SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE & TECHNOLOGY
P.O. Box 345004, Block 10, 6th Floor, Dubai International Academic City, Dubai - U.A.E.
Tel : +971 4 366 4601, Fax : +971 4 366 4607, Email: info@szabist.ac.ae, Web : www.szabist.ac.ae

Design and Development of an IoT Empowered Plant Care System: Climate Adaptive Gardening with Optimal Plant Selection and Water Management

by

Meryum Tahir, Abeeha Imran and Fatimah Munir

A Final Year II Project Presented to The Faculty of Computing and Engineering Sciences

In Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Science (Computer Science) in the Faculty of Computing and Engineering Sciences

Spring 2024

SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE & TECHNOLOGY
P.O. Box 345004, Block 10, 6th Floor, Dubai International Academic City, Dubai - U.A.E.
Tel : +971 4 366 4601, Fax : +971 4 366 4607, Email: info@szabist.ac.ae, Web : www.szabist.ac.ae

APPROVAL SHEET

This Final Year Project II entitled **"Design and Development of an IoT Empowered Plant Care System: Climate Adaptive Gardening with Optimal Plant Selection and Water Management"** presented by **Meryum Tahir, Abeeha Imran** and **Fatimah Munir** with registration number **171234**, **201203** and **201204** in partial fulfillment of the requirements for the degree of **Bachelor of Science (Computer Science) in the Faculty of Computing and Engineering Sciences**, has been evaluated and approved by the Panel of Evaluators.

PANEL OF EVALUATORS

Dr. Rukshanda Jabeen
Advisor

Mr. Muhammad Ali Awan
Juror

Mr. Mohamed Ahamed Amani, MPhil
Program Manager

Dr. Hummayoun Naeem
Research Coordinator

Accepted and approved in partial fulfillment of the requirements for the degree of Bachelor of Science (Information Technology) in the Faculty of Computing.

Dr. Hummayoun Naeem
Head of Campus

Date: ______________

SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE & TECHNOLOGY
P.O. Box 345004, Block 10, 6th Floor, Dubai International Academic City, Dubai - U.A.E.
Tel : +971 4 366 4601, Fax : +971 4 366 4607, Email: info@szabist.ac.ae, Web : www.szabist.ac.ae

Dissertation Copyright Form

As a part of the copyright policy of SZABIST and in compliance with the policy, I **Meryum Tahir**, **Abeeha Imran** and **Fatimah Munir** delegate my rights of this piece of work in favor of SZABIST-Dubai. I assure that the content of this dissertation is the direct result of my own work.

In addition, I am aware that one copy of this dissertation will be submitted in the Library of SZABIST-Dubai for future use by SZABIST students, faculty and other fellows visiting the campus.

In case of sending part or whole of my present research work on the topic of "**Design and Development of an IoT Empowered Plant Care System: Climate Adaptive Gardening with Optimal Plant Selection and Water Management"** at the department of **Computing and Engineering Sciences** in semester **Spring**, year **2024** for (single author) publication by any means, I would require a "No Objection Certificate or NOC" on the prescribed form from the SZABIST Research Office. In case of joint publication with supervisor (only), NOC in written may not be required.

In addition I am authorizing SZABIST Dubai for reproducing, distributing and transmitting part or whole of the dissertation by any means including photocopying, recording, or other electronic (online) or mechanical methods except in the case of brief excerpts or quotations embodied in critical reviews and certain other noncommercial uses permitted by the copyright policy.

Signature (Student/Researcher)
Full Name: Meryum Tahir
Registration No. 171234
Program: BSCS Meryum Tahir

Signature (Student/Researcher)
Full Name: Abeeha Imran
Registration No. 201203
Program: BSCS Abeeha Imran/

Signature (Student/Researcher)
Full Name: Fatimah Munir
Registration No. 201204
Program: BSCS Fatimah Munir/

SZABIST, Dubai International Academic City (DIAC), Block 10, Floor, 6. P.O Box # 345004, Tel: +97143664601 Fax: +97143664607 Email: info@szabist.ac.ae

Document Information

Category	Information
Customer	Home Gardeners
Project Title	Smart Plant Care System
Document	FYP Report
Document Version	1.0
Status	Final
Authors	Abeeha Imran, Fatimah Munir, Meryum Tahir
Approvers	Dr Rukshanda Kamran

Definition of Terms, Acronyms and Abbreviations

PCS	Plant Care System
VNF	Vegetation and Flora
IoT	Internet of Things
SRS	Software Requirements Specification
API	Application Programming Interface
GUI	Graphical User Interface
UX	User Experience
DBMS	Database Management System
SDK	Software Development Kit
FAQs	Frequently Asked Questions

TABLE OF CONTENTS

List of Figures

List of Tables

Acknowledgements

We have made efforts in this project; however, it would not have been possible without the kind support and help of many individuals. We would like to thank all of them. First of all, we are highly indebted to our supervisor, Dr. Rukshanda Kamraan, and would like to express our deepest gratitude to her for providing us with this golden opportunity to develop an "IoT empowered Smart Plant Care System" and for extending her expertise towards us. She was always available to answer our questions and her generous support and constant guidance helped us to complete this project successfully. Moreover, we would also like to thank Mr. Ahmed Amani, Ms. Bushra and Mr. Ali Awan for evaluating our project and for providing us with honest feedback that helped us to improve our project.

Secondly, we are grateful to all the authors for their online publications that have provided us with knowledge about various technologies related to electricity prediction models.

Lastly, our thanks and appreciation also go to all our friends and teachers who willingly helped us to develop this report with their feedback. When asked, they gave us their reviews of our project, and this helped us to identify and rectify the mistakes we made in our project. Their healthy criticism has been a constant motivation for us.

Abstract

In an era marked by the convergence of technology and environmental consciousness, the cultivation of green spaces is taking on a smarter and more sustainable dimension. This research explores the integration of IoT technology into a Smart Plant Care System, a technology-driven solution designed to optimize plant selection, water management and overall gardening practices. This project aims to revolutionize the way we nurture and sustain our gardens.

The core objectives of this research encompass the design and implementation of a comprehensive Smart Plant Care System that harmonizes with prevailing conditions. By coupling sensor data with plant-specific requirements, the system enables users to make informed decisions on plant selection and irrigation strategies. The fusion of sensor technology and user connectivity not only enhances plant health but also contributes to water conservation efforts in an era where resource efficiency is paramount.

The study dives into the technical intricacies of sensor deployment, data acquisition, and user interface development within the Smart Plant Care System. The integration of real-time sensor data provides insights into current environmental conditions, allowing users to adapt their gardening practices proactively. Through an exploration of adaptive watering schedules, this research seeks to empower gardeners with the tools needed to thrive in an ever-changing climate.

The results of this investigation hold the promise of sustainable, adaptive gardening practices that align with the natural world's rhythm. By embedding sensor capabilities within the Smart Plant Care System, the research aims to nurture green spaces that not only thrive today but also remain in trend in the foreseeable future. The project's findings offer a glimpse into the possibilities of IoT-driven plant care, fostering resilience and resourcefulness in our botanical endeavors.

Chapter One: Content and Preliminary Investigation

The surge of technological advancements nowadays has led to an escalating demand for innovative solutions in various sectors. Rapid urbanization, growing awareness of environmental sustainability, and increased reliance on "smart" technology have significantly influenced the development of intelligent systems. [1] Our IoT-Empowered plant care system addresses these trends by integrating advanced sensors and automation to monitor and optimize plant health. By leveraging IoT technology, this system provides real-time data and insights, ensuring efficient water usage and ideal growth conditions for plants.

1.1 Project Selection

The selection of the IoT Empowered Plant Care System project was driven by the growing need for innovative solutions in plant care, especially in regions with harsh climates and transient populations. Around the globe, plant care practices vary widely, but there is a noticeable trend towards integrating advanced technology to manage plant health and maintenance more effectively. [2] This shift is particularly relevant in the UAE, where nearly 70% of residents leave the country to escape the summer heat, often leaving their plants unattended. [3]

To further investigate, we surveyed people anonymously to understand their plant care practices and opinions on the SPCSs currently available in the market. Most respondents were from the Middle East and South Asia. The climates in these regions range from the arid desert conditions of the UAE to the more varied but often extreme climates of Pakistan and India. Countries such as these encounter significant challenges in maintaining plant health due to extreme temperatures, erratic rainfall patterns, and the demanding lifestyles of residents.

The findings from the survey are as follows:

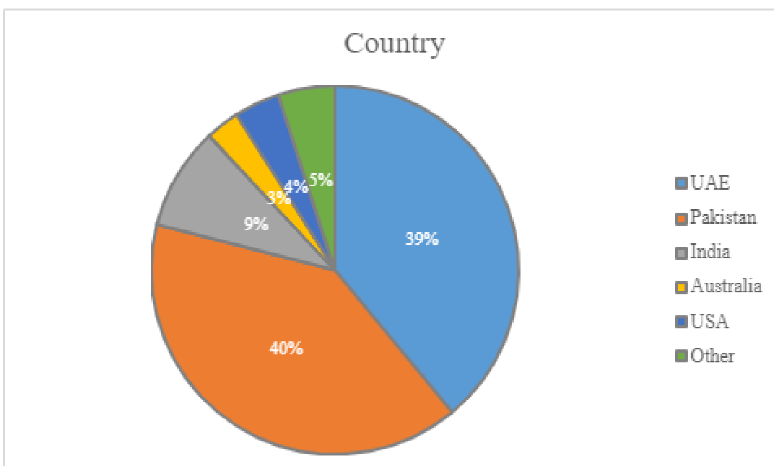


Figure 1.1: Pie Chart of Country of respondents
Source: Author

As mentioned before, most responses came from UAE and Pakistan.

When asked about their experience with home gardening, the majority were not proficient in it, as can be seen in the graph below.

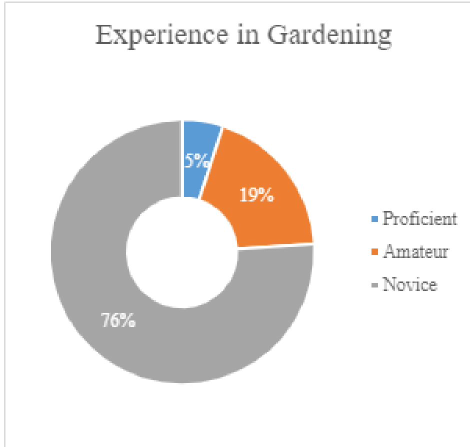


Figure 1.2: Pie Chart of Experience in Gardening
Source: Author

We asked the participants if they felt it was important to keep plants at home and if so, then what they believed was the reason for their importance. The following bar chart visualizes the common answers:

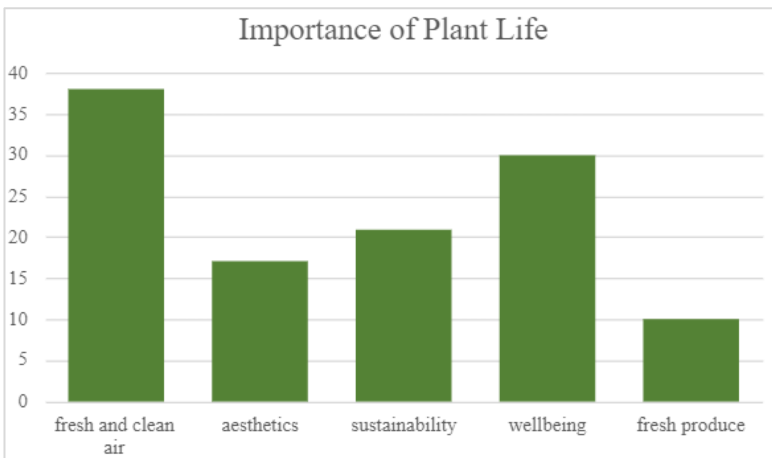


Figure 1.3: Bar Chart of Importance of Plant Life
Source: Author

The respondents were then asked if they had plants at home. Here was the response:

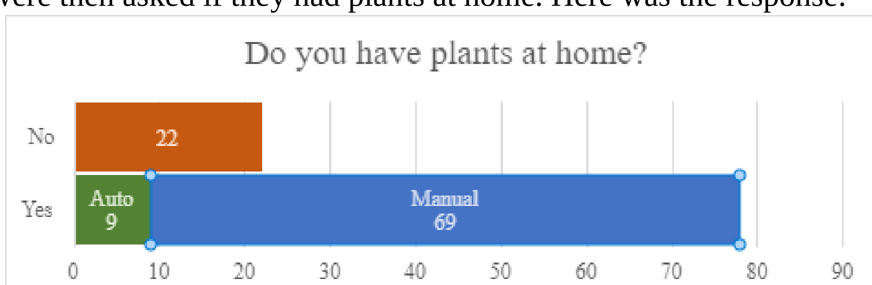


Figure 1.4: Chart of number of plant owners
Source: Author

Out of 100 respondents, 78 of them had plants at home. Out of those, 9 of them used some variation of an automatic watering system while the others manually watered their plants.

Those that did not have plants at home were asked what factors would motivate them to go for home gardening. The following graph shows the trends between the factors with mental wellbeing, hobby and relaxation and improved air quality being the most common motivation factors and creative DIY projects and seasonal décor being the least common motivating factors.

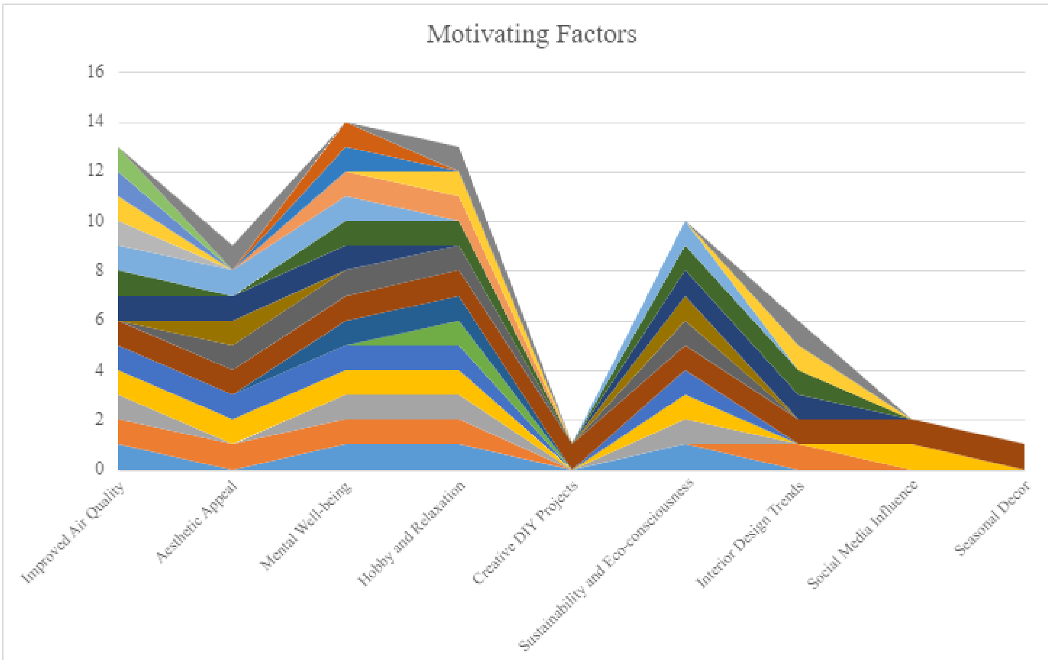


Figure 1.5: Stacked Area Chart Motivating factors
Source: Author

In section two of the survey, we inquired further about the audience's opinions on SPCSs.

We first asked what their views were about the SPCSs that can be currently found in the market and the answers were as follows:

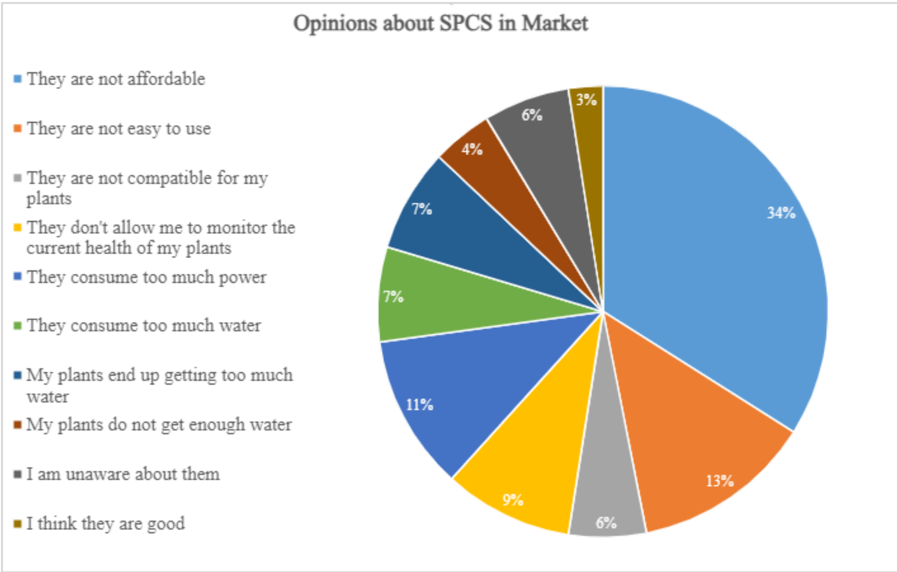


Figure 1.6: Pie Chart about views on SPCS in current market
Source: Author

The main concerns shown are affordability, ease of use, and compatibility with plants.

When asked about what features they would like to see in SPCSs, the responses showed that users highly valued automatic watering, soil moisture sensing and compatibility for indoor and outdoor settings in SPCSs. The following chart reflects the distribution of the preferred features.

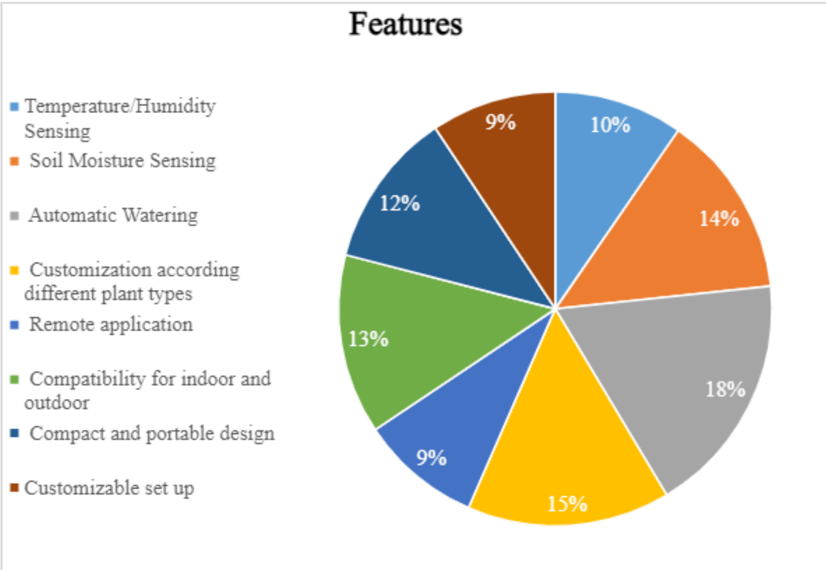


Figure 1.7: Pie Chart on preferred features of SPCS
Source: Author

Considering a solution where all their needs were met, more than half the respondents were willing to buy an SPCS with all their preferred features within 2024.

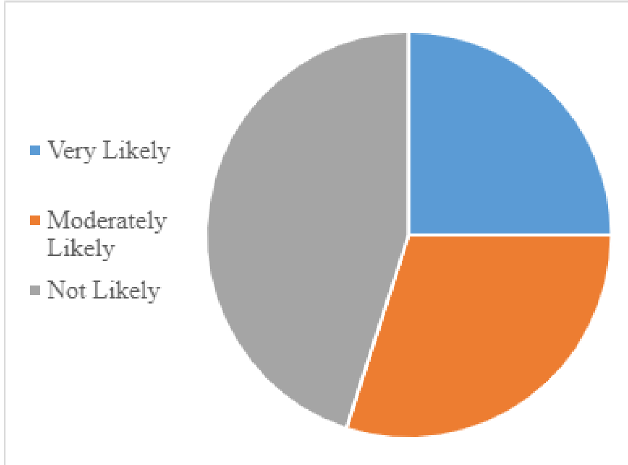


Figure 1.8: Pie Chart on likeliness of purchasing SPCS in 2024
Source: Author

The survey results indicate a strong preference for plant care among respondents, with a significant majority either interested in practicing or currently practicing home gardening. While many value features like air quality and aesthetics, the main motivating factors for adopting SPCSs are ease of use and the system's ability to adapt to the user's lifestyle, highlighting a need for affordable, user-friendly solutions tailored to diverse plant care needs.

1.2 Project Purpose

To optimize plant health and maintenance, our IoT Empowered SPCS focuses on developing and implementing an automated plant care system using IoT technology for enhanced efficiency and sustainability. The project aims to provide a comprehensive analysis of various IoT-based models for plant care and to answer the following questions:

a. Can simple sensor-based models effectively monitor and maintain plant health? If yes, how?
b. Are integrated IoT systems with automated watering and environmental monitoring more efficient in ensuring optimal plant conditions than traditional gardening methods?
c. Can the responsiveness and accuracy of the system be enhanced without increasing complexity or cost?

1.3 Project Objectives

Our project's objectives are as follows:

1. Automate Plant Care:
   Develop an IoT-based system to automate watering and monitor plant health parameters, reducing manual effort and ensuring consistent plant care.

2. Enhance Plant Health Monitoring:
   Integrate soil moisture, temperature, and humidity sensors to continuously monitor and maintain optimal growing conditions for various plant types.

3. Plant Adaptive:
   Implement adaptive algorithms to adjust watering schedules based on plant-specific needs.

4. Cost-Effective Solution:
   Design the system to be affordable and accessible for users of different financial backgrounds, leveraging low-cost components and efficient power management.

5. Scalability and Customization:
   Ensure the system is scalable to accommodate multiple plants and customizable for different plant species and setups.

6. Sustainability:
   Promote sustainable gardening practices by optimizing water usage and reducing resource wastage through intelligent automation and monitoring.

7. Educational Value:
   Offer educational insights to users on plant care, fostering a broader understanding of intelligent gardening technologies.

1.4 Project Scope

This project aims to enhance and revolutionize traditional gardening practices by developing a solution to automate watering and monitor plant health by integrating sensors and actuators controlled via an Arduino-based system. The project aims to provide a scalable, customizable, and cost-effective system that enhances traditional gardening practices through adaptive, data-driven care routines, ensuring optimal plant health and user convenience.

1.5 Project Overview

Throughout the development of the SPCS, the focus will be on implementing IoT technologies to automate plant care and mitigate issues caused by unattended plants during harsh weather conditions. By integrating soil moisture, temperature, and humidity sensors with an Arduino-controlled system, the project aims to ensure consistent plant health. The development will evaluate various environmental monitoring techniques, ultimately integrating these into a comprehensive system for users to manage plant care efficiently. The system will also compare the effectiveness of different automation strategies, enhancing its adaptability and reliability across diverse plant types and growing conditions.

1.6 Document Conventions

PCS	Plant Care System
VNF	Vegetation and Flora
IoT	Internet of Things
UI	User Interface
UX	User Experience

1.7 Brief Literature Review

Plant care systems have evolved significantly, incorporating various technologies to enhance automated plant care. This review explores the development and current state of SPCS, focusing on functionalities like automated watering, environmental monitoring, and user interaction through IoT integration. [4] [5]

1.7.1 IoT-Based Plant Care Techniques

IoT-based plant care techniques are crucial in modern gardening systems. [6] These techniques leverage interconnected sensors and actuators controlled through microcontrollers like Arduino to provide automated plant care. Soil moisture, temperature, and humidity sensors continuously monitor the environment, feeding data into the system to make real-time decisions about watering control. Advanced systems utilize wireless communication modules to allow system connectivity to mobile applications. This integration of sensors and IoT devices facilitates adaptive, data-driven plant care, enhancing efficiency and convenience for users. [7] [8]

Key aspects include:

- ♣ Soil Moisture Sensors: Measure moisture levels, sending data to the controller.
- ♣ Water Pumps and Valves: Regulated by the system to dispense water as needed.
- ♣ Temperature and Humidity Sensors: Helps maintain optimal environmental conditions.

1.7.2 SPCS – Comparative Analysis

In comparing various SPCS, differences in functionalities such as automated watering and sensor integration become apparent. The system excels by combining soil moisture sensing with automated irrigation, addressing limitations found in other products. Systems like Rachio and Rain Bird primarily offer irrigation control without extensive soil moisture sensing. [9] [10] Our system's key differentiator is its holistic approach, offering customization for different plant types, seamless indoor/outdoor compatibility, and affordability, addressing the limitations seen in competing products. [11]

1.8 Project Feasibility Analysis

Project feasibility is an assessment of the practicality and potential success of a proposed project. It involves evaluating various aspects to determine whether the project can be effectively completed and whether it will achieve its intended objectives. Feasibility studies are conducted to analyze the likelihood of the project's success before committing significant resources. [12]

This section describes the ease of developing SPCS in different contexts.

1.8.1 Economic Feasibility

The economic feasibility of a project evaluates the financial aspects, including costs and potential benefits.

The primary costs of this project involve purchasing hardware components (Arduino Uno, sensors, relay modules, water pumps, and Bluetooth Module) and any software tools (e.g., licenses for MySQL Workbench). Additional costs may include maintenance and future upgrades. Overall, the project is economically feasible as the initial investment is quite low, and the potential savings and convenience provided by the system justify the costs.

1.8.2 Technical Feasibility

Technical feasibility assesses whether the project can be successfully completed with the available technology and skills.

This project involves using an Arduino Uno, sensors, and actuators, which are well-documented and supported by a large community which is a major advantage. The necessary skills include basic electronics, programming (C++ for Arduino), and understanding of database management (MySQL). Given the widespread availability of tutorials and resources, the project is technically feasible. Any technical challenges can be addressed through community support such as Arduino’s community on Reddit as well as their available documentation.

1.8.3 Operational Feasibility

Operational feasibility examines the project's ability to function within its intended environment and the support it will receive from users.

This project aims to automate plant care, reducing the need for manual watering and monitoring. Users, such as home gardeners or small-scale farmers, will find the system beneficial due to its ease of use and reliability. The system can be seamlessly integrated into the daily routines of users, making it operationally feasible. The system's usability and minimal maintenance requirements ensure that it will be well-received and effectively utilized by the target audience.

1.8.4 Schedule Feasibility

Schedule feasibility analyzes whether the project can be completed within a reasonable timeframe. The project is divided into two main phases:

First Phase of the Project

In this phase, the Proposal, SRS, SDS and the initial technical documentation are to be prepared and presented. A short demonstration of the incomplete system will also be presented.

Duration	**November 2023-January 2024**
November 4th, 2023	Project Proposal Presentation, Approval, and Submission
December 28th, 2023	Submission of initial Software Requirements Specification and Software Design Specification Document
January 28th, 2024	FYP-1 Final Presentation and the submission of a hard copy of the SRS and SDS documents.

Table 1.1: First Phase of Project

Second Phase of the Project

In the second phase of the Final Year project, the team will work on completing the working model of the SPCS as well as completing the final report, technical document and the user manual.

Duration	**February 2024 – June 2024**
June 6th, 2024	Presentation of the Final Project
June 13th, 2024	Submission of the Final Project

Table 1.2: Second Phase of the Project

1.8.5 Conclusion of the Feasibility Analysis

The feasibility analysis indicates that the project is well grounded from economic, technical, operational, and schedule perspectives. The initial costs are justified because of the long-term benefits and convenience provided by the automated plant care system. The required technology and skills are readily accessible, and the project can be completed within a reasonable timeframe. Additionally, the system is expected to be well-received by users, thus resulting in its operational success.

In conclusion, the project is feasible and worth pursuing, promising significant improvements in plant care efficiency and user satisfaction.

1.9 Deliverables

1.9.1 Hardware

- Sensors: Soil Moisture, temperature, humidity
- Actuators: Water Pump, Relay Module, LCD
- Arduino Setup: Microcontroller with necessary circuits

1.9.2 Software

Arduino IDE code: Software for programming the microcontroller's functions (data acquisitions, data processing, control logic) as well as setting thresholds.

1.10 Report Orientation

This final report for the Final Year Project named IoT empowered Smart Plant Care System is composed of the following three chapters and __ appendices as given below:

- Chapter 1: Context and Preliminary Introduction
  This chapter provides an overview of the project, outlining its objectives, significance, and the motivation behind automating a smart plant care system (SPCS) using Arduino and MySQL. It introduces the problem statement, highlights the benefits of an automated system, and sets the stage for the detailed discussions in subsequent chapters.
- Chapter 2: Research
  This chapter delves into the background research as well as the literature review necessary for the SPCS project. Existing technologies, similar projects, and relevant methodologies are explored in this section.
- Chapter 3: Analysis
  In this chapter, the feasibility of the SPCS is analyzed from various perspectives—economic, technical, operational, and schedule. It includes a detailed feasibility study that evaluates the project's practicality, cost, technical challenges, and potential risks, providing a solid foundation and justification for the design phase.
- Chapter 4: Design
  This chapter presents the system design, detailing both the hardware and software components. It includes circuit diagrams, system architecture, database schema, and UML Diagrams that were made across the various phases as the SPCS was being made.

- Chapter 5: Test Plan
  The test plan chapter outlines the strategies and methodologies for testing the system. It specifies the types of tests to be conducted, the testing environment, and the criteria for success.
- Chapter 6: Critical Evaluation
  This chapter critically evaluates the SPCS, assessing whether the objectives were met and identifying any limitations or challenges encountered throughout the duration of the Final Year Project. It reflects on the successes and areas for improvement as well as laying out future enhancement ideas.
- Appendix A: Glossary
  The glossary appendix provides definitions and explanations of technical terms and abbreviations used throughout the report.
- Appendix B: Code
  This appendix includes the complete source code developed for the project.
- Appendix C: User Manual
  The user manual appendix offers comprehensive instructions for end-users on how to set up, operate, and maintain the automated plant care system.
- Appendix D: Developer's Manual
  The developer's manual appendix is aimed at technical personnel who may need to understand, modify, or extend the system.

Chapter Two: Research

2.1 Academic Research

Research methods refer to the systematic procedures used for collecting and analyzing data. Designing these methods is a crucial aspect of the overall research framework. In developing these methods, there are two critical decisions to be made: selecting the appropriate data collection techniques and determining the best data analysis approaches. These decisions shape the research methodology, ensuring that data is gathered and interpreted in a manner that aligns with the research objectives and design. [13]

2.1.1 Research Methodologies

Qualitative

This type of research focuses on understanding any sort of project, task, activity, process, operation or network through detailed narrative data. It uses methods like interviews, observations, reading, and critical analysis of content. [14]

Quantitative

This research type aims at numeric variables and analyzing statistical relationships. This type utilizes surveys, experiments and statistical models to analyze numerical data. [14]

This project follows a quantitative research method. For more details regarding this part, please refer to chapter one of this document.

2.1.2 Software Development Life Cycle Methodology

The development team must select the system development technique that is most appropriate for the current project to manage it effectively. Every methodology has a unique set of advantages and disadvantages as well as a distinct purpose. [15]

Agile Development Methodology

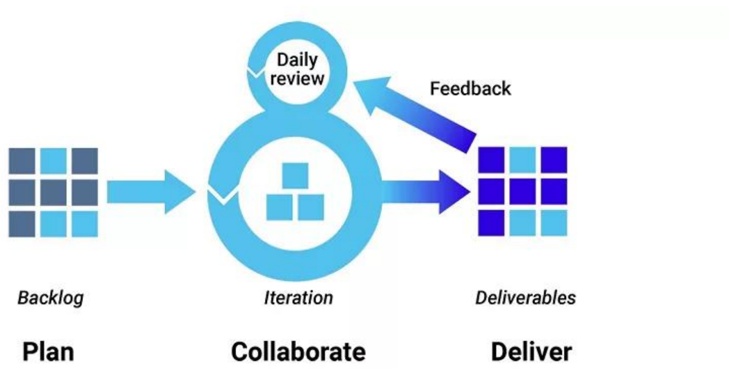


Figure 2.1: Agile Development Methodology
Source: Acquired from [15]

To provide certain features for a release, the tasks in the Agile Model are separated into time boxes, or brief time spans. Every iteration of the technique results in the delivery of a functional software build. Features are added incrementally to each release, with all necessary features included in the final build. [16]

Waterfall Development Methodology

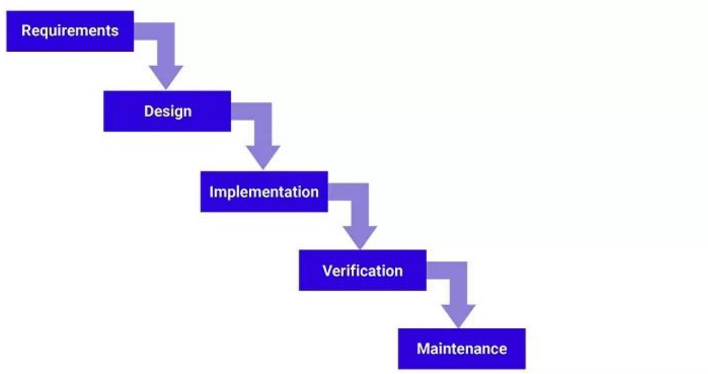


Figure 2.2: Waterfall Development Methodology
Source: Acquired from [15]

The Waterfall methodology is a step-by-step approach to project development that proceeds through all stages of a project, such as analysis, design, development, and testing, like a waterfall. Each stage is finished before the next one starts. The quantity and quality of work completed prior to implementation determines the Waterfall method's success. [17]

Incremental Development Methodology

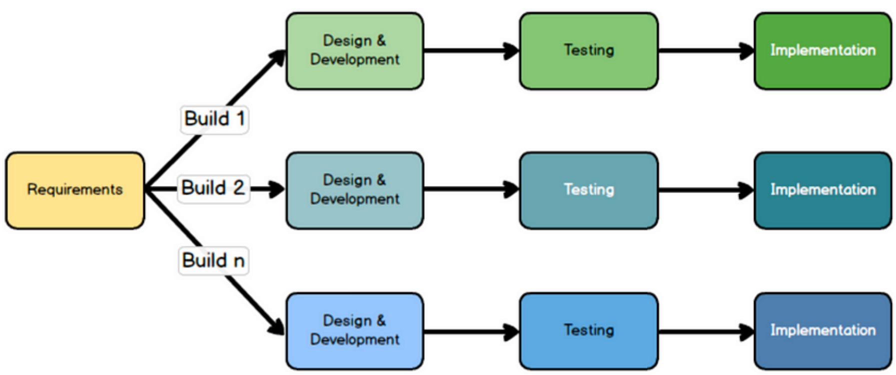


Figure 2.3: Incremental Development Methodology
Source: Acquired from [18]

The incremental model divides the work into smaller sections called builds. Each build goes through its own development cycle, making the process more manageable. Incremental models are similar to Agile models. [18]

Selection of System Development Methodology

After careful consideration and critical analysis of the various development methods, the team chose to implement agile development methodology for the development of this system. This was done for a few reasons:

- Agile Development Methodology was supported by the continuous testing methods
- Agile Development Methodology allowed for work to be done nearly twice as fast as compared to other methods, resulting in the system being fully developed over the span of 5 months.
- Agile Development Methodology supported the simultaneous inclusion and testing of various functions throughout the development process.

2.1.3 Development Tools

Several development tools were required to create the SPCS and make it functional.

Appropriate Tools Available:

Database Management Tools:

- Microsoft SQL Server Management Studio

Figure 2.4: SSMS logo
Source: Acquired from [19]

Any SQL infrastructure can be managed with the integrated environment of SQL Server Management Studio (SSMS). SSMS is a single, all-inclusive solution that combines a large array of graphical tools with numerous sophisticated script editors, giving database administrators and developers of all experience levels access to SQL Server. [20]

- MySQL Workbench CE 8.0

Figure 2.5: MySQL logo
Source: Acquired from [21]

Database architects, developers, and database administrators can collaborate with one another using MySQL Workbench, a unified visual database designing or graphical user interface tool. In addition to offering extensive administrative tools for server configuration, user management, backup, and many other tasks, it also offers SQL creation, data modeling, and data movement. [22]

- Microsoft Excel

Figure 2.6: Excel logo
Source: Acquired from [23]

Microsoft Excel is a popular spreadsheet program for businesses. It allows users to format, organize, visualize, and calculate data. The program is mainly used to enter and organize data, create charts and graphs, and manage projects. [24]

Application Development Tools:

- ♣ IoT Remote App by Arduino Cloud

Figure 2.7: IoT Remote App logo
Source: Acquired from [25]

Arduino IoT Cloud is an extensive platform that simplifies the process of developing Internet of Things (IoT) applications. It provides an integrated solution that enables customers to connect their Arduino-based technology to the internet and manage them online via a simple interface. [26]

- ♣ Flutter

Figure 2.8: IoT Remote App logo
Source: Acquired from [27]

Flutter is Google's portable user interface toolkit that allows you to create natively built applications for desktop, mobile, and the web from just one codebase. Flutter integrates with the current code, can be utilized by developers and organizations worldwide, and is free and open source. [28]

- ♣ MIT App Inventor 2

Figure 2.9: IoT Remote App logo
Source: Acquired from [29]

MIT App Inventor is a simple, visual programming environment that lets anyone create fully working programs for Android phones, iPhones, and Android/iOS tablets. The MIT App Inventor initiative aims to democratize software development by enabling everyone, particularly young people, to transition from technology consumption to technology creation. [29]

Programming Tools:

- ♣ Visual Basic

Figure 2.10: Visual Basic logo
Source: Acquired from [30]

Microsoft created Visual Basic, an object-oriented programming language. Making type-safe.NET applications is quick and simple when you use Visual Basic. [31]

Software developers can use GUI-based tools under the VB programming framework to change code segments by dragging and dropping items. It enables programmers to use graphical elements to design the appearance, behavior, and functionality of many objects. [32]

- ♣ Arduino IDE

Figure 2.11: Arduino IDE logo
Source: Acquired from [33]

The open-source electronics platform, Arduino, is built on user-friendly hardware and software. Arduino boards can take inputs, such as a light from a sensor, a finger pressing a button, or a message from Twitter, and convert them into outputs, such as starting a motor, turning on an LED, or posting content to the internet. By delivering a set of instructions to the microcontroller on the board, you can instruct your board on what to do. You use the Arduino Software (IDE), based on Processing, and the Arduino programming language, which is based on Wiring, to accomplish this. [34]

Selection of appropriate tools

The SPCS has been developed using Arduino IDE to support Arduino hardware. The application has been created through MIT App Inventor for facilitating the connectivity between the Hardware and the smart device.
Additionally, MySQL Workbench was used by developers for the intent of convenience in managing plant information in a database.

Chapter Three: Analysis

3.1 User Requirements/Use Cases

3.1.1 Use Cases

The SPCS provides several functionalities for users:

- Set up Sensors: Users can install and configure sensors to monitor various environmental parameters such as soil moisture levels.
- Monitor Real-Time Data on LCD: The system displays real-time data from the sensors on an LCD, allowing users to keep track of their plant environment.
- Choose What to Plant: Users can select different types of plants and configure the system accordingly.
- Get Information About Plant Categories: The system provides detailed information about plant categories, including their care requirements and optimal growing conditions, aiding users in informed plant selection and care.

These functionalities streamline plant care, making it easier for users to maintain healthy plants by automating monitoring and providing essential plant-related information.

3.1.2 Analyst Requirements

Data Analytics:

- Analysts require tools to evaluate historical plant data, including trends in moisture levels and watering patterns.
- Analysts require tools to evaluate market demographics and changes in consumer needs.

Reporting:

- Analysts must be able to generate reports on system performance and plant health metrics.
- Analysts must be able to generate reports on user satisfaction and feedback.

System Monitoring:

- Analysts must be able to access logs and diagnostic information to monitor system operations and identify potential issues.

3.1.3 Developer Requirements

Modular Codebase:

Developers need a well-documented and modular codebase to facilitate easy updates and maintenance.

Integration Framework:

Developers need support for integrating additional sensors or actuators without major system overhauls.

Testing Tools:

Developers need access to simulation and testing tools to verify sensor integration and system response.

Documentation:

Developers need access to comprehensive technical documentation covering system architecture, APIs, and deployment procedures.

Debugging Support:

Developers need features to assist in debugging, such as serial monitors and logging for troubleshooting sensor data and actuator control.

3.2 Use Case Diagrams

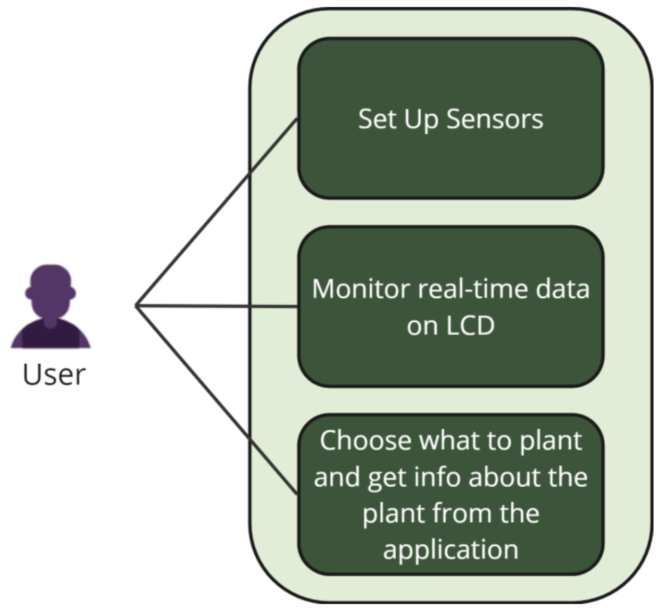


Figure 3.1: Use Case Diagram
Source: Author

This use case diagram illustrates the primary interactions between the user and the SPCS as mentioned before. Users can set up sensors, monitor real-time data on the LCD, and choose what to plant while getting detailed information about the plant from the application, facilitating effective and informed plant care.

3.3 System Specifications

3.3.1 Functional Requirements

Functional requirements define the specific behavior or functions of the system. For the automated plant care system, the functional requirements are as follows:

Data Acquisition:

- ♣ The system must read soil moisture levels from sensors at regular intervals.
- ♣ The system must read environmental data such as temperature and humidity if additional sensors are included. Same can be said for any additional sensors.

Data Processing:

- ♣ The system must process sensor data to determine whether the soil moisture level is below a predefined threshold.

Actuation:

- ♣ Water pumps must be activated by the system (microcontroller) when the soil moisture level falls below the set minimum threshold.
- ♣ Water pumps must be deactivated by the system (microcontroller) when the soil moisture level is above the maximum threshold.

Data Storage:

- ♣ The system must log sensor readings and actions taken after every iteration.

User Interface:

- ♣ The system must provide a user interface for users to see their current plants.

Connectivity:

- ♣ The system must support connectivity via Bluetooth.

3.3.2 Quality Attributes

Quality attributes specify non-functional requirements that the system must meet. For the automated plant care system, the quality attributes are:

Reliability:

- ♣ The system must operate continuously without any setback or failure.
- ♣ The system must have a mechanism to handle sensor or Bluetooth failures gracefully.

Scalability:

- ♣ The system must support adding multiple sensors and water pumps to cover a larger area or more plants.

Usability:

- ♣ The system must have a user-friendly interface that is easy to navigate and configure.

Performance:

- ♣ The system must respond to changes in soil moisture levels within a few seconds.

Maintainability:

- ♣ The system must be designed in a modular way to allow for easy updates and maintenance.

Security:

- The system must ensure that only authorized users can access and configure the system.
- Data transmitted over Bluetooth must be encrypted to prevent unauthorized access.

3.3.3 Domain Requirements

Domain requirements are specific to the environment in which the system will operate and may include:

Environmental Conditions:

- The system must be able to operate in various environmental conditions, including different levels of humidity, temperature, and light.

Power Supply:

- The system must be able to operate on a reliable power source, with backup options in case of power failure.

Integration:

- The system must be compatible with common gardening tools and practices.
- The system must integrate seamlessly with existing irrigation systems if present.

Compliance:

- The system must comply with local regulations regarding electronic devices and water usage.

3.3.4 Interface Requirements

Interface requirements specify how the system will interact with users, other systems, and hardware components:

User Interface:

- The system must provide a graphical user interface (GUI) accessible via a dedicated mobile app.
- It must display real-time sensor data, system status, and historical logs.
- It must also allow users to configure settings such as moisture thresholds and alert preferences.

Hardware Interfaces:

- The system must connect with soil moisture sensors via analog or digital input pins on the Arduino.
- The system must interface with water pumps via relay modules connected to digital output pins on the Arduino.

General Requirements of the SPCS:

Mobile Phone and Bluetooth Connection

Operation System	Space	RAM
Android	25MB	2GB

Table 2.1: Operating System Requirements

Database Interface:

- A MySQL database is used by developers to store and retrieve sensor data and system logs.
- In the future, the system must provide functions to execute SQL queries for data insertion and retrieval.

Chapter Four: Design

4.1 Architectural Strategies

This section focuses on the methodologies and techniques implemented to achieve the previously mentioned target objectives.

4.1.1 Data Preprocessing

The SPCS relies on data from various sensors to monitor the primary data sources like soil moisture, temperature, and humidity. The sensors capture essential raw data of environmental conditions. Once this data is processed, the system defines the appropriate actions for plant care.

Data Collection Process:

- ♣ Soil Moisture Sensors: Measures the water content in the soil. Data is acquired continuously to monitor moisture levels.
- ♣ Temperature and Humidity Sensors: Records ambient conditions to ensure plants are within optimal growing parameters.

Data Preprocessing Steps:

- ♣ Filtering: Raw sensor data is filtered to remove noise and erroneous readings.
- ♣ Normalization: The data is adjusted to a standard range to ensure consistency.
- ♣ Threshold Setting: The system analyzes data for upper and lower bounds and defines thresholds for moisture levels of the sensor corresponding to the soil it is in.

4.1.2 Automated Control System

The system utilizes a microcontroller (Arduino) to process sensor data and manage plant care operations. The microcontroller receives input from sensors and controls actuators such as water pumps and fans.

Control Mechanisms:

Automated Irrigation: The microcontroller compares soil moisture readings against predefined thresholds. Once soil moisture readings are below the minimum threshold, the water pump is activated, and the plant gets watered until the optimal moisture level is reached.

Implementation Techniques:

Relay Modules: Controls high-power devices like water pumps to efficiently manage the watering feature.

4.1.3 System Calibration and Optimization

The system calibrates sensors and tunes control mechanisms to ensure effective data collection. The system must respond appropriately to changes in the environmental conditions.

Calibration Techniques:

- Sensor Calibration: The system calibrates each sensor to account for variability and to ensure accurate data readings.
- Actuator Tuning: The system tunes water pumps to operate efficiently, making the necessary adjustments without overuse.

Optimization Methods:

- Feedback loops are implemented to dynamically adjust thresholds based on historical data and observed plant responses.
- The system is optimized to minimize power consumption by activating devices only when necessary.

4.1.4 Scalability and Modularity

The system is designed for scalability and modularity, allowing for expansion and adaptation to different plant care scenarios:

- Modular Design: Components can be added or removed based on the scale of the application, whether for a single plant or a garden.
- Customizable Settings: Users can set specific parameters for different plant types, making the system adaptable to various environmental needs.

Implementation Strategy:

- Expansion Capability: Supports additional sensors and actuators as needed.
- Customization Options: Sets different thresholds based on various plants.

4.2 Deliverables of Process Modeling

4.2.1 Context Diagram

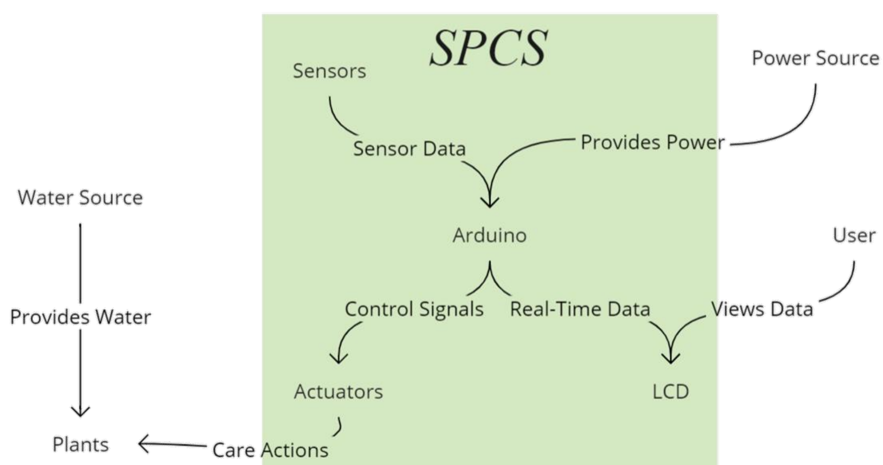


Figure 4.1: Context Diagram for SPCS
Source: Author

The context diagram illustrates the SPCS, which automates plant maintenance by integrating sensors, an Arduino controller, and actuators. Sensors provide soil moisture and temperature data to the Arduino, which processes this information to send control signals to actuators, such as water pumps, for plant care. LCD shows real-time data for monitoring. The system is powered by a central power source and utilizes a water source for irrigation. Users can view data and adjust settings, enhancing plant care efficiency.

4.2.2 System Architecture

The System Architecture details the components and their interactions within the SPCS. The layered architecture is as follows:

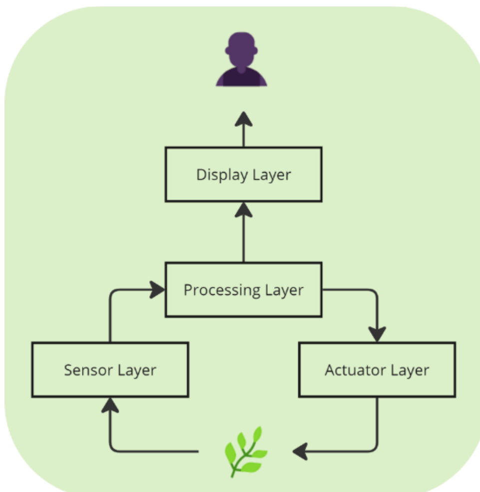


Figure 4.2: Layers in System Architecture
Source: Author

Sensor Layer:

- Components: Soil moisture sensors, temperature sensors, humidity sensors
- Function: Collect data about the plant environment and send it to the Arduino microcontroller.

Actuator Layer:

- Components: Water pumps, relays
- Function: Executes actions like watering plants based on signals from the Arduino.

Processing Layer:

- Component: Arduino microcontroller
- Function: Receives and processes sensor data, applies predefined rules to determine necessary actions, and sends control signals to actuators.

Display Layer:

- Component: LCD Display
- Function: Displays real-time environmental data for the user to view.

4.2.3 UML Diagram

Class Diagram:

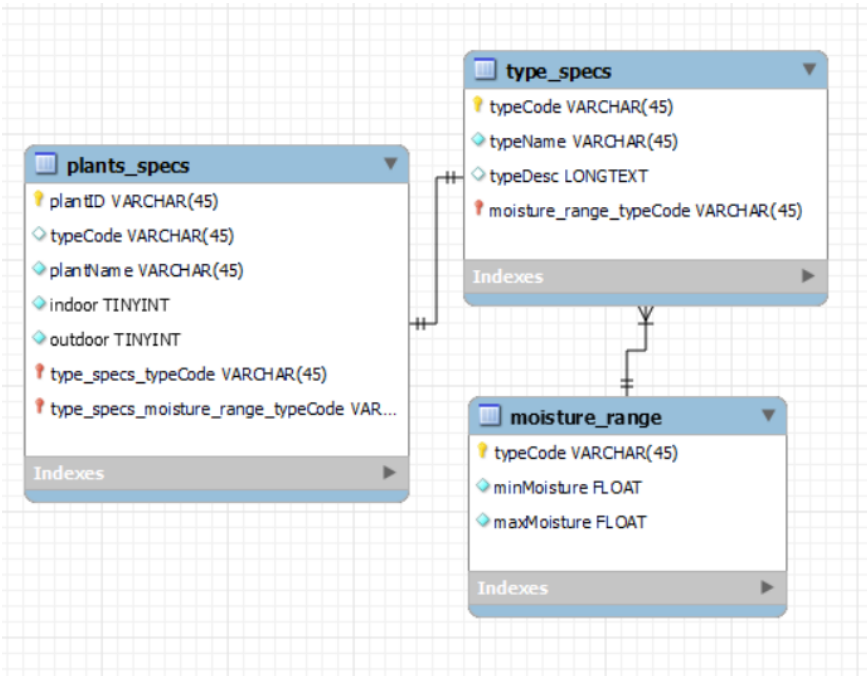


Figure 4.3: Class Diagram
Source: Author

This class diagram depicts the structure of the database for the SPCS. It includes three main entities:

- `plants_specs`: Contains information about specific plants, including their ID, type, name, and whether they are suited for indoor or outdoor environments.
- `type_specs`: Details about plant types, including the code, name, description, and reference to the moisture range.
- `moisture_range`: Defines the moisture requirements for each plant type, including minimum and maximum moisture levels associated with each plant type.

These tables are interconnected to manage plant data, types, and their moisture requirements efficiently.

Sequence Diagram:

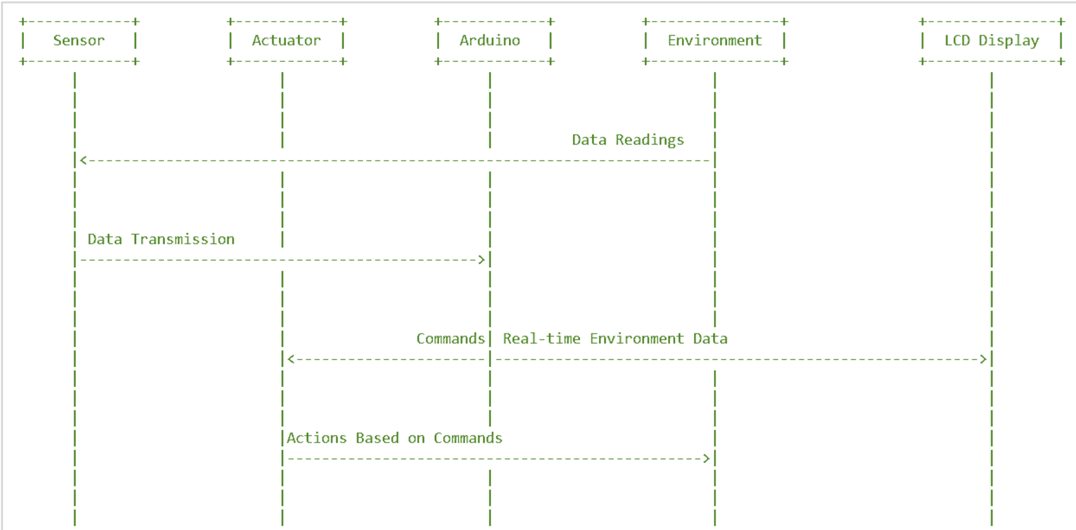


Figure 4.4: Sequence Diagram
Source: Author

This sequence diagram outlines the workflow of the SPCS. Sensors collect environmental data (soil moisture, temperature, humidity) and transmit it to the Arduino. The Arduino processes this data and sends commands to actuators (water pumps) to perform actions based on the current environment. The system also updates real-time data on an LCD for user monitoring, ensuring effective plant care automation by reacting to environmental conditions dynamically.

State Machine Diagram:

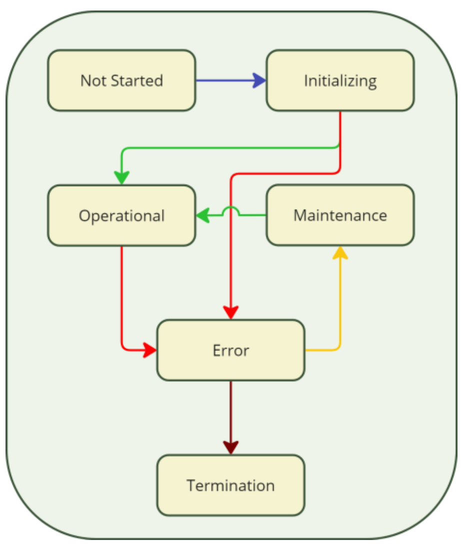


Figure 4.5: State Machine Diagram
Source: Author

Here we can see a simplified version of the different states of the PCS. The process can be in the following states:

- Not Started: the microcontroller is off and has not taken in any data yet
- Initializing: The microcontroller is on and is requesting data readings from sensor
- Operational: The microcontroller is receiving data readings from the sensors and is transmitting commands to the actuators
- Error: The microcontroller has issues during initialization (sensor cannot be found) or there is an issue during operation state (abnormal readings/sensor not found/ actuator not found/ actuator not accepting data)
- Maintenance: The microcontroller analyses the source and nature of error and finds ways to fix it.
- Termination: The microcontroller cannot fix the error, turns system off to avoid catastrophe.

4.3 Policies and Tactics

4.3.1 Application Versions

SPCS will be using the latest version of Arduino IDE and MIT app Inventor. For MySQL, we will be using Workbench 8.0 and MySQL. Data version 6.9.4.0.

4.3.2 Arduino Libraries

For the SPCS, some of the relevant libraries used include:

- DHT.h This library was installed to ensure that correct and accurate readings are received from the DHT11 Temperature and Humidity Sensor.
- LiquidCrystal_I2C.h this library allowed the LCD to communicate with the Arduino microcontroller and display the appropriate data.
- SoftwareSerial.h this library established communication between the Bluetooth module and the Arduino microcontroller.

An additional library which was used was MySQL Connector to attempt integration of the database within the Arduino IDE.

4.3.3 Functional and Object-Oriented Programming

A combination of functional and Object Orientated Programming can improve the overall code modularity, reusability and maintainability.

4.3.4 Software Development Model

Choosing an appropriate software development model ensures the project is completed systematically and efficiently. For this project - the SPCS - the agile methodology proves to be the most suitable due to its iterative and flexible nature. It is a development model that also gives us continuous feedbacks and promotes collaboration between cross-functional teams thus enhancing the overall productivity and innovation.

4.4 GUI's

4.4.1 Sample view of our app

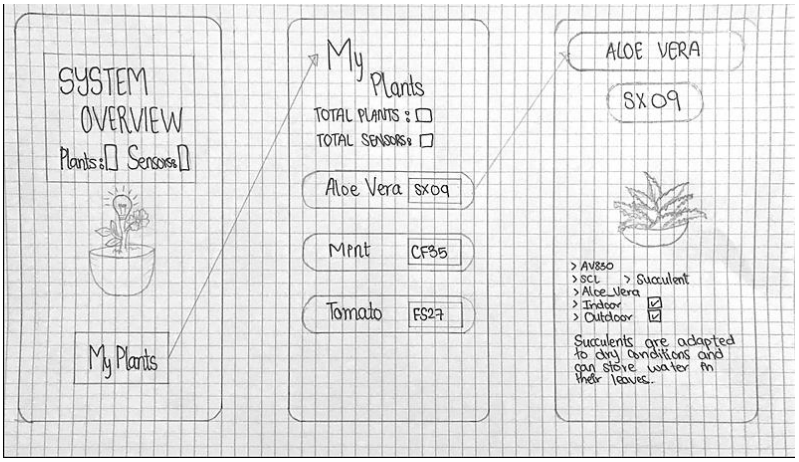


Figure 4.6: Sample view of the app
Source: Author

This was the initial design of the application.

4.4.2 Main Screen

Figure 4.7: Main Screen
Source: Author

This is the screen the user sees when they first open the application. Underneath the greeting message, they can find two options to click on. The bottom one is a button with List Picker functionalities that allows the application to connect to the Arduino Microcontroller via the Bluetooth Module. The button labeled “My plants” leads to the second screen.

4.4.3 My Plants

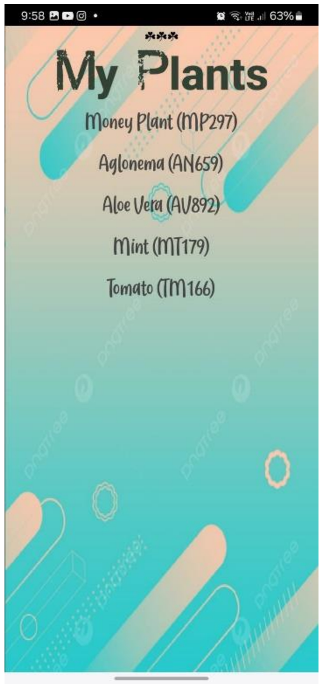


Figure 4.8: Plant choosing screen
Source: Author

The above screen is the second screen the user sees. It shows a list of all their plants as well as the plant codes. The user can click on any of the plant names to view more details about it.

4.4.4 Plant information screens

Figure 4.9: Screens depicting the UI of the plant information
Source: Author

These screens show further details about the category of plant (vegetable, herb, houseplant, etc.) and whether the plant is suited to the indoor environment or the outdoor environment or both.

4.5 Prototype:

4.5.1 Version 1

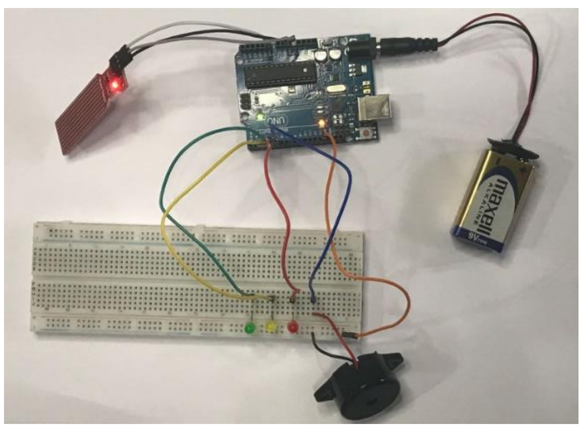


Figure 4.10: Prototype Version 1
Source: Author

The very first prototype consisted of the following main components:

1. A water sensor
2. Buzzer
3. 3 LED lights (Red, Green, Yellow)

The set up was simple with the lights glowing according to the level of the depth of the water sensor. The buzzer would ring if the water level was below a certain threshold with the red LED blinking. The more the water level, the yellow and then the green light would turn on. The yellow LED would indicate that the water level is at optimal level. The green light indicates that there is enough water.

This prototype did not facilitate the use of a soil moisture sensor or connection to the mobile app via Bluetooth.

4.5.2 Version 2

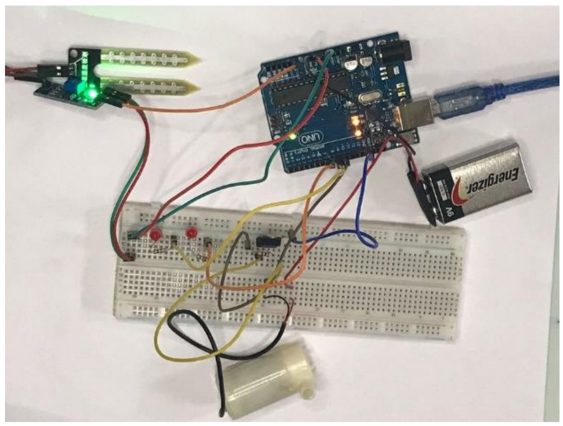


Figure 4.11: Prototype Version 2
Source: Author

In this version, the water sensor is replaced with a soil moisture sensor. A water pump is also added. The working of the system is still nearly the same. With less soil moisture, the pump is supposed to turn on.

The reason why this version was scratched was because of inter-device incompatibility.

4.5.3 Version 3

Figure 4.12: Prototype Version 3
Source: Author

The two LED lights were removed in this version. A relay module was added to control the functioning of the water pump.

4.5.4 Version 4

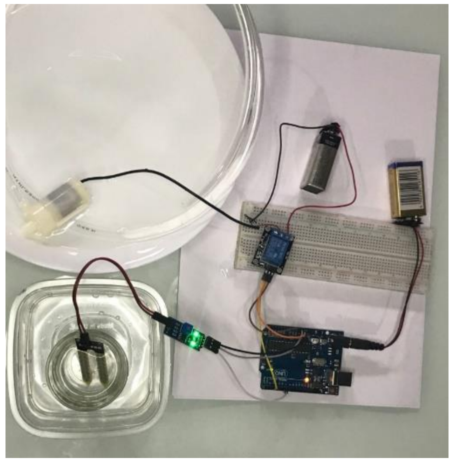


Figure 4.13: Prototype Version 4
Source: Author

In this version, a humidity and temperature sensor were added. Additionally, a small LCD that displayed real-time humidity and temperature was also integrated, further enhancing the efficacy of the SPCS. To get Analogue readings, the MH series moisture sensor was added instead of the digital soil moisture sensor.

4.5.5 Version 5

Figure 4.14: Prototype Version 5
Source: Author

In the final version as of the submission of this report, a Bluetooth module was added to facilitate the connectivity of smart devices with the Arduino microcontroller.

Chapter Five: Test Plan

5.1 Scope of Testing

The scope of testing for the SPCS encompasses verification and validation of all functionalities.

These are:

- sensor data acquisition
- actuator thresholds
- LCD data display
- microcontroller commands

Testing ensures the system accurately monitors soil moisture levels, controls the water pump accordingly, and provides correct information about the plant's environment.

5.2 Test Plan Strategy

5.2.1 Unit Testing

Unit testing is a software development practice where individual components of an application, known as units, are independently examined to ensure they function correctly. This testing is typically performed by software developers and occasionally by QA staff during the development phase. [35]

Unit testing focused on individual components such as the soil moisture sensor, water pump, LCD, and application modules. Each unit was independently tested in isolation to verify its functionality. Tests included

- checking the sensors' ability to detect moisture, temperature, and humidity levels
- the water pump's activation
- LCD data accuracy

5.2.2 Integration Testing

Integration testing, also known as integration and testing (I&T), is a phase in software testing where individual software modules are combined and tested as a group. This testing is performed to assess whether the system or component meets the specified functional requirements. [36]

Integration testing included testing the interactions between different components of the SPC, such as:

- verification of sensor data correctly being processed by the Arduino
- the water pump was activated based on moisture levels
- the LCD showed real-time environment data

5.2.3 System Testing

System testing, also known as system-level testing or system integration testing, involves a quality assurance (QA) team evaluating how the various components of an application work together within the complete, integrated system. [37]

System testing validates the complete and integrated system to ensure it meets the specified requirements. This involves testing the system's response to different moisture levels, overall system performance under various conditions, and the functionality of the entire system for monitoring and managing plant care.

5.3 Test Environment

The test environment for SPCS included:

- an Arduino Uno board
- MH Series moisture sensors
- a 5V relay module
- a water pump
- an LCD

The system was tested in a controlled setting that simulated real-world conditions for indoor and outdoor plants. Various soil types and moisture levels were used to test the sensor's accuracy and the system's response.

5.4 Schedule

5.4.1 Unit Testing

Weeks 1-2: Conducted thorough unit testing for each component. This included verifying the individual functionality of sensors, water pumps, the LCD, and application modules. The aim was to ensure each component worked correctly in isolation before integration.

5.4.2 Integration Testing

Weeks 3-7: Performed comprehensive integration testing. This involved testing the interactions between different components (e.g., sensors with Arduino, Arduino with actuators, Arduino with the LCD). Ensured that integrated components communicate effectively, and that the system performs as expected when components interact.

5.4.3 System Testing

Weeks 8-9: Conducted system testing to validate the overall system functionality, performance, and reliability. This phase helped check if the system would meet the specified requirements and handle real-world scenarios effectively.

5.4.4 Review and Documentation

Week 10-12:

- ♣ Review and documentation of test results
- ♣ Analysis of test results
- ♣ Final adjustments in documentation
- ♣ Deployment preparation (strategizing and planning)

5.5 Control Activities

Throughout the project, the team consistently participated in regular test reviews and progress meetings to effectively oversee and manage testing activities. These meetings served several purposes: reviewing test results, resolving issues identified during different testing phases, and carefully documenting findings. Each meeting enabled the team to comprehensively analyze the test results, ensuring prompt resolution of any problems.

Adjustments to the testing strategy were made based on the feedback received during these reviews. This flexible approach was essential in refining the system and making necessary improvements to meet the desired performance standards. By being responsive to the insights gained from testing, the team aimed to enhance system readiness and ensure a robust deployment process.

This continuous cycle of reviewing, addressing, and documenting played a key role in keeping the project on track with its objectives and maintaining high-quality standards throughout the development lifecycle.

5.6 Resources

5.6.1 Human

The development and implementation of the Smart Plant Care system (SPCS) required the following team members and the supervisor of the project:

- Fatimah Munir
- Abeeha Imran
- Meryum Tahir
- Dr. Rukshanda Kamran (Supervisor)

5.6.2 Hardware

The hardware components of the SPCS included:

- Arduino Uno R3: The central microcontroller for processing the sensor data and sending controls and protocols to the actuators.
- Sensor: MH series moisture sensor, DHT11 (temperature and humidity sensors)
- Relay module: Used to interface the Arduino with devices such as the water pump
- Water pump: Responsible for irrigating the plants
- LCD display: Provided real time data visualization of the humidity and temperature

- Power supply: 9V battery to power the Arduino and sensors.

As for the applications used to make this entire project possible, they mainly involved the use of the computer systems that belonged to the team members. The minimum requirements needed to run the applications are Windows 10 and higher and RAM of 8GB.

5.6.3 Software

The entire project was carried out using the following software resources:

- Arduino IDE: Used for writing and uploading Arduino code to the Arduino microcontroller
- MySQL: Used to make a database that will be used by developers for the integration of the application and database.
- MIT App inventor 2: Used for building the application and establishing connectivity with the Arduino via Bluetooth.

5.7 Test Case Design and Description

Test Case ID	**TC_001**
Test Engineer	Abeeha
Functional Area	Water Sensor (Version 1)
Test Name	Verification of the working of the water sensor
Objective	The purpose of this test is to ensure the working of the water sensor.
Environment	Hardware system of SPCS
Strategy	1. Connect all the modules of the system 2. Place the water sensor in the water
Expected Result	The buzzer will turn on if the water level is below the set threshold set in the Arduino code. If the water level is above the given threshold, the yellow LED will blink. The green LED will turn on if the water level are perfect.
Test Result	Pass

Table 5.1: Test Case 1

Test Case ID	**TC_002**
Test Engineer	Fatimah
Functional Area	Soil moisture sensor and water pump (Version 2)
Test Name	Verification of the working of the moisture sensor and the water pump.
Objective	The purpose of this test is to ensure the working of the moisture sensor and the water pump.
Environment	Hardware system of SPCS
Strategy	1. Connect all the modules of the system 2. Place the moisture sensor in the soil of the plant or water
Expected Result	The water pump should turn on if the soil moisture is less. The red LED should also turn on in this case. In case there is more moisture, the water pump should deactivate.
Test Result	Pass

Table 5.2: Test Case 2

Test Case ID **Test Engineer**	**TC_003** Meryum
Functional Area	Soil moisture sensor, water pump and application. (Version 3)
Test Name	Verification of the working of the moisture sensor and the water pump as well as integration of the SPCS with the IoT remote control Arduino application.
Objective	The purpose of this test is to ensure the working of the moisture sensor and the water pump. The integration of the IoT remote control Arduino application is a crucial factor.
Environment	Hardware system and IoT remote control Arduino application.
Strategy	1. Connect all the modules of the system 2. Place the moisture sensor in the soil of the plant or water 3. Connect the IoT remote control application of Arduino with the SPCS.
Expected Result	The water pump should turn on if the soil moisture is less. The red LED should also turn on in this case. In case there is more moisture, the water pump should deactivate. The IoT remote control application should show real time changes in the Arduino board.
Test Result	Partial failure in terms of the IoT remote control application not being compatible with our Arduino Uno version.

Table 5.3: Test Case 3

Test Case ID	**TC_004**
Test Engineer	Abeeha
Functional Area	Soil moisture sensor, Humidity sensor, temperature sensor, LCD display and water pump. (Version 4)
Test Name	Verification of the working of the moisture sensor, temperature and humidity sensor, LCD display and the water pump.
Objective	The purpose of this test is to ensure the working of the moisture sensor, temperature and humidity sensor, LCD display and the water pump.
Environment	Hardware system
Strategy	1. Connect all the modules of the system 2. Place the moisture sensor in the soil of the plant or water
Expected Result	The water pump should turn on if the soil moisture is less. In case there is more moisture, the water pump should deactivate. The LCD should display real time humidity and temperature.
Test Result	Pass

Table 5.4: Test Case 4

Test Case ID	**TC_005**
Test Engineer	Fatimah
Functional Area	Soil moisture sensor, Humidity sensor, temperature sensor, LCD display, water pump, bluetooth and mobile application. (Version 4)
Test Name	Verification of the working of the moisture sensor, temperature and humidity sensor, LCD display and the water pump. The Bluetooth connectivity with the mobile application was the crucial aspect.
Objective	The purpose of this test is to ensure the working of the moisture sensor, temperature and humidity sensor, LCD display and the water pump, as well as Bluetooth connectivity with mobile application
Environment	Hardware and software system.
Strategy	1. Connect all the modules of the system 2. Place the moisture sensor in the soil of the plant or water 3. Connect Bluetooth with the SPCS.
Expected Result	The water pump should turn on if the soil moisture is less. In case there is more moisture, the water pump should deactivate. The LCD should display real time humidity and temperature. The application should connect with the SPCS.
Test Result	Pass

Table 5.5: Test Case 5

5.8 Status Reporting

A status report is a compilation of data regarding a project's present state. Project team members and other stakeholders are informed about the state of a project through project status reports. [38]

To make sure there were no deviations from the intended plan, the progress was compared to the plan.

5.9 Major Deliverables

Deliverables, to put it simply, are the products of project operations. The size and quantity of project deliverables vary based on the nature of the project. During the project planning stage, they are decided upon by the stakeholders and the project management team. [39]

5.9.1 Test Plan

This specifies the testing strategies that were taken for the testing of the SPCS such as continuous testing to achieve a high-quality end solution.

5.9.2 Test Cases

This focuses on the expected output and the actual output that the developers received in the end when various aspects i.e. the hardware modules and the software application were tested.

5.10 Risks and Assumptions

An assumption is a belief that is taken for granted without any degree of proof. Furthermore, assumptions carry a risk until they've been verified. Future events with a chance of happening and an expected consequence constitute risks. Risks can be threats (negative) or opportunities (positive). [40]

The following are the risks of the SPCS:

- Hardware Failure: There can be a risk of the sensors or actuators malfunctioning leading to incorrect readings or failure in the watering of the plants
- Data inaccuracy: If the sensors are not calibrated properly or there are some issues in their calibration, here is a potential for incorrect sensor data to be sent to the Arduino leading to further faulty actions.
- Integration Challenges: Difficulties in ensuring the smooth communication between the sensors, Arduino and the rest of the actuators.
- Power Supply issues: Without a stable power supply the system operations can be disrupted, negatively affecting the performance and reliability of the system.

The following are the assumptions of the SPCS

- Consistent power supply is supplied across all the system components
- The sensors provide accurate readings
- All hardware components are assumed to be compatible

5.11 Exit Criteria

Conditions known as exit criteria must be accomplished in order to end a specific stage and go on to the next. Team members divide a project into several phases when they apply exit criteria. [41]

The testing of the system will end when:

- All system functionalities have been successfully validated through system integration and testing.
- All critical and major issues identified during testing have been resolved and no new critical issues have emerged during the final testing phase.

Chapter Six: Critical Evaluation

6.1 Success Criteria

In order for the SPCS to be successful, the accuracy of the system must be perfect at all times. The system must monitor and report the soil moisture levels with great accuracy. The temperature and humidity sensor should also be accurate.

Another factor to be noted is that the automation of the system must be working. The system should automatically activate and deactivate the water pump.

Furthermore, the usability of the system should satisfy the customer. Users should be able to set up sensors and access plant information easily.

6.2 Degree of Success

For the final year project, the initial plan was compared with the final project to determine whether it was successful or not. Critical analysis allowed the team to establish the degree of success at 85.56%. The remaining 14.44% could not be achieved due to the following limitations:

- Time constraints
- Incompatibility of devices
- Budget constraints
- Application malfunction

6.3 Learning Experience

Throughout the development of the SPCS, the team gained valuable insights:

- Hardware integration: The team members learned effective techniques for integrating various modules including sensors and actuators with the Arduino microcontrollers.
- System Calibration: The developers gained experience in calibrating sensors by testing out different models of the sensors for accurate data collection.
- Testing Methodologies: The team members developed systematic approach to testing, which improved the reliability and working of the system.
- Documentation: The team members improved their skills in creating comprehensive documentation.

The tools that were used for the development of the SPCS included:

- Arduino IDE
- MIT App inventor 2 for coding
- Microsoft Word and Google Docs for documentation
- PowerPoint, Slidesgo and Canva for presentations
- Zoom and Google Meets for collaboration

6.4 Assumptions and Limitations

The Smart Plant Care System (SPCS) operates under several key assumptions to ensure its functionality and reliability. Firstly, it is assumed that a consistent power supply is provided across all system components, including the Arduino, sensors, and actuators, to maintain uninterrupted operation. Secondly, the system relies on the accuracy of the sensors, assuming they deliver precise readings within their specified ranges to effectively monitor soil moisture levels. Lastly, it is presumed that all hardware components are compatible and work seamlessly together, facilitating smooth integration and reliable performance of the system. These assumptions are critical for the system's successful deployment and operation. [40]

6.5 Future Enhancements

In the future, the team plans on making a few enhancements such as:

- ♣ Further enhancing the scope of the system by commercializing it in different industries such as the agriculture sector.
- ♣ Expansion along new regions of the world where the climate is different from the Middle East and South Asia.
- ♣ Integration of renewable energy sources such as solar power
- ♣ Redesigning data collection methods by integrating a database into the hardware to get more accurate insights on the health of the plant.

References

[1] Z. Yan, L. Jiang, X. Huang, L. Zhang and X. Zhou, "Intelligent urbanism with artificial intelligence in shaping tomorrow’s smart cities: current developments, trends, and future directions," *Journal of Cloud Computing,* vol. 12, no. 1, p. 179, 2023.

[2] E. M. B. M. Karunathilake, T. L. Anh, S. Heo, S. C. Yong and M. Sheikh, "The Path to Smart Farming: Innovations and Opportunities in Precision Agriculture," *Agriculture,* vol. 13, no. 8, p. 1593, 2023.

[3] B. Singh, "UAE & KSA residents are likely to spend more time in their own country this summer season," YouGov, 30 May 2023. [Online]. Available: https://business.yougov.com/content/46707-uae-ksa-residents-are-likely-to-spend-more-time-in-their-own-country-this-summer-season.

[4] G. LiVorio, "Smart Gardening: How Technology Is Revolutionizing Horticulture," 27 Mar 2024. [Online]. Available: https://gardensavvy.com/gardening-blog/gardening-tips/smart-gardening-how-technology-is-revolutionizing-horticulture/.

[5] S. Chavez, "EFFORTLESS GARDENING: MASTERING SMART WATERING FOR YOUR POTTED PLANTS," The Connected Shop, 15 Feb 2024. [Online]. Available: https://theconnectedshop.com/blogs/tech-talk/effortless-gardening-mastering-smart-watering-for-your-potted-plants.

[6] The Bloom Up, "How Smart Planters Work: A Comprehensive Guide to Simplifying Indoor Gardening," 29 Sept 2023. [Online]. Available: https://www.thebloomup.com/how-do-smart-planters-work/.

[7] D. Mowitz, "9-point checklist for impact sprinklers," Successful Farming, 28 Feb 2011. [Online]. Available: https://www.agriculture.com/machinery/irrigation-equipment/9point-checklist-f-impact-sprinklers_269-ar15019.

[8] B. Peck, "Hardware 101: Smart Irrigation Controllers," Gardenista, 23 Feb 2017. [Online]. Available: https://www.gardenista.com/posts/hardscaping-101-smart-irrigation-systems-controllers/.

[9] Rain Bird, "Product Comparison Center," Rain Bird Services and Training, [Online]. Available: https://www.rainbird.com/product-comparison-center?term_node_tid_depth=4448&nids=13198+10380+2892+5544+2852+5462+3619+2935.

[10] Rachio, "Watering Products | Rachio Market | Rachio," Rachio, [Online]. Available: Watering Products | Rachio Market | Rachio.

[11] Galcon, "Residential Gardening," Galcon, [Online]. Available: https://www.galconc.com/field/residential-gardening/.

[12] N. Pearson, P.-J. Naylor, M. C. Ashe, M. Fernandez, S. L. Yoong and L. Wolfenden, "Guidance for conducting feasibility and pilot studies for implementation trials," *Pilot and Feasibility Studies,* vol. 6, no. 1, p. 167, 2020.

[13] Scribbr, "Research Methods | Definitions, Types, Examples," Sribbr, [Online]. Available: https://www.scribbr.com/category/methodology/. [Accessed June 2024].

[14] National University, "What Is Qualitative vs. Quantitative Study?," National University, [Online]. Available: https://www.nu.edu/blog/qualitative-vs-quantitative-study/.

[15] M. McGuire, "Top 4 software development methodologies," Synopsys, 24 March 2024. [Online]. Available: https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies.html. [Accessed June 2024`].

[16] JavaTpoint, "Agile Model - javatpoint," JavaTpoint, 2011. [Online]. Available: https://www.javatpoint.com/software-engineering-agile-model. [Accessed June 2024].

[17] Adobe Communications Team, "Waterfall Methodology: A Complete Guide," Adobe, 18 March 2022. [Online]. Available: https://business.adobe.com/blog/basics/waterfall. [Accessed June 2024].

[18] J. More, "Incremental Development," Bootcamp, 30 Aug 2023. [Online]. Available: https://bootcamp.uxdesign.cc/dlc-software-development-model-incremental-development-7598ae6500e6. [Accessed 2024].

[19] 5. cm, "How to Install Microsoft SQL Server 2019 & SSMS on Windows (On-Prem)," Medium, 20 Nov 2023. [Online]. Available: https://medium.com/@sounder.rahul/how-to-install-microsoft-sql-server-2019-ssms-on-windows-on-prem-3f688d0e98cc. [Accessed June 2024].

[20] CarlRabeler, Matteo-T, craigg-msft, nschonni, MashaMSFT, MikeRayMSFT, damabe, v-surgos, julieMSFT, v-shmck, john-par, DCtheGeek, alexbuckgit, LitKnd, rothja, rwestMSFT and markingmyname, "What is SQL Server Management Studio (SSMS)?," Microsoft, 31 March 2023. [Online]. Available: https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16. [Accessed June 2024].

[21] R. A. S, "MySQL Workbench Installation: A Step-by-Step Guide," Simplilearn, 15 Feb 2024. [Online]. Available: https://www.simplilearn.com/tutorials/mysql-tutorial/mysql-workbench-installation. [Accessed June 2024].

[22] JavaTPoint, "MySQL Workbench (Download and Installation)," JavaTpoint, [Online]. Available: https://www.javatpoint.com/mysql-workbench#:~:text=MySQL%20Workbench%20is%20a%20unified,developed%20and%20maintained%20by%20Oracle.. [Accessed June 2024].

[23] yannzys, "Download Excel Logo, Excel, Logo. Royalty-Free Stock Illustration Image," Pixabay, [Online]. Available: https://pixabay.com/illustrations/excel-logo-excel-logo-generated-8033473/. [Accessed June 2024].

[24] "Microsoft Excel: Easy Guide for Beginners With Formulas and More," HubSpot, [Online]. Available: https://blog.hubspot.com/marketing/microsoft-excel. [Accessed June 2024].

[25] Aptoide, "Arduino IoT Cloud Remote - APK Download for Android," Aptoide, [Online]. Available: https://iot-remote-arduino.en.aptoide.com/app. [Accessed June 2024].

[26] S. Anand, "How does Arduino IoT Cloud work?," 11 Feb 2024. [Online]. Available: https://www.linkedin.com/pulse/how-does-arduino-iot-cloud-work-saurabh-anand-k7goc/. [Accessed June 2024].

[27] Milú, "Flutter for Cross-Platform Applications," Milú, [Online]. Available: https://mrmilu.com/en/flutter. [Accessed June 2024].

[28] P. Lougheed, "FAQ," Flutter, May 2024. [Online]. Available: https://docs.flutter.dev/resources/faq#what-is-flutter. [Accessed June 2024].

[29] Massachusetts Institute of Technology, "About Us," Massachusetts Institute of Technology, [Online]. Available: https://appinventor.mit.edu/about-us. [Accessed June 2024].

[30] A. Kumari, "Visual Basic," 21 Oct 2022. [Online]. Available: https://www.linkedin.com/pulse/visual-basi-anjali-kumari/. [Accessed June 2024].

[31] Microsoft Learn, "Visual Basic docs - get started, tutorials, reference," Microsoft , [Online]. Available: https://learn.microsoft.com/en-us/dotnet/visual-basic/. [Accessed June 2024].

[32] V. Kanade, "What Is Visual Basic? Working, Features, and Applications," Spice Works, 28 April 2023. [Online]. Available: https://www.spiceworks.com/tech/devops/articles/what-is-visual-basic/. [Accessed June 2024].

[33] Arduino Staff, "Software | Arduino," Arduino, [Online]. Available: https://www.arduino.cc/en/software. [Accessed June 2024].

[34] Arduino Staff, "What is Arduino?," Arduino, [Online]. Available: https://www.arduino.cc/en/Guide/Introduction. [Accessed June 2024].

[35] TechTarget Contributor, "Unit Testing," TechTarget, [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/unit-testing. [Accessed June 2024].

[36] Wikipedia Contributors, "Integration testing," Wikipedia, 7 Aug 2019. [Online]. Available: https://en.wikipedia.org/wiki/Integration_testing. [Accessed June 2024].

[37] K. Yasar and R. Black, "System Testing," TechTarget, March 2023. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/system-testing. [Accessed June 2024].

[38] A. C. Team, "Status Reports," Adobe, 18 March 2022. [Online]. Available: https://business.adobe.com/blog/basics/status-reports. [Accessed 11 June 2024].

[39] W. Malsam, "What Is a Project Deliverable? Definition, Examples & More," Project Manager, 2 May 2023. [Online]. Available: https://www.projectmanager.com/blog/project-deliverable. [Accessed June 2024].

[40] A. Zucker, "Project Assumptions & Risks," Project managment essentials, 1 December 2021. [Online]. Available: https://pmessentials.us/assumptions-risks/. [Accessed 12 June 2024].

[41] IsixSigma Staff, "Exit Criteria vs. Acceptance Criteria," ISIXSIGMA, 25 February 2024. [Online]. Available: https://www.isixsigma.com/methodology/exit-criteria-vs-acceptance-criteria/. [Accessed 12 June 2024].

Appendix A: Glossary

Arduino: A microcontroller platform used for building electronic projects.
Automated Watering System: A mechanism that automatically waters plants based on predefined criteria, such as soil moisture levels.
Data Access Controls: Mechanisms that restrict access to sensitive functionalities and user data based on user roles.
Data Logging: The process of capturing, storing, and displaying one or more datasets to analyze activity and trends, and predict future events.
Energy Efficiency: The design consideration focused on minimizing power consumption in both the mobile application and hardware components.
Hardware Interfaces: Connections between physical components
IoT (Internet of Things): The network of interconnected devices that communicate and exchange data to perform smart and automated tasks.
List Picker: An option present in the MIT app inventor. It is a button that when clicked or tapped on, displays a list of options that the user can choose.
Moisture Sensor: A device that measures the moisture content in soil to assess the watering needs of plants.
Native Plants: Plants that naturally occur and thrive in a specific geographic region without human intervention.
Neophytes: Individuals who are new to a subject or activity
Notification Preferences[1]: Customizable settings that allow users to define the frequency and type of alerts received from the system.
Plant Database: A digital repository containing information on various plant species, including care requirements and compatibility.
Real-time Alerts: Immediate notifications based on system data.
Remote Control: Ability to control the system from a distance.
Software Interfaces: Connections between software components
Soil Moisture Level: The amount of water present in the soil, measured as a percentage, indicates whether the soil is dry, moist, or saturated.
Temperature Range: The optimal temperature conditions required for different plant types to thrive.
User Authentication: The process of verifying the identity of a user before granting access to the system.
User Classes: Distinct categories of system users
User Roles: Various levels of access and permissions assigned to users, such as standard users and administrators.

[1] This is specific to the OS and settings of the mobile device.

Appendix B: Code

Version 1

```
const int analogInPin = A0;
int sensorValue = 0;

void setup() {
  // declare pin  to be an output:
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  sensorValue = analogRead(analogInPin);
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\n");
  delay(2);
  if((sensorValue>=50)&&(sensorValue<=600)){
    digitalWrite(2,HIGH);
    delay(100);
    }
  else if((sensorValue>=601)&&(sensorValue<=649)){
   digitalWrite(3,HIGH);
   delay(100);
    }
  else if((sensorValue>=650)&&(sensorValue<=700)){
    digitalWrite(4,HIGH);
    digitalWrite(5,HIGH);
    }
  else{
    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);
    delay(100);
    }
}

```

Version 2

```
#include "thingProperties.h"
#define PumpRunner 0 // set the pin for the pump
int val = 0; // returned value from soil moisture sensor
int soilPin = A0; // pin for reading from the soil moisture sensor
int soilPower = 7; // pin for powering the the soil moisture sensor.
int setMLevel;
void setup() {  // Initialize serial and wait for port to open:
  Serial.begin(9600); // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);  // Defined in thingProperties.h
  initProperties(); // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
  pinMode(PumpRunner, OUTPUT);
  pinMode(soilPower, OUTPUT);
  digitalWrite(soilPower, LOW);
}
void loop() {
  ArduinoCloud.update();
  // Your code here
  moisture = readSoil();
  moisture = map(moisture, 0, 700, 0, 100);
  Serial.println(moisture);
  digitalWrite(PumpRunner, LOW);
  delay(1000);
  if(moisture <= setMLevel && moisture >= 10){
    RunPump = true;
  }
  if(RunPump){
    digitalWrite(PumpRunner, HIGH);
    delay(1000); // pump takes a second or two to start up
    digitalWrite(PumpRunner, LOW);
    delay(1000); // one second delay so water can settle
    RunPump = false;
  }
  delay(1000);// remaining delay, go to value minus 3010
}
int readSoil(){
  digitalWrite(soilPower, HIGH);
  delay(10);
  val = analogRead(soilPin);
  digitalWrite(soilPower, LOW);
  return val;
}
```

Version 3

```
int water; // random variable
const int digitalPin = 6; // digital input pin for soil moisture sensor
const int relayPin = 3;   // output pin for relay module

void setup() {
  pinMode(relayPin, OUTPUT); // output pin for relay board
  pinMode(digitalPin, INPUT); // input pin coming from soil sensor
  Serial.begin(9600); // initialize serial communication
}

void loop() {
  water = digitalRead(digitalPin);  // read the digital signal coming from the soil sensor

  // Print digital readings to the serial monitor
  Serial.print("  |  Digital Sensor Reading: ");
  Serial.println(water);

  // Control the relay based on the digital sensor reading
  if (water == HIGH) { // if water level is low (sensor is dry), then activate the relay and pump
    Serial.println("PUMP ACTIVE");
    digitalWrite(relayPin, HIGH); // HIGH to continue providing signal and water supply
  } else {
    Serial.println("PUMP INACTIVE");
    digitalWrite(relayPin, LOW);  // LOW is to cut the relay
  }

  delay(400);
}
```

Version 4

```
#include "DHT.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Constants
#define DHTPIN 2      // Pin connected to the DHT sensor
#define DHTTYPE DHT11 // DHT 11

const int analogPin = A0; // analog input pin for soil moisture sensor
const int relayPin = 3;   // output pin for relay module

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2); // Set the LCD address to 0x27 for a 16 chars and 2 line display

void setup() {
  pinMode(relayPin, OUTPUT); // output pin for relay board
  Serial.begin(9600); // initialize serial communication
  dht.begin(); // initialize the DHT sensor
  lcd.init(); // initialize the LCD with 16 columns and 2 rows
  lcd.backlight(); // turn on the LCD backlight
}

void loop() {
  // Read the analog value from the soil moisture sensor
  int analogValue = analogRead(analogPin);

  // Read temperature and humidity values from the DHT11 sensor
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  // Print sensor readings to the serial monitor
  Serial.print("Analog Sensor Reading: ");
  Serial.print(analogValue);
  Serial.print("  |  Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C  |  Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  // Display sensor readings on the LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print(" %");

  // Control the relay based on the analog sensor reading
  if (analogValue < 300) { // if analog value is less than 300 (soil is dry), then activate the relay and pump
    Serial.println("PUMP ACTIVE");
    digitalWrite(relayPin, HIGH); // HIGH to continue providing signal and water supply
  } else {
    Serial.println("PUMP INACTIVE");
    digitalWrite(relayPin, LOW);  // LOW is to cut the relay
  }

  delay(3000); // update every 2 seconds
}
```

Version 5

```
#include "DHT.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Constants
#define DHTPIN 2      // Pin connected to the DHT sensor
#define DHTTYPE DHT11 // DHT 11

const int analogPin = A0; // analog input pin for soil moisture sensor
const int relayPin = 3;   // output pin for relay module

char caracter;
int temperature = 0;
int humidity = 0;
int hplmin = 350;
int hplmax = 716;
String tempera_humidity;

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2); // Set the LCD address to 0x27 for a 16 chars and 2 line display

void setup() {
  pinMode(relayPin, OUTPUT); // output pin for relay board
  Serial.begin(9600); // initialize serial communication
  dht.begin(); // initialize the DHT sensor
  lcd.init(); // initialize the LCD with 16 columns and 2 rows
  lcd.backlight(); // turn on the LCD backlight
}

void loop() {
  // Read the analog value from the soil moisture sensor
  int analogValue = analogRead(analogPin);

  // Read temperature and humidity values from the DHT11 sensor
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  //Print sensor readings to the serial monitor
  Serial.print("Soil Moisture: ");
  Serial.print(analogValue);
  Serial.print("\nTemperature: ");
  Serial.print(temperature);
  Serial.print(" °C\nHumidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  // Display sensor readings on the LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print(" %");

  // Control the relay based on the analog sensor reading
  if (analogValue < 350) { // if analog value is less than 300 (soil is dry), then activate the relay and pump
    Serial.println("PUMP ACTIVE");
    digitalWrite(relayPin, HIGH); // HIGH to continue providing signal and water supply
  } else {
   digitalWrite(relayPin, LOW);
  }

  // Check if there's any data sent from the HC-05 Bluetooth module
 if (Serial.available()) {
   char c = Serial.read();
   Serial.write(c);
  }

    Serial.println(tempera_humidity);

  delay(1000); // update every second
}
```

Appendix C: User Manual

-

IoT Empowered Smart Plant Care System

Version 1.0

3/6/2024

Table of Contents

List of Figures

List of Tables

1.0 Introduction

The purpose of the user manual is to provide the intended users of SPCS with information needed to use the SPCS effectively and properly.

2.0 Overview

Technology is advancing at alarming rates but so is climate change. In a world where this technology caters to the changing needs of society in various aspects, we find that the part of society most affected by climate- plant life- is not at the focus point of any major development. There is a dire need for a system to care for the plants while keeping in mind the fluctuation of the weather that we are coming to face more frequently these days. This is where our IoT Empowered Plant Care system comes into play.

The SPCS is a sophisticated integrated solution designed to automate and enhance the care of plants. At the heart of the system is the Arduino microcontroller that coordinates the operations of various sensors and components and is considered the "main brain" of the entire system.

This system continuously monitors vital parameters such as the soil moisture, humidity, and temperature using specialized sensors. The real-time data collected is displayed on a small easy-to-read LCD screen, providing instant feedback on the health of your plants. Additionally, the system includes a Bluetooth module that at the moment, can connect to a mobile application. The mobile application, as in the release of the current version of this report, can only view different plants that the developers have added.

In the future, users can view real time data of their soil moisture, humidity and temperature on the mobile app as well as manually control the watering of the plants via the app.

2.1 Conventions

PCS	Plant Care System
VNF	Vegetation and Flora
IoT	Internet of Things
SRS	Software Requirements Specification
API	Application Programming Interface
GUI	Graphical User Interface
UX	User Experience
DBMS	Database Management System
SDK	Software Development Kit
FAQs	Frequently Asked Questions

Table 2.1: Document Conventions

3.0 Getting Started

3.1 Set-up Considerations

Unbox the Smart Plant Care System and ensure that all the components are included. The device needs to be placed near the plants with an active mobile phone for Bluetooth connection.

Download and install the SPCS application on the mobile phone and connect it to the SPCS system using Bluetooth. The application has been designed to work on Android only currently. Please ensure that the mobile phone fulfills the following:

- Android 10 or above
- A minimum of 2GB RAM

3.2 User Access Considerations

The system is mainly developed for those people who want to do home gardening in a more effective, modern and safer way.

3.3 Accessing the Application

The application can be accessed by installing it in the mobile through a QR code that is given when the user obtains the SPCS. Once the QR code is scanned, the application will be installed after which they will have to give the application permission to access Bluetooth.

3.4 System Organization

The project has been developed using Arduino C++ and its libraries.

Developing and maintaining it requires the use of technical skills such as knowledge related to Arduino modules and their language as well as an understanding of the agile method.

MySQL is also included but has very little use in the system at the time of this report submission.

3.4.1 Main Menu

The main menu is the screen where the user can connect their application to the SPCS. By tapping on "connect your device", the application will detect any nearby Bluetooth devices.

Tap on "HC-05" and put in the password "1234". The application is now connected to the SPCS.

3.4.2 My Plants

In the "My Plants" section, several plants have been added by the developers and at the moment, only their information such as what category they belong to or are they indoor and outdoor, have been added.

3.4.3 Exiting the application

To exit the application, simply swipe out of the application or tap the “back” button.

4.0 Using the system

Place the SPCS near the plant. Push in the pipe and the soil moisture sensor in the soil of the plant.

Open the mobile app and connect to the “HC-05”. The password is “1234”. The mobile app is now connected to the SPCS. In future updates, users can view real time humidity, temperature, pump status and soil moisture level.

Figure 4.1: Main screen of the mobile app

Source: Author

After certain iterations, the Arduino will send commands to the pump to water the plant. If the soil moisture is less than the threshold, the pump will water the plants. If it is more than the threshold, the pump will remain inactive and not water the plant.

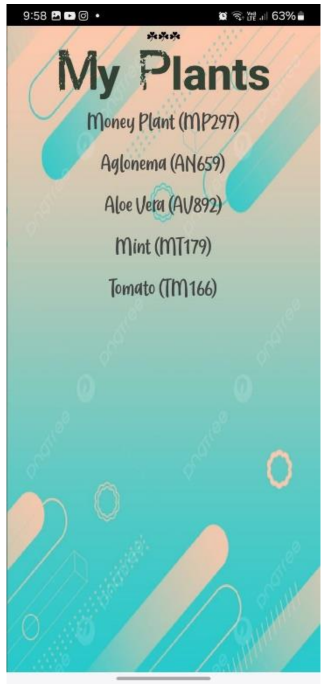


Figure 4.2: Plant choosing screen

Source: Author

The above screen is the second screen the user sees. It shows a list of all their plants as well as the plant codes. The user can click on any of the plant names to view more details about it.

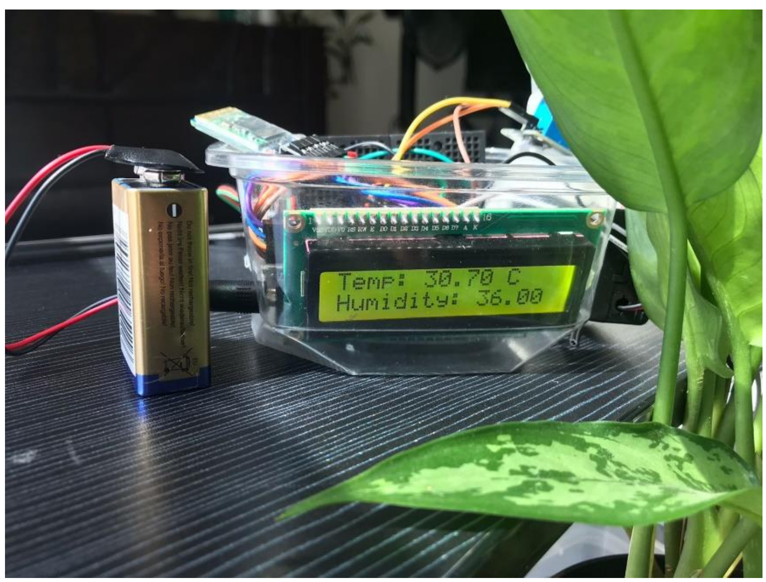


Figure 4.3: Screens depicting the UI of the plant information

Source: Author

These screens show further details about the category of plant (vegetable, herb, houseplant, etc.) and whether the plant is suited to the indoor environment or the outdoor environment or both.

5.0 Troubleshooting:

5.1 Low battery

Right now, 9V batteries are being used to power the system. If the system does not work, check the battery and replace it as they might be low or depleted.

5.2 No Connection

If the mobile app is not connecting to Bluetooth, the issue may be due to the visibility of the Bluetooth module. Make sure you are in range of the Bluetooth module and that the Bluetooth is blinking red on the SPCS.

If the issue persists, go to the settings of the mobile app and under the permissions tab, allow it to connect to other devices.

5.3 Support

In case of any other issue, please contact the following students:

Contact	Organization	Email
Abeeha Imran	SZABIST, Dubai	abeehai61@gmail.com
Fatimah Munir	SZABIST, Dubai	fatimahmunir9876@gmail.com
Meryum Tahir	SZABIST, Dubai	meryumtahir786@gmail.com

Table 5.1: Support Point of Contact

Appendix D: Developer Manual
-
IoT Empowered Smart Plant Care System (SPCS)

Version 1.0

13/6/24

Table of Contents

List of Figures

List of Tables

1.0 Introduction

Welcome to the IoT Empowered Smart Plant Care System Developer Manual. This Manual will guide a developer though the process of building the Smart Plant Care system with the help of Arduino IDE and MIT app inventor 2.

The Smart Plant Care System (SPCS) is designed to automate the care of plants by integrating IoT technology with environmental sensors. This system monitors soil moisture, temperature, and humidity, and automatically activates water pumps to maintain optimal conditions for plant health. The core of the system is built around the Arduino microcontroller, which processes sensor data and controls the actuators. This manual provides a comprehensive guide for developers to set up, configure, and extend the SPCS.

2.0 Prerequisites

Before starting the designing and making of the system, it is essential for the developer to have the following prerequisites:

Hardware which will include the Arduino Uno R3, sensors, relay modules, water pump, LCD display, power supply, breadboard and the connecting wires. It is also needed for the developer to have basic knowledge on how the connections between these modules are made.

Furthermore, the software needed will include Arduino IDE, the necessary libraries (e.g., LiquidCrystal for LCD, DHT for temperature and humidity sensors), and the usage of MIT app inventor 2 for the creation of the mobile application.

In conclusion, the developer will mainly need basic knowledge on Arduino programming, MIT app inventor 2 and the understanding of electronic circuits and sensor integration

3.0 Mounting – Hardware setup

Place the following components on your workstation:

- ♣ Arduino Uno
- ♣ DHT11 Temperature and Humidity Sensor
- ♣ Soil Moisture Sensor
- ♣ 16x2 LCD with I2C Adapter
- ♣ Single Channel 5V Relay Module
- ♣ Water Pump
- ♣ Breadboard and Jumper Wires

3.1 Mounting and Wiring Steps

3.1.1 DHT11 Sensor

Connections:
VCC (Red Wire): Connect to 5V on the Arduino.
GND (Black Wire): Connect to GND on the Arduino.
Data (Green Wire): Connect to digital pin 2 on the Arduino.

Pins:
S: Signal (Data) -> Connect to Digital Pin 2
+: Power (VCC) -> Connect to 5V
-: Ground (GND) -> Connect to GND

3.1.2 Soil Moisture Sensor

Connections:
VCC: Connect to 5V on the Arduino.
GND: Connect to GND on the Arduino.
Analog Output (AO): Connect to analog pin A0 on the Arduino.

Pins:
AO: Analog output -> Connect to Analog Pin A0
GND: Ground -> Connect to GND
VCC: Power -> Connect to 5V

3.1.3 16x2 LCD with I2C Adapter

Connections:
GND: Connect to GND on the Arduino.
VCC: Connect to 5V on the Arduino.
SDA: Connect to A4 on the Arduino.
SCL: Connect to A5 on the Arduino.

Pins on the I2C Adapter:
GND: Ground -> Connect to GND
VCC: Power -> Connect to 5V
SDA: Serial Data -> Connect to A4
SCL: Serial Clock -> Connect to A5

3.1.4 Single Channel 5V Relay Module

Connections:
VCC: Connect to 5V on the Arduino.
GND: Connect to GND on the Arduino.
IN: Connect to digital pin 3 on the Arduino.
COM: Common -> Connect to one terminal of the water pump.
NO: Normally Open -> Connect to the positive terminal of the power supply for the pump.

Pins:
VCC: Power -> Connect to 5V
GND: Ground -> Connect to GND
IN: Signal -> Connect to Digital Pin 3

3.1.5 Water Pump

Connections:
Connect the negative terminal of the power supply to the pump.

Connect the positive terminal to the relay NO pin.
Connect the other terminal of the pump to the relay COM pin.

Pins:
Pump Positive: -> Connect to Relay NO
Pump Negative: -> Connect to Relay COM

3.1.6 Power Supply (9V Battery):

Connections:
Positive: Connect to the positive rail on the breadboard (if using breadboard power).
Negative: Connect to the GND rail on the breadboard.

Connect as described above, ensuring the relay switches power to the pump.

4.0 Installation and setup of application

This section will guide a developer through the installation process of Arduino IDE and the necessary libraries.

4.1 Arduino IDE installation:

The minimum system requirements as of 2024-6-13 [42]:

- Windows 7 and above
- Windows server 2012 (64-bit)
- Window server 2008 R2 SP1 (64-bit)
- A processor of Pentium 2 266 MHz processor and 128MB of RAM.
- About 600MB of free disk space for the installation

To install Arduino on Windows:

1. Download and install Arduino IDE from the official website.
2. Follow the instructions to install Arduino IDE.

4.2 Libraries installation:

1. In **sketch**, go to **Include Library** and then click on **Manage Libraries**

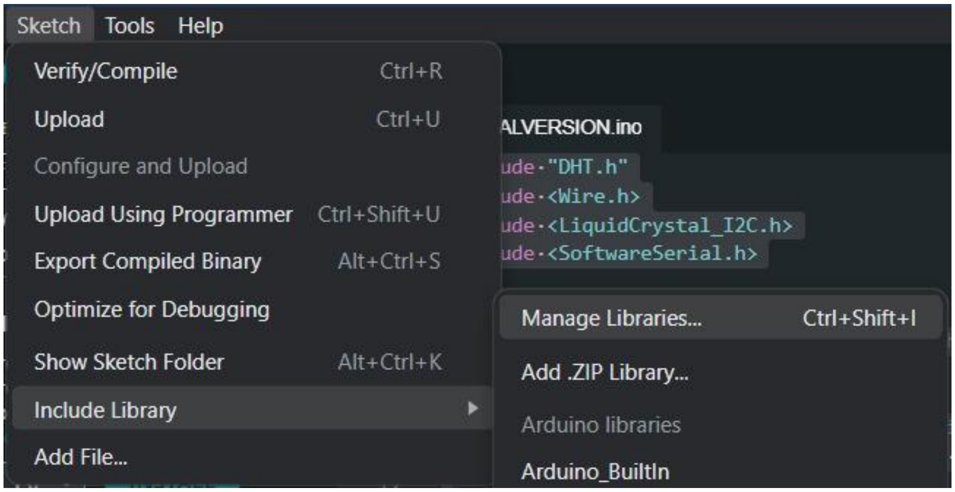


Figure 4.1: Choose Manage Libraries

Source: Author

2. A **Library Manager** will open on the right side of the screen. Search for the needed libraries. Here are the libraries that are needed in this system.

- **DHT Sensor Library**:
  - Name: DHT sensor library
  - Author: Adafruit
  - Function: Used for reading data from DHT11 sensors.
  - Installation: Available through the Arduino Library Manager.
- **Adafruit Unified Sensor**:
  - Name: Adafruit Unified Sensor
  - Author: Adafruit
  - Function: Provides a common framework for various sensors, required by the DHT sensor library.
  - Installation: Available through the Arduino Library Manager.
- **LiquidCrystal I2C**:
  - Name: LiquidCrystal_I2C
  - Author: Frank de Brabander
  - Function: Used for controlling the 16x2 LCD via the I2C interface.
  - Installation: Available through the Arduino Library Manager.
- **Software Serial**:
  - Name: SoftwareSerial
  - Author: Arduino (built-in with the Arduino IDE)
  - Function: Provides serial communication on digital pins other than the standard TX/RX.
  - Installation: This library is included with the Arduino IDE by default.

3. Search for the libraries given above and install them.

Figure 4.2: Install DHT sensor library

Source: Author

4. For some libraries, another dependency will also need to be installed. If the required dependency is not installed, a message **"Install library dependencies"** will come up. Click on **INSTALL ALL**.

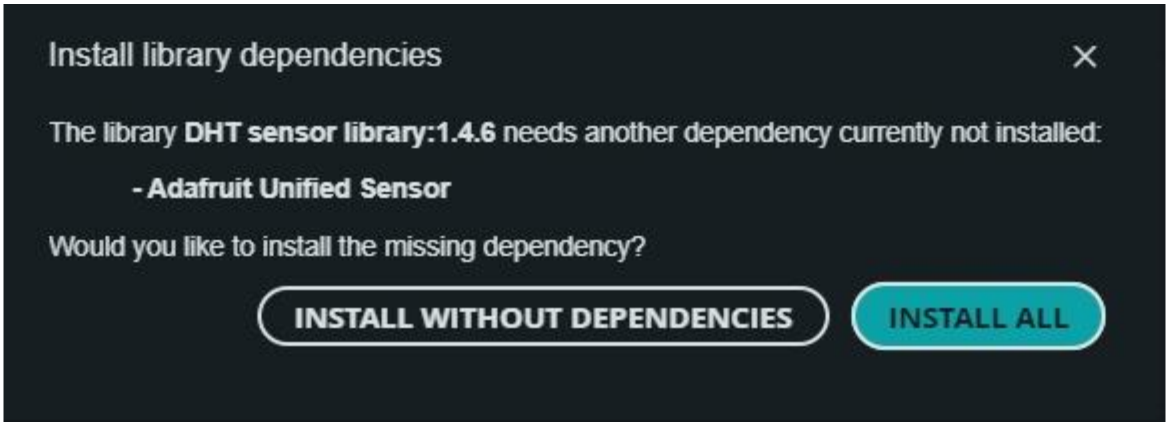


Figure 4.3: Install library dependencies

Source: Author

5. After the libraries are installed, they should show up in the library list that the developer has.

To see them, click on **Sketch**, hover down to **Include Libraries**. At the bottom, the installed libraries should be seen.

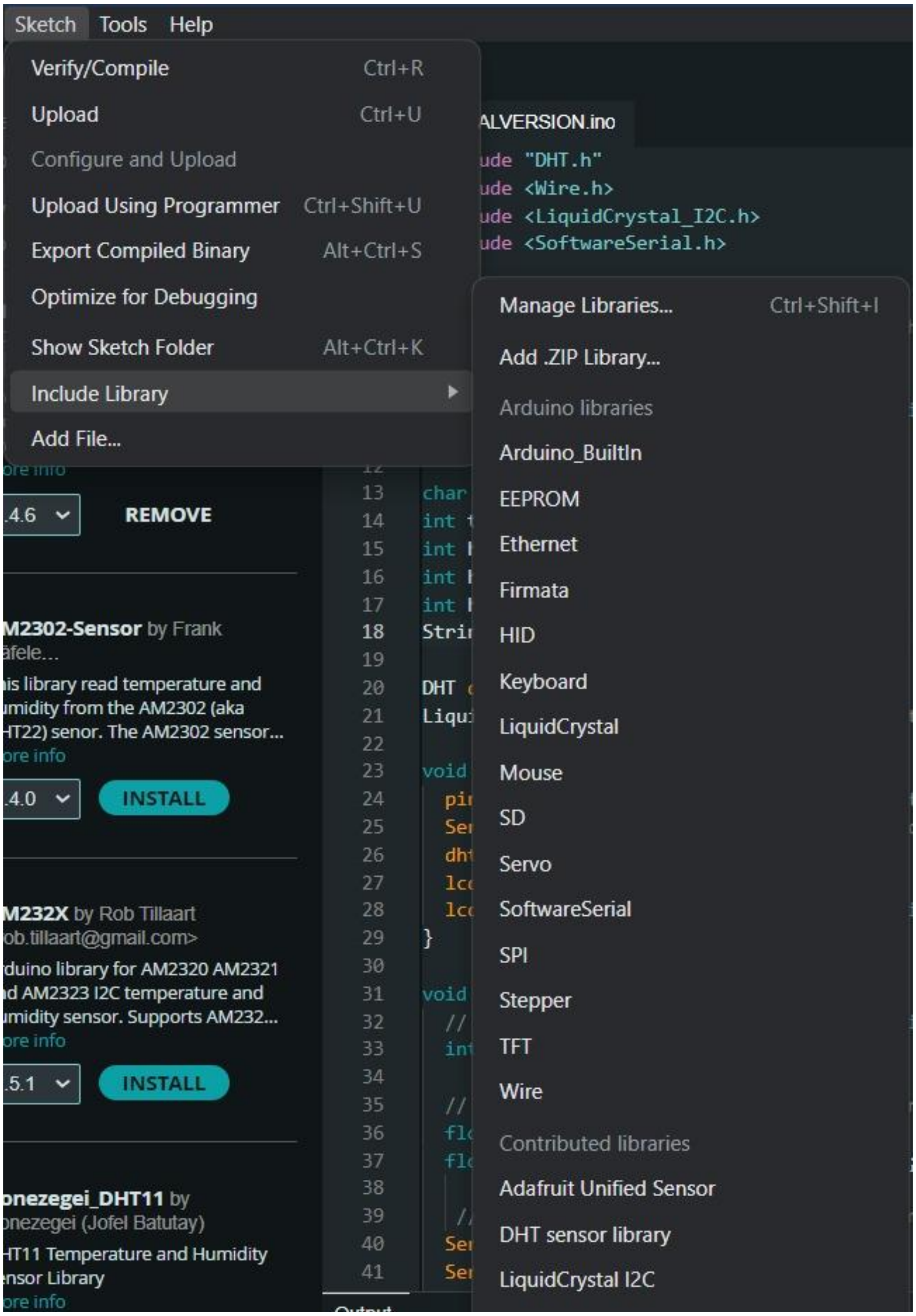


Figure 4.4: Library list

Source: Author

4.3 MIT App Inventor 2

MIT App Inventor is an online platform which is free to use and helps in creating mobile applications easily without the need for coding. It mainly uses If else statements but there are known as “when” and “do” in the block view for those people who are not familiar with coding.

1. Go to [MIT App inventor website](https://appinventor.mit.edu).

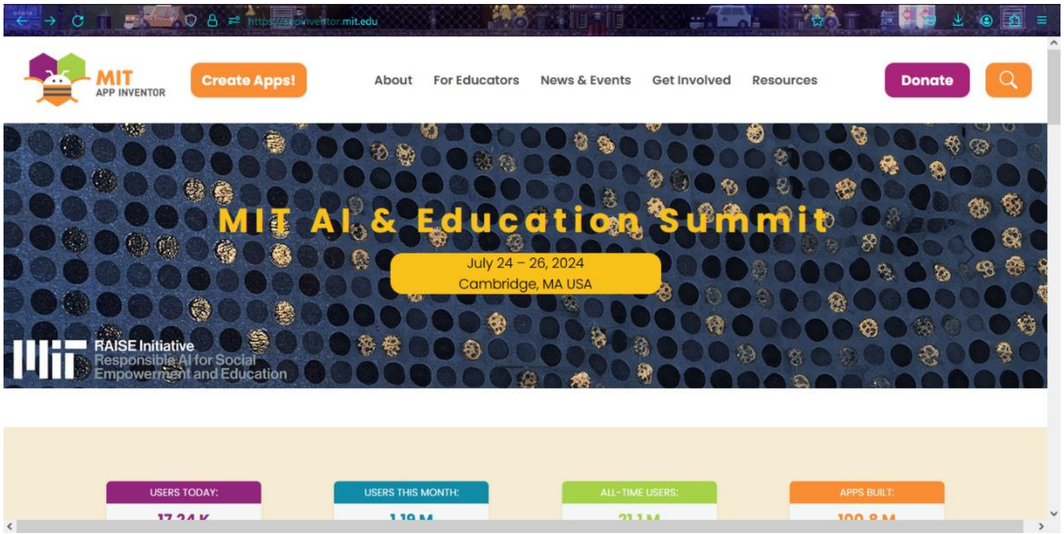


Figure 4.5: MIT App inventor main website

Source: Author

2. Click on **Create Apps!** On the top left of the screen
3. Log in with your google account.
4. Once logged in, the **My Projects** tab will open up. Here the developer can view their projects and also follow tutorials to make projects.

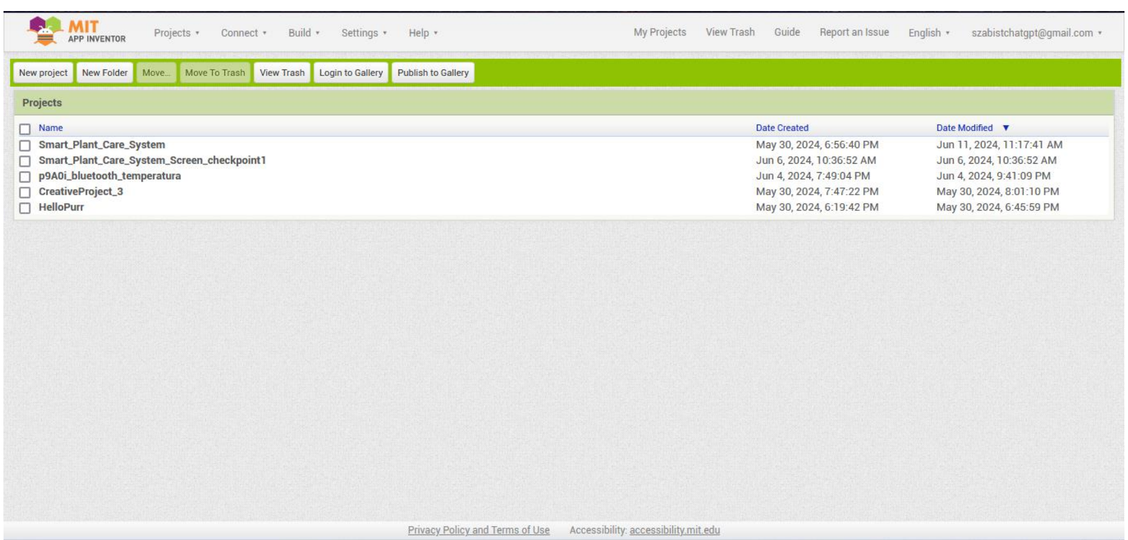


Figure 4.6: My projects

Source: Author

5. Click on **New Project**

6. In the popped up box, write in the name of the project. Also choose the toolkit which can be either Beginner, intermediate, expert or custom. The toolkit shows the set of components needed to build the application. The less complex toolkit chosen, the less components it will have. The theme changes the appearance (Dark or light) of the app based on the device.

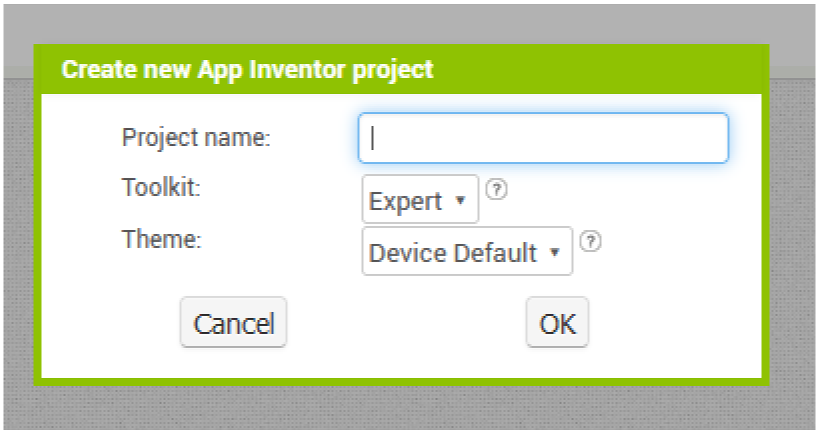


Figure 4.7: Create new App Inventor project

Source: Author

7. The application will now open and now through the use of drag and drop, the working of the application can be modified according to your needs.

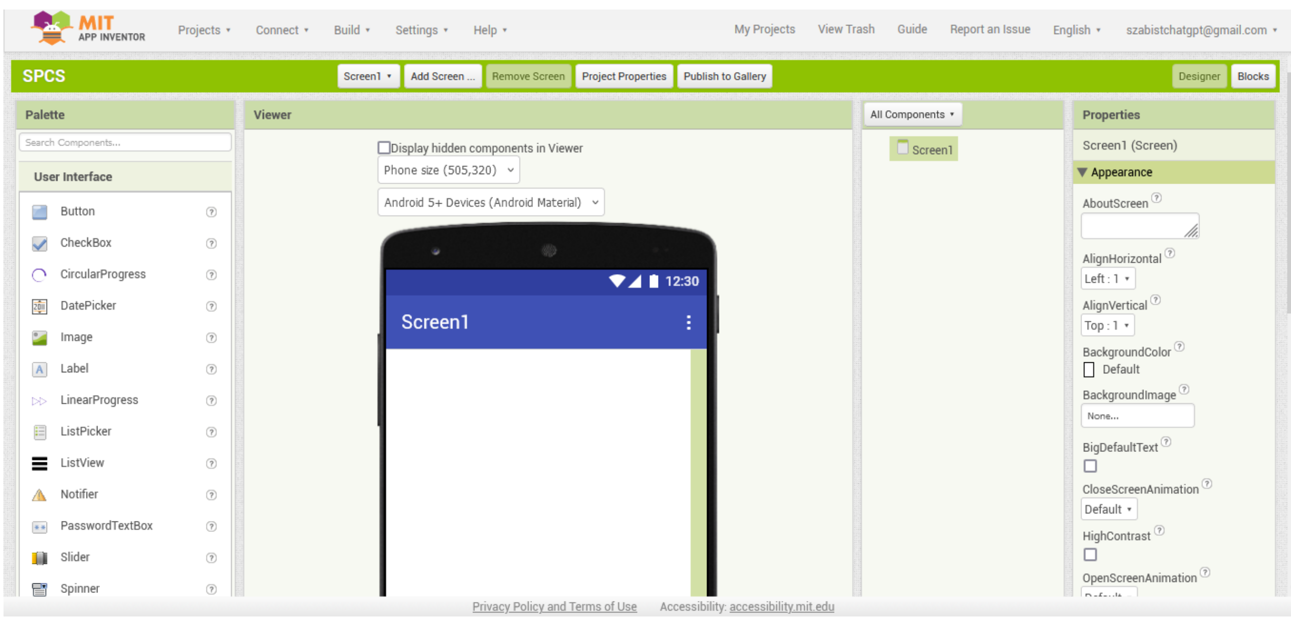


Figure 4.8: Project Dashboard

8. There are two views, a designer and block view. The designer view mainly focuses on drag and drop of components such as a button and listpicker and the designer of the UI for the customers. The Blocks view focuses on the backend of the application.

Following are the components added in every screen:

4.3.1 Main Screen Components

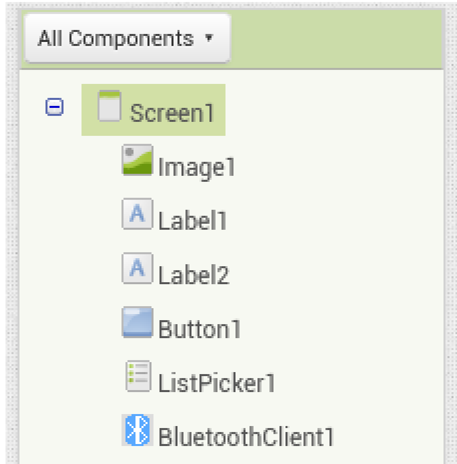


Figure 4.9: First Screen Components

Source Author

4.3.2 My Plants Components

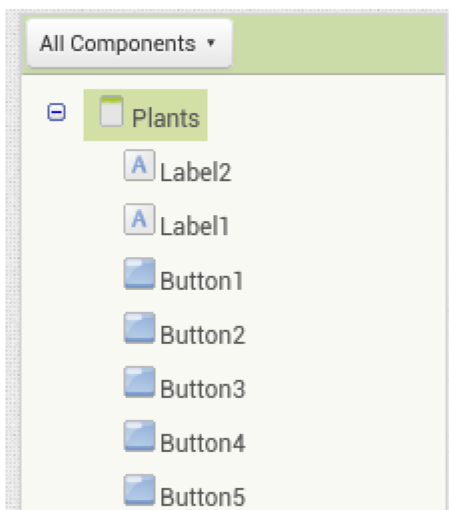


Figure 4.10: My Plants components

Source Author

4.3.3 Money Plant (and other plant information screens) Components

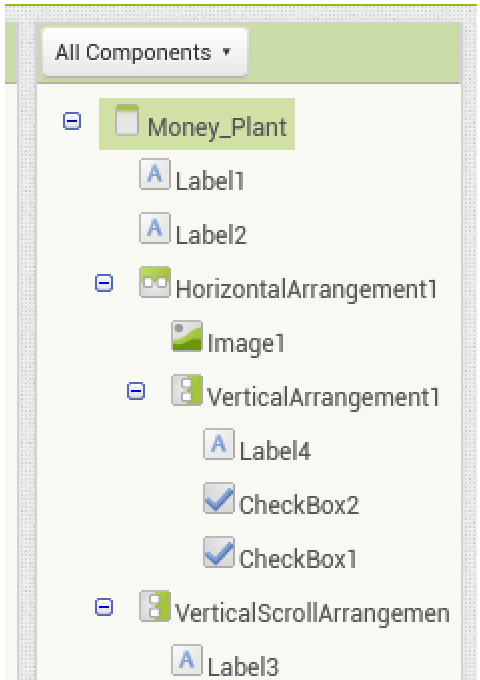


Figure 4.11: Plant Screen components

Source Author

These same components apply on the rest of the plant information screens as well with only difference being in the name, image and description of the plant.

As for the Block view, it only applies on the main screen and my plants. The following blocks should be assembled for the backend working of the app.

4.3.4 Block view for the main screen:

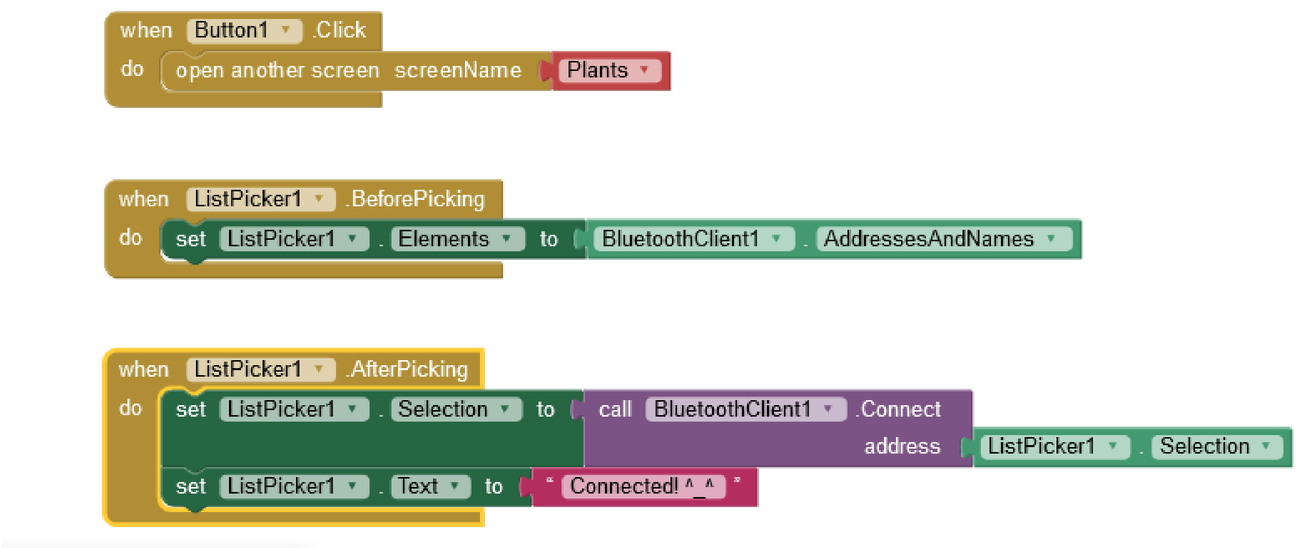


Figure 4.12: Welcome to SPCS block view

Source Author

4.3.5 Block view for My Plants screen:

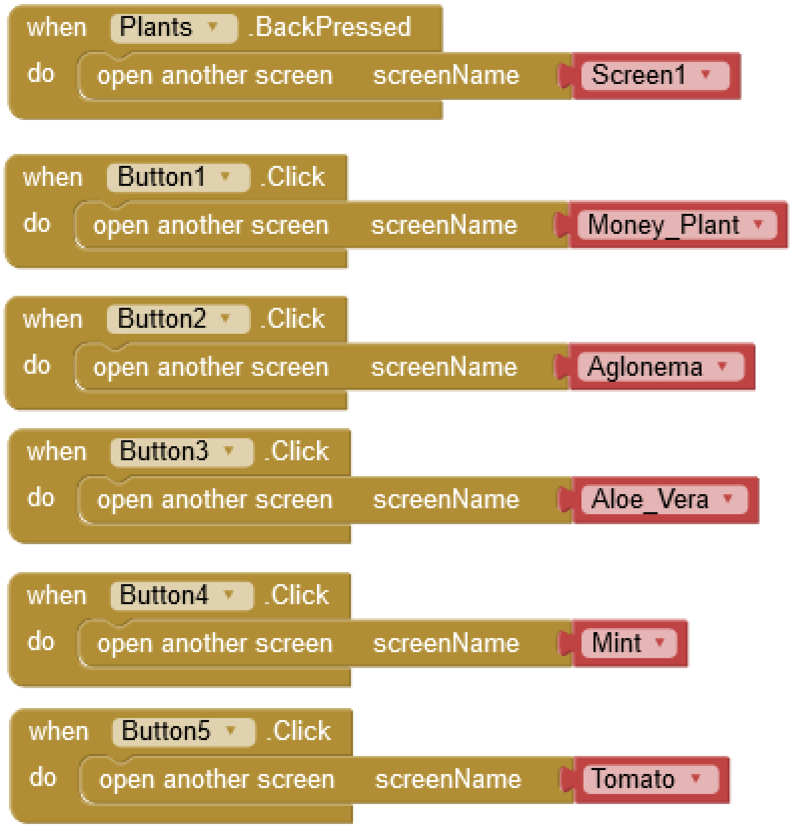


Figure 4.13: My Plants Block view

Source Author

9. Once these steps have been done, the app has to be build. Click on **Build** and then **Android App (APK)**. The **APK** and the **AIA** (file that can be imported and tested in the MIT app inventor) has been provided in the USB.

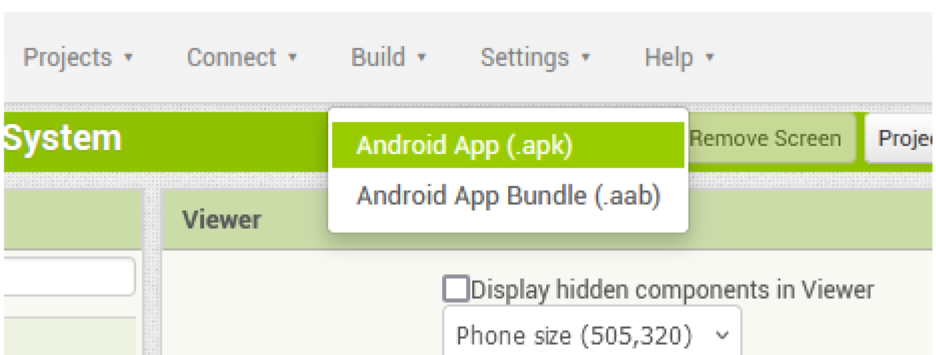


Figure 4.14: Build application

Source: Author

10. Wait for the application to be compiled and packaged.

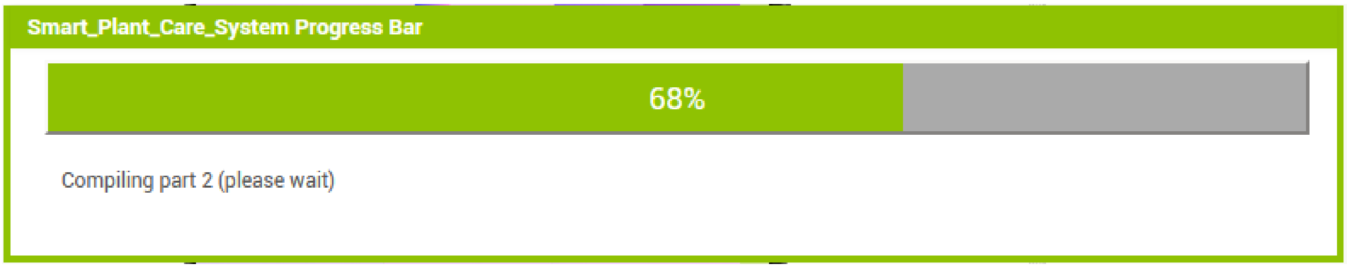


Figure 4.15: Compiling of project

Source: Author

11. A QR code will be generated. The dev. has the option to either scan it through their mobile device or download the **APK** file. The QR code is only valid for 2 hours.

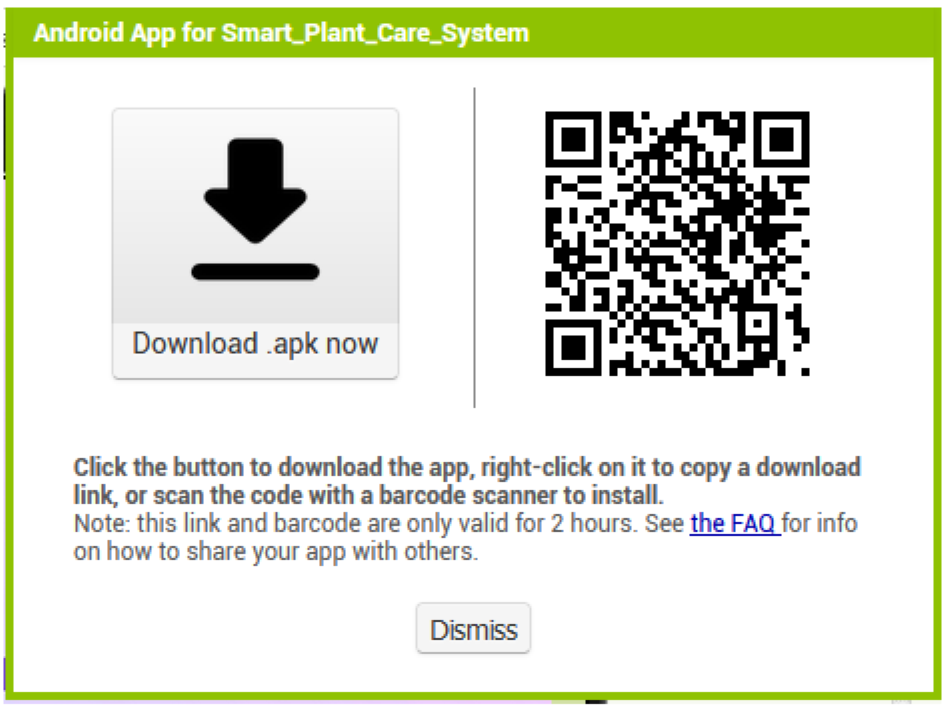


Figure 4.16: Download APK File

Source: Author

5.0 Support

In case of any other issue, please contact the following students:

Contact	Organization	Email
Abeeha Imran	SZABIST, Dubai	abeehai61@gmail.com
Fatimah Munir	SZABIST, Dubai	fatimahmunir9876@gmail.com
Meryum Tahir	SZABIST, Dubai	meryumtahir786@gmail.com

Table: Support Point of Contact

References

[42] Arduino Staff, Arduino Forms, [Online]. Available: https://forum.arduino.cc/t/minimum-system-requirements/297006/2. [Accessed 13 June 2024].

[43] T. Thamaraimanalan, S. P. Vivekk, G. Satheeshkumar and P. Saravanan, "Smart Garden Monitoring System Using IOT," *Asian Journal of Applied Science and Technology ,* vol. 2, no. 2, p. 192, 2018.

[44] M. Sheth and P. Rupani, "Smart Gardening Automation using IoT With BLYNK App," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019, pp. 266-270.

[45] S. Sharma, A. Sharma, T. Goel, R. Deoli and S. Mohan, "Smart Home Gardening Management System: A Cloud-Based Internet-of-Things (IoT) Application in VANET," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020, pp. 1-5.

[46] K. N. Baluprithviraj, M. M. Madhan, T. K. Devi, R. K. Bharathi, S. Chendhuran and P. Lokeshwaran, "Design and Development of Automatic Gardening System using IoT," in *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, Bangalore, India, 2021, pp. 750-754.

[47] M. S. Kumar, T. R. Chandra, D. P. Kumar and M. S. Manikandan, "Monitoring moisture of soil using low cost homemade Soil moisture sensor and Arduino UNO," in *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2016.

[48] S. Pushkar and S. Sanghamitra, "Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module," in *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Agra, India, 2016, pp. 1-4.

[49] D. D. P V S, M. Pachiyannan, J. S. Rani K and G. Pravallika, "A Low-Cost Arduino based Automatic Irrigation System using Soil Moisture Sensor: Design and Analysis," in *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*, Coimbatore, India, 2019, pp. 104-108.

[50] P. Bhadani and V. Vashisht, "Soil Moisture, Temperature and Humidity Measurement Using Arduino," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2019, pp. 567-571.

[51] S. A. Hamoodi, A. N. Hamoodi and G. M. Haydar, Automated irrigation system based on soil moisture using arduino board, Iraq: Institute of Advanced Engineering and Science (IAES), 2020.

[52] D. Schrock, "Water-Efficient Gardening and Landscaping," UNIVERSITY OF MISSOURI-COLUMBIA , Missouri-Columbia, 1999.

[53] R. Cardell-Oliver, M. Kranz, K. Smettem and K. Mayer, "A Reactive Soil Moisture Sensor Network: Design and Field Evaluation," *International Journal of Distributed Sensor Networks,* vol. 1, no. 2, pp. 149-162, 2005.

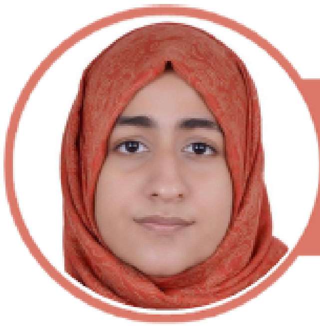
[54] F. T. Ulaby and P. P. Baltivala, "Optimum Radar Parameters for Mapping Soil Moisture," *IEEE Transactions on Geoscience Electronics,* vol. 14, no. 2, pp. 81-93, 1976.

[55] D. I. Shuman, Nayyar Ashutosh, A. Mahajan, Y. Goykhman, K. Li, M. Liu, D. Teneketzis, M. Moghaddam and D. Entekhabi, "Measurement Scheduling for Soil Moisture Sensing: From Physical Models to Optimal Control," *Proceedings of the IEEE,* vol. 98, no. 11, pp. 1918-1933, 2010.

[56] S. Vaishali, S. Suraj, G. Vignesh, S. Dhivya and S. Udhayakumar, "Mobile integrated smart irrigation management and monitoring system using IOT," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2017, pp. 2164-2167.

[57] S. S. Kanumalli, S. J. Mantena, S. Kandula, K. Doppalapudi and T. Atluri, "Automated Irrigation Management System using IoT," in *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2022, pp. 476-482.

[58] Arduino Staff, "Downloads," Arduino, [Online]. Available: https://www.arduino.cc/en/software. [Accessed 13 June 2024].



Abeeha Imran

abeehaimran.official@gmail.com
+971 50 632 3672

OBJECTIVE

To work in a well-established and career-promising organization that may offer a pleasant working environment with ample opportunities of future prospects in line with my qualifications.

EDUCATION

Bachelors (Computer Science)	Oct 2020 - June 2024	SZABIST University, Dubai

SKILLS

- Proficient in Microsoft Office
- Experienced in technical aspects and sales of several Zoho products
- Skilled in HTML, CSS, C++, SQL, and JAVA
- Experienced in VS, Eclipse, Notepad, SQL Server Management Studio, MySQL & Sublime Text
- Experienced in Arduino based technology
- Knowledge of basic, logical, and advanced circuits using programs such as Proteus
- Proficient in using tools and techniques to enhance workflow using AI
- Experienced in event organization
- Fluent in English and Urdu

EXPERIENCE

Senior Technical Consultant & Sales Advisor | IT Enablers Global | Nov 2023 - Feb 2024

- Met with clients to understand their IT needs, business processes, and areas of concern in order to provide recommendations, on integrated solutions using Zoho technologies.
- Performed end-to-end project management e.g. scoping projects, creating schedules, allocating resources effectively and ensuring project completion within the specified timeframe.
- Strategically planned, executed, and monitored multiple LinkedIn campaigns aimed at targeting new customers and prospects.
- Trained new staff on marketing and sales, along with providing them product knowledge.

Technical Consultant and Sales Advisor | IT Enablers Global | July 2023 - Oct 2023

- Assisted in lead generation and marketing
- Engaged with prospects on LinkedIn and initiated conversations to understand their needs.
- Managed data migration for clients from legacy systems to the Zoho platforms.
- Customized Zoho applications such as CRM, Zoho One, and Zoho Creator while considering their integration with Microsoft tools.
- Maintained and updated the lead pipeline, ensuring all relevant information is accurately documented.

MY WORK

- Designed and developed:
    - Multiplayer game in C++
    - Single player game in C++
    - Arcade system with multiple games in JAVA
    - Circuit simulator with several interconnected electrical functions
    - IoT Empowered Smart Plant Care system with database using Arduino Technology
    - Database system which can be used by companies to store and process data and can assist in managing the back-end on a daily basis
- Have written multiple columns in magazines and newsletters
- Social Media Manager

Currently working on Research Paper in collaboration with Murdoch University on the topic of AI and ML Domains.

ACHIEVEMENTS

- Won a prize for Best Research Paper in the ISRC 2024 at Manipal University Dubai
- MUN (Model United Nations)
- Member of IV KenKen team representing my school and the Emirate of Dubai.
- 3rd position in U.A.E Mathletics Competition (based on mental and written mathematics)
- Secured several top positions in STEM project evaluations

Professional references can be provided upon request.

FATIMAH MUNIR

050-5484091
https://www.linkedin.com/in/fatimah-munir-325129235/
fatimahmunir9876@gmail.com
Dubai, U.A.E.

PROFILE

Resourceful tech professional with a flair for turning complex challenges into streamlined solutions. Proficient in applying advanced technology strategies to optimize operational efficiency. Known for a strategic and methodical approach, I am ready to drive impactful results and contribute to the success of a forward-thinking team.

TECH/IT SKILLS

Software:

- Visual Studio
- Eclipse
- PyCharm
- Jupyter
- Microsoft Office
- Zoho
- MySQL
- Shot cut
- Cisco Packet Tracer
- Dreamweaver
- Canva
- MIT App Inventor

Languages:
C (Programming language), C++, JAVA, Python

MY WORK

- Webpages using HTML and Dreamweaver
- Games in JAVA, C++ and Python
- Database system of a museum using SQL Management System
- Automatic Plant watering system using Arduino
- Collaborating with Murdoch University on a Research paper on AI and ML

EXPERIENCE

ZOHO DEVELOPER
ITEG (IT Enablers) Dubai
2023-2024

- Customized and optimized Zoho CRM to align with clients' business processes, enhancing customer relationship management and reporting capabilities.
- Developed and integrated Zoho applications, leveraging Deluge scripting and Zoho Flow to automate workflows and improve system efficiency.

IT CONSULTANT
ITEG (IT Enablers) Dubai
2023-2024

- Provided strategic IT consulting to clients, assessing their technology needs and implementing solutions to streamline operations and drive digital transformation.
- Managed IT infrastructure projects, ensuring successful deployment of hardware, software, and network solutions within scope, time, and budget constraints.

EDUCATION

2015-2019
PRE-MEDICAL F.SC
Pakistan Education Academy (U.A.E)

2020-2024
BACHELOR'S IN COMPUTER SCIENCE (BSCS)
SZABIST University Dubai

SOFT SKILLS

Leadership
Team Work
Problem Management
Time Management
Communication

Resume

Meryum Tahir

Documentation & Verification

Meticulous and methodical document management professional with 2-year background in Telecommunication environments. Coordinates files and maintains workflows in compliance with business. Solid team player with outgoing, positive demeanor and proven skills in establishing rapport with clients. Motivated to maintain customer satisfaction and contribute to company success. Specialize in quality, speed and process optimization.

Contact

Address
Sharjah, UAE

Phone
055-1068835

E-mail
meryumtahir786@gmail.com

Skills

Quality Management

Document organization

Call management software

Microsoft Office

Customer service

Work History

2019-06 - Current

Document Specialist

Sharaf Electronics LLC, Dubai, United Arab Emirates

- Transmitted documents, organized in proper Folders month / day wise in compliance with company guidelines.
- Enforced use of standardized forms and templates to achieve uniformity of communications across departments and functions.
- Delivered scripted verification calls to customers reached via manual dialing systems.
- Ensuring the verification is done by attempting to reach customers 4 times at different timings & days.
- Overcame objections using friendly, persuasive strategies.
- Escalated concerns or problem calls to management staff.
- Recorded contact information of customers and potential customers.
- Trained and mentored new telemarketers on best practices, communication strategies and performance standards.
- Troubleshot any issues and escalated issues to

proper department.

- Answered questions with knowledgeable responses.
- Set up appointments with interested customers according to schedule availability.

2018-06 - 2019-05

Inbound Customer Service Representative

Americana Group, Sharjah, United Arab Emirates

- Answered phone with friendly greeting to create positive inbound calling experience for customers.
- Documented and detailed calls and complaints using call center's CRM database.
- Sought out training opportunities to enhance customer relationship management abilities and further boost satisfaction scores.
- Asked fact-finding questions to determine customer needs and expectations and recommended specific products and solutions.
- Maintained organized and secure customer files to facilitate customer support and follow-up.
- Resolved complaints efficiently to satisfy customers and encourage future transactions.
- Educated customers on current promotions, upgrades or new offerings available under current plan.
- Answered up to 250 incoming calls in busy, fast-paced global call center.
- Resolved customer complaints and addressed emergency requests and needs.
- Referred complex issues relating to online order system or technology to help desk for further evaluation.

Education

2015-04 - 2017-04

Matric: Computer Science

HIS HIGHNESS SHEIKH RASHID AL MAKHTOUM PAKISTANI - Dubai

SCHOOL DUBAI (IN 2015). "ICS" FROM HISS

HIGHNESS SHIEKH RASHID AL MAKHTOUM PAKISTANI SCHOOL DUBAI (IN 2017).

Bachelors: Computer Science

Szabist University Dubai

Additional Information

- Holding valid UAE Driving License
- Member of School Girls Cricket Team
- Travel Enthusiast

Document viewer

Turnitin Originality Report

Processed on: 21-Jun-2024 12:27 PKT
ID: 2406177344
Word Count: 16273
Submitted: 1

IoT Empowered Plant Care: Climate Adaptive Ga... By Meryum Tahir Fatimah Munir Abeeha Imran Muhammad

Similarity Index	Similarity by Source	
5%	Internet Sources:	N/A
	Publications:	5%
	Student Papers:	N/A

include quoted | include bibliography | exclude small matches | mode: quickview (classic) report | print | refresh | download

<1% match (Somnath Meikap, Nihal Kumar, Chandan Kumar. "An Arduino Based Modular Battery Management System for Protection and Real Time Monitoring of Lithium-Ion Cell", 2023 11th National Power Electronics Conference (NPEC), 2023)
Somnath Meikap, Nihal Kumar, Chandan Kumar. "An Arduino Based Modular Battery Management System for Protection and Real Time Monitoring of Lithium-Ion Cell", 2023 11th National Power Electronics Conference (NPEC), 2023

<1% match (publications)
Ton Duc Thang University

<1% match (Makhubela, Vusimuzi Paulus. "The Influence of Selected Consumer-Based Brand Equity Elements on Brand Preference and Purchase Intention Towards Store Brands", Vaal University of Technology (South Africa), 2024)
Makhubela, Vusimuzi Paulus. "The Influence of Selected Consumer-Based Brand Equity Elements on Brand Preference and Purchase Intention Towards Store Brands", Vaal University of Technology (South Africa), 2024

<1% match (Mabaso, C.. "The Influence of Rewards on Talent Attraction and Retention at a Further Education and Training College in Gauteng", Vaal University of Technology (South Africa), 2024)
Mabaso, C.. "The Influence of Rewards on Talent Attraction and Retention at a Further Education and Training College in Gauteng", Vaal University of Technology (South Africa), 2024

<1% match (Rishi Manoj, Athira H, Anudev Asokan, Aneesha A R, Ken Thomas Abraham, Darsana G. "Upper Gastrointestinal Endoscopic Imaging Using Vine Robots", 2022 IEEE 19th India Council International Conference (INDICON), 2022)
Rishi Manoj, Athira H, Anudev Asokan, Aneesha A R, Ken Thomas Abraham, Darsana G. "Upper Gastrointestinal Endoscopic Imaging Using Vine Robots", 2022 IEEE 19th India Council International Conference (INDICON), 2022

<1% match (Abuhishmeh, Khaled Saleh Khaled. "Risk Assessment of Reinforced Concrete Sewer Pipes Under External Loading and Adverse Environmental Conditions Using an Adaptive Neuro-Fuzzy System", The University of Texas at Arlington, 2024)
Abuhishmeh, Khaled Saleh Khaled. "Risk Assessment of Reinforced Concrete Sewer Pipes Under External Loading and Adverse Environmental Conditions Using an Adaptive Neuro-Fuzzy System", The University of Texas at Arlington, 2024

<1% match (Nunes, Patrícia Alexandra Dias Nunes. "Linking Process, Product and Performance by Raman Imaging Analysis", Universidade NOVA de Lisboa (Portugal), 2024)
Nunes, Patrícia Alexandra Dias Nunes. "Linking Process, Product and Performance by Raman Imaging Analysis", Universidade NOVA de Lisboa (Portugal), 2024

<1% match (Sudha Senthilkumar, Meghena Danasekar, Brindha K.. "chapter 15 IoT-Based Smart Gardening System Using the Cloud", IGI Global,