Mohammed V University - Rabat
National School of Computer Science
and Systems Analysis

Summer Internship 1 Year Report

Major :

IDSIT

Subject :

The use of MARS questionnaire to evaluate mobile Oceanography apps

---

Supervised by :

Pr.——

Pr.——

Defensed By :

BOUZID Abdelfattah

Defensed dd/mm/yy in front of the Jury Composed of :

Mr.———-

Mr.———-

Academic Year 2022-2023

Dedication

“

We dedicate this humble work with sincerity and pride : To our dear parents, our families, our teachers, our friends and to all the people who have supported us along our journey.

It is creditable to your love, your encouragement, the trust you have given us that we are able to accomplish this work today. I hope you will find in this work the testimony of our deep gratitude. No dedication can express our feelings towards you for your moral support.

Thank you for all the moments we spend with you and for the help you have offered us.

Thank you...

”

BOUZID Abdelfattah

Acknowledgements

Before any development on this professional experience, it seems appropriate to start by thanking those who taught us a lot during this semester, and even those who were kind enough to make this project a very fruitful experience.

Regarding this work, we would like to express our deep gratitude to our teacher **Mr. RACHAD Taoufik** for bringing all the necessary help to achieve the project goals during the Android course.
We would like to avail ourselves of this opportunity to thank our academic supervisors **Mr.RADOUANE** and **Mr.KOBBANE** for their availability and valuable advice.

Please find here the expression of our most profound respect.

Abstract

This project helps the farmers to do the humidification control and monitor the condition of hydroponic plants in greenhouses in real-time. When it comes to watching the hydroponic plants in greenhouses, the farmers usually experience difficulties because they still do it manually. Activities such as checking the temperature, air humidity, and also water quality in hydroponic plants by coming directly to the greenhouse are still ineffective. Therefore this project aims to make a smart greenhouse prototype for hydroponic plants. Smart greenhouse hardware was built based on the Arduino microcontroller, DHT11 sensor, pH sensor, TDS, DS18b20 temperature, ultrasonic, and esp8266 wifi module. The monitoring system features information on water quality from hydroponic plants and the ability to record farming activities from planting preparation to mobile app-based harvesting. The test results of smart greenhouse monitoring system can display the hydroponic plant conditions and able to do the humidification control with an upper limit of 35 degrees Celsius because plant can survive with disease under 35 degrees celsius, with small average offset for the sensor, and an average offset of 1.49 from TDS sensors, with temperature of 0.50 and pH of 0.34.

Keywords : Greenhouse, NFT hydroponics, Humidification, Monitoring System, Arduino .

List of Acronyms

IoT *Internet of Things*

UML *Unified Modeling Language*

MVC *Model–View–Controller*

Android | Java

NFT *Non-fungible Token*

List of Figures

Table of Contents

General Introduction

The agricultural production sector is critical, an area that can sustain people's lives. In general, most farmers still use traditional farming systems and are dependent on climate change. Uncertain climate change in recent years has made the timing of harvests difficult for farmers to predict. Prolonged rain increases the risk of pests and diseases, and excessive heat causes the plants to lose a lot of water and wither, resulting in farmers experiencing losses in the form of crop failure. Therefore the existence of a greenhouse in the agricultural world is becoming increasingly important. The greenhouse is a building intended for the cultivation of plants with several variable settings in it to match the growth and development of plants that are cultivated. The greenhouse cultivation system for farmers is relatively new, and thus some of the farmers still traditionally use this system, i.e., the whole work is always done using human labour such as the air humidity monitoring process, temperature and humidification process, and watering plants in the greenhouse building. Hydroponic farming is one of the farming methods in a greenhouse building. Hydroponics is an agricultural system that utilises water as a base for the development of the plant body. The water used is not ordinary, but the water that contains nutrients that can help the process of plant growth. In addition to rain that functions as plant nutrition, several environmental factors must be considered, including the humidity and temperature . To control the environmental factor, a greenhouse building with this hydroponic farming system in it will be perfect. Nutrient Film Technique (NFT) is a hydroponic cultivation model that places the plant roots on shallow layers (of water). The water is circulated and contains nutrients according to plant needs. In the NFT model plants, nutrition is a critical factor, i.e., the dosage of food given must be precise because either the deficiency or excess of nutrients is intoxicating for the plants and eventually will kill them, NFT is better than DFT .

For the hydroponic planting method in the greenhouse, the system monitoring is used to determine the quality of water in hydroponics in the form of information on the levels of nutrients dissolved in water, water pH, water temperature, and water level in a hydroponic water storage container. The monitoring system provides features of crop types, crop cycles, and also harvest prediction to help the farmers recording their farming activities, ranging from planting preparation to the harvest phase.

Chapitre 1

General context of the project

In this chapter, we will give an overview of hydroponic farms and smart greenhouse , the problems that they lead to this technology, and the proposed solution to handle them. Finally, we will talk about the project methodology.

1.1 Introduction

Greenhouse and smart farms have witnessed a remarkable surge in popularity in recent years, fueled by the growing interest in sustainable agriculture and the integration of advanced technologies. With the rise of the ”smart farming” movement and the pressing need for efficient food production, individuals are turning to greenhouse and smart farm systems to take charge of their agricultural endeavors.

Advancements in technology, coupled with the availability of compact and intelligent farming systems, have made it easier than ever for people to cultivate crops in controlled environments. These systems enable precise control over crucial environmental factors such as temperature, humidity, lighting, and nutrient levels. Through automated monitoring and management, individuals can optimize plant growth, improve resource efficiency, and ensure optimal conditions for their crops.

The concept of smart farms goes beyond traditional farming methods by leveraging interconnected devices, sensors, and data analytics. These farms utilize cutting-edge technologies like Internet of Things (IoT) devices, cloud computing, and artificial intelligence (AI) algorithms. By integrating these technologies, farmers can remotely monitor and manage their greenhouse or smart farm operations through a centralized control system or a dedicated mobile application.

The ability to remotely control and monitor greenhouse and smart farms brings numerous benefits. Farmers can receive real-time data on environmental parameters and crop health, allowing them to make informed decisions and promptly address any issues. The application of data analytics and machine learning algorithms further enhances productivity, enabling predictive insights for disease prevention, yield optimization, and resource allocation.

Moreover, smart farms contribute to sustainable agriculture practices by minimizing resource consumption and reducing environmental impact. Precise control over water usage, nutrient distribution, and energy consumption results in more efficient resource utilization, reducing waste and promoting eco-friendly farming methods.

In conclusion, the emergence of greenhouse and smart farms represents a significant shift in agriculture, empowering individuals to grow crops in controlled environments and harness the potential of advanced technologies. Through remote control and monitoring capabilities, farmers can optimize their operations, ensure crop health, and contribute to a more sustainable and secure food production system. The integration of smart technologies paves the way for a future where home farming becomes an accessible and impactful solution for food production and environmental stewardship.

1.2 Presentation of the project

Before jumping into the design and the implementation of the mobile application its challenges. Let’s first have a clear vision of what is hydroponic and smart farms, GreenHouse . And what are the different Materials used in this type of farming nowadays.

1.2.1 smart farms and greenhouse : concept and advantages

Smart farms and greenhouses revolutionize agriculture with advanced technologies. Greenhouses provide controlled environments for optimal plant growth. Hydroponic systems in greenhouses deliver precise nutrients to plants. Monitoring sensors collect real-time data for analysis. Automation and robotics automate tasks and improve efficiency. Smart farming enhances crop yields, resource utilization, and plant health. These innovations create a sustainable and productive future for agriculture.

FIG. 1.1 : GreenHouse

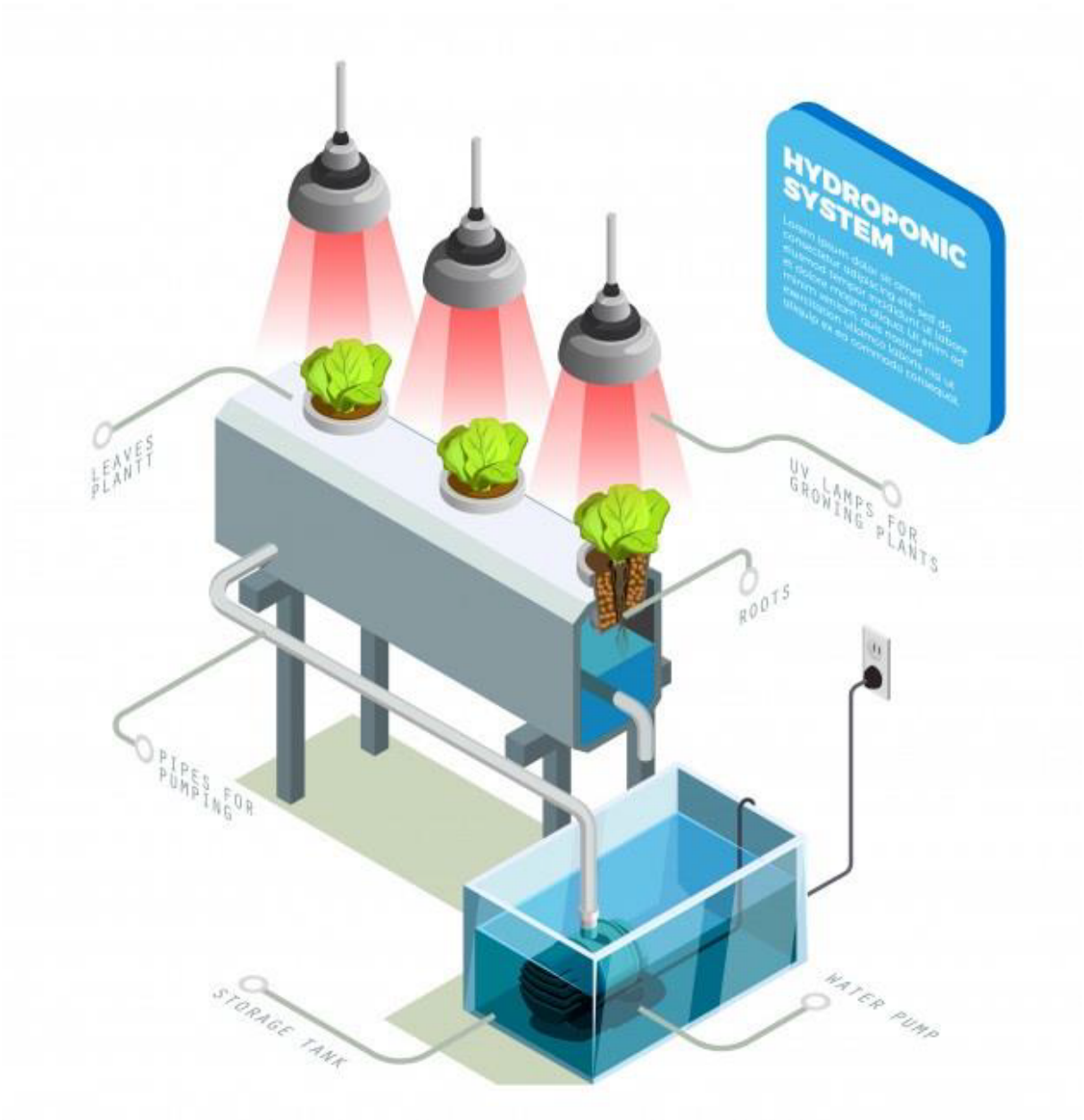

FIG. 1.2 : Hydroponic system

Smart farms and greenhouses offer the ability to control the growing environment, a key advantage of hydroponic agriculture. Precise regulation of temperature, light, and humidity creates optimal conditions for plant growth, leading to higher yields and consistent crop quality. The indoor setup of hydroponic systems enables year-round food production in regions with challenging climates.

Resource efficiency is another benefit of hydroponic agriculture. The closed system minimizes resource loss through evaporation or run-off, and water savings can be substantial compared to traditional soil-based methods. This makes hydroponic agriculture more sustainable, reducing its environmental impact.

However, hydroponic agriculture does come with challenges. It requires more technological complexity and greater technical proficiency than traditional methods. The initial

setup cost can be significant, but it is offset by the numerous benefits and the potential for higher yields in the long run. Smart farms and greenhouses capitalize on these advantages, offering controlled environments and advanced technologies to optimize hydroponic agriculture for sustainable and efficient food production.

1.2.2 Hydroponic system and GreenHouse

To create a hydroponic farm, you will need the following materials :

1. Hydroponic system : There are various hydroponic systems available, including deep water culture, nutrient film technique, ebb and flow, and aeroponic systems.

Fig. 1.3 : hydroponic systems and GreenHouse

2. Growing containers : You will need containers to hold your plants, such as buckets, troughs, or growing trays. Make sure the containers are food-grade and safe for use with plants.

3. Growing medium : This can be a synthetic material such as perlite or rockwool, or a natural material like coconut coir. The growing medium provides support for the roots and helps to retain moisture.

4. Pump and tubing : A water pump is required to circulate the nutrient solution, and tubing is used to transport the water from the pump to the growing containers.

5. Reservoir : A large container is needed to hold the nutrient solution. The reservoir should be located near the growing containers and pump to minimize the distance that the water must travel.

6. Nutrient solution : A special nutrient solution is required for hydroponic farming, as it provides all the necessary elements for plant growth. There are various formulations available, so choose one that is appropriate for the crops you plan to grow.

7. Lighting : Depending on your growing setup, you may need artificial lighting to supplement natural light. LED lights are a popular option for hydroponic farming as they are energy-efficient and provide the right spectrum of light for plant growth.

8. pH and EC meters : These instruments are used to measure the pH and electrical conductivity of the nutrient solution, respectively. Proper nutrient levels are crucial for plant health and growth, so monitoring pH and EC is an important part of hydroponic farming.

9. Air stones and air pumps : These components are used to aerate the water, ensuring that the plants receive adequate oxygen.

10. Net cups : Net cups are small containers that hold the plants and growing medium. They are placed in the growing containers, and the roots grow down into the nutrient solution

1.2.3 Home hydroponic farms

Home hydroponic farms are a type of hydroponic farms designed for indoor or outdoor use in residential settings. They offer a compact, efficient, and convenient way to grow fresh produce year-round, regardless of weather conditions. These farms are tailored to the grower's needs and can be set up in small spaces such as apartments or homes. By using a closed system and recirculated nutrient solution, home hydroponic farms can conserve water and reduce the risk of pests and diseases compared to traditional soil-based gardens.

Fig. 1.4 : Small indoor Hydroponic farm

N.B : In this project, we opt for home hydroponic farms.

1.3 Problematic and motivation

1.3.1 Problematic

The problematic of traditional agriculture and the desire for sustainable food sources has motivated the creation of mobile applications for control and monitoring home hydroponic farms and greenhouse . Hydroponic farming is a method of growing plants without soil, using nutrient-rich water instead. This allows for year-round production, higher yields, and reduced use of water and pesticides. A mobile application allows for easy monitoring and control hydroponic systems and supplies, as well as monitoring and managing of the farm, remotely. The application can provide real-time updates on the health and growth of the plants, enabling the farmer to optimize conditions and make adjustments as needed. This helps to increase efficiency, and productivity and reduce waste, ultimately leading to a more sustainable food production process.

1.3.2 Solution : mobile application GreenHouseTech

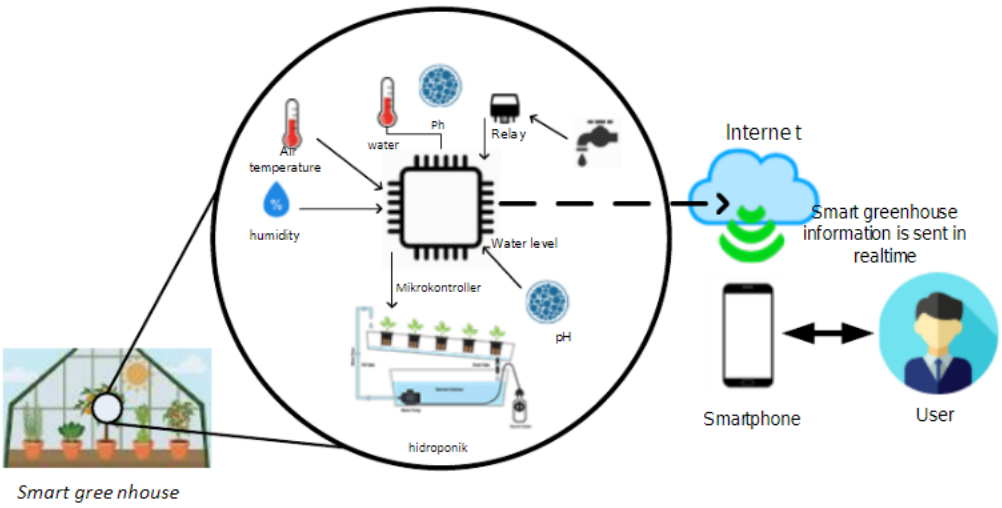

FIG. 1.5 : path of request

The smart greenhouse is intended to provide help to farmers in the form of a humidification control system and a monitoring system that can provide information related to environmental and plants conditions in the greenhouse in the way of water level status, pH, water temperature, and water nutrition through smartphone use. The smart greenhouse divided into two parts, which are hardware and software. The humidification control system works when the humidity in the greenhouse starts to deteriorate. When this happens, the watering pump will start automatically to boost the moisture in the room and set the humidity level as befitting. The scheme of the research design can be seen in Figure 1.5 .

1.3.3 project objective

The goal of this project is to create a mobile application for control and monitoring home hydroponic farms and greenhouse , with a primary focus on providing an efficient and user-friendly management plants,notes and controlling service through the developed platform.

1.4 Project Management

Project Management is the use of knowledge, skills, tools, and techniques to meet the project's requirements. In other words, the goal of project management is to plan and manage a project to successfully achieve its listed goals and deliverables. It involves identifying and managing risks, careful resource management, intelligent budgeting, and clear communication among multiple teams and stakeholders.

Project management is important because it provides leadership and direction to projects. Without project management, a team can be like a ship without a rudder ; moving, but without direction, control, or purpose. Leadership allows team members to do their best to achieve the project aims while respecting deadlines and minimizing costs in order to be efficient.

Fig. 1.6 : Logo "GreenHouseTech".

This allows the customer to make any necessary adjustments to ensure optimal growing conditions for their plants. Additionally, the mobile application can provide reminders for tasks such as watering, fertilizing, and pruning, which helps customers to maintain their hydroponic farms effectively. The mobile application can also offer the ability to receive alerts when any problems arise, such as low nutrient levels or high temperatures. Overall, a mobile application for controlling and monitoring home hydroponic farms can provide

users with a convenient and personalized solution for growing their plants.

1.4.1 Waterfall Methodology

The waterfall model is a sequential software development process that was first introduced in 1970. It is one of the earliest and most widely used models in software project management. In this model, the software development process is divided into distinct stages, each of which must be completed before the next stage can begin. The stages are :

1. Requirements gathering and analysis : In this stage, the software requirements are gathered and analyzed to determine the scope of the project.

2. Design : In this stage, the software design is created, including the architecture and system design.

3. Implementation : In this stage, the software code is written and the software is built.

4. Testing : In this stage, the software is tested to ensure it meets the requirements and performs as expected.

5. Deployment : In this stage, the software is deployed to the production environment and made available to users.

6. Maintenance : In this stage, the software is maintained and updated as necessary.

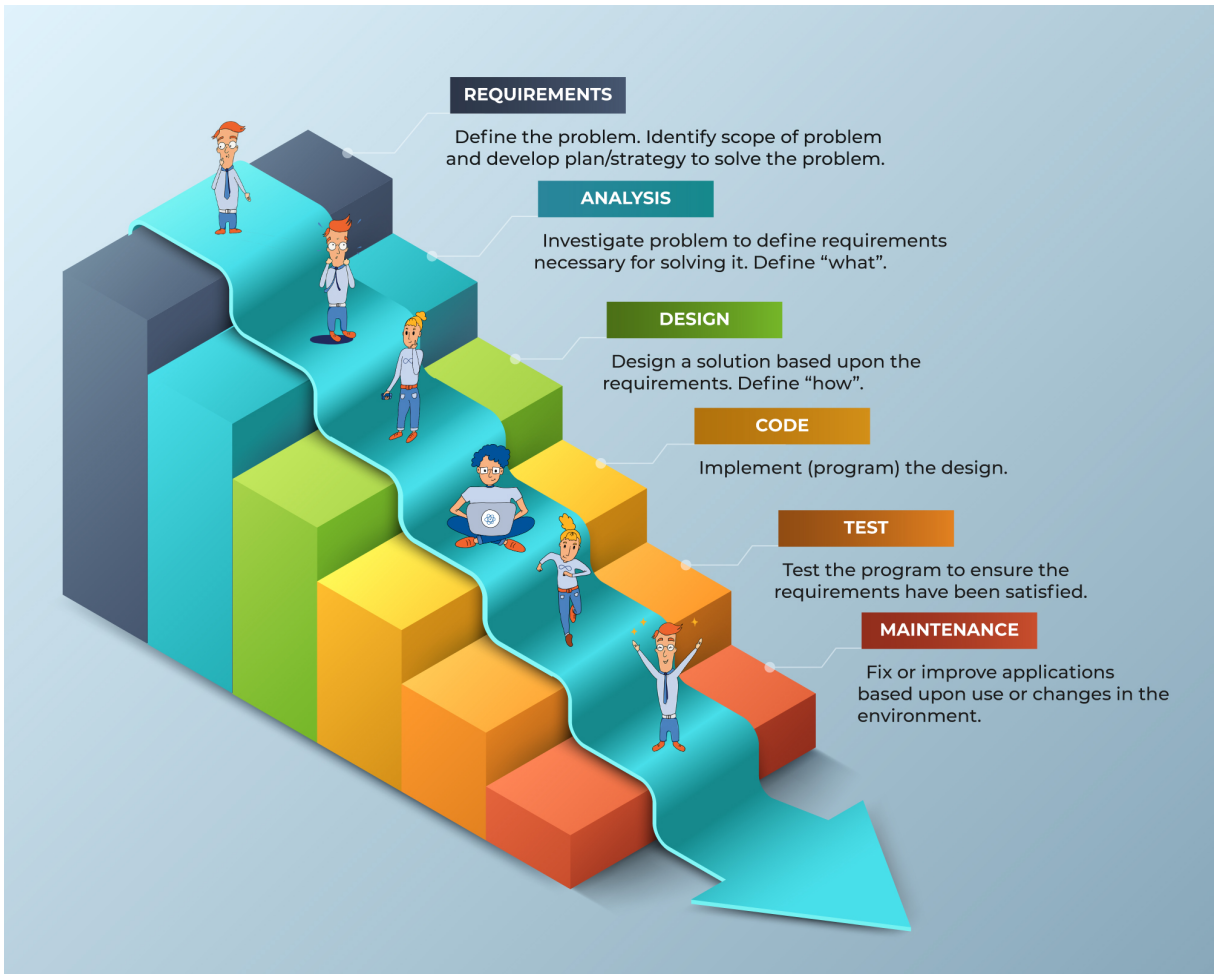

FIG. 1.7 : Stages of waterfall methodology.

1.4.1.1 Advantages of waterfall methodology

The waterfall model has several advantages, including :

1. Clear and well-defined stages : The sequential and linear nature of the waterfall model makes it easy to understand the different stages of the software development process. This makes it easier for team members to understand their responsibilities and for project managers to track progress.

2. Suitable for well-defined projects : The waterfall model is well-suited for projects with well-defined requirements and a clear understanding of the software development process. It provides a structured approach that makes it easier to plan and manage the project.

3. Good documentation : The waterfall model requires documentation at each stage, which makes it easier to maintain the software and provide a clear record of the development process.

4. Cost and time efficiency : The waterfall model can be cost and time efficient because the design, implementation, and testing stages are completed in sequence. This reduces the risk of rework and makes it easier to plan and budget for the project.

5. Predictable outcome : The waterfall model provides a predictable outcome because each stage of the development process is completed before moving on to the next. This makes it easier to anticipate the final outcome and to manage expectations.

6. Good for regulatory compliance : The documentation and clear stages of the waterfall model make it a good choice for projects with regulatory compliance requirements, such as in the healthcare or financial industries.

1.5 Conclusion

In that Chapter, we have seen so far what is a hydroponic and smart farm,greenhouse, especially hydroponic farming and greenhouse. We have seen the problems and challenges that face this technique of agriculture. we saw the need to this technology and its benefits and how the proposed solution will try to satisfy this need . we saw finally the project methodology Waterfall and its advantages.

Chapitre 2

Requirements analysis and Design

This chapter is dedicated to the functional and conceptual study of the project. We will start by capturing the functional and non-functional requirements of the system, then we will analyze these requirements to deduce the actors and the different use cases. Then we will see their sequence diagrams, to ultimately end with the final class diagram.

2.1 Introduction

A design method is a general approach that reflects a philosophy of presentation and system tracking. It provides specific tools for effectively tracking information related to the system. Our choice is theUML which facilitates interactivity with the database using use case diagrams ,class diagram and the diagramm of activity. UML is an object-oriented modeling language that is developed with the purpose of defining a standard notion for modeling applications built using objects. It is used to specify or design software, the model describes classes and use cases seen by the end user of the software. The model produced by object-oriented design is generally an extension of the model from the specification, it enriches the model with so-called technical classes that are not of interest to the end user of the software but only to its designers.

2.2 Functional Analysis

2.2.1 Actor Identification

An actor is an external entity that interacts with the system (operator, remote center, another system...). In response to an actor's action, the system provides a service that meets its need.

In our project i have one main actors :

- The client : refers to the individual who gains advantages from utilizing the mobile application designed to manage and oversee their farms or greenhouse operations.

Fig. 2.1 : The client icon.

2.2.2 Requirements Gathering

The requirements analysis stage is very important since the success of any application depends on the quality of its study. It is necessary to determine the functions expected

by the system well. The requirements are divided into two categories, namely functional and non-functional requirements.

2.2.2.1 Functional requirements

Functional requirements represent the expectations of each actor in the future application. Any conceptual solution must first satisfy functional requirements in order to delimit the functional perimeter of the application and monitor the traceability of requirements during the development phase. They respond to specific points in the specification and are therefore required by the client. They are generally not negotiable. .

Since the **Client** is the main actor, then we will elaborate the functional requirements related to them. Our application allows the client to :

- **Create an account** : The user is able to create an account just by entering their information like : full name, phone number, address, email, and password.

- **Sign-in** : Given that the application handles sensitive personal information, it is essential to identify oneself. To manage the user experience, the session involves retrieving a token from the server that indicates whether the user is still connected or not. Once the user logs out, the token is destroyed. Indeed, for new users, creating an account is mandatory to use the different features of the application.

- **Sign-up** : "Considering the sensitive personal information handled by the application, user identification is crucial. In order to facilitate user experience, the sign-up process includes retrieving a token from the server to indicate the user's active session status. When the user chooses to log out, the token is promptly invalidated. Furthermore, to access the various features of the application, creating an account is mandatory for new users. This ensures that users can fully utilize the functionality provided by the application while maintaining the necessary security measures."

- **Manage account** : The client must be able to modify their personal information. Since the application handles sensitive information provided by the user, and to ensure that the latter is well protected, the account may be deleted at any time by will.

- **Manage favorites** : The user should have the ability to modify their favorite plants in their personal collection. As the application handles sensitive information regarding the user's preferred plants, and to ensure the utmost protection, the user may delete their favorite plants from their collection at any time as per their preference.

- **Manage plants** : The user should have the capability to manage their personal plant collection, including the ability to modify the details of their favorite plants. Since the application handles sensitive information about the client's plants, it is essential to prioritize their protection. Therefore, the user retains the freedom to delete any plants from their collection at any time according to their preferences.

- **Manage notes about plants** : The user should have the capability to manage their notes about each plant in their greenhouse. As the application handles sensitive information regarding the user's plant-related observations and insights, it is crucial to prioritize the protection of this data. Therefore, the user has the freedom to modify or delete their notes about each plant in their greenhouse at any time, allowing them to personalize and manage their collection according to their preferences.
- **Track and Monitor the hydroponic system and greenhouse** : GreenHouseTech allows the client to remotely monitor, control, and manage a farm. This would involve creating a user-friendly interface that displays real-time data about the farm and its operations, as well as providing tools for controlling various systems and processes on the farm. This can include monitoring soil and weather conditions, and controlling irrigation systems.
- **Control the humidition system in the hydroponic system and greenhouse** : GreenHouseTech allows the client to remotely monitor, control, and manage a farm. This would involve creating a user-friendly interface that displays real-time data about the farm and its operations, as well as providing tools for controlling various systems and processes on the farm. This can include monitoring soil and weather conditions, and controlling irrigation systems.

2.2.2.2 Non-functional requirements

A non-functional requirement is a restriction or constraint that affects a system service, such as environmental and implementation constraints and performance requirements.

- **Security** : Given the sensitivity of the data managed by the application, it must be secured ; that's why we need to apply an authentication layer. Additionally, sessions are used to ensure a level of security control over the user experience, these sessions use a token obtained from the backend server. Once the token is destroyed, the session ends and the user must log in again to ensure the protection of sensitive user data, especially payment information.
- **Performance** : To ensure a fast and responsive user experience and handle large amounts of data efficiently.
- **Usability** : To provide a user-friendly interface that is intuitive and easy to navigate.
- **Reliability** : To ensure the system is available and functional at all times, with minimal downtime.
- **Compatibility** : To ensure compatibility with various devices and web browsers.

2.2.3 Requirements analysis : Use Case Diagram

In the UML, a use case diagram can summarize the details of your system's users (also called actors) and their interactions with the system. This is modeled by the following

use case diagram :

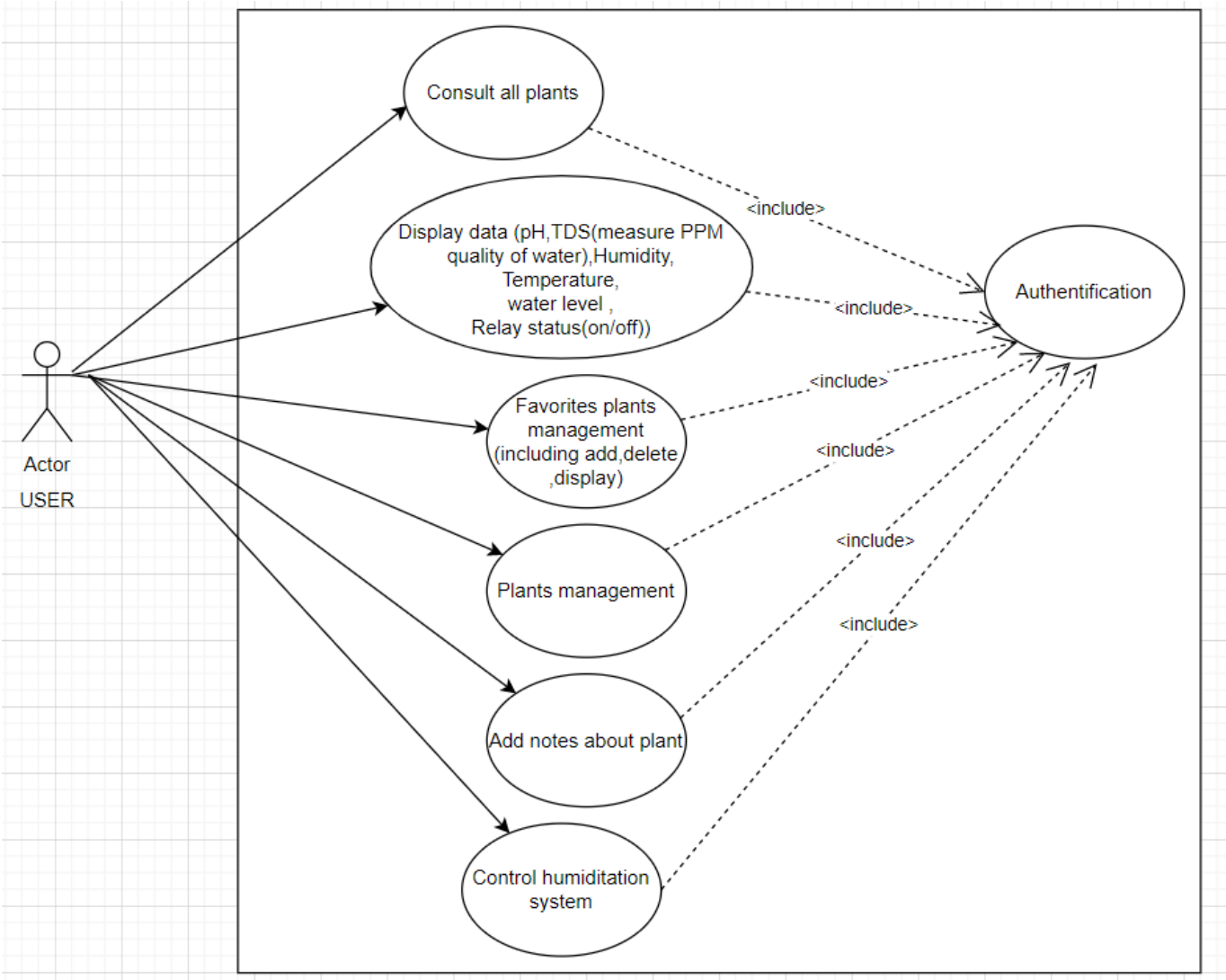

Fig. 2.2 : Global use case diagram.

2.3 Design

2.3.1 Architectural Design of the prototype

2.3.1.1 Green house ,Hydroponic system design

The smart greenhouse is intended to provide help to farmers in the form of a humidification control system and a monitoring system that can provide information related to environmental and plants conditions in the greenhouse in the way of water level status, pH, water temperature, and water nutrition through smartphone use. The smart greenhouse divided into two parts, which are hardware and software. The humidification control system works when the humidity in the greenhouse starts to deteriorate. When this happens, the watering pump will start automatically to boost the moisture in the

room and set the humidity level as befitting. The scheme of the research design can be seen in Figure 2.3 .

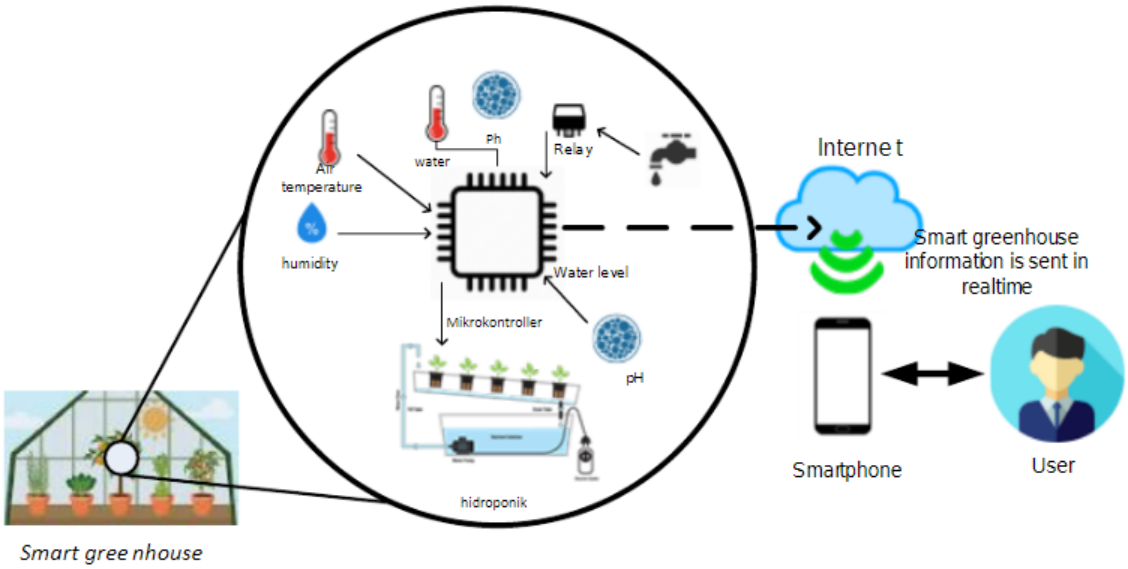

FIG. 2.3 : design GreenHouse.

2.3.1.2 Hardware Design

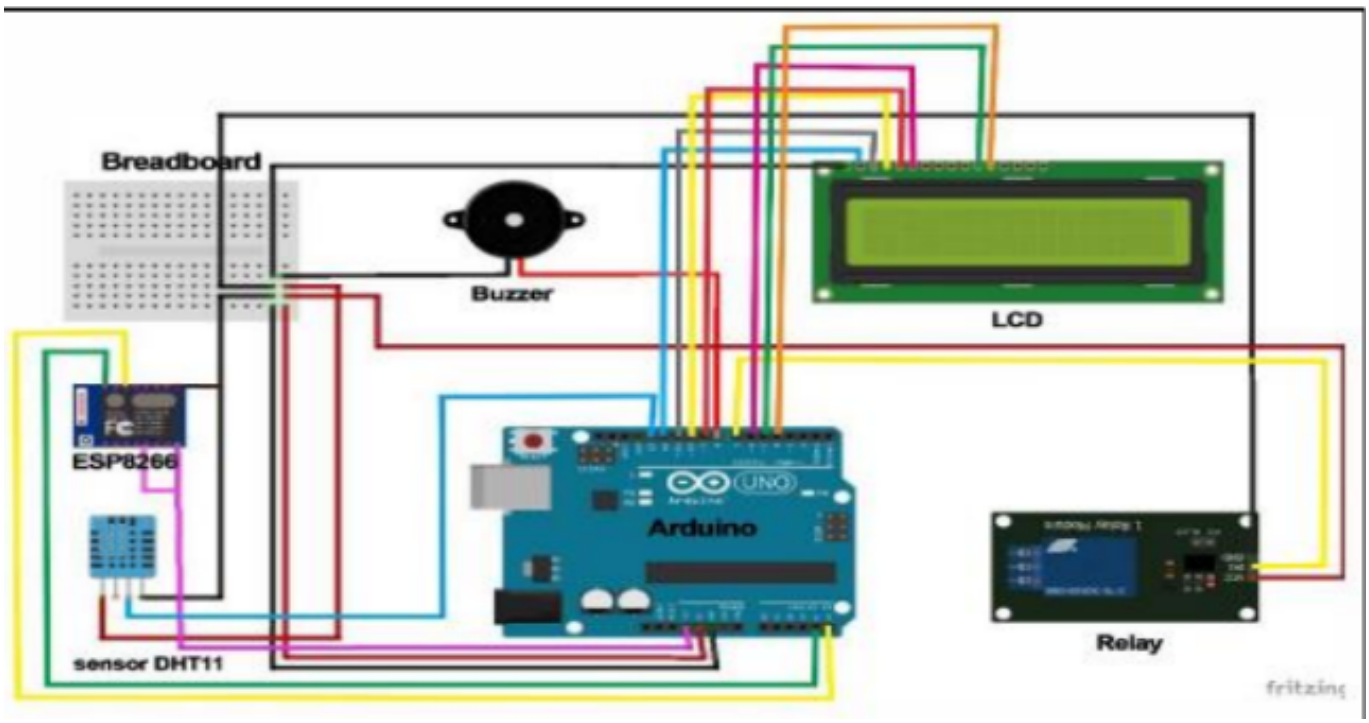

FIG. 2.4 : Hardware Design1.

Figure 2.4 shows the schematic design of the hardware that is used to measure the temperature, humidity and humidification control in a greenhouse. This particular hardware

is called DHT11 Sensor with humidity measurement range 20-90 RH and temperature 0-50 Celcius degrees. It is used to measure temperature and humidity, relay, pump, buzzer, LCD screen and Arduino R3 as a microcontroller.

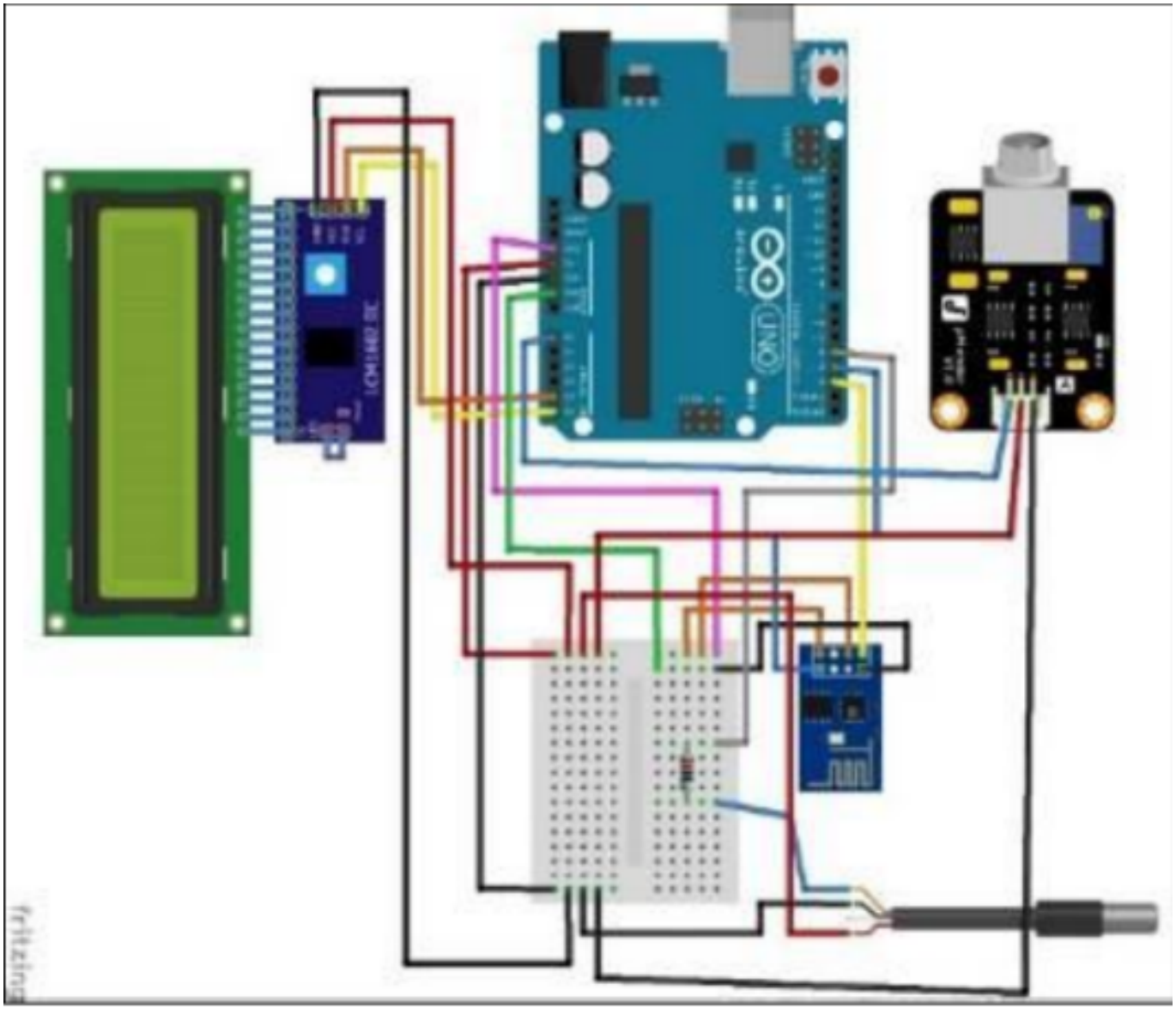

Fig. 2.5 : Hardware Design2.

Figure 2.5 Schematic of a hydroponic monitoring system Figure 3 shows a schematic of a hydroponic monitoring system that is used to measure water quality in a hydroponic water reservoir. It uses hardware such as DF Robot analogue water pH sensor kit, DS18b20 water temperature sensor, TDS meter analogue DF Robot sensor, ultrasonic sensor, esp8266 wifi module, LCD screen, and Arduino as a microcontroller.

2.3.1.3 Program Flowchart

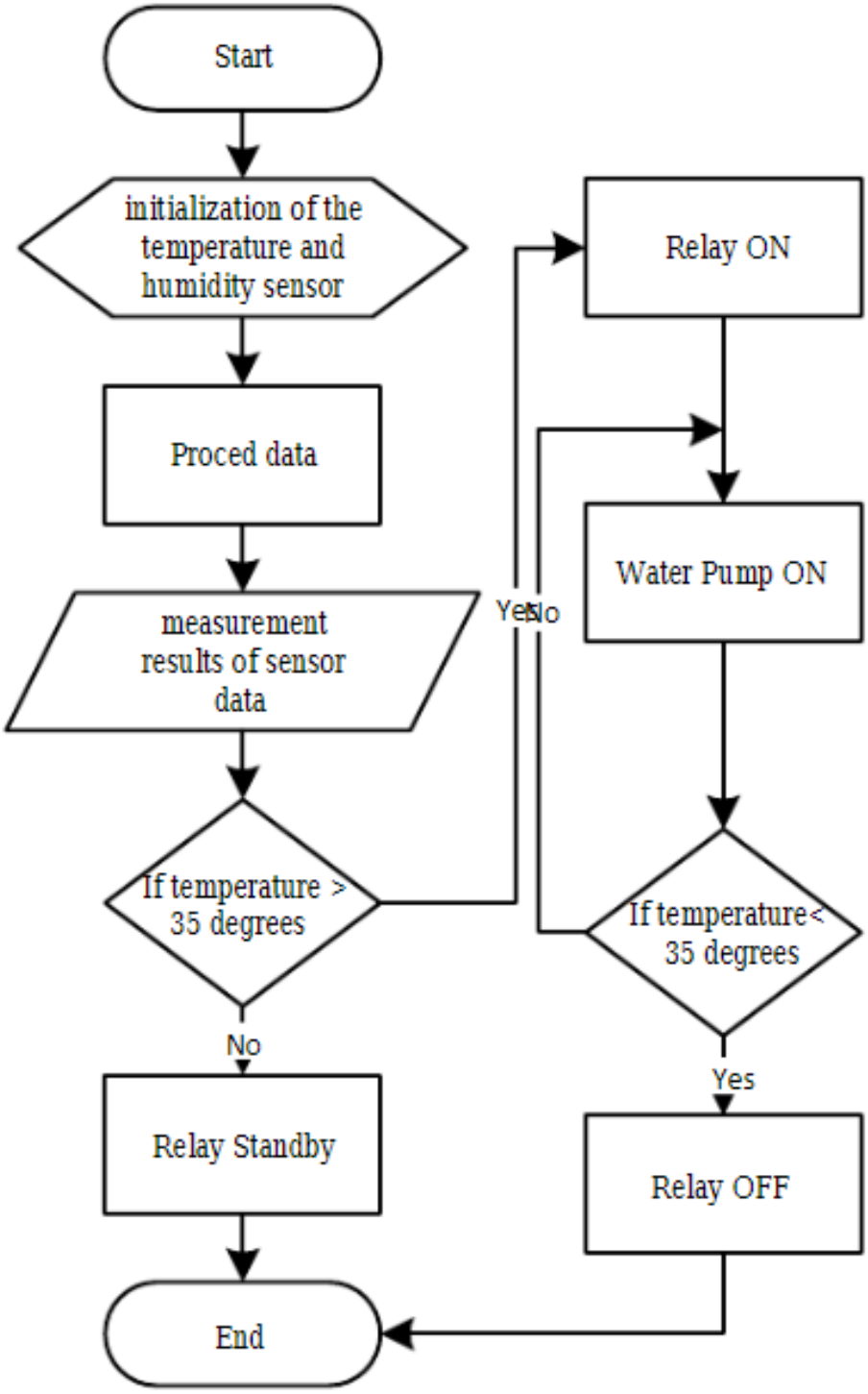

Fig. 2.6 : Flowchart of the humidification control program.

Figure 2.6 shows the humidification control process of the greenhouse. The process begins with the initialisation of the temperature and humidity sensor of the air, then proceed with the process of sensor data reading by the microcontroller. The measurement results of sensor data that have been processed by the microcontroller then displayed

on the LCD screen. The next process is determined when the temperature sensor result shows an exceeding result of a predetermined threshold. If the temperature is above 35 degrees, the microcontroller will command the relay to start the water pump so that the greenhouse room temperature can be adjusted back to normal. This humidification process will continue until the room temperature returns below 35 degrees.

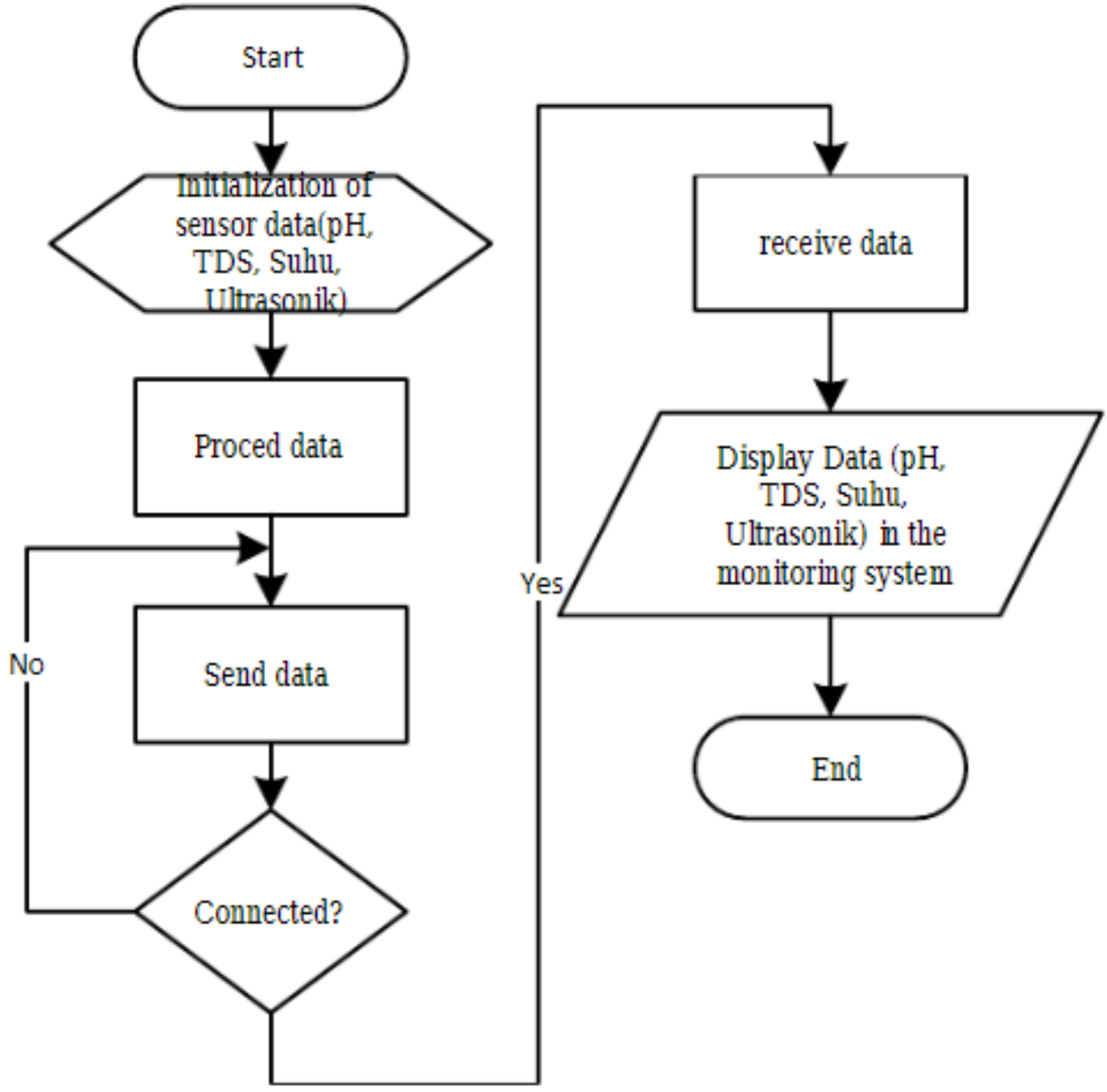

Fig. 2.7 : Flowchart of a hydroponic monitoring program.

Figure 2.7 shows the flow of the process of sensor data distribution up until the sensor data display in the water monitoring system in hydroponics. The process begins with the initialisation of sensor data and then proceed with the process of sensor data reading by the microcontroller. After the sensor data reading process is complete, proceed with

the sensor data distribution process using the wifi module, followed by a condition of connection checkup, if it is connected, the monitoring system will receive data from the sensor reading made by the microcontroller, but if it is not connected, the process of sensor data distribution will repeat again. After the sensor data is received, this data will be directly displayed in the monitoring system in real-time according to the sensor readings.

2.3.1.4 Architecture of my prototype.

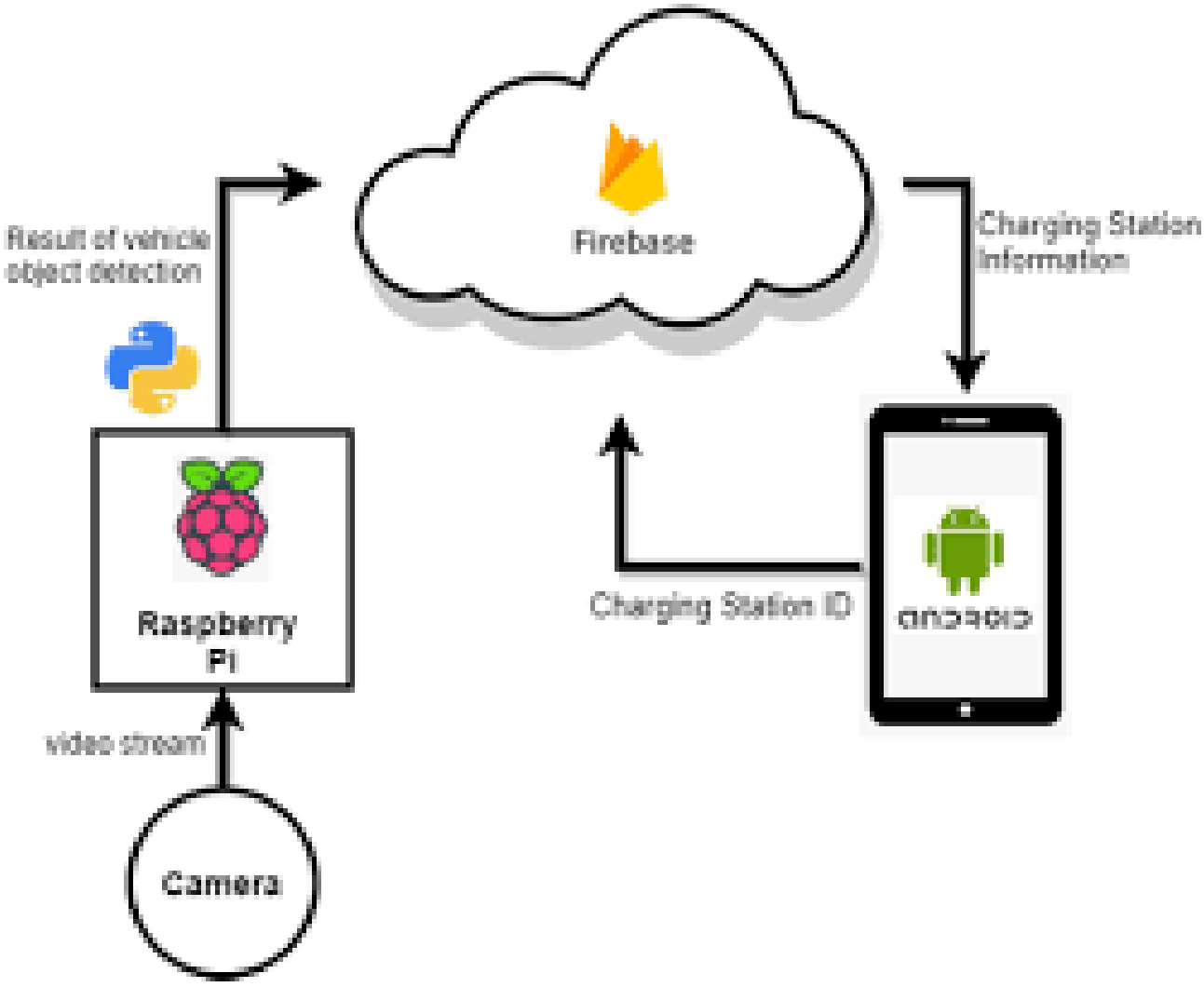

Fig. 2.8 : path of the request between IOT and mobile app.

2.3.2 Architectural Design of the Application

2.3.3 Physical Architecture

In order to thoroughly detail the architecture of our solution, it is recommended to have a general view of the different existing architectural types. In fact, the main architectures to know in this paragraph are three :

Two-tier architecture : consists of two elements, a client and a server, where the tier refers not to a physical entity but to a logical one. It is distinguished by client/server systems in which the client interacts with the interfaces and requests resources and the server directly provides them, using the resources it has available.

Three-tier architecture : As its name suggests, it consists of three elements that are developed and maintained independently on different platforms.

- The client : the resource requester.
- The application server (middleware) : the server responsible for providing the resource but calling on another server.
- The secondary server (usually a database server) : providing a service to the first server.

N-tier Architecture : N-tier architecture refers to the structure of a software application divided into multiple levels. One level is a layer of the application that operates on its own infrastructure or server, where the functions of presentation, processing, and data management are separated both logically and physically.

For the development of our application, we chose a 3-tier architecture which is a client-server implementation that uses a client tier, application tier, and data tier to organize app design.

- **Client tier** : This component interacts with the mobile application through a custom client application.
- **Application tier** : This component provides the runtime environment for the Android application.
- **Data tier** : This component stores the application's data and it is usually accessed via a persistence layer. In this project, we opt for Firebase .

These components interact with each other over a network, typically the internet. The application server communicates with the database to retrieve and store data.

2.3.4 Logical Architecture

2.3.4.1 MVC Architecture

Architecture MVC (Model-View-Controller) : is a model separating three components of main logic namely ; the model which presents all the data as well as the logic of the application, the view that illustrates this data to the user and manages the interactions, and finally the controller which manages the changes between the model and view.

2.3.4.2 Implementing MVC in Android mobile application

Model designs based on MVC architecture follow the MVC design pattern and they separate the application logic from the user interface when designing software. As the name implies MVC pattern has three layers, which are :

- Model : Represents the business layer of the application within **Dao** and **Model**.
- View : Defines the presentation of the application using **XML**.
- Controller : Manages the flow of the application.

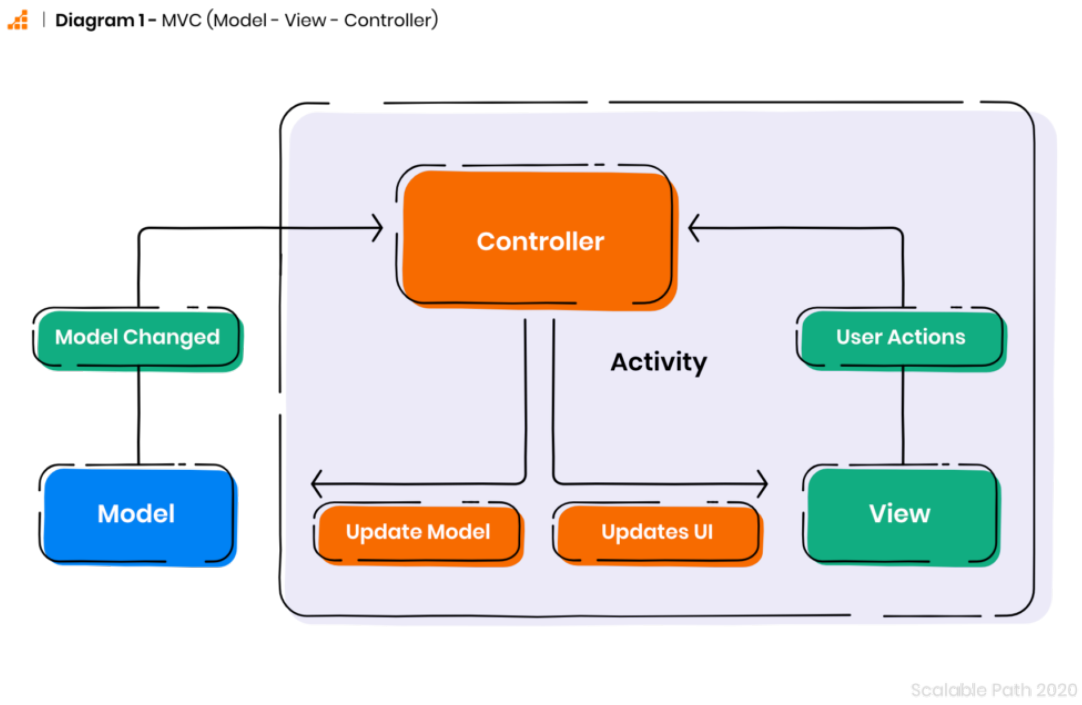

Fig. 2.9 : The process of an action request within the MVC implementation stages.

In the Android application context, the Model consists of simple Java or kotlin classes, the View displays the data via XML tags and the Controller represent a business logic of your code .This separation results in user requests being processed as follows :

1. The user on the client sends a request for a page to the controller present on the server.
2. The controller performs the action of invoking the model, thereby, retrieving the data it needs in response to the request.
3. The controller then gives the retrieved data to the view.
4. The view is rendered and sent back to the client.

2.3.4.3 Dynamic view analysis : Activity diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

- **The user** :
  - process of using GreenHouseTech app :

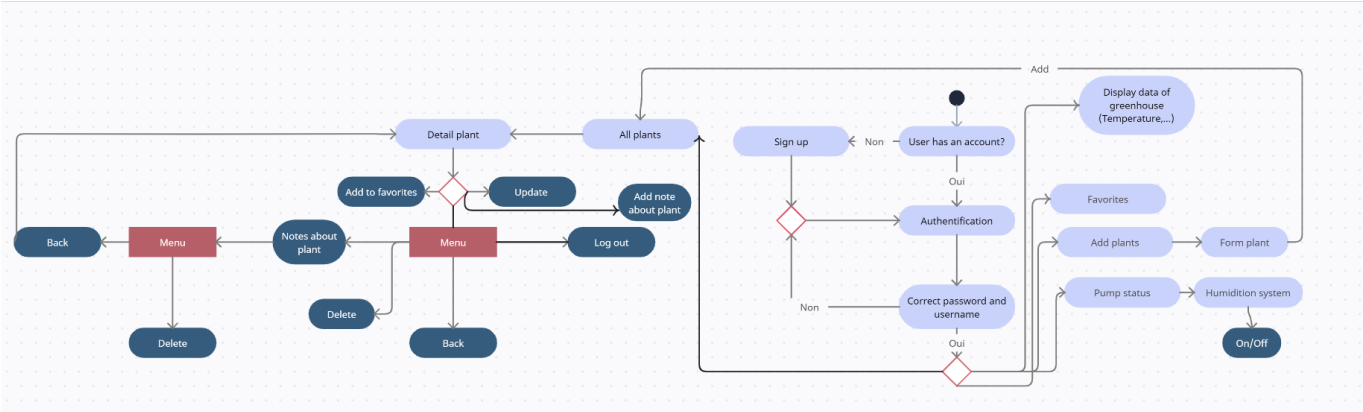

FIG. 2.10 : Activity diagram :process of using GreenHouseTech app .

2.3.4.4 Class Diagram

The class diagram represents the static description of the system to be developed by integrating the data-dedicated part and the processing part into each class. It is a key diagram of the modeling system, this representation focuses on the concept of class and associations.

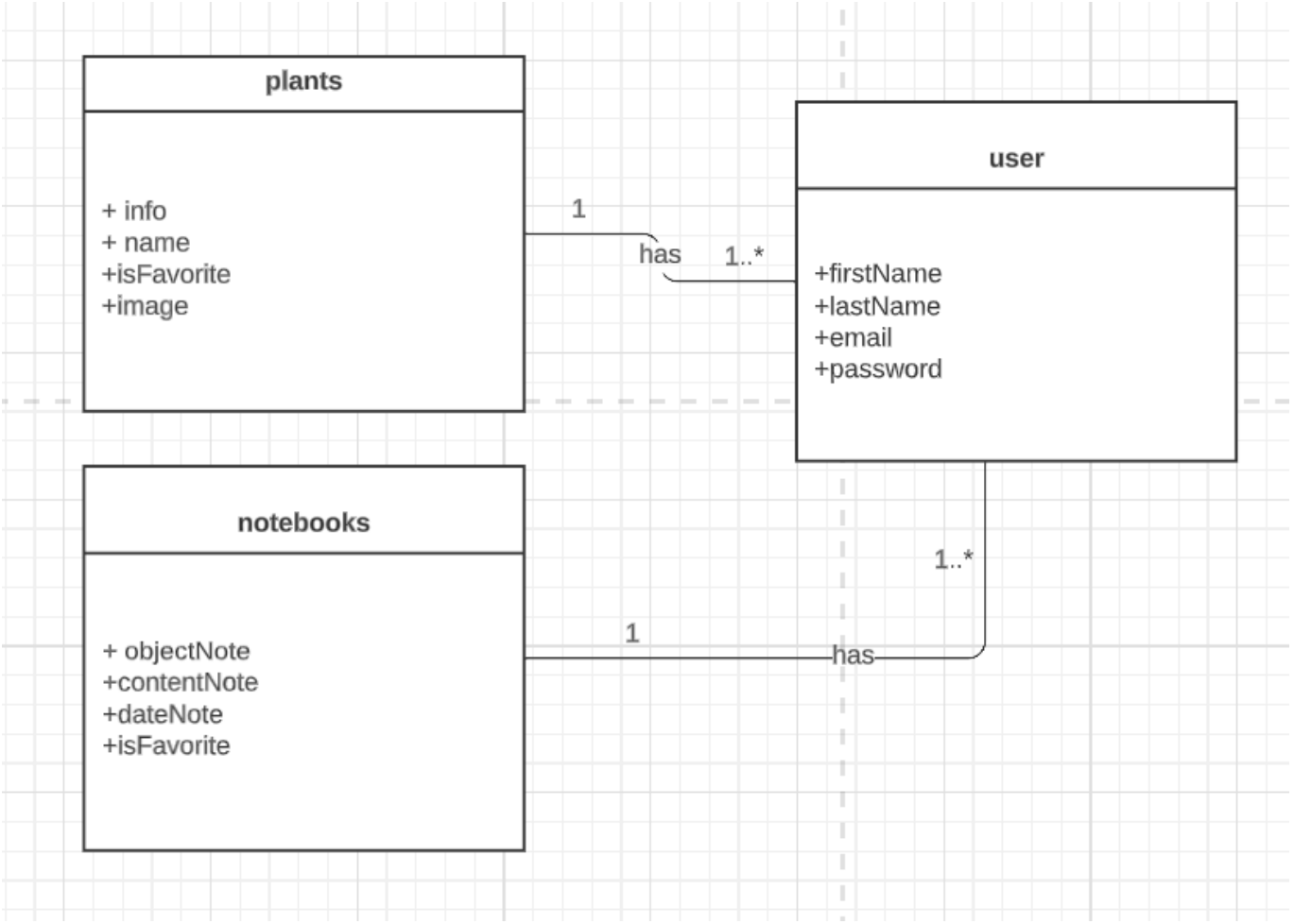

Fig. 2.11 : Class diagram.

A class is an abstract description of a set of objects with similar properties, common behavior, and common relationships with other objects.

2.4 Conclusion

In this chapter, we were able to define the functional and non-functional requirements and a conceptual view of the system to be realized. We were able to define the detailed use case diagram of our project. As well as explaining the operation of our system by presenting sequence diagrams of different scenarios, followed by the detailed class diagram. We can now proceed to the next phase, which is the solution

implementation phase, based on the concepts developed during the modeling phase.

Chapitre 3

The implementation phase

This chapter aims to present the implementation phase which concretely implements the design presented in the previous chapter. It outlines the steps of implementing the developed solution : the development environment and then screenshots of the realized application.

3.1 Introduction

Implementation is the most important phase after the analysis phase. The choice of development tools greatly influences the programming time cost and the flexibility of the product to be realized. This phase consists of implementing the chosen technologies and having an overview of our application .

3.2 Technical study

3.2.1 IOT tools

- TDS meter analogue sensor .
- Sensor of level of water
- LED
- LCD
- Arduino Uno
- Raspberry Pi
- DHT11
- Camera
- DHT11
- Breadboard
- cables ...
  I used python to program the raspberry pi and to send data from Things to Firebase .

3.2.1.1 VNC viewer

VNC Viewer is a versatile remote desktop software that allows users to access and control remote computers from anywhere. It offers a user-friendly interface and a wide range of features tailored for remote desktop access. VNC Viewer enables users to connect to desktops, laptops, and servers running various operating systems, including Windows, macOS, Linux, and Unix. We are used it to control Raspberry Pi .

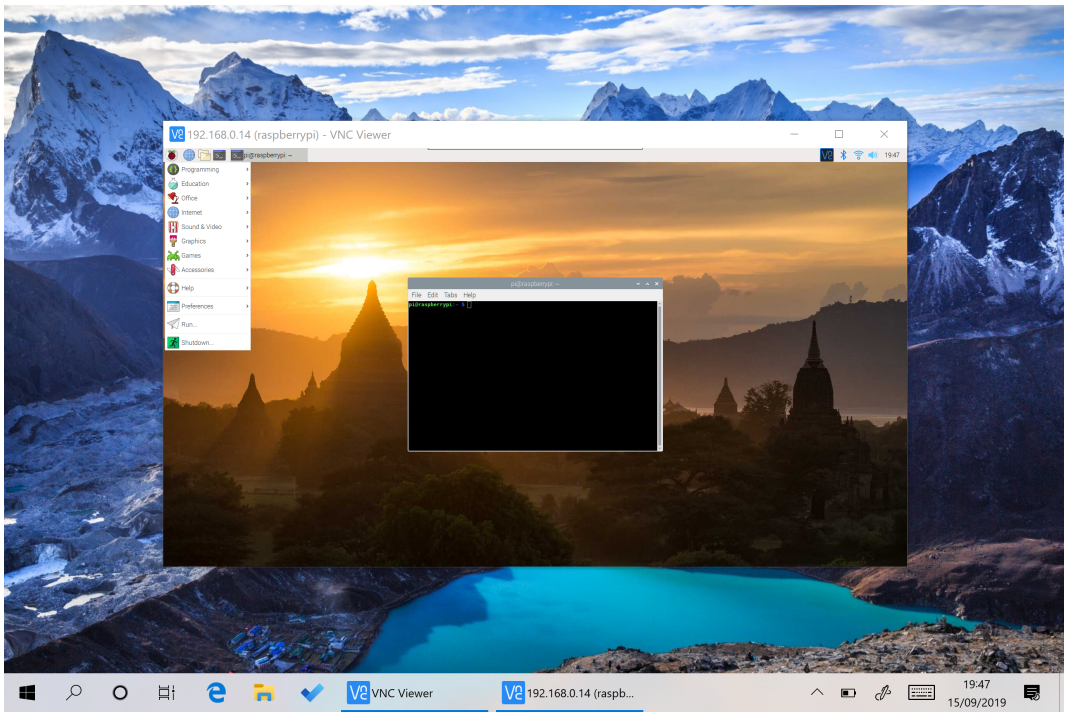

FIG. 3.1 : VNC Viewer .

3.2.1.2 PuTTY

PuTTY is a versatile and widely-used terminal emulator and SSH client, offering a range of tools and features essential for remote access and secure communication. Designed with simplicity and functionality in mind, PuTTY is a popular choice for managing and controlling remote systems.

FIG. 3.2 : Putty .

3.2.2 Development tools

3.2.2.1 Andruid Studio

Android Studio is an integrated development environment (IDE) specifically designed for Android app development. It provides a comprehensive set of tools and features tailored for creating, testing, and deploying Android applications. Android Studio enables developers to build various types of Android apps, including mobile applications, wearable apps, and even apps for smart TVs and IoT devices. With its rich set of libraries, templates, and emulators, Android Studio empowers developers to create high-quality, feature-rich Android applications with ease.

Fig. 3.3 : Android Studio .

Android Studio also provides extensive support for popular frameworks and technologies used in Android app development, such as Kotlin, Java, and the Android SDK. It offers seamless integration with key libraries and tools like Android Jetpack, Firebase, and Google Play services. With Android Studio, developers can take advantage of a unified and comprehensive environment specifically built to streamline the creation, testing, and deployment of Android applications. It empowers developers to efficiently build robust, scalable Android apps by leveraging the rich ecosystem of libraries, APIs, and tools available within the Android development ecosystem .

3.2.2.2 Firebase

Firebase is a comprehensive platform that offers a graphical tool for designing and managing databases, specifically designed for cloud-based applications. It provides a unified visual interface for database architects, developers, and administrators to efficiently work with databases and perform a wide range of tasks. With Firebase, users can seamlessly create and modify database tables, execute powerful queries using Firebase's real-time database, and effectively manage database users and access permissions. Additionally, Firebase goes beyond just database management by offering a suite of services for developing and deploying cloud-based applications, including authentication, cloud storage, hosting, and real-time messaging, making it a versatile and all-encompassing platform for building modern applications .

FIG. 3.4 : Firebase icon.

Firebase Console provides a user-friendly visual interface for managing various aspects of Firebase services and applications. It offers a comprehensive set of tools and functionalities that enable users to perform a wide range of tasks related to Firebase services. With Firebase Console, users can easily manage and configure authentication settings, set up and monitor cloud storage, analyze app performance and usage with analytics, send notifications, and manage various other features provided by Firebase. The console allows users to seamlessly navigate between different Firebase projects and provides a unified interface for efficiently managing and monitoring multiple Firebase applications and services.

3.2.3 Development languages

3.2.3.1 Java for Android

Java for Android is a programming language and platform specifically designed for developing applications on the Android operating system. Java is a widely-used and robust programming language that offers various features and capabilities for building Android applications. With Java, developers can define the behavior and logic of their Android apps, including handling user interactions, implementing data processing, and integrating with various device functionalities. Java code for Android is written using a series of Java classes and methods, which define different components and functionalities of the app. The Java code is then compiled into bytecode that can run on the Android platform, allowing developers to create powerful and interactive applications for Android devices.

Fig. 3.5 : java Android icon.

3.2.3.2 XML for Android

XML (Extensible Markup Language) for Android is a markup language used to structure and organize data in a human-readable format. XML is widely used in Android development to define the structure and content of various resources and configurations within an Android application. It is particularly useful for defining user interface layouts, storing data in a hierarchical manner, and configuring application settings. XML in Android allows developers to separate the presentation and behavior of their applications from the underlying code, making it easier to manage and modify application resources. With XML, developers can define the structure, attributes, and values of different elements to effectively represent and manipulate data within an Android application .

Fig. 3.6 : XML Android icon.

3.3 Project structure

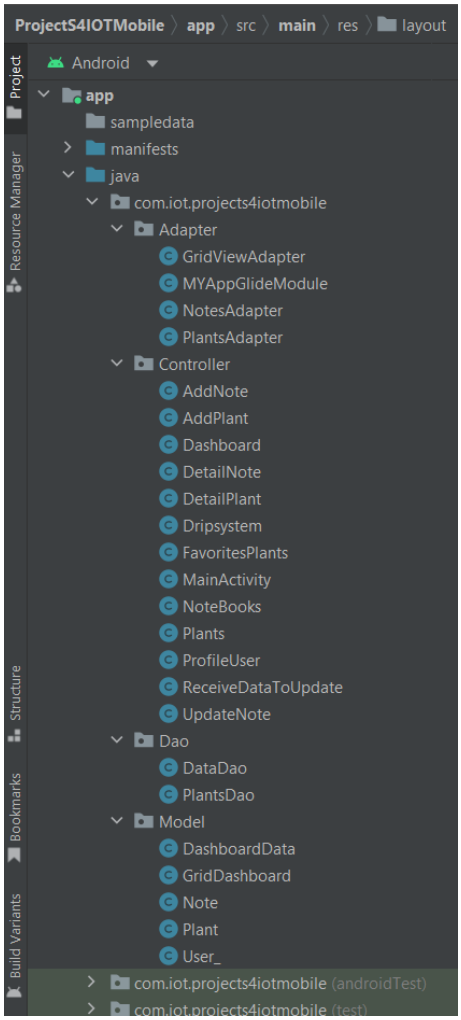

Fig. 3.7 : Code structure.

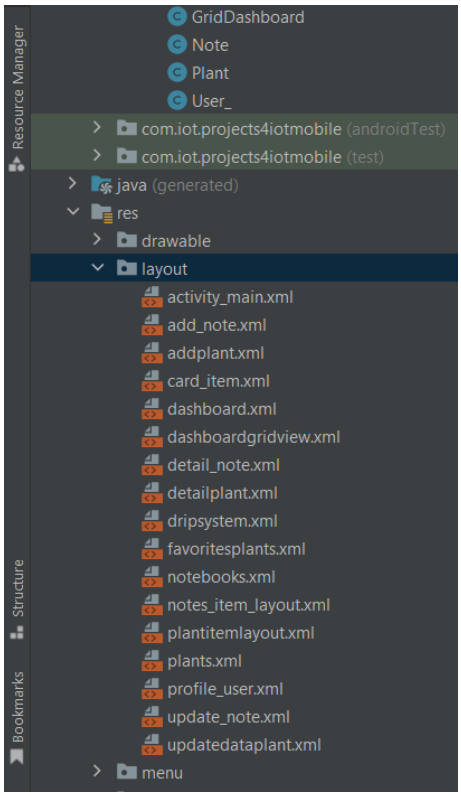

FIG. 3.8 : Code structure.

3.4 Presentation of interfaces

FIG. 3.9 : mobile application logo : GreenHouseTech.

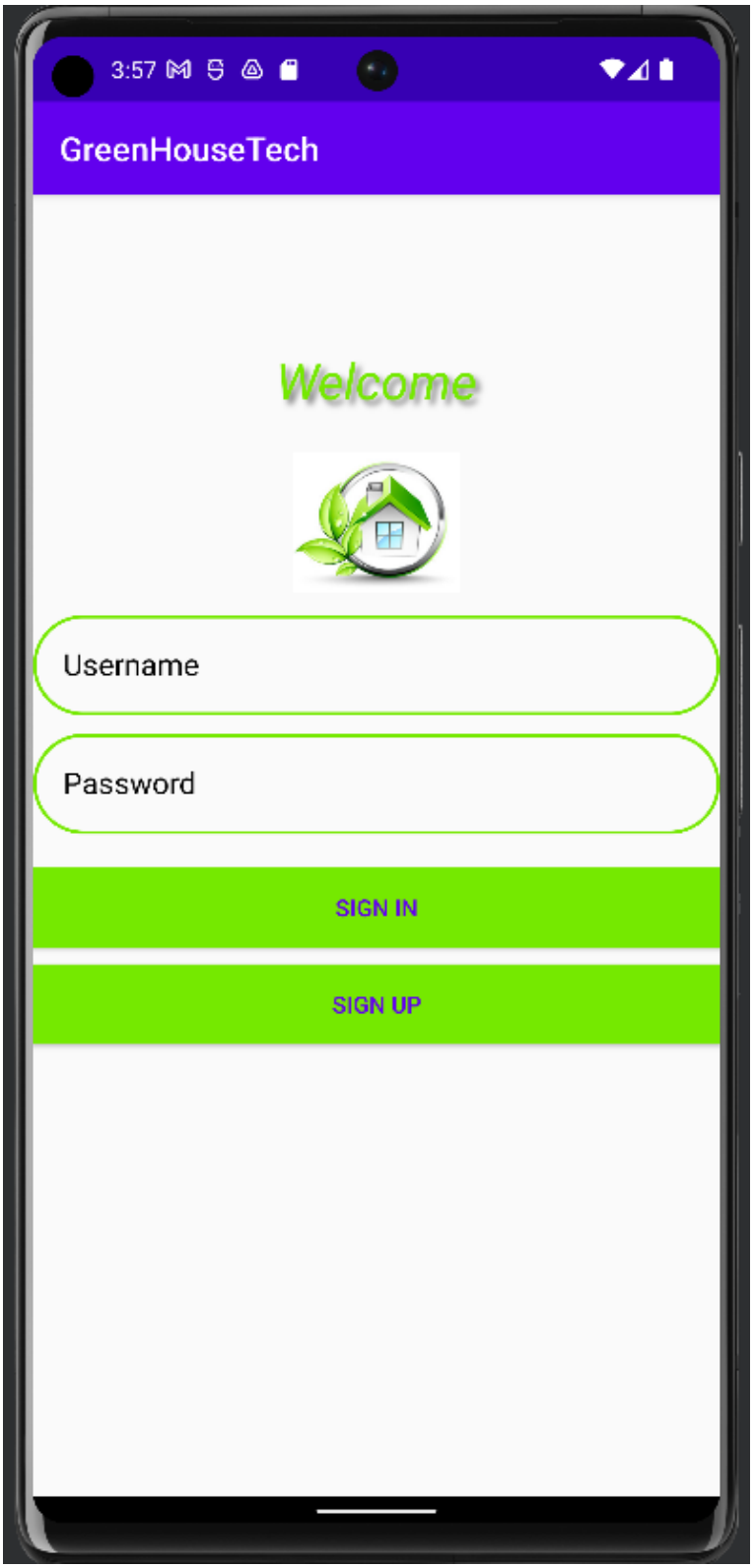

FIG. 3.10 : Log in.

FIG. 3.11 : Register .

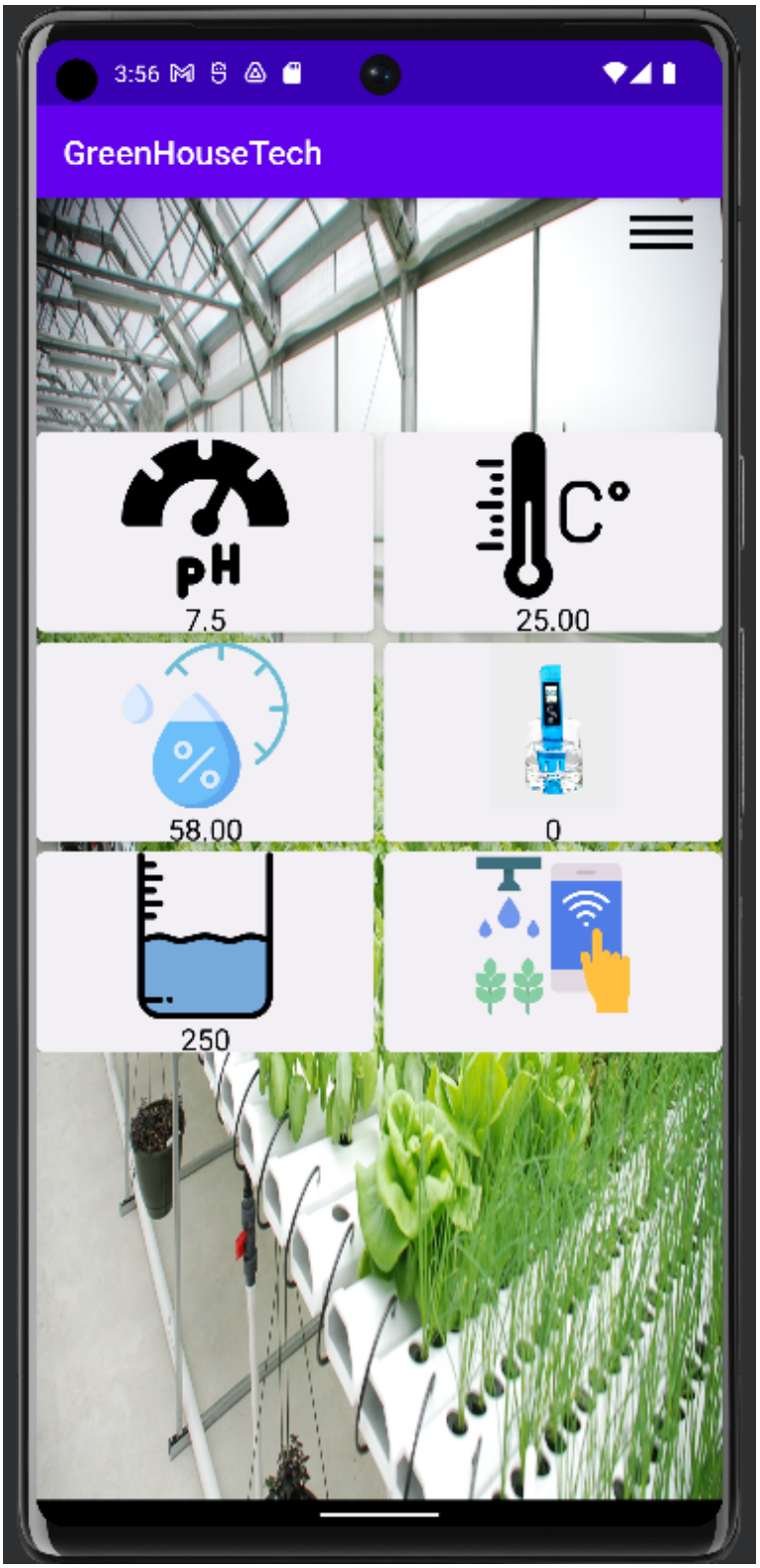

Fig. 3.12 : Dashboard.

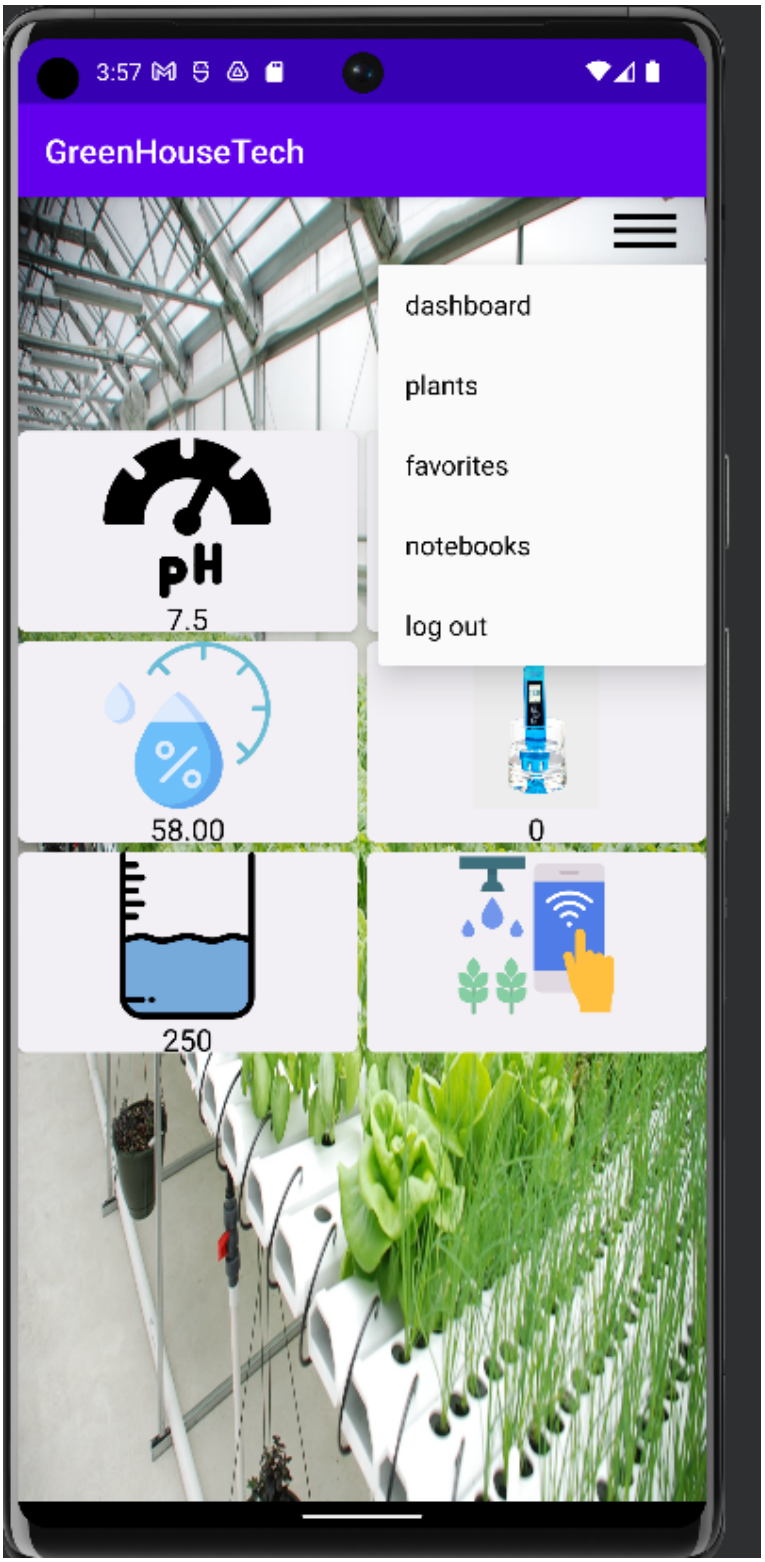

FIG. 3.13 : Dashboard.

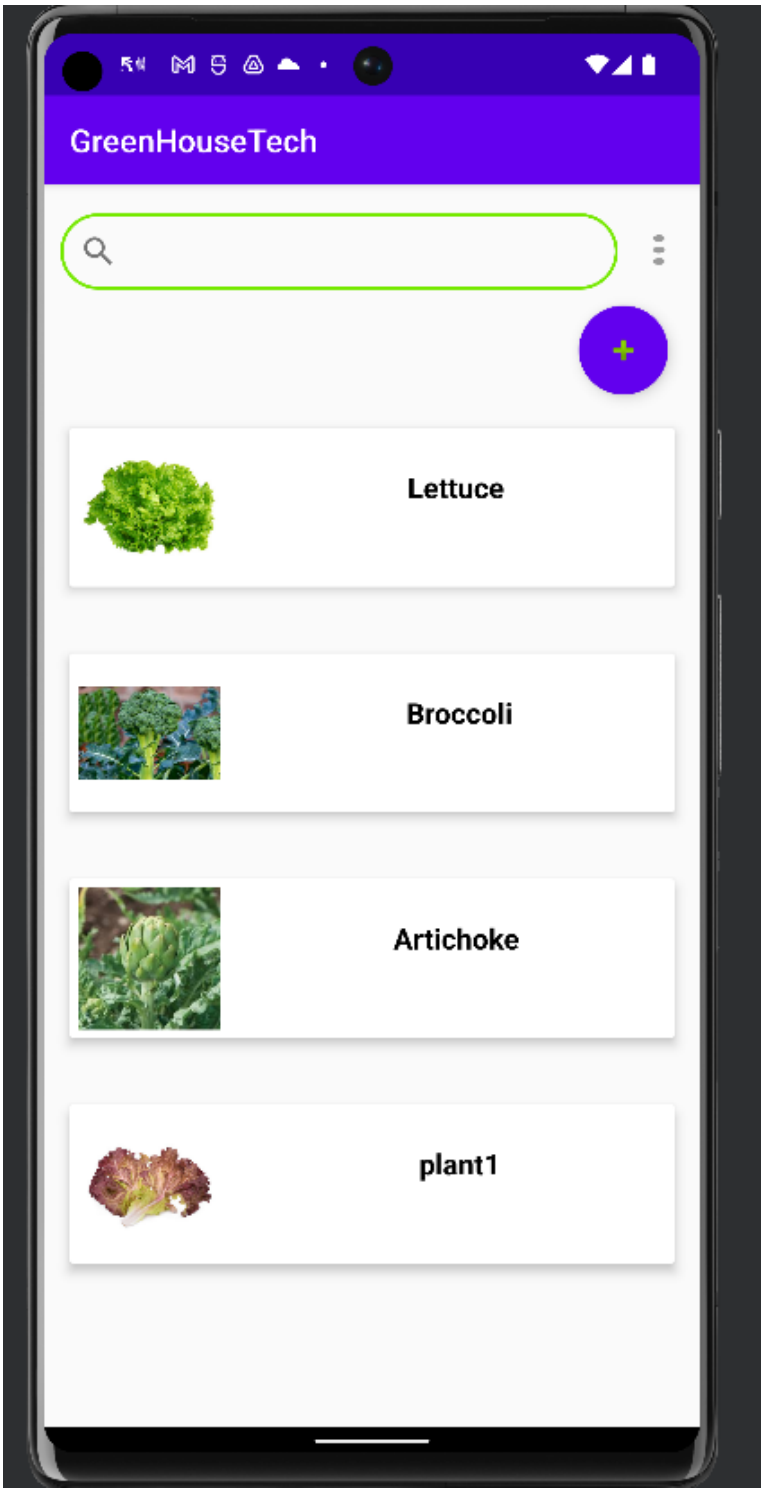

FIG. 3.14 : all plants interface .

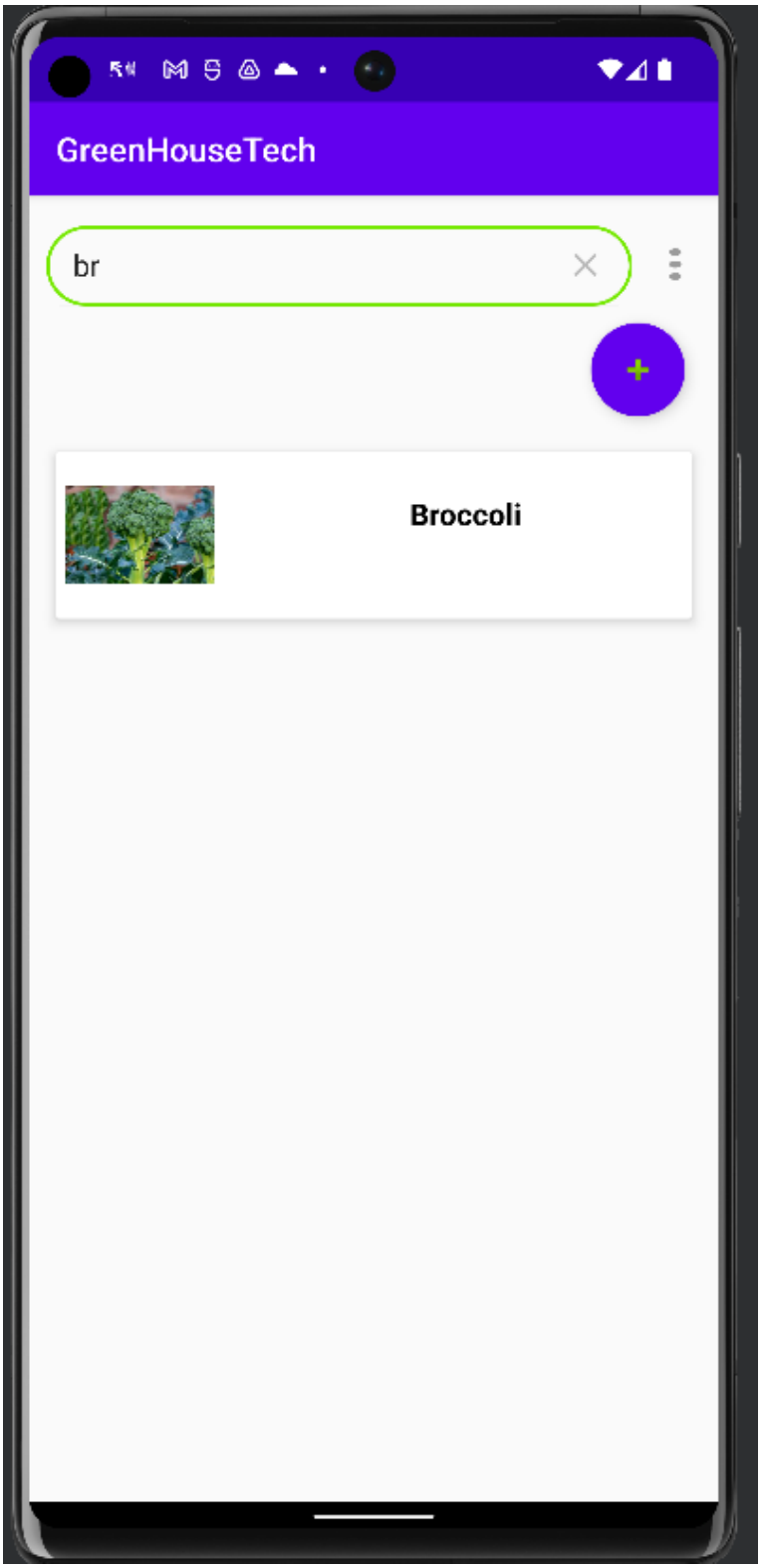

Fig. 3.15 : Search a plant.

FIG. 3.16 : Detail interface of plant.

FIG. 3.17 : Add a plant to greenHouse.

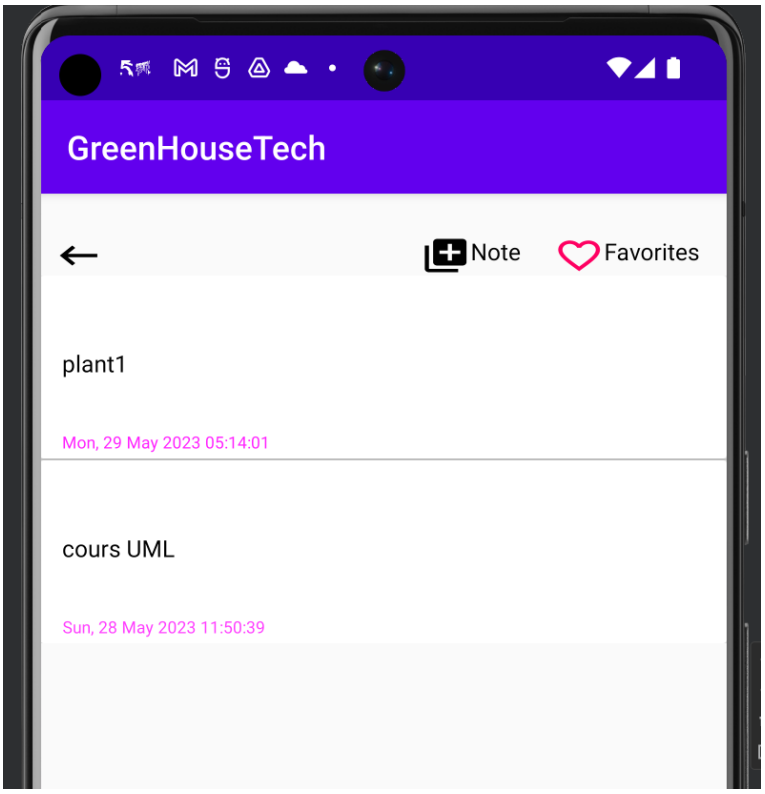

FIG. 3.18 : Notebooks interface .

FIG. 3.19 : Detail note .

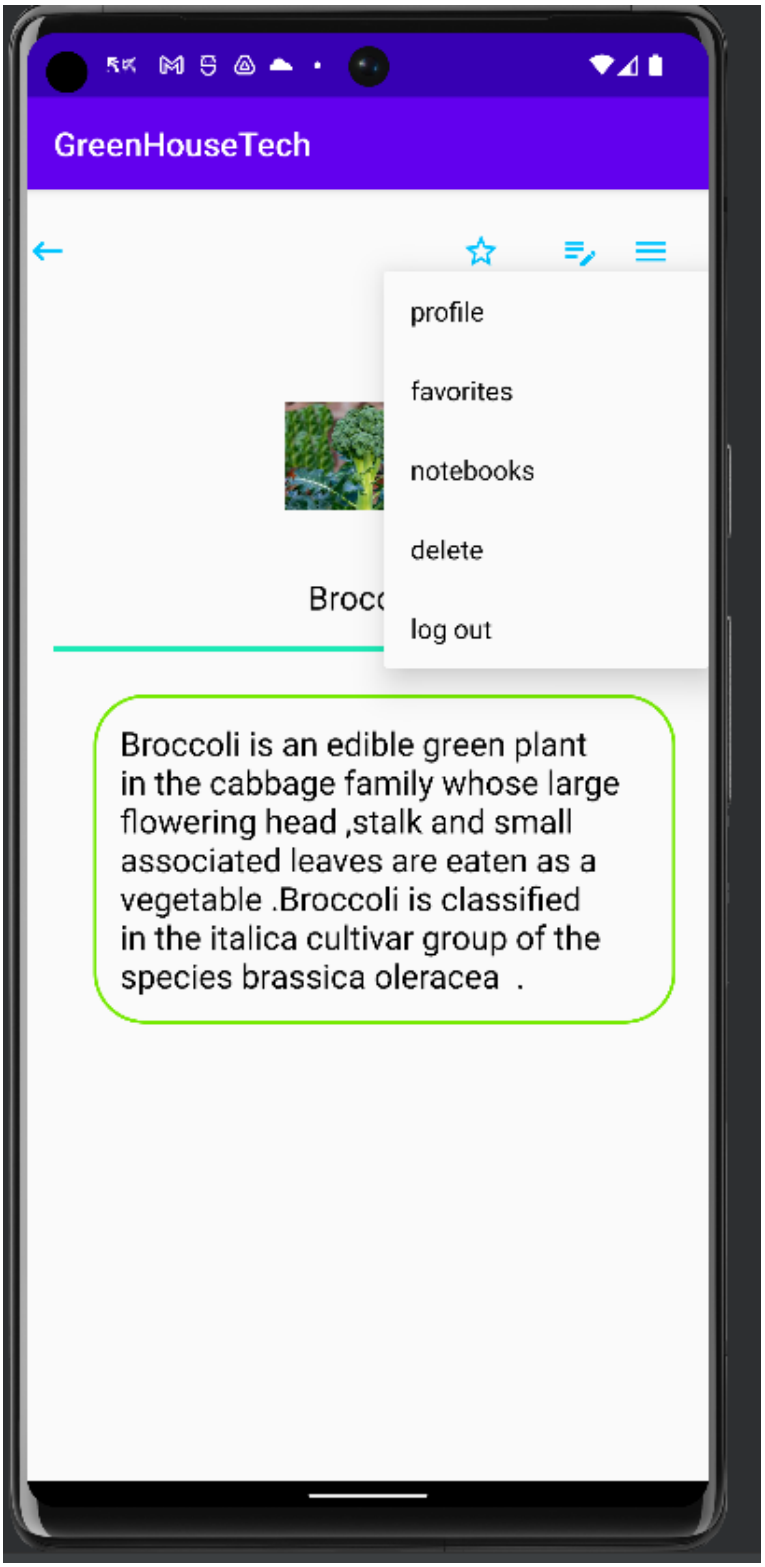

FIG. 3.20 : menu in inetrface detail plant .

FIG. 3.21 : Interface of water pump not completed...

3.5 Conclusion

In this chapter, we have presented the development environment and process, then the general structure of the front-end part of the application. So, We have exposed the development result using screenshots.

General conclusion

The present report presented the various stages of my final year project which focuses on development. The project was a challenging and rewarding experience. The project required extensive planning, design, and development to bring the idea of an online platform for ordering hydroponic farms to life. The end result of this project was a fully functional, user-friendly, and secure web application that enables customers to easily order indoor hydroponic farms online. The application was developed using Java EE and which allowed for a robust, scalable, and efficient solution. The user interface was carefully designed to ensure that customers could easily navigate the platform and customize their orders to meet their specific needs. The application was also designed with security in mind and features multiple layers of protection to ensure the safety of customer data and transactions. To conclude with, the next step in this journey is to focus on monitoring the hydroponic farms once they have been established. This will involve developing new technologies and strategies to ensure the health and productivity of these farms, as well as to minimize waste and ensure efficient resource usage.

Bibliography

[1] Android Documentation

[2] Arduino Documentation

[3] Hydroponic system