**3D Printed Smart Plant Base (Smart Home)**

**A PROJECT REPORT**

***Submitted by***

# Kaarthik Raja S [REGISTER NO: 211417104103]

# Dhana Akarshan V [REGISTER NO: 211417104050]

# Anerudh R [REGISTER NO: 211417104017]

***in partial fulfillment for the award of the degree***

***of***

# BACHELOR OF ENGINEERING

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

# ANNA UNIVERSITY: CHENNAI 600 025

**APRIL 2021**

## BONAFIDE CERTIFICATE

Certified that this project report “**SOUND CLASSIFICATION FOR RESPIRATORY DISEASES USING MACHINE LEARNING TECHNIQUE**” is the bonafide work of “**Kaarthik Raja S (211417104103), Dhana Akarshan V (211417104050), Anerudh R (211417104017)**” who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

### Dr.S.MURUGAVALLI,M.E.,Ph.D., Mr. N.SATHISH,M.E., HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR,

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE,

NAZARATHPETTAI, NAZARATHPETTAI, POONAMALLEE, POONAMALLEE, CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

### ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to express our heartfelt and sincere thanks to our Directors

Tmt.C.VIJAYARAJESWARI, **Dr.C.SAKTHIKUMAR, M.E., Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E., M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI, M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Mr.N.Sathish, M.E.** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

Kaarthik Raja S

Dhana Akarshan V

Anerudh R

#### ABSTRACT

Whether it be for work or pleasure, travelling is always a whole lot of fun. It gives us new experience and opportunities to grow as a “Human”. But what about the growth of your house plant’s while your away. Or what if your in home and you forget to water it or overwater it. That’s going to be a bummer. I have the perfect solution for all these problem’s, why not make an automatic watering system?

This Arduino based, uses a moisture sensor to measure the moisture level of the soil, the sensor is stuck into the soil and then flashes LED’s and provides an OLED display readout telling you whether you’re over or under watering your plant. This system uses AtMega328 microcontroller. It is programmed to sense the moisture content in the soil over a period of time. When the moisture content is less than the limit which is predefined, it will start supplying the desired amount of water till it reaches the limit. So when the soil is dry the pump will automatically water the fields and when the soil is wet the pump will automatically switch off, there by eradicate the need of manpower and conserve the time.

The LED emits red light when you forget to water it and it waits for a while and if you don’t show up to water your plants, it automatically water it. This 3D printed ”Automatic Smart Plant Pot” looks simple on the outside, but inside there are electronics, pumps, and a water reservoir that work together to keep your plants healthy and happy. Engineering and electronics, the solution requires only a little bit knowledge of electronics as well as that knowledge related to botany and plant physiology.

This 3D printed ”**Automatic Smart Plant Pot”** looks simple on the outside, but inside there are electronics, pumps, and a water reservoir that work together to keep your plants healthy and happy. With this I’ve also attached the yet to be 3D-Printed model of the base.

**LIST OF TABLES**

**CHAPTER NO. TITLE PAGE NO.**

#### 3. SYSTEM ANALYSIS

3.1 Minimum Hardware Requirements 11

3.2 Minimum Software Requirement 12

1. **SYSTEM DESIGN**

4.1. Data Dictionary of User 19

1. **SYSTEM TESTING**

6.1. Test Case for User 35

## LIST OF FIGURE

**CHAPTER NO. TITLE PAGE NO.**

### 4. SYSTEM DESIGN

4.1. ER-Diagram 19

4.2. Data Flow Diagram of 22

4.4. Use Case Diagram 23

4.5. Class Diagram 25

4.6 Sequence Diagram 28

1. **SYSTEM ARCHITECTURE**

5.1. System Architecture 30

## TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT**  iv

### 1 INTRODUCTION 1

1.1 Overview 2

1. **LITERATURE SURVEY** 5
2. **SYSTEM ANALYSIS** 8
   1. Existing System 9
   2. Proposed system 9
   3. Requirement Analysis and Specification 10
      1. Input Requirements 10
      2. Output Requirements 10
      3. Functional Requirements 11
   4. Feasibility Study 11
      1. Technical Feasibility 11
      2. Economic Feasibility 12
   5. Hardware Requirements 13
   6. Software Requirements 14

**CHAPTER NO. TITLE PAGE NO.**

1. **SYSTEM DESIGN** 18
   1. ER diagram 26
   2. Data dictionary 26
   3. Data Flow Diagram 27
   4. UML Diagrams 28
2. **SYSTEM ARCHITECTURE** 29
   1. Architecture Overview 30
   2. Module Design Specification 32
      1. Modules Explanation 33
3. **TESTING** 34
   1. System Testing 35
   2. Unit Testing 37
4. **CONCLUSION** 39
   1. Conclusion 39
   2. Future Enhancements 40

### APPENDICES 41

A.1 Sample Screens 41

A.2 Sample Code 49

A.3 Sample Result & Documentation 63

### REFERENCES 68

**CHAPTER 1**

**INTRODUCTION**

## 1.1 OVERVIEW

Watering the plant is the most important cultural practice and one of the labor intensive tasks in daily greenhouse operation. Watering systems ease the burden of getting water to plants when they need it. Knowing when and how much to water is two important aspects of watering process. To make the gardener works easily, the automatic plant watering system is created. There have a various type using automatic watering system that are by using sprinkler system, tube, nozzles and other. This system uses watering sprinkler system because it can water the plants located in the pots. This project uses Arduino board, which consists of ATmega328 Microcontroller. It is programmed in such a way that it will sense the moisture level of the plants and supply the water when required. This type of system is often used for general plant care, as part of caring for small and large gardens. Normally, the plants need to be watered twice daily, morning and evening. So, the microcontroller has to be coded to water the plants in the garden or farms about two times per day. People enjoy plants, their benefits and the feeling related to nurturing them. However for most people it becomes challenging to keep them healthy and alive. To accommodate this challenge we have developed a prototype, which makes a plant more self-sufficient, watering itself from a large water tank and providing itself with artificial sunlight. The pro-To type reports status of its current conditions and also reminds the user to refill the water tank. The system automation is designed to be assistive to the user. We hope that through this prototype people will enjoy having plants without the challenges related to absent or forgetfulness.

The main aim of this project was to provide water to the plants or gardening automatically using microcontroller (Arduino Uno). We can automatically watering the plants when we are going on vacation or don’t we have to bother my neighbours, Sometimes the neighbours do too much of watering and the plants end up dying anyway. the artificial application of water to the land or soil It is used to assist in the growing of agricultural crops, maintenance of landscapes, and re vegetation of disturbed soils in dry areas and during periods of inadequate rainfall. When a zone comes on, the water flows through the lateral lines and ultimately finally ends up at the irrigation electrode (drip) or mechanical device heads. Several sprinklers have pipe thread inlets on the lowest of them that permits a fitting and also the pipe to be connected to them.

The sprinklers are usually used in the top of the head flush with the ground surface. As the method of dripping will reduce huge water losses it became a popular method by reducing the labour cost and increasing the yields. When the components are activated, all the components will read and gives the output signal to the controller, and the information will be displayed to the user (farmer). The sensor readings are analog in nature so the ADC pin in the controller will convert the analog signals into digital format. Then the controller will access information and when the motors are turned On/Off it will be displayed on the LCD Panel, and serial monitor windows.

There are many systems are available to water savings in various crops, from basic ones to more technologically advanced ones. For instance, in one system plant watering status was monitored and irrigation scheduled based on temperature presents in soil content of the plant.Two potentiometers on the monitor allow you to set a maximum and minimum moisture level which then activates either the red LED, to indicate a low moisture level, to tell you that you need to water your plant, or the yellow LED on a high moisture level, to tell you that you’re overwatering your plant. You can also push the button next to the display to turn on the OLED display and to see the exact moisture level as well as the two set points.

# 

# CHAPTER 2

**LITERATURE SURVEY**

## Emmanuel Andres, Amir Hajjam ‘Advances and Perspectives in the Field of Auscultation, with a Special Focus on the Contribution of New Intelligent Communicating Stethoscope’

The stethoscope and the semantic of auscultatory findings were invented more than 200 years ago by Dr. Laennec and over the years very few changes have been made to both the stethoscope itself and the way in which it is used. . However, the ability to differentiate between normal and abnormal sounds or noises (vesicular sounds, wheezes, crackles, etc.) remains essential in clinical practice for correct diagnosis and management. Over the past two decades, much of the progress made in this area has resulted primarily from improvements made to the stethoscope itself. More recently, we have seen advances in the techniques used to process auscultatory signals as well as in the analysis and clarification of the resulting sounds. The availability of novel representations of the sounds, with phono- and spectrograms, also opens interesting perspectives in the context of diagnostic aids, but also in education and pedagogy. It aims to review recent technological advances, evaluate promising innovations and perspectives in the field of auscultation, with a special focus on the development of new intelligent communicating stethoscope systems in clinical practice, and in the context of teaching and telemedicine.

**B. M. Rocha, D. Filos, L. Mendes, I. Vogiatzis, E. Perantoni, E. Kaimakamis, P. Natsiavas, A. Oliveira, C. Jácome, A. Marques, R. P. Paiva, I. Chouvarda, P.**

## Carvalho, N. MaglaverasΑ. ‘Automated Irrigation System with IOT’

In Sensor based Automated Irrigation System with IOT mentioned about using sensor based irrigation in which the irrigation will take place whenever there is a change in temperature and humidity of the surroundings. The flow of water is managed by solenoid valve. The opening and closing of valve is done when a signal is send through microcontroller. The water to the root of plant is done drop by drop using rain gun and when the moisture level again become normal then sensor senses it and send a signal to microcontroller and the value is then closed. The two mobile are connected using GSM. The GSM and microcontroller are connected using MAX232. When moisture of the soil become low moisture sensor sense it and send signal to microcontroller, then the microcontroller gives the signal to mobile and it activate the buzzer. This buzzer indicates that valve needs to be opened by pressing the button in the called function signals are sent back to microcontroller. Microcontroller used can increase System Life and lower the power Consumption. There system is just limited to the automation of irrigation system and lacks in extra ordinary features

## Demir, F, Sengur, A. and Bajaj, V. ‘In Automated Irrigation System Using a Wireless Sensor Network and GPRS Module.

In Automated Irrigation System Using a Wireless Sensor Network and GPRS Module mentioned about using automatic irrigation system in which irrigation will take place by wireless sensor units (WSUs) and a wireless information unit (WIU), linked by radio transceivers that allowed the transfer of soil moisture and temperature data, implementing a WSN that uses ZigBee technology. It takes a measure of temperature and moisture using sensor and controlled by microcontroller. The WIU has also a GPRS module to transmit the data to a web server via the public mobile network. The information can be remotely monitored online through a graphical application through Internet access devices. This irrigation system allows cultivation in places with water scarcity thereby improving sustainability and it is feasible system. But due to Zigbee protocol this system becomes more costly.

**Singh, A., Thakur, N. and Sharma, A. ‘In Wireless Sensor Network based Remote Irrigation Control System and Automation using DTMF code’.**

In Wireless Sensor Network based Remote Irrigation Control System and Automation using DTMF code mentioned about using automated irrigation system for proper yield and handled remotely for farmer safety. Wireless sensor network and Embedded based technique of DTMF (Dual Tone Multiple Frequency) signaling to control water flow for sectored, sprinkler or drip section irrigation. Circuit switching instead of packet switching used by SMS controlled devices available currently in the market. The farmer can use his cell phone or landline phone for the purpose of starting and controlling the irrigation and the pesticide spraying, just by dialing and sending the DTMF commands over the GSM network. This system will be very economical in terms of the hardware cost, power consumption and call charges. Farmers have to control (on/off) the valves time to time (even at night) which increases the running cost because every time we have to make a call to on or off the valves and it is also very inconvenient. Farmers are unable to know the status of power supply at the field

**Georgios Ntritsos, Jacob Franek, Lazaros Belbasis, Maria A Christou, Georgios Markozannes, Pablo Altman, Robert Fogel, Tobias Sayre,Evangelia E Ntzani, Evangelos Evangelou. ‘In Wireless Sensor Network Based Automated Irrigation and Crop Field Monitoring System’**

In Wireless Sensor Network Based Automated Irrigation and Crop Field Monitoring System mentioned about using wireless sensor network based automated irrigation system for optimize water use for agricultural purpose. The system consists of distributed wireless sensor network of soil moisture, and temperature sensors placed in the crop field. To handle the sensor information Zig bee protocol used and control the water quantity programming using an algorithm with threshold values of the sensors to a microcontroller for irrigation system. The system continuously displays the abnormal condition of the land (soil moisture, temperature level). Using a GSM modem with GPRS facility feature provides the information to fanners and interface with PIC 18F77 A microcontroller. The Irrigation system is automatic and manual mode. This system increase the crop fields, improve the crop quality, increase the energy and reduce the non-point source pollution.Due to PIC microcontroller the length of the program will be big because of using RISC (35 instructions).

# CHAPTER 3

**SYSTEM ANALYSIS**

## 3.1 EXISTING SYSTEM

The existing system isn’t a user friendly one, it doesn’t have a user interface and if any changes needed to be made it can only be done by a developer and not by commoners.

The automatic irrigation (i.e: Existing system) system on sensing soil moisture project is intended for the development of an irrigation system that switches submersible pumps on or off by using relays to perform this action on sensing the moisture content of the soil. The main advantage of using this irrigation system is to reduce human interference and ensure proper irrigation.

The Microcontroller acts as a major block of the entire project, and a power supply block is used for supplying power of 5V to the whole circuit with the help of a transformer, [a bridge rectifier circuit](https://www.elprocus.com/bridge-rectifier-circuit-theory-with-working-operation/) and a voltage regulator. The [8051 microcontroller is programmed](https://www.elprocus.com/types-of-interrupts-in-8051-microcontroller-and-interrupt-programming/) in such a way that it receives the input signal from the sensing material which consists of a comparator to know the varying conditions of the moisture in the soil. The OP-AMP which is used as comparator acts as an interface between the sensing material and the microcontroller for transferring the moisture conditions of the soil, viz.wetness, dryness, etc

## 3.2 PROPOSED SYSTEM

The main objective of this project is to water the plants or your garden automatically using microcontroller. Water conservation is one of the major objectives of this project. We can automatically water the plants when we are going on a vacation or stuff. We don’t have to bother our neighbor’s, friends or relatives, sometimes people tend to over water the plants and in such an event the plants end up dying. In large scale it can used in assisting in the growth of agricultural crops, maintenance of landscapes, and re-vegetation of disturbed soils in dry areas and during periods of inadequate rainfall

### 3.2.1 Advantages

* Free yourself from manual labor.
* Prevents Weeds and diseases.
* Preserves soil structure and nutrients.
* Just the right amount of water will be released, While having a Lush Garden. (Unlike when you hand water, when you can easily use more water than you need.)
* The water will be directed to exactly where it is needed. (Unlike with hand watering, when you can easily waste water by splashing it around where plants roots cannot retrieve it easily).
* An automatic shut off can help to keep water usage to a minimum.

## 3.3 REQUIREMENT ANALYSIS AND SPECIFICATION

The requirement engineering process of feasibility study, requirements elicitation and analysis, requirement specification, requirements validation and requirement management. Requirement elicitation and analysis is an iterative process that can be represented as a spiral of activities, namely requirements discovery, requirements classification and organization, requirement negotiation and requirements documentation.

### 3.3.1 INPUT REQUIREMENT

The input requirement at the base requires data from users such as unique IP address of the product to link the product with the user interface.

### 3.3.2 OUTPUT REQUIREMENT

The output requirement depends on the dataset produced by the moisture sensor and it shows whether the plant is healthy or not.

### 3.4 FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

#### 3.4.1 TECHNICAL FEASIBILITY

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible.

#### 3.4.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn’t incur any kind of increase in the expenses. Programming Language for Web-App development is open source and readily available for all users. Since the project is implemented in flutter it is cost efficient.

##### 3.5 HARDWARE REQUIREMENTS

|  |  |
| --- | --- |
| HARD DISK | >90GB |
| PROCESSOR | >Core i3 2.4GHz |
| SYSTEM TYPE | 32bit / 64 bit |
| RAM | >2GB |

##### 3.6 SOFTWARE REQUIREMENTS

|  |  |
| --- | --- |
| OPERATING SYSTEM | WINDOWS 7/8/8.1/10 |
| INTEGRATED DEVELOPMENT KIT | ANACONDA V2019 |
| PROGRAMMING LANGUAGE  (BACKEND) | C# |
| PROGRAMMING LANGUAGE  (FRONTEND) | Java |
| DATABASE | PYTHON SQLITE 3 |
| APPLICATION PROGRAMMING  INTERFACE (API) | Telize |

## 3.7 Hardware Requirements

## Arduino Nano

## Mini sub-emersible pump

## Water level sensor

## Soil moisture sensor

## AWS Cloud

## Half Perma-protoboard(Bread Board)

## Two potentiometers on the monitor which allows you to set a maximum and minimum moisture level which then activates either the red LED, to indicate a low moisture level, or the yellow LED on a high moisture level, to tell you that you’re overwatering your plant. You can also push the button next to the display to turn on the OLED display and to see the exact moisture level as well as the two set points.

## 

**3.7.1 Arduino Nano**

In figure it is showing an Arduino board is an open source platform used for building electronics projects. Arduino is a programmable circuit’s board which we can write a

program based on your projects. Arduino program will be uploading with IDE (Integrated Development Environment) software that runs on your computer, it is used to write and upload computer code to the Arduino physical board. Arduino

language is merely a set of C/C++ functions that can be called from your code

**3.7.2 Soil Moisture Sensor**

Soil moisture sensors measure the volumetric water content in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighting of

a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content. The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity. Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or

gardeners. Soil moisture sensors typically refer to sensors that estimate volumetric water content. Another class of sensors measure another property of moisture in soils called water potential. These sensors are usually referred to as soil water potential sensors and include tension-meters and gypsum blocks.

**3.7.2 Water Pump**

A small pump plus a driver. A driver is to provide enough current for the pump, my application needs a spray distance about one meter, so this pump is enough. But if you

need to make a system that needs a large spray range, you may need larger pumps, or even a pressurized device to make the projectile even farther, such as the watering system in a tea

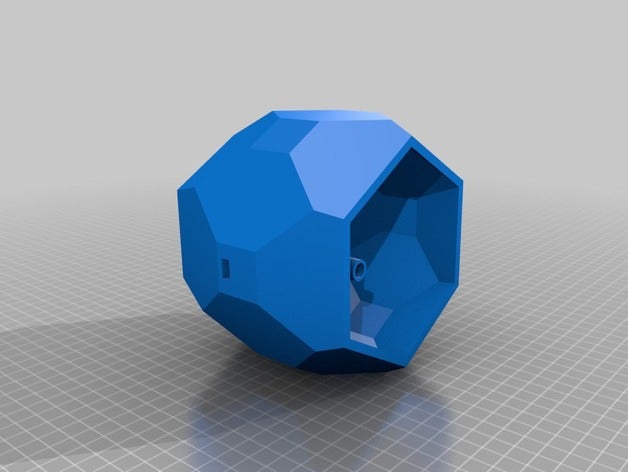
garden.

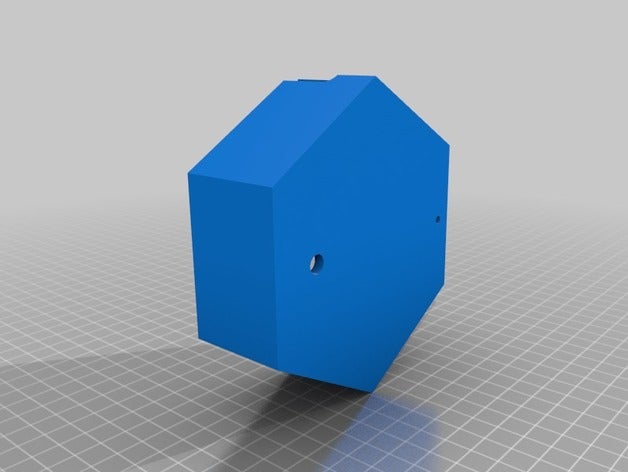
**3.7.2 LCD Display**

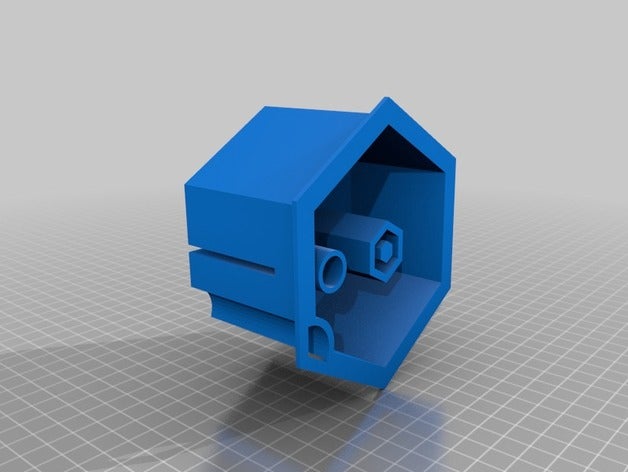
Liquid Crystal Display (LCD) screen is an electronic display module. An LCD has a wide range of applications in electronics. The most basic and commonly used LCD in

circuits is the 16x2 display. LCDs are commonly preferred in display because they are cheap, easy to programme and can display a wide range of characters and animations. A 16x2 LCD have two display lines each capable of displaying 16 characters. This LCD has Command and Data registers. The command register stores command instructions given to the LCD while the Data register stores the data to be displayed by the LCD.

**3.7.3 Custom Designed 3d Models**

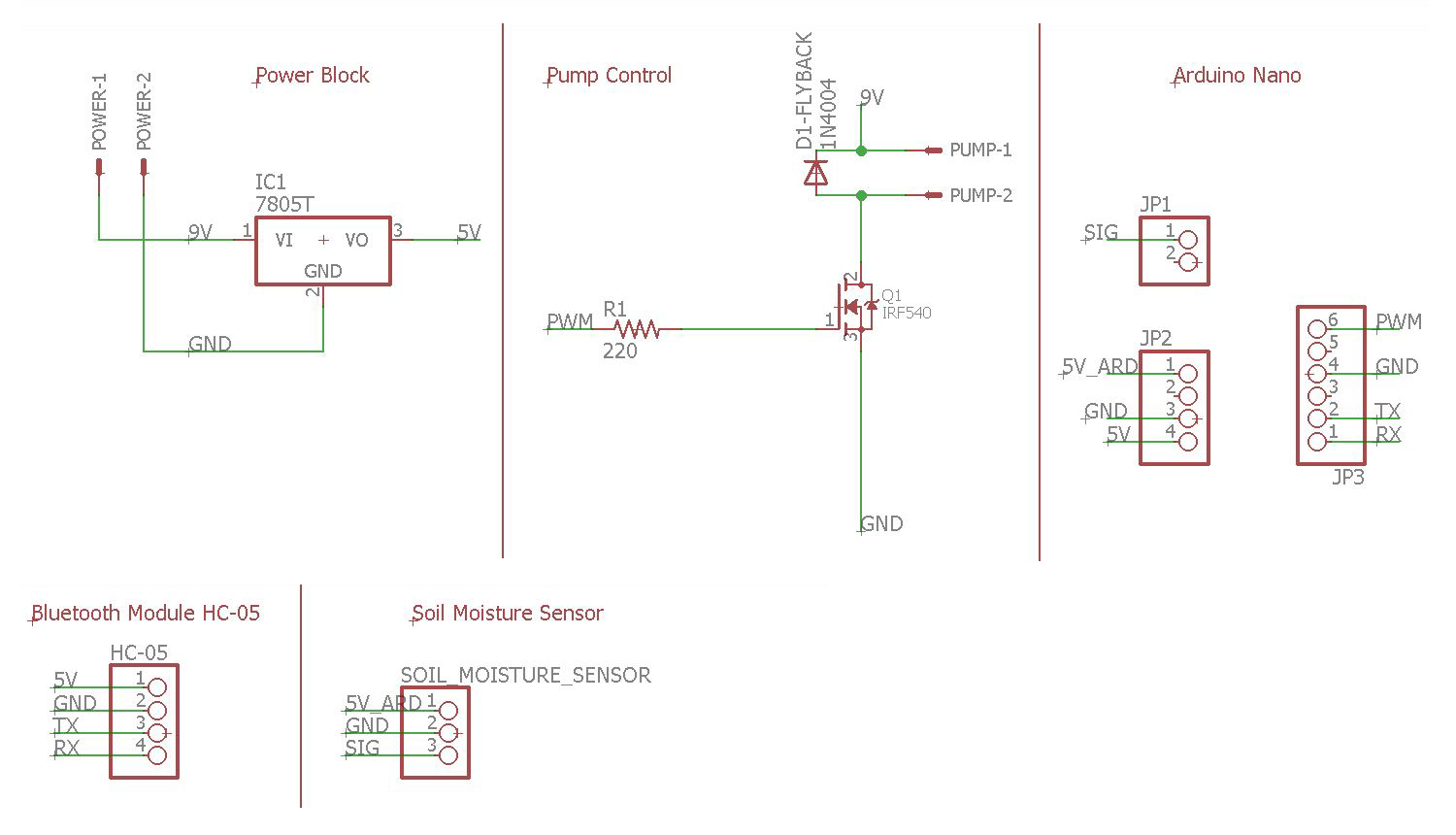
****

****

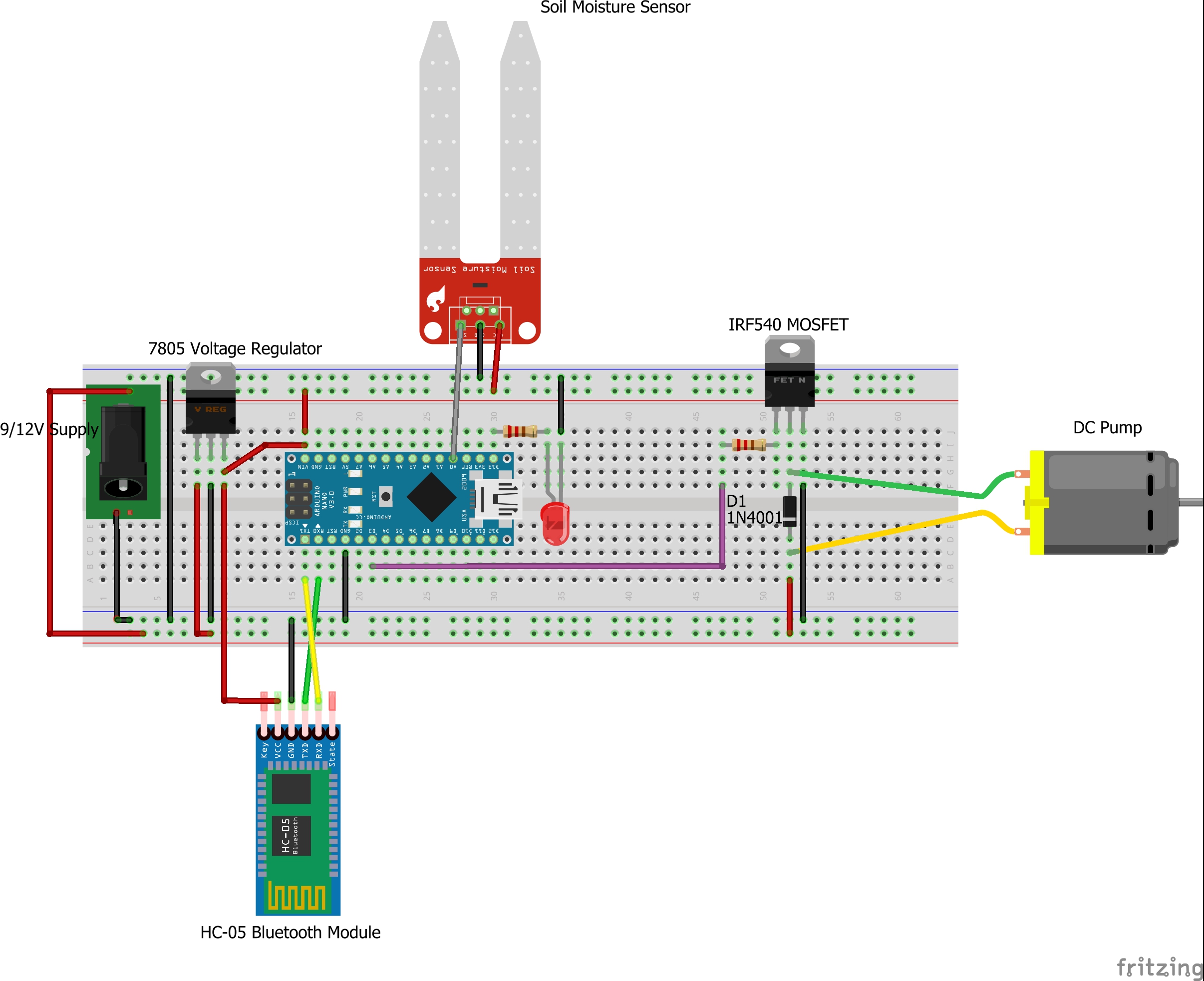
****

**3.7.4 System Implementation**

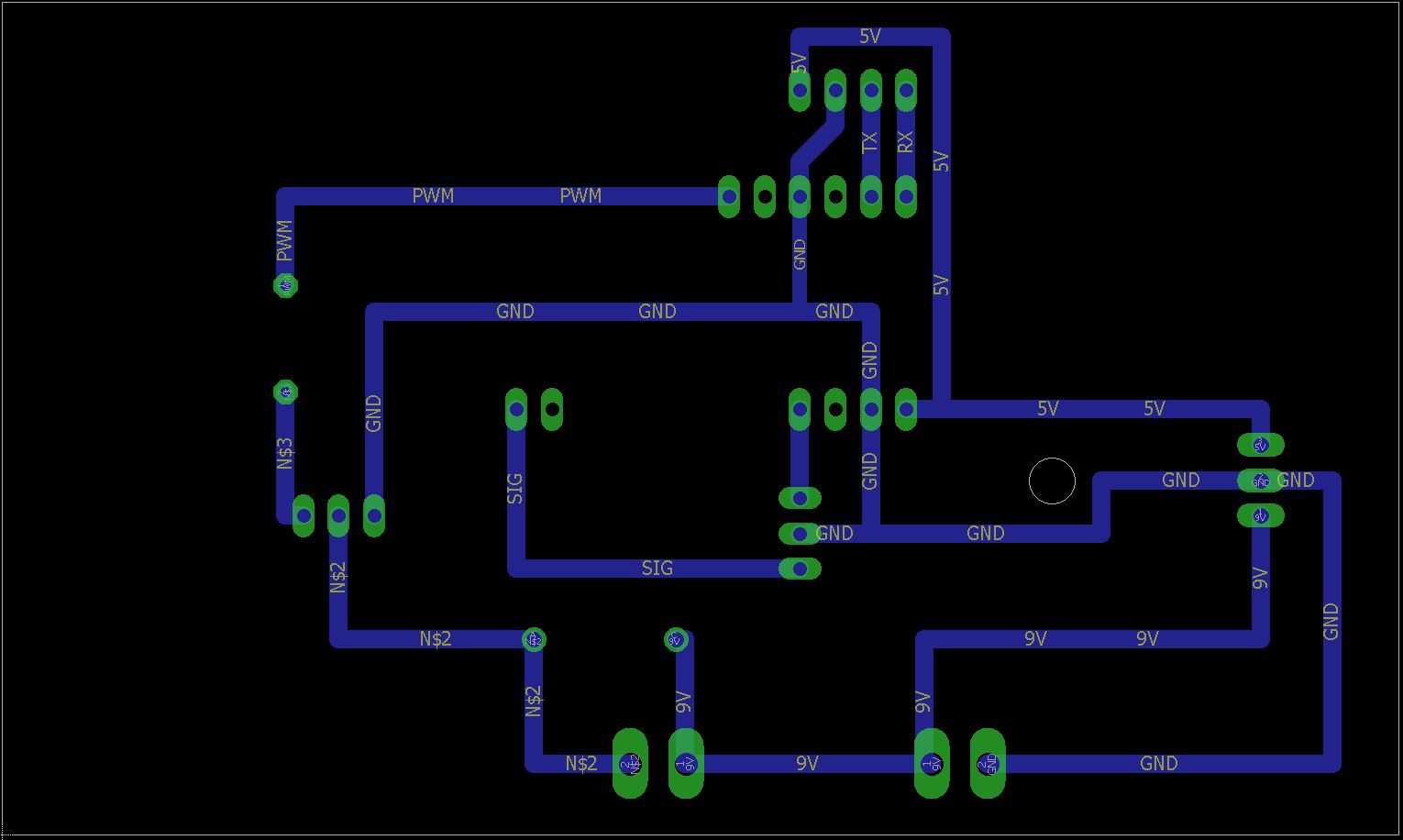
**3.7.4.1 Circuit Schematic Diagram of the System**

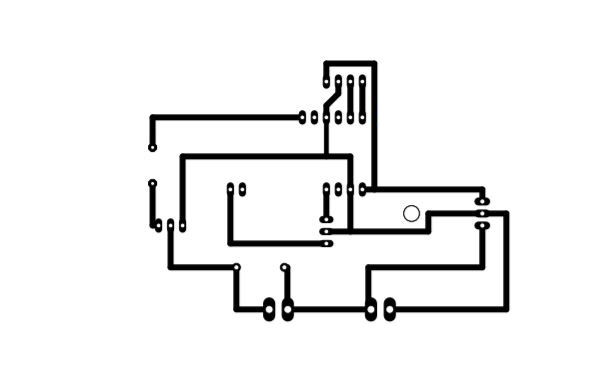
****

**3.7.4.2 Pictorial Circuit Diagram of the System**

****

**3.7.4.3 Custom Built PCB**

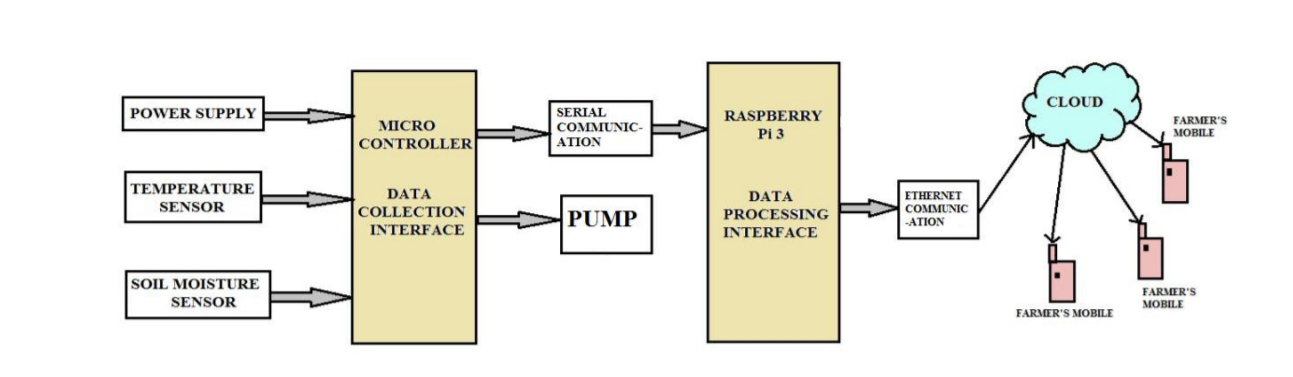
****

****

**Chapter 4**

**SYSTEM DESIGN**

#### 4.1 ER DIAGRAM



#### 4.2 DATA DICTIONARY

A data dictionary, or metadata repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format". Oracle defines it as a collection of tables with metadata. The term can have one of several closely related meanings pertaining to databases and database management systems (DBMS):

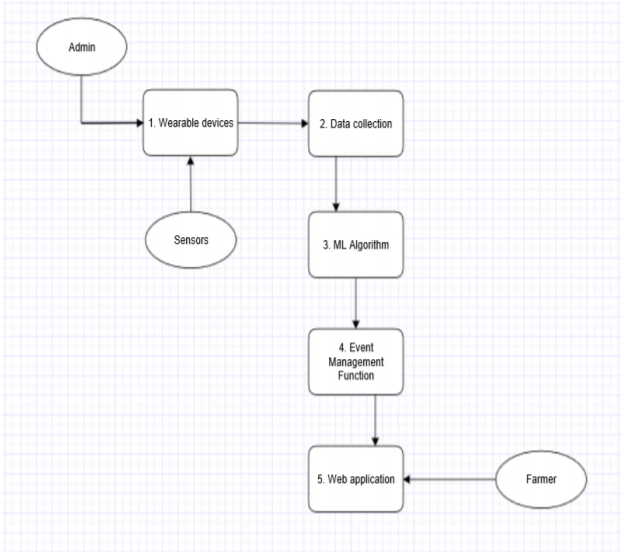
* A document describing a database or collection of databases user
* An integral component of a DBMS that is required to determine its structure
* A piece of middleware that extends or supplants the native data dictionary of a DBMS

**4.2.1 User:**

|  |  |  |
| --- | --- | --- |
| **FIELD** |  | **DATATYPE** |
| User\_id | ObjectId |  |
| Email\_id | String |  |
| Password | String |  |
| Unique IP | Number |  |

#### 4.3 DATA FLOW DIAGRAM

A picture is worth a thousand words. A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system. It is usually beginning with a context diagram as the level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to level 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of details for decomposing function really depends on the complexity that function.



##### 4.4 UML DIAGRAMS

UML stands for Unified Modeling Language. It’s a rich language to model software solutions, application structures, system behavior and business processes. There are 14 UML diagram types to help you model these behaviors. Unified Modeling Language (UML) is a standard visual modeling language intended to be used for

* modeling business and similar processes,
* analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

Specification explained that process:

* provides guidance as to the order of a team’s activities,
* specifies what artifacts should be developed,
* directs the tasks of individual developers and the team as a whole, and
* offers criteria for monitoring and measuring a project’s products and activities.

UML is intentionally process independent and could be applied in the context of different processes. Still, it is most suitable for use case driven, iterative and incremental development processes. An example of such process is Rational Unified Process (RUP).UML is not complete, and it is not completely visual. Given some UML diagram, we can't be sure to understand depicted part or behavior of the system from the diagram alone. Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams. For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases. Name of an abstract classifier is shown in italics while the final classifier has no specific graphical notation, so there is no way to determine whether the classifier is final or not from the diagram.

###### 4.4.1 List of UML Diagram Types

So, what are the different UML diagram types? There are two main categories; structure diagrams and behavioral diagrams. Click on the links to learn more about a specific diagram type.

###### 4.4.2 Structure Diagrams

Structure diagrams show the things in the modeled system. In a more technical term, they show different objects in a system. Behavioral diagrams show what should happen in a system. They describe how the objects interact with each other to create a functioning system.

###### 4.4.3 Class Diagram

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom. In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

###### 4.4.4 Component Diagram

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems with many components. Components communicate with each other using interfaces. The interfaces are linked using connectors. The image below shows a component diagram.

###### 4.4.5 Deployment Diagram

A deployment diagram shows the hardware of your system and the software in that hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration.

###### 4.4.6 Package Diagram

As the name suggests, a package diagram shows the dependencies between different packages in a system. Check out this wiki article to learn more about the dependencies and elements found in package diagrams.

**4.4.7 Composite Structure Diagram**

As the name suggests, a package diagram shows the dependencies between different packages in a system. Check out this wiki article to learn more about the dependencies and elements found in package diagrams.

###### 4.4.8 Use Case Diagram

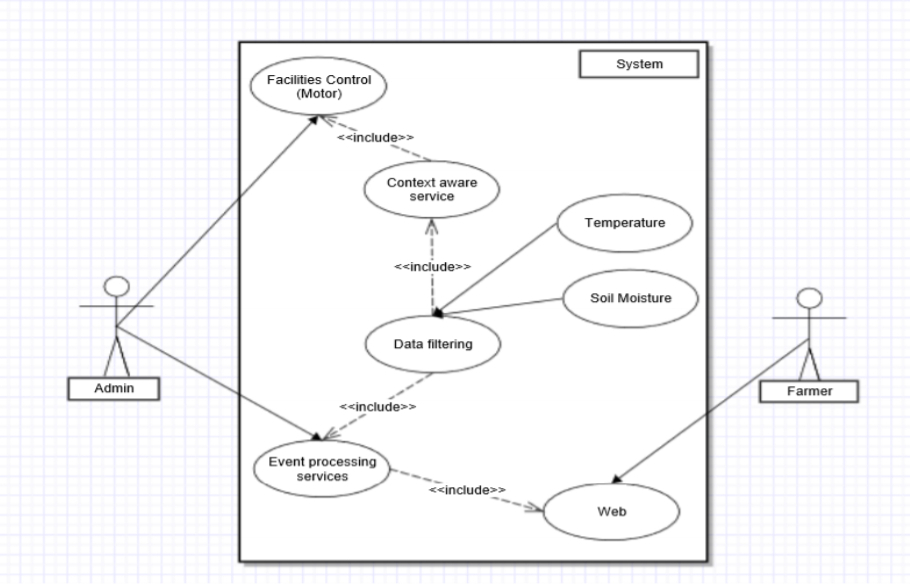
As the most known diagram type of the behavioral UML diagrams, use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact. It’s a great starting point for any project discussion because you can easily identify the main actors involved and the main processes of the system. You can create use case diagrams using our tool and/or get started instantly using our use case templates.

###### 4.4.9 Activity Diagram

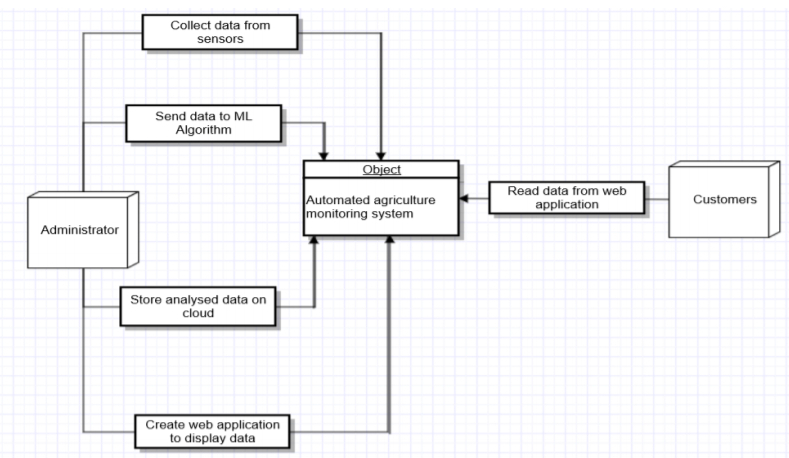
Activity diagrams represent workflows in a graphical way. They can be used to describe the business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams. Check out this wiki article to learn about symbols and usage of activity diagrams.

###### 4.4.10 Sequence Diagram

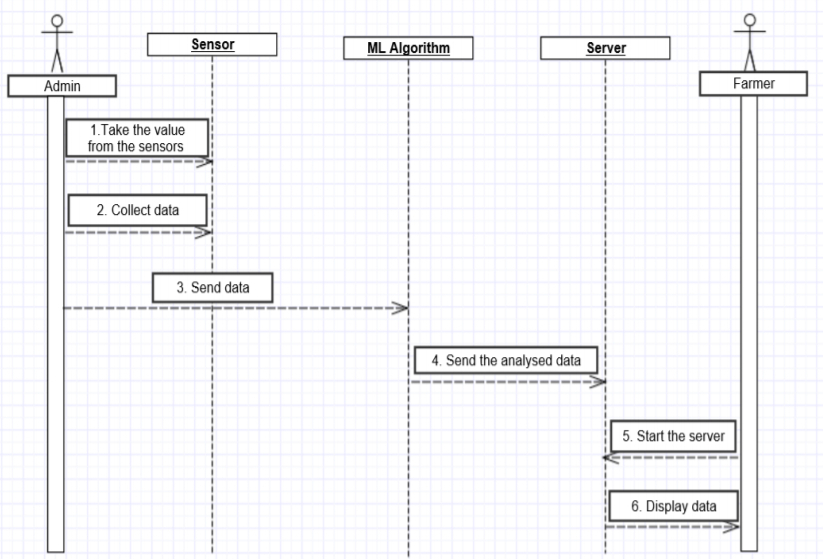
Sequence diagrams in UML show how objects interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a scenario. The processes are represented vertically, and interactions are shown as arrows. This article explains the purpose and the basics of Sequence diagrams. Also, check out this complete Sequence Diagram Tutorial to learn more about sequence diagrams. You can also instantly start drawing using our sequence diagram templates.



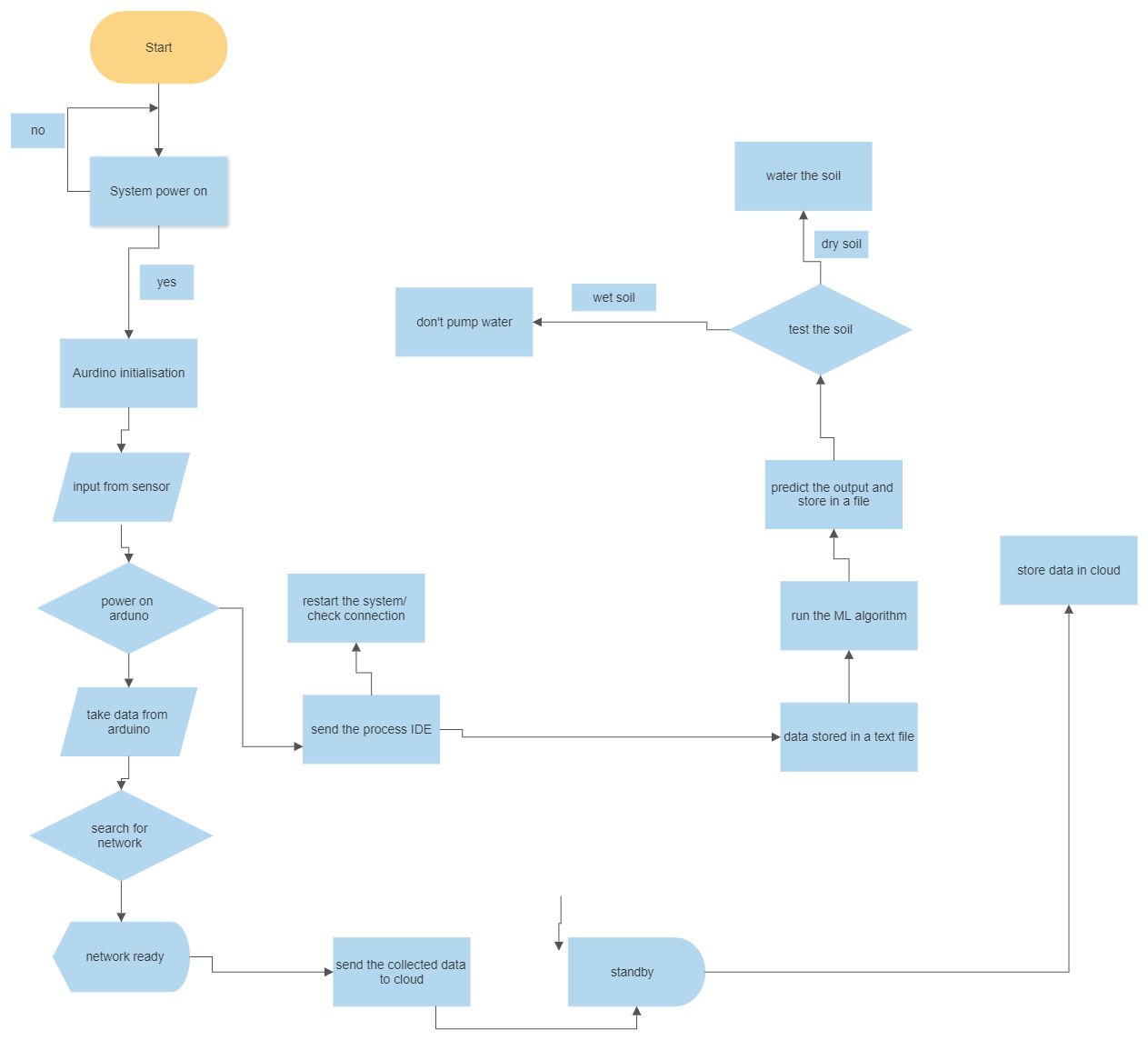
**Fig 4.3: Use case diagram**



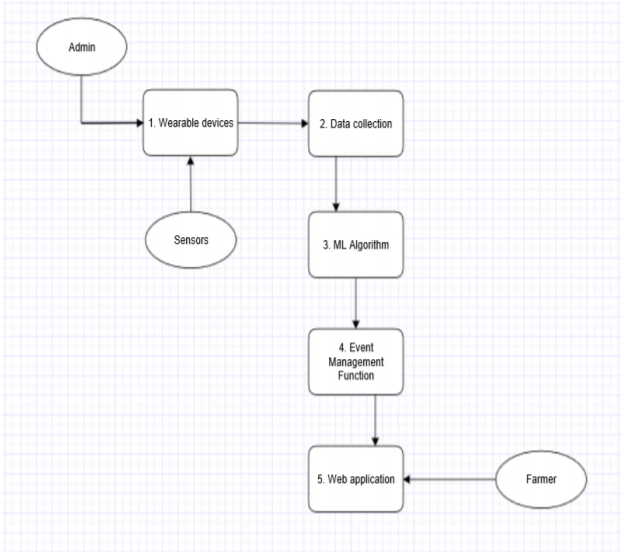
**Fig 4.4: Context diagram**



**Fig 4.5: Sequence diagram**



**Fig 4.5: Flow diagram**

****

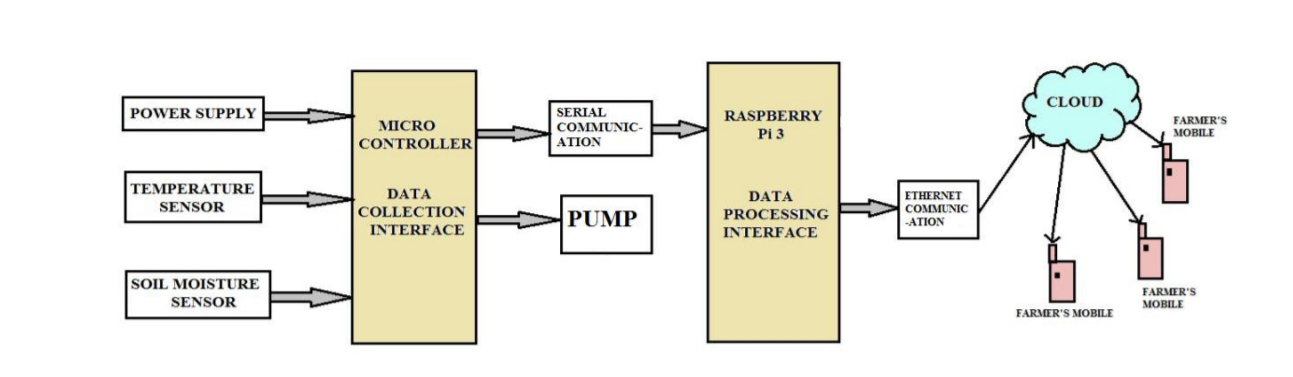
**Fig 4.5: Data Flow diagram**

# CHAPTER 5

**SYSTEM ARCHITECTURE**

## 5.1 ARCHITECTURE OVERVIEW

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



**Fig 5.1 System Architecture Diagram**

## 5.2 SYSTEM Implementation

The Sound Classification System contains different modules functions namely:

* **Step 1:**Connect the device using the Unique IP address method and pair it to Android Mobile.
* **Step 2:** It shows the status every 60 seconds.
* **Step 3:**When the moisture is less than 45 the water will pump through the plant.

Module 1: Registration & Data collection

Module 2: Implementation of algorithm

Module 3: Result

### 5.2.1 MODULES EXPLANATION

**5.1.1.1 METHODOLOGY**

Implementation of the project required the design of the system developed in the design phase of the project to be carefully implemented. The extensive implementation of automated systems in agriculture has proven to successfully reduce cost. The operation of automated agricultural system could potentially revolutionize the irrigation process and the way it has impacted the commercial & industrial sectors. Thus, this project has been an expert or non-expert-system-based method of field monitoring for detecting dryness & treatment of the field. The prototype system food and beverage industry has the potential to be useful for the industry, seeking ways to make agriculture cost effective. Furthermore, the ultimate beneficiaries of the project are the farmers who are the backbone of an agricultural economy.

**5.1.1.2 PROJECT PLAN**

The Objective of the project planning is to provide a framework that enables an owner to make reasonable estimate of the resources, cost and schedule. The project leader is responsible for designing the system precisely according to the requirement specified by the owner/ customer. He is also responsible for maintenance of the system for certain period of time, since in most cases, cost of maintenance is much higher than cost of developing the system. Thus to reduce development and maintenance cost and to provide the system within planned time, proper planning of system is necessary.

**5.1.1.3. Initial Investigation of design**:

The most crucial phase of managing system projects is planning to launch a system investigation, we need a master plan detailing the steps to be taken, the people to be questioned, and outcome expected. The initial investigation has the objective of determining whether the user’s request has potential merits the major steps are defining user requirements, studying the present system and defining the performance expected by candidate system to meet user requirements. The first step in the system development life cycle is the identification of need. There may be a user request to change, improve or enhance an existing system. The initial investigation is one way of handling these needs. The objective is to determine whether the request is valid and feasible before a 22 recommendation is reached to do nothing, improve or modify the existing system, are to build a new one. Thus for an effective test and written paper follow-up data resulting from different circumstances, it is vital to design the APIS.

5.1.1.4 **WORKING**

This project consists of two sections: the external sensor unit, and the inbuilt processing unit. In the external sensor unit, the basic requirement of sensing the moistness of the sand or soil through capacitive reactance is performed, the arms of the sensor are able to detect resistance and provide input to the IC. When the soil becomes dry, it produces large voltage drop due to high resistance, and this is sensed by the soil moisture sensor, and this resistance causes the operational amplifier to produce an output that is above the threshold value required. This causes the relay to change from normally open to closed condition – The relay becomes on. When the relay is turned on, the valve opens and water through the pipes rushes to the crops. When the water content in the soil increases, the soil resistance gets decreases and the transmission of the probes gets starts to make the operational amplifier stop the triggering of the relay. Finally the valve which is connected to the relay is stopped. Op-amp is configured here as a comparator. The comparator monitors the sensors and when sensors sense the dry condition then the project will switch on the motor and it will switch off the motor when the sensors are in wet. The comparator does the above job it receives the signals from the sensors. A transistor is used to drive the relay during the soil wet condition. 5V double pole – double through relay is used to control the water pump. LED indication is provided for visual identification of the relay / load status. A switching diode is connected across the relay to neutralize the reverse EMF. This project works with 5V regulated power supply for the internal blocks and uses regulated 12V power supply for the relay board. Power on LED is connected for visual identification of power status. First, the sensor probes are inserted in the soil at specific locations in the field, at a depth of 5cm from the soil surface at regular intervals in the field. The wiring is made with 23 protective covering so that it is not harmed by any unexpected factors like rocks in the field. Since wet soil is more conductive than dry soil, the soil moisture sensor module has a comparator in it. The voltage from the prongs and the predefined voltage are compared and the output of the comparator is high only when the soil condition is dry. When the moisture in the soil is above the threshold, the relay will be turned on. The relay coil gets energized and turns on the motor. The LED is also turned on as an indicator. The soil begins to get supplied with water, and the water content of the soil increases. When the moisture content of the soil increases and reaches the threshold value, the output of the soil moisture sensor is low and the motor is turned off. This prevents a case of over-watering

**CHAPTER 6**

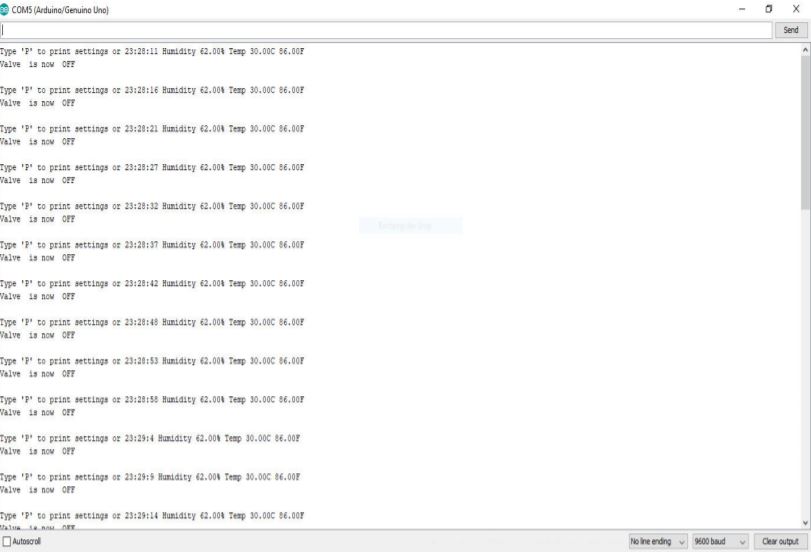
**TESTING**

#### 6.1 SYSTEM TESTING

The testing and the evaluation part are the most important part in any research or development of a model. As we need to understand and check whether our model is performing the way it should be.

#### Decision Tree

The accuracy and the classification report is as follows.

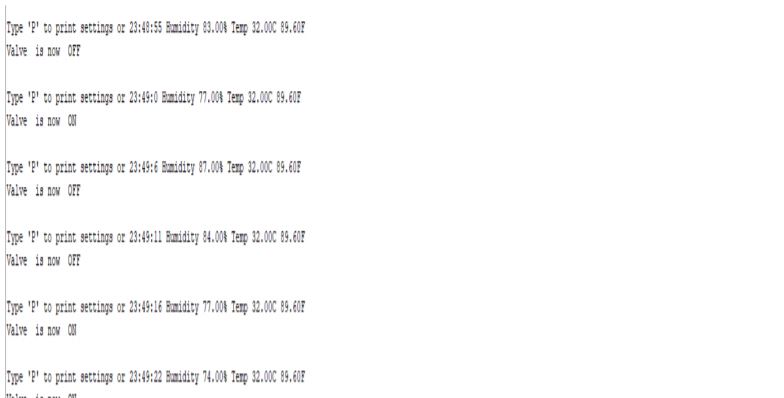


**Print Settings of the Valve**

Valve is Open When Humidity is less than 80% and Timer is Set

###### 

**Valve is Open When Humidity is less than 80% and Timer is Set**



**Valve is Closed When Humidity is More Than 80% Even When the Timer is On**

## 6.2 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

## 6.4 TEST CASES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test  Case Id | Test  Cases | Priority | Input  Test  Data | Test Case  Description | Expected Results | Actual  Results | Pass/  Fail |
| TU0  1 | User registration | A | Enter  Email-id and  Password. | Check  user with the database | User should be  available  in  database | User present in  databas e | Pass |
| TU0  2 | User Enters Unique Ip address | A | 10 digit code | Check for device in database | Training with good accuracy | Connected with your pot | Pass |
| TU0  4 | Check Plants | A | \_ | If moisture less than 45 click water | Successfu  lly  watered plant | Successfu  lly  watered plant | Pass |

## CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENT**

### 7.1 CONCLUSION

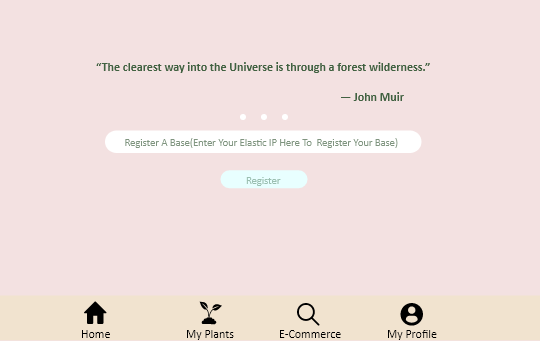
Irrigation becomes easy, accurate and practical with the idea above shared and can be implemented in agricultural fields in future to promote agriculture to next level. The output from moisture sensor and level system plays major role in producing the output. Thus the “AUTOMATIC PLANT Watering System” (APIS) has been designed and tested successfully. It has been developed by integrating all the features of all the hardware components used. Presence of every module has been reasoned above and placed carefully in order to contribute to the best working of the unit. The system has been tested to function automatically, and to the best of its ability. The moisture sensors measure the moisture level (water content) of the different plants. If the moisture level is found to be below the desired level, the moisture sensor sends the signal to the operational amplifier which triggers the DC Motor pump to turn ON and supply the water to respective field area. When the desired moisture level is reached, the system halts on its own and the DC Motor pump is turned OFF. Thus, the functionality of the entire system has been tested thoroughly and it is said to function successfully.

### 7.2 FUTURE ENHANCEMENT

The application certainly is much more advantageous than the manual system. There will be no bias in the regions being covered and the delay is kept as minimal as it can be.

* The operator does not require any previous training because of its user friendliness.
* The operator is free from any technical issues. Extremely simple design makes the circuit easy to implement and maintain.
* Alterations in the system can be done easily if the process of the working changes in future.
* In future according to the user’s requirement it can be updated to meet the user requirements.
* Smart Wifi Irrigation Controllers are next generation controllers that adjust your irrigation system automatically using real-time weather information. Moreover, you can control it from anywhere, anytime.

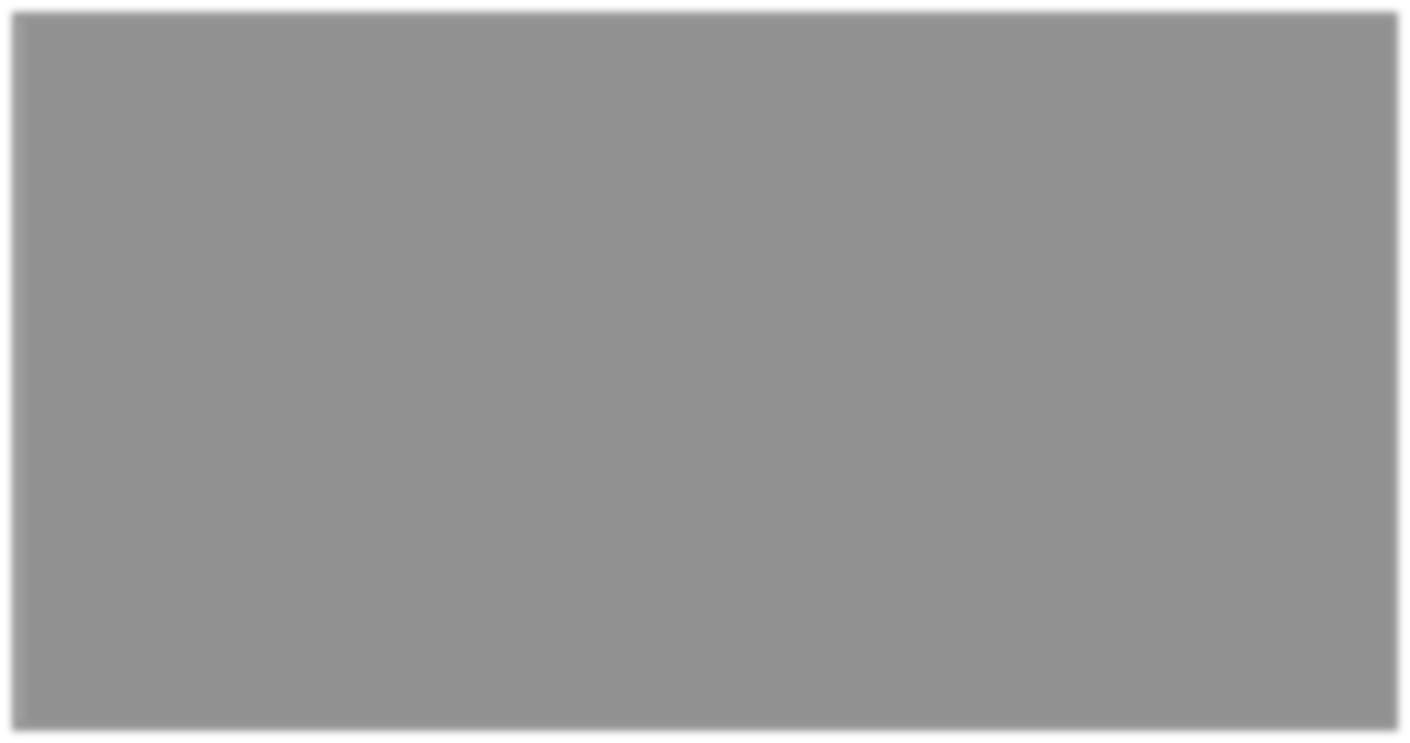
### APPENDICES



**A 1. SAMPLE SCREENS**

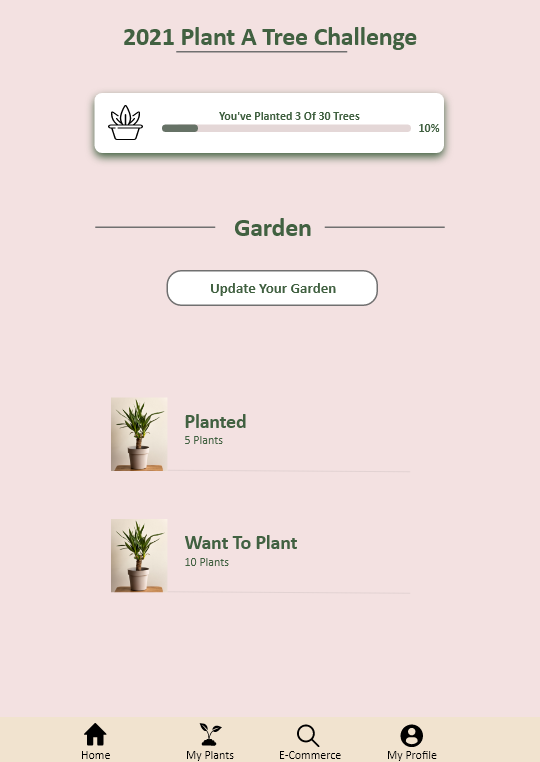
**Fig 1**

**Home Page**





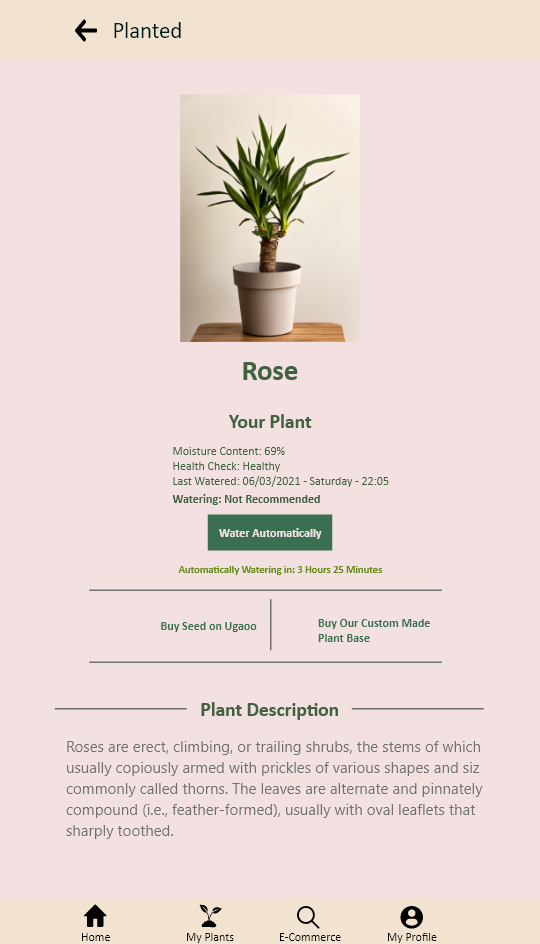
**Fig 2: Successfully Registered Page**



**Fig 3: Dashboard**

****

**Fig 4:Planted**

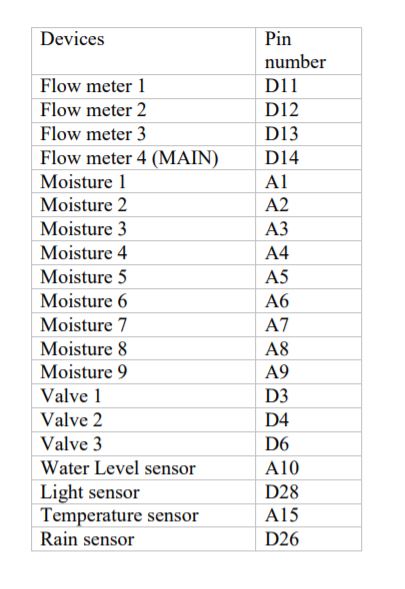


**Fig 5: Description of plant**

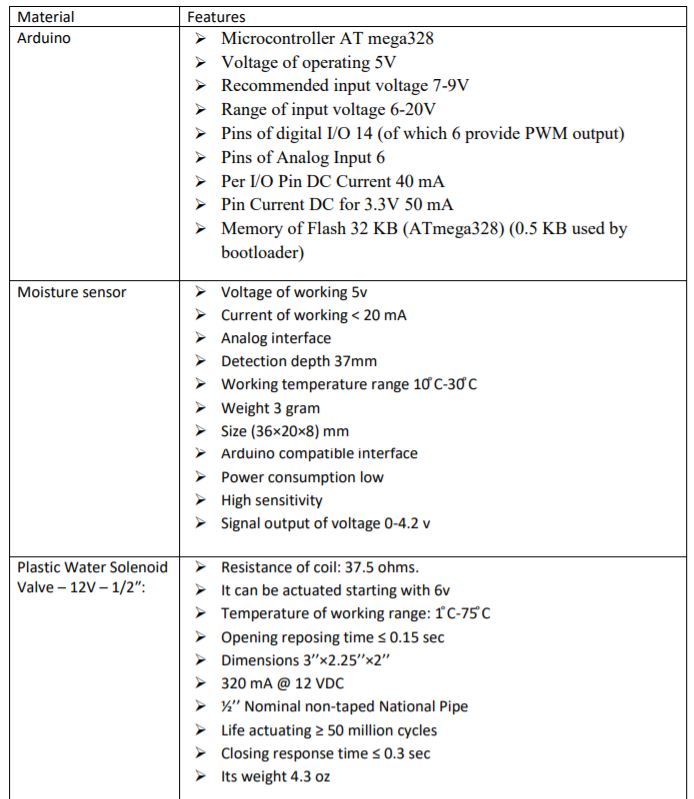


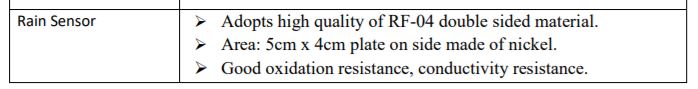
**Fig 6 : Ecommerce Site**

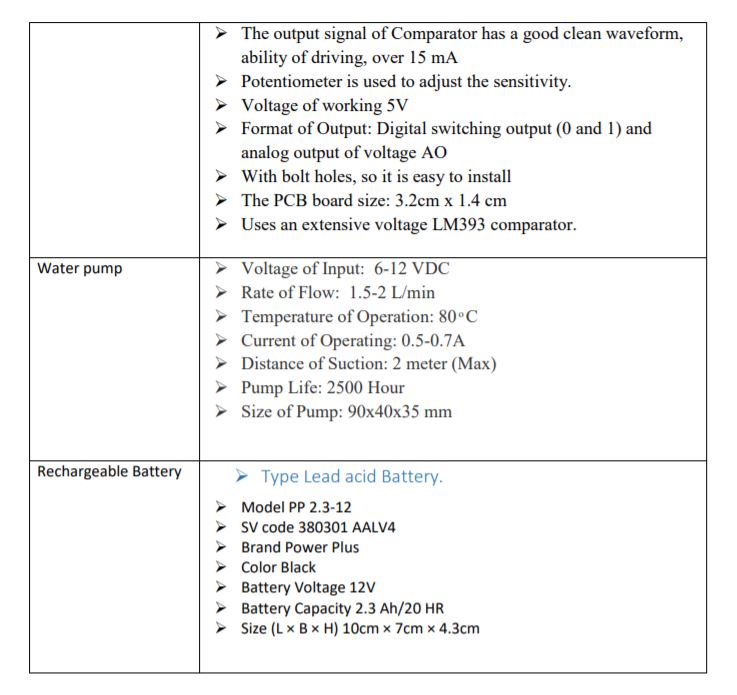
**Appendix 1: Wiring connection between devices and Arduino’s pins.**



**Appendix 2: Material features that have been used in this project.**

****

****

****

**A2. SAMPLE CODE**

### Arduino Code #1

#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

int Flow1 = 11;

int Flow2 = 12;

int Flow3 = 13;

int Flow4 = 14;

int Received = 0;

const int mo1 = 1; // analog

const int mo2 = 2; // analog

const int mo3 = 3; // analog

const int mo4 = 4; // analog

const int mo5 = 5; // analog

const int mo6 = 6; // analog

const int mo7 = 7; // analog

const int mo8 = 8; // analog

const int mo9 = 9; // analog

int pump = 2;

const int vlv = 3; // digital

const int vlv2 = 4; // digital

const int vlv3 = 6; // digital

50

const int water\_level = 10; // analog

const int Temp = 15; // analog

const int light = 28; // digital

const int rain = 26; // digital

void setup() {

// put your setup code here, to run once:

analogWrite (water\_level, LOW);

pinMode (vlv, OUTPUT);

digitalWrite (vlv, LOW);

pinMode (vlv2, OUTPUT);

digitalWrite (vlv2, LOW);

pinMode (vlv3, OUTPUT);

digitalWrite (vlv3, LOW);

pinMode (pump, OUTPUT);

digitalWrite (pump, LOW);

pinMode (light, INPUT);

digitalWrite (light, LOW);

pinMode (rain, INPUT);

digitalWrite (rain, LOW);

51

lcd.begin(16, 4);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

//Tempreture Sensor

// int reading = analogRead(Temp);

// float voltage = reading \* 5.0;

// voltage /= 1024.0;

// Serial.print(voltage); Serial.println(" volts");

// float temperatureC = (voltage - 0.5) \* 10 ;

// Serial.print(temperatureC); Serial.println(" degrees C");

//delay(1000);

int moisture1 = analogRead (mo1);

Serial.println("mo 1");

Serial.println(moisture1);

int moisture2 = analogRead (mo2);

52

// Serial.println("mo 2");

// Serial.println(moisture2);

int moisture3 = analogRead (mo3);

// Serial.println("mo 3");

// Serial.println(moisture3);

int moisture4 = analogRead (mo4);

// Serial.println("mo4");

// Serial.println(moisture4);

int moisture5 = analogRead (mo5);

// Serial.println("mo 5");

// Serial.println(moisture5);

int moisture6 = analogRead (mo6);

// Serial.println("mo 6");

// Serial.println(moisture6);

int moisture7 = analogRead (mo7);

// Serial.println("mo 7");

// Serial.println(moisture7);

int moisture8 = analogRead (mo8);

// Serial.println("mo 8");

// Serial.println(moisture8);

int moisture9 = analogRead (mo9);

// Serial.println("mo 9");

// Serial.println(moisture9);

int level = analogRead (water\_level);

// Serial.print("LEVEL:");

// Serial.println(level);

int light1 = digitalRead (light);

// Serial.print("light:");

// Serial.println(light1);

int rain1 = digitalRead (rain);

// Serial.print("rain:");

// Serial.println(rain1);

// 1, 2 , 3

if (moisture1 > 800 && moisture2 > 800 || moisture1 > 800 && moisture3 > 800 || moisture2 >

800 && moisture3 > 800)

{

digitalWrite(vlv, HIGH);

digitalWrite (pump, HIGH);

}

if (moisture1 > 800 && moisture2 > 800 && level > 800 || moisture1 > 800 && moisture3 > 800

&& level > 800 || moisture2 > 800 && moisture3 > 800 && level > 800)

{

digitalWrite(vlv, LOW);

digitalWrite (pump, LOW);

}

if (moisture1 > 800 && moisture2 > 800 && rain1 == 0 || moisture1 > 800 && moisture3 > 800

&& rain1 == 0 || moisture2 > 800 && moisture3 > 800 && rain1 == 0)

{

digitalWrite(vlv, LOW);

digitalWrite (pump, LOW);

}

if (moisture1 < 700 && moisture2 < 700 && moisture3 < 700)

{

digitalWrite(vlv, LOW);

// digitalWrite (pump, LOW);

}

//4,5,6

if (moisture4 > 800 && moisture5 > 800 || moisture4 > 800 && moisture6 > 800 || moisture5 >

800 && moisture6 > 800 && Temp < 20 && light == 1)

{

digitalWrite (pump, HIGH);

digitalWrite(vlv2, HIGH);

}

if (moisture4 > 800 && moisture5 > 800 && level > 800 || moisture4 > 800 && moisture6 > 800

&& level > 800 || moisture5 > 800 && moisture6 > 800 && level > 800 )

{

digitalWrite (pump, LOW);

digitalWrite(vlv2, LOW);

}

if (moisture4 > 800 && moisture5 > 800 && rain1 == 0 || moisture4 > 800 && moisture6 > 800

&& rain1 == 0 || moisture5 > 800 && moisture6 > 800 && rain1 == 0 )

{

digitalWrite (pump, LOW);

digitalWrite(vlv2, LOW);

}

if (moisture4 < 700 && moisture5 < 700 && moisture6 < 700)

{

digitalWrite(vlv2, LOW);

}

// 7,8,9

//

if (moisture7 > 800 && moisture8 > 800 || moisture7 > 800 && moisture9 > 800 || moisture8 >

800 && moisture9 > 800 && Temp < 20 && light == 1)

{

digitalWrite (pump, HIGH);

digitalWrite(vlv3, HIGH);

}

if (moisture7 > 800 && moisture8 > 800 && level > 800 || moisture7 > 800 && moisture9 > 800

&& level > 800 || moisture8 > 800 && moisture9 > 800 && level > 800 )

{

digitalWrite (pump, LOW);

digitalWrite(vlv3, LOW);

}

if (moisture7 > 800 && moisture8 > 800 && rain1 == 0 || moisture7 > 800 && moisture9 > 800

&& rain1 == 0 || moisture8 > 800 && moisture9 > 800 && rain1 == 0 )

{

digitalWrite (pump, LOW);

digitalWrite(vlv3, LOW);

}

if (moisture7 < 700 && moisture8 < 700 && moisture9 < 700)

{

digitalWrite(vlv3, LOW);

}

//Tempreture & light & rain

//

// if (Temp > 20 && light ) || moisture1 > 800 && moisture3 > 800 || moisture2 > 800 &&

moisture3 > 800)

// {

//

// digitalWrite(vlv, HIGH);

// digitalWrite (pump, HIGH);

// }

float sensorF4 = 0;

float sensorF3 = 0;

float sensorF2 = 0;

float sensorF1 = 0;

sensorF4 = analogRead(Flow4);

sensorF3 = analogRead(Flow3);

sensorF2 = analogRead(Flow2);

sensorF1 = analogRead(Flow1);

lcd.print("Flow2: "); lcd.print(sensorF2 / 1000); lcd.print("L/S"); lcd.println (" ");

lcd.print("Flow3: "); lcd.print(sensorF3 / 1000); lcd.print("L/S"); lcd.println (" ");

lcd.print("Flow4: "); lcd.print(sensorF4 / 1000); lcd.print("L/S"); lcd.println (" ");

lcd.print("Flow1: "); lcd.print(sensorF1 / 1000); lcd.print("L/S"); lcd.println (" ");

if (Serial.available() > 0)

{

Received = Serial.read();

}

if (Received == '1') {

digitalWrite(vlv, HIGH);

digitalWrite (pump, HIGH);

}

if (Received == '0') {

digitalWrite(vlv, LOW);

digitalWrite (pump, LOW);

}

if (Received == '2') {

digitalWrite(vlv2, HIGH);

digitalWrite (pump, HIGH);

}

if (Received == '3') {

digitalWrite(vlv2, LOW);

digitalWrite (pump, LOW);

}

if (Received == '4') {

digitalWrite(vlv3, HIGH);

digitalWrite (pump, HIGH);

}

if (Received == '5') {

digitalWrite(vlv3, LOW);

digitalWrite (pump, LOW);

}

if (Received == '6') {

digitalWrite (pump, HIGH);

digitalWrite(vlv, HIGH);

digitalWrite(vlv2, HIGH);

digitalWrite(vlv3, HIGH);

}

if (Received == '7') {

digitalWrite (pump, LOW);

digitalWrite(vlv, LOW);

digitalWrite(vlv2, LOW);

digitalWrite(vlv3, LOW);

}

}

**Arduino Code #2 (Connecting with Mobile Application)**

#include<SoftwareSerial.h>

SoftwareSerial firstSerial(15,16);

int led = 5;

int pin=A0;

int moisture=0;

String readString;

void setup()

{

Serial.begin(9600);

firstSerial.begin(9600);

Serial.begin(9600);

pinMode(led, OUTPUT);

delay(100); // Baud Rate

}

void loop()

{

while (Serial.available()) //Send data only when you receive data

{

delay(4); //delay time

char c = Serial.read();

readString += c;

}

if (readString.length() > 0)

{

Serial.println(readString);

if (readString == "ON")

{

digitalWrite(led, HIGH); // LED ON

}

if (readString == "OFF")

{

digitalWrite(led, LOW); //LED OFF

}

readString = "";

}

int soilmoisture=analogRead(pin);

if(soilmoisture>=100){

digitalWrite(led,HIGH);

firstSerial.println("AT+CMGF=1");

delay(1000);

firstSerial.println("AT+CMGS=\"+xxxxxxxxxx\"\r"); //Enter Your Mobile Number instead XXXX while Testing

delay(1000);

firstSerial.println("Hai there I need some water");

delay(100);

firstSerial.println((char)23);

delay(1000);

}

else{

digitalWrite(led,LOW);

}

Serial.println(soilmoisture);

Serial.print("%");

delay(20);

}

**Arduino Code #3 (LED Lights)**

int blue = 0; // for incoming serial data

int moisture\_sensor = A0;

int moisture;

int limit = 500;

int pump = 3;

int led = 13;

int pump\_time = 3000;

void setup()

{

Serial.begin(9600); // opens serial port, sets data rate to 9600 bps

pinMode(moisture\_sensor,INPUT);

pinMode(pump,OUTPUT);

pinMode(led,OUTPUT);

}

void loop()

{

moisture = analogRead(moisture\_sensor);

Serial.print("Moisture=");

Serial.print(moisture);

Serial.println();

while(Bluetooth()==1)

{

Pump(1);

blue = 0;

break;

}

while(Bluetooth()==0)

{

if(moisture<limit)

{

Pump(1);

break;

}

else

{

Pump(0);

break;

}

}

}

int Bluetooth()

{

// send data only when you receive data:

if(Serial.available() > 0)

{

// read the incoming byte:

blue = Serial.read();

// say what you got:

Serial.print("Bluetooth");

Serial.println(blue);

if(blue == 48)

return 0;

if(blue == 49)

return 1;

}

else

return 0;

}

void Pump(int stat)

{

if(stat == 0)

{

digitalWrite(pump,stat);

digitalWrite(led,stat);

Serial.println("Pump OFF");

}

if(stat == 1)

{

digitalWrite(pump,stat);

digitalWrite(led,stat);

Serial.println("Pump ON");

delay(pump\_time);

}

}

**A.3 Result & Documentation**

**Data Analysis**

I recorded the process of manual irrigations system of three types of plants (lemon, mint, and mango) for three weeks. The irrigation process takes place twice a week in the morning and the other day in the evening. To analyze data I take data of chosen plant twice a day one between 9 am - 5 pm and the second one between 6pm-11pm. The line graph in (figure 34) shows the humidity data of the mint. Y-axis shows the number

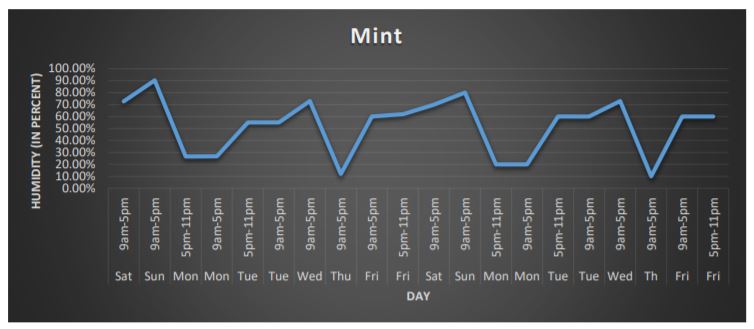
of humidity and X-axis shows the day and time. Sunday and Wednesday have the largest amount

of humidity which is 900-800. When the humidity is more than 800 means the plant is dry and need

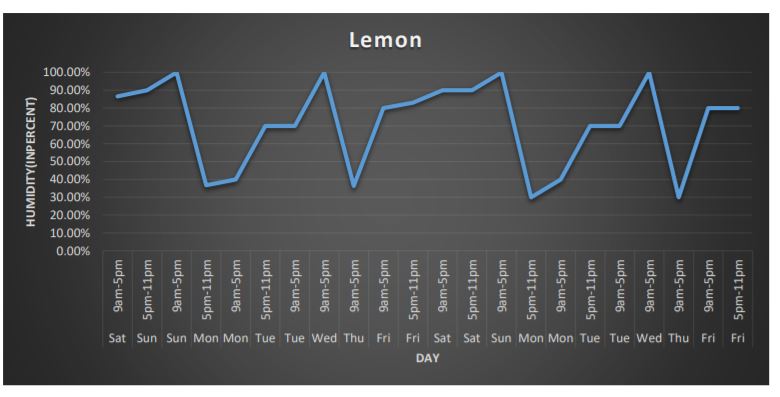
to be irrigated. The number of humidity decreases on Monday in the evening and Thursday in the

morning and reach 200-120 which mean the plant is not dry and does not need watering. From this

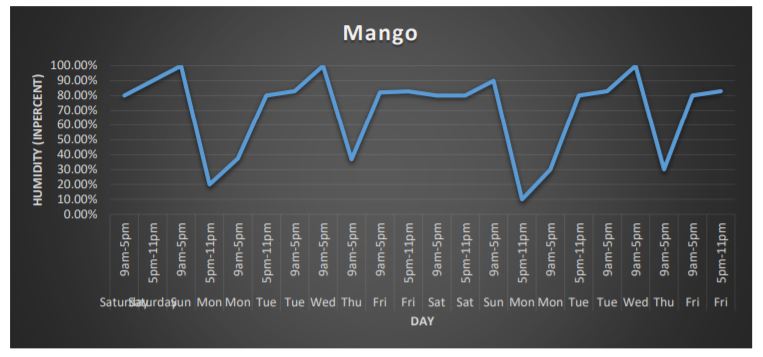
data, we can say that the mint gradually dries up and it maintains water more than other plants



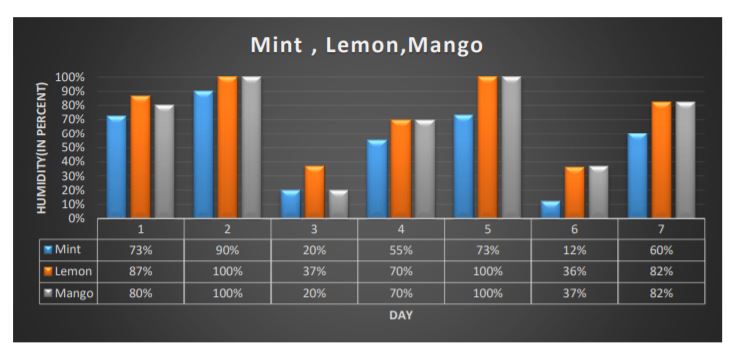
The line graph in shows the humidity data of lemon y-axis shows the number of humidity and x-axis shows the day and time. Sunday and Wednesday have the largest number which is 1000 which mean the plant is very dray and it needs to be irrigated as soon as possible. On Monday and Thursday the humidity decreased but what I notice from this data than the lemon needs to irrigate more than twice a week because it loses water faster. So we can say that lemon needs water more than mint.



The line graph in shows the humidity data of mango. Sunday, Thursday, and Saturday have the largest number of humidity which means the plant is very dray. We can see also there is rapid decrease and rapid increase in the amount of moisture which mean mango lose water very fast and it needs to irrigate more than twice a week.



Through the data we can observe the following in (Figure33), All the plants were watered at one time and in equal quantities almost twice a week, but after comparing the results we found that the lemon and mango needed to water more than two times a week because the humidity ratio is significantly reduced taking into account the temperature and the time of irrigation. When they irrigate in the morning plants lose humidity quickly and very significantly because they irrigate at the wrong time so the sun evaporates water much faster. As for the mint plant through the results, we found that this plant that does not lose moisture quickly as it can withstand a longer period of mango and lemon. Through these results, we can conclude by saying that using automatic irrigation system is more efficient because it solves almost all the problems of using the manual system. Automatic irrigation system works based on the needs of plants, so it solves the problem of irrigation in wrong time by using light and temperature sensors together.



**Discussion**

After completing the plan and the collection of the components of the smart Irrigation system, it's been met the goal. Also, all of the requirements were implemented in order to finish this smart Irrigation system, so that it becomes full production and finalize. After that, the system became tested, and the end result became as required.

The system will now not work until two or three of moisture sensors from any line of the 3 fields send a sign to the Arduino that the soil is dry and crop needs water. After the sign reaches the Arduino, it will send a command to the relay of that specific line field valve to be energized to open the valve and a command to the relay of the pump to exchange it directly to irrigate that field. Also, all 3 fields can be irrigated at a same of time, if two or three of all 3 plants moisture sensor are activated. So, all solenoid valve relays may be energized to open all valves and the pump will run to irrigate all 3 plants.

There has been a problem at the beginning to choose a suitable pump to paintings to irrigate all flowers at the same moment. The program of the system has been configured and the system will no longer operate unless (or three moisture sensors) are activated. But if one sensor is activated of any lines the system will no longer perform, because that sensor can also it be defective. If the water tank level is low the system will not operate in any respect, even all plant sensors are activated to protect the water pump.

Furthermore, this smart Irrigation system has been configured that if there is a raining, it's going to not work, because the raining sensor will activate, and it will send a sign to Arduino to stop the water pump and to close all valves too. Further, on the daytime the system will no longer work, due to the mild sensor will activate at the daytime and that will reason to close the plant's value as well as to switch OFF the pump. For the system programming, it has been precisely chosen as stated in previous chapters to apply UNO Arduino, the wires connections from the controllers to the Arduino have been pretty difficult, due to a single mistake can damage any electric element.

It was not easy to program the smart Irrigation system and upload it in Arduino to run the water pump and starting valves with eighteen sensors, but with the assist of Arduino library, this system turned into completed with best results. Connecting the wires from 20 devices to the Arduino become very hard and complicated, but by means of using the plastic breadboard became to facilitate the connection of those wires.

For designing the plant, special flexible pipes had been used to facilitate the connection from the water tank to the plant, however, we faced problem to connect pipes together. So, two types of pipe joints (T-joint and L-joint) were used to clear up this issue.

### REFERENCES

[1] Klute, A. (ed.), 1986: Methods of Soil Analysis, Part 1: Physical a nd Mineralogical Methods. American Society of Agronomy, Madison, Wisconsin, United States, 1188 pp.

[2] Knight, J.H., 1992: Sensitivity of time domain reflectometry measurements to lateral variations in soil water content. Water Resources Research, 28, pp. 2345–2352.

[3] Magagi, R.D., Kerr, Y.H., 1997. Retrieval of soil moisture and vegetation characteristics by use of ERS-1 wind scatterometer over arid and semi-arid areas. Journal of Hydrology 188-189, 361–384.

[4] Marthaler, H.P., W. Vogelsanger, F. Richard and J.P. Wierenga,1983: A pressure transducer for field tensiometers. SoilScience Society of America Journal, 47, pp. 624– 627.

[5] Attema, Evert, Pierre Bargellini, Peter Edwards, Guido Levrini, SveinLokas, Ludwig Moeller, BetlemRosich-Tell, et al 2007. Sentinel-1 - the radar mission for GMES operational land and sea services. ESA Bulletin 131: 10-17.

[6] Bircher, S., Skou, N., Jensen, K.H., Walker, J.P., & Rasmussen, L.(2011). A soil moisture and temperature network forSMOS validation in Western Denmark. Hydrol. Earth Syst. Sci. Discuss.,8, 9961-10006.

[6] ADVERSE IMPACTS OF DROUGHT ON CROPS AND CROP PRODUCERS IN THE WEST James Johnson and Vince Smith Montana State University Department of Agricultural Economics and Economics <http://ageconsearch.umn.edu/bitstream/27974/1/02010009.pdf>

[7] How Drought and Extreme Heat Are Killing the World’s Crops - Justin Worlan

http://time.com/4170029/crop-production-extreme-heat-climate-change/

[9] SSRG, S. (2017). Engineering Science and Technology Journals, SSRG International Journal. [online]

Internationaljournalssrg.org. Available at: http://www.internationaljournalssrg.org

[10] Scribd. (2017). Automatic Irrigation System on Sensing Soil Moisture Content | Irrigation | Soil.

[online] Available at: https://www.scribd.com/document/362464538/Automatic-Irrigation-System-onSensing-Soil-Moisture-Content.

[11] Arresearchpublication.com. (2017). Cite a Website - Cite This For Me. [online] Available at:

http://www.arresearchpublication.com/images/shortpdf/1478954748\_161\_ijeee .pdf

[12] Ijcit.com. (2017). Cite a Website - Cite This For Me. [online] Available at:

https://www.ijcit.com/archives/volume4/issue3/Paper040304 .pdf

[13] Anon, (2017). Embedded Systems and Robotics with Open Source Tools. [online] Available at:

https://vigyanashram.files.wordpress.com/2015/05/plant-watering-system .pdf

[ 14] Vagulabranan, R., Karthikeyan, M., & Sasikala, V. (2016). Automatic Irrigation System on Sensing

Soil Moisture Content. International Research Journal of Engineering and Technology (IRJET), 3.

[15]Dr. Al Humairi, A. (2016). Introduction to Arduino. Embedded Systems Course Material.

[16] Arduino.cc. (2018). Arduino - LiquidCrystal:

https://www.arduino.cc/en/Reference/LiquidCrystal .

[17] Arduino Project Hub. (2018). Arduino Bluetooth Basic Tutorial:

https://create.arduino.cc/projecthub/user206876468/arduino-bluetooth-basic-tutorial-d8b737 .

[18] Arduino.cc. (2018). Arduino – SD

https://www.arduino.cc/en/Reference/SD

[19] Psomas, S. (2007). The Five Competencies of User Experience Design :: UXmatters.

Uxmatters.com. Retrieved 16 May 2017, from http://www.uxmatters.com/mt/archives/2007/11/the-fivecompetencies-of-user-experience-design.php

[20] Arduino.cc. (2018). Arduino – Introduction:

https://www.arduino.cc/en/Guide/Introduction .

[21] Instructables.com. (2018). Arduino Soil Moisture Sensor: http://www.instructables.com/id/ArduinoSoil-Moisture-Sensor/

[22] (White), B., Pack), J., TMP36, T., SMD, A. and Arduino, S. (2018). SIK Experiment Guide for

Arduino - V3.2 - learn.sparkfun.com. [online] Learn.sparkfun.com.:

https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v32/experiment-7-reading-atemperature-sensor .

[23] Electronics Hub. (2018). Arduino Light Sensor:

https://www.electronicshub.org/arduino-light-sensor/ .

[24] Instructables.com. (2018). WATER LEVEL INDICATOR USING ARDUINO:

http://www.instructables.com/id/WATER-LEVEL-INDICATOR-USING-ARDUINO/ .

[25] Goodluckbuy.com. (2018). 12V DC Cooling Brushless Motor Water Pump 840mA, 4.0M $24.10

Free Shipping @GoodLuckBuy.com. [online]: http://www.goodluckbuy.com/12v-dc-cooling-brushlessmotor-water-pump-840ma-4-0m.html [Accessed 17 May 2018].

[26] Appinventor.mit.edu. (2018). About our new MIT App Inventor logo | Explore MIT App Inventor.

[online] Available at: http://appinventor.mit.edu/explore/blogs/karen/2017/08/about.html [Accessed 23

May 2018].

[ 27] http://www.worldbank.org/en/news/feature/2012/05/17/india-agriculture-issues-priorities

[28] Clint Richards, T. (2018). Japan’s Agriculture Dilemma. Available at:

https://thediplomat.com/2014/09/japans-agriculture-dilemma/