Development of Large-Scale Precision Farming Monitoring System based on REST API and SignalR

2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE) | 979-8-3503-2095-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCoSITE57641.2023.10127746

I Made Adhiarta Wikantyasa
Institut Teknologi Sepuluh Nopember
Informatic Engineering
Surabaya,Indonesia
6025221008@mhs.its.ac.id

Riyanarto Sarno
Institut Teknologi Sepuluh Nopember
Informatic Engineering
Surabaya,Indonesia
riyanarto@if.its.ac.id

Sulaiman Triajo
Institut Teknologi Sepuluh Nopember
Informatic Engineering
Surabaya,Indonesia
6025211006@mhs.its.ac.id

Abstract— **Farming is one of the primary resources to provide human food therefore, it takes a large amount of cultivated land to meet food needs. Currently, technologies such as IoT have significant role in advancing agriculture in this digitalize era by improving conventional agriculture methods by not only making it optimal but also making it cost efficient for farmer to produce quality of crops. The aims of this paper to develop system in the form of REST API service by using .net 6 as framework to efficiently manage the growth of IoT devices from the ground up in medium scale to large scale cultivated land with easy to scale up based on the SignalR library and crops health monitoring. The system will assist institute farmer and technician by visualizing data from each deployed IoT device (Temperature, humidity, soil moisture, Ph) from each section of cultivated land and management of each new IoT device such as sensor, actuator and microcontroller that will be deployed and by using SignalR library as an interconnection between device to API. To ensure the quality of the system author measured using the JMeter application by conducting simulation with high peak of user access and transmitted data of IoT device to give a validation of load testing that the system will run smoothly without any problem. The research concludes based on the testing that been done has successfully delivered and shown no latency or error with performance result on each test 21ms functional and 0.50ms on SignalR testing.**

Keywords—Internet of Thing, REST API, Precision Farming, SignalR, System Development

I. Introduction

In the last year, the increase in resident global growth is accompanied by an increase in demand for food production with a significant decrease in farmers throughout the time[1]which lead to a much harsher way for the farmer to cultivate their land to monitor the health of their crops, monitor health of the soil and ensure the safety of their plant from pest [2]. Many challenges impede crop production, leading to decreased crop productivity. Such as soil salinity in arid conditions, in addition, the climate also plays a big role in how quantity and quality of crops and may lead to an increase in soil sensitivity to desertification[3].

Internet of things (IoT) technology has been a popular solution in the farming environment, for example in monitoring the quality of beef being produced in the research that took place in the year 2019 gives a result implementation of its proposed method by combining machine learning techniques and IoT device to filter electronic nose signal to produce a much better quality of beef in agriculture environment[4], in a recent survey that conducts in 2021 provide an insight of solution to enhance crop production and cultivation practices such as crop monitoring, nutrition and management of resource planning from farmers, transporter, distribution and end-consumer that can be made possible by the implementation IoT technology in the farming environment[5].

Commonly practice of IoT technology in farming is usually used in medium-scale environments with land ranging from 5 to 100 hectares. Generally, it consists of a small percentage of sensors that monitor the crops on its land to give a response of analytic result , but as the cultivated land transition to a much large environment and more IoT devices are deployed, it will lead to a harsher way to manage on the new deployed IoT device that can be crucial on knowing the information on each section deployed device resides and differentiate the transmitted data from each deployed IoT device.

Research in 2019 develop a precision agriculture system to optimally water crops based on the humidity of soil and temperature and provide data management via smartphone and web applications [6]. Research conducted in 2019 proposed a design of an application monitoring system for modern eco-agriculture based on cloud computing where it shows a result of a modern agricultural IoT monitoring system, the intelligent gateway of the IoT, MQTT message service, cloud-based big data analysis, and data mining projects[7]. Research in 2020 presents a design of a smart agriculture system that combines with an irrigation system to cope with unpredictable weather most feature that the application provider is to detect temperature and humidity in the surroundings and moisture level where the output is displayed via smartphone and computer [8]. Recent research in 2022 develop a cost-effective system for cultivating land in India to give a simple to utilize innovation that would help in the quality of the crops by temperature, dampness, moistness, and pH sensor at the selective areas to check harvest condition[9].

Most of the previous research that has been provided focus solely on one scope of owned land based on the chosen case study by the author and no feature to provide scalability or management of new IoT device that will be deployed as the cultivate owned land growth and some of the research focus mainly on monitoring crops health based on the humidity or temperature of the soil and environment but no feature to maintain the crops that been infected by the disease. With these problems on hand, the research will focus on creating a precision farming based on REST API backend server that aim to provide a solution on the growth of IoT device from medium scale farming environment to larger scale in term of management connected IoT device and differentiate the transmitted data from IoT to REST API server with the help of SignalR library as a bridge to process IoT device and monitoring system based on scheduler that provide the needs for user to monitor and manage the disease that infected the crops in each section of the owned land, this feature can be used as a single use scheduler or repeated scheduler which will notify user on daily basis based on the need of the user.

II. METHODOLOGY

The research in developing the system used the Agile Software Development Life Cycle which is method with a short timeframe based on repetitive work, where the rules and solutions that have been agreed upon by each team member are carried out in a structured and organized collaboration[10]. From the agile approach model scrum is being used in the process of making the system, in general scrum is considered able to produce quality software in accordance with the requirement of user need, can be used in large or small scale projects, and is easy to adopt changes if there some phase that not meet the requirement, then in the next phase a system can be developed according to user evaluation [11].

A. Product Backlog

The product backlog consisted of a list of priority work for the development team (developers) derived from the roadmap and the requirements needed to build the application usually carried out in the early phase of the development [12]. The tasks in this phase can be in form of functional needs or other related requirements to make an ideal system based on the scrum team vision [13].

1) Case Study Indentification

Case study identification aims to identify the general concept of the case that the author brings out based on the deployment of IoT devices generally in each institute cultivate land respectively that can be seen at Fig. 1 consisting of many sensors that are connected directly to a microcontroller to transmit data through WebSocket generally, and here its shown that each section of the cultivated land will have one Mini PC that is Raspberry PI that acts as a central to collect data from the microcontroller.

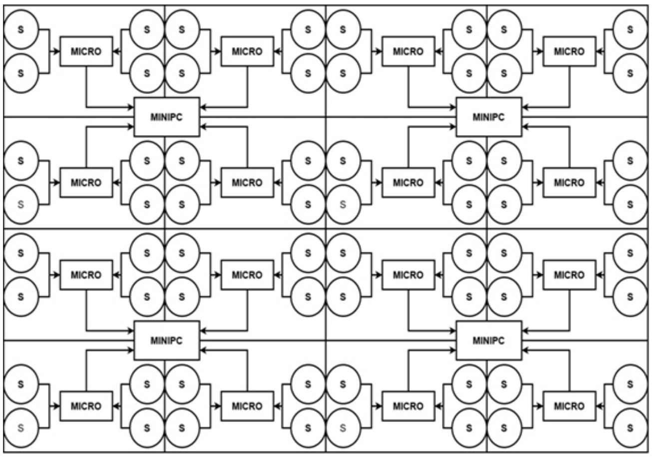

Fig. 1. Case Study

2) Use Case Identification

At this phase, the identification of the actor is a depiction process that is carried out to show the relationship between the actor and the designed system. The resulting representation of the schema is made simply and aims to make it easier for the user to read the information provided[14].The use case identification of the proposed model consists of 3 type of actor that is super admin which is the owner of the system that have higher privilege to access the system while admin or an institute technician can access all feature but have restriction to add super admin and users as an regular office worker have role to only read / view data in the field. Use Case List can be seen in TABLE 1.

TABLE 1. Use Case List

Actor	Privilege	Feature
Super Admin	Create Update Delete Read	Plant MiniPc Schedule Plant Health Micro UserAdmin Region Actuator Sensor Land
Admin	Create Update Delete Read	Plant MiniPc Schedule Plant Health Micro Institute Admin Region Actuator Sensor Land
User	Read	Plant MiniPc Schedule Plant Health Micro Region Actuator Sensor Land

3) Identification of System Requirement

Identification of the requirement system stage is carried out to analyses list of need that will be used in the system to provide the best of its utility and benefits [15]. If there are not transparent or the analysis is not appropriately done, it might lead to a failure of the project no matter how good the concept and design are [16]. An illustration of the requirements of the system can be seen in TABLE 2 which consists of 15 backlog items that have been defined based on the needs in the system that are being made by the team.

TABLE 2. Back-End Sprint Planning

No	Backlog Item
1	Authentication User
2	User Role
3	Create Update Delete Read API Plant
4	Create Update Delete Read API Plant Parameter
5	Create Update Delete Read API Land
6	Create Update Delete Read API Region
7	Create Update Delete Read API Mini PC
8	Create Update Delete Read API Microcontroller
9	Create Update Delete Read API Sensor
10	Create Update Delete Read API Actuator
11	Create Update Delete Read Multiple Image
12	API IoT Device Hub
13	Create Read Data Sensor
14	Create Update Delete Read API Disease Monitor
15	Create Update Delete Read API Disease Health Scheduler

B. Sprint

Sprint is the next stage after defining the product backlog. Sprints can be considered the core of Scrum life cycle. Usually, last up to 1 month or less that is filled with work products that are additional, functional and released to end consumer [17]. The new sprint will be executed after the previous sprint is finished. This phase consists of two activity that will be carried out on each sprint as follows.

1) Sprint Planning

Sprint planning is the first stage of every sprint life cycle that is executed after the product backlog during this stage team will conduct a meeting at every sprint that aims to discuss the goal and mission of each feature [18]. Sprint planning of the research can be seen in form of a list in

TABLE 3 which total to 76 days or around 2 and a half months.

TABLE 3. Back-End Sprint Planning

No	Feature	Estimated Day
1	Authentication User	5
2	User Role	5
3	Create Update Delete Read API Plant	5
4	Create Update Delete Read API Plant Parameter	5
5	Create Update Delete Read API Land	5
6	Create Update Delete Read API Region	5
7	Create Update Delete Read API Mini PC	5
8	Create Update Delete Read API Microcontroller	5
9	Create Update Delete Read API Sensor	5
10	Create Update Delete Read API Actuator	5
11	Create Update Delete Read Multiple Image	5
12	API IoT Device Hub	6
13	Create Read Data Sensor	6
14	Create Update Delete Read API Disease Monitor	5
15	Create Update Delete Read API Disease Health Scheduler	5

2) Sprint Backlog

The sprint backlog is an artifact that is executed after sprint planning is a list of the requirements that is a priority with a starting time and end time that has been determined by the scrum team [19]. At this stage, it shows complete features that have been defined based on the sprint planning. The development sprint is divided into 6 sprints which can be seen at TABLE 4.

The first sprint consists of user configuration such as adding or updating users and setting up role configuration for each privileges access that will be given to a user in instituted that will be created later on and in this segment, also the creation of a database of each function related to the research.

The second sprint illustrated the creation of the API function of each feature that has been defined before this sprint consisting of the creation of plant API and parameters to know the min or max value of the soil and humidity that will result to turn on the actuator to start watering. Lastly, it creates an API to manage Land owned by the institute.

The third sprint illustrated the API that in the span of the date given has been developed consisted of creation API to manage each section of cultivated land where each land has many regions/sections and the other half of the sprint more to manage Mini PC or Raspberry from this case on each region and same as API for microcontroller management.

The fourth sprint illustrated the creation of management for the sensor and actuator also this sprint, the creation of image API will be made to manage a single and multiple bytes of an image that will be used later on.

The fifth sprint illustrated the phase of creating the Hub using SignalR which from this phase it takes quite a bit of time because it not only involves the backend system, it

TABLE 4. Sprint Backlog Artifact

Backlog Id	Backlog Name	Estimate time	Start	End	Sprint
1	JWT Configuration	5	12-Sep	16-Sep	First Sprint
2	User Manager Configuration	5	12-Sep	16-Sep	First Sprint
3	CRUD (Create, Update, Delete) User Account	5	12-Sep	16-Sep	First Sprint
5	Current User Configuration	5	19-Sep	23-Sep	First Sprint
6	User Role Configuration	5	19-Sep	23-Sep	First Sprint
7	CRUD (Create, Update, Delete) Role API	5	19-Sep	23-Sep	First Sprint
8	Create Database	5	26-Sep	30-Sep	First Sprint
9	CRUD (Create, Update, Delete) Plant API	5	03-Oct	07-Oct	Second Sprint
10	CRUD (Create, Update, Delete) Plant Parameter API	5	03-Oct	07-Oct	Second Sprint
11	CRUD (Create, Update, Delete) Land API	5	10-Oct	14-Oct	Second Sprint
12	CRUD (Create, Update, Delete) Region API	5	17-Oct	21-Oct	Third Sprint
13	CRUD (Create, Update, Delete) Mini Pc API	5	24-Oct	28-Oct	Third Sprint
14	CRUD (Create, Update, Delete) Micro API	5	24-Oct	28-Oct	Third Sprint
15	CRUD (Create, Update, Delete) Sensor API	5	31-Oct	04-Nov	Fourth Sprint
16	CRUD (Create, Update, Delete) Actuator API	5	31-Oct	04-Nov	Fourth Sprint
15	CRUD (Create, Update, Delete) Single and Multiple Image API	5	07-Nov	11-Nov	Fourth Sprint
16	Setting Up SignalR Backend Hub	6	14-Nov	19-Nov	Fifth Sprint
17	Configuration Event in Raspberry	6	14-Nov	19-Nov	Fifth Sprint
18	Initialize Device Backend	6	14-Nov	19-Nov	Fifth Sprint
19	CRUD (Create, Update, Delete) Data Sensor API	6	21-Nov	25-Nov	Fifth Sprint
20	CRUD (Create, Update, Delete) Disease Monitor API	5	21-Nov	25-Nov	Sixth Sprint
21	CRUD (Create, Update, Delete) Disease Scheduler API	5	21-Nov	25-Nov	Sixth Sprint

involves setting up the event of the IoT device as well in this case the Raspberry device. creation of the data function will be made in the given sprint which will be used to save data that has been transmitted from the sensor to its Mini PC to the backend.

The Sixth sprint illustrated the phase of creating the scheduler and its detailed task on detected disease API these two functions aim to manage the task on each detected disease in the disease monitor API and transmitted into the disease scheduler to give information about the observation that been made by the user that accept the task.

C. Daily Scrum

The daily scrum in scrum life cycle is a stage that executed on every finished sprint, it consists of meeting activities online through a video conferencing platform such as zoom or offline meetings based on the agreement of the scrum team [20], this stage aims to analyses and give a progress report based on the feature that been completed.

D. Sprint Planning

The sprint planning in scrum life cycle is a stage at the end of each sprint that aim to evaluate features that have been defined and already met the requirement that the ideal team vision based on the evaluation result at this stage scrum team will also forecast the next work based on the product backlog and sprint backlog that have been listed [21].

III. PROPOSED MODEL

The proposed model aims to give an overview of how the system will operate based on the case study defined in Fig. 1. The proposed design is combined with three-layer that is the user layer, the azure layer/cloud layer, and the IoT device layer each of these layer interconnects with each other to transmit its data.

A. Design of Large-Scale Precision Farming System

The general concept of large-scale farming environment is a type of environment where farmer have multiple of cultivated land which usually divided to more than one section in each land depends on the needed of institute in aspect to differentiate type of crop to increase the productivity and quality in contrast of a conventional farming environment that is medium scale farming environment where it only needs to manage one of its cultivated lands. Based on the defined concept of large-scale farming environment proposed model author provide is combine of 3 type of layer which is user layer consist of hardware that connected, azure layer / backend layer where it will process the request and IoT device layer consist of cultivated land institute own where each sensor connected to microcontroller and mini pc as its central to transfer the data to the backend. The design of the proposed system can be seen on Fig. 2.

B. SignalR and JMeter Application

SignalR is an open-source library which used to make it easier to process real-time web function in the application they create such as ability to create server code to send (push) content or data client who already connected to the server, so that the server does not have to wait for the client to request new data. The role of SignalR in this system mostly act as a bridge or signaling server to transmit IoT data and receive event based on the request.

JMeter is an open-source application mostly used to load testing functional behavior and measuring performance of application. JMeter works by simulating parameter based on the inputted group user that will send request to a server locally or publicly the returning statistics of result based on the testing that been done. The role of JMeter in the research act as a validator that conclude that the system can run smoothly with high peak user or IoT device accessing the API with no error or latency.

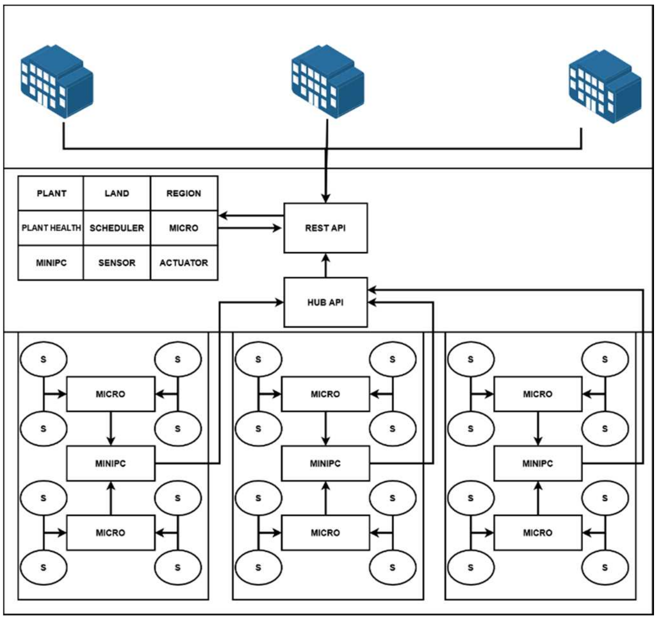

Fig. 2. Proposed System

C. Database Design

This system that has been proposed is designed by implementing the MySQL relation database as its database management. The design of the relational database consists of 25 tables each having a role as support for each functional component in the application TABLE 5 illustrated a description of 12 main tables in the system that acts as a parent / central that are connected to each related relation of 25 tables that have been made to store information based on created admin on each institute that uses the service.

TABLE 5. Database Dictionary

No	Table	Description
1	User	institute user account information and role
2	Plant	Plant information
3	Parameter	Parameter of sensor data in plant based on its sensor name.
4	Land	Land information
5	Region	Region Information
6	Mini PC	Mini PC that has been or is about to be deployed in cultivating the land
7	Microcontroller	The microcontroller that has been or is about to be deployed in cultivating land
8	Actuator	The actuator that has been / about to be deployed in cultivating land
10	Disease_Monitor	Information on detected diseases in cultivated land
11	Longitude_Latitude Disease	Pinpoint of disease in cultivating land
12	Disease_Scheduler	Information on disease tasks for institute workers to maintain crops that have been infected

D. Implementation and Performance Analysis

Implementation of the system will be carried out two types of tests, application trial and IoT Device by conducting testing on the REST API that has been made to determine the feasibility of the function made using Postman on each general method that is mostly used for frontend request and IoT device SignalR testing to determine the feasibility of process data to the backend based on the IoT device that has been deployed.

Performance tests are carried out after implementation phase by simulating many users accessing the services simultaneously and finding the maximum load that can be handled by the application system based on its response time which is a measurement that can be felt the first time by the customer or client [21]. The tests that concluded divide into 2 types of experiment performance test based on response time of functional system and SignalR IoT Device.

IV. Result and Discussion

The tests carried out in this study include 3 parts testing from the postman application by sending a request to main respective feature in the REST API system that has been made, implementation of IoT device to the backend server and lastly the performance analyst simulation that conducted to validate the performance of the system based on the process time of request that been send in the simulation.

A. Postman

This section aimed to give the validate feasibility of each functional request from the REST API that has been developed that is regularly used by the front-end system. The author used Postman as a testing application that provides tools for testing a function through its HTTP request and reading their responses. The test result of the research shown starting from TABLE 6 provide a result of successful response of the first observation before maintain where it provides the coordinate in which the disease being detected and image of the infected crops. TABLE 7 provide a result of 2 type scheduler that been generated based on the tag value it uses (Single Use or Repeated) which will generate response on TABLE 8 that show repeated scheduler generate 3 values based on its start and end date and lastly TABLE 9 is response based on the ongoing scheduler where user give a report of it maintain status.

TABLE 6. Database Response Detected Disease

DiseaseId	RegionId	MonitorStatusId	CreatedById
1	1	1	2

TABLE 7. Database Response Scheduler

Id	1	2
Start	2022-01-01	2022-01-01
End	2022-01-01	2022-01-29
Note	No Loop	Loop
Tag	1	2
LandId	1	1
DiseaseMId	1	1

TABLE 8. Database Response Schedule Occurrences

Date	ScheduleId
2022-01-01	1
2022-01-01	2
2022-01-15	2
2022-01-29	2

TABLE 9. Database Response Schedule Log

TimeStamp	Description	ScheduleId
2022-01-13 14:02:42	Monitor Crops Disease	2

B. IoT Device Testing SignalR

The section aimed to validate the designed system based on the experiment that will be conducted on real-time scenarios of an IoT device that has been deployed. The IoT device being used in the experiment consists of 4 sensors that are Ph sensor, Soil Moisture, Air moisture, and soil temperature which are directly connected to Arduino UNO it will forward the sensor data to its respective Raspberry Pi which will interconnect via SignalR the implementation can be seen in Fig. 3 that took place on 26 November at 12:30:20 and the experiment that was conducted that took 30 minutes to collect data from its sensor and from TABLE 10 illustrated the result of the response from a database. The list consists of 4 various values based on the connected sensors which show each transmitted value has a cooldown time of 5 minutes.

C. Performance Analysis

The section concludes the performance analyses of each functional component (feature) by conducting simulation based on application of JMeter to test the feasibility of performance proposed model with a parameter (user = 50, ramp-up=1, loop count infinite, duration(second) = 1200(20 minute)). The summary report can be seen start from TABLE 11 that conclude from the simulation that it been done from 2:04:40 to 2:24:10 the response time grown sharply most out of time with average of below 8ms and hitting highest approximately 12ms with throughput total of 68.1/sec where each feature value of 4.7/sec to 7.7/sec with no error in simulation that conclude that the test successfully sending the majority of messages to the server and TABLE 12 provide information of report based on the response time of SignalR Testing that conclude from the simulation that it been done from 14:21:40 to 14:41:10 the response time of SignalR hub function at start of initialize the request from JMeter WebSocket shown an increase of connectivity from 12ms and proceed steadily at 0.50ms with throughput of 91.0/sec the cause of the high percentage of connectivity happened only span of a time cause by the JMeter first initialize the request, overall the simulation has provide a good result throughout the testing with no latency and error on each packet that been transmitted that conclude the feasibility of the SignalR hub function.

Fig. 3. Deployed IoT Device

TABLE 10. Result of Response Sensor Data from Database

Microcontroller	Sensor	Value	Created At
ESP 1	Air Temperature	0.1	2022-11-26 12:35:20
ESP 1	Air Moisture	0.2	2022-11-26 12:35:20
ESP 1	Soil Temperature	0.2	2022-11-26 12:35:20
ESP 1	Ph	0.3	2022-11-26 12:35:20
ESP 1	Soil Moisture	0.3	2022-11-26 12:35:20

TABLE 11. Summary Report of Functional Component

Average	Deviation	Through put	Error	Byte Send	Byte Receive
702	1780.11	68.1/sec	0.0%	1374.23	1595.40

TABLE 12. Summary Report of SignalR function

Average	Deviation	Through Put	Error	Byte Send	Byte Receive
399	295.64	91.0/sec	0.0%	8.17	49.30

V. CONCLUSION

Based on the validation result by conducting the implementation and the performance testing of the proposed system conclude that the proposed system has successfully delivered the needs and requirement based on the case study that the author vision with performance testing that shown no error with connectivity signal stable at 21ms on each functional component and signalR testing with average of 0.50ms.

For the next research it can be in form of stress testing or load balancer with much higher percentage of high peak user that access the system with much longer period of time.

ACKNOWLEDGMENT

This research was funded by Lembaga Pengelola Dana Pendidikan (LPDP) under Riset Inovatif-Produktif (RISPRO) Program; the Indonesian Ministry of Education and Culture under Penelitian Terapan Unggulan Perguruan Tinggi (PT UPT) Program; and Institut Teknologi Sepuluh Nopember (ITS) under project scheme of the Publication Writing and IPR Incentive Program (PPHKI).

REFERENCES

[1] A. A. R. Madushanki, M. N, W. A., and A. Syed, "Adoption of the Internet of Things (IoT) in Agriculture and Smart Farming towards Urban Greening: A Review," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, 2019, doi: 10.14569/IJACSA.2019.0100402.

[2] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture," *Comput Electron Agric*, vol. 157, pp. 218–231, Feb. 2019, doi: 10.1016/j.compag.2018.12.039.

[3] U. Shafi, R. Mumtaz, J. García-Nieto, S. A. Hassan, S. A. R. Zaidi, and N. Iqbal, "Precision Agriculture Techniques and Practices: From Considerations to Applications," *Sensors*, vol. 19, no. 17, p. 3796, Sep. 2019, doi: 10.3390/s19173796.

[4] D. R. Wijaya, R. Sarno, and E. Zulaika, "Noise filtering framework for electronic nose signals: An application for beef quality monitoring," *Comput Electron Agric*, vol. 157, pp. 305–321, Feb. 2019, doi: 10.1016/j.compag.2019.01.001.

[5] M. Raj, S. Gupta, V. Chamola, A. Elhence, T. Garg, M. Atiquzzaman, and D. Niyato, "A survey on the role of Internet of Things for adopting and promoting Agriculture 4.0," *Journal of Network and Computer Applications*, vol. 187, p. 103107, Aug. 2021, doi: 10.1016/j.jnca.2021.103107.

[6] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, "IoT and agriculture data analysis for smart farm," *Comput Electron Agric*, vol. 156, pp. 467–474, Jan. 2019, doi: 10.1016/j.compag.2018.12.011.

[7] S. Liu, L. Guo, H. Webb, X. Ya, and X. Chang, "Internet of Things Monitoring System of Modern Eco-Agriculture Based on Cloud Computing," *IEEE Access*, vol. 7, pp. 37050–37058, 2019, doi: 10.1109/ACCESS.2019.2903720.

[8] Z. Muhammad, M. A. A. M. Hafez, N. A. M. Leh, Z. M. Yusoff, and S. A. Hamid, "Smart Agriculture Using Internet of Things with Raspberry Pi," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Aug. 2020, pp. 85–90. doi: 10.1109/ICCSCE50387.2020.9204927.

[9] V. Shankar, S. Kandagatla, K. Dorthi, V. C. S. Rao, P. Ankam, and N. Swathi, "Smart farming using IOT," 2022, p. 030074. doi: 10.1063/5.0082563.

[10] A. H. M. Shani, R. Sarno, K. R. Sungkono, and C. S. Wahyuni, "Time Performance Evaluation of Agile Software Development," in *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2019, pp. 202–207. doi: 10.1109/ISEMANTIC.2019.8884304.

[11] H. R. Herdika and E. K. Budiardjo, "Variability and Commonality Requirement Specification on Agile Software Development: Scrum, XP, Lean, and Kanban," in *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, Sep. 2020, pp. 323–329. doi: 10.1109/IC2IE50715.2020.9274564.

[12] R. T. Prasetio, Y. Ramdhani, and D. P. Alamsyah, "Scrum Method in Help-desk Ticketing and Project Management System," in *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, Oct. 2021, pp. 1–6. doi: 10.1109/ICORIS52787.2021.9649626.

[13] T. Sedano, P. Ralph, and C. Peraire, "The Product Backlog," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, May 2019, pp. 200–211. doi: 10.1109/ICSE.2019.00036.

[14] P.-S. Chen, G. Y.-H. Chen, S.-F. Lien, and W.-T. Huang, "Using Scrum and unified modelling language to analyze and design an automatic course scheduling system," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 6, pp. 534–543, Aug. 2019, doi: 10.1080/02533839.2019.1613930.

[15] E. C. Arango and O. L. Loaiza, "SCRUM Framework Extended with Clean Architecture Practices for Software Maintainability," 2021, pp. 667–681. doi: 10.1007/978-3-030-77442-4_56.

[16] R. Setiawan, A. D. Supriatna, S. Hudawiguna, and F. F. Roji, "Electronic culinary reservations based on Android with the Scrum methodology and Firebase database," *IOP Conf Ser Mater Sci Eng*, vol. 1098, no. 5, p. 052091, Mar. 2021, doi: 10.1088/1757-899X/1098/5/052091.

[17] A. Ramadhan, M. Lubis, W. Puspitasari, and A. R. Lubis, "Development of Web Stock Opname Application with SAP Business One Using Scrum Method," in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, Nov. 2019, pp. 1–5. doi: 10.1109/ICoSNIKOM48755.2019.9111526.

[18] F. F. Kurniawan, F. R. Shidiq, and E. Sutoyo, "WeCare Project: Development of Web-based Platform for Online Psychological Consultation using Scrum Framework," *Bulletin of Computer Science and Electrical Engineering*, vol. 1, no. 1, pp. 33–41, Jun. 2020, doi: 10.25008/bcsee. v1i1.6.

[19] P. L. Ayunda and E. K. Budiardjo, "Evaluation of Scrum Practice Maturity in Software Development of Mobile Communication Application," in *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, Sep. 2020, pp. 317–322. doi: 10.1109/IC2IE50715.2020.9274625.

[20] G. Rodriguez, I. Gasparini, A. Kemczinski, and A. Veloso de Matos, "Students' Perception of Scrum in a Course Project," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 16, no. 4, pp. 329–336, Nov. 2021, doi: 10.1109/RITA.2021.3136436.

[21] M. Hasnain, M. F. Pasha, I. Ghani, and M. I. Babar, "An Efficient Performance Testing of Web Services," in *2019 22nd International Multitopic Conference (INMIC)*, Nov. 2019, pp. 1–8. doi: 10.1109/INMIC48123.2019.9022763.