

Original Research Paper

Intelligent Egg Incubator

Kaiyisah Baiduri Azahar¹, Ericka Ensima Sekudan¹, Ahmad Mirza Azhar¹

¹ Program of Computer Science, Department of Computing, Faculty of Arts, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, Malaysia.

Article history

Received:
30.08.2020

Revised:
17.09.2020

Accepted:
26.09.2020

*Corresponding Author:

Kaiyisah Baiduri Azahar

Email:

kesyabaiduri@gmail.com

This is an open access article,
licensed under: [CC-BY-SA](#)



Abstract: This research presents the design and development of a chicken egg incubator. The aim is to design and construct an “Intelligent Egg Incubator” that can hatch chicken eggs automatically. In the design, author used a temperature control system that controls the temperature of the incubator. This system consists of a temperature sensor, heating and cooling elements and a controller. If the temperature of the incubator exceeds a certain value, the cooling unit will operate to reduce the temperature while if the temperature goes below another threshold value, the heating element will start to operate to increase the temperature. Furthermore, author used a motion sensor to detect the movements of the eggs. If there is a movement inside the incubator, the owner of the incubator will get the notification about the conditions of the eggs.

Keyword: Egg Incubator, Intelligent, Internet of Things, Temperature Control.



1. Introduction

An egg incubator is used to hatch bird eggs. The incubator keeps the eggs warm, allowing the fetuses inside them to grow and hatch without the mother present. It helps farmers to hatch eggs automatically without the need of human intervention. Furthermore, egg incubator will not only improve poultry production considerably, but it will also allow regularity in income making, enabling subsistence farmers to transition into possible rural entrepreneurship. As incubation proceeds, the eggs will normally become lighter, and the air space within the egg will normally become larger, owing to evaporation from the egg [1]. The challenge is to keep and maintain all conditions required for the eggs to hatch such as temperature, lighting, humidity and airflow.

The Intelligent Egg Incubator is designed to improve the available eggs incubator in order to change the traditional farming method into advance and modern farming method. Nowadays, agriculture and livestock farming is an example of important field that continues to grow with further technologies developments. More technologies were created to further develop this field. To achieve this goal, the project is created with efficient, quick and easy to operate. The Project is "Intelligent Egg Incubator" that will focus more on automate controlling the process using electronic system.

The main purpose of this project is to design so that the hatching eggs were more successful than the incubation manually. In this projects, microcontroller software applied to control temperature. This microprocessor is connected to the relay circuit so that it can control the bulbs and fans automatically. If the temperature is supplied in excess of 37°C-38°C and it can be detected by a temperature sensor circuit, while the relay circuit will turn off the bulb and turn on the fan to reduce the temperature inside the incubator. If the temperature is less than 37°C, the bulb will turn on and the fan will turn off. The project requires a temperature of about 37°C to 38°C to ensure that the incubation of the eggs were more evenly so that the result obtained are of better quality. Egg incubation takes about 20 days before he was transferred to the hatchery.

To build this system, the hardware part consists of electronic circuit for the microcontroller, temperature sensor, motion sensor, bulb, fan, power supply circuits and power supply 12V. The software consists of Microcontroller. Two microcontrollers are used for control the entire project, Arduino Uno and ESP8266. Then, temperature sensors are used to detect temperature of the incubator and motion sensor are used to detect movements of the eggs. This project started with the research about the related information via the books and internet. Besides that, the lecture's guidance is also important for the development. Sketch about the project will be done based on the information gathered.

Overall, there are several objectives that have been identified in order to achieve the aim of the project. The objectives of this project are to investigate any current egg incubator, to design the suitable egg incubator based on IoT, to develop an egg incubator based on IoT, namely "Inkubator Pintar", and to evaluate the performance of the "Inkubator Pintar".

2. Literature Review

Recent advanced development on IoT technology has drive many low-cost innovative projects within broad fields. Take for the example, recent development by displaying whether pulse rate is normal or not [2] and water level indicator for resident's early warnings [4]. The same things also happen in agriculture and farming fields. Omar et al. [5] design Smart Egg Incubator System based on Arduino Uno and use water and fan to control the humidity and ventilation. Separately, Sanjaya et.al [6] develop smart incubator based on Arduino for quail eggs. The results show the system successfully hatch a quail eggs normally with 87.55% success rate over 490 eggs during 17 days incubation period.

2.1. Methodology

Methodology is the general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it. These methods, described in the methodology, define the means or modes of data collection or, sometimes, how a specific result is to be calculated. Methodology does not define specific methods, even though much attention is given to the nature and kinds of processes to be followed in a particular procedure or to attain an objective.

When proper to a study of methodology, such processes constitute a constructive generic framework, and may therefore be broken down into sub-processes, combined, or their sequence changed.

2.1.1. Agile

AGILE methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall model. The agile software development emphasizes on four core values. Firstly, individual and team interactions over processes and tools. Secondly, working software over comprehensive documentation. Thirdly, customer collaboration over contract negotiation. Lastly, responding to change over following a plan.



Figure 1. Agile Methodology

2.1.2. Rational Unified Process (RUP)

Short for Rational Unified Process, a software development methodology from Rational. Based on UML, RUP organizes the development of software into four phases, each consisting of one or more executable iterations of the software at that stage of development. Firstly, inception, in this stage, the projects business case is stated and the team decides if the project is worth doing or if it is even possible. It is important to the process to first formulate the scope of the project and also determine what resources will be needed. Next elaboration, in this stage, the developers take a closer look at the project to determine its architecture foundation and to evaluate the architecture in relation to the project. This stage is important to the RUP because it is here that developers analyze the risks associated with changing the scope of the project or adding new technologies along the way. Next construction, in this stage, the development of the project is completed. The application design is finished and the source code is written. It is in this stage that the software is tested to determine if the project has met its goal laid out in the inception phase. Next transition, in this stage, any fine-tuning is performed. Any final adjustments can be based on user feedback, usability or installation issues. RUP is similar in concept to Extreme Programming in that only what is useful and required is produced and the development plan is updated throughout the process. Both methods seek to develop a system of best practices in software development.

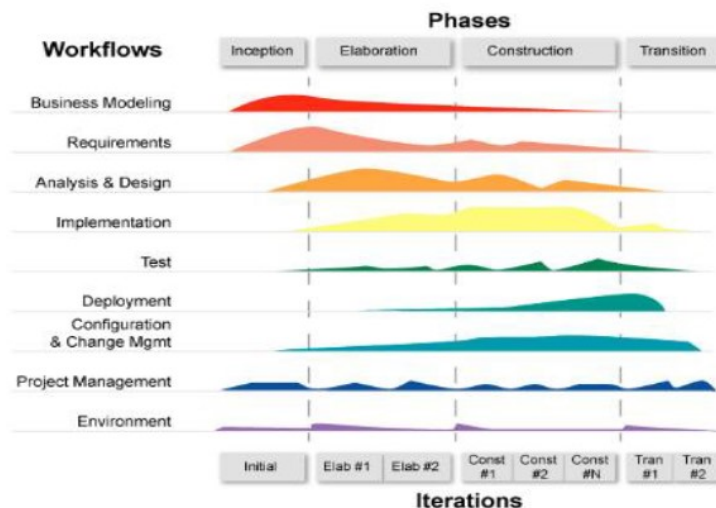


Figure 2. RUP Methodology

2.1.3. Scrum

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process.

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. This is why in Scrum development, for example, a sprint planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint) instead of a set of Entry criteria, Task definitions, Validation criteria, Exit criteria (ETVX) and so on, as would be provided in most methodologies.

Scrum relies on a self-organizing, cross-functional team. The scrum team is self-organizing in that there is no overall team leader who decides which person will do which task or how a problem will be solved. Those are issues that are decided by the team as a whole. And in Scrum, a team is cross functional, meaning everyone is needed to take a feature from idea to implementation.

Within agile development, Scrum teams are supported by two specific roles. The first is a ScrumMaster, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level. The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

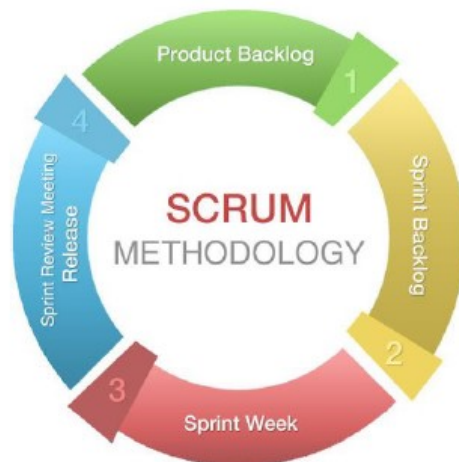


Figure 3. Scrum Model

2.1.4. Prototype

Prototype methodology is defined as a Software Development model in which a prototype is built, test, and then reworked when needed until an acceptable prototype is achieved. It also creates a base to produce the final system. Software prototyping model works best in scenarios where the project's requirement are not known. It is an iterative, trial, and error method which take place between the developer and the client. There are four types of Prototyping models it is Rapid Throwing prototypes, Evolutionary prototype, Incremental prototype and Extreme prototype. Rapid throwaway is based on the preliminary requirement. It is quickly developed to show how the requirement will look visually. The customer's feedback helps drives changes to the requirement, and the prototype is again created until the requirement is baselined. In this method, a developed prototype will be discarded and will not be a part of the ultimately accepted prototype. This technique is useful for exploring ideas and getting instant feedback for customer requirements. Here, the prototype developed is incrementally refined based on customer's feedback until it is finally accepted. It helps you to save time as well as effort. That's because developing a prototype from scratch for every interaction of the process can sometimes be very frustrating. This model is helpful for a project which uses a new technology that is not well understood. It is also used for a complex project where every

functionality must be checked once. It is helpful when the requirement is not stable or not understood clearly at the initial stage.

In incremental Prototyping, the final product is decimated into different small prototypes and developed individually. Eventually, the different prototypes are merged into a single product. This method is helpful to reduce the feedback time between the user and the application development team. Extreme prototyping method is mostly used for web development. It consists of three sequential phases. Firstly, basic prototype with all the existing page is present in the HTML format. Secondly, simulate data process using a prototype services layer. Lastly, the services are implemented and integrated into the final prototype.

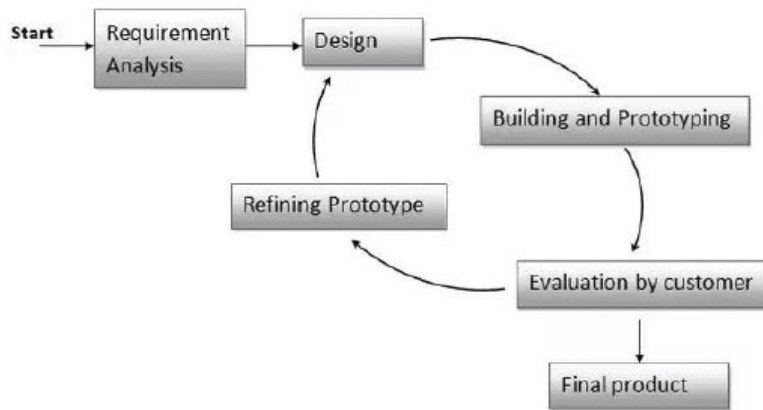


Figure 4. Prototype Model

3. Methodology

This section will be explained in details about the methodology that has been choose for this project. It will be discussed about the phases involved, the activities for each phase, the deliverables and the objectives to be met for each phase.

3.1. Reason of Choosing Prototype Model

The lean startup methodology says that prototyping is a high-priority stage since it helps verify your idea from the technical and market points of view in a very short term and using minimal resources. Methodology used in developing Intelligent Egg Incubator is Prototyping methodology. The reason why author choose this methodology are users can actively involved in development. Therefore, errors can be detected in the initial stage of the software development process. Missing functionality can be identified, which helps to reduce the risk of failure as Prototyping is also considered as a risk reduction activity. Helps team member to communicate effectively. Customer satisfaction exists because the customer can feel the product at a very early stage. There will be hardly any chance of software rejection. Quicker user feedback helps you to achieve better software development solutions. Allows the client to compare if the software code matches the software specification. It helps to find out the missing functionality in the system. It also identifies the complex or difficult functions. Encourages innovation and flexible designing. It is a straightforward model, so it is easy to understand. No need for specialized experts to build the model. The prototype serves as a basis for deriving a system specification. The prototype helps to gain a better understanding of the customer's needs. Prototypes can be changed and even discarded. A prototype also serves as the basis for operational specifications. Prototypes may offer early training for future users of the software system.

3.2. Overview Project Methodology

The prototyping model is a systems development method in which a prototype is built, tested and then reworked as necessary until an acceptable outcome is achieved from which the complete system or product can be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes

place between the developers and the users. In most cases, the steps of the prototyping model are as follows, firstly, the new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system. Secondly, a preliminary, simple design is created for the new system. Thirdly, first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product. Fourthly, the users thoroughly evaluate the first prototype and note its strengths and weaknesses, what needs to be added and what should to be removed. The developer collects and analyzes the remarks from the users. Fifth, the first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed. Sixth, the second prototype is evaluated in the same manner as was the first prototype. Seventh, the preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired. Eighth, the final system is constructed, based on the final prototype. Ninth, the final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

3.3. Requirements Gathering and Analysis Phase

The prototyping model is applied when detailed information related to input and output requirements of the system is not available. In this model, it is assumed that all the requirements may not be known at the start of the development of the system. It is usually used when a system does not exist or in case of a large and complex system where there is no manual process to determine the requirements. This model allows the users to interact and experiment with a working model of the system known as prototype. The prototype gives the user an actual feel of the system. At any stage, if the user is not satisfied with the prototype, it can be discarded and an entirely new system can be developed. A prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what their expectation from the system is.

3.4. Quick Design Phase

A quick design is carried out and a prototype is built. The developed prototype is submitted to the customer for evaluation. Based on the customer feedback, the requirements are refined and the prototype is suitably modified. The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype.

3.5. Build a Prototype Phase

By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

3.6. Initial user Evaluation Phase

Testing a prototype / developed design is a very important part of the design and manufacturing process. Testing and evaluation, simply confirms that the product will work as it is supposed to, or if it needs refinement. In general, testing a prototype allows the designer and client to assess the viability of a design. In this stage, the proposed system is presented to the client for an initial evaluation. It helps to find out the strength and weakness of the working model. Comment and suggestion are collected from the customer and provided to the developer.

3.7. Refining Prototype Phase

Prototyping is a development methodology in which a model is quickly constructed to test or illustrate design features and ideas, in order to gather user feedback. Subsequent models are created by refining earlier versions, with the aim of convergence on the desired end product. If the user is not happy with the current prototype, you need to refine the prototype according to the user's feedback and

suggestions. This phase will not over until all the requirements specified by the user are met. Once the user is satisfied with the developed prototype, a final system is developed based on the approved final prototype.

3.8. Implement Product and Maintain Phase

The implement product and maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements. Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance for minimizing downtime and prevent large-scale failures.

In conclusion, the prototype model is used because of its advantages. It gives the clear picture of software what is being developed and not to the client earlier and before the implementation stage as well. It helps in saving the later part of issues that might occur. It helps in communication of client and developer to make the product as per requirement and easier to understand them as well for the developers from the viewpoint of the customer. The prototype model provides different types of model as well.

4. Development

This section will discuss about the overview of proposed design and the development of the Intelligent Egg Incubator project. The design and development phase are the next phase after the requirement phase. The process of the development can be done by using the available software. In addition, this section also discuss about the tools and technology used in the development of the project. Moreover, this section will discuss about the interface of the system would flow so that the reader will easily understand.

4.1. Requirement Gathering

Requirement gathering is the initial phase of the Prototype Methodology where the information will be gathered required to the project title. Requirements gathering and analysis, a prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what their expectation from the system is. This will make the project clearly understand by the developer. The second phase is design.

4.2. Design

The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype. Next, the third phase is build a prototype.

4.3. Build the Prototype

In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

4.4. Development

Development is the crucial phase as the developer will start build the system on this phase. During this phase, there are 2 major software that been used to develop Intelligent Egg Incubator, Arduino IDE for microcontroller and Blynk application for the notifications and also Thingier.io for the database. Microcontroller contains the functionality of the system. The main function of the Intelligent Egg Incubator is the control function where the user can just let the incubator at anyplace, anytime, anywhere, because the fan and bulb can automatically switch on and off based on the current temperature and will be control the temperature of the incubator productively.

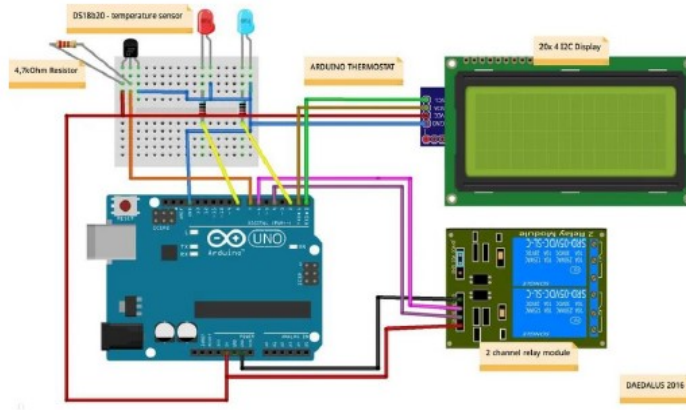


Figure 5. Schematic Diagram – Temperature Controlled Bulb

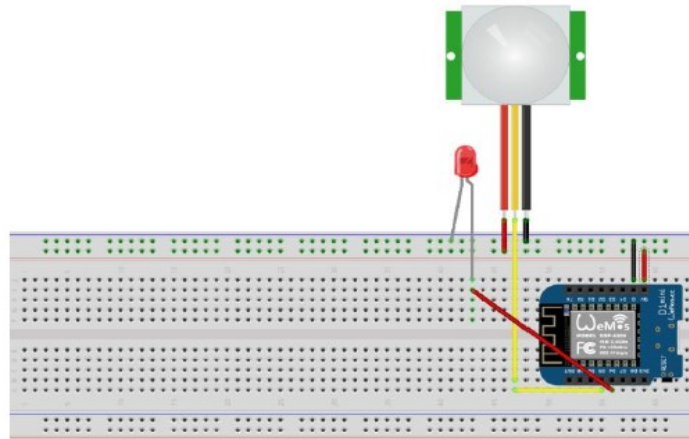


Figure 6. Schematic Diagram – Motion Sensor and Notification

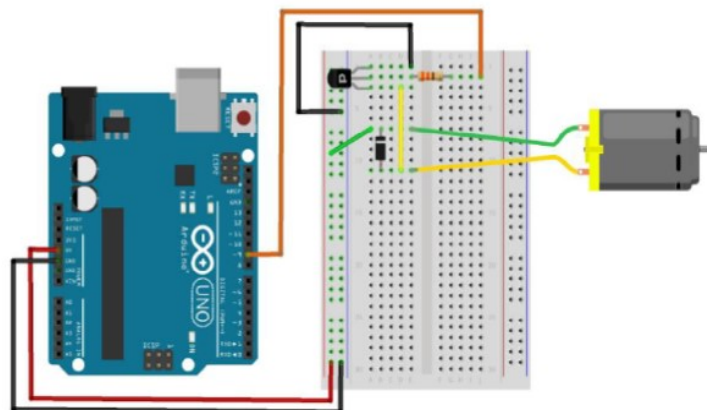


Figure 7. Schematic Diagram – Temperature Controlled Fan

4.5. Components

In this section will show how the components of the project been connected and what type of the components that been used to produce this project. Various type of components used to complete the Intelligent Egg Incubator project based on Table 1.

Table 1. The Components that used in Intelligent Egg Incubator

Name	Description
Arduino Uno	Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, an ICSP header and a reset button.
PIR Sensor	A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They work entirely by detecting infrared radiation (radiant heat) emitted by or reflected from objects.
Fan	Cooling element.
Bulb	Heating element.
5V Power Source	A power source for Arduino Uno.
Miniature Motor	A rotary electrical machine that converts direct current electrical energy into mechanical energy.
LED	A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.
Diode	The fundamental property of a diode is its tendency to conduct electric current in only one direction.
NPN Transistor	NPN transistor amplifies the weak signal enter into the base and produces strong amplify signals at the collector end.
ESP8266	The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability.

These are a component that been used for build Intelligent Egg Incubator project. Combination of this component will make Intelligent Egg Incubator come to life and can be operated based on the requirements that been gathered. The heating components consists of bulb, LED, Arduino Uno, jumper wire, relay, LM35 temperature sensor, resistor, USB cable. While the cooling components consists of fan, Arduino Uno, NPN transistor, miniature motor, diode, jumper wire, USB cable, LM35 temperature sensor. LED as temperature indicator components consists of two differ colours of LEDs, Arduino Uno, USB cable, jumper wire, LM35 temperature sensor, resistor. Finally, the motion detector and notifications components consists of PIR motion sensor, ESP8266, jumper wire, LED.

4.6. Testing and Analysis

A testing been made to see how the Intelligent Egg Incubator work this include:

- Fan and bulb switch on and off automatically based on the current temperature. For temperature testing, Intelligent Egg Incubator been test to ensure the heating and cooling elements functions properly. If the temperature exceed or equal to 37°C the fan will switch on and the bulb will switch off. Then, if the temperature less than 37°C the fan will switch off and the bulb will switch on.
- Motion sensor detect the movement of the egg. If there is a movement the notification will be send to the user. But, if there is no movement, so there will be no notification send.
- LED as temperature indicator

4.7. Responses

In conclusion, design represents as a ‘face’ of the system. Hence in this section, the design of Intelligent Egg Incubator was discussed where which architecture system to apply and what is the flow of the system.



Figure 8. Response for Usability, Functionality, Performance, Capability, and Security Testing

Other than that, storyboard or user interface was created to see the function for each page of the system and how the design for the system should be.

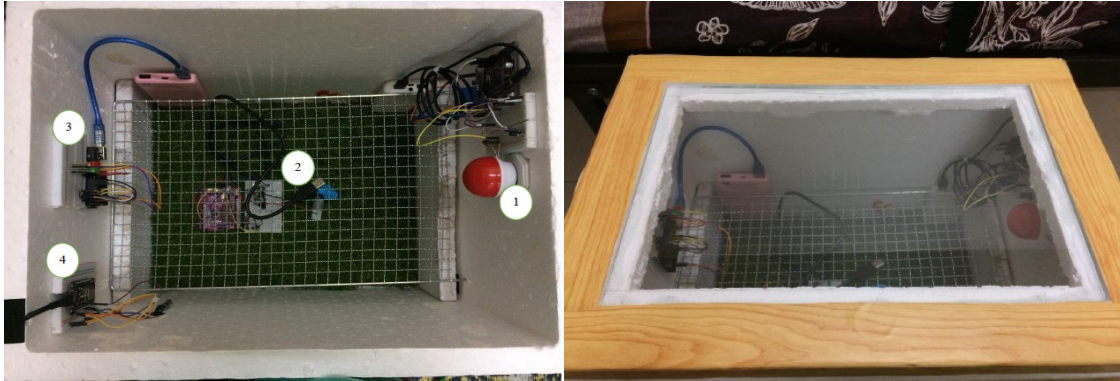


Figure 9. System Prototype

5. Conclusion

This section will point out the conclusion that can be concluded from the project, the further work for the project, the limitation during this project and the recommendation that can be done to enhance this project in the future.

As the system for the project has been completed, the objectives of the project have been achieved. The first objective of this project is to gather and analyze Intelligent Egg Incubator. There are several activities that have been done to achieve the first objective. The activities are gathering the requirements, analyze the requirements and lastly document the requirement. All of the activities are done by interacting with distribute the questionnaire, observing the existing application and reviewing the existing documents resulting in delivering the deliverables for the first objective. Next, to design the suitable diagram for the guideline in development phase. After all requirements had been collected, it must be proceeded to design phase where the development team can have the guideline on how to build the project and meet the requirement. The next objective of this project is to develop a complete system. As the system is completed, the final objective is achieved. Lastly, the project that have been made must be test to ensure the requirement that been gathered earlier had been reached to give full satisfaction to the user who use Intelligent Egg Incubator.

Currently, the limitation of the Intelligent Egg Incubator consists of:

1. Cannot put too many eggs inside incubator at the same time.
2. Perilous if the incubator put under the hot day or rainy day.
3. Do not have a monitoring components in the incubator to make sure that the eggs are in a good conditions.

Further suggestions to improve the work could be done by change LM35 temperature sensor to the DHT11 which is temperature and also humidity sensor and put a camera to monitor the eggs.

In this paper the component designed is easy to maintain, affordable and portable. Thus having result for days of the testing, an average value of temperature was gotten to be 35°C. The IEI also can hatch of different poultry eggs are within a similar range, such as guinea fowl, turkeys, ducks, goose, guinea fowl, quail and ostrich. For recommended for household use, subsistent poultry farmers to increase the production of poultry products. Besides that, another recommended is the solar energy should be used as backup power supply. It is because of the untimely failure of electricity to enhance the efficiency of the system.

References

- [1] M. B. Ramli, H. P. Lim, M. S. Wahab, and M. F. Zin, "Egg Hatching Incubator Using Conveyor Rotation System," *Procedia Manufacturing*, vol. 2, pp. 527-531, 2015. doi:10.1016/j.promfg.2015.07.091.
- [2] M. A. Azizi, and A. Zariman, "Displaying Health Status Based IoT," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 25-35, 2020. [Online]. Available: <https://doi.org/10.36079/lamintang.ijortas-0201.57>. [Accessed: June 2020]
- [3] M. S. Ab Latif, A. A. Ismail, and A. Zariman, "Smart Mirror for Home Automation," *International Journal of Recent Technology and Applied Science*, vol. 1, no. 1, pp. 1-11, 2020. [Online]. Available: <https://doi.org/10.36079/lamintang.ijortas-0101.55>. [Accessed: June 2020]
- [4] A. A. Ismail, M. A. Azizi, and A. Zariman, "Smart Water Level Indicator," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 48-58, 2020. [Online]. Available: <https://doi.org/10.36079/lamintang.ijortas-0201.59>. [Accessed: June 2020]
- [5] M. Omar, H. Haris, M. Hidayat, I. Ismail, and M. Seroji, "Smart eggs incubator system," *IJSSST*, vol. 17, 2018. doi:10.5013/IJSSST.a.17.41.35
- [6] W. S. Sanjaya, S. Maryanti, C. Wardoyo, D. Anggraeni, M. A. Aziz, L. Marlina, A. Kusumorini, "The development of quail eggs smart incubator for hatching system based on microcontroller and Internet of Things (IoT)," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018. doi:10.1109/icoiact.2018.8350682.
- [7] H. Hermansyah, K. Kasim, and I. K. Yusri, "Solar Panel Remote Monitoring and Control System on Miniature Weather Stations Based on Web Server and ESP32," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 1-24, 2020. <https://doi.org/10.36079/lamintang.ijortas-0201.56>.
- [8] M. H. Hazhari, M. A. Azizi, and A. Zariman, "Smart Delivery Agent," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 36-47, 2020. [Online]. Available: <https://doi.org/10.36079/lamintang.ijortas-0201.58>. [Accessed: June 2020]
- [9] I. D. Hashim, A. A. Ismail, and M. A. Azizi, "Solar Tracker," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 59-65, 2020. <https://doi.org/10.36079/lamintang.ijortas-0201.60>. [Accessed: June 2020]
- [10] G. D. Shivamadhu, A. Venkatesh, A. Alva, D. Nausheer, and K. A. S. Devi, "PortaX Secure Automation System Using IoT – A Survey," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 66-74, 2020. [Online]. Available: <https://doi.org/10.36079/lamintang.ijortas-0201.61>. [Accessed: June 2020]
- [11] O. P. Singh, R. Ranjith, S. Reddy, and R. Soni, "Survey Report on Sending Data Securely using IOT Over Cloud", *ijortas*, vol. 2, no. 1, pp. 75-80, Mar. 2020.