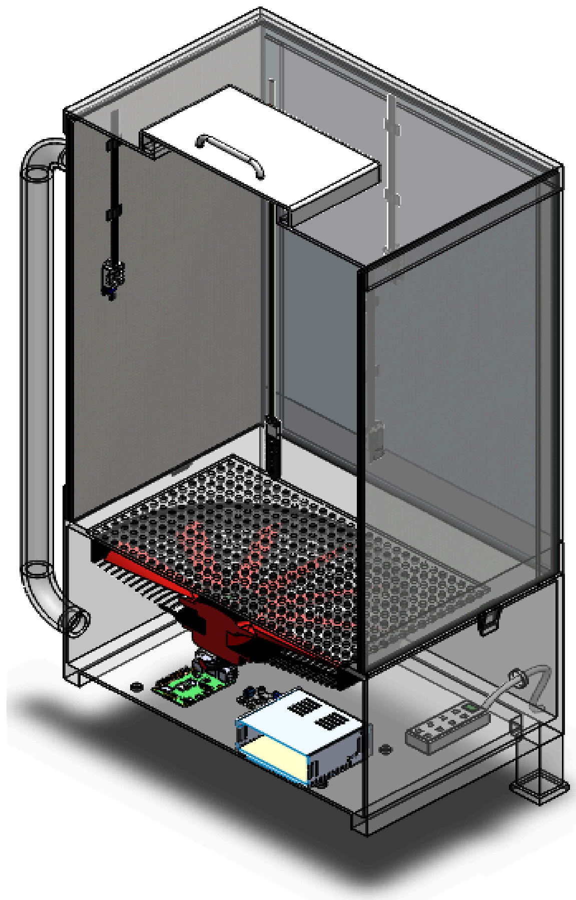


Figure 5-9. dotBean Section View

Figure 5-9 depicts the section view of the final design. The design shows its electric components below the radiator fan and the sensors' placement.

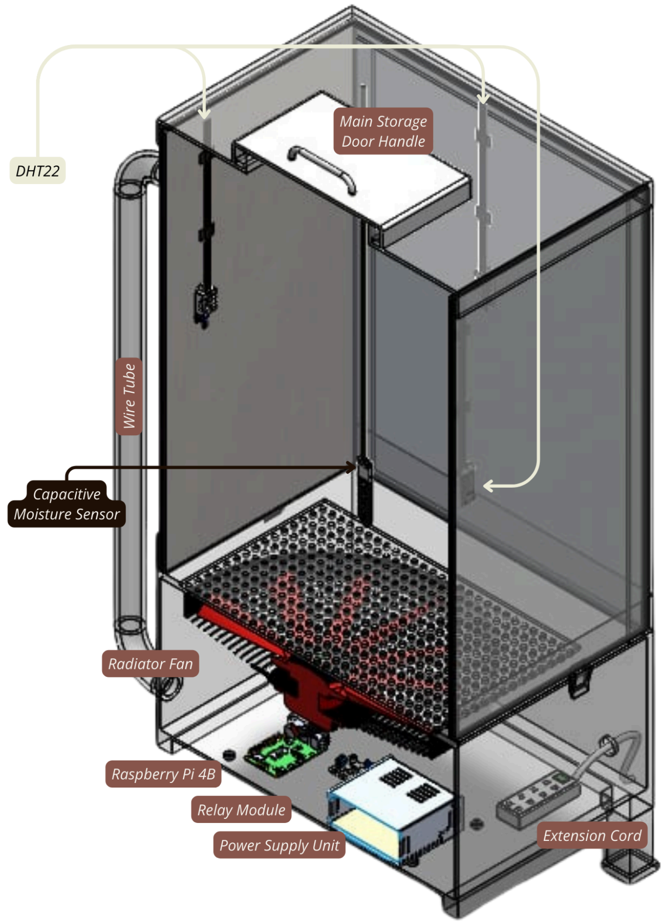


Figure 5-10. dotBean Section View with Labels

The upper part of the body is made of stainless steel, and the walls are made of acrylic glass. Polyolefin foam, which absorbs moisture and acts as an insulator, is between the acrylic glass. Three DHT Sensors are placed on the storage's front, back, and right sides to detect temperature and humidity. Additionally, a capacitive moisture sensor is hung in the middle to detect the moisture content of the stored beans.

Below the primary storage is the electric compartment, which contains the Raspberry Pi, relay module for the fan, extension cord, and power supply unit. The Radiator fan is also mounted upward so the heat can rise toward the primary storage. The wire tube is the pathway for the wire of the sensors from the primary storage to the electric compartment to avoid physical contact between the Green Coffee Beans and the cables (*see Figure 5-10*).

5.4.4 Hardware Components

In the context of this discussion, hardware components refer to physical parts that make up electronic systems. Each element has specific functions and specifications that contribute to the overall performance of devices.

Raspberry Pi 4B 8GB

The Raspberry Pi 4B came in the Raspberry Pi 4 series, a tiny single-board computer designed to serve various applications. It offers a blend of performance, connectivity, and expandability that caters to many users. It is powered by a Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC running at 1.5GHz. This allows it to handle more demanding applications, such as running virtual machines, compiling software, and performing complex calculations. In this case, the developers integrated this microprocessor, which enhances the capability to transfer sensors and actuators data to the Firebase DB through its built-in WiFi. This integration facilitates real-time data storage, updates, and remote control functionalities.

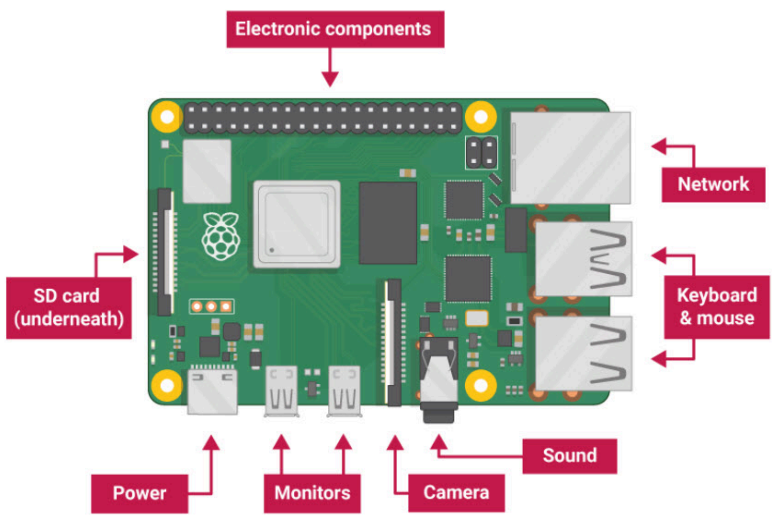


Figure 5-11. Raspberry Pi 4B General Parts

As depicted in Figure 5-11, the Raspberry Pi has numerous functional parts, including a 40-pin GPIO header for interacting with external hardware, Gigabit Ethernet for fast networking, and four USB ports (including two USB 3.0) for peripherals. Multimedia capabilities include a 3.5mm audio jack and a dual micro HDMI port supporting 4K resolution at 60fps. Power is managed via USB Type-C and storage via a microSD slot. It also supports a Camera Serial Interface for high-resolution imaging projects and integration with a GSM module. The developers utilized the 40-GPIO pins to connect the sensors and relay modules for their data, VCC, and ground. On the other hand, Gigabit Ethernet is also employed for internet connectivity. Furthermore, USB Type-C is also used to power/boot up the microprocessor, and the microSD slot is used to store its OS, files, and other configurations within the Raspberry Pi 4B.

As mentioned above, the 40-GPIO pins were utilized to establish a connection between the sensors, the relay modules, and the momentary push button. Each of the 40-GPIO pins had its role, and specific pins were only where, for example, the DHT22 sensors and relay modules could be inserted.

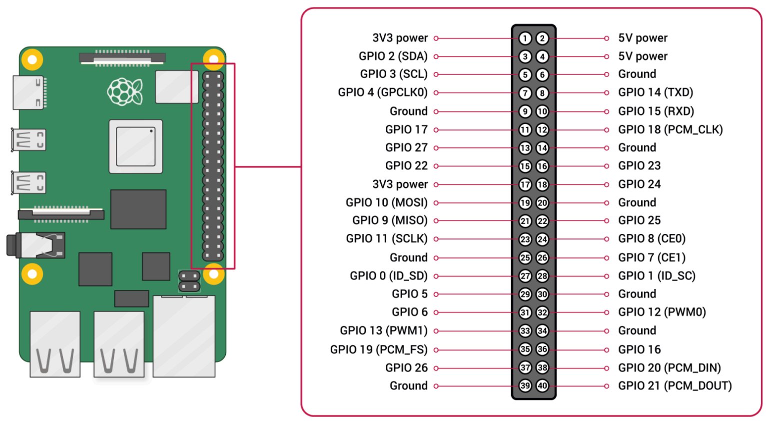


Figure 5-12. Raspberry Pi 4B 40-GPIO Pins

GPIO pins are used for general-purpose input/output operations. Figure 5-11 shows that this allows for controlling various digital devices, such as LEDs, buttons, and sensors. These pins also can read data from these devices. On the contrary, PWM pins are designed to generate pulse width modulation signals, which control the amount of power supplied to devices like motors or LEDs by adjusting the signal's duty cycle. Moreover, SPI pins are synchronous serial communication interfaces used for short-distance communication, primarily in embedded systems. It enables full-duplex communication between a controller device and one or more secondary devices.

Additionally, other pins are known as I2C pins, a communication protocol used for short-distance, intra-board communication. It allows multiple low-speed devices to communicate with each other using only two wires. Furthermore, UART, TX, and RX pins are used for Universal Asynchronous Receiver/Transmitter communication. UART is a hardware device or software implementation that translates data between parallel and serial forms. It is used for serial communication between devices. Lastly, other pins include ID_SD, which supports SD card communication, while ID_SC pins support SPI communication. SD cards are a type of flash memory card used for storing digital data, while SPI is a synchronous serial communication interface.

Capacitive Soil Moisture Sensor v2.0

Figure 5-13. Capacitive Soil Moisture Sensor

This sensor, as shown in Figure 5-13, is calibrated to detect the moisture levels in coffee beans instead of soil. While initially intended to measure the moisture levels in soil, its method, which evaluates the capacitance in an object, can be used for coffee beans or other objects when adequately calibrated.

The sensor operates under voltages between 3.3 to 5.5V, making it compatible with all significant microcontrollers. The sensor module includes an onboard voltage regulator, ensuring functionality within this voltage range. It works by measuring changes in capacitance caused by variations in the dielectric, calibrated explicitly for detecting moisture in coffee beans. Instead of directly measuring the coffee beans' moisture (as pure water poorly conducts electricity), it detects ions dissolved in the moisture. The sensor's capacitance is measured using a 555-timer-based circuit that generates a voltage proportional to the soil's capacitance. This voltage is then converted by an Analog to Digital Converter (ADC), producing a percentage value representing moisture content.

DHT22

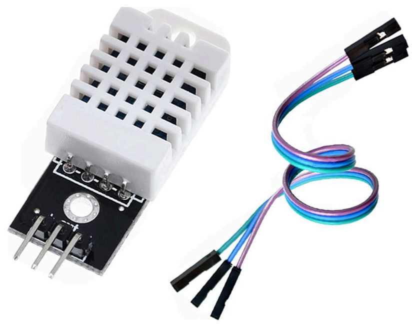


Figure 5-14. DHT22 Sensor

The DHT22, as shown in Figure 5-14, can measure temperature and humidity. It's easy to use and doesn't cost much. It's a digital sensor with an analog-to-digital converter built in, which makes it easy to connect to the Raspberry Pi. The single data pin can link to the GPIO on the Raspberry Pi.

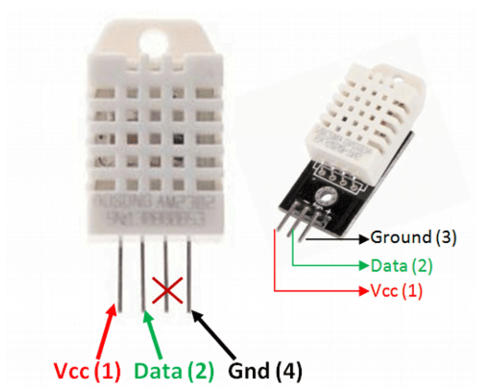


Figure 5-15. DHT22 Pin Configuration

The power source is the Vcc pin, which needs a voltage of 3.5V to 5.5V. The temperature and humidity values are sent in a serial format through the values pin, the output pin. Finally, the

Ground pin is linked to the circuit's ground, as its name suggests. For the DHT22 module to work right, these pins must be set up this way (*see Figure 5-15*).

Acrylic Plastic

Figure 5-16. Acrylic Plastic

The primary purpose of Acrylic Plastic in this project is to serve as a cover material when combined with Polyolefin Foam. Acrylic plastic, known as Plexiglas (*see Figure 5-15*), is highly versatile due to its optical clarity and impact resistance. It's lightweight, chemically resistant, and easy to machine, making it ideal for various components.

316 Food-grade Stainless Metal Steel

316 Food-grade stainless steel is known for being resistant to corrosion and for its resistance to chlorides and acids, making it ideal for use in environments where salt exposure is expected. In this design, it will be used as a storage compartment for the components.

Polyolefin Foam

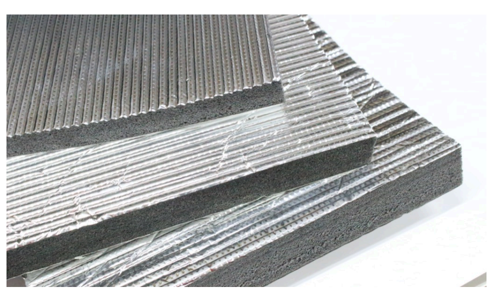


Figure 5-17. Polyolefin Foam

Polyolefin foam is highly valued for its dual insulation and moisture absorption capabilities. Its exceptional thermal insulation properties effectively minimize heat transfer and maintain temperature stability, making it suitable for diverse environments and temperatures. Its low water absorption rate also ensures it resists moisture penetration, which is crucial for applications such as HVAC systems and packaging, where preventing moisture buildup is essential. Polyolefin foam's chemical resistance further enhances its durability, making it resilient against oils, solvents, and various chemicals.

A combination of polyolefin foam and acrylic plastic can protect items like coffee beans, which should not be exposed to light due to the risk of flavor and aroma degradation. This combination helps maintain a stable temperature and prevents moisture buildup, ensuring the beans stay fresh and retain their quality during storage and transportation.

12V Radiator Fan

Figure 5-18. 12V Radiator Fan

The general purpose of the radiator fan, as displayed in Figure 5-18, is to regulate the air inside the storage when the temperature drops and ensure it returns to its optimal range. Since the Radiator Fan operates at a voltage of 12, and the microprocessor, on the other hand, can only handle components with a voltage of 5, the Radiator Fan needs to be connected through a 5V Relay Module for the microprocessor to control its status.

12V Switching Power Supply Unit (16.7Amps)

Figure 5-19. 12V Switching Power Supply

A 12V switching power supply is designed to convert alternating current (AC) from the mains power supply into direct current (DC) at a regulated 12 volts. "Switching" refers to the technology used to regulate the voltage efficiently by rapidly switching the power on and off. This method allows for smaller, lighter power supplies that are more energy-efficient than traditional linear ones.

64GB SD Card

Figure 5-20. Sandisk Extreme 64gb SD Card

The SanDisk Extreme 64GB SD Card, shown in Figure 5-20, is an ideal storage solution for the Raspberry Pi. It accommodates the operating system, applications, and data storage needs. With a storage capacity of 64GB, it boasts a transfer rate of 100MB/s and a read speed of up to 90MB/s. Being UHS-I compliant, it supports high-speed data transfer. Utilizing a microSD card form factor, it seamlessly fits into the Raspberry Pi's microSD slot, ensuring compatibility and efficient performance.

2-channel 5V Relay Module

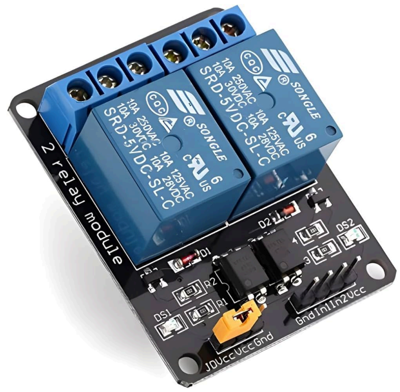


Figure 5-21. 2-Channel 5V Relay Module

Figure 5-21 presents the 2-channel 5V relay module that allows low-voltage microcontrollers like Raspberry Pi to control higher-voltage devices. For example, this project's Radiator Fan and Peltiers operate at 12V.

The Relay Module is connected to the Raspberry Pi, where the pin RY-VCC (or JD-VCC)of the Jumper is connected to the VCC at 5V of the microprocessor. The GND from the Jumper and the other GND, as well as the IN1 and IN2, are connected to the GND of the microprocessor. Additionally, the VCC is connected to the GPIO-23 of the Raspberry Pi. Furthermore, the fans connected to the relay module are connected to COMM (the Negative Power) and the Normally Closed.

Momentary Push Button Switch PBS-110

Figure 5-22. Momentary Push Button Switch PBS-110

The Momentary Push Button Switch PBS-110, shown in Figure 5-22, temporarily closes the normally open contact to the common contact. This action enables brief circuit activation, often utilized in reset mechanisms or as manual control inputs in electronic devices and consumer electronics. It facilitates system operation by allowing users to open and close circuits as needed quickly.

The push button's positive is connected to the Raspberry Pi's GPIO-3, and the negative is connected to the Raspberry Pi's GND. This project utilizes this project to power on and off the Raspberry Pi.

Raspberry Pi Charger Cable

Figure 5-23. Raspberry Pi Charger Cable

The Okdo Raspberry Pi Power Supply features a USB Type-C connector, a 1.5m cable, and an EU plug type, making it a dependable option for powering Raspberry Pi boards. It accommodates various Raspberry Pi models and provides ample power for consistent and reliable operation.

Extension Cord

Figure 5-24. Extension Cord

An extension cord, illustrated in Figure 5-24, is designed to extend the reach of power sources to devices that need placement beyond the proximity of standard outlets. In this project, the extension cord's primary role is to create an extended pathway for the Raspberry Pi and the power supply to access power from a distant location within the building or area.

5.5 Software Design and Development

5.5.1 Graphical User Interface

dotBean offers two applications: for the Administrator and the User. The Administrator mainly handles the flow of the data received from the Raspberry Pi through its Website Application, which consists primarily of user logs, sensor logs, and actuator logs. On the other hand, the user or the client can access the mobile application to monitor the updates inside the dotBean device.

5.5.1.1 Client Side - Mobile Application

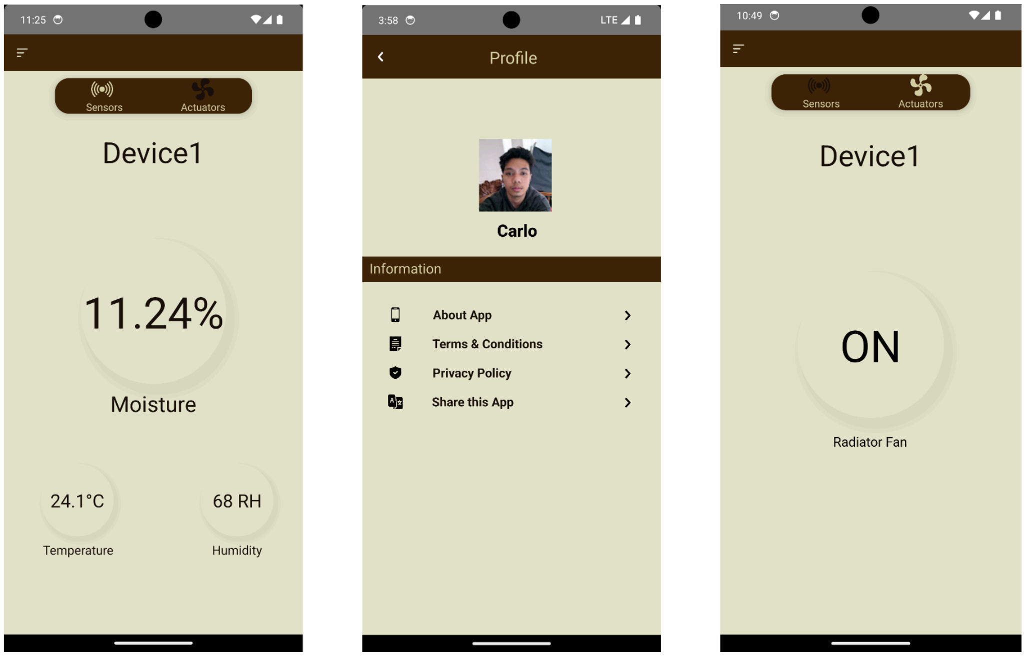


Figure 5-25. dotBean Mobile Application UI

For the client side, the developers focused on designing a mobile application to offer real-time and user-friendly monitoring capabilities. In this application, as depicted in Figure 5-25, users are directed to the sensor dashboard once they log in. The main page of the mobile application consists of two tabs: "Sensors" and "Actuators." Under the "Sensors" tab, users can view the values for Temperature, Humidity, and Moisture levels. On the other hand, the "Actuators" tab allows clients to monitor the status of the Fan.

5.5.1.2 Admin Side - Website Application

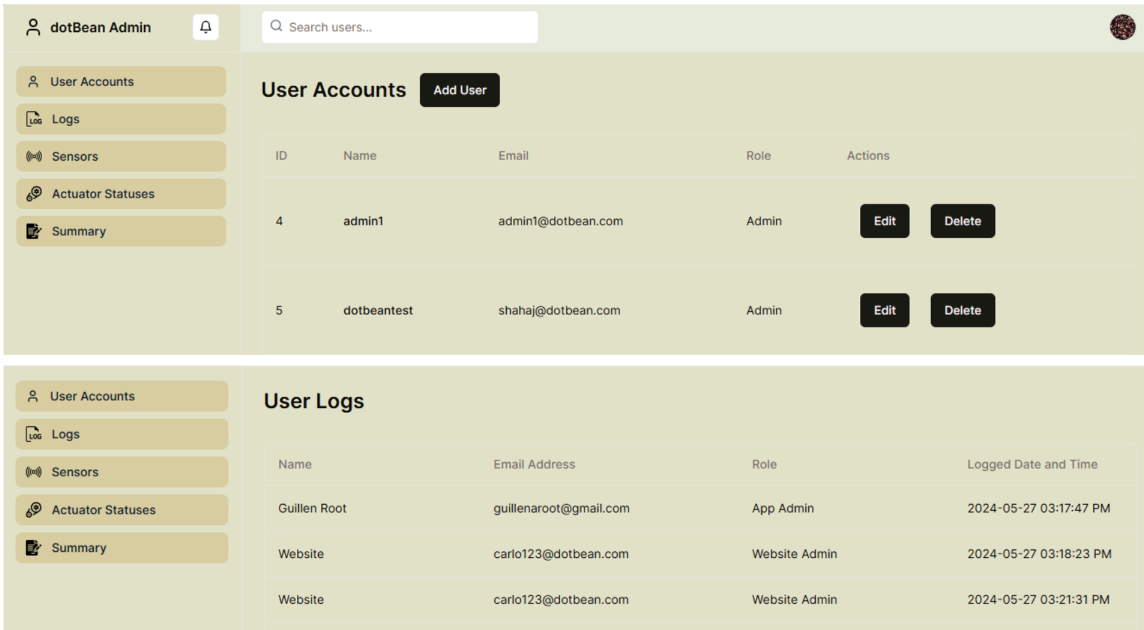


Figure 5-26. dotBean Website Application UI - User Accounts & Logs Pages

Once the admin successfully accesses the website, they will automatically redirect to the User Accounts Page homepage, as presented in Figure 5-26. In this user accounts page, the admin can view, add, edit, and delete any accounts logged in the given storage device. The sidebar menus are located on the left side of the page, where all the accessible pages can be viewed. User longs, on the other hand, are mainly used to monitor all accounts logged into the mobile and website application. The admin can monitor the user's name, email address, designated role, and the date and time they logged in. These pages ensure that all those authorized users are the only ones accessing the storage device data and information.

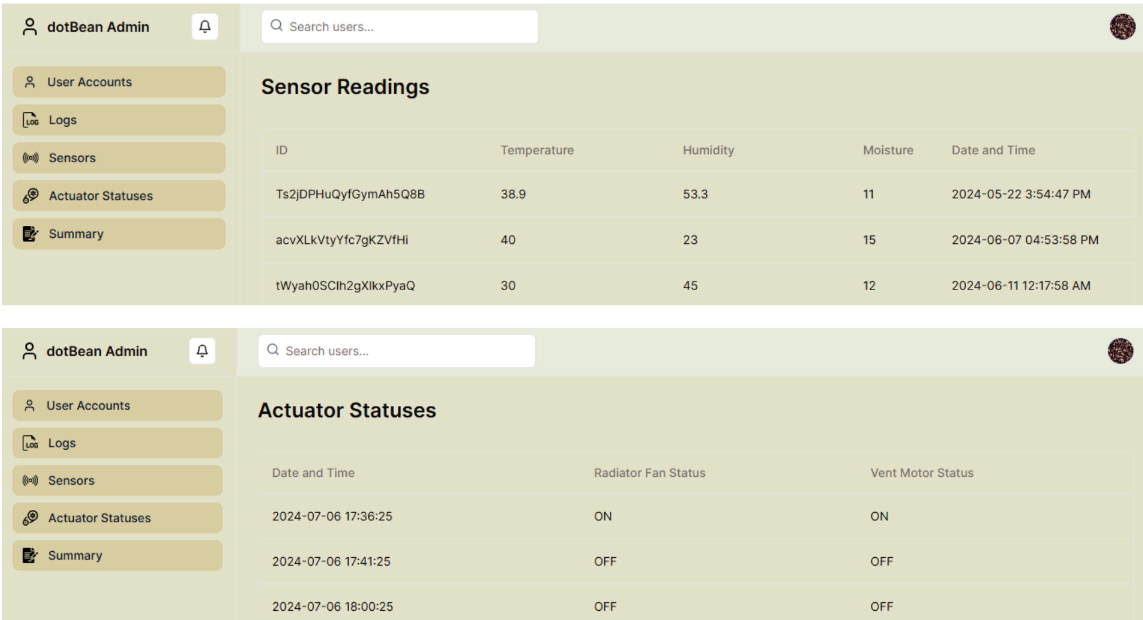


Figure 5-27. dotBean Website Application UI - Sensor Readings & Actuator Statues Page

When the admin viewed the Sensor Readings, all the sensor values in the mobile application were shown, as shown in Figure 5-27. Unlike in the mobile application, users can only see the values of every sensor and actuator one at a time; here on the admin website, the admin has access to all the data the database receives from the Raspberry Pi. Similarly, with the sensor readings page, the actuator statuses have all the logs of all radiator statuses and the date and time it was changed from the Raspberry Pi.

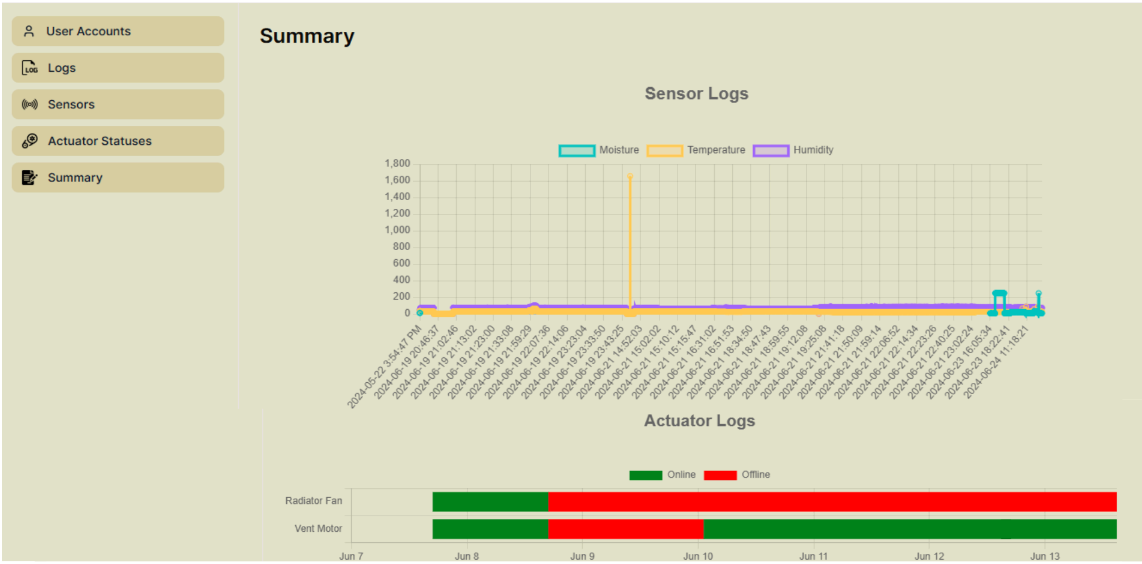


Figure 5-28. dotBean Website Application UI - Summary Page

Figure 5-28 presents the summary page of the admin website, which displays a graph summarizing every sensor log and actuator log. Sensor Logs are graphed in line, whereas actuator logs are graphed in bar.

The website's primary purpose is to manage account users using the system. The admin can set which only user accounts can access the device's application and who can only access the admin site. Once logged in, the admin will redirect to the "User Accounts" page. On its left side, the website has this sidebar menu that all other pages can access; this includes "User Logs", "Sensors Readings," "Actuator Statuses," and "Summary".

5.5.2 Data Flow Diagram

The below diagrams show a detailed discussion of the dotBean data flow during its operation. The DFD Level 0 and DFD Level 1 are presented as follows:

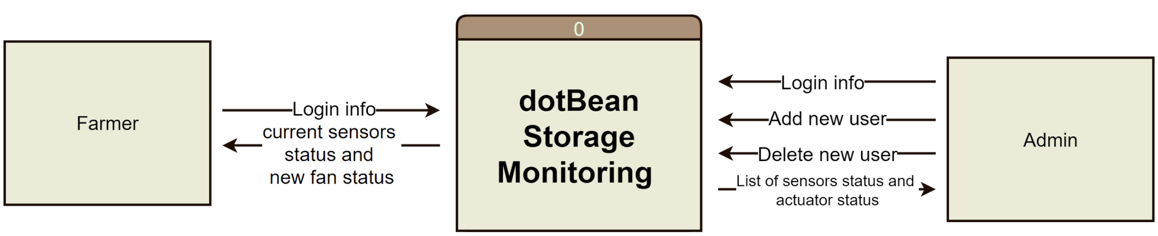


Figure 5-29. dotBean DFD Level 0

A Data Flow Diagram (DFD) is a formal representation of a system's structure that facilitates comprehension by both programmers and users. This visual representation clearly illustrates the flow of data within the system and the processes that manage and manipulate this data. DFDs are essential in system design as they visualize the system's operations and the routes through which data flows. DotBean's DFD level 0 illustrates the data exchanged between the actors and the system and the reverse (see Figure 5-29), providing a general overview of how data flows between external entities and the system. Below is the DFD level 1, which explains more in-depth the processes that occur inside the system:

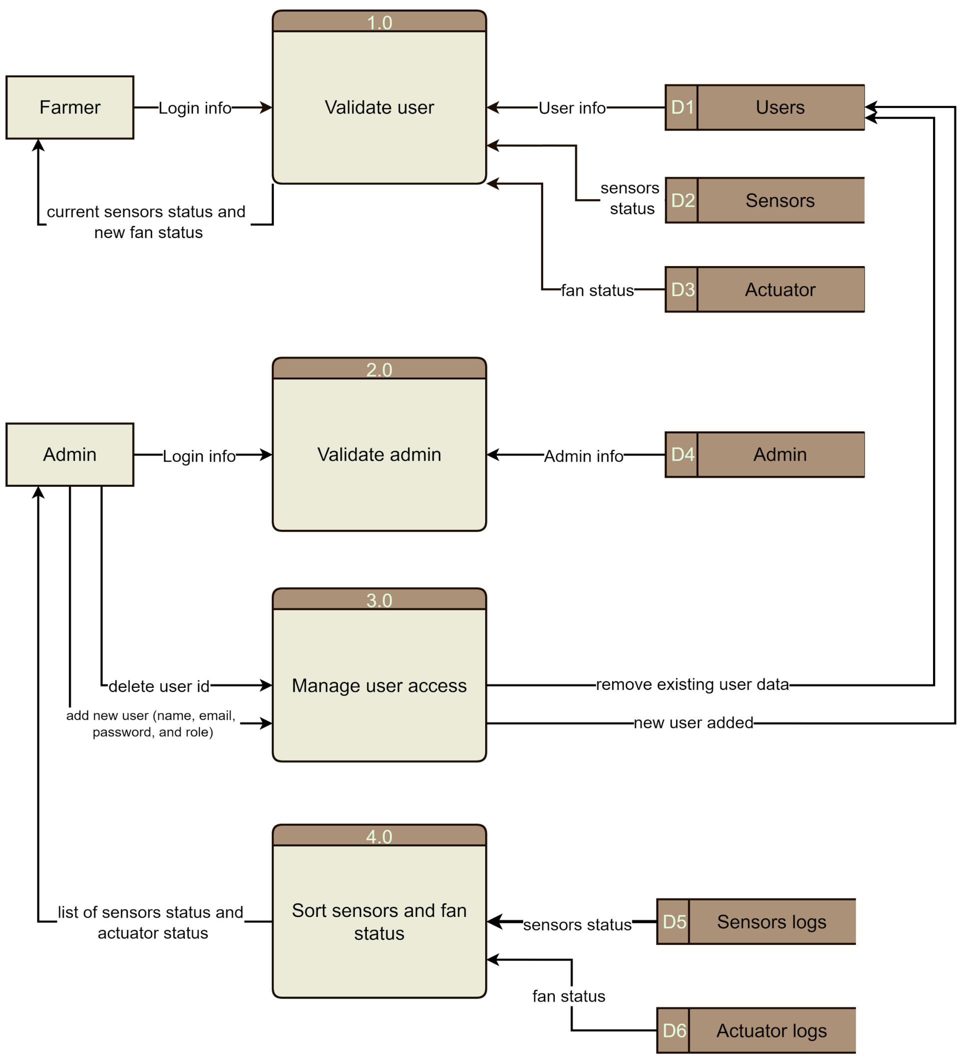


Figure 5-30. dotBean DFD Level 1

The DFD Level 1, as presented in Figure 5-30, details how information flows between the Farmer and Admin users, the processes that handle their requests, and the associated data stores that keep critical system information.

The diagram begins with Process 1.0: Validate User, where the Farmer provides their login credentials, including their email and password, to access the system. The login information is processed through the "Validate User" function, which cross-references the provided credentials with those stored in Data Store D1: Users. If the credentials match, the system retrieves the user information and confirms the user's access to the platform. Once logged in, the Farmer can view the current status of sensors and fans. This data is fetched from Data Store D2: Sensors, which holds information on environmental metrics such as temperature, humidity, and moisture content, and Data Store D3: Actuator, which stores the status of the fan. If there are updates to the fan status, the new information is sent back to the Actuator data store to keep the records current.

Parallel to this, Process 2.0: Validate Admin shows the actions for an Admin logging into the system. Like the Farmer login, the Admin provides login credentials verified against Data Store D4: Admin. The Admin can manage the system's user data and view relevant status information if the credentials are authenticated.

The Admin can use Process 3.0: Manage User Access to add or remove users upon successful authentication. This process involves two key actions: adding a new user or deleting an existing one. To add a new user, the Admin inputs details such as the user's name, email, password, and role. This information is then stored in Data Store D1: User, and the newly added user gains access to the application. Conversely, if the Admin decides to delete a user, they specify the user ID to remove, and the system removes the relevant information from the data store, effectively denying future access to that user.

The Admin also can monitor environmental data and fan statuses through Process 4.0: Sort Sensors and Fan Status. In this process, the Admin can view the list of sensor statuses, which the system retrieves from Data Store D5: Sensors Logs. Additionally, the Admin can view information regarding the fan status, which is fetched from Data Store D6: Actuator Logs in

Process 4.0. This gives the Admin a comprehensive view of the average temperature, humidity, calibrated moisture content, and fan status.

5.5.3 Algorithm

A static threshold algorithm can help regulate and optimize the system's operations based on predefined thresholds when implementing an automated control system that depends on the storage conditions. As suggested in its name, threshold refers to the maximum score of any tuple that hasn't been encountered by reading from the various indices (Zhang, H. & Li, D., 2014). The threshold-based algorithm has two types: dynamic and static.

The static threshold-based algorithm maintains stability and efficiency in control systems by providing a fixed criterion for triggering actions based on sensor data or system states. Ge et al. (2019) emphasize the significance of static event-triggered conditions in distributed event-triggered filtering or estimation algorithms, where thresholds remain constant throughout the system's operation. By having predetermined thresholds, the algorithm can make decisions promptly without needing continuous adjustments, ensuring a streamlined and predictable control process.

Conversely, as He et al. (2021) discussed, dynamic threshold-based algorithms introduce flexibility by allowing thresholds to adapt based on changing conditions. In the context of control input and system output signals, dynamic thresholds enable the algorithm to adjust the triggering conditions dynamically, optimizing the transmission of signals and enhancing overall system performance. This adaptability is particularly beneficial when environmental conditions or system dynamics vary over time, requiring a more responsive control mechanism.

Threshold-based algorithms are commonly utilized in IoT systems to make decisions based on predefined conditions. A static threshold-based algorithm controls the system's operation in the project context. The system initializes devices like DHT devices, capacitive moisture sensors, and relay modules and then displays temperature, humidity, and moisture content on a GUI. The algorithm checks if the average temperature is below a specific threshold, in this case, 19.8 degrees, to determine whether to turn on the fan. Once these decisions are made, the fan and sensor data status are sent to Firebase for user access.

In this scenario, a static threshold-based algorithm simplifies decision-making by setting fixed temperature and moisture content values that trigger specific actions. This approach is efficient and straightforward, suitable for scenarios where the conditions for action are well understood and do not require frequent adjustments. Using static thresholds, the system can quickly determine when to activate the fan based on temperature and moisture readings without complex computations or continuous monitoring.

This algorithm aligns with optimizing spatial capacity under constraints like reliability and security, as mentioned in Lenong's work (2024). The static thresholds provide a clear decision-making framework, ensuring that actions are taken promptly based on predefined conditions. Additionally, the system's reliance on fixed thresholds resonates with age-dependent random access protocols, where devices access channels based on specific thresholds (Chen et al., 2020).

5.6 System Testing

5.6.1 Testing Protocol

The following procedures are conducted after the drying process:

1. Two beans are stored: parchment and GCB (green coffee beans). The parchment still has skin, while the GCB has no skin.
2. Parchment and GCB have the same protocol. Once received, the GCB should contain at least 11% to 12% moisture content. If it's 14%, it'll still be stored, but the staff should air dry it to lower the moisture content for more extended storage.
3. Even with a 12.5% moisture content, it's okay if the ideal room temperature is not too cold, ideally 20°C —25°C.
4. It has to be stored at 11%—12% moisture content until the GCB is ready for roasting. The same parchment or GCB must have at least 11% moisture content to store and roast. If not, it should be air-dried again and stored until it's ready for roasting.

This is where the dotBean system comes into play after the drying process. The GCB is placed inside the storage, and the system will initialize the moisture content of the coffee beans and the storage temperature. If the coffee beans' moisture content is below 14%, they will be stored at the intended time until they are ready for roasting. Otherwise, instead of air drying them outside, the system will dry them using the fan, and since a Polyolefin Foam surrounds the cover material, it'll absorb moisture.

5.6.2 Moisture Sensor Calibration

Table 5-1. Moisture Tester vs. Sensor Value Data Points

Moisture Tester	Sensor Value
11.6	573
12.3	567
11.9	574
12.1	576
11.7	572
12.1	576
12.1	574
12.1	576
12.3	574
12.3	577
12.3	576
11.6	573
12.1	575
12.3	571
11.9	574
12.3	576
12.4	571
11.9	574

Moisture Tester	Sensor Value
12.1	575
11.6	573
12.1	576
12.5	577
11.7	571
11.9	572
12.3	577
12.1	576
12.5	577
12.1	575
12.1	571
11.6	573
11.9	574
11.7	573
12.5	571
11.9	573
12.3	572
12.5	572
12.1	574
11.9	574
12.3	573
12.3	574
11.9	573

Moisture Tester	Sensor Value
11.7	575
12.5	570
11.7	574
11.9	572
11.6	571
12.1	572
11.7	575
11.9	573
12.3	572

Before setting the slope and intercept that will be used for the moisture sensor, the developers first identified the corresponding sensor value of the moisture from the tester's output. The developers conducted at least 50 tests and set the 622 at 0; this means that when the sensor value is at 622, it's not touching any materials or coffee beans. With this, the slope and intercept are determined, and using the code below (*see Figure 5-31*), the developers continued their testing for the moisture sensor.

```
float slope = -0.04; // Calculated slope based on data points
float intercept = 34.65; // Calculated intercept based on one data point
float calibratedMoistureContent = (slope * sensorValue + intercept); // y
= mx + b
```

Figure 5-31. Getting the Moisture Value Snippet Code

Before determining the adjusted moisture reading from the sensor, it's first required to determine the slope and intercept. With this, Figure 5-31 shows the calculated mean and intercept of the sensor based on the given data shown in Table 5-2. The slope indicates that the calibrated moisture content decreases at -0.04 as the sensor value increases. The intercept adjusts the baseline measurement, setting the calibrated moisture content to 34.65 when the sensor value is zero. The final line calculates the calibratedMoistureContent using the linear equation y = mx + b, where m is the slope, x is the sensor value, and b is the intercept. This equation transforms the raw sensor readings into a calibrated value, aligning them with the expected moisture content by applying the specified linear adjustment.

Table 5-2. Moisture Tester vs. Calibrated Sensor Value Data Points

Test #	Moisture Tester	Calibrated Sensor Value
1	11.7	11.65
2	11.9	11.93
3	11.6	11.61
4	11.7	11.73
5	11.7	11.65
6	11.9	11.89
7	11.7	11.73
8	11.7	11.69
9	11.9	11.89
10	11.7	11.65
11	11.9	11.89
12	11.7	11.77
13	11.7	11.69
14	11.7	11.73
15	11.7	11.69
16	11.4	11.41
17	11.7	11.61
18	11.9	11.93
19	11.9	11.85
20	11.7	11.65

Test #	Moisture Tester	Calibrated Sensor Value
21	11.9	11.97
22	12.3	12.29
23	11.7	11.69
24	11.7	11.85
25	11.9	11.93
26	11.7	11.73
27	11.7	11.69
28	11.7	11.77
29	11.9	11.89
30	11.9	11.85
31	11.9	11.89
32	11.9	11.93
33	11.9	11.93
34	11.6	11.61
35	11.9	11.97
36	11.9	11.89
37	12.1	12.05
38	11.9	11.93
39	11.9	11.89
40	12.5	12.49
41	12.5	12.53
42	12.3	12.25
43	12.1	12.13

Test #	Moisture Tester	Calibrated Sensor Value
44	12.1	12.09
45	11.9	11.89
46	12.1	12.05
47	11.9	11.93
48	11.9	11.89
49	11.9	11.93
50	11.9	11.85

After getting the slope and y-intercept from the data points, the developers conducted at least 50 tests (see Figure 5-3) to determine how close the moisture result of the moisture tester and the calibrated moisture sensor. Below are the two datasets' mean, variance, and standard deviation.

```
Moisture Tester - Mean: 11.87 Variance: 0.05 Standard Deviation: 0.21
Moisture Sensor - Mean: 11.87 Variance: 0.04 Standard Deviation: 0.21
```

Figure 5-32. Moisture Tester vs. Calibrated Sensor Value Mean, Variance, and Standard Deviation

The statistical analysis reveals that the Moisture Tester and the Moisture Sensor exhibit similar performance characteristics, as Figure 5-32 illustrates. Both devices have a mean moisture reading of 11.87, indicating that their average measurements are identical. Regarding variability, the Moisture Tester has a variance of 0.05, while the Moisture Sensor has a slightly lower variance of 0.04. This means that the Moisture Sensor's readings are centered around the mean compared to the Moisture Tester's.

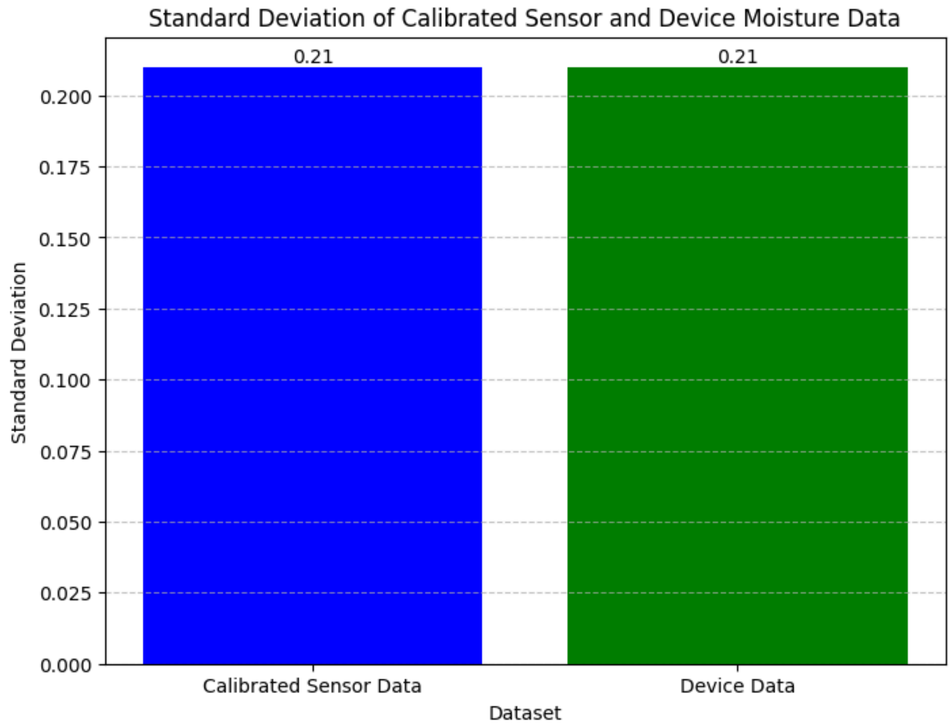


Figure 5-33. Standard Deviation of Moisture Tester and Calibrated Sensor Value Data

Despite this minor difference in variance, both devices have the same standard deviation of 0.21, as presented in Figure 5-33, which confirms that the overall spread of their measurements around the mean is equivalent. Thus, the analysis suggests that both devices provide comparable moisture readings and exhibit similar levels of consistency in their measurements.

5.6.2 Performance

Unlike during the Design Options' testing and validation phase, the developers conducted testing for this constraint on real-life scenarios, this time to confidently confirm whether the simulation on SolidWorks was realistically correct.

In this scenario, the developers set the temperature inside the dotBean at 18°C to test whether the fan can reach the least optimal temperature of 20°C within less than 2 minutes, as suggested in Design Option 2's testing and validation of the Performance Constraint.

Figure 5-34. dotBean Performance Physical Testing

Figure 5-34 shows the dotBean currently running at a given storage temperature of 18.5°C. The developers set the temperature to determine if the final design meets the criteria. The figure below shows the result of the conducted testing:

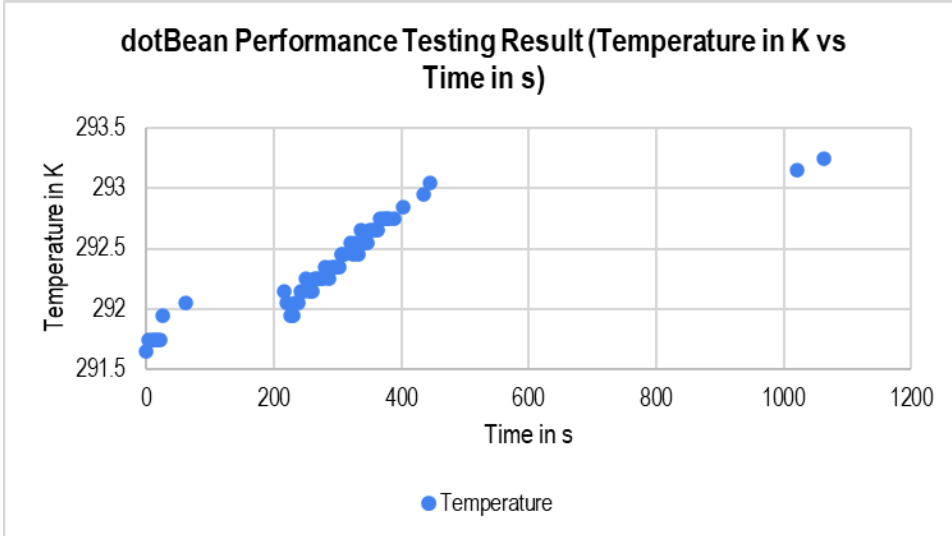


Figure 5-35. dotBean Performance Result (Temperature in K vs Time in s)

Figure 5-35 illustrates the system testing results in performance constraints in seconds. The initial temperature is noted 291.5 K or 18.5°C, the testing started at 7:48:28 PM and ended at 8:06:11 PM where the temperature is reached at 293.15 K or 20°C. The simulation lasted for 17 minutes and 43 seconds or 1063 seconds. As the system's running over time, the temperature also increased achieving the minimum optimal temeprature.

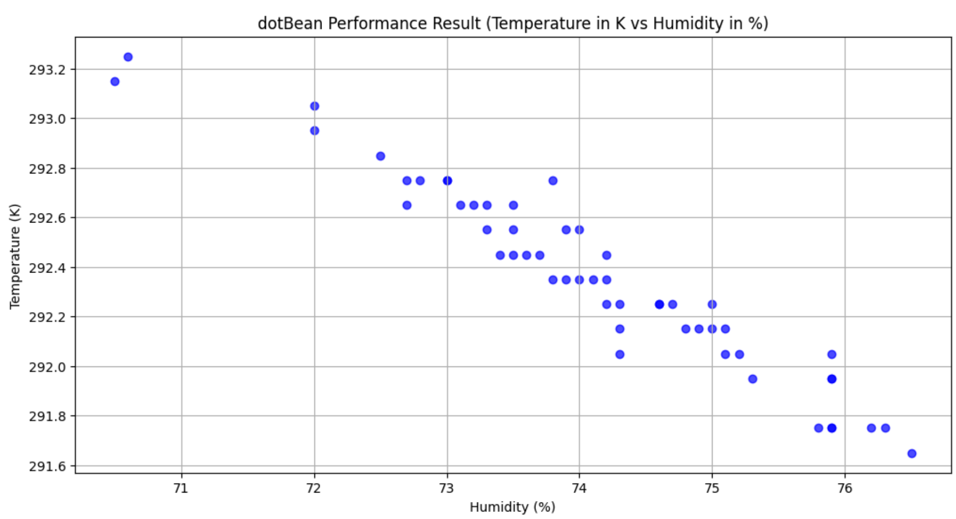


Figure 5-36. dotBean Performance Result (Temperature in K vs Humidity in %)

Figure 5-36 shows that as the temperature increases, the humidity drops until they both reach the optimal condition. At the end of the testing, the temperature is 293.25 K, and the humidity is 75%.

5.6.3 Response Time

During this testing, the developers measured how long the fan would turn on or off when activated by the Raspberry Pi. This testing ensures that the system reliably reacts to changes in temperature when detected.

Table 5-3. Final Design Response Time in Microseconds (μs) Result

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry pi Received the Reading in microseconds (μs)*	*Time the Fan Activated in microseconds (μs)*	*Response Time in microseconds (μs)*
1	19.7	ON	100	300	200
2	20	OFF	100	300	200
3	20	OFF	100	300	200
4	20	OFF	100	300	200
5	20	OFF	100	200	100
6	20	OFF	100	300	200
7	20.1	OFF	200	500	300
8	20	OFF	200	500	300
9	19.8	ON	100	300	200
10	20	OFF	100	300	200
11	20	OFF	100	300	200
12	20	OFF	0	100	100
13	20	OFF	200	400	200
14	20	OFF	100	300	200
15	19.9	ON	200	400	200
16	20	OFF	200	400	200
17	20	OFF	300	500	200
18	20	OFF	100	300	200
19	20	OFF	100	300	200

Test #	*Temperature Reading*	*Fan Status*	*Time the Raspberry pi Received the Reading in microseconds (μs)*	*Time the Fan Activated in microseconds (μs)*	*Response Time in microseconds (μs)*
20	20	OFF	100	300	200
21	19.8	ON	100	300	200
22	20	OFF	100	300	200
23	20.1	OFF	100	300	200
24	20	OFF	100	300	200
25	19.9	ON	100	400	300
26	20	OFF	0	100	100
27	20	OFF	100	300	200
28	20	OFF	100	300	200
29	19.7	ON	100	300	200
30	20	OFF	100	300	200
31	20	OFF	100	400	300
32	20.1	OFF	100	500	400
33	20.2	OFF	100	300	200
34	20	OFF	100	300	200
35	20	OFF	0	100	100
36	20.1	OFF	100	300	200
37	19.9	ON	100	300	200
38	20.1	OFF	100	300	200
39	20.1	OFF	100	300	200
40	20.1	OFF	0	100	100
41	19.9	ON	100	300	200
42	20	OFF	100	300	200
43	20.1	OFF	100	300	200
44	20.1	OFF	100	300	200
45	19.9	ON	100	300	200
46	20.1	OFF	100	300	200
47	20	OFF	100	300	200
48	20	OFF	100	300	200

Test #	Temperature Reading	Fan Status	Time the Raspberry pi Received the Reading in microseconds (μs)	Time the Fan Activated in microseconds (μs)	Response Time in microseconds (μs)
49	20	OFF	100	300	200
50	20.1	OFF	100	300	200
Average Response Time in microseconds (μs)					**202.04**

Table 5-3 shows the testing result of the dotBean response time. The tests were done by recording 50 tests of the time the Raspberry Pi received the reading and another 50 tests for the time the radiator fan reacted. Based on the response time testing results, it has an average of 202.04 microseconds.

5.6.4 Power Consumption

While validating the final design power consumption constraint, the developers still utilized the computation used during the testing and validation phase, as presented in Chapter 3.

Each component's dataset was used as a reference from different online marketplaces and articles to complete this computation. According to (Makerlab Electronics, n.d.-c), Raspberry Pi 4B 8GB has an operating voltage of 5V at 3A. Arduino Uno R3, on the other hand, has a voltage of 5V at 0.02A (Arduino Tips, Tricks, and Techniques, 2024). Additionally, according to Edmallon (2024), a Capacitive Soil Moisture Sensor has a voltage of 5V and a current of 0.03A. Moreover, the DHT22 Module has a voltage and current of 3.3V and 0.0025 mA, respectively (LME Editorial Staff, 2022), while the Radiator Fan has 12V voltage and a current of 10A (Powering a Radiator Fan With 12V DC Power Supply, n.d.). On the other hand, a Switching Power Supply Unit has a voltage of 12V and 16.7A (Makerlab Electronics, n.d.-a). Lastly, a 2-channel 12V-5V relay module has a voltage of 5V and a current of 10A (Makerlab Electronics, n.d.-a), while a Momentary Push Button, as stated by Smoot (2022), has a voltage of 5V and a 1A current.

Herewith, the succeeding table illustrates the computation and the result:

Table 5-4. Final Design Power Consumption Computation of Power in kWh

COMPONENTS	Qty.	V	I	P (in W)	kWh
Idle State					
Raspberry Pi 4B 8GB	1	5	3	15	0.015
Arduino Uno R3	1	5	0.02	0.1	0.0001
Capacitive Soil Moisture Sensor v2.0	1	5	0.03	0.15	0.00015
DHT22 Module	3	3.3	0.0025	0.02475	0.000025
Active State					
Radiator Fan	1	12	10	120	0.12
2-channel 5V Relay Module	1	5	10	50	0.05

Table 5-4 shows the power consumption of each item used in the final design. The radiator fan used for heating has a power consumption of 0.12kWh. Each item has its value in volts and amperes. These values are used to calculate the power in watts in each item. The Raspberry Pi contains 0.015 kWh, the capacitive soil moisture contains 0.015 kWh, the DHT22 Module with the quantity of 3 includes 0.000025 kWh, the 2-channel 5V relay module contains 0.05 kWh, and the momentary push button switch consists of 5 watts.

The developers have excluded the computation of the momentary push button because it doesn't consume significant power, with the electrical current involved being minimal—usually in the microampere range—since it is only used to detect state changes. The power supply is also excluded, as it doesn't add any extra load but supplies power to the connected devices. This means the power consumed by the power supply from the outlet is simply the total of the power used by all connected devices, plus the minimal inefficiency caused by heat and other factors inherent in the power supply.

Table 5-5. Final Design's Power and Energy Consumption Result

Power and Energy Consumption	State	
	Idle (at use)	**Active**
Total Power (W)	15.27475	185.27475
Total Kilowatt-hour (kWh) (1 hr)	0.0153	0.1853

Table 5-5 summarizes the power consumption for the device Design Option 2, categorized into two states: idle and active. In the idle state, the device consumes 15.27475 watts of power, reflecting the energy it requires while not actively performing tasks but remaining on standby. During active use, the power consumption increases to 185.27475 watts, demonstrating the additional energy needed to perform tasks. Power consumption over one hour is measured in kilowatt-hours (kWh), with the device using 0.0153 kWh in idle mode and 0.1853 kWh in active mode.

5.6.5 Client Usability Testing

The dotBean was tested at the Farm to Cup DIY Brew Bar located in La Trinidad, Benguet, for client evaluation under the supervision of Sir ELI. Before conducting any demonstration, the developers introduced the system's main features and explained how it works. The developers also discussed the objectives, scope, and limitations of the dotBean project.

Moving forward, the developers tested and demonstrated the system by running the device, which already contained green coffee arabica beans. After the demonstration, the developers interviewed Sir Eli. During this interview, the developers received feedback from the client. The client approval sheet was also signed during this phase.

Phase 1: Introduction

During this phase, the developers introduced the project objectives of dotBean, which are to maintain and monitor the moisture content of the green coffee arabica beans to prevent mold growth due to moisture absorption. The project scope and limitations were also clarified before the demonstration. The developers then introduced dotBean by presenting the final prototype and mobile application. The developers asked the client to download and install the application and log in using his Gmail account to facilitate the monitoring of storage conditions and the moisture content of the coffee beans.

Phase 2: Simulation

The demonstration followed shortly after the developers introduced the prototype and its application.

Figure 5-37. dotBean Client Simulation: Prototype Running

After the short demonstration and discussion of how the prototype functions, the developers instruct the client and test users to install and download the dotBean application. In this demonstration, the developers explain and discuss the features and functionalities of the application. The client logged in to the application using his Gmail account; the account was already registered on the Admin site. As can be seen on the application, the real-time moisture content of the coffee beans, the temperature of the dotBean and its humidity, and the status of the fan are shown.

Phase 3: Interview

After the client demonstration of the dotBean's operation, the developers conducted an open-ended interview with Mr. Eli to gather his feedback and assess the efficacy and usefulness of the proposed system:

Interview Question #1: What were your first impressions when you received and demonstrated the dotBean storage? Were the instructions clear and easy to follow?

"Yeah, yeah, oo. Yes, yes. Definitely very impressive yung sa una ko nga ini-imagine although make sense diba kasi prototype lang naman sya."

Interview Question #2: How user-friendly do you find the app or interface for monitoring the moisture, temperature, and humidity? Is there any feature you find particularly useful or challenging?

"Yeah. Very easy to use, wala naman na yun lang. I mean yung sa need namin, yon na yon. Yung moisture makita, yung humidity makita, yung other data makita, okay na yon sa'min."

Interview Question #3: How effectively does the dotBean device monitor and report the moisture content of your coffee beans? Have you noticed any changes in the quality of your coffee beans due to using the storage?

"Yes, effective naman sya kasi depende naman talaga si coffee sa moisture."

Interview Question #4: How helpful are the notifications and alerts regarding moisture, temperature, and humidity changes? Do you feel that they are timely and actionable?

“Yes, so yeah I think sabi mo kanina 3 seconds. Kahit na mag 1 minute okay pa rin kasi di naman masyado malaki yung effect nya.”

Interview Question #5: How satisfied are you with the overall performance of the dotBean? What improvements or additional features would you like to see in future versions?

“*Very satisfying. Gusto ko lang din makita na stable yung system and kailangan, yeah, talaga namin sya. Kailangan talaga namin na stable yung system na di sya nag che-change, yun lang.”*

In summary, the client expressed confidence that the system would yield long-term benefits and recommended scaling it up to enhance its broader applicability and efficiency. Following the interview, the developers took a photo with Sir Eli and formally signed the Client Acceptance Sheet (*see Figure 5-38*).

Figure 5-38. dotBean Photo Opportunity with Client

5.7 Summary of Findings

The dotBean system aims to maintain the moisture content of the coffee beans by monitoring the temperature and humidity inside the storage. High moisture absorption leads to high moisture content, which can degrade the quality and profitability of the coffee. Hence, monitoring the storage conditions is essential to maintaining its moisture content.

With this, the table below illustrates the result of the conducted final testing:

Table 5-6. Final Design Summary of Findings Result

Design Constraint	Result	Achieved
Performance (m)	17:43	YES
Response Time (μs)	202.04	YES
Power Consumption (kWh)	0.1853	YES

As previously mentioned, performance refers to how long the system adjusts the storage temperature when it drops below the recommended level. In this case, the final system manages the given storage condition in 17 minutes and 43 seconds. In addition, regarding response time, the developers test how long the fan reacts once the readings are received from Raspberry Pi. The system achieves an average of 202.04 microseconds. Lastly, the performance, on the other hand, is validated through mathematical computation. When in the active state, the system achieved a maximum of 0.1853 kWh.

Overall, the final design can achieve the three criteria: the system must accurately monitor the storage conditions and dampness of the coffee beans and achieve the optimal temperature value when it drops, the system must employ real-time monitoring, and the system must consume low power. The developers also interview the client after the demonstration. The client initially found the dotBean storage impressive, though acknowledging it as a prototype. He described the app interface as very user-friendly, particularly appreciating its ability to monitor moisture, temperature, and humidity effectively. The client noted the device's effectiveness in maintaining coffee bean quality, emphasizing the importance of moisture control for coffee. The user also found notifications timely and actionable, even with slight delays.

5.8 Assessment of the Attainment of Project Objectives

After completing testing and ensuring compliance with the established criteria, the developers moved on to evaluate the achievement of project objectives. This section reviews all test results to determine whether dotBean has met the project's goals. Below are the findings for the following objectives:

Project Objective #1: *Determine the correct temperature and humidity levels by utilizing a temperature and humidity sensor to closely monitor the coffee beans' moisture content inside a closed storage device.*

The developers conducted a three-day testing process, each lasting 12 hours. Every day, they replaced the batch of Arabica Green Coffee Beans with a different pack of coffee beans. With a 10-minute interval between system readings, the developers collected the latest sensor data within the storage. They compared it with actual readings from reference devices, a thermometer for temperature, and a moisture tester for coffee's moisture content.

Table 5-7. Project Objective #1 Test Results

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
August 17, 2024								
1	7:40:20	23.5	56	11.58	OFF	23.7	11.6	26.3
2	7:50:21	23.5	55.9	12.09	OFF	23.8	12.1	27.2
3	8:00:23	23.7	54.7	11.21	OFF	24.3	11.2	27.3
4	8:10:24	23.6	55.3	11.92	OFF	23.9	11.9	27.2
5	8:20:26	23.7	56.5	11.7	OFF	24.1	11.7	25.9
6	8:30:28	23.6	53.5	11.39	OFF	24.1	11.4	26.1
7	8:40:29	23.6	53.3	11.21	OFF	24.1	11.2	26.6
8	8:50:31	23.9	52.9	11.52	OFF	23.8	11.4	25.7
9	9:00:33	23.8	53.1	11.01	OFF	24	11	25.8
10	9:10:34	23.7	51.9	11.24	OFF	23.9	11.2	27.2
11	9:20:36	23.5	52	11.25	OFF	23.8	11.2	26.1
12	9:30:38	23.6	54	11.06	OFF	23.8	11	26.3
13	9:40:40	23.5	53	11.02	OFF	23.9	11	26.8
14	9:50:42	23.4	54.5	11.25	OFF	23.9	11.6	26.7
15	10:00:43	23.5	53.4	11.67	OFF	23.8	11.6	26
16	10:10:45	23.5	53.4	11.65	OFF	23.8	11.6	26.8
17	10:20:47	23.6	54.8	11.43	OFF	23.9	11.4	26.9
18	10:30:49	23.7	54.8	11.65	OFF	24	11.6	27.3

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
19	10:40:50	23.6	53.3	11.14	OFF	23.9	11.2	25.9
20	10:50:52	23.6	53.2	11.35	OFF	23.9	11.4	26.6
21	11:00:54	23.9	53.5	11.65	OFF	24.3	11.7	26.5
22	11:10:56	23.9	53.6	11.1	OFF	24.4	11	26
23	11:20:58	23.9	53.8	11.36	OFF	23.8	11.4	26.4
24	11:31:00	23.9	53.2	11.5	OFF	24	11.4	26.8
25	11:41:02	23.9	53.8	11.37	OFF	24.1	11.4	26.2
26	11:51:03	24	54.9	11.05	OFF	24.1	11	26.9
27	12:01:05	24.7	54.3	11.37	OFF	24.6	11.4	26
28	12:11:07	24.4	53.8	11.12	OFF	24.5	11	27.2
29	12:21:08	24.2	53.5	11.24	OFF	24.4	11.2	27
30	12:31:10	24.2	55.2	11.5	OFF	24.3	11.4	27.3
31	12:41:12	24.2	54.7	11.1	OFF	24.4	11	27.9
32	12:51:14	24.1	55.3	11.62	OFF	24.2	11.6	27.6
33	13:01:15	24.2	54.4	11.62	OFF	24.2	11.6	26.8
34	13:11:17	24.2	52.5	11.3	OFF	24.3	11.4	27
35	13:21:19	24.2	51.6	11.6	OFF	23.6	11.7	27.3
36	13:31:20	24.3	51.5	11.14	OFF	24.1	11.2	27.2
37	13:41:22	24.4	51.9	11.4	OFF	24.1	11.4	27.2
38	13:51:24	24.6	52.6	11.05	OFF	24.7	11	27.7

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
39	14:01:25	24.5	51.8	11.1	OFF	24.2	11	27.1
40	14:11:27	24.6	51.3	11.37	OFF	24.7	11.4	27.6
41	14:21:29	24.8	50.2	11.6	OFF	24.7	11.7	27.2
42	14:31:30	25.2	49.5	11.58	OFF	25.6	11.6	27.2
43	14:41:32	24.8	49.8	11.1	OFF	24.3	11	27.4
44	14:51:34	24.8	49.8	11.62	OFF	24.4	11.6	27.5
45	15:01:36	25	49.8	11.8	OFF	25	11.7	28.4
46	15:11:37	24.8	50.2	11.37	OFF	24.6	11.4	27.3
47	15:21:39	24.9	49.2	11.7	OFF	24.8	11.7	27.7
48	15:31:41	24.9	49	11.5	OFF	25	11.6	27.7
49	15:41:42	25.3	49	11.7	OFF	25.2	11.7	28.3
50	15:51:44	25	49.8	11.65	OFF	25.3	11.6	28.5
51	16:01:46	24.9	50	11.65	OFF	24.8	11.7	27.7
52	16:11:48	25	49.2	11.7	OFF	24.9	11.7	28.3
53	16:21:49	25	50.1	11.57	OFF	24.9	11.6	27.5
54	16:31:51	24.9	50.1	11.57	OFF	24.8	11.6	28.2
55	16:41:53	25	50	11.57	OFF	24.7	11.6	28
56	16:51:54	25	50.3	11.57	OFF	24.7	11.6	27.8
57	17:01:56	25.1	50.3	10.97	OFF	25.3	11	28.4
58	17:11:58	24.8	51	11.6	OFF	24.7	11.6	27.3

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
59	17:22:00	24.9	50.1	11.6	OFF	24.8	11.6	28.6
60	17:32:01	25.1	50	11.17	OFF	25	11.2	28.1
61	17:42:03	24.7	50	11.6	OFF	24.4	11.6	29
62	17:52:05	24.8	50.2	11	OFF	24.7	11	27.7
63	18:02:07	24.8	49.8	11.65	OFF	24.7	11.7	27.9
64	18:12:08	24.9	50	11.5	OFF	24.6	11.7	28.3
65	18:22:10	25	49.5	11.72	OFF	25.2	11.7	29.1
66	18:32:12	25.2	50.5	11.72	OFF	25.1	11.7	28.1
67	18:42:13	24.8	50.4	11.87	OFF	24.6	11.7	28.2
68	18:52:15	24.9	49.5	11.3	OFF	24.8	11.2	27.9
69	19:02:17	24.9	50	11.54	OFF	24.7	11.4	28.2
70	19:12:18	24.5	50.1	11.2	OFF	24.6	11	27.5
71	19:22:20	24.6	51	11.37	OFF	24.5	11.4	27.4
72	19:32:22	25.2	50.8	11.15	OFF	25.3	11.2	28.5
73	19:42:24	24.5	50.5	11.6	OFF	24.3	11.6	28.3
August 18, 2024								
74	7:40:20	22.5	56.2	11.54	OFF	22.5	11.6	26.2
75	7:50:21	22.2	55.3	11.82	OFF	22.1	11.9	26.3
76	8:00:23	22.1	54.8	12.1	OFF	22.3	12.1	27.9
77	8:10:25	22	54.2	11.18	OFF	22.1	11.2	27.3

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
78	8:20:26	21.8	54.4	11.12	OFF	22	11.2	26.4
79	8:30:28	21.7	54.4	11.68	OFF	21.4	11.7	26.1
80	8:40:29	21.7	54.1	11.8	OFF	21.6	11.7	26.6
81	8:50:31	21.6	54.1	11.8	OFF	21.9	11.7	26.8
82	9:00:33	21.5	54.9	11.62	OFF	21.7	11.6	26.4
83	9:10:35	21.7	54.9	11.95	OFF	21.7	11.9	25.9
84	9:20:36	21.6	54.6	11.45	OFF	21.8	11.4	26.2
85	9:30:38	21.3	54.9	11.45	OFF	21	11.4	26.3
86	9:40:40	21.3	55	11.75	OFF	21.5	11.7	26.3
87	9:50:42	21.1	55.1	11.95	OFF	21.3	11.9	25.7
88	10:00:43	21.2	55	11.3	OFF	21.1	11.2	25.9
89	10:10:45	21.3	54.7	11.75	OFF	21.4	11.7	26.1
90	10:20:47	21.4	55.6	12	OFF	21.7	11.9	26
91	10:30:48	21.4	56.3	11.85	OFF	21.6	11.7	25.8
92	10:40:50	21.3	56.2	11.7	OFF	21.2	11.6	26
93	10:50:52	21.4	55.3	11.45	OFF	21.7	11.4	25.8
94	11:00:54	21.4	55.3	11.5	OFF	21.5	11.4	25.7
95	11:10:55	21.5	55.4	12	OFF	21.6	11.9	26.6
96	11:20:57	21.6	54.9	11.8	OFF	21.7	11.7	26.3
97	11:30:59	21.2	56.3	11.85	OFF	21.1	11.7	26.5

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
98	11:41:00	21.4	56.5	12	OFF	22.1	11.9	26.1
99	11:51:02	21.3	56.2	12.16	OFF	21.5	12.1	25.8
100	12:01:04	21.5	57.1	11.5	OFF	21.5	11.4	25.7
101	12:11:06	21.6	56.2	11.14	OFF	21.5	11.2	25.6
102	12:21:07	21.5	56	11.56	OFF	21.6	11.4	25.6
103	12:31:09	22.1	55.8	11.25	OFF	22.3	11.2	25.6
104	12:41:11	22	55.4	11.25	OFF	22.1	11.2	25.2
105	12:51:12	21.9	56.8	11.25	OFF	22.2	11.2	25.3
106	13:01:14	22	55.9	11.3	OFF	22.3	11.2	25.8
107	13:11:16	22	55.9	11.79	OFF	21.8	11.2	26.3
108	13:21:17	22	55.9	11.54	OFF	22.1	11.6	26
109	13:31:19	22.3	56	11.54	OFF	22.4	11.4	26.1
110	13:41:21	22.2	56.1	11.89	OFF	22.5	11.7	26.7
111	13:51:23	22.1	55.9	11.89	OFF	22.4	11.7	26.3
112	14:01:24	22.2	55.1	11.89	OFF	22.1	11.7	25.9
113	14:11:26	22.4	54.9	11.95	OFF	22.4	11.9	25.9
114	14:21:28	22.4	55	11.3	OFF	22.4	11.6	26.2
115	14:31:33	22.2	55.1	11.57	OFF	22.5	11.4	26.3
116	14:41:35	22.1	54.7	11.5	OFF	22.3	11.6	26.3
117	14:51:37	22.7	53.5	11.57	OFF	22.6	11.6	26.1

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
118	15:01:38	22.4	52.9	11.75	OFF	22.5	11.7	25.9
119	15:11:40	22.5	53.2	11.55	OFF	22.6	11.4	26
120	15:21:42	22.7	53	12	OFF	22.8	11.9	26.1
121	15:31:43	23	54.5	11.87	OFF	23.1	11.7	26
122	15:41:45	22.5	53.6	11.65	OFF	22.8	11.6	26
123	15:51:47	22.6	53.2	11.44	OFF	22.7	11.4	26.2
124	16:01:48	23	55.5	12.1	OFF	22.9	11.9	26.1
125	16:11:50	23.1	53.3	11.48	OFF	23.1	11.4	26
126	16:21:52	23.1	53.8	11.67	OFF	23.4	11.6	26.2
127	16:31:54	23.3	53.8	11.59	OFF	23.6	11.6	26.4
128	16:41:55	23.3	53.9	11.64	OFF	23.3	11.6	26.2
129	16:51:57	23.4	53.7	11.23	OFF	23.2	11.2	26
130	17:01:59	23.3	54.3	11.54	OFF	23.5	11.4	26.1
131	17:12:01	23.4	53.9	11.54	OFF	23.5	11.4	26.1
132	17:22:02	23.4	53.8	11.18	OFF	23.5	11.2	26.7
133	17:32:04	23.5	52.9	11.38	OFF	23.4	11.4	25.9
134	17:42:06	23.4	52.1	11.38	OFF	23.6	11.4	26.4
135	17:52:08	23.4	52.5	11.44	OFF	23.4	11.4	26.3
136	18:02:09	23.4	53.4	11.83	OFF	23.5	11.6	26.5
137	18:12:11	23.4	52.2	11.67	OFF	23.4	11.7	26.2

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
138	18:22:13	23.3	52	11.88	OFF	23.1	11.7	26.1
139	18:32:15	23.2	52.5	11.35	OFF	23.1	11.4	26
140	18:42:17	23.2	52.3	11.7	OFF	23.7	11.7	26.2
141	18:52:19	23.2	53	11.47	OFF	23.3	11.4	26
142	19:02:21	23.1	52.5	11.37	OFF	23.3	11.4	26.2
143	19:12:23	23.1	52.4	11.64	OFF	23.1	11.7	25.7
144	19:22:25	23	52.3	11.5	OFF	23	11.4	25.9
145	19:32:26	22.9	52.2	11.5	OFF	23	11.4	25.8
146	19:42:28	23	52.2	11.79	OFF	23.1	11.6	25.8
August 19, 2024								
147	7:40:11	22.8	55.9	12	OFF	22.5	11.9	26.4
148	7:50:12	22.8	54.8	12	OFF	22.3	12.1	25.6
149	8:00:14	22.7	55.4	12.2	OFF	22.4	12.1	25.1
150	8:10:16	22.4	54.8	11.8	OFF	22.3	11.7	25.6
151	8:20:17	22.4	59.4	11.8	OFF	22.3	11.7	25.3
152	8:30:19	22.5	58.1	11.56	OFF	22.1	11.6	25.6
153	8:40:21	22.5	56.9	11.34	OFF	22.3	11.4	26.3
154	8:50:22	22.5	56.9	11.6	OFF	22.3	11.6	25.9
155	9:00:25	22.5	56.1	11.56	OFF	22.4	11.6	25.5
156	9:10:26	22.5	56.7	11.5	OFF	22.2	11.4	25.6

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
157	9:20:28	22.5	56	11.87	OFF	22.3	11.9	25.8
158	9:30:30	22.5	55.5	11.8	OFF	22.1	11.9	26
159	9:40:32	22.6	55	11.44	OFF	22.2	11.6	26
160	9:50:34	22.5	55.7	12.12	OFF	22.3	12.1	25.5
161	10:00:35	22.6	56.4	11.54	OFF	22.1	11.6	25.7
162	10:10:37	22.5	55.5	11.84	OFF	22.1	11.9	26.1
163	10:20:39	22.6	55.9	11.4	OFF	22.2	11.4	25.5
164	10:30:40	22.5	54.5	11.32	OFF	22.6	11.4	26.1
165	10:40:42	22.5	56	11.6	OFF	22.2	11.6	26.4
166	10:50:44	22.5	55.5	11.57	OFF	22.3	11.6	25.9
167	11:00:46	22.6	55.9	11.34	OFF	22.8	11.2	25.3
168	11:10:47	22.5	55.5	11.37	OFF	22.8	11.4	25.1
169	11:20:49	22.6	55.7	11.67	OFF	22.8	11.7	24.9
170	11:30:51	22.6	54.3	11.25	OFF	22.5	11.2	25.7
171	11:40:52	22.6	53.5	11.72	OFF	22.9	11.7	25.2
172	11:50:54	22.6	53.9	11.56	OFF	22.5	11.4	25.4
173	12:00:56	22.6	53.6	11.54	OFF	22.6	11.6	25.9
174	12:10:58	22.5	54.5	11.6	OFF	22.3	11.6	25
175	12:20:59	22.5	54.8	11.34	OFF	22.7	11.4	24.8
176	12:31:01	22.7	55.3	11.47	OFF	22.9	11.4	25.6

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
177	12:41:03	22.6	53.8	11.7	OFF	22.7	11.7	25.4
178	12:51:05	22.6	53.7	11.87	OFF	22.3	11.9	25.8
179	13:01:06	22.7	54.6	11.77	OFF	22.3	11.7	25.4
180	13:11:08	22.8	57.1	11.69	OFF	23.1	11.7	26.2
181	13:21:10	23	55	11.7	OFF	22.7	11.7	25.4
182	13:31:11	23.1	55.9	11.42	OFF	22.9	11.4	25
183	13:41:13	23.2	55.8	11.4	OFF	23.4	11.4	25.3
184	13:51:15	23.3	54.8	11.25	OFF	23.2	11.2	25.4
185	14:01:16	23.3	54.6	11.7	OFF	23.6	11.7	25.1
186	14:11:18	23.5	54.7	11.65	OFF	23.6	11.6	25.5
187	14:21:20	23.4	55.3	11.73	OFF	23.8	11.6	25.1
188	14:31:21	23.5	57.8	11.57	OFF	23.9	11.6	25.7
189	14:41:23	24	58.2	11.9	OFF	24.4	11.9	26.2
190	14:51:25	24.4	56.1	11.64	OFF	24.8	11.6	26.3
191	15:01:26	24.7	53.9	11.64	OFF	24.9	11.6	26.5
192	15:11:28	24.8	54.1	11.7	OFF	25.2	11.7	26.8
193	15:21:30	25.1	52.1	11.75	OFF	25.4	11.7	27.6
194	15:31:32	24.7	52.7	11.7	OFF	24.9	11.7	26.7
195	15:41:33	25	52.3	11.87	OFF	25.1	11.9	26.8
196	15:51:36	24.5	51	11.93	OFF	24.5	11.9	26.9

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
197	16:01:37	24.5	56.2	11.93	OFF	24.9	11.9	27.5
198	16:11:46	24.2	53.5	11.42	OFF	24	11.4	27.1
199	16:21:49	24.2	51.1	11.8	OFF	24.2	11.9	27.4
200	16:31:51	24.2	60	12.1	OFF	25	11.9	26.9
201	16:41:52	24.9	55.2	12	OFF	25.4	12.1	27.3
202	16:51:54	24.8	53.3	11.74	OFF	24.6	11.7	26.4
203	17:01:56	23.9	52.3	11.93	OFF	24	11.9	26.1
204	17:11:58	23.4	51.9	11.93	OFF	23.4	11.9	26.4
205	17:21:59	23.1	51.7	11.54	OFF	23.2	11.6	26.3
206	17:32:01	22.8	51.5	11.65	OFF	22.8	11.7	25.9
207	17:42:03	22.5	51.5	11.54	OFF	22.8	11.6	25.7
208	17:52:04	22.4	51.2	11.45	OFF	22.6	11.4	25.8
209	18:02:06	22.2	51.5	12	OFF	22.5	11.9	25.6
210	18:12:08	22.1	51.7	11.64	OFF	22.3	11.6	25.6
211	18:22:09	22	51.8	12	OFF	22.2	11.9	25.6
212	18:32:11	22	52.2	11.7	OFF	22.3	11.7	25.7
213	18:42:13	22	52.4	11.64	OFF	22.1	11.6	25.3
214	18:52:14	21.8	53.2	11.87	OFF	21.5	11.9	25.2
215	19:02:16	22.1	57.3	11.84	OFF	22.6	11.7	25.5
216	19:12:18	23.3	61.8	11.72	OFF	23.4	11.7	26.1

Test #	Time	dotBean Readings				Actual Readings		Site Testing Condition
		Temperature	Humidity	Moisture	Fan Status	Temperature	Moisture	Temperature
217	19:22:19	24.3	59.9	11.9	OFF	24.7	11.9	26.3
218	19:32:21	24	54.3	11.95	OFF	24	11.9	25.8
219	19:42:23	23.5	53.9	12	OFF	23.6	11.9	25.7

As the table suggests, on the first day, the temperature ranges from 23.4°C to 25.3°C, humidity from 49% - 56.5%, and moisture content from 10.97% - 12.09%. On the second day, the temperature ranges from 21.1°C to 23.5°C, humidity from 52% to 57%, and moisture from 11.12% to 12.16%. Lastly, on the third day, the temperature ranges from 21.8°C to 25.1°C and humidity from 51% to 61.8% from 11.25% to 12.2%. Overall, while the site testing has a temperature ranging from 24.8°C - 29.1°C within three days from 7:40 AM - 7:40 PM, the storage temperature maintains its ideal temperature between 20°C - 25.4°C.

Given the variations observed in the conditions, it's essential to quantify the dispersion of these measurements.The developers can assess the sensors' reliability by determining the standard deviation of the data's variability. This statistical measure will help identify any deviations that might affect the quality and stability of the sensors.

The developers calculated the mean, variance, and standard deviation. Below formulas were used to calculate the mean, variance, and standard deviation according to King (2020):

$$\bar{x} = \frac{\Sigma xi}{n} \qquad \textit{(Equation 8)}$$

Where:

- $\bar{x}$ is the mean value of all observations
- Σxi is the sum of all observations
- n is the number of scores

$$S^2 = \frac{\Sigma(xi - \bar{x})^2}{n - 1} \qquad \textit{(Equation 9)}$$

Where:

- S^2 is the variance
- $\Sigma(xi - \bar{x})^2$ is the sum of all differences between each observation and the mean squared
- n is the number of scores

$$S = \sqrt{S^2} \qquad \textit{(Equation 10)}$$

Where:

- S is the standard deviation
- $\sqrt{S^2}$ is the square root of the variance

```
df = pd.DataFrame({
    'TemperatureSensor': tsensor_data,
    'TemperatureReferencedDevice': tdevice_data
})

df = pd.DataFrame({
    'MoistureSensor': msensor_data,
    'MoistureReferencedDevice': mdevice_data
})

#temperature
#Convert to NumPy arrays for calculation
tsensor_array = np.array(tsensor_data)
tdevice_array = np.array(tdevice_data)
#Calculate mean
sensor_mean = np.mean(tsensor_array)
device_mean = np.mean(tdevice_array)
#Calculate variance
tsensor_variance = np.var(tsensor_array, ddof=1)
tdevice_variance = np.var(tdevice_array, ddof=1)
#Calculate the standard deviation
tsensor_std_dev = np.sqrt(tsensor_variance)
treferenced_device_std_dev = np.sqrt(tdevice_variance)

#moisture
#Convert to NumPy arrays for calculation
msensor_array = np.array(msensor_data)
mdevice_array = np.array(mdevice_data)
#Calculate mean
msensor_mean = np.mean(msensor_array)
mdevice_mean = np.mean(mdevice_array)
#Calculate variance
msensor_variance = np.var(msensor_array, ddof=1)
mdevice_variance = np.var(mdevice_array, ddof=1)
#Calculate the standard deviation
msensor_std_dev = np.sqrt(msensor_variance)
mreferenced_device_std_dev = np.sqrt(mdevice_variance)
```

Figure 5-39. Standard Deviation Snippet Code

Figure 5-39 shows how to compute the mean, variance, and standard deviation using NumPy. Initially, the code creates data frames for the test results for temperature and moisture. Then, the data is transformed into an array, and its average values for both temperature and humidity are found.

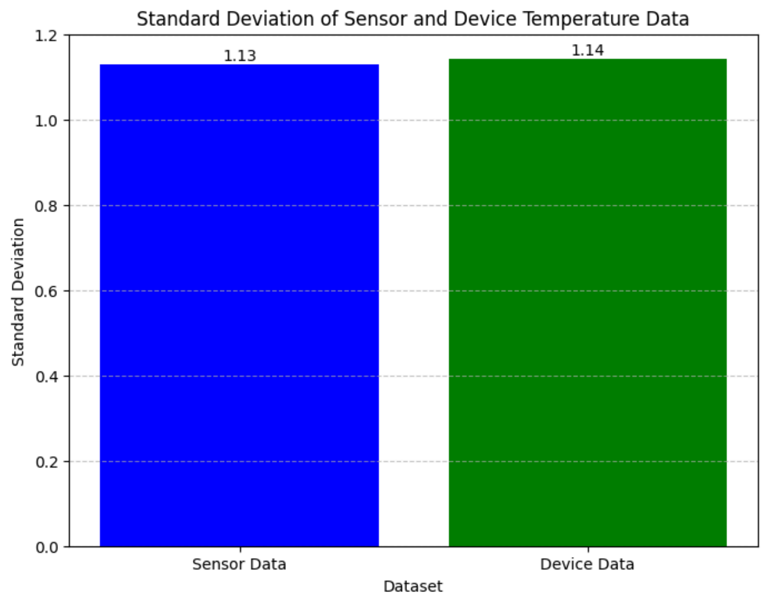


Figure 5-40. Standard Deviation of Sensor and Device Temperature Data

Figure 5-40 shows the bar chart of two comparable data's standard deviations from the previous testing: from sensor data, which is the DHT22, and the reference device, which is the temperature thermometer. The figure reveals that both datasets have similar levels of variability, with standard deviations of 1.13 and 1.14, respectively. Standard deviation refers to how dispersed the data is concerning the mean of given datasets (Omda, S. & Sergeant, S., 2023). This close similarity indicates that the temperature changes recorded by the sensor and the devices are consistent.

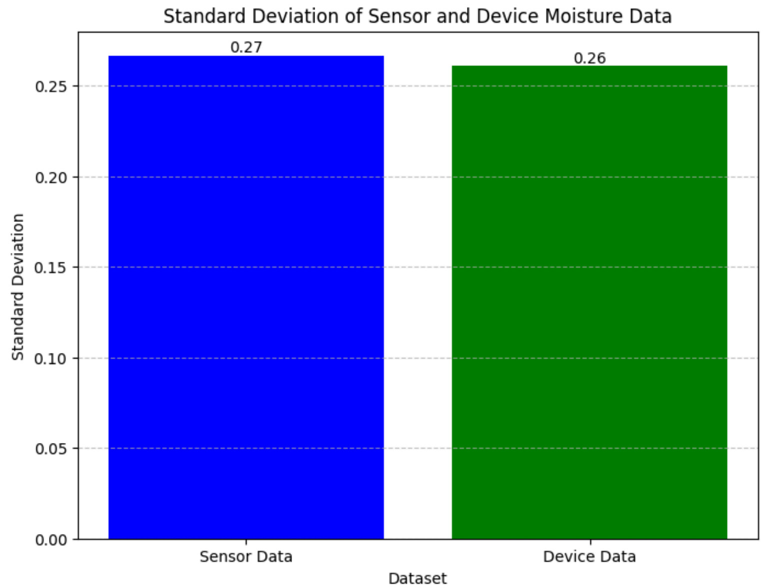


Figure 5-41. Standard Deviation of Sensor and Device Moisture Data

Similarly to Figure 5-40, Figure 5-41 depicts the sensor's and referenced device's standard deviation in terms of moisture data. The two datasets having a standard deviation of 0.27 and 0.26, respectively, suggest that the two devices, the Moisture Sensor and the Moisture Meter, have similar levels of variability. This also indicates that the moisture levels recorded by the sensor are almost consistent with the referenced device used.

For determining the correlation between two variables, the developers used the table below according to the Boston University Medical Campus website on their “PH717 Module 9 - Correlation and Regression” (n.d.):

Table 5-8. Correlation Coefficient and Description

Correlation Coefficient (r)	Description (Rough Guideline)
+1.0	Perfect (positive) + correlation
+0.8 to 1.0	Very strong + association
+0.6 to 0.8	Strong + association
+0.4 to 0.6	Moderate + association
+0.2 to 0.4	Weak + association
0.0 to +0.2	Very weak + or no association
0.0 to -0.2	Very weak - or no association
-0.2 to – 0.4	Weak - association

-0.4 to -0.6	Moderate - association
-0.6 to -0.8	Strong - association
-0.8 to -1.0	Very strong - association
-1.0	Perfect negative association

The Pearson correlation coefficient ranges from +1 to -1, where, as Table 5-7 suggests, +1.0 indicates a perfect positive + correlation. While on the other hand, +0.8 to 1.0 indicates a very strong + association. +0.6 to 0.8 indicates a strong + association, while +0.4 to 0.6 indicates a moderate + association. Moreover, +0.2 to 0.4 indicates weak + association. 0.0 to 0.02 indicates a very weak + or no association. Meanwhile, 0.0 to -0.2 indicates a very weak negative (-) or no association. -0.2 to -0.4 indicates a weak - association, while -0.4 to -0.6 indicates a moderate - association. Furthermore, -0.6 to -0.8 correlation indicates a strong - association. Lastly, -0.8 to -1.0 indicates a very strong - association, and -1.0 indicates a perfect negative association.

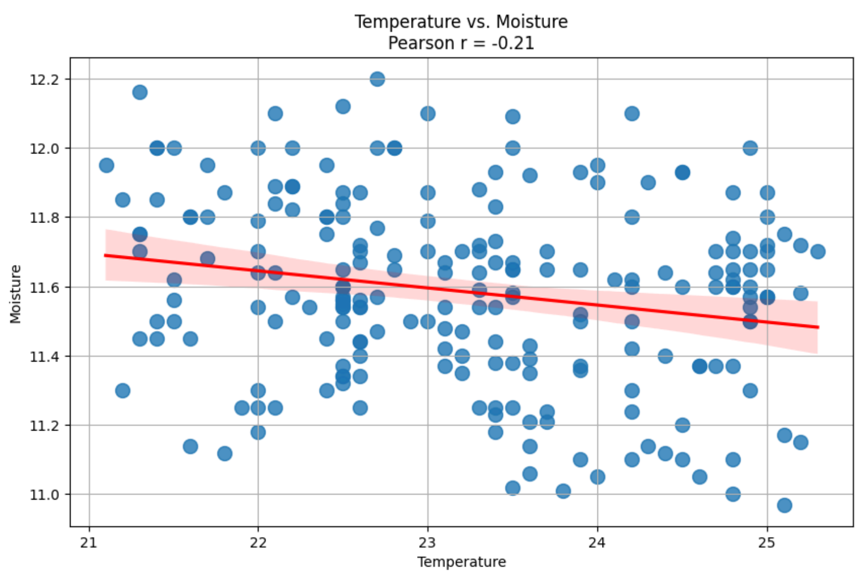


Figure 5-42. Temperature vs. Moisture Pearson Correlation Coefficient

Figure 5-42 presents the Pearson correlation coefficient of Temperature and Moisture data after the three days of testing. A Pearson correlation coefficient -0.21 indicates a weak negative correlation between the two variables. According to the BMJ (2020), between 0.20 - 0.39, there is a “weak negative” correlation; this means that when the temperature increases, the moisture content decreases.

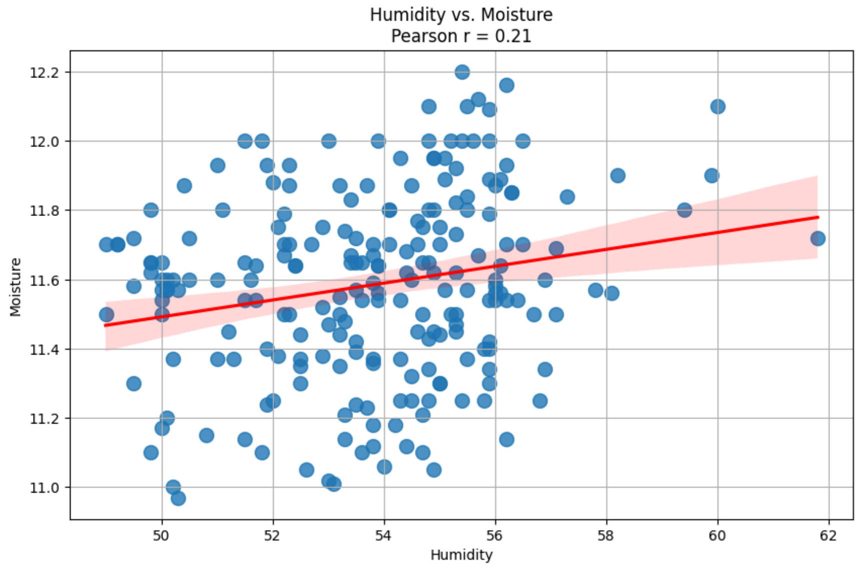


Figure 5-43. Humidity vs. Moisture Pearson Correlation Coefficient

Figure 5-43 shows the relationship between humidity and moisture data after the three-day testing. A Pearson correlation coefficient 0.20 indicates a “weak positive” correlation between the two variables. This suggests that as one variable slightly increases, humidity and the other variable, moisture, tend to increase slightly.

Given the context, a Pearson correlation of -0.21 and 0.20 between the Temperature-Moisture Relationship and Humidity-Moisture Relationship, respectively, are acceptable for this project. Since the requirement is that moisture should be between 11% - 12.5% when the temperature is between 20 - 25.4 and when the humidity is between 50% - 70%, the weak negative correlation indicates some inverse relationship between these variables. Still, it's not strong enough to enforce a strict linear relationship. This means that while there might be a tendency for moisture to decrease slightly as temperature increases, the correlation is not strong enough to guarantee that moisture will always be at 12.5 when the temperature reaches 25.4 or at 11 when the temperature is at 20.

Project Objective #2: *Design a storage device that maintains coffee beans' moisture content from 11% to 12.5%.*

The second objective examines whether the device effectively maintains the moisture content of the coffee beans. To assess the efficacy of the final design, the developers measure the moisture content of the green Arabica coffee beans throughout their storage period. Measurements are taken at hourly intervals. The results of this testing are presented below:

Table 5-9: Moisture Content Reading Result/Remarks

Moisture Content (%)	Reading Result/Remarks
10% - 10.9%	Acceptable, Not Suggested
11% - 12.5%	Acceptable, Normal
12.6 % - 13.9%	Acceptable, Not Suggested
14.0% - up	Not acceptable. Suggested for drying

Tables 5-9 above provide the remarks used as references for this objective's testing results. Based on the client's current scenario and a related review of studies, a moisture content of 11% - 12.5% for coffee beans during storage is considered acceptable and normal. However, if the moisture content rises to 12.6% - 13.9%, it is still acceptable but not recommended. Conversely, if the moisture content reaches 14% or higher, it is no longer sufficient, and the beans should be dried to reduce the moisture content to 11% - 12.5%. Lastly, if the moisture content falls below 10% - 10.9%, it is also acceptable but not recommended for roasting.

Table 5-10: Project Objective #2 Testing Results

Test	Time (hr)	Moisture Content	Reading Result/Remarks
1	7:40:20	11.58	Acceptable, Normal
2	7:50:21	12.09	Acceptable, Normal
3	8:00:23	11.21	Acceptable, Normal
4	8:10:24	11.92	Acceptable, Normal
5	8:20:26	11.7	Acceptable, Normal
6	8:30:28	11.39	Acceptable, Normal
7	8:40:29	11.21	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
8	8:50:31	11.52	Acceptable, Normal
9	9:00:33	11.01	Acceptable, Normal
10	9:10:34	11.24	Acceptable, Normal
11	9:20:36	11.25	Acceptable, Normal
12	9:30:38	11.06	Acceptable, Normal
13	9:40:40	11.02	Acceptable, Normal
14	9:50:42	11.25	Acceptable, Normal
15	10:00:43	11.67	Acceptable, Normal
16	10:10:45	11.65	Acceptable, Normal
17	10:20:47	11.43	Acceptable, Normal
18	10:30:49	11.65	Acceptable, Normal
19	10:40:50	11.14	Acceptable, Normal
20	10:50:52	11.35	Acceptable, Normal
21	11:00:54	11.65	Acceptable, Normal
22	11:10:56	11.1	Acceptable, Normal
23	11:20:58	11.36	Acceptable, Normal
24	11:31:00	11.5	Acceptable, Normal
25	11:41:02	11.37	Acceptable, Normal
26	11:51:03	11.05	Acceptable, Normal
27	12:01:05	11.37	Acceptable, Normal
28	12:11:07	11.12	Acceptable, Normal
29	12:21:08	11.24	Acceptable, Normal
30	12:31:10	11.5	Acceptable, Normal
31	12:41:12	11.1	Acceptable, Normal
32	12:51:14	11.62	Acceptable, Normal
33	13:01:15	11.62	Acceptable, Normal
34	13:11:17	11.3	Acceptable, Normal
35	13:21:19	11.6	Acceptable, Normal
36	13:31:20	11.14	Acceptable, Normal
37	13:41:22	11.4	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
38	13:51:24	11.05	Acceptable, Normal
39	14:01:25	11.1	Acceptable, Normal
40	14:11:27	11.37	Acceptable, Normal
41	14:21:29	11.6	Acceptable, Normal
42	14:31:30	11.58	Acceptable, Normal
43	14:41:32	11.1	Acceptable, Normal
44	14:51:34	11.62	Acceptable, Normal
45	15:01:36	11.8	Acceptable, Normal
46	15:11:37	11.37	Acceptable, Normal
47	15:21:39	11.7	Acceptable, Normal
48	15:31:41	11.5	Acceptable, Normal
49	15:41:42	11.7	Acceptable, Normal
50	15:51:44	11.65	Acceptable, Normal
51	16:01:46	11.65	Acceptable, Normal
52	16:11:48	11.7	Acceptable, Normal
53	16:21:49	11.57	Acceptable, Normal
54	16:31:51	11.57	Acceptable, Normal
55	16:41:53	11.57	Acceptable, Normal
56	16:51:54	11.57	Acceptable, Normal
57	17:01:56	10.97	Acceptable, Not Suggested
58	17:11:58	11.6	Acceptable, Normal
59	17:22:00	11.6	Acceptable, Normal
60	17:32:01	11.17	Acceptable, Normal
61	17:42:03	11.6	Acceptable, Normal
62	17:52:05	11	Acceptable, Normal
63	18:02:07	11.65	Acceptable, Normal
64	18:12:08	11.5	Acceptable, Normal
65	18:22:10	11.72	Acceptable, Normal
66	18:32:12	11.72	Acceptable, Normal
67	18:42:13	11.87	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
68	18:52:15	11.3	Acceptable, Normal
69	19:02:17	11.54	Acceptable, Normal
70	19:12:18	11.2	Acceptable, Normal
71	19:22:20	11.37	Acceptable, Normal
72	19:32:22	11.15	Acceptable, Normal
73	19:42:24	11.6	Acceptable, Normal
74	7:40:20	11.54	Acceptable, Normal
75	7:50:21	11.82	Acceptable, Normal
76	8:00:23	12.1	Acceptable, Normal
77	8:10:25	11.18	Acceptable, Normal
78	8:20:26	11.12	Acceptable, Normal
79	8:30:28	11.68	Acceptable, Normal
80	8:40:29	11.8	Acceptable, Normal
81	8:50:31	11.8	Acceptable, Normal
82	9:00:33	11.62	Acceptable, Normal
83	9:10:35	11.95	Acceptable, Normal
84	9:20:36	11.45	Acceptable, Normal
85	9:30:38	11.45	Acceptable, Normal
86	9:40:40	11.75	Acceptable, Normal
87	9:50:42	11.95	Acceptable, Normal
88	10:00:43	11.3	Acceptable, Normal
89	10:10:45	11.75	Acceptable, Normal
90	10:20:47	12	Acceptable, Normal
91	10:30:48	11.85	Acceptable, Normal
92	10:40:50	11.7	Acceptable, Normal
93	10:50:52	11.45	Acceptable, Normal
94	11:00:54	11.5	Acceptable, Normal
95	11:10:55	12	Acceptable, Normal
96	11:20:57	11.8	Acceptable, Normal
97	11:30:59	11.85	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
98	11:41:00	12	Acceptable, Normal
99	11:51:02	12.16	Acceptable, Normal
100	12:01:04	11.5	Acceptable, Normal
101	12:11:06	11.14	Acceptable, Normal
102	12:21:07	11.56	Acceptable, Normal
103	12:31:09	11.25	Acceptable, Normal
104	12:41:11	11.25	Acceptable, Normal
105	12:51:12	11.25	Acceptable, Normal
106	13:01:14	11.3	Acceptable, Normal
107	13:11:16	11.79	Acceptable, Normal
108	13:21:17	11.54	Acceptable, Normal
109	13:31:19	11.54	Acceptable, Normal
110	13:41:21	11.89	Acceptable, Normal
111	13:51:23	11.89	Acceptable, Normal
112	14:01:24	11.89	Acceptable, Normal
113	14:11:26	11.95	Acceptable, Normal
114	14:21:28	11.3	Acceptable, Normal
115	14:31:33	11.57	Acceptable, Normal
116	14:41:35	11.5	Acceptable, Normal
117	14:51:37	11.57	Acceptable, Normal
118	15:01:38	11.75	Acceptable, Normal
119	15:11:40	11.55	Acceptable, Normal
120	15:21:42	12	Acceptable, Normal
121	15:31:43	11.87	Acceptable, Normal
122	15:41:45	11.65	Acceptable, Normal
123	15:51:47	11.44	Acceptable, Normal
124	16:01:48	12.1	Acceptable, Normal
125	16:11:50	11.48	Acceptable, Normal
126	16:21:52	11.67	Acceptable, Normal
127	16:31:54	11.59	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
128	16:41:55	11.64	Acceptable, Normal
129	16:51:57	11.23	Acceptable, Normal
130	17:01:59	11.54	Acceptable, Normal
131	17:12:00	11.54	Acceptable, Normal
132	17:22:02	11.18	Acceptable, Normal
133	17:32:04	11.38	Acceptable, Normal
134	17:42:06	11.38	Acceptable, Normal
135	17:52:08	11.44	Acceptable, Normal
136	18:02:09	11.83	Acceptable, Normal
137	18:12:11	11.67	Acceptable, Normal
138	18:22:13	11.88	Acceptable, Normal
139	18:32:15	11.35	Acceptable, Normal
140	18:42:17	11.35	Acceptable, Normal
141	18:52:19	11.47	Acceptable, Normal
142	19:02:21	11.37	Acceptable, Normal
143	19:12:23	11.64	Acceptable, Normal
144	19:22:25	11.5	Acceptable, Normal
145	19:32:26	11.5	Acceptable, Normal
146	19:42:28	11.79	Acceptable, Normal
147	7:40:11	12	Acceptable, Normal
148	7:50:12	12	Acceptable, Normal
149	8:00:14	12.2	Acceptable, Normal
150	8:10:16	11.8	Acceptable, Normal
151	8:20:17	11.8	Acceptable, Normal
152	8:30:19	11.56	Acceptable, Normal
153	8:40:21	11.34	Acceptable, Normal
154	8:50:22	11.6	Acceptable, Normal
155	9:00:25	11.56	Acceptable, Normal
156	9:10:26	11.5	Acceptable, Normal
157	9:20:28	11.87	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
158	9:30:30	11.8	Acceptable, Normal
159	9:40:32	11.44	Acceptable, Normal
160	9:50:34	12.12	Acceptable, Normal
161	10:00:35	11.54	Acceptable, Normal
162	10:10:37	11.84	Acceptable, Normal
163	10:20:39	11.4	Acceptable, Normal
164	10:30:40	11.32	Acceptable, Normal
165	10:40:42	11.6	Acceptable, Normal
166	10:50:44	11.57	Acceptable, Normal
167	11:00:46	11.34	Acceptable, Normal
168	11:10:47	11.37	Acceptable, Normal
169	11:20:49	11.67	Acceptable, Normal
170	11:30:51	11.25	Acceptable, Normal
171	11:40:52	11.72	Acceptable, Normal
172	11:50:54	11.56	Acceptable, Normal
173	12:00:56	11.54	Acceptable, Normal
174	12:10:58	11.6	Acceptable, Normal
175	12:20:59	11.34	Acceptable, Normal
176	12:31:01	11.47	Acceptable, Normal
177	12:41:03	11.7	Acceptable, Normal
178	12:51:05	11.87	Acceptable, Normal
179	13:01:06	11.77	Acceptable, Normal
180	13:11:08	11.69	Acceptable, Normal
181	13:21:10	11.7	Acceptable, Normal
182	13:31:11	11.42	Acceptable, Normal
183	13:41:13	11.4	Acceptable, Normal
184	13:51:15	11.25	Acceptable, Normal
185	14:01:16	11.7	Acceptable, Normal
186	14:11:18	11.65	Acceptable, Normal
187	14:21:20	11.73	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
188	14:31:21	11.57	Acceptable, Normal
189	14:41:23	11.9	Acceptable, Normal
190	14:51:25	11.64	Acceptable, Normal
191	15:01:26	11.64	Acceptable, Normal
192	15:11:28	11.7	Acceptable, Normal
193	15:21:30	11.75	Acceptable, Normal
194	15:31:32	11.7	Acceptable, Normal
195	15:41:33	11.87	Acceptable, Normal
196	15:51:36	11.93	Acceptable, Normal
197	16:01:37	11.93	Acceptable, Normal
198	16:11:46	11.42	Acceptable, Normal
199	16:21:49	11.8	Acceptable, Normal
200	16:31:51	12.1	Acceptable, Normal
201	16:41:52	12	Acceptable, Normal
202	16:51:54	11.74	Acceptable, Normal
203	17:01:56	11.93	Acceptable, Normal
204	17:11:58	11.93	Acceptable, Normal
205	17:21:59	11.54	Acceptable, Normal
206	17:32:01	11.65	Acceptable, Normal
207	17:42:03	11.54	Acceptable, Normal
208	17:52:04	11.45	Acceptable, Normal
209	18:02:06	12	Acceptable, Normal
210	18:12:08	11.64	Acceptable, Normal
211	18:22:09	12	Acceptable, Normal
212	18:32:11	11.7	Acceptable, Normal
213	18:42:13	11.64	Acceptable, Normal
214	18:52:14	11.87	Acceptable, Normal
215	19:02:16	11.84	Acceptable, Normal
216	19:12:18	11.72	Acceptable, Normal
217	19:22:19	11.9	Acceptable, Normal

Test	Time (hr)	Moisture Content	Reading Result/Remarks
218	19:32:21	11.95	Acceptable, Normal
219	19:42:23	12	Acceptable, Normal

Table 5-10 shows the summary results of the conducted testing. It reveals that all 219 tests had a reading result of Acceptable. However, 1 out of 219 (0.46%) testings were noted as Not Suggested, leaving 99.54% of data indicated as "Acceptable, Normal."

Based on the results, the developers proved that the materials used for the final design, which include Acrylic Plastic and Polyolefin Foam, effectively maintained the moisture content of the green coffee beans.

Project Objective #3: *Develop an automated fan control system that activates the fan based on a predefined threshold.*

In this objective, the developers assess whether the Raspberry Pi and the fan are connected by checking if every time the microprocessor has a new sensor reading, it'll be followed by the corresponding command to the fan and how long before it reacts. With this, the developers conducted 30 tests and counter-checked whether the fan was turned on or off from the Raspberry Pi terminal and how long it took to activate. Below are the test results:

Table 5-11: Project Objective #3 Testing Results

Test #	Temperature Reading	Threshold Value	Fan Status (On/Off)	Expected Outcome (On/Off)	Actual Outcome (On/Off)	Response Time (µs)	Remarks
1	19.1°C	19.9°C	ON	On	On	200	It matches the expected outcome.
2	20.0°C	19.9°C	OFF	Off	Off	100	It matches the expected outcome.
3	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
4	20.0°C	19.9°C	OFF	Off	Off	100	It matches the expected outcome.
5	19.6°C	19.9°C	ON	On	On	200	It matches the expected outcome.
6	20.2°C	19.9°C	OFF	Off	Off	200	It matches the expected outcome.
7	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
8	20.0°C	19.9°C	OFF	Off	Off	200	It matches the expected outcome.
9	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
10	20.2°C	19.9°C	OFF	Off	Off	100	It matches the expected outcome.
11	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
12	20.0°C	19.9°C	OFF	Off	Off	300	It matches the expected outcome.
13	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
14	20.1°C	19.9°C	OFF	Off	Off	200	It matches the expected outcome.
15	19.9°C	19.9°C	ON	On	On	300	It matches the expected outcome.
16	20.1°C	19.9°C	OFF	Off	Off	200	It matches the expected outcome.
17	19.9°C	19.9°C	ON	On	On	300	It matches the expected outcome.
18	20.1°C	19.9°C	OFF	Off	Off	200	It matches the expected outcome.
19	19.9°C	19.9°C	ON	On	On	100	It matches the expected outcome.

Test #	Temperature Reading	Threshold Value	Fan Status (On/Off)	Expected Outcome (On/Off)	Actual Outcome (On/Off)	Response Time (µs)	Remarks
20	20.1°C	19.9°C	OFF	Off	Off	100	It matches the expected outcome.
21	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
22	20.1°C	19.9°C	OFF	Off	Off	100	It matches the expected outcome.
23	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
24	20.1°C	19.9°C	OFF	Off	Off	100	It matches the expected outcome.
25	19.9°C	19.9°C	ON	On	On	300	It matches the expected outcome.
26	20.1°C	19.9°C	OFF	Off	Off	300	It matches the expected outcome.
27	19.9°C	19.9°C	ON	On	On	300	It matches the expected outcome.
28	20.1°C	19.9°C	OFF	Off	Off	200	It matches the expected outcome.
29	19.9°C	19.9°C	ON	On	On	200	It matches the expected outcome.
30	20.1°C	19.9°C	OFF	Off	Off	200	It matches the expected outcome.

Table 5-11 presents a series of tests designed to check how well the system controls the fan based on temperature readings. The system used a temperature threshold of 19.9°C, or below 20°C, to decide whether the fan should be turned ON or OFF. If the temperature is below this threshold, the fan should switch to on; otherwise, it should be off. The results from all 30 tests show that the system's decisions were correct 100% of the time, and the fan's actual status matched what was expected; the average response time of the fan to activate was 196.67μs.

Project Objective #4: *Develop a device that maintains an average power consumption below 1000W per hour.*

The developers conducted three experiments to validate how much the device consumes in an hour. As mentioned, the first experiment had an interval of 5 minutes per check of the current power consumption in wattage, the second experiment had an interval of 15 minutes, and the third experiment had an interval of 30 minutes. All experiments included conditions where the device was in an idle state (with only the sensors working) and an active state (with the fan, sensors, and microcontroller all working).

Table 5-12. Project Objective #4 Experimental Results - Interval every 10 minutes

Test #	Time	Measurement in Wattage (W)	State	Interval (mins.)
1	3:47 PM	10.17	Idle (at use) state	10
2	3:57 PM	9.27	Idle (at use) state	10
3	4:07 PM	9.25	Idle (at use) state	10
4	4:17 PM	9.04	Idle (at use) state	10
5	4:27 PM	160.97	Active state	10
6	4:37 PM	8.52	Idle (at use) state	10

The results from Project Objective #4, measured every 5 minutes, as depicted in Table 5-10, show apparent differences in power consumption between the device's idle (at use) and active states. When the device was idle, power consumption gradually decreased from 10.17 watts at 3:47 PM to 9.04 watts by 4:17 PM. This shows a steady drop in power use over time. However, when the device was active, there was a significant increase in power consumption, reaching 160.97 watts at 4:27 PM. This spike highlights the much higher power needed when the device is fully operational. After the active state, power consumption dropped to 8.52 watts by 4:37 PM. It has a total average power consumption of 29.60 W or 0.0296 kWh.

Table 5-13: Project Objective #4 Experimental Results - Interval every 15 minutes

Test #	Time	Measurement in Wattage (W)	State	Interval (mins.)
1	5:00 PM	10.04	Idle (at use) state	15
2	5:15 PM	9.72	Idle (at use) state	15
3	5:30 PM	9.9	Idle (at use) state	15
4	5:45 PM	163.26	Active state	15
5	6:00 PM	9.68	Idle (at use) state	15

Table 5-13 presents the results at 15-minute intervals, revealing the device's power consumption at an hour. In the idle state, power consumption ranged from 9.68 to 10.04 watts throughout the test, showing relatively stable power usage. The measurement at 5:00 PM was 10.04 watts, which decreased slightly to 9.68 watts by 6:00 PM. However, a significant increase in power consumption was observed during the active state, with a measurement of 163.26 watts at 5:45 PM. This sharp rise highlights the additional power required when the device is actively running. Following the active state, power consumption returned to idle levels, demonstrating the impact of device activity on overall energy usage. This result achieved a total average power consumption of 40.52 W or 0.0405 kWh.

Table 5-14: Project Objective #4 Experimental Results - Interval every 30 minutes

Test #	Time	Measurement in Wattage (W)	State	Interval (mins.)
1	6:00 PM	9.68	Idle (at use) state	30
2	6:30 PM	9.93	Idle (at use) state	30
3	7:30 PM	9.69	Idle (at use) state	30

The results from the final test, conducted at 30-minute intervals, show consistent power consumption readings. At 6:00 PM, the device consumed 9.68 watts. By 6:30 PM, the consumption increased slightly to 9.93 watts and then decreased to 9.69 watts by 7:30 PM. The total power consumption over the test period was 9.7667 watts, equivalent to 0.009767 kWh. These measurements indicate that power consumption remains relatively stable over extended periods, demonstrating consistent energy usage patterns.

5.9 Conclusion

The dotBean project aims to maintain the moisture content of the green coffee arabica beans by monitoring the temperature and humidity of the storage and the coffee beans' moisture content. This project sought to address the issue of the client from La Trinidad, Benguet; low temperatures resulting in high humidity leads to moisture absorption that causes high moisture content, leading to losses to the farmers due to possible mold growth, loss of aroma, and low profitability rate in the market. With this, the developers implement an automated storage system that offers real-time monitoring through its mobile application, which intends to provide prompt and easy tracking.

To achieve this possibility, the developers, concerning the client's requirements, sought to complete the following constraints and their criteria: performance, to reach optimal condition when the temperature drops within 30 minutes at 17 minutes and 43 seconds, response time, forwarding the command to the fan in just 202.04 microseconds, achieving the criteria less than 1 second, and lastly, power consumption, the device should at least consumed 1 kWh while still reliable and not compromising its functionality, and the device has a total power consumption of 0.1853 kWh.

Consequently, the system was able to achieve all these three criteria along with the following objectives: to determine the correct temperature and humidity levels by utilizing a temperature and humidity sensor to closely monitor the coffee beans moisture content inside a closed storage device, where the temperature and humidity sensor achieved a standard deviation of 1.13 during the final testing while the moisture sensor achieved a standard deviation of 0.27; to design a storage device that's capable of maintaining coffee beans' moisture content from 11% to 12.5%. The device maintained the moisture content at an acceptable range of 99.45% (218 out of 219 testing). Although 0.46% was still in the acceptable range, it was not suggested as the moisture content of the coffee beans for roasting to develop an automated fan control system based on a predefined threshold. The system achieved 100% of the testing where the fan followed the correct instructions given by the Raspberry Pi at an average response time of 196.67μs, and lastly, to develop a device that consumes less than 1000W per hour. Based on the experiments, it satisfies the condition where it maintains an average power consumption well below the 1000-watt per hour threshold across various testing intervals—5, 15, and 30 minutes—power consumption remained consistent with average readings of 29.6W, 40.52W, and 9.77W, respectively.

The above-mentioned specific objectives align with the primary purpose of the project: to develop an automated temperature-controlled storage system for moisture content monitoring for arabica coffee beans, assuring that it can still carry the usual capacity that the client stores while at the same time being carried or dragged at ease. In conclusion, the developers successfully maintained the green coffee arabica beans' storage conditions and moisture content. Furthermore, the client also found the system impressive regarding the implementation and purpose of the dotBean project.

Overall, dotBean successfully achieved the constraints, criteria, and project objectives, along with positive client feedback that could be improved further in future research endeavors.

5.10 Recommendations

From the conducted testing and validation of each constraint and attainment of specified objectives derived from the client’s problem and requirements, the dotBean can be further improved through:

- Customizing a moisture measuring device to enhance the accuracy of determining the moisture content of stored coffee beans, rather than relying on a general-purpose sensor designed for soil, which requires calibration.
- Allowing the storage device to operate without being tethered to an electric power supply/outlet, enabling greater portability.
- It can be customized according to the client's preferences, for example, where they manually operate the temperature the fan produces and set a specific time to open the fan or turn it off.
- Make the mobile application compatible with iOS users, preferably supporting iOS 15 or later versions alongside Android 12 and API level 33.

- Remove the mobile device requirement and integrate an LCD directly within the device.
- Further improvement could be achieved by exploring opportunities with other clients in warmer regions, where the system could be repurposed as a cooling mechanism in addition to its current heating function.
- Design a device that can be used anywhere, even in high-temperature environments.

5.11 Impact of the Engineering Solution

5.11.1 Public Safety and Health

The dotBean effectively manages moisture levels and storage conditions, preventing mold growth. This ensures that the coffee beans remain safe for consumption and helps maintain their quality, reducing the risks of off-flavors and health hazards linked to mold.

By minimizing moisture absorption-related issues, the project supports better food safety and hygiene, which is crucial for consumers and coffee producers. As coffee beans are a high-value commodity, their cost depends on their quality. By effectively managing moisture levels and storage conditions, the dotBean project helps preserve the moisture content of the coffee beans—one indicator of coffee beans' quality. This contributes to reduced risks of off-flavors and health hazards associated with mold. This proactive approach significantly reduces the risk of health hazards related to mold exposure, ultimately contributing to public health and safety by ensuring that coffee beans are free from contaminants and fit for consumption. Additionally, it supports coffee producers by preventing potential financial losses from spoilage and maintaining the high standards required in the coffee industry.

5.11.2 Societal

The dotBean project addresses challenges faced by small-scale coffee farmers in Benguet Province by monitoring and controlling green coffee beans' storage conditions, helping maintain their moisture content. This improvement could enhance the coffee's taste and marketability, potentially providing local farmers better market opportunities.

Including a mobile application for real-time monitoring ensures that the technology is accessible and user-friendly. This aspect encourages coffee farmers, who may need more experience with advanced monitoring systems, to adopt new technology. By focusing on ease of use, the dotBean project aims to enhance digital literacy and empower local communities to integrate technology into their traditional practices. Monitoring moisture content, storage temperature, and humidity in real-time also gives farmers greater assurance in the quality of their stored beans, contributing to stability in their operations. By addressing these critical aspects, the dotBean project supports the sustainability and resilience of coffee-producing communities.

5.11.3 Economical

Economically, the dotBean system is designed to help reduce losses caused by inadequate storage conditions. Maintaining the optimal temperature and humidity levels reduces the risk of moisture absorption, which can lead to mold growth, which can be financially burdensome for small-scale coffee producers. This project demonstrates the potential for cost savings by minimizing waste and improving the consistency of coffee beans' quality.

Additionally, the system's low power consumption is an important consideration, as it keeps operational costs manageable, making the solution more attractive to producers who are conscious of energy expenses. Although the project is not yet ready for large-scale implementation, its development provides valuable insights into how technology can be integrated into traditional agricultural practices. This initial prototype could lead to further iterations that are more scalable and adaptable to different storage needs, offering a foundation for future improvements and potential economic benefits.

REFERENCES

11. Correlation and regression | The BMJ. (2020, October 28). The BMJ | the BMJ: Leading General Medical Journal. Research. Education. Comment. https://www.bmj.com/about-bmj/resources-readers/publications/statistics-square-one/11-correlation-and-regression

Ada, L. (2024). Arduino Tips, Tricks, and Techniques. https://learn.adafruit.com/arduino-tips-tricks-and-techniques

Ahmed, S., Brinkley, S., Smith, E. M., Sela, A., Theisen, M., Thibodeau, C., Warne, T., Anderson, E. P., Van Dusen, N., Giuliano, P., Ionescu, K. E., & Cash, S. B. (2021). Climate Change and Coffee Quality: Systematic Review on the Effects of Environmental and Management Variation on Secondary Metabolites and Sensory Attributes of Coffea arabica and Coffea canephora. *Frontiers in Plant Science*, *12*. https://doi.org/10.3389/fpls.2021.708013

Alibayan, Jan & Bobadilla, Ian & Carnicer, Mark & Pascua, Reynaldo & Teodósio, Gabriel & Arago, Nilo & Tolentino, Lean Karlo & Fernandez, Edmon & Valenzuela, Ira. (2019). Green Coffee Bean Sorter and Corrector based on Moisture Content using Capacitive Method. 1-4. 10.1109/HNICEM48295.2019.9073477.

alldatasheet.com. (n.d.). HS1101 Datasheet(PDF). Humirel. https://www.alldatasheet.com/datasheet-pdf/pdf/47866/HUMIREL/HS1101.html

Lady Ada (2024). *Arduino tips, tricks, and techniques*. Retrieved from: https://cdn-learn.adafruit.com/downloads/pdf/arduino-tips-tricks-and-techniques.pdf

Audrey Glenn L. Culliao & Jonathan M. Barcelo (2015) Fungal and mycotoxin contamination of coffee beans in Benguet province, Philippines, Food Additives & Contaminants: Part A, 32:2, 250-260, DOI: 10.1080/19440049.2014.1001796

Barea-Ramos, J. D., Cascos, G., Mesías, M., Lozano, J., & Martín-Vertedor, D. (2022). Evaluation of the olfactory quality of roasted coffee beans using a digital nose. Sensors, 22(22), 8654. https://doi.org/10.3390/s22228654

Bastiaans, J. a. H. P. (2001). *CA2454317A1 - Moisture barrier in foods - Google Patents*. https://patents.google.com/patent/CA2454317A1/en

Batani, R. S. (2020). *Growing arabica coffee in the mountain:* http://portal.bsu.edu.ph:8083/index.php/BRJ/article/view/266

Błaszkiewicz J, Nowakowska-Bogdan E, Barabosz K, Kulesza R, Dresler E, Woszczyński P, Biłos Ł, Matuszek DB, Szkutnik K. Effect of green and roasted coffee storage conditions on selected characteristic quality parameters. Sci Rep. 2023 Apr 20;13(1):6447. doi: 10.1038/s41598-023-33609-x. PMID: 37081041; PMCID: PMC10119118.

Bogdan, O., Zlobin, D., Murav'eva, O., Molin, S., & Platunov, A. (2021). Evaluation of nonuniformity of elastic properties of sheets made from closed-cell polyolefin foams by acoustic method. Devices and Methods of Measurements, 12(1), 58-66. https://doi.org/10.21122/2220-9506-2021-12-1-58-66

Borchert, S. (2022, July 6). *SOLIDWORKS flow tips and tricks*. Engineers Rule. https://www.engineersrule.com/solidworks-flow-tips-and-tricks/

Chen, H., Gu, Y. T., & Liew, S. (2020). Age-of-information dependent random access for massive iot networks. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). https://doi.org/10.1109/infocomwkshps50562.2020.9162973

Culliao, A. G. L., & Barcelo, J. M. (2015). Fungal and mycotoxin contamination of coffee beans in Benguet province, Philippines. *Food Additives & Contaminants. Part a. Chemistry, Analysis, Control, Exposure & Risk Assessment*, *32*(2), 250–260. https://doi.org/10.1080/19440049.2014.1001796

Cunningham, M. (2022, November 15). How to remove moisture from coffee beans. Brewing Coffees. https://brewingcoffees.com/remove-moisture-from-coffee-beans/

Da Rosa, S., Carvalho, A., McDonald, M., Von Pinho, E., Ap, S., & Veiga, A. (2011). The effect of storage conditions on coffee seed and seedling quality. Seed Science and Technology, 39(1), 151–164. https://doi.org/10.15258/sst.2011.39.1.13

Dejan. (2022, February 18). DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino. How to Mechatronics. https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

Department of Agriculture - Bureau of Agricultural Research, UPLB Foundation, Inc., & Philippine Council for Agriculture and Fisheries. (2022). *The Philippine Coffee Industry Roadmap (2021-2025)*. Retrieved July 27, 2024, from https://www.pcaf.da.gov.ph/wp-content/uploads/2022/06/Philippine-Coffee-Industry-Roadmap-2021-2025.pdf

Dey, A. K. (2023). What are Agitators? Selection and Types of Agitators (PDF). What Is Piping. https://whatispiping.com/agitators/#google_vignette

Edmallon. (2024, July 12). *Hacking a capacitive soil moisture sensor (V1.2) for frequency output*. Underwater Arduino Data Loggers. https://thecavepearlproject.org/2020/10/27/hacking-a-capacitive-soil-moisture-sensor-for-frequency-output/

Elert, G. (2023). *Electric Power - Summary*. The Physics Hypertextbook. https://physics.info/electric-power/summary.shtml

Farah, A. (Ed.). (2019). Coffee: Production, Quality and Chemistry. Royal Society of Chemistry. 267-268

Farahani, H., Wagiran, R., & Hamidon, M. N. (2014). Humidity Sensors Principle, Mechanism, and Fabrication Technologies: A Comprehensive Review. *Sensors*, *14*(5), 7881–7939. https://doi.org/10.3390/s140507881

Firmansyah, D. A., Negara, B. J., Maulana, Y. P., Firmansyah, V., & Juliastuti, E. (2020). Simple measurement instrument of moisture content for Indonesia coffee powder based on capacitive sensor. Journal of Physics, 1568(1), 012010. https://doi.org/10.1088/1742-6596/1568/1/012010

Flow. (2023). Autmix Blog | Industrial agitation's for the food industry. https://autmix.com/en/blog/agitation-for-food-industry

Foam Factory, Inc. & Author Foam Factory, Inc. (2024). *Polyethylene foam: Its uses, characteristics, and varieties*. Foam Factory, Inc. Blog. https://www.foambymail.com/blog/polyethylene-foam-its-uses-characteristics-and-varieties/

Fraden, J. (2016). Humidity and Moisture Sensors. In: Handbook of Modern Sensors. Springer, Cham. https://doi.org/10.1007/978-3-319-19303-8_14

Garcia, A. (2020). The Bittersweet Situation of the Local Coffee Industry. The Coffee Heritage Project. Retrieved from https://www.coffeeheritageproject.com/news/the-bittersweet-situation-of-the-local-coffee-industry

Ge, X., Han, Q., & Wang, Z. (2019). A threshold-parameter-dependent approach to designing distributed event-triggered h_{∞} consensus filters over sensor networks. IEEE Transactions on Cybernetics, 49(4), 1148-1159. https://doi.org/10.1109/tcyb.2017.2789296

Gilani, I. E., Sayadi, S., Zouari, N., & Al-Ghouti, M. A. (2023b). Plastic waste impact and biotechnology: Exploring polymer degradation, microbial role, and sustainable development implications. Bioresource Technology Reports, 24, 101606. https://doi.org/10.1016/j.biteb.2023.101606

Glkinst (2005). 4-Wire Pulse Width Modulation (PWM) Controlled Fans Specification. https://www.glkinst.com/cables/cable_pics/4_Wire_PWM_Spec.pdf

Goodmann, W. (2022). Food grade plastic: Which plastics are safe for food storage. Palmetto Industries. https://www.palmetto-industries.com/safe-food-grade-plastic/

Habaradas, R. B., & Mia, I. B. R. (2021). Bote Central: creating a chain of happiness for Philippine coffee farmers. *International Journal of Business and Society*, *22*(2), 941-959.

Haile, M., & Kang, W. (2019). The Harvest and Post-Harvest Management Practices' Impact on. Retrieved from https://www.intechopen.com/chapters/70151

Haixiao, H., Jiming, S., & Dengji, H. (2017). Airtightness detection device of coffee capsule and detection method. SciSpace - Paper. https://typeset.io/papers/airtightness-detection-device-of-coffee-capsule-and-2sttwbgijp

Harris, S., & Harris, D. (2022). From zero to one. In *Elsevier eBooks* (pp. 1–50). https://doi.org/10.1016/b978-0-12-820064-3.00001-5

He, D., Wang, H., Tian, Y., & Zimenko, K. (2021). Event-triggered discrete extended state observer–based model-free controller for quadrotor position and attitude trajectory tracking. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 236(4), 754-771. https://doi.org/10.1177/09596518211055362

He, X., Yin, J., Yang, J., Qiao, L., & Wu, S. (2019). Effect of dry-wet circulation on moisture absorption of autoclaved aerated concrete. Advances in Materials Science and Engineering, 2019, 1-12. https://doi.org/10.1155/2019/4165482

Hrisko, J. (2022, November 16). Capacitive Soil Moisture Sensor Calibration with Arduino — Maker Portal. Maker Portal. https://makersportal.com/blog/2020/5/26/capacitive-soil-moisture-calibration-with-arduino

Indeed Editorial Team (2022). How To Measure Accuracy and Precision in 5 Steps. Retrieved from: https://www.indeed.com/career-advice/career-development/how-to-measure-accuracy

ISO 8873-2:2007. (n.d.). ISO. https://www.iso.org/standard/39406.html

ISO/IEC 29341-26-12:2017. (n.d.). ISO. https://www.iso.org/standard/69172.html#:~:text=ISO%2FIEC%2029341%2D26%2D12%3A2017%20is%20compliant,a%20Telephony%20Client%20(TC).

Jameswilson. (2020, September 27). *LM2596 Buck Converter Datasheet, Pinout, Features, Applications*. The Engineering Projects. https://www.theengineeringprojects.com/2020/09/lm2596-buck-converter-datasheet-pinout-features-applications.html

Jennifer Madonna G. Dait, "*Factors Affecting the Economic Viability of the Coffee Industry in Ifugao*", International Journal of Science and Research (IJSR), Volume 5 Issue 10, October 2016, pp. 890-895, https://www.ijsr.net/getabstract.php?paperid=ART20162251

Joaquin, A. C., & Esteves, D. T. (1978, May 24). *TECHNICAL BULLETIN*. TECHNICAL BULLETIN. Retrieved October 2, 2023, from https://www.philmech.gov.ph/assets/publication/Technical%20Bulletin/2019/Technical%20Bulletin%2014.pdf

Johnson, R. (2023). Using Silica Gel for Food Storage: Is it Safe? The Prepper's Apprentice. https://preppersapprentice.com/is-silica-gel-food-safe/

Kenton, W. (2023, December 20). Sensitivity Analysis Definition. Investopedia. https://www.investopedia.com/terms/s/sensitivityanalysis.asp#:~:text=Sensitivityanalysisisafinancial,acertainrangeofvariables.

Keyence. (n.d.). Tolerance and Measurement Accuracy. Retrieved from https://www.keyence.com/ss/products/measure/library/basic/tolerance.jsp

King, K. (2020, October 27). *How to find Mean, variance, and standard deviation — Krista King Math | Online math help*. Krista King Math | Online Math Help. https://www.kristakingmath.com/blog/mean-variance-standard-deviation

Kverh, D. (2020, October 27). Humidity sensor what type should you choose? - ANDIVI. ANDIVI. https://www.andivi.com/humidity-sensor-what-type-should-you-choose/

L298N Motor Driver Module - Motor Drivers - Motor and Wheels. (n.d.). http://www.astanadigital.com/products/L298N-Motor-Driver-Module/1447

LaPoint-Tufts, L. (2021). Climate could ruin the flavor of your coffee. Futurity. https://www.futurity.org/coffee-climate-flavor-2647962-2/

Lauren. (2023, May 15). Drying Your Coffee: Methods for Removing Moisture from Beans. Liberty Beans Coffee Company. https://libertybeanscoffee.com/blogs/methods-for-removing-moisture-from-beans/

LearnMetrics. (2022, November 10). *Watts to kWh: Watts to Kilowatt-Hours Calculator + Chart*. LearnMetrics. https://learnmetrics.com/watts-to-kilowatt-hours-watts-to-kwh/

Lenong, C. and Chun, Y. J. (2024). Snr threshold scheduling for iot uplink network. IEEE Internet of Things Journal, 11(14), 25124-25135. https://doi.org/10.1109/jiot.2024.3391368

Li, I. (2021, July 1). *Five things you didn't know about Philippine coffee*. IndoChina Coffee. https://indochinacoffee.com/5-things-you-didnt-know-about-philippine-coffee/

LME Editorial Staff. (2022, November 25). *Interfacing DHT11 and DHT22 Sensors with Arduino*. Last Minute Engineers. https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/

MacDonnell, K. (2022). Coffee Shops in the Philippines: 12 Statistics to Know in 2023. Coffee Affection. https://coffeeaffection.com/coffee-shops-in-the-philippines-statistics/

Makerlab Electronics. (n.d.). *SIM800L V2 5V Wireless GSM GPRS Module*. https://www.makerlab-electronics.com/products/sim800l-v2-5v-wireless-gsm-gprs-module

Makerlab Electronics. (n.d.-a). *1 2 4 8 Channel 5V 12V 10A Relay Module with Optocoupler*. https://www.makerlab-electronics.com/products/1-2-4-8-channel-5v-12v-10a-relay-module-

Makerlab Electronics. (n.d.-b). *12V power supply 200W Philippines - Makerlab Electronics*. https://www.makerlab-electronics.com/products/12v-switching-power-supply-200w?_pos=1&_psq=12v+16.7a+s&_ss=e&_v=1.0?variant=42292814839999

Makerlab Electronics. (n.d.-c). *Official Raspberry Pi Power Supply 5.1V 3A USB C*. https://www.makerlab-electronics.com/products/official-raspberry-pi-power-supply-5-1v-3a-usb-c

Makerlab Electronics. (n.d.-d). *Thermoelectric TEC12706 TEC-12706 Peltier Cooling System Heatsink Kit*. https://www.makerlab-electronics.com/products/thermoelectric-tec12706-tec-12706-peltier-cooling-system-heatsink-kit-air-cooled-liquid-cooled?_pos=2&_sid=bc4e8c26f&_ss=r?variant=43321210405055

Makerlab Electronics. (n.d.-e). *Tower Pro Digital Robot Servo Motor (180 rotation) – MG996R MG996*. https://www.makerlab-electronics.com/products/tower-pro-digital-robot-servo-motor-180-rotation-mg996r-mg996

Meralco (2022). *2022 1H Average Rate for ALL Meralco Residential Customers Category: Ref and Freezers.* https://www.meralco.com.ph/sites/default/files/inline-files/0f5b366bc24705a8c0d1ac0d487ffcfd.pdf

Merchant, J. (2021, November 25). How to measure the moisture content of green coffee. MTPak Coffee. https://mtpak.coffee/2021/05/how-to-measure-moisture-content-green-coffee/

Micabalo, Ardessa Tiffany P., "A field practice report on processing of arabica coffee (Coffea arabica L.) at Coffee for Peace, Matina, Davao City, Philippines with focus on storage and roasting" (2015). Undergraduate Theses. 4540. https://www.ukdr.uplb.edu.ph/etd-undergrad/4540

Muhammed, B. L., Seid, M. H., & Habte, A. T. (2021). Determination of caffeine and hydrogen peroxide antioxidant activity of raw and roasted coffee beans Around Habru Woreda, Ethiopia using UV-VIS spectroscopy. Clinical Pharmacology, Volume 13, 101–113. https://doi.org/10.2147/cpaa.s311032

Murthy, M. N., & AjaySaiKiran, P. (2018). A smart office automation system using Raspberry Pi (Model-B). *2018 International Conference on Current Trends Towards Converging Technologies (ICCTCT)*. https://doi.org/10.1109/icctct.2018.8550894

Natalie, E. (2023). The impact of humidity on coffee bean storage - CoffeeReso. CoffeeReso. https://coffeereso.com/impact-humidity-coffee-bean-storage/

NTCDB. (2018). NATIONAL STRATEGIC AND CAPACITY DEVELOPMENT PLAN FOR COFFEE SUB-SECTOR DEVELOPMENT. Retrieved from https://www.teacoffee.gov.np/public/images/pdf-111546843047.pdf

Parallax. (n.d.). HS1101 Relative Humidity Sensor (#27920). mousser. https://www.mouser.com/datasheet/2/321/27920-HS1101-v1.0-217883.pdf

Parker, J. (2023, March 6). *Why is system architecture important? - Architecture*. Design Your World - Explore Architecture. https://www.architecturemaker.com/why-is-system-architecture-important/

Pazmiño, Jhonathan & Gallardo, Cecilia & González, Tzitziki & Winkler, Robert. (2022). Loss of Sensory Cup Quality: Physiological and Chemical Changes during Green Coffee Storage. Plant Foods for Human Nutrition. 77. 1-11. 10.1007/s11130-022-00953-8.

Pecina, E. (2023, December 16). Raspberry Pi 4 GPIO Pinout. Maker Pro. https://maker.pro/raspberry-pi/tutorial/raspberry-pi-4-gpio-pinout

Peltier module (Thermoelectric module) - Energy Conversion Devices - KYOCERA. (n.d.). KYOCERA GROUP GLOBAL SITE. https://global.kyocera.com/prdct/ecd/peltier/

Piltch, A. (2022, September 17). *Raspberry Pi 4: Review, buying guide and how to use.* Tom's Hardware. https://www.tomshardware.com/reviews/raspberry-pi-4

Pomelo. (2022, November 29). Humidity Sensor Types | Atlas Scientific. Atlas Scientific. https://atlas-scientific.com/blog/humidity-sensor-types/

Powering a radiator fan with 12V DC power supply. (n.d.). Electrical Engineering Stack Exchange. https://electronics.stackexchange.com/questions/421297/powering-a-radiator-fan-with-12v-dc-power-supply

ProtoSupplies.com. (2023). Thermoelectric Peltier TEC1-12706 Cooler Kit - ProtoSupplies. ProtoSupplies. https://protosupplies.com/product/thermoelectric-peltier-tec1-12706-cooler-kit/

PU cold chains help keep goods fresh and save energy. (n.d.). Covestro AG. https://solutions.covestro.com/en/highlights/articles/stories/2021/polyurethane-pu-cold-chain-applications

Quann, M. (2003). Air-tight coffee storage container. SciSpace - Paper. https://typeset.io/papers/air-tight-coffee-storage-container-5813f5xg2d

Ramteke, S. (n.d.). *Buy SIM800L V2.0 5V Wireless GSM GPRS MODULE Quad-Band with Antenna Online in India | Robocraze*. Robocraze. https://robocraze.com/products/sim800l-gsm-gprs-module

Raspberry Pi (Trading) Ltd., 2019. Raspberry Pi 4 Model B Datasheet. from https://www.farnell.com/datasheets/4015971.pdf

Red LED. (2021, January 24). Exploring Arduino. https://www.exploringarduino.com/parts/red-led/#:~:text=An%20average%205mm%20Red%20LED,an%20LED%20to%20your%20Arduino!

Rosa, S., Carvalho, A., Mcdonald, M., Pinho, É. V., & Silva, A. P. (2011). *The effect of storage conditions on coffee seed and seedling quality*. https://www.semanticscholar.org/paper/The-effect-of-storage-conditions-on-coffee-seed-and-Rosa-Carvalho/0d3abd7337166c7339d6e11d8208844450ef1838

Ryszard, Rywotycki. (2003). Electric sensor for prompt measurement of moisture content in solid food products. Journal of Food Process Engineering, 25(6):473-483. doi: 10.1111/J.1745-4530.2003.TB00646.X

Schirn, A. (2024, January 11). *ISO/IEC/IEEE 23026:2023—Systems and Software Engineering*. The ANSI Blog. https://blog.ansi.org/iso-iec-ieee-23026-2023-systems-and-software-engineering/#:~:text=What%20Is%20ISO%2FIEC%2FIEEE,for%20intranet%20and%20extranet%20environments.

Sensor Terminology - NI. (2023, March 30). National Instruments. Retrieved October 2, 2023, from https://www.ni.com/en/shop/data-acquisition/sensor-fundamentals/sensor-terminology.html

Servo Motor SG-90. (n.d.). Components101. https://components101.com/motors/servo-motor-basics-pinout-datasheet

Shrinithivihahshini, N. (2024). A study based on bis standard is9845:1998 for assessing the migration of bisphenol a from food contact plastics.. https://doi.org/10.21203/rs.3.rs-4363762/v1

Siddique, A., Bozorgi, M., Venkateshwar, K., Tasnim, S. H., & Mahmud, S. (2023). Phase change material-enhanced solid-state thermoelectric cooling technology for food refrigeration and storage applications. *Journal of Energy Storage*, *60*, 106569. https://doi.org/10.1016/j.est.2022.106569

SIM800L GPRS moduledule. (n.d.). Future Electronics Egypt. https://store.fut-electronics.com/products/sim800l-gprs-gsm-module#:~:text=It's%20operating%20voltage%20is%203.7,V%2C%20peak%20current%20is%201A.&text=Send%20and%20receive%20GPRS%20data,IP%2C%20HTTP%2C%20etc.)

Singh, R. A., Puluckul, P. P., Berkvens, R., & Weyn, M. (2020). Energy consumption analysis of LPWAN technologies and lifetime estimation for IoT application. Sensors, 20(17), 4794. https://doi.org/10.3390/s20174794

Śmietańska, J. (2024, January 18). Watt-hour calculator. Omni Calculator. https://www.omnicalculator.com/physics/watt-hours

Smoot, B. R. (2022, February 11). *Push Button Switches 101 | CUI Devices*. CUI Devices. https://www.cuidevices.com/blog/push-button-switches-101

Soil moisture hygrometer detection sensor module for Arduino. (n.d.). https://www.icstation.com/mobile/soil-moisture-hygrometer-detection-humidity-sensor-module-corrosion-resistance-probe-arduino-p-12322.html

Soil Moisture Sensor Detector Module and Humidity Test Sensor with Cable. (2024). RoboticsDNA. https://roboticsdna.in/product/soil-moisture-sensor-detector-module-and-humidity-test-sensor-with-cable/

Soil moisture sensor module. (2020, October 25). *Hatchnhack Cart*. https://www.hnhcart.com/blogs/sensors-modules/soil-moisture-sensor-module#:~:text=Input%20Voltage%3A%203.3%E2%80%935V,Adjustable%20trigger%20level%20from%20preset

Sucafina specialty: Consistency in measuring moisture content. (n.d.). https://sucafina.com/apac/news/consistency-in-measuring-moisture-content

Stell, J. (2023, March 27). Do coffee beans absorb smell? A comprehensive guide. *Coffee Pursuing*. https://coffeepursuing.com/do-coffee-beans-absorb-smell/

System Architecture - Detailed explanation. (2023, June 22). InterviewBit. https://www.interviewbit.com/blog/system-architecture/

Tampon, V. (2023, July 29). Coffee statistics you need to know for 2023 and beyond. https://hillsandvalleys.ph/coffee-statistics-philippines/

Tark, S. (2020, July 20). How to protect coffee quality during sea Transit. Perfect Daily Grind. https://perfectdailygrind.com/2020/04/how-to-protect-coffee-quality-during-sea-transit/

Technology, S. (n.d.). Polyurethane in the food industry: efficiency and safety. https://blog.synthesia.com/en/polyurethane-in-the-food-industry-efficiency-and-safety

Thermoelectric Peltier Refrigeration Cooling System Kit - RAM Electronics. (2020). RAM Electronics. https://ram-e-shop.com/product/hs10/

The Correlation Coefficient (r). (n.d.-b). https://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717-Module9-Correlation-Regression/PH717-Module9-Correlation-Regression4.html

The Importance of Humidity Control in Coffee Storage Facilities. (n.d.). Polygon Group. Retrieved from https://www.polygongroup.com/en-US/blog/the-importance-of-humidity-control-in-coffee-storage-facilities/

Use Case Diagram Tutorial (Guide with Examples) | Creately. (2022, December 2). https://creately.com/guides/use-case-diagram-tutorial/

Walker, M. (2023, March 15). Acrylic vs glass food storage. Build a Stash. https://www.buildastash.com/post/acrylic-vs-glass-food-storage

Wang, B., Ruud, C., Price, J., Kim, H., & Giebink, N. (2019). Graded-index fluoropolymer antireflection coatings for invisible plastic optics. Nano Letters, 19(2), 787-792. https://doi.org/10.1021/acs.nanolett.8b03886

Wati*, E. K., Hidayanti, F., & Prasetya, A. (2020). Design of Temperature and Humidity Control on Arabica Coffee Storage. In International Journal of Innovative Technology and Exploring Engineering (Vol. 9, Issue 4, pp. 1076–1080). Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication - BEIESP. https://doi.org/10.35940/ijitee.d1542.029420

Waynne, & Waynne. (2023, August 26). Effects of Humidity on stored coffee. Coffee Revolution. https://www.coffeerevolution.net/humidity-on-coffee/#:~:text=Here%E2%80%99s%20how%20humidity%20affects%20the%20flavor%20of%20stored,coffee%2C%20humidity%20can%20affect%20the%20extraction%20process.%20

WebstaurantStore. (n.d.). What is silica gel? https://www.webstaurantstore.com/blog/4004/what-is-silica-gel.html#:~:text=Is%20Silica%20Gel%20Safe%20for,drawers%20to%20slow%20down%20sprouting

What is a Block Diagram and How to Create One | Miro. (n.d.). https://miro.com/. https://miro.com/diagramming/what-is-a-block-diagram/

What is Burst time, Arrival time, Exit time, Response time, Waiting time, Turnaround time, and Throughput? (n.d.).

https://afteracademy.com/blog/what-is-burst-arrival-exit-response-waiting-turnaround-time-and-throughput/

Wiener-Bronner (2022). Climate Change is Coming for Our Coffee. CNN Philippines. Retrieved from http://www.cnnphilippines.com/business/2022/1/27/Climate-change-coffee.html

Yazawa, M. (2023). *How Long Ago You Bought Your Coffee Beans Isn't a Key Determinant of How Fresh They Are—Here's How To Tell if Your Beans Have Gone Stale*. Well+Good. Retrieved Retrieved from https://www.wellandgood.com/stale-coffee/

Yin, Jianhui., Cheng, Shengli., Wu, Guojian. (2015). Resistive grain moisture measurement system.

Yongsheng, Xie., Hong, Ding., Fendong, Huang. (2019). Design of Temperature and Humidity Measuring System Based on Raspberry Pie. 610-615. doi: 10.1145/3349341.3349477

Yoong, K. (2022). How much does light affect roasted coffee? MTPak Coffee. https://www.mtpak.coffee/2022/05/how-much-does-light-affect-roasted-coffee/

Yu, J., Chu, M., Park, H., Park, J., & Lee, K. (2021). Analysis of volatile compounds in coffee prepared by various brewing and roasting methods. Foods, 10(6), 1347. https://doi.org/10.3390/foods10061347

Yu, L., Gao, W., Shamshiri, R. R., Tao, S., Ren, Y., Zhang, Y., & Su, G. (2021). Review of research progress on soil moisture sensor technology. Int J Agric & Biol Eng, 14(4), 32–42. https://doi.org/10.25165/j.ijabe.20211404.6404

APPENDICES

Appendix A:
Acceptance Sheet

ACCEPTANCE SHEET

This design project entitled "**dotBean: A Design of an Automated Temperature-Controlled Storage with Threshold-based Algorithm and Real-time Moisture Monitoring via Mobile App for Green Coffee Beans in La Trinidad, Benguet**" designed and submitted by Rome Angelo A. Gagabi, Reinier D. Mariscotes, Guillen Minerva A. Root, Carlo R. Rotoni, and Raven D. Agcaoili, in partial fulfillment of the requirements for the course CPE413 - CPE Design Project 2 is hereby accepted.

Dr. Jennalyn N. Mindoro
Design Project Adviser

Dr. Jennalyn N. Mindoro
Course Adviser

Dr. Jennifer B. Enriquez
Design Project Lead Panel

Engr. Marte Nipas
Design Project Panel

Mr. Dennis Nava
Design Project Panel

Accepted by:

Dr. Jennalyn N. Mindoro
Assistant Program Chair, Computer Engineering Department

Dr. Jennifer B. Enriquez
Program Chair, Computer Engineering Department

Dr. Marianne L. Yumul, ASEAN Eng.
Dean, College of Architecture and Engineering

Appendix B:

Endorsement for Final Defense

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
Manila

COMPUTER ENGINEERING DEPARTMENT

CERTIFICATION OF READINESS FOR PROPOSAL PRESENTATION

This project proposal entitled, **dotBean: A Smart-storing Device for Green Coffee Bean Arabica's Moisture Content Monitoring and Storage Temperature Control** prepared and submitted by:

Root, Guillen Minerva A.

Gagabi, Rome Angelo A.

Mariscotes, Reinier D.

Rotoni, Carlo R.

Agcaoili, Raven D.

In partial fulfillment for the course CPE 413 Design Project 2 has been reviewed, examined and is recommended for Proposal Presentation.

Attached is the requirements checklist for proposal presentation.

Engr. Yuneza Claire Mortos
Project Adviser

Dr. Jennalyn N. Mindoro
Class Adviser

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
Manila

COMPUTER ENGINEERING DEPARTMENT

ASSESSMENT AND EVALUATION OF READINESS FOR PROPOSAL PRESENTATION
DESIGN PROJECT COURSE

Project Title: dotBean: A Smart-storing Device for Green Coffee Bean Arabica's Moisture Content Monitoring and Storage Temperature Control

Project Members: Root, Guillen Minerva A.; Gagabi, Rome Angelo A.; Mariscotes, Reinier D.; Rotoni, Carlo R.; Agcaoili, Raven D.

Project Component	Requirements	Remarks
Manuscript	**CHAPTER 1: PROJECT BACKGROUND**	
	The Project	Ok
	The Client	Ok
	Project Objectives	Ok
	Scope and Limitation	Ok
	Project Development	Ok
	CHAPTER 2: DESIGN INPUTS	
	Client's Requirements	Ok
	Design Criteria and Design Constraints	Ok
	StoryBoard	Ok
	Other Relevant Information	Ok
	CHAPTER 3: PROJECT DESIGN	
	System Architecture (General)	Ok
	Design Process based on the System Architecture	Ok
	System Architecture for Design Option 1	Ok
	System Architecture for Design Option 2	Ok
	System Architecture for Design Option 3	Ok
	CHAPTER 4: CONSTRAINTS, STANDARDS, AND TRADE-OFFS	

	Design Constraints	OK yr.
	Importance Weighting of Constraints	OK yr.
	Trade-off Analysis	OK yr.
	Sensitivity Analysis 1	OK yr.
	Sensitivity Analysis 2	OK yr.
	Sensitivity Analysis 3	OK yr.
	Summary of All Sensitivity Analyses	OK yr.
	Design Standards	OK yr.
	CHAPTER 5: FINAL DESIGN	
	System Architecture	Checked
	System Flowchart	Checked
	Hardware Design and Development	Checked
	Software Design and Development	Checked
	System Testing	Checked
	Summary of Findings	Checked
	Assessments of the Attainment of Project Objectives	Checked
	Conclusion	Checked
	Recommendations	Checked
	Impact of Engineering Solutions	Checked
	References	Checked
	Appendices	Checked
Project Management	Time Table	Checked

8/13/2024

review

Checked by:

Date: July 12, 2024

yr. 8/13/2024

Engr. Yuneza Claire Mortos
Project Adviser

Appendix C:
Risk Management Plan

Risk Management Plan			
OBJECTIVES	**RISKS AND ASSUMPTIONS**	**STRATEGIES**	**ACTIONS TO BE TAKEN**
Determine the correct temperature and humidity levels by utilizing a temperature and humidity sensor to closely monitor the coffee beans moisture content inside a closed storage device.	**Risks:** ● Sensor inaccuracies or malfunctions. ● Data transmission errors. ● Environmental factors affecting sensor performance. **Assumptions:** ● Sensors are calibrated correctly and will provide accurate data within their specified range. ● Data transmission between sensors and the monitoring system is reliable. ● Environmental conditions are controlled.	● Regular calibration and testing of sensors. ● Use high-quality sensors. ● Implement robust data transmission protocols.	● Perform initial calibration of sensors. ● Conduct regular accuracy checks. ● Replace or repair sensors as needed. ● Use redundant sensors for verification. ● Test data transmission reliability.
Design a storage device that's capable of maintaining coffee beans' moisture content from 11% to 12.5%.	**Risks:** ● Inability to maintain precise moisture levels due to environmental fluctuations. ● Device overheating. ● Mechanical failures in the control system.	● Implement effective insulation and ventilation. ● Regularly test mechanical components.	● Design and build the storage device with temperature and humidity control. ● Ensure adequate ventilation and cooling. ● Adjust design based on test results. ● Perform routine maintenance and checks.

Risk Management Plan			
	Assumptions: ● The storage device will have effective insulation and control mechanisms to maintain stable conditions. ● Cooling and ventilation systems are adequately designed. ● Mechanical components are reliable and durable.		
Develop an automated fan control system based on a pre-defined threshold.	**Risks:** ● Fan control system may not respond accurately to changes in environmental conditions. ● Potential for fan failure. ● Programming errors or bugs. **Assumptions:** ● The fan control system will be programmed with accurate threshold values and responsive controls. ● Fans are of reliable quality. ● The control software is tested and free of major	● Implement a reliable threshold-based control algorithm. ● Use high-quality, durable fans. ● Conduct thorough testing of software.	● Develop and program the control system. ● Test the system's response to varying conditions. ● Implement fan failure detection mechanisms. ● Fine-tune thresholds and response mechanisms. ● Debug and refine software as needed.

Risk Management Plan			
	bugs.		
Develop a device that maintains an average power consumption of below 1000W per hour.	**Risks:** ● Device might exceed the power consumption limit due to inefficiencies or design flaws. ● Potential for unexpected power spikes. ● Inaccurate power consumption measurements. **Assumptions:** ● Power consumption calculations and device components will be optimized to ensure efficiency. ● Electrical systems are stable and properly rated. ● Power consumption measurements are accurate and reliable.	Optimize device design and component selection. Implement power monitoring and management. Use precise power measurement tools.	● Design the device with energy-efficient components. ● Monitor and test power usage.

Appendix D:
Time Table

the dotBean Team Timetable

Phase	Tasks/Activities	Responsibilty	Duration	Start Date	End Date	Deliverables	Remarks
Requirement Definition	Brainstorming	Gagabi Mariscotes Root Rotoni	1 day	August 14, 2023	August 14, 2023	Project design topic	Completed
	Problem Identification	Gagabi Mariscotes Root Rotoni	2 days	August 14, 2023	August 15, 2023	Focus for project design	Completed
	Client Identification	Gagabi Mariscotes Root Rotoni	1 day	August 15, 2023	August 15, 2023	Specified target clients for the project	Completed
	Client Interview	Gagabi Mariscotes Root	1 day	August 16, 2023	August 16, 2023	Current scenario	Completed
	Human Problem Identification	Gagabi Mariscotes Root Rotoni	2 days	August 16, 2023	August 17, 2023	Human problem based on client's current scenario	Completed
	Problem Validation 1 (Interview to Customers)	Gagabi Mariscotes	10 days	August 18, 2023	August 27, 2023	Validating the human problem	Completed
	Problem Validation 2 (Survey to Customers)	Gagabi Root	10 days	August 31, 2023	September 10, 2023	Statistics of customers who shared the similar problem	Completed
	Technical Problem Identification	Gagabi Root Rotoni	2 weeks	September 13, 2023	September 20, 2023	Gaps and technical problems of existing technologies	Completed
	Project Management Plan Creation	Gagabi Mariscotes Root Rotoni	5 days	September 23, 2023	September 27, 2023	Project management plan	Completed
	Design Problem Identification	Mariscotes Root	4 days	September 24, 2023	September 27, 2023	Design problem	Completed
	Project Objectives Identifcation	Gagabi Mariscotes Root Rotoni	3 days	September 25, 2023	September 27, 2023	Project objectives	Completed
	Follow Up Client Interview	Gagabi Root	1 day	October 2, 2023	October 2, 2023	Client's requirements	Completed
	Client's Requirements Idenfication	Gagabi Root	1 day	October 2, 2023	October 2, 2023	List of client's needs and wants regarding proposed solution	Completed
	Design Criteria and Constraints Idenfication	Root Rotoni	1 day	October 2, 2023	October 2, 2023	Design criteria and design constraints	Completed
	Storyboard Creation	Gagabi Mariscotes	1 day	October 2, 2023	October 3, 2023	Storyboard of propsed solution	Completed
	RRL Matrix Creation	Gagabi Root	2 days	October 2, 2023	October 3, 2023	RRL Matrix and Lit Map	Completed
	Problem Validation 3 (Interview to End-users)	Gagabi Root	4 days	October 3, 2023	October 6. 2023	End-users for testing	Completed
	Patent Search Report Finalization	Root Rotoni	9 days	October 5, 2023	October 11, 2023	Patent search report	Completed
	Relevant Information: Devices Identification	Root Rotoni	1 day	October 16, 2023	October 16, 2023	List of components/devices to use for chapter 3	Completed
	Relevant Information: Literature Map Creation	Gagabi Rotoni	3 days	October 16, 2023	October 18, 2023	Literature Map	Completed
	Design Solutions Conceptualization	Mariscotes Root	3 days	October 16, 2023	October 18, 2023	Components specifications and cost breakdown	Completed

☑		General System Architecture Creation	Mariscotes Root	2 days	October 17, 2023	October 18, 2023	System architecture	Completed
☑		Design Process Flow Creation	Mariscotes Root	2 days	October 17, 2023	October 18, 2023	Design process based on general system architecture	Completed
☑		Design Option 1 System Architecture Creation	Mariscotes Root	1 day	October 18, 2023	October 18, 2023	Design Option 1 System Architecture	Completed
☑		Design Option 2 System Architecture Creation	Mariscotes Root	1 day	October 18, 2023	October 18, 2023	Design Option 2 System Architecture	Completed
☑		Design Option 3 System Architecture Creation	Mariscotes Root	1 day	October 18, 2023	October 18, 2023	Design Option 3 System Architecture	Completed
☑		Design Option 1 Visual Prototype Creation	Gagabi Mariscotes Root	3 days	October 25, 2023	October 27, 2023	Design Option 1 Visual Prototype	Completed
☑		Components Setting up and Calibration	Mariscotes Rotoni	3 months	October 27, 2023	February 1, 2024	Ready for testing of design options	Completed
☑		Design Option 2 Visual Prototype Creation	Gagabi Mariscotes Root	3 days	November 1, 2023	November 3, 2023	Design Option 2 Visual Prototype	Completed
☑		Design Option 3 Visual Prototype Creation	Gagabi Mariscotes Root	3 days	November 4, 2023	November 6, 2023	Design Option 3 Visual Prototype	Completed
☑		Design Options Hierarchy Creation	Root	1 day	November 6, 2023	November 6, 2023	Design Options Hierarchy	Completed
☑		Data Flow Diagram Level 0 & Level 1, and Entity Relationship Diagram Creation	Root	4 days	November 6, 2023	November 9, 2023	Data Flow Diagram Level 0 & Level 1, and Entity Relationship Diagram	Completed
☑		Title Defense Propoal	Gagabi Mariscotes Root Rotoni	1 day	Novermber 22, 2023	Novermber 22, 2023	Copy of REVISION-MATRIX DotBean.docx.pdf	Completed
☑		Revising the manuscript according to the Revision Matrix	Gagabi Root Rotoni	3 weeks	November 27, 2023	December 20, 2023	dotBean Documentation [0129]	Completed
☑		Consultation with Engr. Jimbo from ME regarding Prototype	Gagabi Mariscotes	1 day	November 29, 2023	November 29, 2023	Consultation with Sir Jimbo 11/29/23	Completed
☑		Creating the Prototype layout of 3 Design Options	Gagabi Mariscotes Root	16 days	December 1, 2023	December 14, 2023	Design Options	Completed
☑		Cover Materials Canvassing	Gagabi Mariscotes Root Rotoni	3 days	January 17, 2024	January 19, 2024	Price quotation for Acrylic Plastic and Stainless Steel	Completed
☑		Meeting with the Project Adviser	Gagabi Mariscotes Root Rotoni	1 day	January 19, 2024	January 19, 2024	Revision updates approval, and additional comments for updating the Design Options	Completed
☑	Design	Radiator Fans and Peltier Modules Procuring	Gagabi Mariscotes Root Rotoni	4 days	January 19, 2024	January 23, 2024	Fan/s to be used for the final design	Completed
☑		Revision updates regarding Flowchart, Use Case Diagram, and Data Flow Diagram	Gagabi Mariscotes Root Rotoni	1 day	January 24, 2024	January 31, 2024	Updated diagrams	Completed
☑		Design Options' Prototype Layout Simulation Using Solid Works	Gagabi Mariscotes Root Rotoni	19 days	January 24, 2024	February 12, 2024	Ready for design options' testing and validation	Completed

☑		Revising Design Options' Prototype Layout based on Simulation Error Reports	Agcaoili Gagabi Mariscotes Root Rotoni	23 days	January 24, 2024	February 17, 2024	Design Options	Completed
☑		Design Options' Testing and Validation Documentation for Design Option 1 - 3	Agcaoili Gagabi Mariscotes Root Rotoni	4 months	January 30, 2024	May 31, 2024	Documentation of design criteria and constraints testing for design options	Completed
☑		Schematic Diagrams for Design Options 2 & 3 Finalization	Agcaoili	1 Week	February 04, 2024	February 12, 2024	Schematic Diagram of Design Options Components and Pin Configurations	Completed
☑		Panelist consultation and request for approval of Revision Matrix	Agcaoili Gagabi Mariscotes Root Rotoni	1 Week	February 15, 2024	February 21, 2024	Approved and Signed Revision Matrix	Completed
☑		SolidWorks Simulation Data Gathering Design Options 1-3	Gagabi Root Rotoni	5 days	February 19, 2024	February 23, 2024	dotBean Simulation Results	Completed
☑		Procurring of Agitator	Agcaoili Gagabi Mariscotes Root Rotoni	3 days	February 19, 2024	February 21, 2024	Agitator to be used for testing and validation of design option 1	Completed
☑		Setting Up and Configuration of Mobile Application, SMS, and EMail in Raspberry Pi	Mariscotes Root Rotoni	2 weeks	April 19, 2024	May 03, 2024	Chapter 3 data (testing, validation, and report)	Completed
☑		Sending of data from sensor to raspberry pi to Mobile Application/SMS/EMail	Mariscotes Root Rotoni	2 weeks	April 19, 2024	May 03, 2024	Chapter 3 Testing	Completed
☑		Revision Matrix Action Taken Form Consultation and Validation	Gagabi Mariscotes Root Rotoni	4 months	January 25, 2024	May 16, 2024	dotBean Signed Revision Matrix Action Taken Form.pdf	Completed
☑		Testing and Validation of Design Options based on Design Constraints and Criteria	Gagabi Mariscotes Root Rotoni	10 days	May 21, 2024	May 31, 2024	CHAPTER 3: PROJECT DESIGN - Computation	Completed
☑		Design Standards Research	Agcaoili Root	1 day	May 27, 2024	May 27, 2024	Computation: CHAPTER 3 PROJECT DESIGN & CHAPTER 4: CONSTRAINTS, STANDARDS, AND TRADE-OFFS	Completed
☑		Design Trade-off Analysis	Agcaoili Root	6 days	May 31, 2024	June 05, 2024		Completed
☑		Development of dotBean Hardware Project	Agcaoili Gagabi Mariscotes Root Rotoni	2 weeks	June 06, 2024	June 18, 2024	dotBean Casing Fabcrication Agreement	Completed
☑		Initial Chapter 5 Documentation	Agcaoili Root Rotoni	1 day	June 13, 2024	June 20, 2024	dotBean Documentation [0715]	Completed
☑		Adviser Checking of Chapter 3 & 4	Root	1 day	June 18, 2024	June 18, 2024		Completed
☑		Follow Up Client for the Final Draft Prototype Design pre-Testing Phase	Gagabi Root	1 day	June 18, 2024	June 18, 2024	Client Approval for Field Testing	Completed
☑	Development	Connectivity Testing of raspberry pi, sensors, actuators, and other components	Agcaoili Gagabi Mariscotes Root Rotoni	1 day	June 19, 2024	June 20, 2024		Completed

Appendix E:
Task Dependencies

Phase	Task No.	Tasks/Activities	Dependencies
Requirement Definition	1	Brainstorming	0
	2	Problem Identification	1
	3	Client Identification	2
	4	Client Interview	2,3
	5	Human Problem Identification	4
	6	Problem Validation 1 (Interview to Customers)	5
	7	Problem Validation 2 (Survey to Customers)	5,6
	8	Technical Problem Identification	5
	9	Project Management Plan Creation	0
	10	Design Problem Identification	5,8
	11	Project Objectives Identification	10
	12	Follow Up Client Interview	11
	13	Client's Requirements Identification	12
	14	Design Criteria and Constraints Identification	13
	15	Storyboard Creation	5,13
	16	RRL Matrix Creation	5,8,10,11
	17	Problem Validation 3 (Interview to End-users)	15
	18	Patent Search Report Finalization	8,16
	19	Relevant Information: Devices Identification	11,15
	20	Relevant Information: Literature Map Creation	16
Design	21	Design Solutions Conceptualization	13,14,19
	22	General System Architecture Creation	15,21
	23	Design Process Flow Creation	22
	24	Design Option 1 System Architecture Creation	21,22
	25	Design Option 2 System Architecture Creation	21,22
	26	Design Option 3 System Architecture Creation	21,22
	27	Design Option 1 Visual Prototype Creation	24
	28	Components Setting up and Calibration	24-26
	29	Design Option 2 Visual Prototype Creation	25

Phase	Task No.	Tasks/Activities	Dependencies
	30	Design Option 3 Visual Prototype Creation	26
	31	Design Options Hierarchy Creation	24-26
	32	Data Flow Diagram Level 0 & Level 1, and Entity Relationship Diagram Creation	23
	33	Title Defense Proposal	
	34	Revising the manuscript according to the Revision Matrix	33
	35	Consultation with Engr. Jimbo from ME regarding Prototype	33
	36	Creating the Prototype layout of 3 Design Options	34,35
	37	Cover Materials Canvassing	36
	38	Meeting with the Project Adviser	36
	39	Radiator Fans and Peltier Modules Procuring	36
	40	Revision updates regarding Flowchart, Use Case Diagram, and Data Flow Diagram	33
	41	Design Options' Prototype Layout Simulation Using Solid Works	35,36
	42	Revising Design Options' Prototype Layout based on Simulation Error Reports	41
	43	Design Options' Testing and Validation Documentation for Design Option 1 - 3	41,42
	44	Schematic Diagrams for Design Options 2 & 3 Finalization	24-26
	45	Panelist consultation and request for approval of Revision Matrix	34
	46	SolidWorks Simulation Data Gathering Design Options 1-3	41,42,45
	47	Procurring of Agitator	37
	48	Setting Up and Configuration of Mobile Application, SMS, and EMail in Raspberry Pi	45
	49	Sending of data from sensor to raspberry pi to Mobile Application/SMS/EMail	48,49
	50	Revision Matrix Action Taken Form Consultation and Validation	45,46,48,49

Phase	Task No.	Tasks/Activities	Dependencies
	51	Testing and Validation of Design Options based on Design Constraints and Criteria	50
	52	Design Standards Research	
	53	Design Trade-off Analysis	51
Development	54	Development of Storage Cover Material	53
	55	Initial Chapter 5 Documentation	54
	56	Adviser Checking of Chapter 3 & 4	51,53
	57	Follow Up Client for the Final Draft Prototype Design pre-Testing Phase	56
	58	Connectivity Testing of raspberry pi, sensors, actuators, and other components	54,56
	59	Initial Final Draft Prototype Design Testing and Validation Before Actual Testing On-field (Alpha testing)	58
	60	Implementation of Necessary Revision based on Initial Final Draft Prototype Design Testing	59
	61	On-site visit for Field Testing (Beta testing part 1)	58,60
Testing	62	Testing and Validation Documentation during Field Testing	61
	63	Implementation of Necessary Revision based on Field Testing	62
	64	On-site visit for Field Testing (Beta testing part 2)	63
	65	Client Approval for Final System Design	64
	66	Manuscript Finalization (Chapter 1 - 5)	65
Deployment	67	Releasing of System	65
	68	Manuscript Finalization (Chapter 1 - 5 and Appendices) and Checking	67
	69	Approval of Proposal for Final Defense	67,68

Appendix F:
Problem Validation Survey Questionnaire

dotBean Survey

To Respondents,

We, the researchers, from the **Technological Institute of the Philippines - Manila**, currently enrolled in the **Computer Engineering** program and taking the course **Project Design 1.** As part of our academic requirements for this semester, we have undertaken a research project focusing on storing coffee beans during post-harvest production.

Understanding the significance of your expertise and experience in this field, we kindly request your valuable insights and inputs on this matter. Your contribution will significantly enrich our research and aid in developing effective and practical solutions for coffee bean storage. Your expertise will play a crucial role in shaping the direction of our research and contributing to the overall success of our project.

Problem: Coffee beans are left piled up in a storage facility before distributing. The top threats of coffee beans are the varying of temperature and humidity, hence the storage conditions. Exposure of coffee beans to low temperature and high levels of humidity can lead to an absorption of moisture, resulting in building up of molds and deterioration of freshness. While high temperature and low humidity can degrade flavor and aroma.
Target Clients: Small-scale farmers and/or coffee shop owners
Proposed Solution: A smart-storage solution with built-in IoT technology for monitoring the moisture content of the coffee beans.

The purpose of this solution is to address the issue about maintaining coffee beans moisture content when stored. It is common for farmers and coffee shops to store their coffee before selling it to maintain and keep its quality. The assurance of keeping it in a usual storing process is tainted due to the fact that alongside the changes in environmental and storage conditions that highly impact the quality of coffee beans, these storing process won't keep up. There are still instances that coffee beans are go waste during the storing process, and this what the dotBean trying to minimize. Adapting to a modern technology that addresses the issue in sustainability is what the dotBean aiming for.

We eagerly anticipate the possibility of collaborating with you to achieve our research objectives. We respect your time and commitment and assure you that any information shared will be handled with utmost confidentiality and used solely for academic purposes.

Shall you have any concerns or questions, please don't hesitate to reach our representative through email:
Rome Angelo A. Gagabi (mraagagabi@tip.edu.ph)

Thank you for considering our request.

Yours truly,
The Researchers

mgmaroot01@tip.edu.ph Switch account

Not shared

* Indicates required question

Gender *

- Male
- Female
- Other:

Age *

Choose

Location *

Choose

Work *

- ○ Farmer
- ○ Distributor
- ○ Coffee Shop owner/Retailer
- ○ Barista
- ○ Consumer/Home Brewer
- ○ Other:

Do you keep your coffee beans stored? *

- ○ Yes
- ○ No

Where do you typically store your coffee beans? *

- ○ Glass/insulated jar
- ○ Vacuum-sealed container
- ○ Hermetic bag
- ○ Sack rice
- ○ Other:

Could the following storage conditions cause moisture absorption in the coffee beans? *

	Yes	No
Temperature	○	○
Humidity	○	○

Is there a significant relation between storage conditions and moisture absorption leading to mold growth? *

- ○ Yes
- ○ No

Have your encounter a situation where your coffee beans have turn into a spoiled state. *

- ○ Yes
- ○ No

If there'll be an applicable technology-based coffee bean storage in the market that ensures the preservation and maintaining the quality/freshness of coffee beans, will you be interested to purchase it? *

- ○ Yes
- ○ No

Submit

Clear form

Appendix G:
Problem Validation Survey Results

dotBean Survey

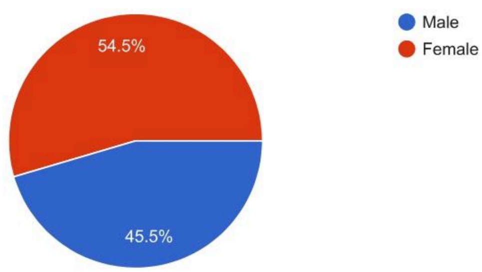
22 responses

Publish analytics

Gender

Copy

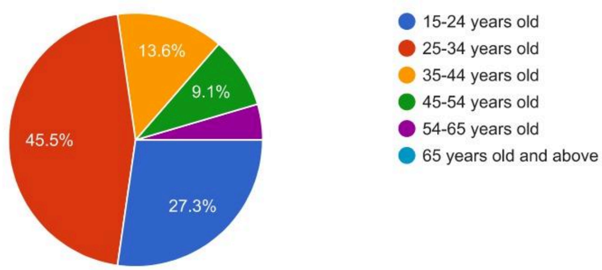
22 responses



Age

Copy

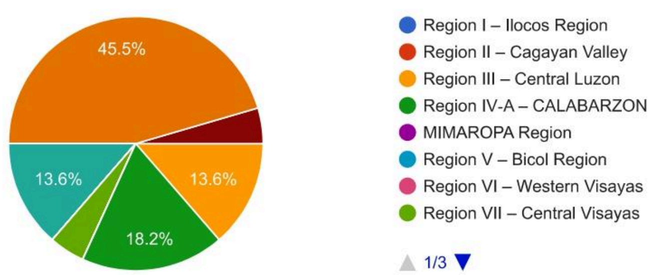
22 responses



Location

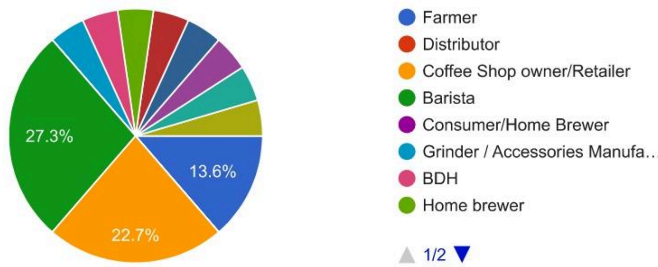
Copy

22 responses



Work Copy

22 responses

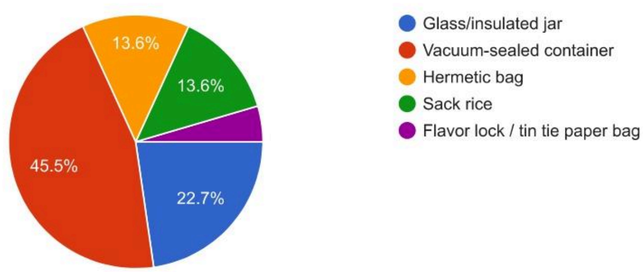


Do you keep your coffee beans stored? Copy

22 responses

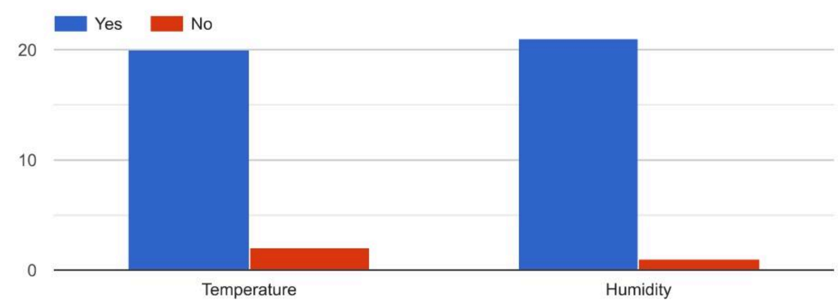
Where do you typically store your coffee beans? Copy

22 responses



Could the following storage conditions cause moisture absorption in the coffee beans?

Copy



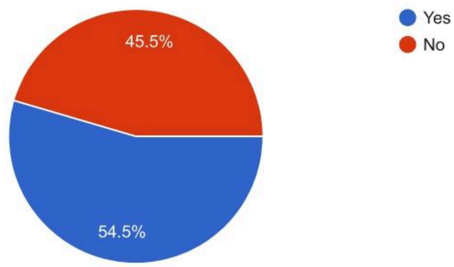
Is there a significant relation between storage conditions and moisture absorption leading to mold growth?

Copy

22 responses

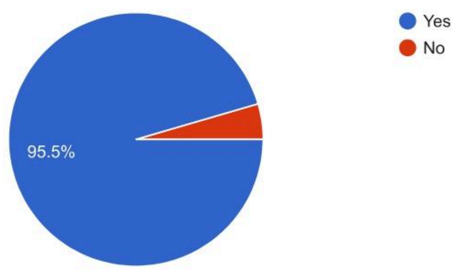
Have your encounter a situation where your coffee beans have turn into a spoiled state.

22 responses



If there'll be an applicable technology-based coffee bean storage in the market that ensures the preservation and maintaining the quality/freshness of coffee beans, will you be interested to purchase it?

22 responses



Appendix H:

Review of Related Literature Matrix

Title of the Study	Authors	Copyright	Discussion	Keysentence
Factors Affecting Coffee Beans' Attributes				
Temperature and Humidity				
Effect of green and roasted coffee storage conditions on selected characteristic quality parameters	Błaszkiewicz J. Nowakowska-Bogdan E. Barabosz K. Kulesza R. Dresler E. Woszczyński P. Biłos Ł. Matuszek DB. Szkutnik K.	2023	In order to attest the relation of these two variables: coffee beans and storage conditions, the developers used washed Arabica (W) coffee and natural Arabica (N) coffee placed in a two separate packaging materials, GrainPro (G) and Jute bag (J) for coffee samples. In terms of storage conditions, these two coffee samples, washed arabica coffee samples in a GrainPro (WG), natural Arabica coffee samples in a grain pro (NG), washed Arabica coffee samples in a Jute bag (WJ), and lastly, natural Arabica samples in a Jute bag (NJ) were placed in a controlled temperature conditions at -10°C, 10°C, and 20°C. Study suggests that the findings clearly indicate that storage of coffee beans at a temperature of 20°C does not maintain their freshness and high quality as well as the other two tested temperature chambers. The	The findings suggest that storing coffee beans at 20°C does not preserve their freshness and quality as effectively as the other two tested temperature settings. The -10°C and 10°C chambers did not yield significantly different results

Title of the Study	Authors	Copyright	Discussion	Keysentence
			−10°C and 10°C chambers did not show significant differences in results. The results obtained for the N grains showed better quality parameters, while no clear differences were observed between J and G bags.	
Climate Change and Coffee Quality: Systematic Review on the Effects of Environmental and Management Variation on Secondary Metabolites and Sensory Attributes of Coffea arabica and Coffea canephora	Ahmed, S.	2021	According to the author Ahmed, S. et al. (2021), temperature has a great impact on the quality of the coffee beans. As per Ahmed, S. coffee that grew in high temperature sometimes increased sensory attributes, while at other times, it decreased. For instance, warmer temperatures during seed development affect the coffee's sensory negatively which result in bad flavors. Additionally, there will be change in coffee flavor quality due to the negative relationship of coffee with the warm temperature. Based on the study, if there is an increase in caffeine and phenolics above the limited threshold, it means there will be a buildup of bitterness due to the decrease in its sensory attributes. In	Warmer temperature increase sensory attributes resulting in bad flavors. Increase in caffeine and phenolics above the limited threshold, means decrease in its sensory attributes.

Title of the Study	Authors	Copyright	Discussion	Keysentence
			contrast, the increase of trigonelline improves the sensory attributes that result in good flavor and aroma.	
Moisture Content				
From Plantation to Cup: Changes in Bioactive Compounds during Coffee Processing	Bastian F Hutabarat OS Dirpan A Nainu F Harapan H Emran TB Simal-Gandara J.	2021	The study suggested that moisture, among other factors, can significantly influence the quality of coffee beans. During the maturation process, the beans absorb moisture from the environment, which can affect their chemical composition and hence their flavor profile.	Moisture absorption affects coffee beans flavor profile.
Light Exposure				
Climate Change and Coffee Quality: Systematic Review on the Effects of Environmental and Management Variation on Secondary Metabolites and Sensory Attributes of Coffea arabica and Coffea canephora	Schwabe, A. L. Hansen, C. J. Hyslop, R. M. McGlaughlin, M. E.	2021	This study investigates how light exposure affects the sensory qualities of coffee, examining 12 parameters. Among the 12 articles analyzed, 8 found that increased light exposure decreased coffee quality, while 4 reported an increase in sensory attributes. Notably, changes in shade levels also played a role; decreased shade below 45% resulted in bitter, grassy, astringent, and aroma-lacking coffee, while increasing shade from 37% to 61%	Increase in light exposure decreased coffee quality and shade below 45% resulted in bitter, grassy, astringment, and aroma-lacking coffee.

Title of the Study	Authors	Copyright	Discussion	Keysentence
			improved the body of brewed coffee.	
External Odors				
Evaluation of the Olfactory Quality of Roasted Coffee Beans Using a Digital Nose	Barea-Ramos, J.D. Cascos, G. Mesias, M. Lozano, J. Martin-Vertedor, D.	2022	the volatile organic compound in the coffee beans may cause changes in their concentration and sensory strength due to the external odors that can impact them. These changes can result in a change in the coffee's aroma and flavor.	External odors can impact concentration and sensory strenght of the coffee resulting in change of its aroma and flavor
Storage				
Storage fungi and ochratoxin A associated with arabica coffee bean in postharvest processes in Northern Thailand	Mantana Maman Somsiri Sangchote Onuma Piasai Wiphawee LeesutthiphonchaiHeni k Sukorini Netnapis Khewkhom	2021	The study investigated the impact of different packaging types on fungal contamination in green coffee beans. Among the packaging types tested, coffee stored in polypropylene woven bags showed significantly lower Aspergillus contamination than those stored in ramie sacks or plastic mesh bags.	Polypropylene woven bags showed significantly lower Aspergillus contamination than those stored in ramie sacks or plastic mesh bags
Carbon Monoxide Release From Whole Bean Roasted Coffee in Storage	Alan McCarrick Benjamin Letter Shannon O'Dwyer Matthew Knighton Sara Jane Nea	2019	Beans are typically sealed in vented bags with one-way valves that release these gases, preventing oxygen from entering and degrading the beans. Storing roasted coffee in unventilated spaces can lead to dangerous CO concentrations,	Storing roasted coffee in unventilated spaces can lead to dangerous CO concentrations.

Title of the Study	Authors	Copyright	Discussion	Keysentence
			particularly in shipboard storage, raising concerns for Navy personnel safety. Some bags lost their vacuum seal, suggesting coffee delivered shortly after packaging may off-gas CO more rapidly.	
Effects of different coffee storage methods on coffee freshness after opening of packages	Samo Smrke, Jan Adam, Samuel Mühlemann, Ingo Lantz, Chahan Yeretzian	2022	The study began three weeks after roasting, focusing on freshness decline and coffee staling under various storage conditions, assessed using Gas chromatography mass spectrometry (GC/MS) for headspace above ground coffee, freshness indices, and principal component analysis (PCA). Four storage methods were assessed for coffee freshness: airtight canister, tape-resealed original package, clip-sealed package, and screw cap packaging. The screw cap method maintained freshness best, while clip, tape, or container transfer caused faster aroma loss, corresponding to oxygen and carbon dioxide levels. Roasted Arabica coffee beans from Tchibo were stored in nitrogen-filled	The screw cap method maintained freshness best, while clip, tape, or container transfer caused faster aroma loss, corresponding to oxygen and carbon dioxide levels.

Title of the Study	Authors	Copyright	Discussion	Keysentence
			1 kg bags with one-way valves. Regular oxygen and carbon dioxide measurements and freshness indices were recorded. Reference samples in sealed bags were also analyzed.	
Implementation of Automation				
Storage Conditions				
Design of Temperture and Humidity Control on Arabica Coffee Storage	Erna Kusuma Wati Fitria Hidayanti Aji Prasetya	2020	Wati, E., Hidayanti, F., and Prasetya, A. developed a post-harvest Arabica coffee bean storage system. This system maintains the storage room temperature within 19°C to 27°C and humidity between 60% to 70%, as per Badan Standardisasi Nasional (2008) guidelines. The system utilizes a fan as an actuator and a DHT 11 sensor to measure room temperature and humidity, with control managed by an Arduino-Uno. The study used two bags of Toraja Arabica green coffee, each containing 500 grams of coffee. Two DHT 11 sensors were placed on opposite sides of the storage box walls, while the Arduino-Uno	Controlled environment storage successfully maintained coffee temperature within 19°C to 27°C and humidity at 60% to 70% for approximately 20 hours, accomplishing this within less than 30 hours.

Title of the Study	Authors	Copyright	Discussion	Keysentence
			controlled the temperature. Two experiments were conducted: one in a non-conditioned storage room and the other in a non-conditioned storage room.The results revealed that the controlled environment storage successfully maintained coffee temperature within 19°C to 27°C and humidity at 60% to 70% for approximately 20 hours, accomplishing this within less than 30 hours.	
Comparison of chemical compounds and their influence on the taste of coffee depending on green beans storage conditions	Zarebska, M. Stanek, N. Barabosz, K. et al.	2022	This study assessed the impact of storage conditions on green coffee beans' chemical composition and taste quality to ensure ongoing excellence.Over three months, both natural and washed coffees initially had similar attribute levels, with aroma and acidity being prominent, and aftertaste less so. Washed coffees generally saw a 5-7% decrease in quality, while natural coffees experienced smaller drops of 4-5%. After 12 months, natural coffees maintained higher quality, with scores just below 80 points.	The 10°C storage consistently yielded the highest quality, especially for natural coffees in jute bags, which retained their quality remarkably well.

Title of the Study	Authors	Copyright	Discussion	Keysentence
			Washed coffees stored at 18°C and 20°C, in both jute and GrainPro bags, recorded significantly lower scores, losing around 12% of their initial values, especially in aroma and acidity. The 10°C storage consistently yielded the highest quality, particularly for natural coffees in jute bags, which retained their quality impressively.	
Measuring Moisture Content				
Green Coffee Bean Sorter and Corrector based on Moisture Content using Capacitive Method	Alibayan, Jan Bobadilla, Ian Carnicer, Mark Pascua, Reynaldo Teodósio, Gabriel Arago, Nilo Tolentino, Lean Karlo Fernandez, Edmon Valenzuela, Ira	2019	In a research authored by Alibayan, J., Bobadilla, I., Carnicer, M., et al.(2019) from the Technological University of the Philippines, they introduced a moisture measurement system based on capacitance technology. What distinguishes this system is its dual functionality for measuring moisture levels and also incorporates a quality correction feature. Equipped with advanced components such as electrical plates, a microcontroller, an integrated dryer, and a user-friendly display—its custom-designed software enhances precision by	Uses a electrical platers, mcu, and integrated dryer, it introduced a moisture measurement system based on capacitance technology. The system achieved an accuracy rate of approximately 91.67%

Title of the Study	Authors	Copyright	Discussion	Keysentence
			evaluating bean moisture content and categorizing them based on predefined criteria. Research findings underscore the system's proficiency in sorting beans by moisture content, achieving an impressive accuracy rate of approximately 91.67%.	
Development of a Non-Destructive Moldy Coffee Beans Detection System Based on Electronic Nose	Tang chang Lin Tang chang LinSang-Ren Yang Yi-Jhen Lin Shih-Wen Chiu Sheng-Wei Lee Kea-Tiong Tang Zhong-Kai Y	2023	This study aimed to develop an electronic nose (E-nose) system capable of detecting the odor of coffee beans, particularly focusing on identifying moldy coffee beans, which can have negative health effects when used in coffee drinks. The E-nose system consists of three main components: an environmental control system, a sensor array, and a data signal readout system. Its primary function is to differentiate between various levels of mold contamination in coffee beans based on the beans' odor. To achieve this, the researchers established a standardized procedure for collecting gas samples emitted by coffee beans in a controlled	E-nose, a system based on detecting the odor of coffee beans to identify moldy beans that are not comercially consumable. It consists of an environmental control, sensor array, and data signal readout system. The system got an accuracy of 91.77%.

Title of the Study	Authors	Copyright	Discussion	Keysentence
			environment with regulated temperature and humidity. They then recorded changes in sensor signals when these samples were introduced. Feature extraction was carried out on the collected data, followed by the application of dimensionality reduction techniques like principal component analysis (PCA) and linear discriminant analysis (LDA) to simplify the data and remove noise. For the classification task, K-nearest neighbor (KNN) and support vector machine (SVM) algorithms were employed. The resulting E-nose system demonstrated a high classification accuracy of 91.77%.	

Title of the Study	Authors	Copyright	Discussion	Keysentence
An Implementation of Convolutional Neural Network for Coffee Beans Quality Classification in a Mobile Information System	Robby Janandi Tjeng Wawan Cenggoro	2020	A mobile app, utilizing a deep learning model, automatically assesses coffee bean quality through the phone's camera. The choice between ResNet-152 and VGG16 was based on their classification performance. ResNet-152 achieved the highest accuracy at 73.3% and was successfully integrated into a functional mobile app.	A mobile app, utilizing a deep learning model, automatically assesses coffee bean quality through the phone's camera. The choice between ResNet-152 and VGG16 was based on their classification performance. ResNet-152 achieved the highest accuracy at 73.3% and was successfully integrated into a functional mobile app.

Appendix I:
Patent Search Report

PATENT SEARCH REPORT

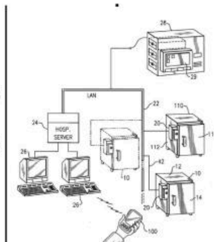
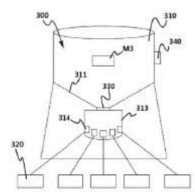
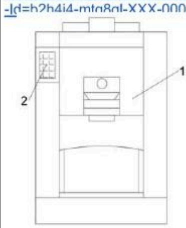
Title: dotBean: A Smart-Storage Solution on Maintaining the Coffee Bean's Quality for Small-Scale Farmers in the Philippines	
Submission Date:	
Abstract: The dotBean project aims to address the pressing issue faced by small-scale coffee farmers in the Philippines regarding climatic conditions that have impacted the coffee beans during storage. In collaboration with the Farm to Cup Philippines Association, dotBean aims to provide a comprehensive solution in the form of a smart-storage system embedded with Internet of Things (IoT) technology and its mobile application to detect, monitor, and control the temperature and humidity. The proposed solution offers an innovative approach to overcome the challenges of climate changes during the storing process. By incorporating IoT technology, the storage conditions can be continuously monitored and controlled, ensuring optimal parameters for preserving the coffee bean's quality. The air-sealed design of the storage prevents moisture from entering and safeguards the beans against mold growth. Furthermore, the ability to regulate temperature and humidity levels within the storage and through mobile application, enables the preservation of the beans' flavor and aroma. By leveraging IoT technology and a user-friendly smart-storage solution, this project offers a comprehensive solution that protects the freshness, quality, and marketability of coffee beans.	
International Patent Classification A01F 25/14 - Containers specially adapted for storing [2006.01] A01F 25/08 - Ventilating means [2006.01] *A23F 5/105 - {Treating in vacuum or with inert or noble gases; Storing in gaseous atmosphere; Packaging}* G16Y 10/05 - Agriculture [2020.01] G16Y 20/10 - relating to the environment, e.g. temperature; relating to location [2020.01] G16Y 40/10 - Detection; Monitoring [2020.01]	
Keywords Coffee, Storage, Internet of Things (IoT), Cabinet, Maintain, Quality, Temperature, Humidity, Mobile Application, Sensor, Control, Monitor	

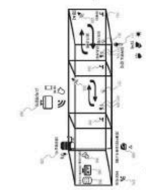
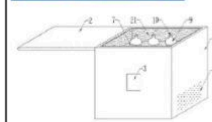
SEARCH STRINGS AND DATABASES

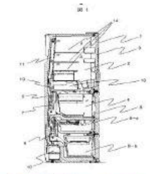
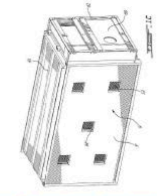
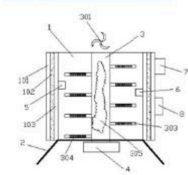
Database	Search String	Number of Hits
(Ex: USPTO, Espacenet, Patentscope, JPO-IPDL, Thomson-Reuters)		*[Ex: 205]*

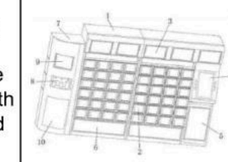
1	Espacenet	(Coffee OR "Coffee Beans") AND (Storage OR Stockpiling) AND (Temperature OR "Heat Level") AND (Humidity OR Moisture) AND ("Internet of Things" OR IoT) AND (Monitor OR Track OR Check)	914
2	Espacenet	("Storage" OR "Container") AND ("Maintain" OR "Preserve") AND ("Monitor" OR "Track") AND ("Control" OR "Regulate") AND ("IoT") AND ("Coffee Berry" OR "Coffee Beans")	27
3	Espacenet	("Coffee" OR "coffee beans") AND ("storage" OR "cabinet") AND ("quality" OR "preservation" OR "maintenance") AND ("temperature") AND ("humidity") AND ("internet of things" OR "IoT") AND ("mobile application" OR "app") AND ("sensors") AND ("monitor") AND ("control")	262
4	Espacenet	(Coffee OR Caffeine) AND (Storage OR Stockpiling) AND (Temperature OR "Heat Level") AND (Humidity OR Moisture OR Dampness) AND (Quality OR "High Standard") AND (IoT OR "Internet of Things" OR "Smart Device")	898
5	Espacenet	(Storage OR Stockpiling) AND (Temperature OR "Heat Level") AND (Humidity OR Moisture OR Dampness) AND (Monitor OR Track OR Check) AND ("Coffee Beans" OR Coffee) AND ("Mobile Application" OR "Software for mobile devices") AND (Quality OR "High Standard")	313
6	Espacenet	("Automatic") AND ("Humidity") AND ("Temperature") AND ("Control") AND ("Cabinet") AND ("Ventilation")	8054
7	WIPO	("Smart Storage" OR "Smart-rack) AND (IoT OR "Internet of Things" OR "Smart Device") AND (Coffee OR Caffeine) AND (Control OR Manage)	13
8	WIPO	(Temperature OR "Heat Level") AND (Humidity OR Moisture OR Dampness) AND (Sensor OR Detector) AND (Monitor OR Track OR Watch) AND ("Coffee Beans" OR Coffee) AND ("Mobile Application")	733
9	WIPO	(coffee OR "coffee beans") AND ("cabinet") AND ("monitor" OR "track") AND ("temperature" OR "heat level") AND ("humidity" OR "moisture") AND ("quality" OR "freshness")	947
10	WIPO	(storage OR container) AND (quality OR freshness) AND (maintain OR preserve) AND ("storing conditions") AND (sensor)	574

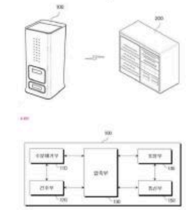
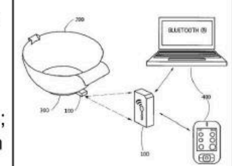
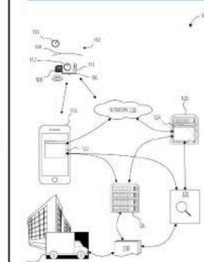
DOCUMENT INFORMATION					
Patent No.	**Ttitle**	**Publication Date**	**No of Claims**	**List of Claims**	**Illustrations**
[Ex: US5233532]	*[Ex: System for Mailing and Collecting Items]*	*[Ex: 08/03/1993]*	*Ex: Claims 1, 2*	*Ex: Claims 1, 2*	

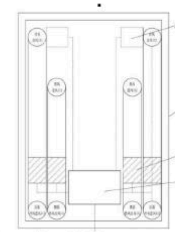
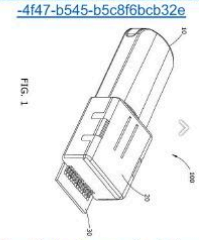
1 US20070125100	REMOTELY OR LOCALLY ACTUATED REFRIGERATOR LOCK WITH TEMPERATURE AND HUMIDITY DETECTION https://patentscope.wipo.int/search/en/detail.jsf?docId=US42108027&_cid=P10-LMYDJL-43648-1	07/06/2007	Claims 1-10, 19-25, 27, 33-39	1. A portable or mobile controlled temperature storage cabinet and door lock therefor, wherein the storage cabinet has a cabinet body in which temperature sensitive materials are to be stored and a door which closes against said cabinet body, the door lock having a body portion that mounts onto the cabinet body and a door portion that mounts onto the door to align with the body portion when the storage cabinet door is closed:	 https://patentscope.wipo.int/search/en/detail.jsf?docId=US42108027&_cid=P10-LMYDJL-43648-1
2 KR102012584B1	Smart factory type for coffee and grain of standard capacity https://worldwide.espacenet.com/patent/search?q=pn%3DKR20180089323A	08/08/2018	Claim 1	1. The apparatus according to claim 1, wherein a plurality of the discharges are arranged at predetermined intervals along a circumferential direction of the reservoir, and the work table A smart factory-type dispenser for coffee beans and cereals, characterized by a fixed amount. Wherein when it is recognized that the work is completed among the plurality of workbenches, the discharge	 htt[illegible]o m/[illegible]oc uments/KR/20180089323/A/formats/png/pages/12?EPO-Trace-Id=b2h4j4-mtq8ql-XXX-000431
3 CN216060211U	Food safety traceability coffee machine with moisture-proof function https://worldwide.espacenet.com/patent/search?q=pn%3DCN216060211U	03/18/2023	Claims 1,4,5	1.A food safety traceability coffee machine with moisture-proof function, characterized in that it comprises a coffee machine main body (1) for processing coffee, and a food safety traceability component for food safety traceability is arranged inside the coffee machine main body (1). (2) and provided with a coffee bean moisture-proof	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-706b02d0-3d9a-47bd-90e5-d0e89d8654ca

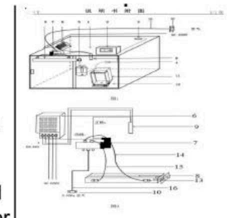
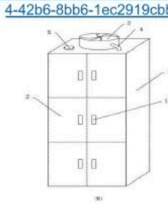
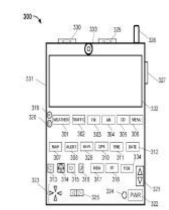
4	KR20180095329A	COFFEE NEW CROP CELLER SYSTEM AND CONTAINERIZED COFFEE NEW CROP CELLER https://worldwide.espacenet.com/patent/search?q=pn%3DKR20180095329A	08/27/2018	Claim 1,2,7	The method of claim 1, wherein the container-type coffee bean salaer further comprises at least one camera for photographing the inside of the container, and the image information photographed through the camera is transmitted to the manager server through the communication network Featured coffee bean sala	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-ebcf149a-eca4-4c49-9387-6c6be1f773c6
5	KR20150131599A	SYSTEM AND METHOD FOR CONTROLLING COFFEE ROASTING https://worldwide.espacenet.com/patent/search?q=pn%3DKR20150131599A	11/25/2015	Claims 1,5	According to claim 1, The roasting control module 110, Temperature detecting unit 110a for detecting the internal and external temperature of the coffee roaster 100, Humidity detector 110b for detecting the external humidity of the coffee roaster 100), A wind speed direction detection unit 110c for	 https:// net.com/patent/drawing?channel=espacenet_channel-ef2eacd3-e38c-418f-b014-1b324e830c51
6	CN213523729U	Coffee high-temperature processing pre-fermentation device https://worldwide.espacenet.com/patent/search?q=pn%3DCN213523729U	06/25/2021	Claim 1	1.A coffee high- temperature processing pre-fermentation device, which is characterized by comprising: an outer barrel wall, a discharge port, a control part, a feeding port sealing door, a water inlet, an exhaust fan, an exhaust pipe, a bracket, an inner barrel wall, a motor, a	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-54f8d214-12e3-476e-9257-76da9204487f
7	CN208470573U	Coffee beans storage box of traceable location https://worldwide.espacenet.com/patent/search?q=pn%3DCN208470573U	02/05/2019	Claims 1, 2	1.A coffee bean storage box with traceable positioning, comprising a box body (1) and a box cover (2), characterized in that one side of the top end of the box body (1) is hinged to one side of the box cover (2), The inside of the box (1) is	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-d1b4e617-c0a2-4b2f-9f55-e69276105d4b

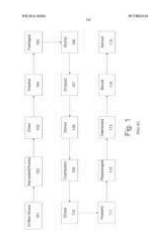
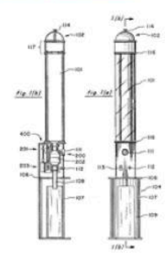
8	JP2009052855A	REFRIGERATOR https://worldwide.espacenet.com/patent/search?q=pn%3DJP2009052855A	03/12/2009	Claims 1, 2	1. In a refrigerator having a refrigerator main body formed with a plurality of storage chambers, a low-pressure chamber disposed in the storage chamber, and a decompression means for decompressing the low-pressure chamber, when the dried food is put into the	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-f43dca6c-9e3a-4adf-91b7-b814b5babfed
9	WO2012083444	COMPARTMENTED TEMPERATURE AND HUMIDITY CONTROLLED MODULAR HOUSING FOR THE STORAGE AND PRESERVATION OF WINE BOTTLES https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2012083444&_cid=P10-LMYDJL-43648-1	6/28/2012	Claims 1-34	1 . A temperature and humidity controlled housing for storing and preserving wine bottles, said housing having a storage compartment for supporting a plurality of wine bottles; a division wall in said storage compartment defining a front and a rear chamber, said housing having a door for access to said front chamber for	 https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2012083444&_cid=P10-LMYDJL-43648-1
10	CN112437934A	DELIVERY SYSTEM HAVING ROBOT VEHICLES WITH TEMPERATURE AND HUMIDITY CONTROL COMPARTMENTS https://worldwide.espacenet.com/patent/search?q=pn%3DCN112437934A	03/02/2021	Claims 1,2,3,4,5,6,7,8,9,10,15,17,20	A storage compartment, coupled to the transportation system, and configured to receive at least one item stored in the at least one storage location, the storage compartment including a temperature control module configured to The storage compartment is maintained within a predetermined temperature range to provide temperature control for the at	 https://worldwide.espacenet.com/3.2/rest-services/images/documents/CN/112437934/A/formats/png/pages/20?EPO-Trace-Id=jhti34-pyhim5-XXX-000088
11	CN210144499U	Rice seed storage device https://worldwide.espacenet.com/patent/search?q=pn%3DCN207083536U	03/13/2018	Claims 1,3,4,5	1. A rice seed storage device includes a storage box (1) disposed on the bracket (2), wherein the storage box (1) is provided with a longitudinal through storage box (1) in the middle of the storage box (1). The ventilation duct (3) is provided with a ventilation fan (301) at the top of the	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-d45d538c-1773-4330-a4ea-355c16ab6e6a

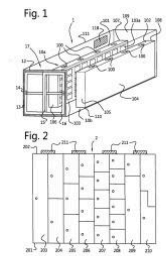
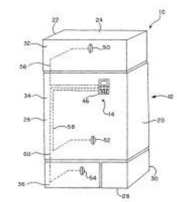
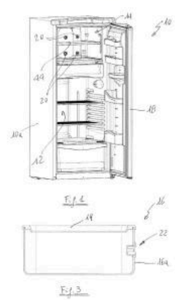
12	CN207764902U	Food storage cabinet https://worldwide.espacenet.com/patent/search?q=pn%3DCN207764902U	08/24/2018	Claims 1,2	1. A food storage cabinet comprises a food cabinet (1) and a food storage compartment (2), wherein the food cabinet (1) is provided with a plurality of cavities, the food storage compartment (2) is placed in the cavity, and the food cabinet (1) An industrial computer is arranged on one side; the food storage	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-767f217d-db9f-4d68-9e01-862c7f24c122
13	CN217408488U	Coffee machine storage device capable of preventing raw materials from being moist and deteriorated https://worldwide.espacenet.com/patent/search?q=pn%3DCN217408488U	09/13/2022	Claims 1,2,3,4,5,6,7,8	1.A material storage device for a coffee machine for preventing raw materials from being wet and deteriorating, comprising a coffee machine body (1), characterized in that one outer wall of the coffee machine body (1) is fixedly connected with two or more discharge ports (5).), the outer wall of one side of the coffee machine body (1) close to each outlet (5) is provided with a label (2); the outer wall of one side of the coffee machine body (1) is provided with the outlet (5) The same number of pressing parts: the outer wall of	 https://worldwide.espacenet.com/3.2/rest-services/images/documents/CN/217408488/U/formats/png/pages/7?EPO-Trace-Id=jhti34-qhjha9-XXX-002187
14	CN115039897A	Internet of Things control method and device suitable for coffee bean production https://worldwide.espacenet.com/patent/search?q=pn%3DCN115039897A	09/13/2022	Claims 1,2,3,4,6,7,8	1.An Internet of Things control method suitable for coffee bean production, characterized in that an Internet of Things acquisition unit is respectively set at a washing device and a drying device in advance, and an Internet of Things controller obtains the acquisition data sent by the Internet of Things acquisition unit based on the	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-0c2b88dc-0c3f-496a-bd9c-c4234b57c7c9

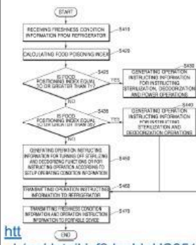
15	KR20210042475A	IoT BASED MESURING APPARATUS FOR COLLECTING COFFEE GROUNDS https://worldwide.espacenet.com/patent/search?q=pn%3DKR20210042475A	04/20/2021	Claim 1	1. A moisture removal unit that measures the input amount and volume of coffee grounds, and removes moisture through a strainer; a drying unit that dries the coffee grounds that have passed through the moisture removal unit while maintaining a preset temperature using a heating wire; and passes through the drying unit. A compression unit that	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-1ba487dd-11ed-4fa4-9539-ecf0bbfc7a44
16	EP2843965A1	Wireless apparatus https://worldwide.espacenet.com/patent/search/family/049111011/publication/EP2843965A1?q=pn%3DEP2843965A1	03/04/2015	Claim 1,2,5	1. A wireless apparatus monitoring system, comprising: a) an apparatus housing a sensor unit for monitoring conditions within the apparatus; wherein said unit comprises an EEPROM card, a sensor, a magnet and a controller. b) the monitoring device	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-7b430481-39d6-4651-b2ee-49674d54df04
17	US20220268620	CONTAINERIZED TRACKING AND REORDER SYSTEM https://patentscope.wipo.int/search/en/detail.jsf?docId=US372516142&_cid=P11-LJZVE0-35928-8	08/25/2022	Claim 1-7, 9, 10, 12-20	1. A system comprising: a plurality of vendor systems; a plurality of containers each comprising a product; wherein a first set of the containers is assigned to a first customer and a second set of the containers is assigned to a second customer; a sensor arrangement comprising at least a weight sensor for each container, and at least one motion sensor;	 https://patentscope.wipo.int/search/en/detail.jsf?docId=US372516142

18	CN211656709U	AUTOMATIC TEMPERATURE CONTROL SYSTEM OF OUTDOOR CABINET https://worldwide.espacenet.com/patent/search/family/072692590/publication/CN211656709U?q=pn%3DCN211656709U	10/09/2020	Claim 1, 2 , 3, 4, 5, 6, 7, 8	1.A temperature automatic control system for an outdoor cabinet includes a controller arranged inside the cabinet body, and the controller is electrically connected with a ventilation system, a refrigeration system, a cabinet external temperature and humidity sensor, and a cabinet internal temperature and humidity sensor. The system	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-44c91beb-859c-40ed-ae6c-33f0ccaedb30
19	CN116231502A	AUTOMATIC TEMPERATURE CONTROL SWITCH CABINET AND AUTOMATIC TEMPERATURE CONTROL METHOD https://worldwide.espacenet.com/patent/search?q=pn%3DCN116231502A	06/06/2023	Claim 1,2,3,4,5,6,7,8	1. An automatic temperature control switch cabinet, including a cabinet body (1), a base (1a), a cabinet door (2), a control module (3) and a component mounting plate (4), characterized in that: it also includes Fan I (5a), fan II (5b), fan III (5c), fan IV (5d), temperature and humidity sensor (6), exhaust tube (7)	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-e91ac626-e781-4f47-b545-b5c8f6bcb32e
20	EP3396557	WI-KEY ELECTRONIC MONITORING DEVICE AND METHOD OF USE https://patentscope.wipo.int/search/en/detail.jsf?docId=EP232545719&_cid=P11-LK00F5-93278-36	10/31/2018	Claim 1,2,4,6,8,11	1. A Wi-Key device insert-able into a compatible apparatus comprising one or more sensors able to monitor conditions within and around the compatible apparatus, the Wi-Key device comprising: a. a pluggable unit able to attach and detach from a connector unit on a long end, wherein the pluggable unit houses a micro-controller able	 https://patentscope.wipo.int/search/en/detail.jsf?docId=EP232545719&_cid=P11-LK00F5-93278-36

21	CN216848578U	TEMPERATURE AND HUMIDITY AUTOMATIC CONTROL DEVICE OF ELECTRONIC BALANCE https://worldwide.espacenet.com/patent/search?q=pn%3DCN216848578U	06/28/2022	Claim 1,2,3,4,5,6,7,8,9,10	1.An electronic balance temperature and humidity automatic control device, characterized in that it comprises a glass cabinet (1), an electronic balance (11) arranged inside the glass cabinet (1), a temperature and humidity sensor (3), a humidifier (5) and A controller (2) capable of displaying the temperature and humidity in the glass	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-7034a0e5-b2d4-42b6-8bb6-1ec2919cbb36
22	CN208480838U	FILING CABINET WITH AUTOMATIC TEMPERATURE CONTROL FUNCTION https://worldwide.espacenet.com/patent/search?q=pn%3DCN208480838U	02/12/2019	Claims 1,2,3,4,5,6,7,8,9,10	1.A filing cabinet with automatic temperature control function, comprising: a cabinet body, a cabinet door, an air exhausting mechanism and a temperature control mechanism, wherein the cabinet door is hinged with the cabinet body, and the air exhausting mechanism comprises an air exhausting device and a guiding device a	 https://worldwide.espacenet.com/patent/drawing?channel=espacenet_channel-46cac82b-fa29-47b7-a15d-25fa7ad601ba
23	US09528861	REMOTE LOCATION MONITORING https://patentscope.wipo.int/search/en/detail.jsf?docId=US189858292&_cid=P22-LJW3NG-47924-3	12/27/2016	Claims 1-20	1. A method, comprising: receiving, by a mobile computing device, a user input via an input component of the mobile computing device, the user input associated with a request by a user to monitor an output of a first sensor device; activating, by the mobile computing device, at least one of a digital camera, scanner, or barcode reader in response to the user input, to capture	 https://patentscope.wipo.int/search/en/detail.jsf?docId=US189858292&_cid=P22-LJW3NG-47924-3

24	US20140272078	COFFEE PRESERVATION METHODS https://patentscope.wipo.int/search/en/detail.jsf?docId=US107416056&_fid=WO2014140836	09/18/2014	Claims 1,3,5	What is claimed is: 1. A method for preserving green coffee beans comprising: flushing a low-oxygen permeability container with a gas, wherein the gas is an inert gas safe for food contact, wherein the gas minimally reacts with green coffee beans; including at least one oxygen absorbent in said container; filling said container with green coffee beans; and sealing said container. 3. A method for preserving green coffee beans comprising:	 https://patentscope.wipo.int/search/docs2/pct/WO2014140836/pic/2sohRFUL55-FAILRVAGYppa7ADHhUKwutISD-vNWJEFAppzt5bGgBxZVwuPITj5x
25	US5542583	DUAL CHAMBER VACUUM STORAGE AND DISPENSER FOR COFFEE BEANS https://patentscope.wipo.int/search/en/detail.jsf?docId=US38558455&_cid=P12-LJVNCE-47438-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO	08/06/1996	Claims 1,4,5,6,7	1. A system for storing and dispensing flowable food product into an external environment at atmospheric pressure comprising: means including a first chamber for storing the flowable food product; air-tight seals coupled to the first chamber to isolate the first chamber from the external environment; and means including a second chamber for removing a	 https://patentscope.wipo.int/search/en/detail.jsf?docId=US38558455&_cid=P12-LJVNCE-47438-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO
26	IN202011002260	AUTOMATED SMART COLD STORAGE SYSTEM WITH AN INTELLIGENT PERISHABLE GOODS MONITORING DEVICE https://patentscope.wipo.int/search/en/detail.jsf?docId=IN334863891&_cid=P12-LJVL0O-25392-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO	08/20/2021	Claims 1,3,4,5,6,7,10	3. Temperature chamber according to claim 1, where the dehumidifier is a solid desiccant, preferably silica, and where the container for containing the dehumidifier comprises an initial main surface comprising at least one opening, the at least one opening being provided with a multiple of moisture-permeable openings which are	not available

27	NL2019796	COOL AND DRY REMOTE STORAGE https://patentscope.wipo.int/search/en/detail.jsf?docId=NL290766641&_cid=P12-LJVL0O-25392-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO	05/02/2019	Claims 1,3,4,5,6,7,19	1. A temperature chamber (1, 3, 4, 6, 7, 9) comprising a chamber (17, 309, 407) and a cooler (118, 310) for maintaining the temperature in the chamber within a predetermined temperature range, where the chamber comprises at least one side with at least one opening (12, 304), characterized by the temperature chamber further encompassing: (1) a frame (13, 38, 22a, 22b) provided with at least one	 https://patentscope.wipo.int/search/en/detail.jsf?docId=NL290766641&_cid=P12-LJVL0O-25392-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO
28	EP1247052	STORAGE CONDITION CONTROLLER https://patentscope.wipo.int/search/en/detail.jsf?docId=EP13812213&_cid=P12-LJVL0O-25392-5&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO	10/09/2002	Claims 1,4,6,8,9,10,11	1. A refrigerator (10), having a housing (12), having multiple compartments (32, 34, 36), a cooling mechanism (16) capable of independently cooling said compartments, and a storage condition controller (14) having programmable electronic circuitry (46) that includes memory circuitry containing preprogrammed optimal storage conditions for the preservation of numerous	 FIG.1 https://patentscope.wipo.int/search/en/detail.jsf?docId=EP13812213&_cid=P12-LJVL0O-25392-5&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO
29	EP3006871	REFRIGERATOR WITH A COMPARTMENT STORING REMOVABLE FOOD CONTAINERS HAVING A CONTROLLED ENVIRONMENT https://patentscope.wipo.int/search/en/detail.jsf?docId=EP161834464&_cid=P12-LJVO3F-54883-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO	04/13/2016	Claims 1,2,3,5,6	1. Refrigerator comprising at least a compartment (14) for storing food items placed in Closed and removable containers (16) and an air regulation system (24, 24a, 26, 26a), the refrigerator comprising a plurality of docking devices (20) connected to the air regulation system in parallel and configured to be coupled with a plurality of said containers (16), wherein said air regulation system (24, 24a, 26, 26a) and said docking devices (20) are configured to draw air from the containers in order to	 https://patentscope.wipo.int/search/en/detail.jsf?docId=EP161834464&_cid=P12-LJVO3F-54883-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO

30 US20190264976	REFRIGERATOR CONTROLLING FRESHNESS ON BASIS OF CONTEXT AWARENESS AND ARTIFICIAL INTELLIGENCE, SERVER, PORTABLE DEVICE, AND METHOD FOR CONTROLLING FRESHNESS https://patentscope.wipo.int/search/en/detail.jsf?docId=EP161834464&_cid=P12-LJVO3F-54883-1&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO	08/29/2019	Claims 1,4,6,7,8,11,12,13	1. A refrigerator for controlling operations for freshness of contents based on assessed conditions, comprising: one or more partitioned storage spaces, any one or more of the storage spaces being operated based on a setup operating condition information that is preset; one or more freshness sensors configured to sense conditions outside the storage spaces or	 htt ar ch/en/detail.jsf?docId=US251054347&_cid=P12-LJVO3F-54883-2&_wv=true#detailMainForm:MyTabViewId:NATIONALBIBLIO

List of features and description (features of your proposed topic that are in the patent documents)
Implementation of Internet of Things (IoT) technology
Development of Mobile Application for remote access of controlling and monitoring the storage conditions (e.g. temperature and humidity)
Application of sensors
Application of heating/cooling and demuhidifying devices to act as actuators
Automation

List of features and description (features of your proposed topic that are NOT in the patent documents)
Design and development of smart-storage solution where:
Moisture of the roasted coffee beans are measured and monitor throughout the storage duration;
Specifically designed for storing the 4 available types of coffee beans in the Philippines; and
Portable; can be carried or travelled with at ease.

Conclusion
Developing a smart-storage solution for storing coffee beans under the post-harvest process is beneficial for farmers because:
Choosing this specific design as the main storing equipment for harvested green coffee beans, it maximizes storage capacity while utilizing minimal space applicable for limited storage area;
Monitoring the temperature, humidity, and moisture levels (automatically) capabilities ensure that the coffee beans are stored in the ideal conditions, which is crucial for maintaining their quality;
Maintaining the right temperature and humidity levels can prevent the moisture absorption that causes mold growth;
Implementing a specific cover material for coffee beans storage needs, prevent odor transfer as they're susceptible to odors leading to aroma contamination; and
This solution, similar to a mobile storage facility, can be carried/dragged at ease and convenient.

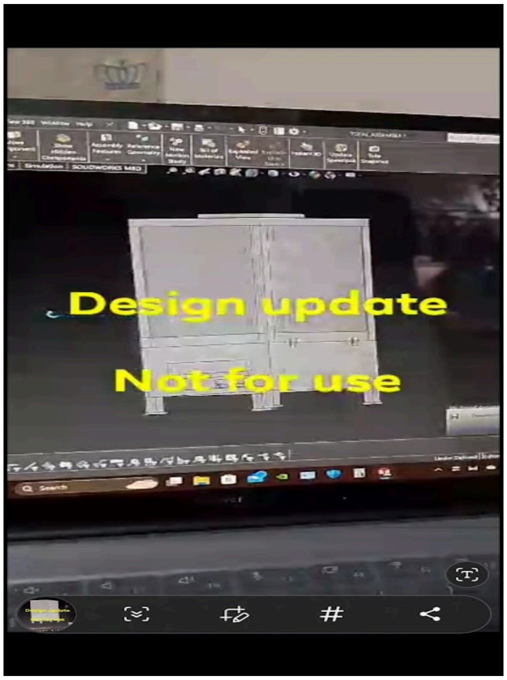
Lastly, this solution also allows remote monitoring and management through integration of IoT technology and a mobile application, which gives user at peace knowing they have constant oversight and control over their coffee bean storage, even when they are not physically present at the storage facility.

Plan in finalizing the proposed design

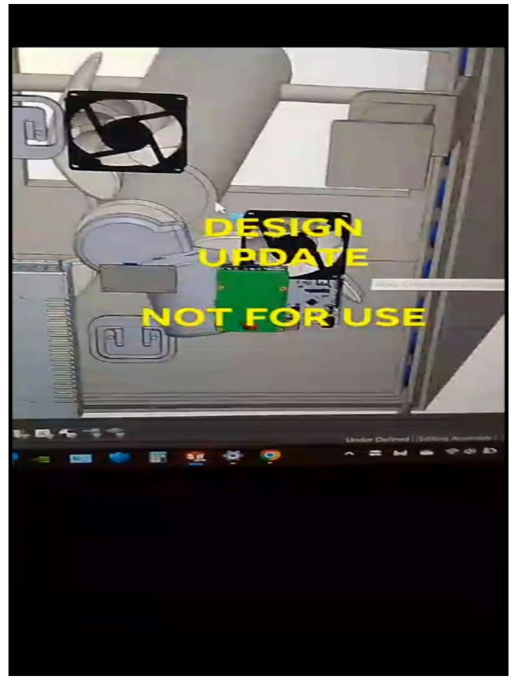
Initial research involved a comprehensive assessment of its technological feasibility; cost components and hardware specificationsand functionalities. Additionally, current efforts include collobarating and connecting with various professionals that are expert of this field. Ultimately, the developers are very open for any suggestions or inputs from reliable professionals to produce a more suitable and working prototype.

Appendix J:

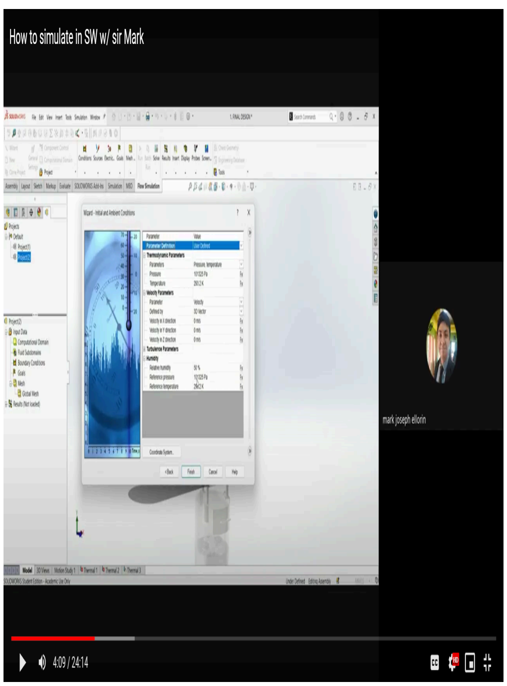
Design Options Development and Testing Documentation



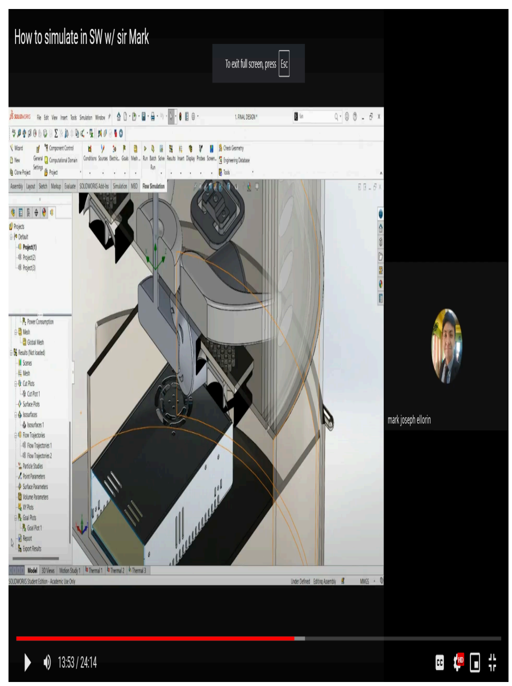
Final design of the prototype, view from the SolidWorks.



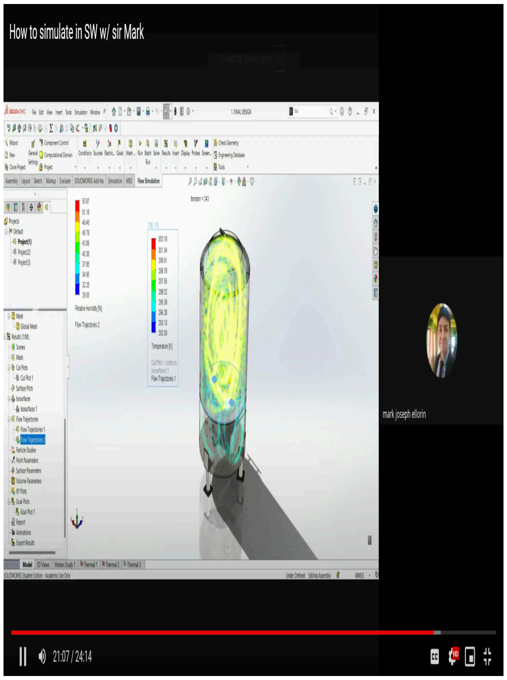
Initial placement of all the components use in the prototype using SolidWorks.



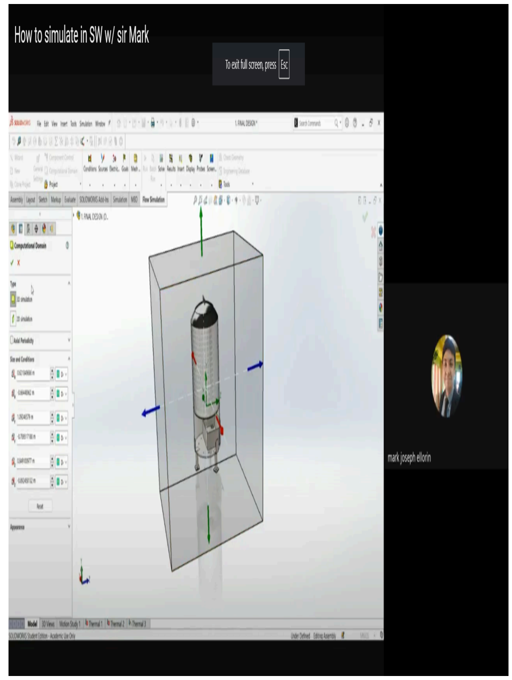
Demo on how to run the simulation, includes setting up conditions, etc.



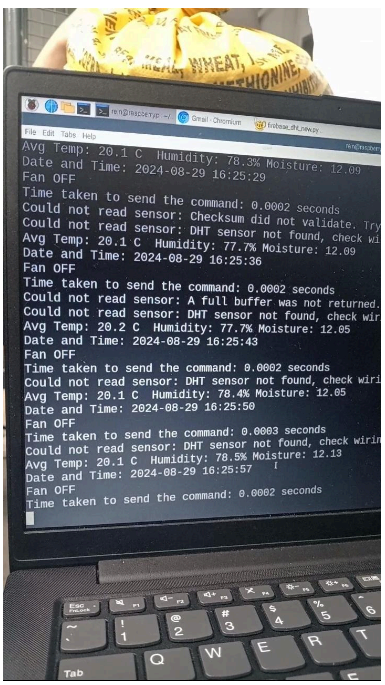
Building the component compartment step-by-step using SolidWorks.



Testing the flow simulation on every design to determine inside storage condition.

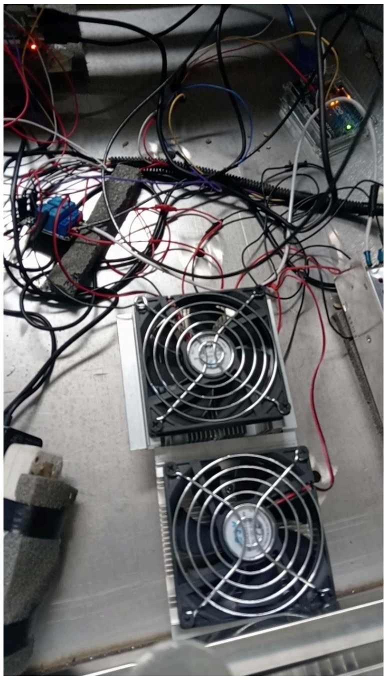


Initial design of the prototype, brainstorming on other design that could perform well.

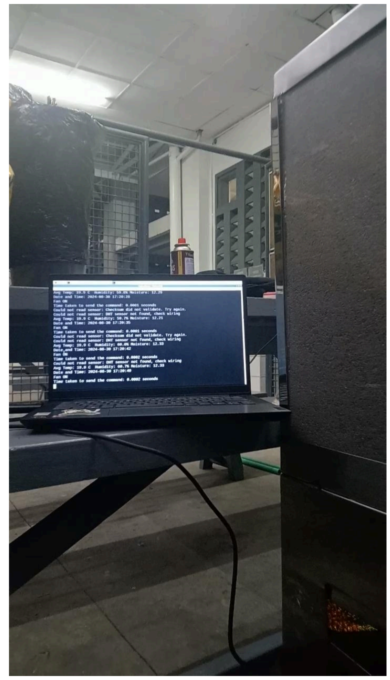


Response Time testing for Design Option 2: Raspberry Pi terminal display output (results).

The prototype in T.I.P. Project Area.



Response Time testing for Design Option 1 and Design Option 3: Thermoelectric Peltier (Fan).



Response Time testing for Design Option 1 and Design Option 3: Raspberry Pi terminal display output (results).

Appendix K:
Casing Development Fabrication Agreement

Agreement for Manufacturing of Client-Designed Casing Storage

June 05, 2024

Dear dotBean Proponents and Mr. Rodelio Orante,

This letter serves as an agreement between dotBean Proponents (the "Client") and Mr. Rodelio Orante (the "Manufacturer") for the manufacturing of storage casing. The Client will provide all designs, drawings, and specifications for the storage casings, and the Manufacturer will build the casing strictly according to these design specifications. The Manufacturer will not alter or modify the designs without prior written approval from the Client.

The total cost for the project is provided by the Manufacturer amounting (₱ 30,000.00); the Client agrees to pay 50% of the total cost, as a downpayment before manufacturing begins. The remaining 50% will be paid upon completion of the storage casing.

Any modifications requested by the Client will be communicated in writing, and any additional costs will be agreed upon before modifications commence. If the Manufacturer identifies any part of the design that seems unfitted or impractical for the proposed design, the Manufacturer must communicate these concerns to the Client in writing and await their approval before making any alterations. The Manufacturer warrants that the storage casing will be free from defects in materials and workmanship for a period of [30] from the date of completion (June 19, 2024). The Manufacturer is not liable for any defects or damages arising from the Client's designs.

Sincerely,

The dotBean Proponents

Signed by the Client's and Manufacturer's Representative/s

06-05-2024
Rome Angelo A. Gagabi
dotBean's Representative Signature Over Name and Date

Rodelio Orante
The Manufacturer's Representative Signature Over Name and Date

Reinier D. Mariscotes 06-05-2024
dotBean's Representative Signature Over Name and Date

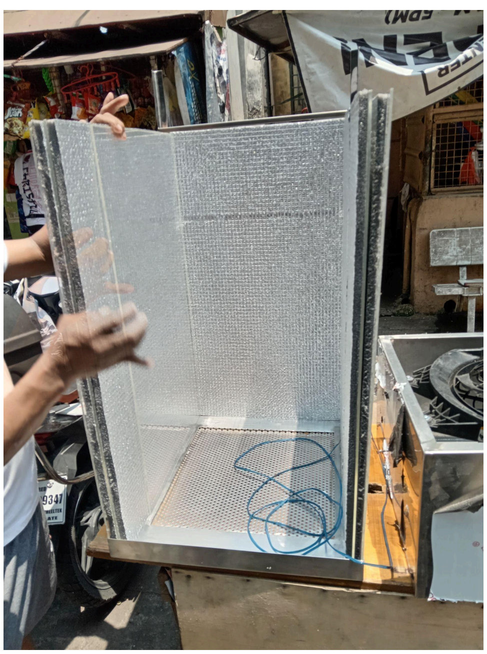
Appendix L:

Final Design Development and Field Testing Documentation

Prototype Casing Fabrication & Development

Development of the prototype component compartment, the radiator fan is now attached.

Fixing the lower part of the main storage.



Development of the main storage, wherein it has a 3-layer consisting of acrylic - polyolefin foam - and acrylic.

The final look of the prototype after attaching the components compartment and the main storage.

Prototype Setting Up Wires Connection and Components

Assembling all the wires for the three DHT22 sensors, the moisture sensor, and the fan.

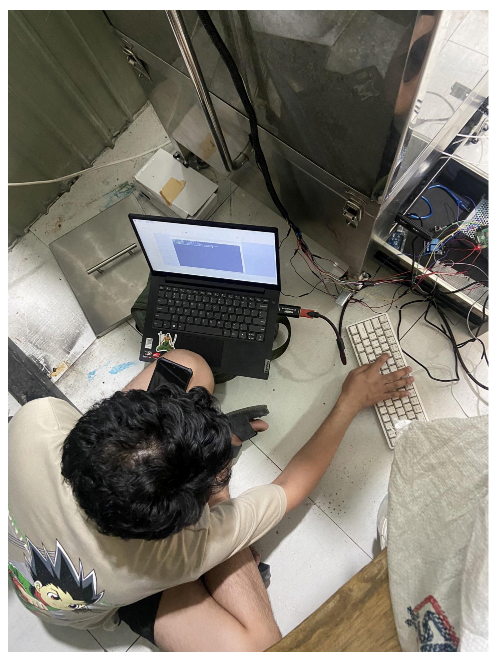
Sensors have been attached inside, and all the wires are tied to the wiring tube up to the compartment.

On Site Field Testing (June 20 - 23, 2024 & July 14 - 16, 2024) at La Trinidad, Benguet

First day in Baguio, working on how to manipulate the relay module for the radiator fan.

The prototype is now being brought to the client's place.

The prototype has been brought by the use of its built-in wheels for easy transport.



Testing the prototype in the client's storage room and monitor its performance.

Testing the prototype, gathering the temperature reading outside and inside the device.

Comparing the reading of mobile application to the actual location's temperature.

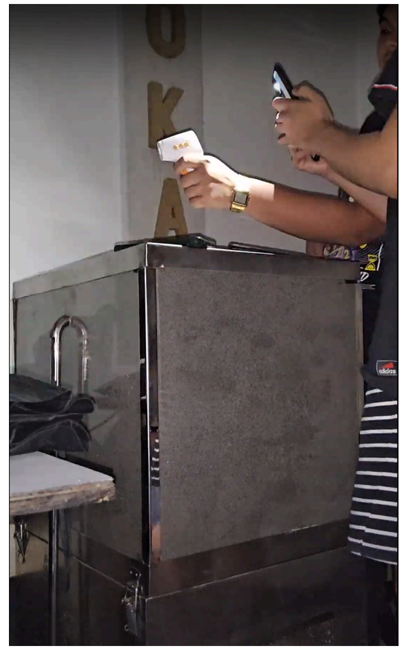
Testing at T.I.P. Project Area

Preparing the prototype for the testing of power consumption inside the school.

Testing the power consumption, gather the reading for a minute and for an hour of running.

Mocked Cold Environment Testing (3 Days Testing for Project Objective #1 and #2)

Three day testing of the device, gathering of raw reading of the prototype.



Recording the reading of the reference device and the actual prototype's reading.

Appendix M:
Client Acceptance Sheet

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

1338 Arlegui St., Quiapo, Manila

DEPARTMENT OF COMPUTER ENGINEERING

CLIENT ACCEPTANCE SHEET

The system entitled "**dotBean: A Smart-storing Device for Green Coffee Bean Arabica's Moisture Content Monitoring and Storage Temperature Control**," which was presented on the 16th of July 2024 by the proponents:

Gagabi, Rome Angelo A.

Mariscotes, Reinier D.

Root, Guillen Minerva A.

Rotoni, Carlo R.

Agcaoili, Raven D.

is hereby **ACCEPTED** by the client:

Mr. Eli Natividad
Farm to Cup Benguet

Appendix N:
User Manual

dotBean User Manual

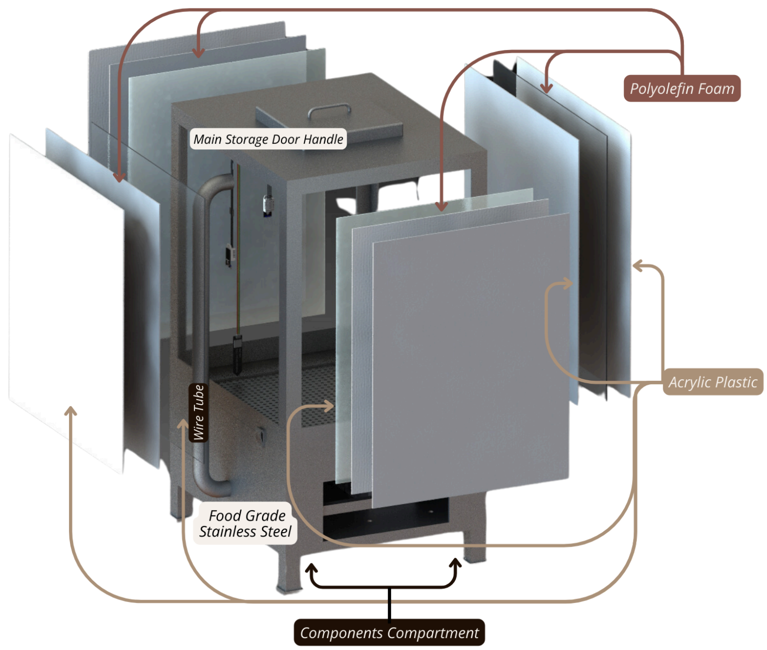
INTENDED USE

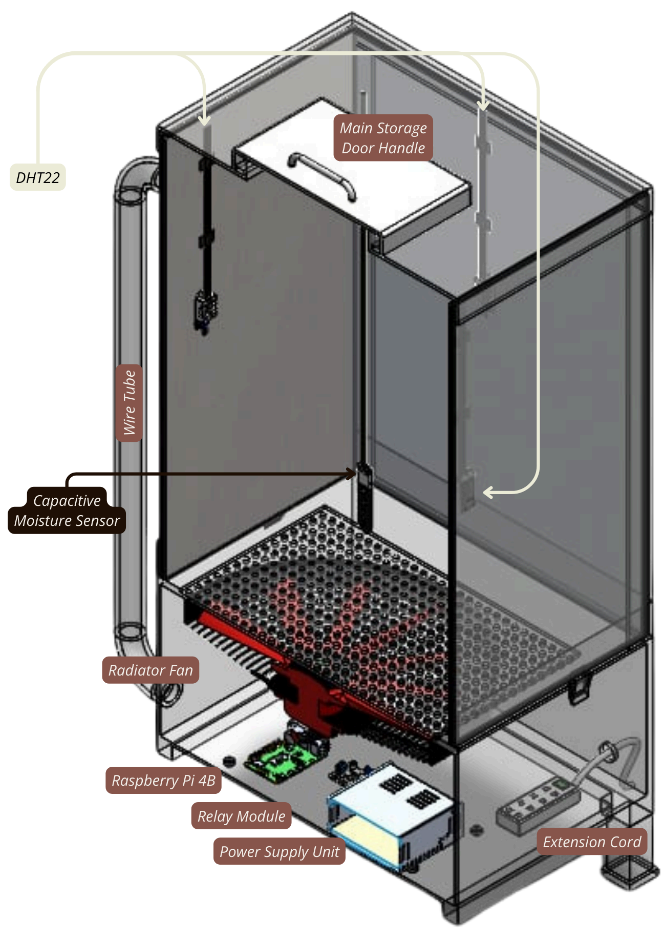
The dotBean is an automated storage device designed for farmers to store their coffee beans. This system monitors and controls temperature and humidity to maintain optimal moisture content. The dotBean can notify users of any changes via its mobile application, allowing farmers to monitor the storage conditions in real time with ease.

ACCESSORIES

- dotBean storing device
- dotBean compartment for the following components:
    - Raspberry Pi 4B
    - 12V-5V 2-channel Relay Module
    - Breadboard
    - 5V 3A Type C Charger
    - Arduino Uno R3
    - USB 2.0 Cable Type A/B
    - Power Supply
    - Extension Cord

DEVICE PARTS





HOW TO USE - Hardware

Ensure that the power cords of the Raspberry Pi, Power Supply, and Fan are connected to the extension cord and connected to an outlet.

Step 1: Let the system boot up and initialize the process.

Step 2: Transfer the dried coffee beans inside. Wait for the system to catch the initial conditions and check it with the pre-defined parameters.

Step 3: Install and load the dotBean app.

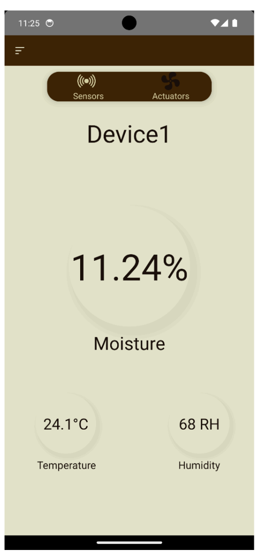
Step 4: Log in using your Google email account.

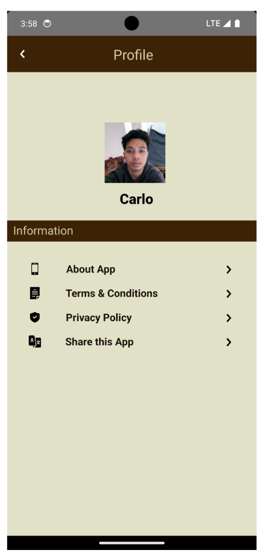
Step 5: The mobile app will begin to receive the data received by the Raspberry Pi from the sensors and the fan.

To turn it off, push the power button twice with 2-3-second intervals.

HOW TO USE - Software

Mobile Application (User Side)





Step 1: To log in, open the app on your mobile device and log in using your gmail account credentials.

Step 2: Once successfully logged in, you'll be immediately redirected to the main menu where you can see the sensor tab, where it shows the Moisture, Temperature, and Humidity.

Step 3: To access the actuator tab, click the "Actuators" icon beside the "Sensor" which currently selected. The status of the fan is presented whether is currently on or off.

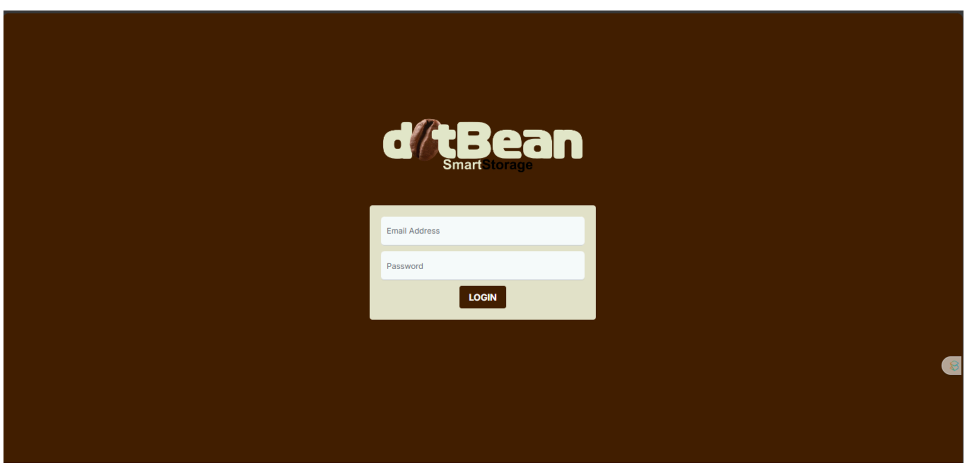
Additional Guide:

Navigating the Menu Bar:

1. Tap on the **Menu Bar** to open the profile page.
2. **User Profile**: View your profile information linked to your Gmail account.
3. **User Name**: See your registered name.
4. **Information Section**: Includes:
    a. **About App**: Learn more about the app.
    b. **Terms & Conditions**: Review the app's terms of use.
    c. **Privacy Policy**: Understand how your data is protected.
    d. **Share App**: Option to share the app with others

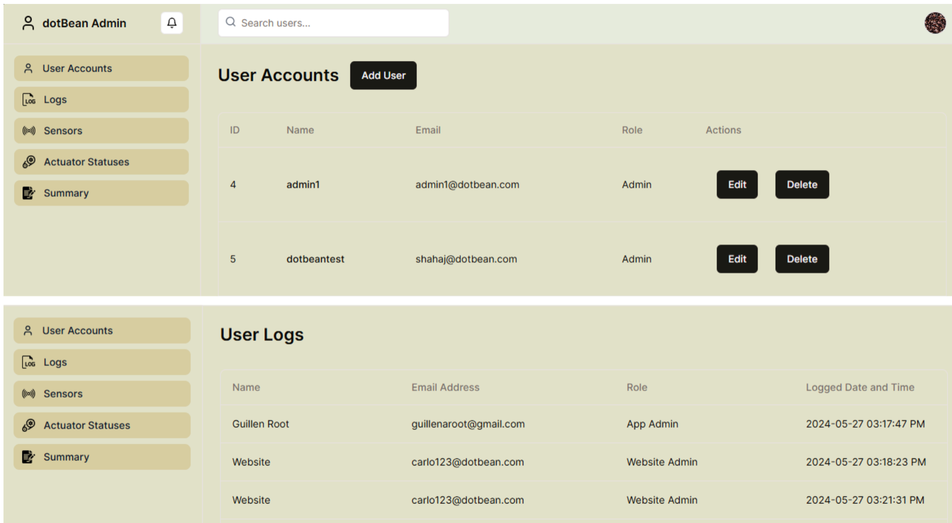
View Device Information: Once on the profile page, find the **Device Name** section to see the current dotBean device connected to your account.

Website (Admin Side)



Step 1: To begin using the website, first open it in your browser. On the login page, locate the **Email Address Input Field** and enter your pre-registered email address. Next, find the **Password Input Field** and input your corresponding password. Once both fields are

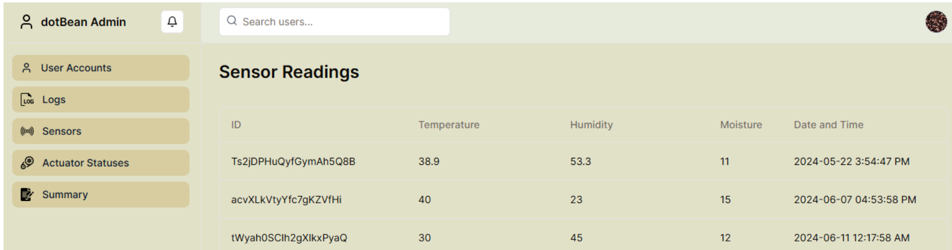
completed, click the **Submit** button. If your credentials are correct, you will be redirected to the User Accounts page, where you can start managing the website's features.

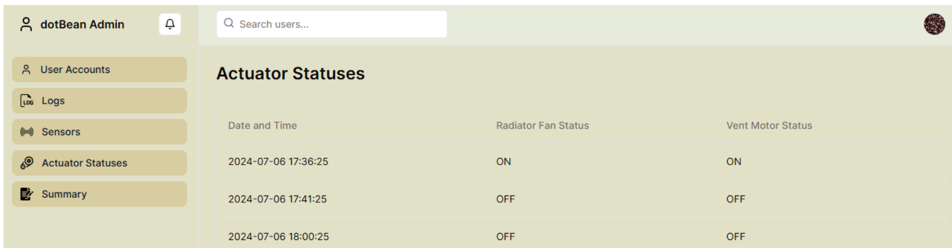


Step 2: After logging in, you will be directed to the User Accounts page. On the left side of the screen, you'll see the **Side Menu**, which provides access to various pages and tabs within the website. The current page you are viewing will be indicated by the **Page Title** at the top of the screen. This title helps you understand which section of the website you are currently on.

Step 3: To manage user accounts, click on the **User Accounts Tab** located in the Side Menu. This will bring up a comprehensive list of all registered user accounts. You can add a new user by clicking the **Add User** button. The User Accounts table will display columns including **Id** (an auto-generated ID for each account), **Name** (the name associated with the account), **Email** (the email address used), **Role** (such as App Admin, Website Admin, or Super Admin), and **Actions** (options to edit or delete the account).

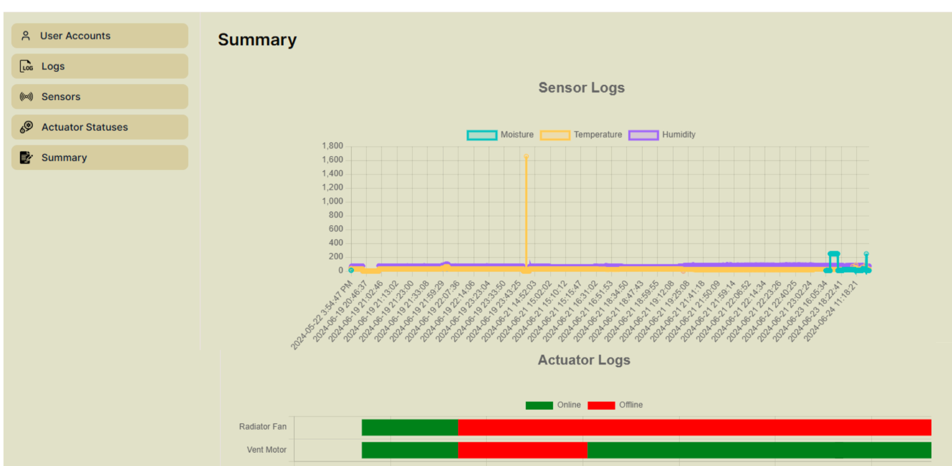
Step 4: For accessing user logs, click on the **User Logs Tab** in the Side Menu. This tab provides a table with detailed information about users who have logged into the website. The table includes columns for **Name** (the name of the logged-in user), **Email Address** (the email address used for login), **Role** (the user's role), and **Logged Date and Time** (the date and time when the user logged in).





Step 5: To view sensor data, click on the **Sensors Tab** found in the Side Menu. This tab presents a table with the latest sensor data collected from the dotBean device. The table features columns such as **Id** (an auto-generated ID for each data entry), **Temperature** (the temperature level from the DHT22 sensor), **Humidity** (the humidity level from the DHT22 sensor), and **Moisture** (the moisture content level of the coffee beans), along with the **Date and Time** when the data was received by Firebase.

Step 6: To check the statuses of actuators, click on the **Actuator Statuses Tab** in the Side Menu. The tab will display a table showing the current status of the radiator fan. Columns include **Date and Time** (when the data was received by Firebase) and **Radiator Fan Status** (whether the fan is ON or OFF).



Step 7: For a summary of sensor and actuator logs, click on the **Summary Tab** in the Side Menu. This tab provides graphs that summarize the data. The **Sensor Logs Graph** displays a daily summary of sensor logs in a line chart format, while the **Actuator Logs Graph** shows a daily summary of actuator logs in a bar chart format, providing a visual overview of the data trends.

MANAGING/MAINTAINING

1. **Weekly Sensor Checking:** Regular checking of the temperature and moisture readings that can be done by comparing the sensor reading using the reading of mobile application to the reading of Temperature Scanner and Agratronix. If there are fluctuations or inaccurate sensor readings, contact the dotBean support team for assistance.
2. **Monthly Hardware Inspection**: Inspecting all the hardware components physically such as the Raspberry Pi, Arduino Uno, and the 2-channel relay module, for any signs of tear or damage wires. Ensure all connections are secure and intact.
3. **Regular Cleaning**: The DotBean storage should be clean regularly both the below hardware compartment and the main storage to avoid any dust accumulation, which can affect sensor reading and the functionality of the radiator fan. A soft cloth can be used to wipe/clean the exterior device, and a compressed air/vacuum can be used to clean any dust/beans parchment inside the compartment and the main storage.

TROUBLESHOOTING

1. No Data Displayed on Mobile App

- **Check Internet Connection**: Ensure that both the dotBean device and your mobile phone are connected to internet.
- **Restart the Raspberry Pi**: Connectivity issues could just be resolved by rebooting the Raspberry Pi. To restart, press the push button once, wait 3 seconds and press the push button again, wait until the green light disappear. Then, press the push button again to power on.

2. Inaccurate Sensor Readings

- **Check the sensor**: If the readings seem different or unusual, look for the specific sensor that causes inaccurate reading and manually check the sensor physically if it was damaged or the wire has been removed.
- **Inspect Wiring**: Loose or damaged wires can cause inaccurate readings. Check all connections end to end using tester and replace any faulty wires.

3. **Radiator Fan Not Operating**

- **Check Power Supply**: Ensure that the fan is connected to a working power supply. Check the power supply if there is green LED light, if yes, then inspect the wiring if they are intact, else no LED light means the power supply isn't connected to the main power outlet or the power supply isn't receiving any power.
- **Inspect Relay Module**: The relay module can turn the fan on/ off. If the relay module is not functioning, it may need to be replaced.

4. **Unable to Log In to Admin Website/Mobile Application**

- **Verify Email and Password**: Double-check that you are using the correct email and password. Remember that pre-registered admin emails only are allowed access the website.
- **Contact Support**: If the issue persists, contact the dotBean support team for assistance.

Appendix O:

Source Code

Raspberry Pi 4B Source Code

```
import time
import adafruit_dht
import RPi.GPIO as GPIO
import board
import logging
import datetime
import firebase_admin
from firebase_admin import credentials, firestore
import threading
from time import sleep
import os
import logging
import serial


# Set the root logger to WARNING level to ignore INFO and DEBUG messages
logging.getLogger().setLevel(logging.WARNING)


# Load the credentials and initialize the app
cred = credentials.Certificate("dotbean.json")
firebase_admin.initialize_app(cred)

# Open serial port
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)


# Get a reference to the Firestore service
db = firestore.client()
containName= "dotbeancontainer1"
containID= 1

# Initialize the DHT devices
dht_Device1 = adafruit_dht.DHT22(board.D4, use_pulseio=False) # violet
dht_Device2 = adafruit_dht.DHT22(board.D27, use_pulseio=False) # blue
dht_Device3 = adafruit_dht.DHT22(board.D22, use_pulseio=False) # green

# Relay module pin gpio23
GPIO.setmode(GPIO.BCM)
RELAY_PIN = 23  # Change this to match your relay module's input pin
GPIO.setup(RELAY_PIN, GPIO.OUT)


# Set the initial values in Firestore
doc_ref = db.collection('sensors').document(containName)
doc_ref.set({
       'containerId': containID,
       'humidity':  12,
    'temperature':  45
})

contain_reg = db.collection('container').document(containName)
```

```
contain_reg.set({
       'connection': True,
       'containerId': containID,
    'containerName':  containName,
       'userId': 'Reinier',
})

sensorlog_ref=db.collection('sensorlog')
actuator_ref = db.collection('actuator').document('actuatorstatus')
actuator_ref.set({
       'radiator': ""
})
actuatorlog_ref=db.collection('actuatorlog')

def read_sensor(sensor):
       try:
       temperature_c = sensor.temperature
       humidity = sensor.humidity
       return temperature_c, humidity
       except RuntimeError as error:
         print(f"Could not read sensor: {error}")
       return None, None
       except Exception as error:
         print(f"An error occurred: {error}")
       sensor.exit()
       raise error

# Main loop to read the sensor values and control the fan
try:
       while True:
       temperatures = []
       humidities = []

       for sensor in [dht_Device1, dht_Device2, dht_Device3]:
             temperature, humidity = read_sensor(sensor)
              if temperature is not None and humidity is not None:
                 temperatures.append(temperature)
                 humidities.append(humidity)

       if temperatures and humidities:
             avg_temperature_c = (sum(temperatures) / len(temperatures))
             avg_humidity = (sum(humidities) / len(humidities))
if ser.in_waiting > 0:
              line = ser.readline().decode('utf-8').strip()

              # Record the start time
              start_time = time.time()

              fan_status = "OFF" if GPIO.input(RELAY_PIN) == GPIO.LOW else
"ON"

              # Fan control logic
              if 20 <= avg_temperature_c < 100:  # Condition for turning the
fan off
```

```
                print("Fan OFF")
                GPIO.output(RELAY_PIN, GPIO.LOW)  # Turn the fan off
             else:  # Default case where the fan should be off
                print("Fan ON")
                GPIO.output(RELAY_PIN, GPIO.HIGH)  # Turn the fan on

             # Record the end time
             end_time = time.time()

             # Print the averaged values to the console
            current_datetime = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
             message = (f"Avg Temp: {avg_temperature_c:.1f} C  Humidity:
{avg_humidity:.1f}% Moisture: {line}\nDate and Time: {current_datetime}")

            print(message)

             # Calculate the duration in seconds with millisecond precision
             duration = end_time - start_time
            print(f"Time taken to send the command: {duration:.4f} seconds")

            avg_temperature_c_rounded = round(avg_temperature_c, 1)
            avg_humidity_rounded = round(avg_humidity, 1)

             # Update the values in Firestore
            doc_ref.update({
                'humidity': avg_humidity_rounded,
                'temperature': avg_temperature_c_rounded,
                'moisture': line,
                'dateAndTime': current_datetime
             })
            actuator_ref.update({
                'radiator': fan_status,
                'DateAndTime': current_datetime
             })
            sensorlog_ref.add({
                'humidity': avg_humidity_rounded,
                'temperature': avg_temperature_c_rounded,
                'moisture': line,
                'dateAndTime': current_datetime
             })
            actuatorlog_ref.add({
                'radiator': fan_status,
                'DateAndTime': current_datetime
             })

        time.sleep(60)  # Adjust the sleep duration as needed
except KeyboardInterrupt:
    print("\nExiting...")
    GPIO.cleanup()  # Clean up GPIO on exit
```

Arduino Source Code

```
float slope = -0.4913; // Slope calculated from the data points
float intercept = 250.563; // Intercept calculated from the data points
const int numReadings = 3; //array number
int readings[numReadings]; //array to store
int readIndex = 0; //index of current
float total = 0; //sum of readings
float average = 0; //average of readings in an array

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0); // Read the sensor value
  // Calibrate the sensor value to match the Agratronix measurements
  float calibratedMoistureContent = (slope * sensorValue + intercept);
  // Ensure the calibrated moisture content is positive
  calibratedMoistureContent = abs(calibratedMoistureContent);

  total = total - readings[readIndex];
  readings[readIndex] = calibratedMoistureContent;
  total = total + readings[readIndex];
  readIndex = readIndex + 1;

  if (readIndex >= numReadings){
    readIndex = 0;
  }

  average = total/numReadings;
  Serial.println(average);
  delay(1000); // Delay for 1 second
}
```

Mobile Application Source Code

Homepage.js

```
import React, { useEffect, useState } from 'react';
import { View, Text, Image, TouchableOpacity, TextInput, Button } from
'react-native';
import LinearGradient from 'react-native-linear-gradient';
import { onSnapshot } from 'firebase/firestore';
import { useNavigation } from '@react-navigation/native';
import Toast from 'react-native-toast-message';
import styles from '../../components/styles';
import { FireBase_DB } from '../../FireBaseConf.ts';
import { doc } from 'firebase/firestore';
import { Shadow } from 'react-native-shadow-2';


const Homepage = () => {
 const [devicemoisture, setdevicemoisture] = useState(0);
 const [devicehumidity, setdevicehumidity] = useState(0);
 const [devicetemp, setdevicetemp] = useState(0);

 const navigation = useNavigation();

 useEffect(() => {
    const docRef = doc(FireBase_DB, 'sensors', 'dotbeancontainer1');

    // Set up real-time listener
    const unsubscribe = onSnapshot(docRef, (docSnap) => {
      if (docSnap.exists()) {
        setdevicetemp(docSnap.data().temperature);
        setdevicehumidity(docSnap.data().humidity);
        setdevicemoisture(docSnap.data().moisture);
      } else {
        console.log('No such document!');
      }
    });

    // Clean up the listener when the component unmounts
    return () => unsubscribe();
 }, []);

 const openDrawer = () => {
    navigation.openDrawer();
 };

 const showToast = () => {
    Toast.show({
      type: 'success',
      text1: 'dotBean',
      text2: 'This Feature is coming soon',
      position: 'top',
```

```
      visibilityTime: 2000,
      autoHide: true,
    });
 };

 return (
    <LinearGradient
      colors={['#E5E5CB', '#E5E5CB']}
      start={{ x: 0, y: 0 }}
      end={{ x: 0, y: 0 }}
      style={styles.container}
    >
      <View style={{ position: 'absolute', top: 0, left: 0, right: 0 }}>
        <View style={{ backgroundColor: '#3F2305', height: 50, flex: 1 }}>
          <TouchableOpacity onPress={openDrawer}
style={[styles.drawer.Button, { left: 15, top: 15 }]}>
            <Image source={require('../../assets/menu.png')}
style={[styles.drawer.Icon]} />
          </TouchableOpacity>
        </View>
      </View>

      <View style={{ flexDirection: 'row', top: -70 }}>
        <Text style={{ fontSize: 40, color: '#1A120B' }}>Device1</Text>
      </View>

      <View style={{ flexDirection: 'row', top: 230 }}>
        <Text style={{ fontSize: 30, color: '#1A120B' }}>Moisture</Text>
      </View>

      <View style={{ flexDirection: 'row', top: 400, left: 45 }}>
        <Text style={{ fontSize: 17, color: '#1A120B', right: 130, position:
'absolute' }}>Temperature </Text>
        <Text style={{ fontSize: 17, color: '#1A120B', left: 70
}}>Humidity</Text>
      </View>

      <View style={{ flexDirection: 'row', bottom: -370, left: 20 }}>
        <Shadow // TEMPERATURE CIRCLE
          distance={8}
          startColor={'#00000010'}
          containerViewStyle={{ marginVertical: 20 }}
          radius={8}
          offset={[-71, -95]}
        >
          <View style={{
            top: -100,
            right: 75,
            backgroundColor: '#E5E5CB',
            borderRadius: 50, // This makes the view circular
            width: 100,
            height: 100,
            alignItems: 'center',
            justifyContent: 'center',
```

```
          }}>
            <Text style={{ fontSize: 25, color: '#1A120B'
}}>{devicetemp}°C</Text>
          </View>
        </Shadow>

        <Shadow // HUMIDITY CIRCLE
          distance={8}
          startColor={'#00000010'}
          containerViewStyle={{ marginVertical: 20 }}
          radius={8}
          offset={[49, -95]}
        >
          <View style={{
            top: -100,
            left: 45,
            backgroundColor: '#E5E5CB',
            borderRadius: 50, // This makes the view circular
            width: 100,
            height: 100,
            alignItems: 'center',
            justifyContent: 'center',
          }}>
            <Text style={{ fontSize: 25, color: '#1A120B'
}}>{devicehumidity} RH</Text>
          </View>
        </Shadow>
      </View>

      <Shadow // MOISTURE CIRCLE
        distance={15}
        startColor={'#00000010'}
        containerViewStyle={{ marginVertical: 20 }}
        radius={8}
        offset={[7, -130]}
      >
        <View style={{
          top: -140,
          backgroundColor: '#E5E5CB',
          borderRadius: 100, // This makes the view circular
          width: 200,
          height: 200,
          alignItems: 'center',
          justifyContent: 'center',
        }}>
          <Text style={{ fontSize: 60, color: '#1A120B'
}}>{devicemoisture}%</Text>
        </View>
      </Shadow>
    </LinearGradient>
 );
};

export default Homepage;
```

Actuator.js

```
import React, { useEffect, useState } from 'react';
import { View, Text, Image, TouchableOpacity} from 'react-native';
import { Shadow } from 'react-native-shadow-2';
import LinearGradient from 'react-native-linear-gradient';
import { useNavigation } from '@react-navigation/native';
import styles from '../../components/styles';
import { FireBase_DB } from '../../FireBaseConf';
import { doc } from 'firebase/firestore';
import { onSnapshot } from 'firebase/firestore';

const Actuator = () => {

 const navigation = useNavigation();
 const [deviceradiator, setradiator] = useState(0);

  useEffect(() => {
    const docRef = doc(FireBase_DB, 'actuator', 'actuatorstatus');

    // Set up real-time listener
    const unsubscribe = onSnapshot(docRef, (docSnap) => {
      if (docSnap.exists()) {
        setradiator(docSnap.data().radiator);
      } else {
        console.log('No such document!');
      }
    });

    // Clean up the listener when the component unmounts
    return () => unsubscribe();
 }, []);

  const openDrawer = () => {
    navigation.openDrawer();
  };

  return (
    <LinearGradient
      colors={['#E5E5CB', '#E5E5CB']}
      start={{ x: 0, y: 0 }}
      end={{ x: 0, y: 0 }}
      style={styles.container}
    >

      <View style={{ position: 'absolute', top: 0, left: 0, right: 0 }}>


       <View style={{backgroundColor:'#3F2305',height:50,flex:1}}>
          <View>


          </View>
```

```
          <TouchableOpacity onPress={openDrawer}
style={[styles.drawer.Button, { left: 15, top: 15}]}>

            <Image source={require('../../assets/menu.png')}
style={[styles.drawer.Icon]} />
          </TouchableOpacity>

        </View>

      </View>

      <View style={{ flexDirection: 'row', top: -143 }}>
        <Text style={{ fontSize: 40, color: '#1A120B' }}>Device1</Text>
      </View>

    <Shadow
        distance={15}
        startColor={'#00000010'}
        containerViewStyle={{ marginVertical: 20 }}
        radius={8}
        offset={[7, 0]}>
        <View style={{
          top: -10,
          backgroundColor: '#E5E5CB',
          borderRadius: 100, // This makes the view circular
          width: 200,
          height: 200,
          alignItems: 'center',
          justifyContent: 'center',
        }}>
          <Text style  = {{fontSize:60 , color:
'black'}}>{deviceradiator}</Text>
        </View>
      </Shadow>

      <View style={{flexDirection: 'row',top:10, left:50}}>
        <Text style={{fontSize:18, color: 'black',right:0, position:
'absolute'}}>Radiator Fan</Text>
      </View>

    </LinearGradient>
  );
};

export default Actuator;
```

FireBaseConf.ts

```
// Import the functions you need from the SDKs you need
// Dev Entry : Put credentials in .env file

import { initializeApp } from "firebase/app";
import { getMessaging } from 'firebase/messaging';
import { getFirestore } from "@firebase/firestore";
import '@react-native-firebase/messaging';
import { initializeAuth, getReactNativePersistence, getAuth } from
'firebase/auth';
import ReactNativeAsyncStorage from
'@react-native-async-storage/async-storage';

const firebaseConfig = {
  apiKey: "AIzaSyAK-lOa2sr0jlFSQENNGDioUuSV2kPMqk0",
  authDomain: "dotbean-1dc7e.firebaseapp.com",
  databaseURL:
"https://dotbean-1dc7e-default-rtdb.asia-southeast1.firebasedatabase.app",
  projectId: "dotbean-1dc7e",
  storageBucket: "dotbean-1dc7e.appspot.com",
  messagingSenderId: "607873532464",
  appId: "1:607873532464:web:3234d1a7650ac261a2e3f5",
  measurementId: "G-XS03GBKKNV"
};



// Initialize Firebase
export const FireBase_APP = initializeApp(firebaseConfig);
export const FireBase_AUTH = initializeAuth(FireBase_APP, {
  persistence: getReactNativePersistence(ReactNativeAsyncStorage)
});

export const FireBase_DB = getFirestore(FireBase_APP)
```

Website Application Source Code

AdminPage.jsx

```
import React, { useEffect, useState } from 'react';
import { collection, getDocs, doc, deleteDoc } from 'firebase/firestore';
import { FireBase_DB } from "../../../firebaseConf";
import Link from "next/link";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { DropdownMenuTrigger, DropdownMenuLabel, DropdownMenuSeparator,
DropdownMenuItem, DropdownMenuContent, DropdownMenu } from
"@/components/ui/dropdown-menu";
import { TableHead, TableRow, TableHeader, TableCell, TableBody, Table }
from "@/components/ui/table";
import { Sidebar } from './sidebar';


export function AdminPage() {
  const [admins, setUsers] = useState([]);

  const fetchUsers= async () => {
    try {
      const querySnapshot = await getDocs(collection(FireBase_DB, 'admin'));
      const adminList = querySnapshot.docs.map(doc => ({ ...doc.data(), id:
doc.id }));
      setUsers(adminList);
    } catch (e) {
      console.error("Error getting documents: ", e);
    }
  };

  const redirectToLoginPage = () => {
    router.push('/');
  };


  const handleLogout = async () => {
    try {
      await signOut(auth);
      redirectToLoginPage();
    } catch (error) {
      console.error("Failed to log out:", error);
    }
  };

 useEffect(() => {
    fetchUsers();
 }, []);

  const deleteUser = async (adminId) => {
    try {
      const adminDoc = doc(FireBase_DB, 'admin', adminId);
      await deleteDoc(adminDoc);
```

```
      fetchUsers();
    } catch (error) {
      console.log(adminId)
      console.error("Error removing document: ", error);
    }
  };

  return (
    (<div style={{ backgroundColor: '#E5E5CB' }} className="grid
min-h-screen w-full lg:grid-cols-[280px_1fr]">
      <div style={{ backgroundColor: '#E5E5CB' }} className="hidden border-r
bg-gray-100/40 lg:block dark:bg-gray-800/40">
        <div className="flex h-full max-h-screen flex-col gap-2">
          <div className="flex h-[60px] items-center border-b px-6">
            <Link className="flex items-center gap-2 font-semibold"
href="#">
              <UserIcon className="h-6 w-6" />
              <span className="">dotBean Admin</span>
            </Link>
            <Button className="ml-auto h-8 w-8" size="icon"
variant="outline">
              <BellIcon className="h-4 w-4" />
              <span className="sr-only">Toggle notifications</span>
            </Button>
          </div>
          <Sidebar/>
        </div>
      </div>
      <div className="flex flex-col">
        <header
          className="flex h-14 lg:h-[60px] items-center gap-4 border-b
bg-gray-100/40 px-6 dark:bg-gray-800/40">
          <Link className="lg:hidden" href="#">
            <UserIcon className="h-6 w-6" />
            <span className="sr-only">Home</span>
          </Link>
          <div className="w-full flex-1">
            <form>
              <div className="relative">
                <SearchIcon
                  className="absolute left-2.5 top-2.5 h-4 w-4 text-gray-500
dark:text-gray-400" />
                <Input
                  className="w-full bg-white shadow-none appearance-none
pl-8 md:w-2/3 lg:w-1/3 dark:bg-gray-950"
                  placeholder="Search users..."
                  type="search" />
              </div>
            </form>
          </div>
          <DropdownMenu>
            <DropdownMenuTrigger asChild>
              <Button
                className="rounded-full border border-gray-200 w-8 h-8
```

```
dark:border-gray-800"
                size="icon"
                variant="ghost">
                <img
                  alt="Avatar"
                  className="rounded-full"
                  height="32"

src="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?auto
=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
                  style={{
                    aspectRatio: "32/32",
                    objectFit: "cover",
                  }}
                  width="32" />
                <span className="sr-only">Toggle user menu</span>
              </Button>
            </DropdownMenuTrigger>
            <DropdownMenuContent align="end">
              <DropdownMenuLabel>Device 1</DropdownMenuLabel>
              <DropdownMenuSeparator />
              <DropdownMenuItem><a href="/"
onClick={handleLogout}>Logout</a></DropdownMenuItem>
              <DropdownMenuSeparator />

            </DropdownMenuContent>
          </DropdownMenu>
        </header>

        <main className="flex flex-1 flex-col gap-4 p-4 md:gap-8 md:p-6">

        <div className="flex items-center gap-4">
  <h1 className="font-semibold text-lg md:text-2xl">User Accounts</h1>
  <Link href="/dashboard/Add">
    <Button className="ml-auto bg-primary text-primary-foreground
hover:bg-green-500/90">
      Add User
    </Button>
  </Link>
</div>
          <div className="border shadow-sm rounded-lg">
          <Table>
          <TableHeader>
                <TableRow>
                  <TableHead className="w-[80px]">ID</TableHead>
                  <TableHead className="max-w-[150px]">Name</TableHead>
                  <TableHead className="hidden
md:table-cell">Email</TableHead>
                  <TableHead className="hidden
md:table-cell">Role</TableHead>
                  <TableHead>Actions</TableHead>
                </TableRow>
              </TableHeader>
      <TableBody>
```

```
        {admins.map((admin) => (

          <TableRow key={admin.id}>

            <TableCell>{admin.uid}</TableCell>
            <TableCell className="font-medium">{admin.username}</TableCell>
            <TableCell className="hidden
md:table-cell">{admin.email}</TableCell>
            <TableCell>{admin.role}</TableCell>
            <TableCell className="hidden md:table-cell">
            <TableCell className="hidden md:table-cell">

            <Button className="bg-primary text-primary-foreground
hover:bg-yellow-500/90 button-with-margin">Edit</Button>
  <Button  className ="bg-primary text-primary-foreground
hover:bg-red-500/90"  onClick={() => deleteUser(admin.id)} >
  Delete
</Button>


                  </TableCell>
            </TableCell>
          </TableRow>
        ))}
      </TableBody>
         </Table>
          </div>
        </main>
      </div>
    </div>)
  );
}


function UserIcon(props) {
  return (
    (<svg
      {...props}

xmlns="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?au
to=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M19 21v-2a4 4 0 0 0-4-4H9a4 4 0 0 0-4 4v2" />
      <circle cx="12" cy="7" r="4" />
    </svg>)
  );
}
```

```
function BellIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M6 8a6 6 0 0 1 12 0c0 7 3 9 3 9H3s3-2 3-9" />
      <path d="M10.3 21a1.94 1.94 0 0 0 3.4 0" />
    </svg>)
  );
}


function HomeIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="m3 9 9-7 9 7v11a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2z" />
      <polyline points="9 22 9 12 15 12 15 22" />
    </svg>)
  );
}


function UsersIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
```

```
      strokeLinejoin="round">
      <path d="M16 21v-2a4 4 0 0 0-4-4H6a4 4 0 0 0-4 4v2" />
      <circle cx="9" cy="7" r="4" />
      <path d="M22 21v-2a4 4 0 0 0-3-3.87" />
      <path d="M16 3.13a4 4 0 0 1 0 7.75" />
    </svg>)
  );
}


function SearchIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <circle cx="11" cy="11" r="8" />
      <path d="m21 21-4.3-4.3" />
    </svg>)
  );
}
```

LogsPage.jsx

```
import React, { useEffect, useState } from 'react';
import { collection, getDocs} from 'firebase/firestore';
import { FireBase_DB } from "../../../firebaseConf";
import Link from "next/link";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { DropdownMenuTrigger, DropdownMenuLabel, DropdownMenuSeparator,
DropdownMenuItem, DropdownMenuContent, DropdownMenu } from
"@/components/ui/dropdown-menu";
import { TableHead, TableRow, TableHeader, TableCell, TableBody, Table }
from "@/components/ui/table";
import { Sidebar } from './sidebar';


export function LogsPage() {
  const [sensorReadings, setSensorReadings] = useState([]);

  const fetchSensorReadings = async () => {
    try {
      const querySnapshot = await getDocs(collection(FireBase_DB, 'logs'));
      const readingsList = querySnapshot.docs.map(doc => ({ ...doc.data(),
id: doc.id }));
```

```
       const sortedReadings = readingsList.sort((a, b) => new
Date(a.dateAndTime).getTime() - new Date(b.dateAndTime).getTime());
       setSensorReadings(sortedReadings);
    } catch (e) {
       console.error("Error getting sensor readings: ", e);
    }
   };

   const redirectToLoginPage = () => {
    router.push('/');
  };


  const handleLogout = async () => {
    try {
      await signOut(auth);
      redirectToLoginPage();
    } catch (error) {
      console.error("Failed to log out:", error);
    }
  };

 useEffect(() => {
    fetchSensorReadings();
 }, []);


  return (
    (<div style={{ backgroundColor: '#E5E5CB' }} className="grid
min-h-screen w-full lg:grid-cols-[280px_1fr]">
      <div style={{ backgroundColor: '#E5E5CB' }} className="hidden border-r
bg-gray-100/40 lg:block dark:bg-gray-800/40">
        <div className="flex h-full max-h-screen flex-col gap-2">
          <div className="flex h-[60px] items-center border-b px-6">
            <Link className="flex items-center gap-2 font-semibold"
href="#">
              <UserIcon className="h-6 w-6" />
              <span className="">dotBean Admin</span>
            </Link>
            <Button className="ml-auto h-8 w-8" size="icon"
variant="outline">
              <BellIcon className="h-4 w-4" />
              <span className="sr-only">Toggle notifications</span>
            </Button>
          </div>
          <Sidebar/>
        </div>
      </div>
      <div className="flex flex-col">
        <header
          className="flex h-14 lg:h-[60px] items-center gap-4 border-b
bg-gray-100/40 px-6 dark:bg-gray-800/40">
          <Link className="lg:hidden" href="#">
            <UserIcon className="h-6 w-6" />
```

```
            <span className="sr-only">Home</span>
          </Link>
          <div className="w-full flex-1">
            <form>
              <div className="relative">
                <SearchIcon
                  className="absolute left-2.5 top-2.5 h-4 w-4 text-gray-500
dark:text-gray-400" />
                <Input
                  className="w-full bg-white shadow-none appearance-none
pl-8 md:w-2/3 lg:w-1/3 dark:bg-gray-950"
                  placeholder="Search users..."
                  type="search" />
              </div>
            </form>
          </div>
          <DropdownMenu>
            <DropdownMenuTrigger asChild>
              <Button
                className="rounded-full border border-gray-200 w-8 h-8
dark:border-gray-800"
                size="icon"
                variant="ghost">
                <img
                  alt="Avatar"
                  className="rounded-full"
                  height="32"

src="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?auto
=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
                  style={{
                    aspectRatio: "32/32",
                    objectFit: "cover",
                  }}
                  width="32" />
                <span className="sr-only">Toggle user menu</span>
              </Button>
            </DropdownMenuTrigger>
            <DropdownMenuContent align="end">
              <DropdownMenuLabel>Device 1</DropdownMenuLabel>
              <DropdownMenuSeparator />
              <DropdownMenuItem><a href="/"
onClick={handleLogout}>Logout</a></DropdownMenuItem>
            </DropdownMenuContent>
          </DropdownMenu>
        </header>

        <main className="flex flex-1 flex-col gap-4 p-4 md:gap-8 md:p-6">

        <div className="flex items-center gap-4">
  <h1 className="font-semibold text-lg md:text-2xl">User Logs</h1>
</div>
          <div className="border shadow-sm rounded-lg">
            <Table>
```

```
                <TableHeader>
                    <TableRow>
                    <TableHead className="w-[250px]">Name</TableHead>
                    <TableHead className="w-[300px]">Email
Address</TableHead>
                    <TableHead className="w-[250px]">Role</TableHead>
                    <TableHead className="max-w-[250px]">Logged Date and
Time</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    {sensorReadings.map((reading) => (
                    <TableRow key={reading.id}>
                        <TableCell>{reading.fullName}</TableCell>
                        <TableCell>{reading.email}</TableCell>
                        <TableCell>{reading.role}</TableCell>
                        <TableCell>{reading.dateAndTime}</TableCell>
                    </TableRow>
                    ))}
                </TableBody>
            </Table>
          </div>
        </main>
      </div>
    </div>)
  );
}


function UserIcon(props) {
  return (
    (<svg
      {...props}

xmlns="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?au
to=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M19 21v-2a4 4 0 0 0-4-4H9a4 4 0 0 0-4 4v2" />
      <circle cx="12" cy="7" r="4" />
    </svg>)
  );
}


function BellIcon(props) {
  return (
    (<svg
```

```
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M6 8a6 6 0 0 1 12 0c0 7 3 9 3 9H3s3-2 3-9" />
      <path d="M10.3 21a1.94 1.94 0 0 0 3.4 0" />
    </svg>)
  );
}


function HomeIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="m3 9 9-7 9 7v11a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2z" />
      <polyline points="9 22 9 12 15 12 15 22" />
    </svg>)
  );
}


function UsersIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M16 21v-2a4 4 0 0 0-4-4H6a4 4 0 0 0-4 4v2" />
      <circle cx="9" cy="7" r="4" />
      <path d="M22 21v-2a4 4 0 0 0-3-3.87" />
      <path d="M16 3.13a4 4 0 0 1 0 7.75" />
```

```
    </svg>)
  );
}


function SearchIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <circle cx="11" cy="11" r="8" />
      <path d="m21 21-4.3-4.3" />
    </svg>)
  );
}
```

SensorPage.jsx

```
import React, { useEffect, useState } from 'react';
import { collection, getDocs} from 'firebase/firestore';
import { FireBase_DB } from "../../../firebaseConf";
import Link from "next/link";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { DropdownMenuTrigger, DropdownMenuLabel, DropdownMenuSeparator,
DropdownMenuItem, DropdownMenuContent, DropdownMenu } from
"@/components/ui/dropdown-menu";
import { TableHead, TableRow, TableHeader, TableCell, TableBody, Table }
from "@/components/ui/table";
import { Sidebar } from './sidebar';


export function SensorPage() {
  const [sensorReadings, setSensorReadings] = useState([]);

  const fetchSensorReadings = async () => {
    try {
       const querySnapshot = await getDocs(collection(FireBase_DB,
'sensorlog'));
       const readingsList = querySnapshot.docs.map(doc => ({ ...doc.data(),
id: doc.id }));
       const sortedReadings = readingsList.sort((a, b) => new
Date(a.dateAndTime).getTime() - new Date(b.dateAndTime).getTime());
       setSensorReadings(sortedReadings);
    } catch (e) {
```

```
        console.error("Error getting sensor readings: ", e);
     }
    };

    const redirectToLoginPage = () => {
     router.push('/');
   };


   const handleLogout = async () => {
     try {
       await signOut(auth);
       redirectToLoginPage();
     } catch (error) {
       console.error("Failed to log out:", error);
     }
   };

  useEffect(() => {
     fetchSensorReadings();
  }, []);
   return (
     (<div style={{ backgroundColor: '#E5E5CB' }} className="grid
min-h-screen w-full lg:grid-cols-[280px_1fr]">
       <div style={{ backgroundColor: '#E5E5CB' }} className="hidden border-r
bg-gray-100/40 lg:block dark:bg-gray-800/40">
         <div className="flex h-full max-h-screen flex-col gap-2">
           <div className="flex h-[60px] items-center border-b px-6">
             <Link className="flex items-center gap-2 font-semibold"
href="#">
               <UserIcon className="h-6 w-6" />
               <span className="">dotBean Admin</span>
             </Link>
             <Button className="ml-auto h-8 w-8" size="icon"
variant="outline">
               <BellIcon className="h-4 w-4" />
               <span className="sr-only">Toggle notifications</span>
             </Button>
           </div>
           <Sidebar/>
         </div>
       </div>
       <div className="flex flex-col">
         <header
           className="flex h-14 lg:h-[60px] items-center gap-4 border-b
bg-gray-100/40 px-6 dark:bg-gray-800/40">
           <Link className="lg:hidden" href="#">
             <UserIcon className="h-6 w-6" />
             <span className="sr-only">Home</span>
           </Link>
           <div className="w-full flex-1">
             <form>
               <div className="relative">
                 <SearchIcon
```

```
                  className="absolute left-2.5 top-2.5 h-4 w-4 text-gray-500
dark:text-gray-400" />
                <Input
                  className="w-full bg-white shadow-none appearance-none
pl-8 md:w-2/3 lg:w-1/3 dark:bg-gray-950"
                  placeholder="Search users..."
                  type="search" />
              </div>
            </form>
          </div>
          <DropdownMenu>
            <DropdownMenuTrigger asChild>
              <Button
                className="rounded-full border border-gray-200 w-8 h-8
dark:border-gray-800"
                size="icon"
                variant="ghost">
                <img
                  alt="Avatar"
                  className="rounded-full"
                  height="32"

src="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?auto
=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
                  style={{
                    aspectRatio: "32/32",
                    objectFit: "cover",
                  }}
                  width="32" />
                <span className="sr-only">Toggle user menu</span>
              </Button>
            </DropdownMenuTrigger>
            <DropdownMenuContent align="end">
              <DropdownMenuLabel>Device 1</DropdownMenuLabel>
              <DropdownMenuSeparator />
              <DropdownMenuItem><a href="/"
onClick={handleLogout}>Logout</a></DropdownMenuItem>
            </DropdownMenuContent>
          </DropdownMenu>
        </header>

        <main className="flex flex-1 flex-col gap-4 p-4 md:gap-8 md:p-6">

        <div className="flex items-center gap-4">
  <h1 className="font-semibold text-lg md:text-2xl">Sensor Readings</h1>
</div>
          <div className="border shadow-sm rounded-lg">
            <Table>
                <TableHeader>
                <TableRow>
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[100px]">ID</TableHead>
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[150px]">Temperature</TableHead>
```

```
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[150px]">Humidity</TableHead>
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[150px]">Moisture</TableHead>
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[150px]">Date and Time</TableHead>
                </TableRow>
                </TableHeader>
                <TableBody>
                  {sensorReadings.map((reading) => (
                    <TableRow key={reading.id}>
                      <TableCell>{reading.id}</TableCell>
                      <TableCell>{reading.temperature}</TableCell>
                      <TableCell>{reading.humidity}</TableCell>
                      <TableCell>{reading.moisture}</TableCell>
                      <TableCell>{reading.dateAndTime}</TableCell>
                    </TableRow>
                  ))}
                </TableBody>
            </Table>
          </div>
        </main>
      </div>
    </div>)
  );
}


function UserIcon(props) {
  return (
    (<svg
      {...props}

xmlns="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?au
to=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M19 21v-2a4 4 0 0 0-4-4H9a4 4 0 0 0-4 4v2" />
      <circle cx="12" cy="7" r="4" />
    </svg>)
  );
}


function BellIcon(props) {
  return (
    (<svg
      {...props}
```

```
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M6 8a6 6 0 0 1 12 0c0 7 3 9 3 9H3s3-2 3-9" />
      <path d="M10.3 21a1.94 1.94 0 0 0 3.4 0" />
    </svg>)
  );
}


function HomeIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="m3 9 9-7 9 7v11a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2z" />
      <polyline points="9 22 9 12 15 12 15 22" />
    </svg>)
  );
}


function UsersIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M16 21v-2a4 4 0 0 0-4-4H6a4 4 0 0 0-4 4v2" />
      <circle cx="9" cy="7" r="4" />
      <path d="M22 21v-2a4 4 0 0 0-3-3.87" />
      <path d="M16 3.13a4 4 0 0 1 0 7.75" />
    </svg>)
```

```
  );
}


function SearchIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <circle cx="11" cy="11" r="8" />
      <path d="m21 21-4.3-4.3" />
    </svg>)
  );
}
```

ActuatorPage.jsx

```
import React, { useEffect, useState } from 'react';
import { collection, getDocs} from 'firebase/firestore';
import { FireBase_DB } from "../../../firebaseConf";
import Link from "next/link";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { DropdownMenuTrigger, DropdownMenuLabel, DropdownMenuSeparator,
DropdownMenuItem, DropdownMenuContent, DropdownMenu } from
"@/components/ui/dropdown-menu";
import { TableHead, TableRow, TableHeader, TableCell, TableBody, Table }
from "@/components/ui/table";
import { Sidebar } from './sidebar';


export function ActuatorPage() {
  const [sensorReadings, setSensorReadings] = useState([]);

  const fetchSensorReadings = async () => {
    try {
       const querySnapshot = await getDocs(collection(FireBase_DB,
'actuatorlog'));
       const readingsList = querySnapshot.docs.map(doc => ({ ...doc.data(),
id: doc.id }));
       setSensorReadings(readingsList);
    } catch (e) {
       console.error("Error getting sensor readings: ", e);
    }
   };
```

```
 useEffect(() => {
    fetchSensorReadings();
 }, []);


  return (
    (<div style={{ backgroundColor: '#E5E5CB' }} className="grid
min-h-screen w-full lg:grid-cols-[280px_1fr]">
      <div style={{ backgroundColor: '#E5E5CB' }} className="hidden border-r
bg-gray-100/40 lg:block dark:bg-gray-800/40">
        <div className="flex h-full max-h-screen flex-col gap-2">
          <div className="flex h-[60px] items-center border-b px-6">
            <Link className="flex items-center gap-2 font-semibold"
href="#">
              <UserIcon className="h-6 w-6" />
              <span className="">dotBean Admin</span>
            </Link>
            <Button className="ml-auto h-8 w-8" size="icon"
variant="outline">
              <BellIcon className="h-4 w-4" />
              <span className="sr-only">Toggle notifications</span>
            </Button>
          </div>
          <Sidebar/>
        </div>
      </div>
      <div className="flex flex-col">
        <header
          className="flex h-14 lg:h-[60px] items-center gap-4 border-b
bg-gray-100/40 px-6 dark:bg-gray-800/40">
          <Link className="lg:hidden" href="#">
            <UserIcon className="h-6 w-6" />
            <span className="sr-only">Home</span>
          </Link>
          <div className="w-full flex-1">
            <form>
              <div className="relative">
                <SearchIcon
                  className="absolute left-2.5 top-2.5 h-4 w-4 text-gray-500
dark:text-gray-400" />
                <Input
                  className="w-full bg-white shadow-none appearance-none
pl-8 md:w-2/3 lg:w-1/3 dark:bg-gray-950"
                  placeholder="Search users..."
                  type="search" />
              </div>
            </form>
          </div>
          <DropdownMenu>
            <DropdownMenuTrigger asChild>
              <Button
                className="rounded-full border border-gray-200 w-8 h-8
dark:border-gray-800"
                size="icon"
```

```
                variant="ghost">
                <img
                  alt="Avatar"
                  className="rounded-full"
                  height="32"

src="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?auto
=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
                  style={{
                    aspectRatio: "32/32",
                    objectFit: "cover",
                  }}
                  width="32" />
                <span className="sr-only">Toggle user menu</span>
              </Button>
            </DropdownMenuTrigger>
            <DropdownMenuContent align="end">
              <DropdownMenuLabel>My Account</DropdownMenuLabel>
              <DropdownMenuSeparator />
              <DropdownMenuItem>Settings</DropdownMenuItem>
              <DropdownMenuItem>Support</DropdownMenuItem>
              <DropdownMenuSeparator />
              <DropdownMenuItem>Logout</DropdownMenuItem>
            </DropdownMenuContent>
          </DropdownMenu>
        </header>

        <main className="flex flex-1 flex-col gap-4 p-4 md:gap-8 md:p-6">

        <div className="flex items-center gap-4">
  <h1 className="font-semibold text-lg md:text-2xl">Actuator Statuses</h1>
</div>
          <div className="border shadow-sm rounded-lg">
            <Table>
                <TableHeader>
                <TableRow>
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[60px]">Date and Time</TableHead>
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[150px]">Radiator Fan Status</TableHead>
                  <TableHead className="flex-grow-0 flex-shrink-0
max-w-[150px]">Vent Motor Status</TableHead>
                </TableRow>
                </TableHeader>
                <TableBody>
                  {sensorReadings.map((reading) => (
                    <TableRow key={reading.id}>
                      <TableCell>{reading.DateAndTime}</TableCell>
                      <TableCell>{reading.radiator}</TableCell>
                      <TableCell>{reading.vent}</TableCell>
                    </TableRow>
                  ))}
                </TableBody>
            </Table>
```

```
          </div>
        </main>
      </div>
    </div>)
  );
}
function UserIcon(props) {
  return (
    (<svg
      {...props}

xmlns="https://images.pexels.com/photos/1695052/pexels-photo-1695052.jpeg?au
to=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M19 21v-2a4 4 0 0 0-4-4H9a4 4 0 0 0-4 4v2" />
      <circle cx="12" cy="7" r="4" />
    </svg>)
  );
}


function BellIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M6 8a6 6 0 0 1 12 0c0 7 3 9 3 9H3s3-2 3-9" />
      <path d="M10.3 21a1.94 1.94 0 0 0 3.4 0" />
    </svg>)
  );
}


function HomeIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
```

```
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="m3 9 9-7 9 7v11a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2z" />
      <polyline points="9 22 9 12 15 12 15 22" />
    </svg>)
  );
}

function UsersIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <path d="M16 21v-2a4 4 0 0 0-4-4H6a4 4 0 0 0-4 4v2" />
      <circle cx="9" cy="7" r="4" />
      <path d="M22 21v-2a4 4 0 0 0-3-3.87" />
      <path d="M16 3.13a4 4 0 0 1 0 7.75" />
    </svg>)
  );
}

function SearchIcon(props) {
  return (
    (<svg
      {...props}
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
      stroke="currentColor"
      strokeWidth="2"
      strokeLinecap="round"
      strokeLinejoin="round">
      <circle cx="11" cy="11" r="8" />
      <path d="m21 21-4.3-4.3" />
    </svg>)
  );
}
```

FirebaseConf.ts

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { initializeAuth, getAuth } from 'firebase/auth';
import { getFirestore } from "@firebase/firestore";

const firebaseConfig = {
  apiKey: "AIzaSyAK-lOa2sr0jlFSQENNGDioUuSV2kPMqk0",
  authDomain: "dotbean-1dc7e.firebaseapp.com",
  projectId: "dotbean-1dc7e",
  storageBucket: "dotbean-1dc7e.appspot.com",
  messagingSenderId: "607873532464",
  appId: "1:607873532464:web:3234d1a7650ac261a2e3f5",
  measurementId: "G-XS03GBKKNV"
};

// Initialize Firebase

export const FireBase_APP = initializeApp(firebaseConfig);
export const FireBase_AUTH = initializeAuth(FireBase_APP, {});
export const FireBase_DB = getFirestore(FireBase_APP)
export const SignUp_AUTH = getAuth(FireBase_APP)
```

Appendix P:

Roles and Responsibilities

Team Members	Roles	Responsibilities
Root, Guillen Minerva A.	*System Analyst*	Gather and analyze requirements, design system solutions, and align with project goals. Conduct feasibility studies, and cost-benefit analysis, and collaborate with stakeholders and technical teams.
	Documentation Analyst	Facilitates the overall progress and accuracy of the contents of the documentation.
	System Architect	Designs and develops the overall project blueprint that encompasses both the structural and elemental aspects, covering hardware and software.
	Test Specialist	Plans and conducts thorough testing, including unit, integration, and user acceptance testing.
	Financier	Provides an adequate budget contribution to support the development and progress of the project.
Gagabi, Rome Angelo A.	*Client Relations Coordinator*	Facilitates communication between the project team and the client, ensuring the integration of client expectations, feedback, and requirements into the project.
	Documentation Specialist	Maintains comprehensive project documentation to facilitate effective project management and knowledge sharing.
	Hardware Engineer Assistant	Assisting on designing and developing the necessary hardware for the system.
	Financier	Provides an adequate budget contribution to support the development and progress of the project.
Mariscotes, Reinier D.	*Software Engineer*	Leads on designing and developing the necessary software for the system

	Hardware Engineer	Leads on designing and developing the necessary hardware for the system.
	Hardware Specialist	Leads on researching and assembling the hardware of the system according to the project blueprint.
	Financier	Provides an adequate budget contribution to support the development and progress of the project.
Rotoni, Carlo R.	*Documentation Specialist*	Maintains comprehensive project documentation to facilitate effective project management and knowledge sharing.
	Software Engineer Assistant	Assisting on the researching and assembling of the hardware of the system according to the project blueprint.
	Hardware Specialist Assistant	Assisting on designing and developing the necessary hardware for the system.
	Financier	Provides an adequate budget contribution to support the development and progress of the project.
Agcaoili, Raven D.	*Documentation Specialist*	Maintains comprehensive project documentation to facilitate effective project management and knowledge sharing.
	Financier	Provides an adequate budget contribution to support the development and progress of the project.

Appendix Q:

Curriculum Vitae

ROME ANGELO GAGABI

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) M

Address: **2117 Old Balagtas Street Pandacan Manila / Brgy 833 Manila**

Email Address: **rome.angelo15@gmail.com**

Cellular No.: **+639774677460**

CAREER OBJECTIVE

To establish a career in the engineering field where I can exhibit the educational outcomes that I gained from the Computer Engineering program in Technological Institute of the Philippines (TIP), and I aspire to become a Cyber Security Specialist in the future by applying technical skills, use of modern techniques and tools of the computing practice in complex activities.

DESIGN PROJECT COMPLETED/ RESEARCH

Random Pop-Up
This project is an educational system that uses MySQL, PHP, HTML, CSS, JavaScript, jQuery, and Sublime Text that permits the students and the instructors to interact with each other and to make sure that the students learn from the instructor's discussion. The website will present a popup with questions that the students need to answer. The instructor can create a question that will randomly pop up and will notify the students. The instructor will have the results of the student's response. Moreover, this is to make sure that the students are listening and paying attention while the instructors are discussing. This system is composed of multiple programming languages such as PHP, jQuery, and Sublime Text which makes the website more effective and user-friendly to anyone in the local region. To verify if the system's functionalities have been met, we asked our users to evaluate the system by answering a set of questions that relates to the system that we are making. Upon obtaining the results of the survey, users find that PopUp has met the objectives, and functionalities were supplied accordingly.

ON-THE-JOB TRAINING/ PRACTICUM/ INTERNSHIP EXPERIENCE

- Language Assessor
  TEST, Inc. PH
  197 Salcedo, Legazpi Village, Makati, Metro M
  October 31, 2022 - February 01, 2023

  Language Assessor

  - Efficiently evaluated language skills,met quotas, and collaboratedwithin a team, showcasing adaptable timemanagement.
  - Provided valuable feedback, fostering language growthwhile juggling assessment and teaching.
  - Enhanced engagementthrough clear communication, fostering an environmentforimproved language skills.
  - Achieved targets by effective collaboration, seeking guidance, andmaintaining quality assessment and feedback.

KNOWLEDGE, SKILLS AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills and attitudes) necessary to the practice of the computing profession:

- *Analyze complex problems and identify and define the computing requirements appropriate for solution.*
- *Use modern techniques and tools of the computing practice in complex activities.*
- *Understand professional, ethical, legal, security and social issues and responsibilities relevant to professional computing.*

REINIER MARISCOTES

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) M

Address: **182 San Pedro St. Balut, Tondo Manila**

Email Address: **mrdmariscotes@tip.edu.ph**

Cellular No.: **+639218200430**

CAREER OBJECTIVE

Highly motivated and driven computer engineering student seeking an internship/entry-level position to apply and expand my technical knowledge, problem-solving skills, and passion for innovation in the field. With a solid foundation in computer science principles and a keen interest in cutting-edge technologies, I aim to contribute to a dynamic team, collaborate on challenging projects, and gain hands-on experience in software development, hardware design, and system analysis. Through my dedication, adaptability, and continuous learning mindset, I aspire to make a positive impact in the industry while furthering my own professional growth.

DESIGN PROJECTS COMPLETED/ RESEARCHES

Palengk-E
Palengk-e is a mobile application that revolutionizes the way farmers and buyers connect in local markets. It serves as a platform for farmers to showcase their fresh produce directly to buyers, eliminating intermediaries and ensuring fair prices. With Palengk-E, buyers gain access to high-quality, locally sourced goods while supporting sustainable agriculture and empowering local farming communities.

dotBean
Optimizing Storage Conditions for Preserving Quality and Freshness of Coffee Beans

IoT Air Quality Monitoring System
I have experience in designing and implementing IoT solutions, with a focus on air quality monitoring systems. My expertise includes integrating various sensors to measure pollutants such as CO2, CO, PM, and others. I am skilled in developing embedded systems using microcontrollers like Arduino and Raspberry Pi, enabling real-time data processing and communication, I have contributed to creating web and mobile interfaces for users to access air quality data. My work emphasizes environmental awareness and data-driven decision-making for improving air quality in diverse settings.

Embedded System
Experienced embedded systems engineer with a proven track record in hardware design, firmware development, and system integration. Proficient with Arduino, Raspberry Pi, and various microcontrollers. Skilled in sensor interfacing, programming (C, C++, Python), and optimizing system performance. Well-versed in real-time operating systems (RTOS) for precision. Committed to staying current with emerging technologies.

Zombie Game
I developed a zombie-themed game using 001 Game Creator, showcasing my game development skills. The game went through three stages: concept and story creation, implementation with graphics and interactivity, and testing and optimization.

Mechanical Non-invasive Ventilator
Developed a mechanical non-invasive ventilator with integrated temperature, humidity, air flow sensor, and pulse sensor using Arduino. Demonstrated expertise in designing and implementing a comprehensive healthcare solution with real-time monitoring capabilities.

ON-THE-JOB TRAINING/ PRACTICUM/ INTERNSHIP EXPERIENCES

- Graphic Designer
  Freelancer
  Online
  March 02, 2021 - November 18, 2022
  As a freelance graphic designer, I have successfully collaborated with numerous clients to deliver visually stunning and impactful designs. With a strong command of design principles, color theory, and typography, I have created compelling logos, brand identities, marketing collateral, and digital assets. By combining my creativity with a deep understanding of client requirements and target audiences, I have consistently produced designs that effectively co

- Social Media Manager
  Freelancer
  Online
  March 08, 2021 - November 18, 2022
  As a social media manager, my primary responsibility is to develop and implement effective social media strategies to enhance brand visibility, engage with the target audience, and drive business growth. In my freelance work, I have successfully managed social media accounts for various clients, creating and curating compelling content that aligns with their brand image and objectives. I am experienced in conducting market research and utilizing analytics tools to identify trends, track performance, and make data-driven decisions to optimize social media campaigns

KNOWLEDGE, SKILLS AND ATTITUDE

- **Problem Analysis & Solution Definition**: Expertise in analyzing complex problems to define computing requirements for effective solutions.
- **Modern Computing Techniques & Tools**: Proficient in utilizing modern techniques and tools, emphasizing Next.js and React.js for dynamic front-end development.
- **Front-End Development Skills**: Advanced skills in front-end development, incorporating Next.js and React.js for creating responsive and interactive web applications.
- **Professional Ethics & Legal Awareness**: Strong understanding of professional, ethical, legal, security, and social issues relevant to the computing profession.
- **Cybersecurity Practices**: Specialized knowledge in cybersecurity practices, ensuring secure computing environments and applications.
- **Embedded Systems Development**: Skilled in developing efficient embedded systems for various applications.
- **User-Friendly Front-End Solutions**: Adept at creating intuitive and user-friendly front-end interfaces using React.js and Next.js.
- **C++ Proficiency**: Strong foundation in C++ programming, enabling efficient software development and system-level programming.
- **Versatile Professional**: Equipped with a comprehensive skill set, including cybersecurity and embedded systems, ready to address the evolving challenges of the computing industry dynamically.

LEADERSHIP ACTIVITY

As a dedicated member of our software design group, I excelled as a Project Leader, overseeing and coordinating front-end development efforts. My primary focus was on conceptualizing, designing, and implementing user-friendly interfaces that seamlessly integrated with back-end functionality.

In addition to leadership, I actively contributed to back-end development tasks, fostering a cohesive and efficient system. This involvement provided valuable insights into the full software development lifecycle, enhancing my understanding of the project''s technical requirements.

Beyond my software expertise, I have embraced a parallel role as a badminton coach. Leveraging my coaching skills, I bring a unique perspective to teamwork, communication, and adaptability. This dual experience reflects my ability to excel in diverse roles, combining technical proficiency with effective leadership and coaching.

SEMINARS AND TRAININGS ATTENDED

- Game Development
  TIP Manila
  June 30, 2022

- Web Development
  TIP Manila
  July 25, 2022

- Leadership Training
  St. Joseph School of Gagalangin
  August 26, 2016

- Freelance Academy
  Online
  February 15, 2021

- Introduction to Azure Machine Learning
  Technoligical Institute of the Philippines
  April 03, 2024

- Introduction to Next JS.
  Technoligical Institute of the Philippines
  March 29, 2024

EXTRA AND CO-CURRICULAR ENGAGEMENTS AND VOLUNTEER WORKS

- St. Joseph Varsity
  Badminton Player
  June 15, 2015 - March 15, 2018

- Computer Engineering Student Society
  Badminton Team Manager/ Player
  January 09, 2023 - May 24, 2023

- Point Blank (Band)
  Lead Guitarist/ Vocalist
  January 09, 2023 - May 24, 2023

- YMCA
  Basic Member
  July 13, 2022 - Present

OTHER SKILLS

Badminton Player
Musician
Excellent Communication Skills
Hardworker
Organized

REFERENCES

Dr. Jennifer Enriquez
Program Chair
Technological Institute of the Philippines
jennifer.enriquez@tip.edu.ph
0925244785

Dr. Jennalyn Mindoro
Assistant Program Chair
Technological Institute of the Philippines- Manila
jmindoro@tip.edu.ph
09088813829

GUILLEN MINERVA ROOT

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) M

Address: **391 Gomez St., 16th ISU, Brgy. Pinagsama Taguig City**

Email Address: **guillenaroot@gmail.com**

Cellular No.: **+639451224484**

CAREER OBJECTIVE

A Computer Engineering student seeking an entry-level position to apply my academic knowledge and technical skills. Eager to learn from experienced professionals and adapt to the company's work environment, while committed to growing into a proficient IT professional.

DESIGN PROJECTS COMPLETED/ RESEARCHES

dotBean: A Smart-storing Device for Coffee Arabica Bean's Moisture Content Monitoring and Storage Temperature Control (2024)
The dotBean project, in collaboration with the Farm to Cup Philippines Benguet, addresses the storage challenges faced by small-scale coffee farmers in La Trinidad, Benguet due to low temperatures. By developing an IoT-enabled smart storage device and mobile app, dotBean monitors and controls temperature, humidity, and moisture, ensuring optimal storage conditions. This innovative solution prevents mold growth and preserves the beans' quality, flavor, and aroma, protecting their freshness and marketability.

RateUp: A Reviewing Site with Supervised Machine Learning for Text-Review Classification (2022)
The researchers developed an advanced website application using Supervised Machine Learning and a custom Support Vector Machine algorithm. Hosted on Scalingo, a cloud platform, the user-friendly web application analyzes customer feedback. Emphasizing sentiment analysis enables businesses to distinguish authentic reviews, thereby elevating service standards. The application, featuring a review page and restaurant information, aims to enhance customer satisfaction amidst the heightened concerns about falsified reviews.

Journey through Human Anatomy: A Strategic Web-based Platformer on the Study of Human Body Systems (2020)
Educational institutions leverage technological progress to embrace innovative teaching methods, such as web-based platformer games. These games, accessible through web browsers, provide an interactive learning experience, especially in Grade 11 STEM's Human Body Systems. Research suggests video games serve as effective educational tools, offering dynamic and engaging learning environments compared to traditional classrooms. The study not only assesses learning outcomes but also gauges user acceptance of the web-based platformer game.

KNOWLEDGE, SKILLS AND ATTITUDE

Soft Skills·

- Problem-Solving·
- Time Management·
- Fast Learner·
- Adaptable·
- Goal Setting·
- Emotional Intelligence

Technical Skills·

- Computer NetworkingCybersecurity Management·
- Data Science & Machine Learning: Python Programming·
- Full-stack Website Development
  - ASP.net Core Framework and Blazor-Syncfusion as UI Web-Framework and Blazor-CSS with C# Programming with MS SQL DB
  - Next.js Framework and Tailwind-CSS with JS Programming with Firebase DB
- Photo-Video Editing
  - Adobe Photoshop & Illustrator
  - Davinci Resolve

LEADERSHIP ACTIVITIES

- **Vice President for Internal Affairs,** *Computer Engineering Student Society (August - October 2023)*
- **Public Relations Officer**, *Institute of Computer Engineers of the Philippines Student Edition – National Capital Region Chapter (S.Y. 2022 – 2023)*
- **Vice President for External Affairs**, *Computer Engineering Student Society (S.Y. 2022 – 2023)*
- **Treasurer**, *Computer Engineering Student Society (S.Y. 2021 – 2022)*

SEMINARS AND TRAININGS ATTENDED

- Enhancing the Capacity of National IP Offices to Use Artificial Intelligence: Introductory Workshop and Mentoring Program
  Multi Purpose Hall, IPOPHL
  April 26, 2024

- AIoT+: Unleashing Machine Learning in IoT Era
  T.I.P. Manila Casal Seminar Room
  July 11, 2023

- AgriThink!: Developing Agricultural Solutions for a Sustainable Agriculture Industry
  T.I.P. Manila Arlegui Seminar Room
  June 22, 2023

- Sustainability thru Smart Factory Operations: An IoT Application and Implementation and Implementation via Zoom
  ZOOM
  April 22, 2023

- ReENGINEER+ 2021: Empowering Engineer Leaders of Tomorrow (3 Days)
  ZOOM
  November 20, 2021

- Cyber Security: Are You at Risk?
  ZOOM
  November 20, 2020

EXTRA AND CO-CURRICULAR ENGAGEMENTS AND VOLUNTEER WORKS

- Edustria High School by FEU & T.I.P
  Facilitator
  June 03, 2023

- T.I.P. Manila CpE Department & CESO COP
  Facilitator
  February 11, 2023

- Edustria High School by FEU & T.I.P.
  Facilitator
  November 26, 2022

- T.I.P. Manila CpE Department & CESO COP
  Facilitator
  August 19, 2022 - September 06, 2022

REFERENCES

Dr. Jay-ar Lalata, PCpE
Organization Adviser
ICPEP.SE NCR Chapter
jlalata@icpep.org.ph
09369497501

Engr. Yuneza Claire Mortos
Faculty
T.I.P. Manila CpE Department
mortosycy.cpe@tip.edu.ph
09999255056

CARLO ROTONI

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) M

Address: **1642-D Gerardo Tuazon St, Sampaloc Manila**

Email Address: **rotonicarl@gmail.com**

Cellular No.: **+639987224047**

CAREER OBJECTIVE

To pursue job opportunities in a competitive environment where I can apply my expertise in computer networking and cybersecurity in a dynamic and challenging environment. Committed to leveraging my skills and knowledge to propel the company forward, consistently pushing beyond established limitations and making a meaningful impact on the team.

DESIGN PROJECTS COMPLETED/ RESEARCHES

Barang-Gabay: QR Code
QR Code based E-Governance solution which provides an online systematic approach in accessing and managing barangay services and transactions. It manages the residents profile and gives access for effective planning and management of each barangay in each city.

RateUp: A Reviewing Site with Supervised Machine Learning for Text-Review Classification
Reputable consumer feedback is essential for evaluating the quality of services provided, which is why software with sentiment analysis capabilities has been developed utilizing Supervised Machine Learning (SVM) in particular.

eTapon
Develop a smart drop-off container for plastic bottles that uses internet connectivity to enhance garbage disposal. This incorporates functions including sensor-triggered alarms, automated lids, and real-time information to a website for monitoring.

dotBean
A Smart-storing Device for Coffea arabica's Moisture Content Monitoring and Temperature-Humidity Control

ON-THE-JOB TRAINING/ PRACTICUM/ INTERNSHIP EXPERIENCE

- Intern
  Legaspi Billders Construction
  776 San Sebastian Street, Quiapo, Manila
  February 28, 2024 - May 09, 2024

  - Stay up-to-date with the latest technologies and incorporate new technology into existing units.
  - Responding to computer-related issues and providing technical support to staff members.
  - Responsible for documenting their work, including writing technical reports, creating user manuals, or updating project documentation.
  - Assisting in the setup, maintenance, and troubleshooting of computer networks, servers, and other IT infrastructure components.
  - Installing and inspecting LAN and CCTV cable equipment for setting up internet networks and security cameras.
  - Working on documentation from Microsoft Excel and Word.

KNOWLEDGE, SKILLS AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills and attitudes) necessary to the practice of the computing profession:

- Knowledge in Cybersecurity
- Knowledge in IntelliJ IDEA
- Skills in front-end and back-end development

SEMINARS AND TRAININGS ATTENDED

- AgriThink!: Developing Agricultural Solutions for an Agriculture Industry
  TIP-MANILA
  June 12, 2023

- Getting Started with Python: A Beginner's Guide to Programming
  via Zoom
  January 28, 2023

- Cyber Security: Are You at Risk?
  via Zoom
  November 29, 2023

- CCNAv7:Introduction to Networks
  Cisco Networking Academy
  August 11, 2022

- CCNAv7: Switching, Routing, and Wireless Essentials
  Cisco Networking Academy
  November 09, 2022

- CCNAv7: Enterprise Networking, Security, and Automation
  Cisco Networking Academy
  May 22, 2023

- CyberOps Associate
  Cisco Networking Academy
  December 14, 2023

OTHER SKILLS

Soft Skills

- Teamwork
- Time Management
- Adaptability
- Decision-Making

Technical Skills

- Computer Networking
- Cybersecurity Management
- Basic knowledge on Python, C++, JavaScript, PHP, and CSS
- Basic Operating System Fundamentals and microcontroller programming
- Basic knowledge on Mobile Application development
- Proficient in Microsoft Office Applications (Word, Excel. and PowerPoint) and Google Applications

REFERENCE

Jennalyn Mindoro
Assistant Program Chair, Computer Engineering Department
Technological Institute of the Philippines
jmindoro.cpe@tip.edu.ph
09088132829

RAVEN AGCAOILI
COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) M

Address: **3422 Dona Imelda Quezon City**

Email Address: **agcaoiliraven7@gmail.com**

Cellular No.: **+630966417057**

CAREER OBJECTIVE

To achieve a better and successful career in engineering where I can demonstrate the learning outcomes of the Computer Engineering Program of the Technological Institute of the Philippines (TIP) and by applying the graduate attributes being taught by the institution. I am hoping to build my career and utilize my skills to further work towards personal and professional development and contribute towards the prosperity of the organization.

DESIGN PROJECTS COMPLETED/ RESEARCHES

Deep Learning-Based Gamefowl's Dropping Recognition : A Classification Approach for Battle Preparedness An approved software proposal in System Analysis and Design (2020)
Deep learning based gamefowl dropping recognition system that aims to recognize a dropping from a peaked gamefowl.

dotBean: A Smart-storing Device for Green Coffee Bean Arabica?s Moisture Content Monitoring and Storage Temperature Control
A smart storing device for moisture content in maintaining the quality of coffee beans.

KNOWLEDGE, SKILLS AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills and attitudes) necessary to the practice of the computing profession:

- *Analyze complex problems and identify and define the computing requirements appropriate for solution.*
- *Use modern techniques and tools of the computing practice in complex activities.*
- *Understand professional, ethical, legal, security and social issues and responsibilities relevant to professional computing.*

SEMINARS AND TRAININGS ATTENDED

- Stay Connected: A Deep Dive to the Evolving World of Cloud Computing
  Technological Institute of the Philippines - Manila
  December 15, 2020

- IT Project Management: A Practical Approach
  Technological Institute of the Philippines - Manila
  December 12, 2020

- Block Chain: A Security for Everyone
  Technological Institute of the Philippines - Manila
  January 16, 2021

OTHER SKILLS

- HTML
- CSS
- JAVA
- C++
- Figma
- Python
- UI Design
- Assembly Language

REFERENCES

Debbie Anne Orantoy
Helpdesk Technician 1
Stefanini Group
debbieorantoy@gmail.com
09287516949

Engr. Jennifer Enriquez
Computer Engineering Department Chairperson
Technological Institute of the Philippines
jennifer.enriquez@tip.edu.ph
09225244785