APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 2 (2025) 1216-1230
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Smart Autoplanter: Automatic Plant Cultivation System

Hendry Firmansah Zulkifly Ngin[1], Suhaila Mohd Yasin[1]*, Shuhaida Ismail[1]

[1] *Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: ysuhaila@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.065

Article Info

Received: 16 June 2025
Accepted: 20 November 2025
Available online: 30 November 2025

Keywords

Prototype, Smart Gardening, Urban Gardening, Plant Cultivation and Automation, mobile application

Abstract

This project aims to address the inefficiencies in cultivating plants, especially in urban areas. The Smart Autoplanter is a system that provides a way to automate plant cultivation, like watering and fertilization needs and plan harvest schedules. In addition, the system's objective is to design and develop a working prototype and have it tested to meet the user's expectation. The system's modules manages the planter , gardening routine , harvest planning , custom plant cultivation and modular gardening . By implementing this system, the project aims to reduce the plants' tending and nurturing period in urban settings, and enhance their cultivation while providing a way for users to learn along the way. The chosen methodology for this project is the Prototyping model, which includes iterative development, user feedback and continuous improvement. This project provides an innovative approach to urban gardening trend that is in line with rapidly evolving automation technologies.

1. Introduction

In the present day, technological advancement has a positive impact for human life, more so with the advancement of autonomous technology. Some parts of our everyday life are already being automated, with the use of microcontrollers and sensors such as light fixtures and window curtains. With the myriad of uses and functionality of autonomous technology in improving our daily life, one of the emerging fields of interest is smart autonomous farming. Farming traditionally centered around the usage of simple tools and manpower and usually cultivated in rural area, in which a large area of land is cleared and cultivated with the crop of choice. As people started to migrate from rural areas to a more urban area in a modern era, a large percentage of farmland is still cultivated in rural areas, making transporting the produce from rural area to populated urban areas an issue. Some people started to farm on a smaller scale or called gardening, in the urban area to decrease the dependence on produce from the rural area or even to make sure that the food that reaches their plate is in the best condition.

As people started to veer towards urban gardening to ensure their food security, several other issues started to occur. Most urban dwellers are very busy with their lifestyles, be it with their careers, education and even commuting. All of these activities took up a large chunk of their time daily, causing them to neglect their urban garden. Some people who just started gardening also have issues such as lack of knowledge and skills, causing their garden to fail to produce or even fail to grow altogether.

The aim of this system is to help user meet these challenges and allow the user to cultivate their own garden, automate manual labour and ensure timely harvest. This project constitutes attaining the three major objectives as follows:

i. To design Smart Autoplanter system with microcontrollers using the structured approach.

ii.To develop a working mobile application for Smart Autoplanter system.
iii.To test the developed Smart Autoplanter system using Functionality Testing.

The developed Smart AutoPlanter shall assist urban dwellers to deal with the issues faced by them. Smart AutoPlanter can tend their garden automatically by utilizing microcontroller and several sensors to automate the processes, so people only need to start their garden by germinating the seeds and let the system take over the cultivation until harvest. Next, the system also uses vertical towers for cultivation process, effectively reducing space needed for gardening. Other than that, the system also can help people learn about good gardening practices and plan their harvest to maximize their produce output. With the utilization of this system, users can decrease their reliance on traditional gardening practices that need tedious manual labour that can consume a large portion of their time. The system also can ensure the users of their harvest quality as long as the users adhere to the cultivation practices taught.

The rest of the paper is organized as follows: Section 2 discusses related works and existing technologies in the field; Section 3 describes the methodology; Section 4 describes the analysis and design of the proposed system; Section 5 focuses on implementation and testing and finally, the concluding section summarizes the findings and suggests potential improvements that could be made in future versions of the application.

2. Related Work

This section will explain urban gardening, soilless planting methods, vertical farming, smart farming and comparison between existing system and proposed system.

2.1 Urban Gardening

Urban gardening is part of a trend towards more parks and green areas in cities, consumption of organic, locally grown products, and a closer relationship with one's own living environment. In theory, urban gardening is the practice of cultivating plants in an urban environment, transforming concrete jungles into vibrant green spaces. It encompasses a wide range of activities, from tending a windowsill herb garden to managing sprawling rooftop farms. There are numerous benefits of urban gardening in several aspects, such as health. In a study written in Science of The Total Environment, experience of urban gardening may lead to positive health effect such as an improvement in psychological health [1].

2.2 Soilless Planting Method

Soil-less cultivation represents a modern approach to enhance the cultivation of diverse vegetable crops, involving the growth of vegetables without traditional soil. In this method, inorganic nutrients essential for plant growth are provided through irrigation water. The techniques encompassed within soil-less cultivation include hydroponics, aeroponics, and aquaponics. Hydroponics entails cultivating vegetables in a solution rich in a blend of macro and micro-nutrients. Aquaponics combines the growth of aquatic animals like snails, fish, crayfish, with hydroponic vegetable cultivation in a symbiotic environment. In aeroponics, a sealed root chamber serves as a reservoir for nutrient solution, with plants above covered by polystyrene or other materials and misted with nutrient solution to maintain constant moisture. This method has been successfully applied to various vegetables, ranging from tomatoes and chilies to onions and lettuce. Soil-less cultivation facilitates early nursery raising, disease-free seedling production, and offers benefits such as year-round production, higher productivity, and streamlined pest and weed management. In the Journal of Pharmacognosy and Phytochemistry, the study stated that with its potential for popularity in future generations, the advanced technologies and techniques involved in soilless vegetable cultivation could be considered the next generation of crop science, even serving as a gateway to establishing agricultural practices in outer space [2].

2.3 Vertical Farming

Vertical farming is the practice of growing crops in vertically stacked layers [3] instead of traditional horizontal swath of land that we are used to. Vertical farming involves growing crops throughout structures (such as a skyscraper or an old warehouse) rather than in the ground, which saves water and eliminates the need for soil. There are no weather or other natural factors that can stop food production in a vertical farm. A wide variety of plant species can achieve optimal growth rates year-round when grown in controlled environments with constant monitoring and manipulation of environmental factors like light, humidity, and temperature. Mastu Patel stated that the vertical farming strategy aims to boost efficient rate of food production due to the artificially controller environment [4].

2.4 Smart Farming

Smart farming, also known as smart agriculture, is the adoption of advanced technologies and data-driven farm operations to optimize and improve sustainability in agricultural production. Mohamed *et al*. explained that smart farming is a technology that relies on its implementation on the use of AI and IoT in cyber-physical farm management [5]. Smart agriculture addresses many issues related to crop production as it allows monitoring of the changes of climate factors, soil characteristics, soil moisture, and many more. The Internet of Things (IoT) technology is able to link various remote sensors such as robots, ground sensors, and drones, as this technology allows devices to be linked together using the internet to be operated automatically [6].

2.5 Existing smart gardening system

This section discusses the existing systems which are similar to the proposed smart gardening system. The existing systems are AeroGarden, Lettuce Grow Farmstand and Vivosun Hydroponic Grow Kit.

2.5.1 AeroGarden

An AeroGarden is a self-contained indoor gardening system that uses hydroponics and LED lights to grow plants without soil. It's essentially a countertop greenhouse that simplifies the process of growing fresh herbs, vegetables, and even some flowers all year round, regardless of the weather for gardening experience. Some key features of AeroGarden are utilization of hydroponic technique instead of soil-based techniques, full spectrum LED light to provide optimal light for plant growth and some automation features such as automatically controlled light cycle and water level monitoring [7]. Figure 1 shows AeroGarden device.

Fig 1 AeroGarden [7]

2.5.2 Lettuce Grow Farmstand

Lettuce Grow Farmstand is another player in the indoor gardening game, aiming to simplify fresh produce cultivation for everyone. Essentially, it's a modular hydroponic system that can be used to grow a variety of herbs, vegetables, fruits, and even edible flowers indoors or outdoors. Some notable key features include stackable modular design to increase planting capacity, automatic watering and fertilization and can be used outdoor. Figure 2 shows Lettuce grow Farmstand.

Fig 2 Lettuce Grow Farmstand

2.5.3 Vivosun Hydroponic Grow Kit

Vivosun hydroponic grow kits are complete setups designed for indoor gardening using hydroponics, a method where plants grow in nutrient-rich water instead of soil. These kits come in various sizes and configurations, catering to different plant types, budgets, and grower experience levels. Notable features include a fully featured hydroponic system, water pump and timer, grow light and included nutrient kit that can be diluted in water before use [8]. Figure 3 shows Vivosun Hydroponic Grow Kit.

Fig 3 Vivosun Hydroponic Grow Kit [8]

2.6 Comparison Between Existing System and Proposed System

Table 1 shows the comparison between existing smart gardening systems which are AeroGarden, Lettuce Grow Farmstand and Vivosun Hydroponic Grow Kit with the proposed system which is Smart Autoplanter. The features included login, internet of things (IoT) hardware, soilless method, vertical cultivation, grow light, automation, information module, modularity and control panel design.

Table 1 Comparison between similar system with proposed system

Features	AeroGarden [7]	Lettuce Grow Farmstand	Vivosun Hydroponic Grow Kit [8]	Proposed System
Login	Yes	Yes	No	Yes
Internet of Things (Iot) hardware	Yes	Yes	No	Yes
Soilless method	Yes	Yes	Yes	Yes
Vertical Cultivation	No	Yes	No	Yes
Grow light	Yes	Yes	Yes	Yes
Automation	Yes	Yes	Partial	Yes
Information module	Yes	Yes	No	Yes

Modularity	No	Yes	Yes	Yes
Control Panel Design	On device	No	No	On device, web interface

For summary, all three existing systems and the proposed system have their own set of advantages and disadvantages. For that, the proposed system will integrate all the beneficial features of all existing systems into it.

3. Methodology

The model chosen to be used for the development of this system is the prototyping model. Prototyping model is software or system development method that involves the development of the initial version of the system, which is a prototype to describe and explain user requirement. The selection of this methodology depends on the suitability of the project development, which involves implementing iterative improvements based on user feedback to generate a prototype that matches with the defined user requirements[9]. Figure 4 shows Prototyping model illustration of each phase.

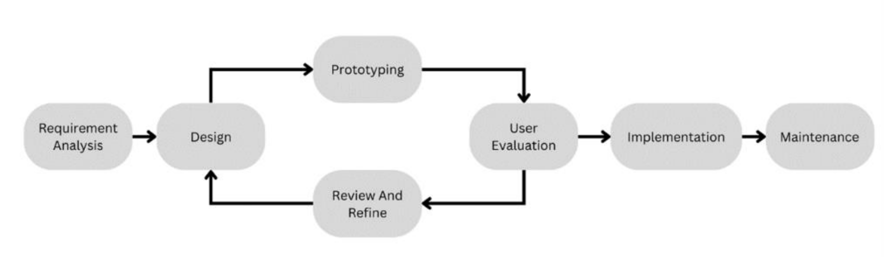


Fig 4 Prototyping Model

3.1 Requirement Analysis Phase

This requirement analysis phase is the first phase of the entire project development. In this phase there is also a process of gathering information through observation sessions and interviews with stakeholders to obtain information such as user requirements in the system, identifying problem statements, making research and references through the internet and various sources, determining the required modules in system to use and so on. In addition, the creation of a Gantt chart is also done to plan the duration of the project according to the phases in the prototyping model so that the development of this system can be developed according to the set time by using Microsoft Excel software. Going through this planning phase helps in examining the problems faced by stakeholders and planning how to solve those problems through the development of the system that will be developed to achieve the scope of the project.

In this analysis phase, the focus on the existing flow chart system used by the stakeholders, produced entity relationship diagram (ERD) and data flow diagram (DFD). The stakeholders' understanding and documentation of the current system are the main priorities throughout the requirement analysis phase. To obtain insight into the current processes and interactions, a thorough analysis of the current flow chart system is necessary. The ERD and the DFD are two important diagrams created to give a visual depiction of the system's structure. In addition to providing a clear picture of how data is arranged and integrated, the ERD shows the relationships between various entities inside the system while DFD shows how data moves across the system and emphasizes the various data flows, storage, and processes that make up the overall architecture of the system.

Furthermore, the analysis process of functional and non-functional system requirements is also important to include in documentation, in addition to graphical representations. Functional requirements specify the features and capabilities of the system that are intended, whereas non-functional requirements deal with elements like performance, security, and usability. To guarantee that the system satisfies the requirements and expectations of its intended users, user requirements are also established. The technological foundation required for the system's effective implementation is established by outlining the hardware and software requirements. The system development phases that follows are built upon this thorough analysis, which is supported by several diagrams such as DFD, ERD, and flowcharts.

3.2 Design Phase

In the design phase, various tools, such as Draw.io, are employed to create comprehensive designs, particularly for the analysis frames and outputs related to ERD and DFD structural approaches. The initial focus is on the user interface, integrating graphic database design. This involves creating the interface design in initial sketches and developing the database schema, which includes defining entity elements, attributes, and establishing relationships between them. Emphasizing the creation of system architecture, database structure, and user interface blueprint is paramount during this stage. Utilizing online platforms, especially for ERD and DFD, streamlines the process by leveraging tools like draw.io.

A pivotal outcome of this phase is system architecture, providing a high-level overview of the project scope and illustrating the arrangement and interaction of modules. Developing a database schema and data dictionaries is equally important, involving the characterization of entities, their attributes, and establishing connections. The schema aids database management by offering a visual representation of the database structure, while data dictionaries document the attributes of each component.

Concurrently, the user interface is crafted, beginning with wireframes that outline the structure and layout of the interface. These wireframes serve as a visual guide for component placement and system interaction. Graphic database design is also integrated during this phase to ensure a seamless transition between the user interface and the underlying database. This encompasses thoughtful consideration of how users access information, leading to the creation of an intuitive and user-friendly interface. The comprehensive design process encompasses the development of a detailed system architecture, specification of data dictionaries and database structure, and the design of the user interface using wireframes and visual components. Collectively, these elements establish a robust foundation for subsequent phases in system development.

3.3 Prototyping Phase

The prototyping phase holds significant importance as it marks the initial construction of the system in the early stages of development. During this phase, a system prototype is built, incorporating several forms of designs including diagrams, databases, system modules, and interfaces formulated in the analysis phase. This construction is facilitated by utilizing integrated development environments such as Arduino IDE. The development of this system prototype serves the crucial purpose of testing and validating ideas, features, and functionalities before progressing to full-scale development. This iterative process allows for refinement and enhancement based on feedback, ensuring the eventual creation of a well-functioning and effective system.

3.4 User Evaluation

During the user evaluation phase, the system prototype undergoes a comprehensive assessment and testing. This evaluation aims to gather feedback from end users, focusing on the assessment of functionality, usability, and overall user experience. The goal is to pinpoint any issues that may arise during the prototype evaluation, identify areas for improvement in both the user interface and the system's functionality, and ensure alignment with the specified user requirements. This iterative process involves implementing recommendations from user evaluations to refine the user interface, enhance the user experience of the prototype system, and optimize functionality. By integrating user feedback into the development process, continuous improvements are made, contributing to the ongoing enhancement of the final product.

3.5 Review and Refine Phase

The review and refinement phase occurs subsequent to the user evaluation phase, drawing upon the feedback gathered from users. This feedback is subjected to careful and technical review and analysis to enhance various aspects of the prototype, including design, code quality, features, and overall functionality. The iterative nature of this process necessitates thorough documentation of each analysis, and subsequent updates to the prototype are made based on the received feedback. The goal is to ensure that the developed prototype aligns seamlessly with the defined scope and objectives of the system, with continuous improvements implemented to refine and optimize its performance.

3.6 Implementation Phase

The implementation phase encompasses the comprehensive development of the prototype, building upon the input gained from the review and refinement phase. This stage involves the complete realization of the prototype, including the writing of code, designing the database schema, and integrating various components to produce the final iteration. Continuous enhancements are made throughout this phase, refining the system until it becomes a fully functional and detailed solution. It ensures the inclusion of all necessary modules to fulfil the requirements of users effectively.

3.7 Maintenance Phase

The maintenance phase in system prototype development follows the implementation stage. It constitutes an ongoing process aimed at ensuring and enhancing the functionality of the system prototype post-development and usage. Continuous monitoring of the system is a key aspect of this phase, enabling the identification of any issues or performance problems that may arise. This monitoring process serves as a valuable tool for developers to update the system in response to user needs and feedback, contributing to the sustained improvement and optimization of the overall system performance.

4. System Analysis and Design

System analysis and design show the system requirements, functional and non-functional requirements, user requirement analysis, context diagram, data flow diagram, activity diagram, entity relationship diagram (ERD) and user interface design.

4.1 Functional and Non-Functional Requirements

The system requirements serve to elaborate on the functions, services, and operational constraints of the system to be developed in greater detail. This comprehensive understanding is crucial as it necessitates careful examination and implementation for the effective functioning of the constructed system. Furthermore, the system requirements are categorized into two distinct categories: functional and non-functional. Tables 2 and 3 present the functional and non-functional requirements, respectively.

Table 2: Functional requirements.

No	Module	Description
1	Registration Module	• Users must fill out the necessary information. • The web interface should include the functionality for new users to register before logging in. • The system should be designed to display any encountered errors, such as empty fields, invalid email addresses, or email addresses that already exist in the system.
2	Log in Module	• To access this system, both users must log in using registered email address and password. They will have the option of logging out or remain in the application as needed. • The system should provide notifications to users in the case of any invalid input. • The system should automatically lead the user to the right pages based on their assigned roles upon successful login.
3	Gardening Routine Module	• The system allows the user to initiate a cultivation cycle. • Users can select available cultivation plans and initiate the cycle on the auto-planter system. • The system should show any status update to inform the user of cultivation status.
4	Harvest Planning Module	• Users can choose to use predefined harvest plan or create a new harvest plan. • Remaining harvest time will be shown on the OLED screen on the auto-planter device.
5	Custom Plant Cultivation Module	• The system will provide a way for users to add custom cultivation plans according to the cultivation needs. • Added cultivation plan can be used in the Gardening Routine module to automate the cultivation process.

Table 3: Non -Functional requirements.

No	Requirement	Description
1	Operational	• This mobile app can be used on android mobile phone.

		• The auto-planter device needs to be connected to the local network at least during initial cultivation plan upload process. • The mobile app can be accessed from anywhere as long there is internet connection.
2	Implementation	• The interface should be straightforward to use. • The auto-planter can be used indoor and outdoor as long there is electrical source.
3	Security	• Only the Administrator Login has been set by the application developer.

4.2 User Requirements Analysis

In terms of user requirements, the Smart Autoplanter has two users: the Administrator, the user added by the system developer for maintenance purposes, and the User, who is the auto-planter owner. The needs of both users on this system are described in Table 4.

Table 4: Explanation about User Requirements.

No	Requirement
1	All users should be able to enter their email address and password for registration and login requirements.
2	Users can start a cultivation cycle based on the plans available.
3	Users will get the status of current cultivation plan status on OLED screen on the auto-planter.
4	All users can add custom cultivation plans for the system.
5	All users can add custom harvest plans for the system.
6	Users should be able to register their auto-planter under their account in the mobile application.
7	Users should be able to connect to their auto-planter registered to their account from the mobile application.
8	Users should be able to insert the size of their planting module stack to the system.
9	Users should be able to view the system resources such as light intensity, remaining nutrient availability and water level.

4.3 Context Diagram

Context diagrams present an overview of interaction between the system and its user. Context diagrams also show the input and output to and from its user and system. Figure 5 shows the context diagram of the Smart Autoplanter system.

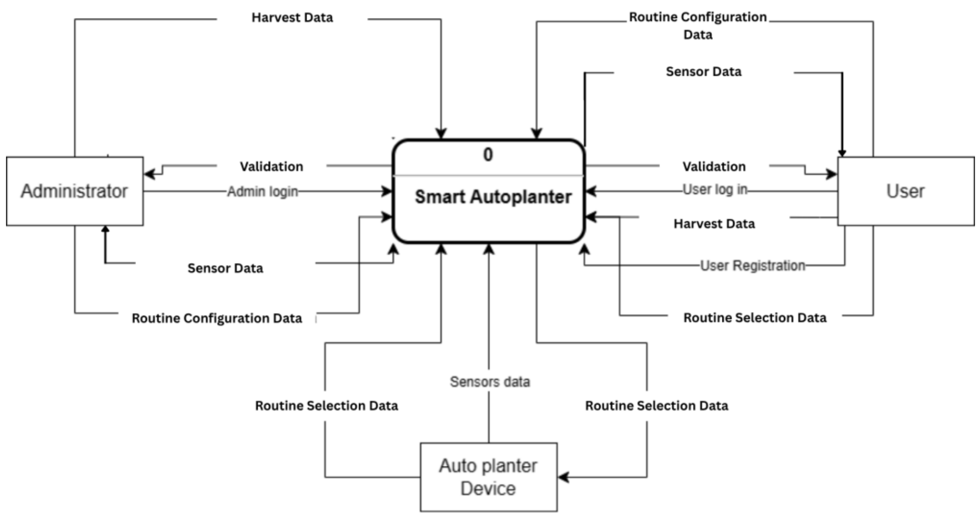


Fig 5 Context Diagram

Figure 5 shows the context diagram with three external entities namely Administrator, User, and Autoplanter device. The administrator and user represent end users who will use the system to automatically cultivate plants based on the selected parameters. The administrator's entity roles show the function of the first system operator who will assign predefined information and parameters for the system to work out of the box. The administrator entity has complete control over the system, while the user entity has slightly limited control over the system. For the auto-planter entity, it's a representation of the auto-planter device that will automate the cultivation process. This diagram makes it simple to understand how data flows between the Smart Autoplanter system and its external partners. It also puts the system's features in perspective within the larger framework of user and administrative control.

4.4 Data Flow Diagram

DFD is a graphical representation of the flow of data or input from an entity through a process, which then generates output either to another entity or stored in data storage. DFD shows each input and output for each entity and process. Figure 6 shows the Level 0 Data Flow Diagram (DFD 0) of the developed system.

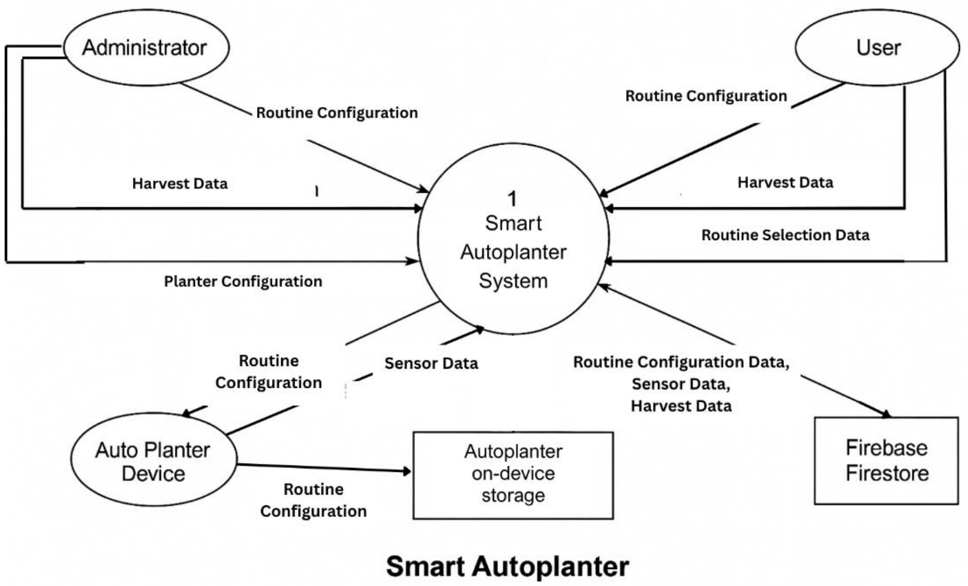
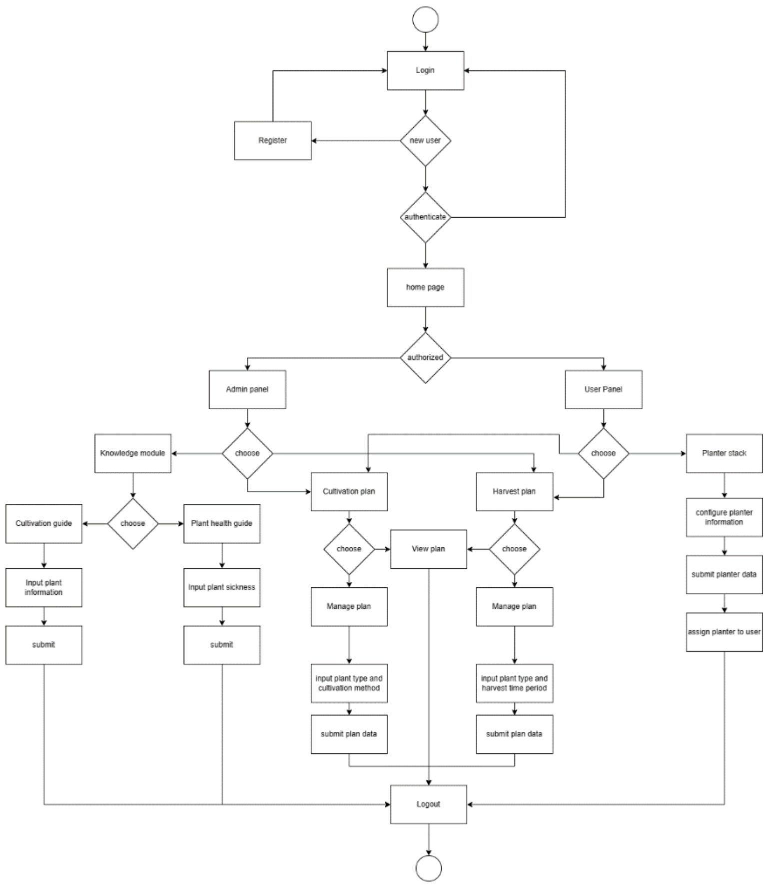


Fig 6 Data Flow Diagram

4.5 Flow Chart Diagram

Structure 7 shows the flow chart diagram of the proposed system. New users will need to register first before logging in to the system. To register, the user will need to input their email, username, UserID and password. Next, to login, users need to use their UserID and password. After that, the system will verify the users and redirect the user to the main menu page based on their account privilege. Both users will have a similar interface with minor differences between privilege access. In the dashboard interface, there are several options that can be selected by users. For Administrator, they can choose to enter knowledge module with the option for cultivation guide. They can also choose a cultivation plan and harvest plan to set predefined parameter for the auto-planter system to work. For Users, they can view and commit cultivation plan and harvest plan. Users also can assign planters to their account.



Struct 7 Flowchart Diagram

4.6 Entity Relationship Diagram

An ERD visually represents the relationships among entities in a database system, utilizing various symbols to illustrate the organizational structure of the database. These relationships define how actual objects or conceptual entities are interlinked. Additionally, entity attributes convey details about the characteristics or qualities of these entities. ERDs serve as a valuable tool for conceptualizing, planning, and communicating the database structure, aiding database designers and stakeholders in comprehending, and designing the relationships between different data elements. This facilitates the maintenance of integrity and effectiveness in the overall architecture of the database system.

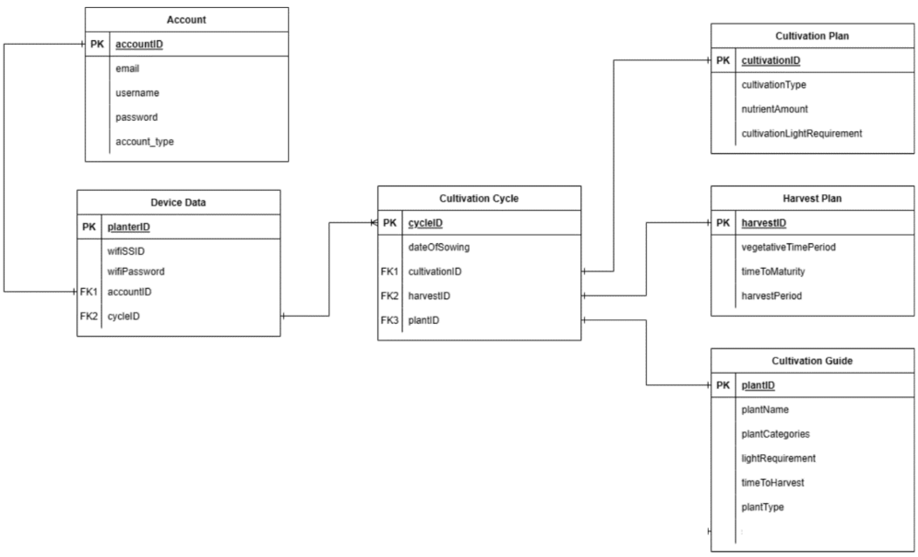


Fig 8 Entity Relationship Diagram (ERD)

4.7 User Interface Design

The interface design is an early drawing of the Smart Autoplanter interface. The interfaces draft that will be used in the development of this system are shown in Figures 9, 10, 11 and 12 respectively.

4.7.1 Login Page

The user interface draft for login page into the system using UserID and password is shown in Figure 8. Users enter their UserID and password in order to log in. After the information is typed in, press the Login button to login into the system. Users will be forwarded to dashboard according to their respective account privilege, normal user, or administrator.

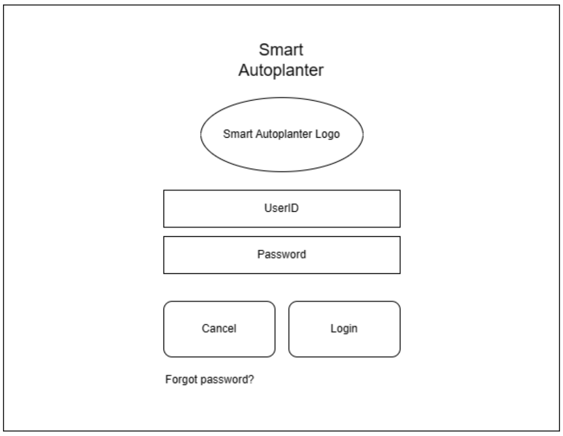


Fig 9 Login Page Wireframe Draft

4.7.2 Registration Page

The user interface draft for account registration into the system using email, username, UserID and password is shown in Figure 9. Users must submit their email, username, UserID and password in order to register. After the information is typed in, press the register button to register account.

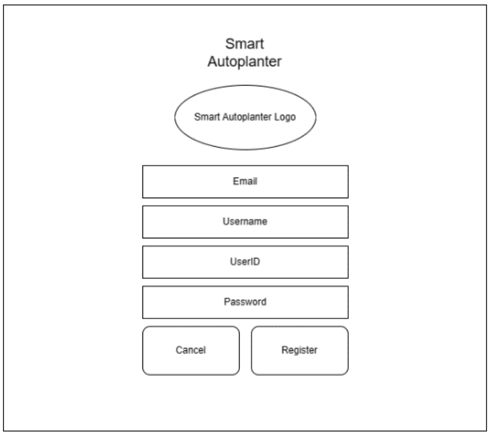


Fig 10 Registration Page Wireframe Draft

4.7.3 Cultivation Plan Page

The user interface draft for cultivation plan page is shown in Figure 10. Users enter new cultivationID to create new cultivation plans or select from list of predefined cultivationID to customize or send to planter. Select a plant type from the list or enter a new plant type if the type is not available on the list. Select the number of days from the calendar or enter the growing period in day into the data field. Finally, select predefined nutrition amount from the list, either leafy, flowering, or fruiting nutrition plan. Users can also enter custom nutrition amount if they wish to do so.

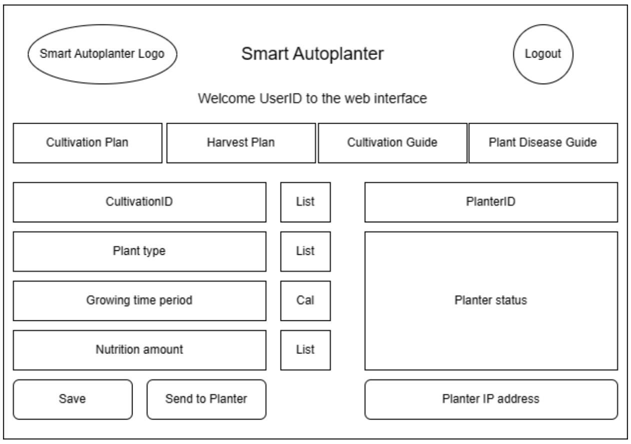


Fig 11 Cultivation Plan Page Wireframe Draft

4.7.4 Harvest Plan Page

The user interface draft for harvest plan page is shown in Figure 4.11. Users enter a new harvestID to create new harvest plan or select from list of predefined harvestID to customize or send to planter. Select the sowing date from the calendar date popup. Select the number of days from the calendar or enter vegetative growth period in day into the data field. Finally, enter time needed for the plant to mature or select from the calendar popup. After all the parameters are set, click Send to Planter to initiate cycle.

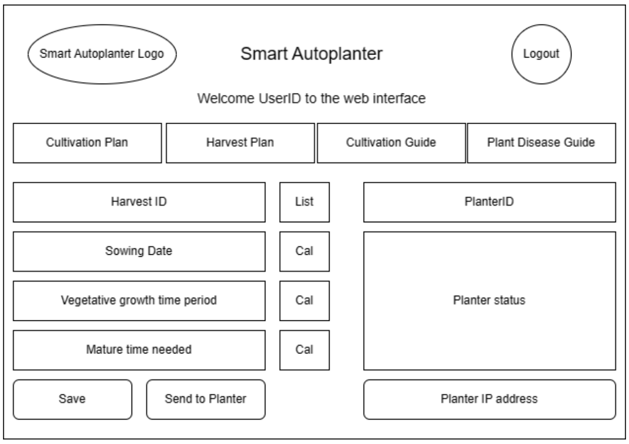


Fig 12 Harvest Plan Page Wireframe Draft

4.8 Functionality Testing

The testing phase focuses on assessing the functionality of the system. System testing is conducted to ensure that the developed Smart Autoplanter system meets both functional and non-functional requirements. Completion of system testing is determined by the proper functionality of all modules, as aligned with the specified system requirements. The list of test cases and their outcomes is presented in Table 5.1.

Test Case ID	Description	Test Status
TC_100	System Initialization and User Interface	
TC_100_01	The system must allow users to power on and initialize the Smart Autoplanter.	PASS
TC_100_02	The interface must allow users to configure basic settings (network SSID, network password).	PASS
TC_100_03	The system must validate user input for configuration.	PASS
TC_100_04	The system shall display configuration status without restarting the system.	PASS
TC_100_05	The system must notify the user of hardware or connection errors.	PASS

TC_200	Registration Module	
TC_200_01	The system must be able to receive email and password from user	PASS
TC_200_02	The system must be able to authenticate with Firebase Auth for Authentication	PASS
TC_200_03	The system must be able to store user registration data in database.	PASS
TC_200_04	The system must alert users if their account already registered.	PASS
TC_200_05	The system must alert user if their registration is successful.	PASS
TC_300	Login Module	
TC_300_01	The system must able to receive email and password from user.	PASS
TC_300_02	The system must be able to authenticate user with firebase Auth	PASS
TC_300_03	The system must be able to redirect user to dashboard after successful login	PASS
TC_300_04	The system must be able to alert user if their account is not registered or the password entered is wrong.	PASS
TC_400	Gardening Routine Module	
TC_400_01	The system must able to add routine configuration to the system	PASS
TC_400_02	The system must able to save added routine into the database	PASS
TC_400_03	The system must able to send the configuration to the planter	PASS
TC_400_04	The system must be able to edit the configuration according to user needs	PASS
TC_500	Planter Module	
TC_500_01	The system must be able to add planter to the system	PASS
TC_500_02	The system must be able to assign planter to the user	PASS
TC_500_03	The system must be able to sync configuration of planter with the system and the device	PASS
TC_500_04	The system shall be able to edit planter configuration as needed	PASS
TC_500_05	The system must be able to delete unused planter as needed by user	PASS
TC_600	Harvest Calendar Module	
TC_600_01	The system must be able to display calendar	PASS
TC_600_02	The system must be able to calculate harvest date from the routine	PASS
TC_600_03	The system must be able to provide current harvest period for specific planter	PASS
TC_600_04	The system must be able to provide coming harvest for specific planter	PASS

5. Conclusion

This project is conducted to solve problems 1, 2 and 3. The development of the Smart AutoPlanter has successfully fulfilled the three main objectives established at the beginning of the project.

The first objective is to design an autonomous plant cultivation system using structured approaches. The project employed a systematic and structured development methodology, prototyping methodologies and iterative refinement. Through careful planning and modular system architecture, the Smart AutoPlanter was designed with automation in mind, integrating key components such as water sensors, nutrient sensors, and microcontroller logic. The resulting design enables automatic planting operations with minimal user intervention, aligning closely with the first objective.

Next, to develop a working mobile application for the automatic plant cultivation device system. A mobile application was successfully developed to provide user interaction with the Smart AutoPlanter system. The app allows users to input cultivation parameters, monitor real-time data from sensors, and initiate or stop planting operations remotely. The user interface is intuitive and functional, demonstrating the successful completion of the second objective by offering a convenient control interface for users to manage the cultivation process.

Finally, the last objective is to test the developed Smart AutoPlanter system using Functionality Testing. The results indicated that the system met the expected functions and provided satisfactory user experience, thus validating the effectiveness of the Smart AutoPlanter in real-world scenarios. This confirms the achievement of the third and final objective.

In conclusion, all three core objectives of the project were achieved, demonstrating the feasibility and practicality of the Smart AutoPlanter as an effective automated cultivation solution. the development of smart auto-planters will make automating plant cultivation easier within urban settings. Furthermore, it will reduce the dependence on outside food sources while ensuring food security. This is because the harvest can be produced in urban environments, reducing the dependance on rural produce that might not be able to cultivate the food of our choice. The system also will make cultivation process easier for users as most of the tedious tending process is automated by the system.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Hendry Firmansah Bin Zulkifly Ngin, Suhaila Mohd. Yasin, Suhaida Ismail;* ***data collection:*** *Hendry Firmansah Bin Zulkifly Ngin, Suhaila Mohd. Yasin, Suhaida Ismail;* ***analysis and interpretation of results:*** *Hendry Firmansah Bin Zulkifly Ngin, Suhaila Mohd. Yasin, Suhaida Ismail;* ***draft manuscript preparation:*** *Hendry Firmansah Bin Zulkifly Ngin, Suhaila Mohd. Yasin, Suhaida Ismail. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] D. Schram-Bijkerk, P. Otte, L. Dirven, and A. M. Breure, "Indicators to support healthy urban gardening in urban management," *Science of The Total Environment*, vol. 621, pp. 863–871, Apr. 2018, doi: 10.1016/J.SCITOTENV.2017.11.160.

[2] K. Maya Waiba, P. Sharma, A. Sharma, S. Chadha, M. Kaur, and C. Author, "Soil-less vegetable cultivation: A review," ~ *631 ~ Journal of Pharmacognosy and Phytochemistry*, vol. 9, no. 1, 2020, [Online]. Available: http://www.phytojournal.com

[3] J. Birkby, "A program of the National Center for Appropriate Technology • 1-800-346-9140 • www.attra.ncat.org," 2016. [Online]. Retrieved June 11, 2025, from: www.ncat.org

[4] M. Patel, "Future Food Production Methods Include Vertical Farming and Agriculture in Controlled Environments," 2023. [Online]. Available: https://www.researchgate.net/publication/369753620

[5] E. Said Mohamed, A. A. Belal, S. Kotb Abd-Elmabod, M. A. El-Shirbeny, A. Gad, and M. B. Zahran, "Smart farming for improving agricultural management," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 24, no. 3, pp. 971–981, Dec. 2021, doi: 10.1016/J.EJRS.2021.08.007.

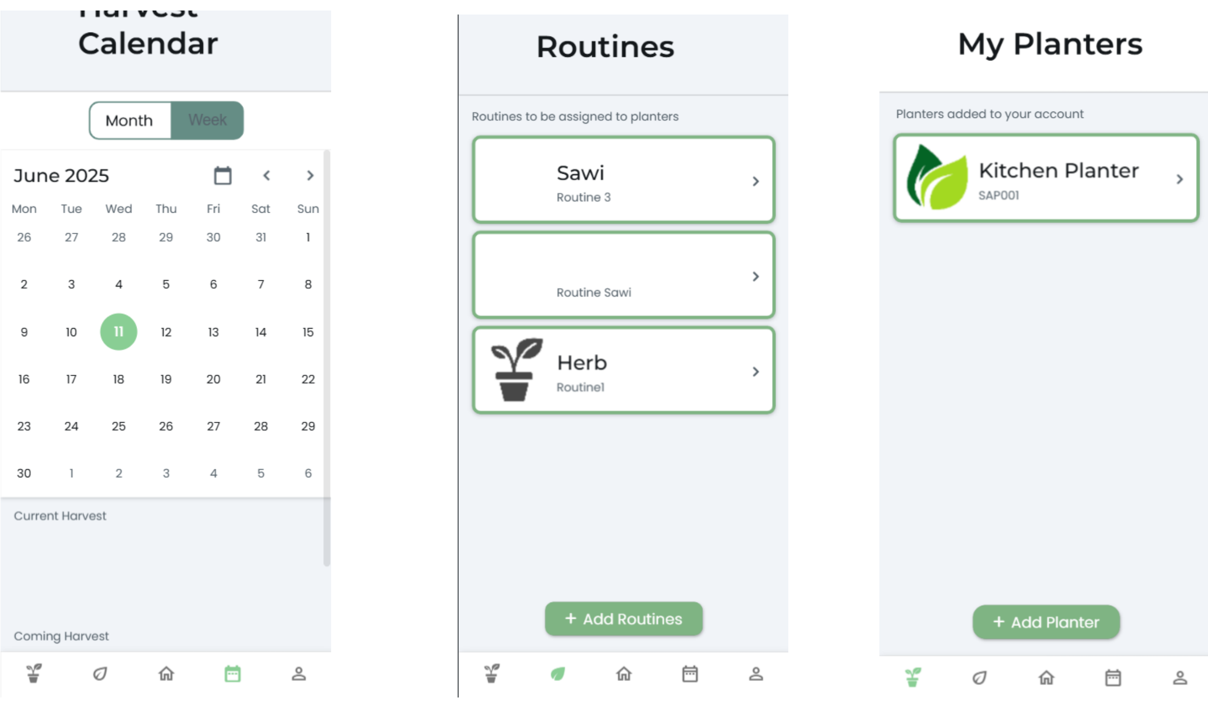
[6] E. Said Mohamed, A. A. Belal, S. Kotb Abd-Elmabod, M. A. El-Shirbeny, A. Gad, and M. B. Zahran, "Smart farming for improving agricultural management," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 24, no. 3, pp. 971–981, Dec. 2021, doi: 10.1016/J.EJRS.2021.08.007.

[7] M. Foo, E. Ganzert, T. Horita, A. Khoshnavaz, and M. Tatari, *EcoGarden: Self-contained aeroponic growth system*, 2016.

[8] M. Taylor, A. Newman, and R. Ribbons, "Exploring the rhizosphere microbiome of hydroponically grown leafy greens," *Proc. Wisconsin Space Conf.*, 2022.

[9] A. M. Davis, *Software Prototyping*, in *Advances in Computers*, vol. 40, M. C. Yovits and M. Zelkowitz, Eds. San Diego, CA, USA: Elsevier, 1995, pp. 39–63, ISSN 0065-2458, ISBN 978-0-12-012140-3, doi: 10.1016/S0065-2458(08)60544-6.

Appendix A : User Interface



Appendix A : (a) Harvest Calendar Page (b) Routine Page (c) Planter Page

Appendix B: Autoplanter Device

Appendix B: (a) Vertical tower of Autoplanter Device (b) Reservoir for Device