Vol. 08, No. 01, January 2024, pp. 69-86

INTERNATIONAL JOURNAL OF INNOVATION IN ENTERPRISE SYSTEM

Journal homepage: https://ijies.sie.telkomuniversity.ac.id
e-ISSN: 2580-3050, DOI: https://doi.org/10.25124/ijies.v8i01.206

# E-Farm Livestock Platform Requirements Engineering Using Loucopoulos and Karakostas Iterative Process Model

Liana Trihardianingsih[1*], Maie Istighosah[2], Ariel Yonatan Alin[3], Muhammad Ryandy Ghonim Asgar[4]

[1] Master of Informatics Enginering
Amikom University Yogyakarta
liana.trihardianingsih@students.amikom.ac.id

[2] Master of Informatics Enginering
Amikom University Yogyakarta
maie.is@students.amikom.ac.id

[3] Master of Informatics Enginering
Amikom University Yogyakarta
ariel.s2@students.amikom.ac.id

[4] Master of Informatics Enginering
Amikom University Yogyakarta
muhammadryandy@students.amikom.ac.id

*liana.trihardianingsih@students.amikom.ac.id

ARTICLE INFO

Article history:
Received 11 January 2023
Accepted 11 October 2023
Published 31 January 2024

ABSTRACT

Global human population growth has forced farms to evolve in order to produce more livestock products more efficiently while also paying attention to public health, environmental sustainability, and animal welfare. However, problems arise when some diseases appear to affect farm animals and large companies providing livestock products dominate the market. It is necessary to develop a platform or application that can be used to solve these two problems, especially for breeders who have farms on a small scale. This study aims to outline the process of understanding engineering requirements by utilizing the Loucopoulos and Karakostas Requirements Engineering Process Model method, which consists of elicitation of requirements, specification of requirements, as well as validation and verification of requirements. The development process is carried out by hiring breeders and potential customers to determine the priority needs of the platform. The results showed that of the 25 defined functional needs, there were 22 final functional needs that were validated with values above 50%. The E Farm platform should be further developed based on the defined demands since a total of 22 validated needs have been determined to be able to represent 88% of the needs required by users.

Keywords:
Engineering Requirements;
Requirement Elicitation;
Requirement Specification;
Requirement Validation and Verification; Farm.

# 1. INTRODUCTION

In the era of the industrial revolution 4.0, technology has successfully entered the world of animal husbandry or commonly referred to as smart Livestock. Smart Livestock aims to transform traditional techniques towards innovative solutions based on Information Communication Technology (ICT). Concretely, technologies such as Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs), Image Processing, Machine Learning, Big Data, Cloud Computing, and Wireless Sensor Networks (WSNs), with the hope of being able to bring significant changes in the field of animal husbandry [1]. One form of livestock digitization that has been carried out is the existence of Precision Livestock Farming (PLF). The use of PLF allows farmers to maximize the potential of Livestock because farmers are able to overcome problems such as monitoring disease outbreaks that attack Livestock and preventing food-related economic losses [2].

The use of IoT, Artificial Intelligence, and big data to develop software is also able to provide convenience in the field of animal husbandry because the application and development of software can improve user service to users [3]. However, to create software that is able to solve user problems, it is necessary to carry out the process of engineering software needs. Software needs engineering is an approach process that is carried out in building software through several stages, namely collecting, analyzing, and implementing all the requirements of the software product needs [4]. This stage is an important stage in the software development process, as it has a very significant impact on the results to be developed [5]. That is because the success or failure rate of information technology projects depends on the results of the requirements engineering process [6].

Requirements such as user requirements and system requirements are gathered, comprehended, and analyzed during the first stage of the software development life cycle, which is called requirement engineering [7]. Because it significantly affects how the application will turn out, this is a crucial point in the development process [8]. Every stakeholder, including the user, is involved in the requirements engineering phase, which in general results in a definition of the necessary application features and capabilities. Because it determines whether an information technology project will succeed or fail, the requirement engineering process is essential. Requirement engineering is a branch of systems engineering that focuses on the discovery, creation, tracking, analysis, specification, communication, and support of requirements at various abstraction levels [9]. As a result, the user's requirements based on business processes are specified and examined during the requirement engineering phase of software development. An approach for getting a better understanding of and insight into user expectations and demands is called "user-centered requirement engineering" (UCRE).

In this study, researchers engineered the need for software for livestock platforms, E-Farm, with the aim of being able to assist farmers in managing their farms and providing understanding and documentation for the development of these platforms. This is also related to the results of the 2013 Agricultural Census, which states that the number of livestock households in Indonesia reaches 13.56 million households. This condition calls for the availability of livestock products to be able to meet the consumption needs of people's calories and animal protein to improve the quality of human resources [10]. Based on the survey, the idea emerged from engineering software needs that can overcome several problems that occur in animal husbandry. To address the failure of platform development, as previously explained, researchers focused on understanding the engineering needs for platform development in the livestock sector by using the Loucopoulos and Karakostas Iterative Requirements Engineering Process Model methods through three stages, namely requirement elicitation, requirement specification, and requirement validation and verification. Research conducted by Bangkalang et al. [5] used the Loucopoulos and Karakostas Iterative Requirements Engineering Process Model methods to meet platform needs related to women's violence. This model was successfully applied to define in detail the problems and needs of helping victims of violence against women using use cases and prototyping as a result of documentation. However, the use of this method has only been described in a simple project environment and does not involve many complex requirements.

E-farm is a platform used to help breeders manage livestock and sell livestock and their livestock products. This platform involves many requirements because it adopts AI, IoT, and big data technologies in its implementation. IoT implementation is used to monitor animal health and stable conditions so that it can prevent the spread of diseases that attack livestock. Animal health will be monitored using an IoT collar, which can automatically help detect animal health by processing data on several indicators such as heart rate, temperature, and breathing patterns. Furthermore, sensors are used to monitor the condition of the cage by calculating the temperature and ammonia levels in the cage. The complexity of the features involved means that the requirements must be clearly defined. This study tries to test and analyze the effectiveness of the Loucopoulos and Karakostas Iterative Requirements Engineering Process Model method in a project environment that involves complex requirements and diverse needs. To maximize the performance of the method, the authors added a focus group discussion (FGD) involving ten potential users to identify, analyze, and validate needs.

According to Hina et al [11] Requirement engineering is carried out to collect requirements from users and develop the system by a team of developers to meet user requirements. In software development, requirements engineering plays an important role as a product blueprint. This is because software development is directly related to product quality and user satisfaction. Prospective users need to be involved in the process of identifying their requirements and expectations/feedback regarding the system, therefore that the development team can validate these requirements for successful system development [12]. Although the process for gathering requirements is difficult, it must be done to understand what the user wants. Requirements contain a variety of good information about the problem, statement, system behavior, nature, design constraints and implementation [13]**.** Therefore, to facilitate the collection of requirements, it is necessary to use a requirement engineering model.

Research [14] succeeded in designing an agricultural information guide application for farmers by applying the user-centered requirements engineering method. This user-centered requirements engineering method is used to help the process of classifying farmers' information needs which will then be packaged using an integrated agricultural information application design model. Other research conducted [15] built a web-based Learning Management System (LMS) application using Kotonya and Sommerville Linear Process Model as an integrated system. Needs engineering is carried out through the stages of elicitation, analysis and negotiation documentation, and validation. Based on the engineering of these needs, the results of the prototype are obtained to be able to provide an overview of the entire system built through the validation stage.

In the research of Nadeem et al. [16], of the four RE models compared, the Loucopoulos and Karakostas models have better performance than the other models in terms of security issues. Requirements engineering in the Loucopoulos and Karakostas models is carried out by the process of developing requirements through an iterative cooperative process to analyze problems systematically, then documenting the results of observations in various representation formats, as well as checking and validating the knowledge obtained [17]**.** One of the studies that adopted the Loucopoulos and Karakostas model was a study conducted by Bangkalang et al. [5] This research designed a platform for victims of violence against women to report and provide psychological assistance directly by applying the Loucopoulos and Karakostas model. The model was applied as a guide for application development and succeeded in uncovering details of the problem and the need to help victims of violence against women by using documentation in the form of use case diagrams, activity diagrams, database designs, and easy-to-understand user interface designs. So that in this study researchers will also adopt the Loucopoulos & Karakostas model to identify the engineering requirements needed for the development of the e-farm platform by adding an interview method to define needs and give priority to each need based on the assessment given by the user.

## 2. METHOD

In software engineering development, the first phase that must be passed is Requirement Engineering. Requirement Engineering is the process of collecting, documenting, and defining the needs of the system, where software requirements are sourced from users and customers in the form of information about the purpose, functions, and limitations of the system to be applied [18]. After the requirements of these various sources are collected, then extraction is carried out, and continued analysis and documentation for software engineering development. In collecting engineering requirements, a lot of good skills are needed because the process is very complicated and difficult [9]. In research [10], Requirements Engineering, which was not well observed, was identified as the main cause of failure in software development.

The study adopted the method Loucopoulos and Karakostas. The Loucopoulos and Karakostas methods provide the idea of a process of repetition and the process model of cycle needs. This model describes the relationships that occur between needs engineering phases such as elicitation, requirement specification, and validation of the problem domain iteratively. This method consists of Elicitation, Specification, and Validation and Verification. Elicitation is also called the needs collection stage because the main activity carried out is the collection of needs from stakeholders [4],[11]. Elicitation activities include identifying problems and formulating system constraints. The next stage is the specification stage. This stage is carried out to check the completeness, consistency, and feasibility of requirements through the process of prioritization, communication, and negotiation. In this arrangement, the requirements are composed in a formal document with the aim of providing a bridge between developers and stakeholders [11]. Therefore, the documents produced by the requirements must be translated clearly, consistently, concisely, and decently [12]. The final stage is validation and verification. At this stage, confirmation is carried out that the user's needs and needs specifications are complete and correct. One of the methods used is prototyping.

Therefore, to solve problems in requirements engineering, systemization of the requirement process between the developer and the user needs to be carried out in 3 processes, namely: elicitation, specification, validation and verification.

This method is known as The Three Dimensions of Requirement Engineering. The process model engineering requirements in this study apply Loucopoulos and Kanakostas, as in Figure 1 [5].

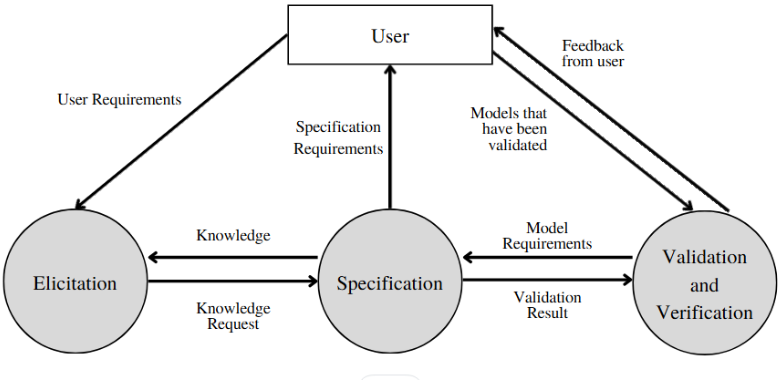



**Figure 1 - Loucopoulos and Kanakostas Iterative Requirements Engineering Process Model**

### 2.1. Elicitation

The first phase in needs engineering that needs to be done is the elimination of needs, where in this process, very critical accuracy is needed because it is prone to errors. The problem that arises stems from the gap in the problem of differences in knowledge of the disciplines owned. Customers are experts in the software domain they want to develop, while developers do not know about the knowledge domain. Even so, it can be overcome by implementing periodic and repetitive interactions (iterations) between developers and customers. According to [13], there are three activities that need to be carried out in the elimination process: (1) identification of stakeholders' problems, needs, and constraints; (2) Making a list of the requirements of each stakeholder with the elicitation technique; (3) integrate the information that has been collected to determine the functionality needs and limitations of the software needed. There are several elimination techniques that can be done, namely: (1) interviews, (2) brainstorming; (3) observation; and (4) case analysis [10]**.** The stages in this first phase are shown in Figure 2

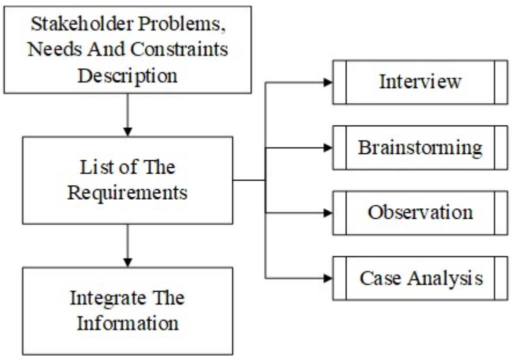



**Figure 2 - Elicitation Stage**

The purpose of the elicitation technique is to obtain requirements and to record as many problems as possible in the process of collecting requirements to find out the appropriate needs of stakeholders.

### 2.2. Specification

The second phase in needs engineering is a specification. The requirement specification is a stage that explains in more detail the stages of elicitation, namely by explaining what and how software that is built is able to work as expected by explaining functionality and non-functionality needs [24]. As illustrated in Figure 3, this requirements specification document comprises function requirements, non-functional requirements, performance requirements, and requirements for external interfaces in addition to function requirements [25]. To help explain in more detail the depiction of functional needs, the tools to be used are use cases.

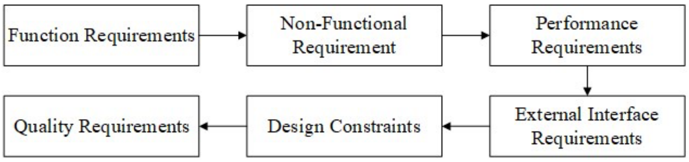



**Figure 3 - Specification Stages**

The Use Case diagram describes the needs of the system to be designed and constructed and provides an overview of the interactions that occur in the system environment [26]. Use Case contains illustrations of a graphic form consisting of actors involved in the system, the needs of actors, and the interactions that occur between these needs. The use of this use case makes clear what needs exist in the system or platform that is designed or built and who are the users who can use the defined functions [27].

### 2.3. Validation and Verification

Requirements Verification and validation are two distinct techniques used to ensure that the requirements gathered satisfy user expectations and increase developer activity to fulfill user expectations or wants [28]. In other words, the purpose of the Verification and validation requirements is to ensure that the product developed is in accordance with customer expectations and conditions [19]. Review, static and dynamic analysis, and formal method testing are just a few of the techniques used in verification and validation to make sure the product functions under the constraints that customers and developers place on it. The difference between the two methods is largely related to their specifications [20]**.**

- Verification is a procedure to determine whether or not the software complies with the specifications provided during the elicitation phase. Additionally, verification serves to validate that the finished product fully complies with the stated specifications. [21].
- The validation process determines whether the specifications created during the elicitation step satisfy all of the user's requirements (users). Developers employ several task sets throughout the validation phase to make sure that the software created can be linked to the needs of the user [21].

Validation requirements can be said to have a role in ensuring that the requirements collected during the elicitation phase are properly documented and meet the needs of customers or users. At the same time, the verification requirements ensure that the software developed meets these specifications or requirements. These two needs are combined to evaluate everything, from built software to user-documented requirements.

## 3. RESULT AND DISCUSSION

### 3.1. Elicitation

According to research conducted by Bernard Ijesunor, this ever-growing population influences the increasing demand for animal food. In addition, managing scarce resources like land, the need to lessen the greenhouse gas emissions caused by livestock, and the necessity of managing intricate, specific, and repetitive daily livestock management routines are difficulties that must be overcome in the production of livestock. The usefulness of the Internet of Things (IoT) in industrial development and its role will be significant if applied to animal husbandry [32]. So, the development of research carried out on the proposed platform is to utilize technology based on IoT, AI, and big data. The proposed platform is named farm, which illustrates that this platform can manage farms digitally. E-farm is not only designed like an information system that can manage sales between farmers and customers but also provides IoT devices that can be used by farmers to monitor Livestock.

The application of IoT on e-arm is the use of smart cameras, cage sensors, and IoT necklaces. Smart cameras are used to monitor Livestock on a wide scale, where the camera will capture several images for further processing on the system and will provide diagnostics related to animal habits. The cage sensor is used to determine the condition of the cage in terms of temperature, humidity, and ammonia gas levels. If abnormal indications are found, handling features such as fans will turn on to maintain the condition of the cage to remain safe and comfortable for Livestock. Furthermore, IoT necklaces are used to check each farm animal's health, as the necklace will be attached to one farm animal. If the resulting diagnosis results look bad, the system will send a notification to the farmer by giving a red status to state that the farm animal needs a doctor's treatment.

The research was conducted by conducting a case analysis of livestock problems in Indonesia. Some of the problems found are; 1) the availability of beef and buffalo in Indonesia in April 2022 experienced a deficit of up to 258.69 thousand tons; 2) farmers find it difficult to recognize diseases that attack farm animals early; 3) some breeders have difficulty in carrying out the maintenance of livestock sheds; 4) livestock waste is not subjected to the maximum processing process; 5) Small breeders are difficult to compete because it is difficult to expand the market network. Based on the problems that have been defined, an Alternative Exploration of Solutions in platform development is presented in Table 1.

**Table 1 - Alternative Exploration of Solutions**

| Aim | Needs | Solution |
|---|---|---|
| Help make the farmer's work easier | Creating a platform that can monitor Livestock to accommodate the sale and purchase of Livestock. | By creating an app that connects farmers with related partners |
| Helping farmers manage cages and monitor the health of farm animals | | By creating an application that is integrated with IoT so that farmers can monitor Livestock through smartphones |
| Expanding the distribution network for buying and selling Livestock | | By creating an application that connects farmers with customers (e-commerce) |

As per the identification carried out in Table 1, the development of a platform always involves user interaction in the system environment. In this development process, as many as ten actors are needed who will interact with each other on this platform. Users for this platform are divided into four groups, namely breeders, customers, admins, and partners. Details of each user on this platform can be seen in Table 2.

**Table 2 - Platform User Details**

| No. | User | Description |
|---|---|---|
| 1. | Admin | The user is in charge of managing all data on the system. |
| 2. | Breeder | Users who have access rights to sell farm animals, manage cages, view partner catalogs, and place partner orders. |
| 3. | Customer | Users who have access rights to view product catalogs and purchase animals or processed Livestock |
| 4 | Friend | |
| | a. Courier | Users who have access rights to send orders |
| | b. Butcher Farm | Users who have access rights to provide livestock slaughter services |
| | c. Cleaner-farm | Users who have access rights to provide cage cleaning services |
| | d. Doctor | Users who have access rights to provide inspection of farm animals |
| | e. Drug Sellers | Users who have access rights to sell veterinary drugs |
| | f. Feed Seller | Users who have access rights to sell animal feed |
| | g. Waste Management | Users who have access rights to manage livestock waste |

The first step in defining the functional requirements of a system or platform is to use the solution in Table 1. In addition to being based on research, defining needs also entails getting feedback from users to ensure that the necessary needs are met. Through the interview step, user feedback is gathered. Stakeholders hold Focus Group Discussions (FGD) to learn more about desired needs. Seven breeders and three users, who could end up as partners or customers, participated in the interview process. The respondents were nevertheless able to describe their demands even if the interviews were semi-structured. Table 3 displays an overview of the traits of the informants.

**Table 3 - Interview Characteristics**

| User | Age | Gender | Use of Personal Computer | Use of Smartphone |
|---|---|---|---|---|
| U1 | 24 | Man | High | Medium |
| U2 | 23 | Man | Medium | High |
| U3 | 25 | Man | High | High |
| U4 | 25 | Man | Medium | Medium |
| U5 | 38 | Man | Medium | Medium |
| U6 | 23 | Man | High | High |
| U7 | 34 | Man | Medium | Medium |
| U8 | 24 | Man | High | High |
| U9 | 24 | Woman | Medium | High |
| U10 | 22 | Man | Medium | Medium |

Based on Table 3, it can be seen that each user has a different level of smartphone and personal computer use. UI, U3, U6, and U8 have a high tendency to use personal computers, but U1 is known to have a tendency to use a smartphone that is neither high nor low (medium). While U4, U5, U7, and U10 have the same preference for personal computers over smartphones, namely medium. U2 and U9 have a strong preference for smartphones and a moderate preference for personal computers. To carry out the development of a platform, the platform that is built must be user-friendly. So, it is very important to consider people's habits when using personal computers and smartphones. Based on Table 3, the platform that is suitable for developing E-Farm is smartphone-based.

### 3.2. Specification

On the basis of the requirements of each user, the criteria that were defined in the previous step will subsequently be individually detailed. Table 4 presents detailed requirements for the functional needs of the platform. Functional needs are the needs of processes used to solve previously identified problems. Functional needs contain what processes and services will later be provided by the system.

**Table 4 - Definition of Functional Application Requirements**

| No | Needs |
|---|---|
| 1 | Farmers can monitor farm animals |
| 2 | Farmers can get information about animal health |
| 3 | Breeders can get information about the cage |
| 4 | Farmers can monitor the location of farm animals |
| 5 | Farmers and customers/users can interact by buying and selling |
| 6 | Breeders can perform detection of illegal objects |
| 7 | Breeders can provide food automatically |
| 8 | Breeders can get an analysis of the farm |
| 9 | Breeders can get financial statements |

| No | Needs |
|---|---|
| 10 | Breeders can manage product catalogs |
| 11 | Breeders can see articles related to animal husbandry |
| 12 | Farmers can choose and contact animal feed seller |
| 13 | Farmers can choose and contact waste managers |
| 14 | Farmers can select and contact Butcher Farm |
| 15 | Farmers can choose and contact Clean Farm |
| 16 | Farmers can choose, contact, and consult a Veterinarian |
| 17 | Farmers can choose and contact the Courier |
| 18 | Breeders can choose and contact the seller of the drug |
| 19 | Breeders can choose and contact recycling providers |
| 20 | Users can view and modify profiles |
| 21 | User can top up, transfer, and view balances |
| 22 | User can see the market price |
| 23 | Users can consult with Customer Support |
| 24 | Users can get the latest information about app activity |
| 25 | Users can get information about the spread of animal diseases in each area |

Non-functional needs are aimed at emphasizing the properties owned by the platform. Table 5 defines details of non-functional needs that have been designed based on the results of previous analyses based on the needs of each user. These non-functional requirements are defined based on the characteristics of the users interviewed, taking into account the users' computer and smartphone usage habits.

**Table 5 - Non-Functional Needs**

| Platform Users | Non-Functional Needs |
|---|---|
| Admin | • Personal Computer/Laptop windows/ios operating system specifications<br>• Intel Core i3/ AMD FX-6300 processor<br>• Internet Network<br>• Web browser |
| Breeder | • Smart Camera<br>• Enclosure Sensors<br>• IoT Necklace<br>• Base Station<br>• Mobile phone with android KitKat 4.4.x or iOS operating system specifications<br>• Internet network |
| Customer | |

| Platform Users | Non-Functional Needs |
|---|---|
| Friend | • Mobile phone with android KitKat 4.4.x or iOS operating system specifications<br>• Internet network |

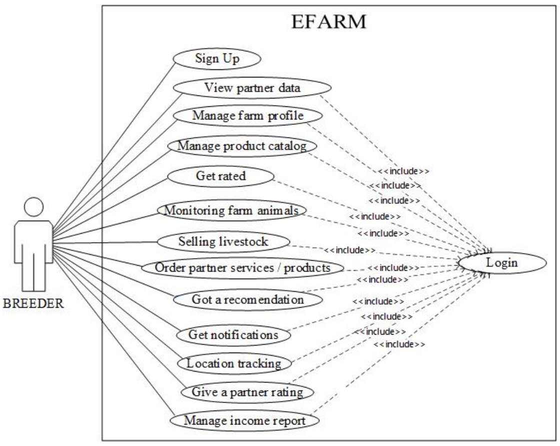
Furthermore, at the stage of specification of functional needs, the requirements that have been defined from the results of the analysis in Table 5 will be translated into a system model using a use case diagram. Before defining a use case, first, identify the use case. Use case identification needs to be done to define the functional picture of the system created to make it easier to create use case diagrams. The identification of use cases can be seen in Table 6. Furthermore, the use case diagram is able to provide details of what functions are in a system and who is entitled to perform these functions.

**Table 6 - Use Case Identification**

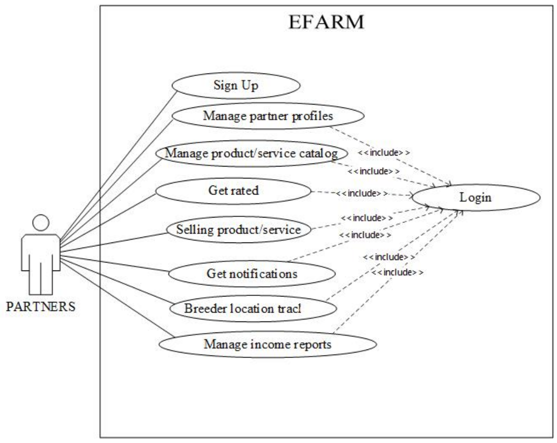
| *Use Case* | Description | Actor |
|---|---|---|
| *Login* | It is the process of inputting *usernames* and passwords that are in accordance with their respective access rights. | Admin, Breeder, Customer, Partner |
| List | Is a process of inputting personal data to create an account according to each access right | Breeders, Customers, Partners |
| View Partner Data | It is a process where farmers can see partner data | Breeder |
| Manage profiles | It is a process admins, breeders, customers, and partners can carry out the process of adding, changing, and deleting personal data profiles. | Admin, Breeder, Customer, Partner |
| Manage catalogs | It is a process that farmers and partners can carry out the process of adding, changing, and removing the catalog of products/services offered. | Breeders, Partners |
| Rated | It is a process for farmers and partners to get an assessment from the customer. | Breeders, Partners |
| Monitoring of farm animals | It is a process where farmers can monitor Livestock and cages through IoT devices. | Breeder |
| Selling farm animals | It is a process where farmers can sell Livestock or processed livestock products. | Breeder |
| Make a partner order | It is a process where farmers can make an order process for products/services provided by partners. | Breeder |
| Get Recommendations | It is a process that includes<br>a. Farmers can get recommendations on the location of partners closest to the location of the farm.<br>b. Customers can get recommendations on the location of the farm closest to the customer's location. | Farmer, Customer |
| Get notified | It is a process by which breeders, customers, and partners can get notifications. | Breeders, Customers, Partners |
| Location Tracking | It is a process that includes<br>a. Farmers can track partner locations<br>b. Partners can track the location of breeders<br>c. Customers can track farm locations | Breeders, Customers, Partners |

| *Use Case* | Description | Actor |
|---|---|---|
| Manage revenue reports | It is a process where farmers and partners can manage income reports | Breeders, Partners |
| Selling Products/Services | It is a process where partners can sell their products/services | Friend |
| View farm data | It is a process where customers can see livestock data. | Customer |
| View a product catalog | It is a process in that customers can see farmer product data. | Customer |
| Make a Payment | It is a process by that customers can pay for the products to be purchased. | Customer |
| View product tracking | It is a process where customers can see the movement of the product being sent. | Customer |
| Rate it | It is a process where customers can see ratings or ratings to couriers (partners) and breeders. | Farmer, Customer |
| Manage partner data | It is a process that the admin can carry out the process of adding, changing, and deleting partner data. | Admin |
| Manage customer data | It is an admin process that can carry out the process of adding, changing, and deleting customer data. | Admin |
| Manage transaction data | It is a process that the admin can carry out the process of adding, changing, and deleting transaction data. | Admin |
| Manage farmer data | It is an admin process that can carry out the process of adding, changing, and deleting farmer data. | Admin |
| Manage farmer product data | It is an admin process that can carry out the process of adding, changing, and deleting farmer product data. | Admin |
| Manage partner product/service data | It is an admin process that can carry out the process of adding, changing, and deleting partner product/service data. | Admin |

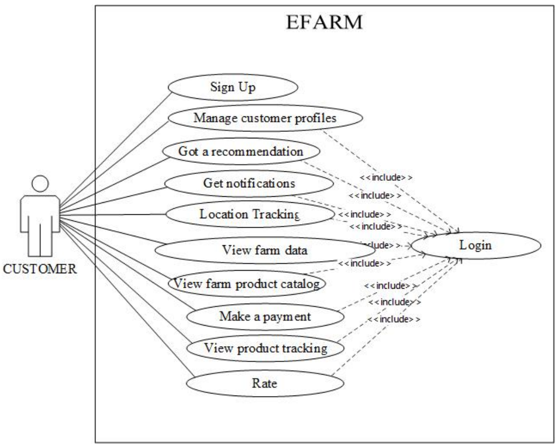
After identifying use cases as described in Table 6, the identification results will be translated into a use case diagram so that the needs of each user can be more easily understood. The results of each use case identification will be grouped into four categories: breeders, administrators, partners, and customers, where each user will be given a use case diagram. Figure 4 depicts the use case diagram in greater detail. Figure 4a is a use case diagram that describes what activities can be carried out by farmers; Figure 4b is a use case diagram that describes what activities partners can do; Figure 4c is a use case diagram that describes what activities can be done by the customer; and Figure 4d is a use case diagram that describes what activities can be done by the administrator.



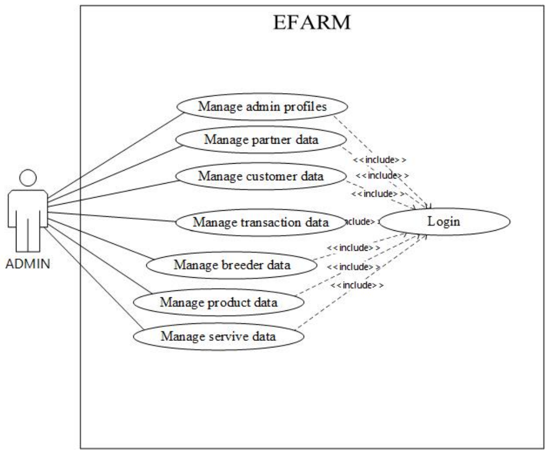

(a) Breeder Use Case




(b) Partner Use Case




(c) Use Case Customer




(d) Use Case Admin

**Figure 4 - User Use Case**

An activity diagram is a graphical flowchart used to explain the operational or business flow of system components. [26]. Activity Diagrams are independent of classes; activity flows in use cases or detailed descriptions of methods. This activity diagram is based on the use case diagram in Figure 4. However, not every use case is made into an activity diagram. In this study, activity diagrams are used to describe the process of connecting farmers and customers, which can be seen in Figure 5, illustrate the process of connecting farmers and partners, which can be seen in Figure 6, illustrate the work process of the IoT necklace, which can be seen in Figure 7, and illustrate the working process of the cage sensor, which can be seen in Figure 8.

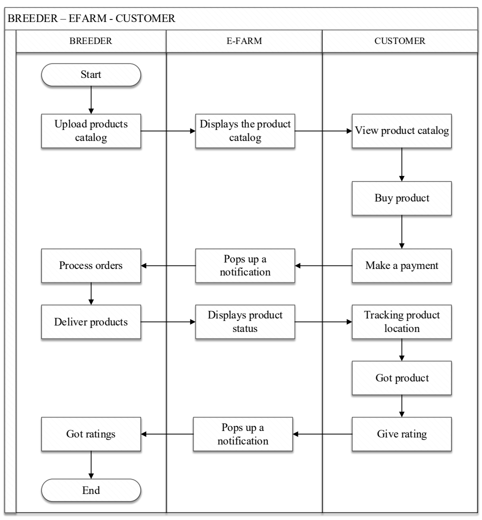



**Figure 5 - Activity Diagram of Breeders with Consumers**

Figure 5 is a feature that illustrates the relationship between breeders and customers in making transactions. The farmer will upload the catalog of the system for the system to be displayed on the system. Customers who see it can immediately make a purchase. If the customer wants to buy it, the customer must immediately make a payment so that the farmer gets a notification and immediately prepares the product, and arranges delivery for the next product to be sent immediately to the customer's address. After the goods arrive at the customer, the customer can give an assessment or rating to the farmer.

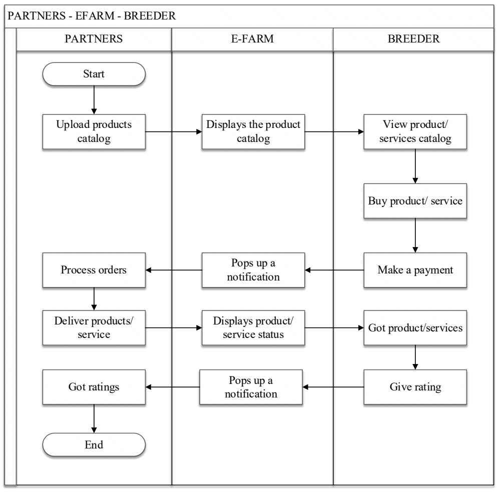



**Figure 6 - Activity Diagram of Breeders with Partners**

Figure 6 is a feature that illustrates the relationship between partners and breeders in making transactions. The partner will upload the catalog to the system for it to be displayed on the system. Farmers who see it can immediately place an order for services or products. If the farmer wants to order, the farmer must immediately make a payment so that the partner gets a notification and immediately prepares the product/service, and arranges the process so that it can be sent immediately to the farmer's address. After the partner has finished processing the order, the farmer can provide an assessment or rating to the partner.

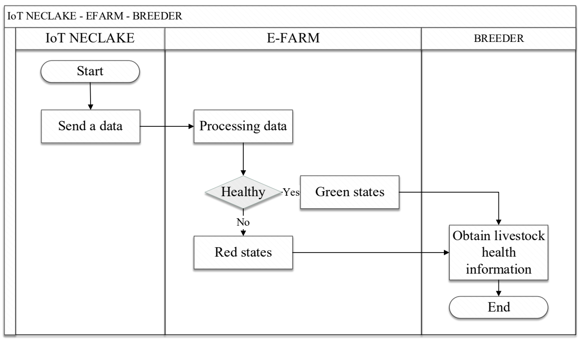



**Figure 7 - IoT Necklace Diagram Activity**

In the IoT necklace diagram activity depicted in Figure 7, the necklace will provide some raw data related to livestock health that can be detected on the necklace, such as heart rate, to subsequently undergo processing in the system. There are two responses that will be generated to the results of the data processing. If the farm animal is declared healthy, then the system will give a green status. On the contrary, if the farm animal is declared unhealthy, the system will give a red status. Based on this status, farmers can look at the health of their animals to determine their next course of action.

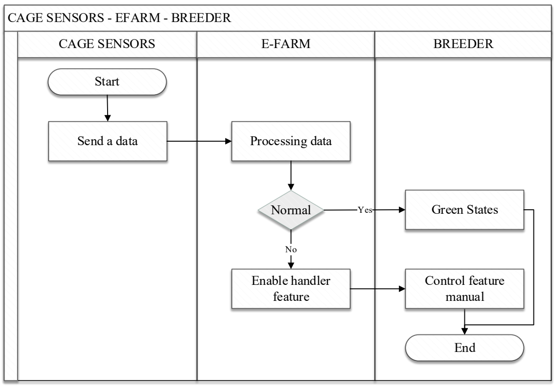



**Figure 8 - Cage Sensor Diagram Activity**

In the cage sensor diagram activity depicted in Figure 8, the sensor will send some raw data such as temperature, humidity, and ammonia gas contained in the cage. Furthermore, the system will process these data and provide a decision on whether the cage has normal or dangerous conditions. When the cage is normal, the farmer will get a green status, while if the condition is declared abnormal, the system will turn on the handling feature automatically. However, breeders can also manually control existing anticipation features.

### 3.3. Validation and Verification

The final stage of requirement engineering is validation. The validation stage is used to confirm that the requirements identified in the previous stages are in accordance with user requests by producing a clearer and more complete understanding and description of the requirements. Validation was carried out by the same respondents as the other stages. Each user is interviewed about existing application features in order to analyze and ensure that the functional requirements of the application being built are met. The results of the interview are shown in Table 7.

**Table 7 - Interview Result**

| No | **Needs** | Percentage | Decision |
|---|---|---|---|
| 1 | Farmers can monitor farm animals | 85.7% | Pass |
| 2 | Farmers can get information about animal health | 85.7% | Pass |
| 3 | Breeders can get information about the cage | 85.7% | Pass |
| 4 | Farmers can monitor the location of farm animals | 85.7% | Pass |
| 5 | Farmers and customers/users can interact by buying and selling | 100% | Pass |
| 6 | Breeders can perform detection of illegal objects | 14.3% | Fail |
| 7 | Breeders can provide food automatically | 14.3% | Fail |
| 8 | Breeders can get an analysis of the farm | 100% | Pass |
| 9 | Breeders can get financial statements | 85.7% | Pass |
| 10 | Breeders can manage product catalogs | 85.7% | Pass |
| 11 | Breeders can see articles related to animal husbandry | 100% | Pass |
| 12 | Farmers can choose and contact animal feed seller | 100% | Pass |
| 13 | Farmers can choose and contact waste managers | 57.1% | Pass |
| 14 | Farmers can select and contact Butcher Farm | 57.1% | Pass |

| No | **Needs** | Percentage | Decision |
|---|---|---|---|
| 15 | Farmers can choose and contact Clean Farm | 57.1% | Pass |
| 16 | Farmers can choose, contact, and consult a Veterinarian | 57.1% | Pass |
| 17 | Farmers can choose and contact the Courier | 57.1% | Pass |
| 18 | Breeders can choose and contact the seller of the drug | 71.4% | Pass |
| 19 | Breeders can choose and contact recycling providers | 57.1% | Pass |
| 20 | Users can view and modify profiles | 100% | Pass |
| 21 | User can top up, transfer, and view balances | 100% | Pass |
| 22 | User can see the market price | 100% | Pass |
| 23 | Users can consult with Customer Support | 100% | Pass |
| 24 | Users can get the latest information about app activity | 90% | Pass |
| 25 | Users can get information about the spread of animal diseases in each area | 10% | Fail |

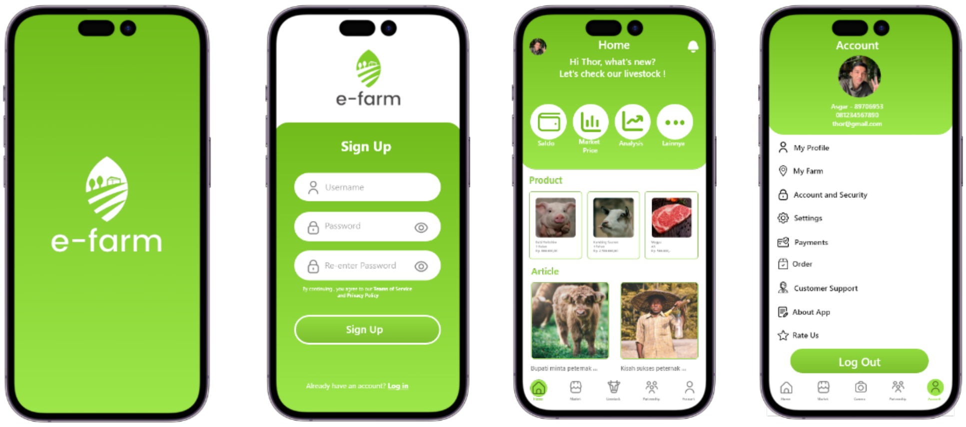
Furthermore, based on the results of interviews with the users shown in Table 7, a selection of needs will be made based on the percentage value of each feature. Features that have a percentage value below 50% will be removed and deemed unsuitable based on the interviews that have been conducted. There are 3 features that have a percentage value below 50%, namely: breeders can perform the detection of illegal objects; breeders can provide food automatically; and users can get information about the spread of animal diseases in each area. Therefore, these three features were removed because they were considered unsuitable or less functional for users. The final results of the final requirements can be seen in Table 8.

**Table 8 - Final Requirement**

| No | **Needs** |
|---|---|
| 1 | Farmers can monitor farm animals |
| 2 | Farmers can get information about animal health |
| 3 | Breeders can get information about the cage |
| 4 | Farmers can monitor the location of farm animals |
| 5 | Farmers and customers/users can interact by buying and selling |
| 6 | Breeders can get an analysis of the farm |
| 7 | Breeders can get financial statements |
| 8 | Breeders can manage product catalogs |
| 9 | Breeders can see articles related to animal husbandry |
| 10 | Farmers can choose and contact animal feed seller |
| 11 | Farmers can choose and contact waste managers |
| 12 | Farmers can select and contact Butcher Farm |
| 13 | Farmers can choose and contact Clean Farm |

| No | Needs |
|---|---|
| 14 | Farmers can choose, contact, and consult a Veterinarian |
| 15 | Farmers can choose and contact the Courier |
| 16 | Breeders can choose and contact the seller of the drug |
| 17 | Breeders can choose and contact recycling providers |
| 18 | Users can view and modify profiles |
| 19 | User can top up, transfer, and view balances |
| 20 | User can see the market price |
| 21 | Users can consult with Customer Support |
| 22 | Users can get the latest information about app activity |

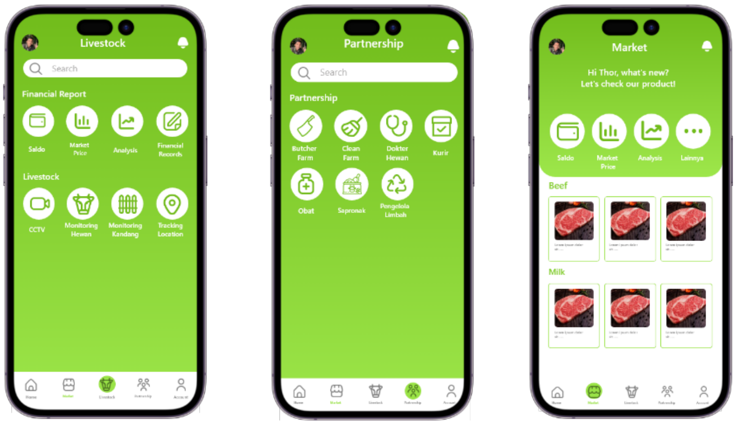
Based on the final requirements in Table 8, the next step is to create an interface description to make it easier for users to recognize how the desired feature will be implemented. A prototyping approach is used during the validation step to assist provide a visual representation [23]. To the user in understanding the features presented. Prototyping is a form of system modeling that is built and contains an overview of the layout of the features you want to create [24], or it can be said to be a basic overview of the platform. At this stage, prototyping is generated based on predefined needs. The prototype of this e-farm platform is shown in Figure 9, Figure 10, and Figure 11.




(a) Initial View (b) Login (c) Home (d) Profile

**Figure 9 - Breeder's Main Page Design**

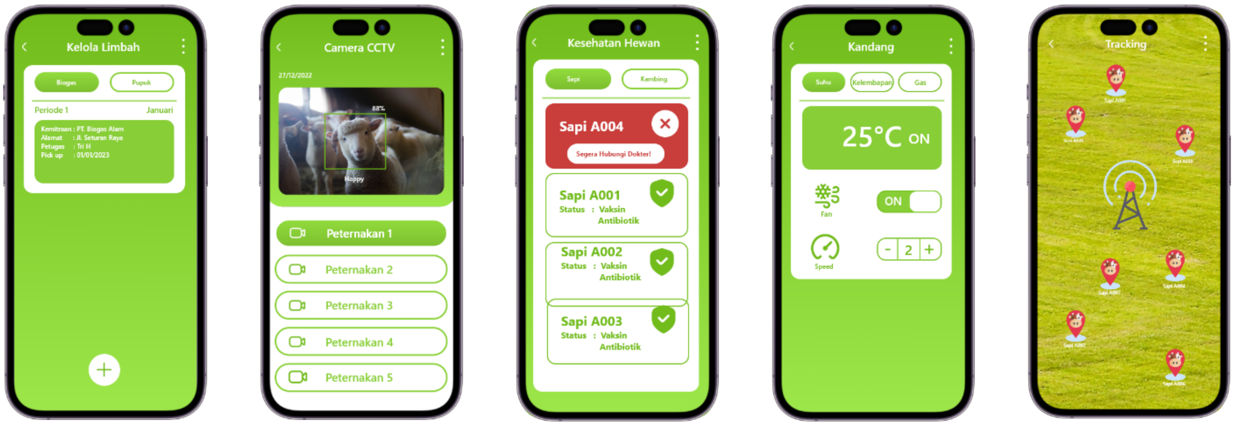
This view is a user interface for access rights as a breeder. Figure 9a is the initial display when accessing the platform, consisting of a plain white e-farm logo. Figure 9b is a display for signup, consisting of the e-farm logo at the top, and then below it is the words Sign Up. Then there is a text input Username, Password, and Re-enter Password, and at the very bottom, there is a click on the Signup button. Figure 9c is the initial display after logging in, and this display consists of an account icon on the top left and a notification icon on the top right. At the bottom, there are icons Balance, Market Price, Analysis, and Other. In the middle part, there are products sold, and below it is the latest article about animal husbandry. At the very bottom are Home, Market, Livestock, Partnership, and Account. Figure 9d is a display for the breeder's profile menu, consisting of Profile Photo, My Profile, My Farm, Account and Security, Setting, Payments, Order, Customer Support, About Apps, Rate Us, and the Log Out button.




(a) Live Stock Menu (b) Partnership Menu (c) Market Menu

**Figure 10 - Breeder Feature Interface Design**

Furthermore, the User interface in Figure 10a is a display of the details of the livestock menu, consisting of two parts. First, the Financial Report consists of "Balance," "Market Price, "Analysis, and "Financial Report," the second part is Livestock, consisting of CCTV, Animal Monitoring, Cage Monitoring, and Tracking Location. Figure 10b is a display of the details of the partnership menu consisting of Butcher Farm, Clean Farm, Veterinarian, Courier, "Medicine, Sapronak, and Waste Manager. The function of partnership is a feature to connect breeders with related partners, such as when farmers need butchers, then breeders can choose the Butcher Farm icon and various related partners, as mentioned in each icon. Figure 10c is the display for the market menu. The market menu function is used by customers to buy livestock products, and the market menu consists of Balance, Market Price, Analysis, and Others.




(a) Waste Management Menu (b) Smart Camera (c) IoT Necklace (d) Cage Sensors (e) Location Tracking

**Figure 11 - Waste Management Interface Design and IoT Features**

Figure 11a is a feature that allows farmers to place waste management orders so that livestock waste can be managed properly. With this feature, farmers can choose how the waste will be managed. There are two options, namely biogas and fertilizer. Each booking will be recorded according to the category as shown above. If you want to add orders, farmers can use the additional feature. Figure 11b is a smart livestock monitor. This feature can be obtained by using a smart camera installed by the farmer. In this feature, the resulting output is in the form of an image that includes a diagnosis of the animal's health condition based on the emotions that appear. Each result of the smart camera can be viewed alternately (depending on the number of cameras installed). Figure 11c is animal health monitoring. This feature is obtained from the installation of necklaces on farm animals. The output obtained is an animal health diagnosis based on the data obtained by the IoT necklace. If the farm animal is healthy, then its status will be green. On the contrary, if the farm animal is sick or needs treatment, its status will be red. Figure 11d monitors the condition of the cage. This feature is obtained with an IoT device in the form of a sensor that is able to identify temperature, humidity, and gas. If the temperature increases, then the fan can be turned on automatically to lower the temperature and vice versa. Furthermore, figure 11e Tracking Location Livestock. This feature is able to track the location of Livestock on a farm. So that if there is Livestock that runs away, this feature is able to detect the loss of this Livestock.

## 4. CONCLUSION

To make it easier for users to connect on a platform, problem specification must be completed. Requirement engineering was carried out by adopting the Loucopoulos and Karakostas Iterative Process Model method with the aim of being able to identify user needs in the process of developing the E-Farm platform. The Loucopoulos and Karakostas iterative requirements engineering process model has three stages: requirement elicitation, requirement specification, and requirement validation and verification. At the requirements elicitation stage, an interview process is carried out, and the results are documented at the requirements specification stage into a use case and activity diagram. Furthermore, in requirement validation and verification, three requirements are obtained that have a value below 50%, so that these requirements will not be used in the development process. The results of this study indicate the complexity of the needs required by users identified by using the Loucopoulos and Karakostas iterative methods. For further research, comparisons can be made using other approaches, and then system or application development is carried out based on the functional requirements that have been identified during the research to develop the application.

## REFERENCES


[1] V. Moysiadis, P. Sarigiannidis, V. Vitsas, and A. Khelifi, “Smart Farming in Europe,” *Comput. Sci. Rev.*, vol. 39, p. 100345, 2021, doi: 10.1016/j.cosrev.2020.100345.

[2] S. Neethirajan and B. Kemp, “Digital Livestock Farming,” *Sens. Bio-Sensing Res.*, vol. 32, no. December 2020, p. 100408, 2021, doi: 10.1016/j.sbsr.2021.100408.

[3] J. C. Pereira and R. de F. S. M. Russo, “Design thinking integrated in agile software development: A systematic literature review,” *Procedia Comput. Sci.*, vol. 138, pp. 775–782, 2018, doi: 10.1016/j.procs.2018.10.101.

[4] D. Pandey, U. Suman, and A. K. Ramani, “An effective requirement engineering process model for software development and requirements management,” *Proc. - 2nd Int. Conf. Adv. Recent Technol. Commun. Comput. ARTCom 2010*, pp. 287–291, 2010, doi: 10.1109/ARTCom.2010.24.

[5] D. H. Bangkalang, N. Setiyawati, R. Tanone, H. P. Chernovita, and Y. T. B. Tacoh, “A Requirement Engineering in Reporting and Counseling-Based Assistance Application for Victims of Violence Against Women,” *J. Ris. Inform.*, vol. 3, no. 4, pp. 311–318, 2021, doi: 10.34288/jri.v3i4.256.

[6] R. P. Soesanto, “Gamification for Student Achievement in Classroom: In Search of Requirement for Student Achievement Application,” *Int. J. Innov. Enterp. Syst.*, vol. 5, no. 02, pp. 90–99, 2021, doi: 10.25124/ijies.v5i02.138.

[7] I. Udousoro, “Effective Requirement Engineering Process Model in Software Engineering,” *Softw. Eng.*, vol. 8, no. 1, p. 1, 2020, doi: 10.11648/j.se.20200801.11.

[8] N. R. Mead and T. Stehney, “Security quality requirements engineering (SQUARE) methodology,” *SESS 2005 - Proc. 2005 Work. Softw. Eng. Secur. Syst. - Build. Trust. Appl.*, pp. 1–7, 2005, doi: 10.1145/1083200.1083214.

[9] J. Dick, E. Hull, and K. Jackson, *Requirements Engineering*. Springer, 2017.

[10] Badan Pusat Statistik, “Peternakan dalam Angka Tahun 2021,” *Badan Pus. Stat. Indones.*, pp. 4–8, 2021.

[11] H. Noor *et al.*, “Emerging Requirement Engineering Models: Identifying Challenges is Important and Providing Solutions is Even Better,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 11, pp. 646–656, 2021, doi: 10.14569/IJACSA.2021.0121174.

[12] J. M. Silva, R. Javales, and J. R. Silva, “A new Requirements Engineering approach for Manufacturing based on Petri Nets,” *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 97–102, 2019, doi: 10.1016/j.ifacol.2019.10.006.

[13] I. Sommerville, *Software Engineering*. 2011. doi: 10.1007/3-540-49477-4_16.

[14] A. Nurrifqhi, S. Widowati, and M. Imrona, “Implementasi User Centered Requirements Engineering pada Perancangan Aplikasi Panduan Informasi Pertanian untuk Petani,” *J. Comput.*, vol. 4, no. 2, pp. 9–20, 2019, doi: 10.21108/indojc.2019.4.2.256.

[15] R. Kusuma and N. Setiyawati, “Rekayasa Kebutuhan Aplikasi Learning Management System ( LMS ) Berbasis Web Menggunakan Kotonya and Sommerville Process Model,” no. 5, pp. 50–59, 2022.

[16] M. A. Nadeem and S. U. J. Lee, “Machine learning evaluation of the requirement engineering process models for cloud computing and security issues,” *Appl. Sci.*, vol. 10, no. 17, pp. 1–13, 2020, doi: 10.3390/app10175851.

[17] P. Loucopoulos and B. Karakostas, *System Requirements Engineering Process*. 1996. doi: 10.1002/j.2334-5837.1996.tb02037.x.

[18] A. Y. Aleryani, “The Impact of the User Experience (UX) on the Quality of the Requirements Elicitation,” *Int. J. Digit. Inf. Wirel. Commun.*, vol. 10, no. 1, pp. 1–9, 2020, doi: 10.17781/p002628.

[19] J. Li *et al.*, “Attributes-Based Decision Making for Selection of Requirement Elicitation Techniques Using the Analytic Network Process,” *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/2156023.

[20] H. F. Martins, A. C. de Oliveira, E. D. Canedo, R. A. D. Kosloski, R. Á. Paldês, and E. C. Oliveira, “Design thinking: Challenges for software requirements elicitation,” *Inf.*, vol. 10, no. 12, pp. 1–27, 2019, doi: 10.3390/info10120371.

[21] N. Ramadan and S. Megahed, “Requirements Engineering in Scrum Framework,” *Int. J. Comput. Appl.*, vol. 149, no. 8, pp. 24–29, 2016, doi: 10.5120/ijca2016911530.

[22] S. Saito, Y. Iimura, A. K. Massey, and A. I. Anton, “How Much Undocumented Knowledge is there in Agile Software Development?: Case Study on Industrial Project Using Issue Tracking System and Version Control System,” *Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. RE 2017*, pp. 194–203, 2017, doi: 10.1109/RE.2017.33.

[23] I. Garcia, C. Pacheco, A. León, and J. A. Calvo-Manzano, “Experiences of using a game for improving learning in software requirements elicitation,” *Comput. Appl. Eng. Educ.*, vol. 27, no. 1, pp. 249–265, 2019, doi: 10.1002/cae.22072.

[24] S. W. Ali, Q. A. Ahmed, and I. Shafi, “Process to enhance the quality of software requirement specification document,” *2018 Int. Conf. Eng. Emerg. Technol. ICEET 2018*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/ICEET1.2018.8338619.

[25] R. S. Wahono, “Menyegarkan Kembali Pemahaman tentang Requirement Engineering,” 2006. https://romisatriawahono.net/2006/04/29/menyegarkan-kembali-pemahaman-tentang-requirement-engineering/ (accessed Jan. 09, 2023).

[26] M. Rachmaniah, *Pengembangan Perangkat Lunak dan Sistem Informasi*. Bogor: Bogor: IPB Press, 2018.

[27] R. A. Sukamto and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Bandung: Informatika, 2016.

[28] C. Jane Radatz, “IEEE Standard Glossary of Software Engineering Terminology,” *Office*, vol. 121990, no. 1, p. 1, 1990, doi: 10.1109/IEEESTD.1990.101064.

[29] A. Speck, S. Witt, S. Feja, A. Lotyzc, and E. Pulvermüller, “Framework for business process verification,” *Lect. Notes Bus. Inf. Process.*, vol. 87 LNBIP, pp. 50–61, 2011, doi: 10.1007/978-3-642-21863-7_5.


*liana.trihardianingsih@students.amikom.ac.id 

[30] S. Maalem and N. Zarour, "Challenge of validation in requirements engineering," *J. Innov. Digit. Ecosyst.*, vol. 3, no. 1, pp. 15–21, 2016, doi: 10.1016/j.jides.2016.05.001.

[31] S. K. R. Peddireddy and S. R. Nidamanuri, "Requirements Validation Techniques and Factors Influencing them," *Master Sci. Softw. Eng.*, no. February, 2021, [Online]. Available: www.bth.se

[32] I. Akhigbe, K. Munir, O. Akinade, L. Akanbi, and L. O. Oyedele, "Iot technologies for livestock management: A review of present status, opportunities, and future trends bernard," *Big Data Cogn. Comput.*, vol. 5, no. 1, 2021, doi: 10.3390/bdcc5010010.

[33] H. Anas, M. Ilyas, Q. Tariq, and M. Hummayun, "Requirements Validation Techniques: An Empirical Study," *Int. J. Comput. Appl.*, vol. 148, no. 14, pp. 5–10, 2016, doi: 10.5120/ijca2016910911.

[34] C. A. Lauff, D. Kotys-Schwartz, and M. E. Rentschler, "What is a prototype? what are the roles of prototypes in companies?," *J. Mech. Des. Trans. ASME*, vol. 140, no. 6, 2018, doi: 10.1115/1.4039340.