International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

Integrated Smart Farming Based on Internet of Things (Iot) System for Figs Cultivation

N. Mohamood, N. Zainal, A. F. Kadmin, S. F. Abd Gani, T. M. F. Tengku Wook

Abstract: ***Fig plants are gaining popularity among farmers across Malaysia, mainly influenced by the high demands for fresh fig fruits and a fairly higher market price for the fruit. Current practices in farm fields are still based on observation and scheduling approach without any quantitative approaches which provide a precise way of determining the crucial elements such as irrigation and fertilization needs. This paper explains the design and development of smart farming system with sensing technology deployment for precision agriculture and the Internet of Things (IoT) approach to get farmers connected to their farm. Raspberry Pi 3 Model B acts as a brain of the entire system, delivering its functionality in performing monitoring and controlling tasks. Database is implemented by using ThingSpeak IoT cloud platform while for mobile application, this project is using Swift 4 programming language within Xcode IDE in implementing the iOS user interface features. The evaluation and validation result shows the microcontrollers and all embedded sensors associated to them are successfully executing their tasks in performing the surrounding humidity, irrigation and fertilization control duties. The developed system also capable in tracing the surrounding temperature and humidity, soil humidity and pH, and fertilizer EC value changes. Assistance in mobile device application implementation and ThingSpeak cloud database deployment in this project also get the farmers connected to their farm. Although this project has been completed successfully, however there are several areas which can be further improved in order to make the entire system more efficient and useful to the user.***

Keywords: ***Fig, Internet of Things, Raspberry Pi 3 Model B, Swift 4.***

Revised Manuscript Received on December 30, 2019.

***** Correspondence Author

Nadzrie B Mohamood*, Faculty of Electrical and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia. Email: nadzrie@utem.edu.my

Nasharuddin B Zainal, Faculty of Engineering and Build Environment, Universiti Kebangsaan Malaysia, Bangi, Malaysia. Email: Nasharuddin.zainal@ukm.edu.my

Ahmad fauzan B Kadmin, Faculty of Electrical and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia. Email: fauzan@utem.edu.my

Shamsul Fakhar B Abd Gani, Faculty of Electrical and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia. Email: shamsulfakhar@utem.edu.my

Tg. Mohd Faisal B Tengku Wook, Faculty of Electrical and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia. Email: tgfaisal@utem.edu.my

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

I. INTRODUCTION

Farming has a pivotal role that requires continuous effort and commitment, and farmers need to be equipped with proper knowledge, skills, and most importantly patience.

A key aspect of farming is the farmer also needs to know the most suitable time to plant, the type of soil composition for specific plants to grow optimally, when and how much water needed for irrigation, how often the fertilizer needs to be applied, and how long it takes a certain plant to harvest. In addition to that, farmers also face challenges in finding suitable and cultivable land, increased need for fresh water supply to water the plants, and unpredictable factors such as climate change.

In recent years, there has been an increasing interest in improving farm productivity is crucial in increasing profitability as well as fulfilling market demands for crops. One of the main obstacles to increase farm productivity, farmers need to understand and possess the capability to forecast crop performance in wider environmental, soil, fertilization, and irrigation condition [1]. However, by adapting technology into agricultural field, all these tedious processes can be replaced by automation. With technology, farms become smarter with huge amounts of sensor data, thus minimizing cost, increasing yield and maximizing profits. Essentially, smart farming applies the use of modern Information Communication Technologies (ICT) and in particular, the Internet of Thing (IoT) into agriculture.

It is very interesting to understand how modern and smart technology plays roles in remodeling the agriculture industry. This project focuses on the development of a smart farming embedded with IoT application comprised of sensors and microcontroller integration, data logging into database and connecting farmers to the entire system by using iOS-based phone app.

II. LITERATURE REVIEW

A. Fig Tree Overview

The common fig or Ficus carica L. is a prolific fruit tree, which is commonly cultivated in mild-temperature climates. The genus Ficus, comprises of over than 800 species and one of the mulberry family, Moraceae [2]. The native place of this tree is said to be origin from eastern Mediterranean area and also to Asia Minor [3].

The climate requirement for fig cultivation is in dry or semi-dry region, which is identical to its place of origin. Fig tree can survive in the temperature range from 20 °C up to as high as 45 °C [4]. The fig tree can usually grow up to 15 to 20 feet tall, with the spreading branches and trunk in about 7 feet in diameter. The root system is typically shallow and spreading sometimes covering 50 feet of ground and some of the roots may descend to 20 feet in permeable soil.

A number of researchers have reported that surrounding humidity determined the quality of the fig's fruit. During fruit development phase, high humidity may result in fruit splitting. On the other hand, several studies have documented that the combination of humid and warm condition also may lead the fungus infections to the leaves and fruits. Infected leaves and fruits would fall off with the prolonged exposure to the disease and not well controlled.

B. Figs Cultivation in Malaysia

Nowadays, it has been established that figs cultivation is not an uncommon in Malaysia. The planters known as FigDirect Sdn Bhd, which is based in Chuping, Perlis is the first commercially planted fig farm in Malaysia and nowadays become the biggest outdoor fig farm in this country. Saf Fa Fig garden, is another commercially figs farming in Malaysia. Having two-acre farm somewhere in Kuala Pilah, Negeri Sembilan, currently equipped with 15 greenhouses homing 2,000 fig trees and saplings. There are almost 500 varieties of figs cultivated in this farm. This farm however currently focusing on cultivating and culturing the two species namely Panache France and Craotian Dalmatie. These premium species are origin from Europe.

Realizing the figs' potentials, there is also an involvement from the local universities, which interested to do further research on this plant. Saf Fa Fig Garden has signed the Memorandum of Agreement (MoA) with Universiti Kebangsaan Malaysia in 2015 through their UKM Biodiesel and Future Crops laboratory [5]. At this laboratory facility, UKM researchers are working on tissue culture and active compounds of the fig fruit and leaf. Later than that, Universiti Sains Malaysia (USM) also follows the footstep with the MoA for five years, begin in 2016 with Fig Direct. In this collaboration, USM done a further research on tissue culture technology in efficient and effective way in order to culture more quality fig saplings [6]. Universiti Teknologi Malaysia (UTM) through their Centre for Community and Industry Network (CCIN) collaborate with the Felda Sening Community Committee in developing and practicing fertigation technique in figs cultivation [7].

C. Automated and Smart System for Plantation

From years back ago, most of the plantation processes were handled manually by human starting from planting the seeds until crop harvesting. Processes including watering and fertilizing the plantations require a lots of human energy especially for large scale farming. The constraint arises here is that human may not always manage or even monitor their farms 24 hours a day.

Questions has been raised about the use of fully automated system in increasing agriculture quality. Agriculture product quality could potentially be increased and enhanced with deploying sensing technology in farming. This technology has been produced a lot of farms become more intelligent and connected thus realizing the precision agriculture and smart farming. Making farming smarter can potentially increase yields and profits, as well as to effectively and efficiently make use of precious water and fertilizer. The Internet of Things (IoT) gives new perspectives to the agriculture enabling wide range of techniques to the farmers such as precision and sustainable agriculture. IoT make farmers to get connected to their farm from anywhere and anytime. Sensors are used to monitoring the farm condition, while micro controllers are used to automate the related process. Smart phone then provides the way to get farmers updated on the ongoing conditions of their farm. In this way, IoT also capable to plays a significant role in precision farming to increase farm's productivity [8].

Smart farming has growth and been the focus of many researchers, which involving numerous studies done in this particular area. [9] proposed an Arduino based automatic plant watering system mainly focusing on controlling the irrigation system to determine the frequency and the amount of water to be supplied. [10] studied the work of rural farming which then replaces some of the traditional techniques. They deployed several external sensors such as leaf wetness, soil moisture, soil pH, and atmospheric pressure sensors. Soil moisture sensor triggers the water sprinkling during water scarcity and turns it off after adequate water is sprinkled. It has been demonstrated that soil pH reading is then sent to base station and communicated with farmer via SMS using GSM modem. [11] worked on automated irrigation system based on wireless sensor, in order to optimize water usage in agriculture. The system comprises of wireless sensor network of soil moisture sensors and temperature sensors. Information from sensor is handled using ZigBee protocol, water quantity for irrigation is then determined by the microcontroller through programming using an algorithm with predetermined threshold value. [12] proposed system using ZigBee technology where the research is on hardware and the software of the network coordinator and sensor nodes. As a result, parameter of the greenhouse environment parameter such as humidity, temperature and carbon dioxide concentration can be captured efficiently by the system. They also conclude that the system capable to deliver good network stability. Although studies have recognized smart farming system, research has yet to systematically investigate the effect of smart farming in fig plant application. This paper describes the design and implementation of smart farming in fig application by using quantitative case study and mixed methodology approach.

III. RESEARCH METHOD

Quantitative methods offer an effective way to evaluate the effectiveness of a smart farming system. Proposed smart farming system model is illustrating in Fig. 1. Basically, the design consists of three main parts which are hardware, cloud database platform and iOS apps development phase.

A. Hardware Architectural Design

The first step from this process was to develop the hardware design. Hardware design determining the capability of the whole system in providing the raw data which its interprets into useful kind of information which later acts as the main input for controlling and monitoring purpose. Several sensor nodes would be integrated into an embedded system, in order to acquire a related data regarding the surrounding temperature and humidity as well as soil humidity and pH value. The brain of this hardware architectural design led by a microprocessor, which orchestrated the main function of the whole hardware system.

Basically, the whole hardware architectural design divided into two parts. This is due to practical point of view of the real application consideration. A green house where the fig plants are cultivated is actually located apart with the certain distance to the fertilizer and water tanks. However, sensor nodes are necessary for these points in the design perspectives. So that, it comes to the proposed solution to provide the two control bases, located at fig plant location and fertilizer and water tanks location respectively. These control bases, which is microcontroller then communicated with each other in completing the tasks.

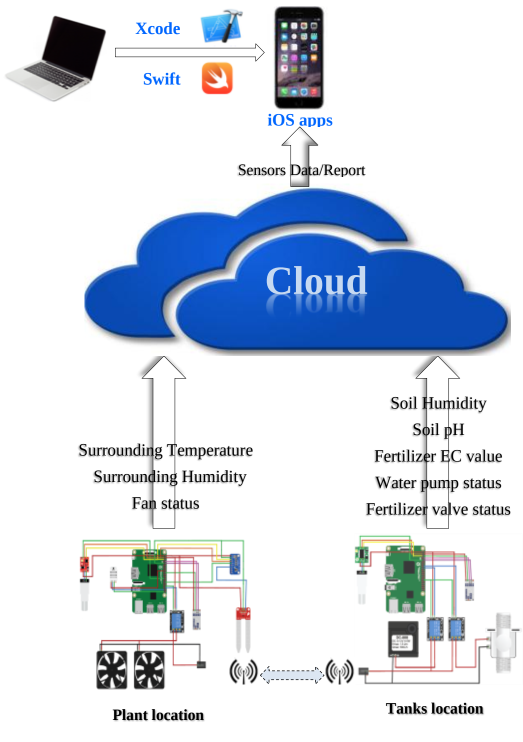

Fig. 1. Smart farming system model diagram

B. Cloud Database Implementation

Cloud database deployment in the other hand can be visualized as the intermediate medium between hardware system and the user's mobile application, or in the other words the IoT devices. Taking an advantage of the fast growing IoT technologies and the emergences of numerous IoT middleware platforms in marketplace, the database development employed this opportunity. IoT middleware platform provides a simple way of data injection from all kind of IoT devices by using a common Application Programming Interface (API) [1]. In this way, time consumption to complete this part would be reduced instead of adopting MySQL database with the PHP programming language.

This project deployed ThingSpeak IoT cloud platform as an intermediate medium in connecting sensor nodes to the iOS mobile apps. ThingSpeaks is an IoT analytics platform service which enable users to accumulate, visualize and at the same time to analyze live data streams in the cloud. For the purpose of monitoring analysis, it also offers instant visualization to the posted data by the devices to its channel.

C. Mobile Application Design

An iOS apps development has taken into consideration the user interface design and arrangement, as well as the features. Main component in succeeding this phase are the Xcode Integrated Development Environment (IDE), and Swift programming language which is compatible with iOS environment. Mobile application in this project has been integrated with a tab bar controller which is closely possible to the standard Apple User Interface. Tab bar controller is kind of interface module that grants for multiple views to be controlled and displayed within the application. In most Apple application, tab bar is used for navigation purposed and usually is placed at the bottom of active application and gives a direct link to the personal view needed for navigating within the application. Fig. 2 shows the common view of tab bar user interface in common Apple apps.

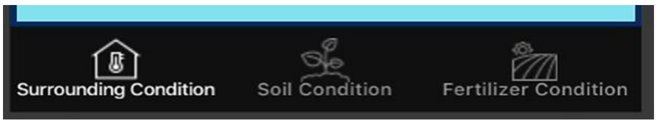

Fig. 2. Tab bar interface in iPhone app

Fig. 3 depicts a block diagram for mobile application main features and interaction flow with user. The incorporated of tab bar function permits user to select features within the feature within the application.

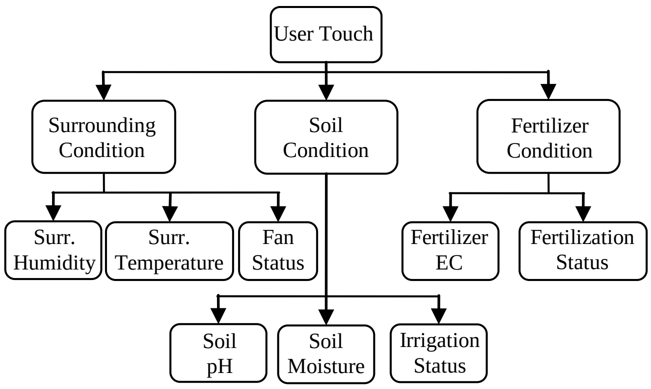

Fig. 3. Mobile apps feature block diagram

IV. RESULT AND DISCUSSION

Basically, the main evaluation and verification result cover all hardware design, cloud database implementation and finally the iOS mobile application's testing phase.

As of hardware part, the evaluation is focused on the entire system design functionality to ensure each of the component is capable in executing its own functional task. While for cloud database is evaluated based on its capability in storing all related data sending from the system and to further validate all those data. Final part would be the iOS mobile device application implementation evaluation comprise of testing phase to analyze the application's requirement verification and performance test.

A. Hardware System Validation and Verification

Hardware section discussed the result of the hardware and system design implementation. This is to conform that the hardware design is capable to deliver the stated specifications as previously discussed in methodology section. Both Raspberry Pi are evaluated based on the timing intervals which comprised of sensing time interval and data posting time intervals for related data to the cloud database. Sensing intervals which specify that all sensors should take the reading and handling the automated elements in every 4 s interval. Update interval comprise of automated system status update (fan status, water pump status and fertilizer valve status) in every 20 s interval and all sensor reading update in every 120 s.

First 200 cycles of sensing intervals for Raspberry Pi #1 which is reflected in about 15 minutes time frame is measured and analyzed. Fig. 4 shows that the minimum sensing interval is 4.05 s but the maximum sensing interval is as much as 7.60 s. The possible delay in sensing interval is potentially caused by the sensing duration itself where it will take into account the durations of DHT and soil humidity sensors input process. Another component which influences the sensing time intervals are fan status and all data update executions to ThingSpeak database which are taking place every 20 s and 120 s respectively.

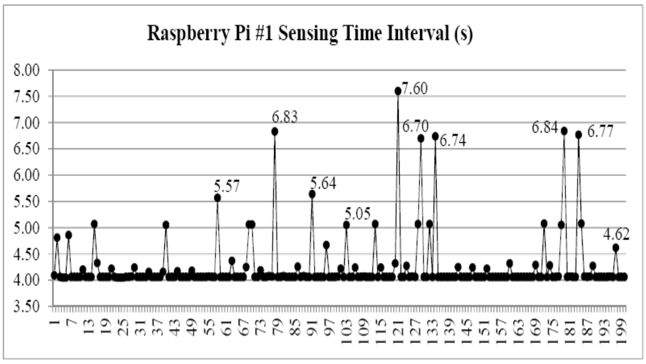

Fig. 4. Raspberry Pi #1 sensing time intervals

Fan status update time interval is evaluated as shown in Fig. 5. This evaluation comprises of the data analysis towards the first 100 cycles of fan status update intervals. It is apparent to observe that the updates are also manipulated by the higher time intervals which are longer than 40.0 s especially at points denoted by 1 and 2 respectively. The maximum fan status interval is taking place at 43.17 s while the minimum time interval is 20.02 s. The higher time interval happens when fan status is updated together with all sensors data, which is taking place after fourth successful fan status update alone. Even though this situation occurs, the average fan status update however lies within the tolerable value of 24.51 s which is not far from the predetermined 20.0 s update interval. The higher update interval also occurs once in every fifth update cycle which is considerably infrequent enough and taking into account that fan status itself is not crucial towards the entire system execution.

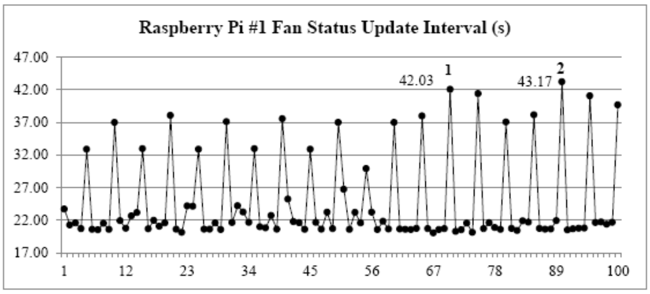

Fig. 5. Fan status update time intervals

All data update intervals for Raspberry Pi #1 is depicted as in Fig. 6. This includes surrounding humidity and temperature together with fan status data update into single channel in ThingSpeak cloud database. The highest and maximum update intervals occur at point 1 which is 127.10 s, while the second highest point is at point 2 with 126.26 s. Even though this time interval is high, it doesn't deviate much from the expected 120 s data update interval.

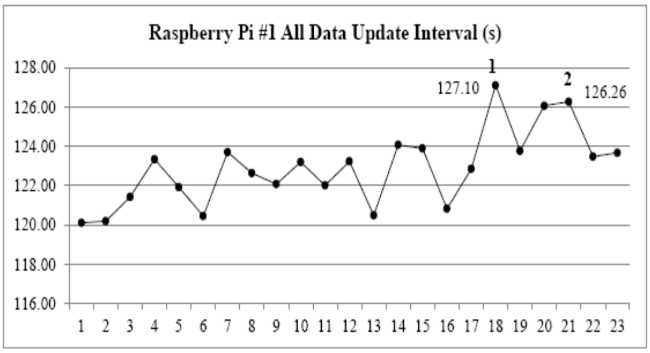

Fig. 6. Raspberry Pi #1 all data update time intervals

Fig. 7 illustrates sensing time intervals for Raspberry Pi #2. As the previous Raspberry Pi #1, this data evaluation and analysis comprises of first 200 cycles of sensing time interval. If compared to Raspberry Pi #1, it seems that Raspberry Pi #2 sensing time intervals are taking longer time during each cycle. First factor which leads to this situation is the sensing interval mechanism for the Raspberry Pi #2 itself, where its operation cycle will begin with receiving the soil humidity and pH value from Raspberry Pi #1 through its RF 433 MHz receiver. Raspberry Pi #2 response time depends on this data arrival at the beginning, then only it will proceed with the remaining functional processes associated with it. Hence, as Raspberry Pi #1 sensing time interval is delayed, so does the interval for Raspberry Pi #2.

The second factor that affects the Raspberry Pi #2 sensing time interval is pump and valve status update duration. This is due to pump and valve status are updated into two different channels, so that the doubling of status update duration for Raspberry Pi #2 is expected.

While considering the factors which lead to the higher sensing time interval within Raspberry Pi #2, however its average value is actually concentrated at 5.10 s with the standard deviation of 1.89 s. This average sensing time interval is considerably accepted and expected with the multiple channels update within Raspberry Pi #2.

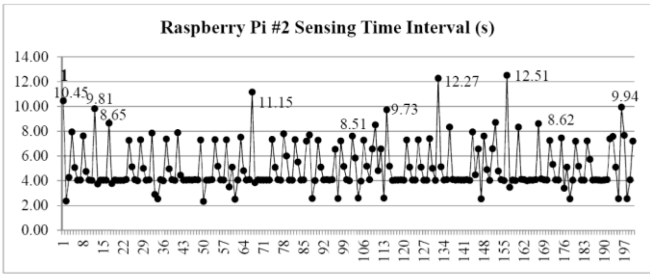

Fig. 7. Raspberry Pi #2 sensing time intervals

Pump and valve status updates in Raspberry Pi #2 indicate the irrigation and fertilization process respectively. Similar to the fan status update interval in Raspberry Pi #1, the designed algorithm for both pump and valve status updates are working in the same way as the fan status update. Status update interval is higher when it comes to the fifth cycle and this pattern is consistent throughout the entire status update operation. The average status update for both pump and valve however lies at 24.45 s with a standard deviation of 5.75 s. Irrigation and fertilization process are not time critical, hence, slightly higher status update interval for both processes are practically acceptable.

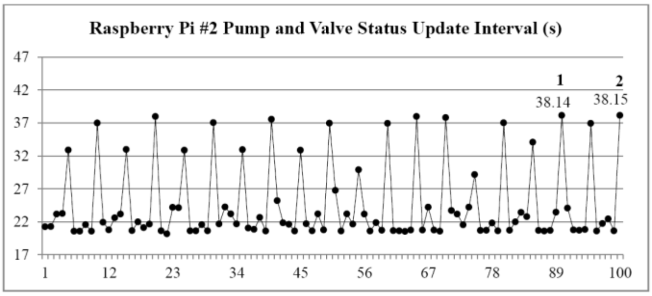

Fig. 8. Pump and valve status update time intervals

Raspberry Pi #2 is also responsible to update other sensor data such as soil humidity and pH values as well as fertilizer EC value. All these data will be updated into ThingSpeak channel once every 120 s. Fig. 9 shows the update intervals pattern of these data update during entire evaluation process. The minimum data update interval is 120.19 s, while the maximum is at 124.19 s interval. The highest and maximum update intervals occur at point 1 which is 124.03 s, while the second highest point is at point 2 with 124.19 s. Throughout the entire data, the average time taken by the Raspberry Pi #2 to update all the data is 122.46 s intervals with the data standard deviation of 1.21 s.

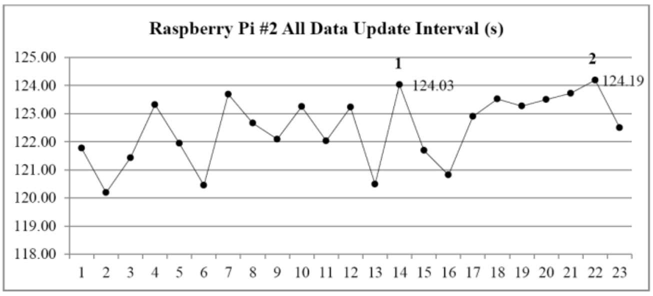

Fig. 9. Raspberry Pi #2 all data update time intervals

All operational and status update intervals for both Raspberry Pi #1 and Raspberry Pi #2 are summarized in Table- I. This concluded that the entire hardware system design capability in determining its efficiency delivery towards the functional specifications.

Table- I: Summary of hardware operational and status update intervals

Item	Category	Min (s)	Max (s)	Avg (s)	Std Dev (s)
Raspberry Pi #1	1. Sensing interval	4.05	7.60	4.25	0.55
	2. Update intervals:				
	a. Fan status	20.02	43.17	24.51	6.52
	b. Surrounding temperature and humidity data	120.11	127.10	122.90	1.91
Raspberry Pi #2	1. Sensing interval	2.33	12.51	5.10	1.89
	2. Update intervals:				
	a. Pump and valve status	20.14	38.15	24.45	5.75
	b. Soil humidity and pH and Fertilizer EC data	120.19	124.19	122.46	1.21

B. Cloud Database Evaluation

In this evaluation and validation process, analyzed data from the hardware system data logging would be compared with the data in the cloud channels data feeds. This is to ensure that all the data comprise of automated system status (fan status, water pump status and fertilizer valve status) and all sensor readings are updated successfully.

As of ThingSpeak database deployment, the expected data update comprises of fan, pump and valve status update together with surrounding humidity and temperature, soil humidity and pH, and fertilizer EC values update are successfully executed. Data verification process shows that there are no missing data during update process where all the expected data update is visible across all ThingSpeak channel fields.

C. Specification Evaluation of iOS Mobile Application

The number of testing have been carried out to verify and validate the iOS mobile application to ensure it meet the requirements that have been specified earlier and to make this mobile application features deliver its expected functionality.

Basically, specifications evaluation comprises of requirement verification and performance testing.

Requirement verification validates the iOS mobile application features with initial requirements specified during development stage. This will determine whether the application conforms to the initial requirement.

As illustrated in Table II, it can be concluded that the developed iOS mobile application fulfills its hardware and software requirements. The iOS application is successfully flashed and functioning well on an actual iPhone SE with the latest iOS version 11.1.2. iOS Simulator also reveals that this application is capable to be run on the latest iPhone 10 with the same iOS version. The simulation is illustrated in Fig. 10. The software development tools comprise of Xcode, as well as its simulator is using the latest released version, updated early November 2017. Development process is also run on a MacBook pro with macOS Sierra 10.12.6 operating system. Swift 4 is chosen as the main programming language in creating the user interface and other related components within the application.

Table- II: Hardware and software requirements result

No	Requirement	Priority	Result
1	The application should run on Apple iPhone SE or later version	Critical	Passed
2	The application should compile and run on iPhone iOS 11 or later	Critical	Passed
3	The application should built and deployed in MacOS Sierra version 10	Critical	Passed
4	The application should compile and run on iPhone Simulator 11.0	Critical	Passed
5	The application should be built using iPhone Xcode 9.1	Critical	Passed
6	The application should be written in Swift 4	Critical	Passed

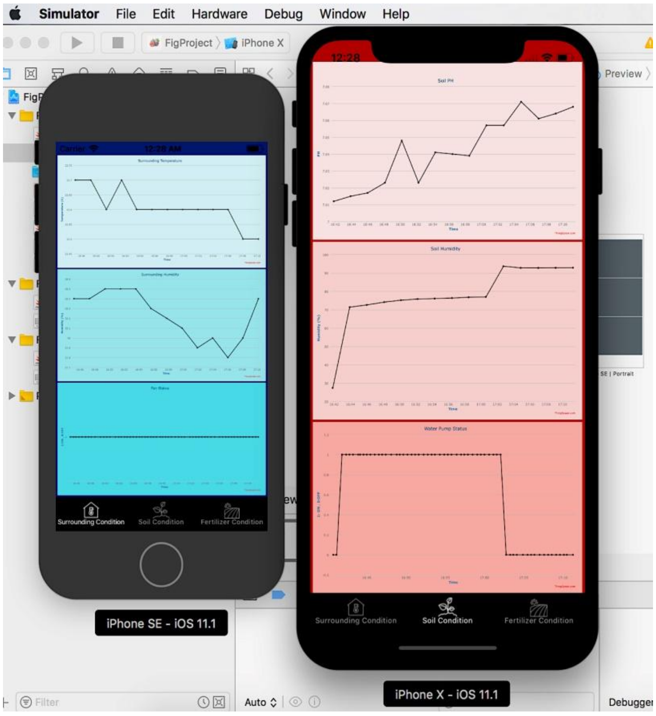

Fig. 10. iPhone SE and iPhone X in Xcode simulator 11

The most important requirement in determining the application's functionality and crucial to users are shown in Table- III. Most of the application development period is mainly focused on fulfilling all those functional requirements. ThingSpeak cloud database plays a significant role in delivering the functional capability to this mobile application. Key features embedded in ThingSpeak includes data visualization through charts which is a real time reflection of updated data and in line with the much hyped IoT.

To distinguish and for better viewing pleasure to the user, tab bar is implemented within the application to further separate between surrounding condition, soil condition and fertilizer condition related data. Fig. 11 shows the touch screen interaction and zooming capability within iPhone application. Figure also shows the tab bar user interface within the application.

Table- III: Functional requirement results

No	Requirement	Priority	Result
1	The application should display real time data for 1. Surrounding temperature and humidity 2. Soil pH and humidity 3. Fertilizer EC value 4. Surrounding humidity fan status 5. Irrigation pump status 6. Fertilizer valve status	Critical	Passed
2	The application should display historical data for 1. Surrounding temperature and humidity 2. Soil pH and humidity 3. Fertilizer EC value 4. Irrigation pump status 5. Fertilizer valve status	Significant	Passed
3	The application should display the time period for each fan operation session		Passed
4	The application should display the time period for each irrigation session	Significant	Passed
5	The application should display the time period for each fertilization session	Significant	Passed
6	The application should contain UI screens for each view: 1. Surrounding temperature 2. Surrounding humidity 3. Surrounding humidity fan status 4. Soil pH 5. Soil humidity 6. Irrigation pump status 7. Fertilizer EC 8. Fertilizer valve status	Significant	passed
7	The application should include a 'tab' view similarly as existing iPhone applications	Critical	Passed
8	The application User Interface should function entirely using the iPhone touch screen technology	Critical	Passed

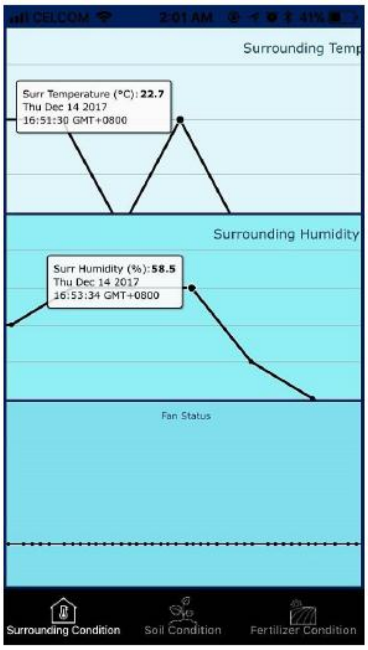

Fig. 11. iPhone application features

Fig. 11. iPhone application icon

Table- IV shows the non-functional requirement evaluation result. Non-functional requirement is not vital to the application perspective but important for the user point of view. So the non-functional requirement verification is necessary in delivering the comfort to the user when using this application. This application is simply straight forward application which only comprise of user touch to the tab which will lead the user to the related data visualization. Running test in long period of time shows no issue at all and no indication of sudden application exit tend to happen. Other necessary mobile phone operation such as answering incoming call and notification of incoming message is working as it normal operation. Thus the entire application layout conforms to the Apple standard application layout. Fig. 12 shows the icon for this application, with the application name of FIGura.

Table- IV: Non-functional requirement results

No	Requirement	Priority	Result
1	The application should require no training for a user with previous experience with iPhone	Significant	Passed
2	The application should preserve the iPhone battery as much as possible	Significant	Passed
3	The application should not contain memory leaks, and program should exits smoothly if error occurs	Critical	Passed
4	The application should handle incoming calls and text messages in an intuitive way	Significant	Passed
5	The application should reduce the user physical interaction with the UI of the mobile device	Significant	Passed
6	The application should have an application icon on the mobile device home screen	Non-vital	Passed
7	The application should conform as near as possible to the Apple standard application layout	Significant	Passed

Mobile device often has a limited resource mainly on its power and memory. The most critical component in developing an application is the memory management which an error in handling mobile device memory will lead to unnecessary performance issue. Test outcome resulting from Instrument leakage test shows no visible leakage object. If any, memory leakage will possibly cause the FIGura application to crash.

V. CONCLUSION

The project's main initial objective is to develop a smart farming system with the Internet of Things (IoT) approach which was aiming for fig cultivation as a target. Smart farming in this context is including the system's capability in delivering its functional requirements of monitoring and controlling tasks. An adaption of sensing nodes within the system is to enable the precision agriculture practice. Other than that, with the cloud database deployment in realizing the IoT features in this project also expecting the connectivity among the farmers and it farms. When looking the outcomes from this project, all mentioned elements are successfully implemented. Both Raspberry Pi #1 and Raspberry Pi #2 microcontrollers and all embedded sensors associated to them are successfully executing their tasks in performing the surrounding humidity, irrigation and fertilization control duties. As of monitoring functions, the developed system also capable in tracing the surrounding temperature and humidity, soil humidity and pH, and fertilizer EC value changes. With the help of mobile device application implementation and ThingSpeak cloud database deployment in this project also get the farmers connected to their farm.

Although this project has been completed successfully, however there are several areas which can be further improved in order to make the entire system more efficient and useful to the user.

The suggested are to improve including:

1. Sensors deployments should consist of better accuracy type of sensors and able to work in efficient way. For example, soil humidity sensor in this project is using resistive electrode which is not practical to be used in actual fields. This is because the measurement would be based on the electrical conductivity principle, which is targeting to measure the presence of water. However, at the same time fertilizer element is having the electrical conductivity which is higher with relative to the irrigation water. Thus, this avoided the soil humidity sensor to deliver its reading accuracy.
2. All related sensor should be further divided and not to concentrate on single microcontroller. The suggestion is to deploy and practice the wireless sensor network instead. Overall system performance in term of its accuracy and efficiency expected to be increased.

ACKNOWLEDGMENT

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) and Universiti Kebangsaan Malaysia (UKM) for providing the support needed to complete the work herein. We are also grateful to Sa Fa Fig Farm through for their kind and help and giving their assistance to complete this project.

REFERENCES

1. P. P. Jayaraman, A. Yavari, D. Georgakopoulos, A. Morshed, and A. Zaslavsky, "Internet of things platform for smart farming: Experiences and lessons learnt," Sensors (Switzerland), vol. 16, no. 11, pp. 1–17, 2016.
2. A. Chawla, R. Kaur, and A. K. Sharma, "Ficus carica Linn .: A Review on its Pharmacognostic , Phytochemical and," Int. J. Pharm. Phytopharm. Res., vol. 1, no. 4, pp. 215–232, 2012.
3. G. Nicotra, S. Vicentini, and A. Mazzolari, "Ficus carica Research and development of a dry extract," vol. 9, no. 3, pp. 27–30, 2010.
4. M. A. Flaishman, V. Rodov, and E. Stover, The Fig: Botany, Horticulture, and Breeding, vol. 34, no. April 2017. 2008.
5. Riceukm. 2017. Makmal Alami. http://www.ukm.my/riceukm/?page_id=53.
6. USM News Portal. 2015. USM-Fig Direct Berkolaborasi Majukan Industri Buah Tin Pertama. https://news.usm.my/index.php/berita-mutakhir/3429-usm-fig-direct-berkolaborasi-majukan-industri-buah-tin-pertama.
7. UTM CCIN. 2016. Penjanaan Sosio Ekonomi Komuniti dan embangunan Teknologi Melalui Pertanian Fertigasi Pokok Tin. http://ccin.utm.my/2016/01/21/penjanaan-sosio-ekonomi-komuniti-dan-pembangunan-teknologi-melalui-pertanian-fertigasi-pokok-tin/.
8. D. K. Sreekantha and Kavya A. M., "Agricultural Crop Monitoring using IOT- A Study," 2017 11th Int. Conf. Intell. Syst. Control (ISCO), Coimbatore, 2017, pp. 134–139, 2017.
9. A. N. and S. D. A. Patil, M. Beldar, "Smart Farming Using Arduino and Data Farming," pp. 1913–1917, 2016.
10. S. Vijayakumar and J. N. Rosario, "Preliminary design for crop monitoring involving water and fertilizer conservation using wireless sensor networks," 2011 IEEE 3rd Int. Conf. Commun. Softw. Networks, ICCSN 2011, pp. 662–666, 2011.
11. G. Nisha and J. Megala, "Wireless sensor network based automated irrigation and crop field monitoring system," 2014.
12. L. Li, S. Yang, L. Wang, and X. Gao, "The greenhouse environment monitoring system based on wireless sensor network technology," 2011 IEEE Int. Conf. Cyber Technol. Autom. Control. Intell. Syst., pp. 265–268, 2011.

AUTHORS PROFILE

Nadzrie Mohamood graduated with a Bachelor Degree in Electronic Engineering from Universiti Sains Malaysia (USM), in 2005, and finished his Master of Engineering (Communication & Computer) from Universiti Kebangsaan Malaysia (UKM), in 2018. He is currently a Senior Teaching Engineer attached to Electronic and Computer Engineering Technology Department, Faculty of Electrical and Electronic Engineering Technology, UTeM. Previously, he worked as a Product/Test Engineer in Texas Instrument (M) Sdn. Bhd., Freescale Semiconductor (M) Sdn. Bhd. and STATChipPAC (M) Sdn. Bhd, with worth of 5 years of industrial experiences prior to joining his current role as an academician. His main research interests include computer networking, image and video processing and programming.

Nasharuddin Zainal received his BEng in Computer Science from Tokyo Institute of Technology (Japan), in 1998, MEng in Communication and Computer Engineering from Universiti Kebangsaan Malaysia (UKM), Malaysia, in 2003, and DrEng degree from Tokyo Institute of Technology (Japan) in 2010. He is currently an Associate Professor at the Centre of Advanced Electronic and Communication Engineering, Universiti Kebangsaan Malaysia, Bangi, Malaysia. His main research interests include computer engineering, image and video processing and pattern recognition. He published several technical and engineering paperwork in various high index engineering journals worldwide. He also currently serving as professional engineer under Board of Engineer Malaysia.

Currently attached with UTeM as a researcher, Ahmad Fauzan B. Kadmin CEng. P.Tech. has over 14 years of experience in electronic & computer engineering field with technical expert in R&D engineering, computer vision & medical electronics. He Graduated with a Bachelor Degree in Electronics Engineering from Universiti Sains Malaysia and Master Degree in Computer & Communication Engineering from Universiti Kebangsaan Malaysia. Previously, he worked as project engineer with Megasteel Sdn. Bhd., Samsung SDI(M) Sdn. Bhd. as research and development engineer and later as research officer in Agensi Angkasa Negara. He published several technical and engineering paperwork in image processing and medical electronics.

Shamsul Fakhar bin Abd Gani graduated from Universiti Malaysia Perlis (UniMAP) in Bachelor of Engineering (Computer Engineering) with honours in 2006 and later received his Master's degree in Internet & Web Computing in 2015 from Royal Melbourne Institute of Technology (RMIT) Australia. He started his career as an R&D electronic engineer specializing in software design for meter cluster development in Siemens VDO Automotive Penang (later known as Continental Automotive Malaysia). Before leaving Continental, Shamsul plays an active role in Mitsubishi Fuso projects as a software developer and also software project manager. Shamsul is now a lecturer in electronic and computer engineering technology department of FTK UTeM.

Tengku Mohd Faisal bin Tengku Wook graduated from Universiti Sains Malaysia (USM), in 2000 and finished his Master of Business Administration (Advanced Operations Management) from Universiti Teknikal Malaysia Melaka (UTeM), in 2011. He is currently a senior Teaching Engineer attached to Electronic and Computer Engineering Technology Department, Faculty of Electrical and Electronic Engineering Technology, UTeM. He started his career as Production Engineer in Soshin Electronics (M) Sdn. Bhd. During in Soshin, Tengku plays an active role in Yield improvement projects and transfer new product from Japan. After 7 years he leave Soshin for Konica Minolta. In Konica Minolta he is person in charge for Western Digital project. After 5 years he decide to join UTeM as an academician.