Online version at https://journal.lenterailmu.com/index.php/josapen

JOSAPEN

E-ISSN: 3031-2272 (Online)

Web-Based Monitoring and Management System for Livestock Operations

Ananda Ade F

Faculty of Computer Science, IGM University, Palembang, Indonesia

ARTICLE INFO

Article history:
Received 02 June 2025
Revised 03 Aug 2025
Accepted 03 Aug 2025

Keywords:
Livestock Monitoring
Web-Based System
Agile Development

ABSTRACT

Monitoring livestock health and productivity is a critical aspect of agricultural management, yet many organizations still rely on manual or disconnected systems that lead to inefficiencies, delayed reporting, and lack of real-time insight. These challenges hinder timely decision-making, especially in identifying diseases, recording births and deaths, and maintaining accurate data. To address this gap, this study introduces a Web-Based Livestock Monitoring System designed to streamline data entry, enable centralized access, and support role-based interaction for Admins, Livestock Staff, and the Head of Department. The system was developed using a structured web application model, with core features including user authentication, livestock categorization, health monitoring, and monthly report generation. A black-box testing approach was used to evaluate its functionality from the user's perspective without delving into internal code logic. Testing results showed that all features—such as login validation, data input, and report generation—operated successfully and provided the expected outputs. This system offers a practical solution for modernizing livestock monitoring processes, enhancing efficiency, data accuracy, and communication across roles. It contributes to more effective livestock management by ensuring that health data is up-to-date and accessible, allowing stakeholders to make informed decisions and respond proactively to issues in the field.

1. Introduction

In recent years, technology has played an increasingly pivotal role in modernizing the agricultural sector. Livestock farming, in particular, has witnessed a growing demand for digital solutions that can support farmers in improving efficiency, productivity, and animal welfare [1]-[3]. Traditional methods of livestock monitoring—such as manual record-keeping or on-site inspections—often lead to inaccuracies, delayed responses to

* *Corresponding author:* Ananda Ade F
email: 2021.21.0114@students.igm.ac.id

health issues, and inefficiencies in daily operations. As the global population grows and the demand for high-quality animal products rises, there is a strong need to integrate smart systems into livestock management practices [4], [5].

Despite these advancements, many farmers, especially those in rural or developing regions, still face significant challenges in managing livestock effectively. Common problems include the lack of real-time health data, limited access to expert veterinary support, and the inability to track livestock movements or conditions remotely. These limitations can lead to economic losses and compromised animal well-being. Furthermore, the absence of centralized data systems makes it difficult for farm owners to make informed decisions quickly, especially in emergencies. These challenges highlight the necessity of an accessible and user-friendly digital solution.

To address these issues, this study proposes the development of a web-based livestock monitoring system utilizing the SDLC development methodology. The method was chosen for its iterative and flexible nature, allowing continuous improvement based on stakeholder feedback throughout the development process. The system is designed to collect, store, and display real-time data on livestock conditions such as temperature, location, and movement, using sensors and web technologies. With a user-centered design and a responsive web interface, the platform ensures that farmers and livestock managers can monitor their herds from anywhere using a computer or smartphone.

The main contribution of this system lies in its ability to empower farmers with timely and accurate information that supports better decision-making and enhances animal welfare. By leveraging web technology and model development, the system ensures adaptability, scalability, and ease of use. Additionally, it bridges the digital gap for small to medium-scale farmers by providing a cost-effective and practical tool for livestock monitoring. Ultimately, this project aims to promote smarter, data-driven farming practices that are sustainable and aligned with the future of agriculture.

2. Methods

The system development stage used is the SDLC model. The implementation model is a method that has the characteristic that each result in the model must be completed first before proceeding to the next phase. The Model is a software development methodology that follows a linear and structured flow [6], [7], [8]. It consists of a series of phases that must be completed sequentially, and each phase is dependent on the completion of the previous phase. Following are some of the main phases in the model:

a. Analysis: The stage where system requirements are gathered and thoroughly understood. It involves interaction with users and stakeholders to define functional and non-functional requirements [9], [10].
b. Design: After the requirements are collected, the next step is to design the system architecture. This includes designing the system structure, identifying algorithms, and preparing the necessary technical specifications [11]-[13].
c. Coding: This stage involves coding the software according to the specifications created at the design stage. The development team creates code based on the approved design.
d. Testing: After implementation, the system is tested to ensure that all requirements have been met and that there are no significant bugs or errors. These tests include functional, performance, and security tests [14], [15].
e. Delivery/Implementation: Once the system passes all the tests, it is ready to be implemented and released into a production environment or used by end users.

4. Result and Discussion

The following diagram (Figure 1) presents the proposed workflow of the Web-Based Livestock Monitoring and Management System. It visually outlines the roles and responsibilities of key users—Admin, Livestock Staff, and the Head of Department—within the system. Each user interacts with the platform through a secure login and validation process, followed by access to specific features based on their role.

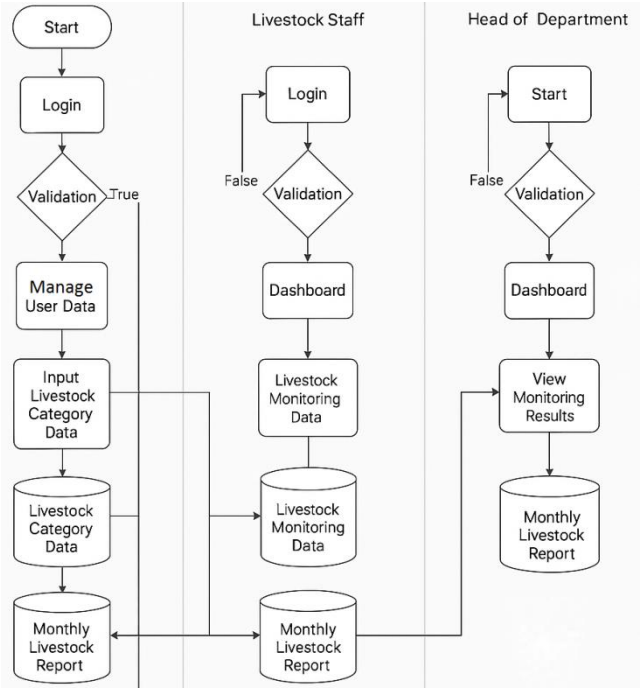

Figure 1 – The Proposed Flowchart Diagram

The proposed diagram in Figure 1 shows the workflow of a Web-Based Livestock Monitoring System by dividing responsibilities among three key user roles: Admin, Livestock Officer, and Department Head. Each role begins with a secure login and validation process to ensure access rights are verified before proceeding. If the login is valid, users are directed to a dashboard that serves as a central hub for their respective tasks. This initial section emphasizes the system's focus on user authentication and role-based access control, ensuring that only authorized personnel can view or manage specific data.

For the Admin, the dashboard offers extensive control over system data. They are responsible for managing user accounts, setting up livestock categories, and entering both livestock records and monitoring data. These inputs serve as the foundation for generating monthly livestock reports, which are shared across the system. This centralized data management ensures consistency and accuracy, allowing the system to maintain up-to-date and reliable livestock information. Admins play a pivotal role in setting up and maintaining the data structure that supports the operations of other users.

The Livestock Officer focuses on monitoring activities in the field. After logging in and accessing their dashboard, they can input real-time livestock monitoring data and contribute to the monthly livestock reports. These reports are then made available to the Department Head, who, upon successful login, can view monitoring results and access summarized reports. This structured flow ensures that monitoring efforts are documented at the operational level and transparently reviewed at the management level. Overall, the system fosters collaboration across different roles while supporting data-driven decision-making in livestock management.

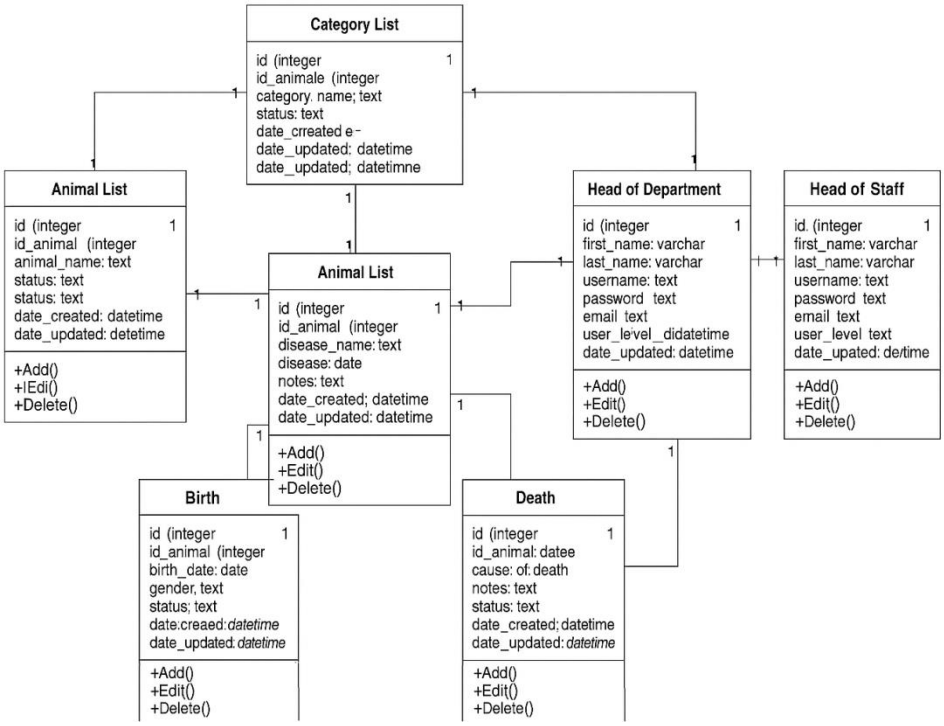

Figure 2 - The Proposed Class Diagram

Figure 2 shows the proposed class diagram. The class diagram illustrates the structural design of the Web-Based Livestock Monitoring System, showcasing the key entities involved and the relationships between them. At the center of the structure is the Animal List, which stores detailed information about each livestock animal, including its category (linked via the Category List) and serves as the connection point for related records such as Birth, Death, and Disease. Each of these lifecycle-related classes is tied to a specific animal and includes essential fields like dates, descriptions, and status updates. This modular structure helps ensure that each event or condition in an animal's life is recorded and traceable, enabling accurate reporting and tracking of livestock health and productivity.

On the user side, the system supports three main roles: Admin, Livestock Staff, and Head of Department. Each user class includes standard fields such as name, username, password, email, and timestamps, along with methods to add, edit, and delete records. The Admin is typically responsible for managing users and inputting category and animal data, while Livestock Staff can enter field data such as disease cases or births. The Head of Department can view summarized reports and monitor overall livestock status. The relationships in the diagram clearly define which users can interact with which types of data, supporting a secure, role-based system design that reflects real-world workflows.

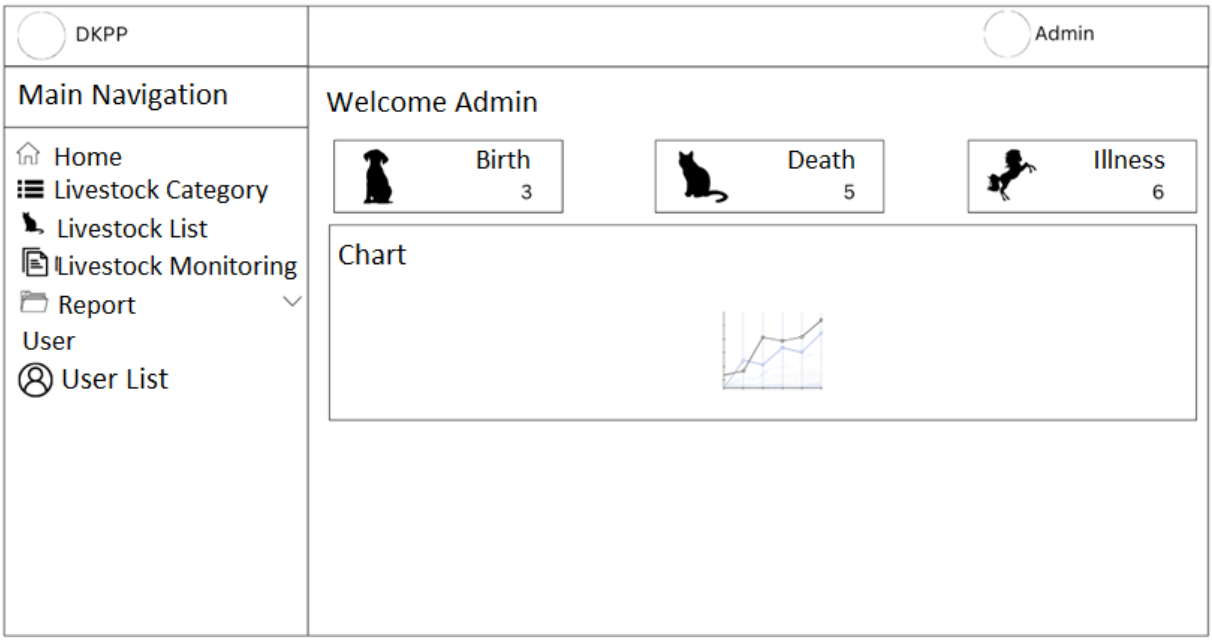

Figure 3 – The Homepage

Figure 3 shows the main page of the proposed system. The main page of the proposed Web-Based Livestock Monitoring System serves as a centralized dashboard for the Admin user. On the left side, the Main Navigation menu provides quick access to essential modules, including Home, Livestock Category, Livestock List, Livestock Monitoring, Report, and User List. Each icon is accompanied by intuitive visual cues, making the interface user-friendly and easy to navigate. This menu ensures that Admins can efficiently manage livestock data, monitor animal health, and oversee system users all in one place.

In the main content area, the system greets the Admin with a summary dashboard that includes key livestock statistics: the number of births, deaths, and illness cases, each represented with animal-themed icons and current counts. Below this, a placeholder for a chart or graph provides visual analytics, likely showing trends over time related to livestock health events. This layout allows the Admin to gain a quick overview of the system's status at a glance, promoting fast, informed decision-making based on real-time data. Table 1 shows the blackbox testing result with several feature tested.

Table 1 – The Blackbox Testing Result

Test Case ID	Feature Tested	Test Description	Input	Expected Output	Actual Output	Status
TC001	Login	Login with valid credentials	Username: admin, Password: admin123	Redirect to dashboard	Redirect to dashboard	Pass
TC002	Login	Login with invalid credentials	Username: admin, Password: wrongpass	Show error message "Invalid credentials"	Error message displayed	Pass
TC003	Add Livestock Category	Add a new livestock category	Category Name: Cattle	Success message and category appears in list	Category added successfully	Pass
TC004	Add Animal	Add a new animal record	Animal Name: Cow A, Category: Cattle	Animal appears in the Livestock List	Animal listed	Pass
TC005	Record Birth	Input birth record for a specific animal	Animal ID: 5, Birth Date: 01/01/2025	Birth record saved and shown in monitoring	Record added successfully	Pass
TC006	Record Illness	Input illness data for an animal	Animal ID: 5, Illness: Flu	Illness recorded and reflected in	Illness listed	Pass

				dashboard		
TC007	Generate Report	Generate monthly report	Month: July 2025	Downloadable report generated	Report generated	Pass
TC008	Navigation Access	Navigate to User List page	Click “User List”	User list table displayed	Table appears	Pass
TC009	Input Validation	Submit empty form for adding animal	(Empty fields)	Show required field validation messages	Validation triggered	Pass
TC010	Logout	Admin logs out from system	Click “Logout”	Redirected to login page	Redirected to login page	Pass

The black-box testing results above show that the core functionalities of the Web-Based Livestock Monitoring System are performing as expected. Key actions such as logging in with both valid and invalid credentials work correctly, ensuring secure access control. Features like adding livestock categories, registering new animals, and recording key events such as births and illnesses all passed their tests. This indicates that the system can successfully handle user inputs and display appropriate responses without exposing or relying on its internal code. Validation mechanisms are also functioning well—for example, when forms are submitted without required fields, the system correctly prompts the user to fill in the missing information.

Additionally, user interface elements such as the main navigation menu and report generation tools are working smoothly. Test cases related to viewing the user list, logging out, and generating monthly reports all passed, which confirms that users can interact with the system efficiently from start to finish. These test outcomes demonstrate that the system is not only functionally stable but also user-friendly and ready for real-world use, supporting essential livestock monitoring tasks with clarity and reliability.

5. Conclusion

The development and testing of the Web-Based Livestock Monitoring System demonstrate that the application effectively supports livestock data management through secure user access, structured record-keeping, and real-time monitoring features. The system allows Admins to manage users and livestock categories, while Livestock Staff can input field data such as births, illnesses, and deaths—information that is then accessible to the Head of Department for oversight and reporting. Based on black-box testing, all major functionalities performed successfully, validating the system's readiness for deployment in real-world agricultural or governmental settings. This solution not only enhances operational efficiency but also promotes informed decision-making in livestock health management.

REFERENCES

[1] G. Papadopoulos *et al.*, “Economic and environmental benefits of digital agricultural technological solutions in livestock farming: A review,” *Smart Agric. Technol.*, vol. 10, no. January, p. 100783, 2025, doi: https://doi.org/10.1016/j.atech.2025.100783.

[2] H. Görge, I. Dittrich, N. Kemper, and J. Krieter, “Solutions and prospects for digital documentation of treatment data on-farm,” *Smart Agric. Technol.*, vol. 5, no. August, 2023, doi: https://doi.org/10.1016/j.atech.2023.100299.

[3] A. Shahbaz, W. Zhang, and M. Smith, “A novel system for automated continuous on-farm assessment of digital dermatitis using artificial intelligence,” *Smart Agric. Technol.*, vol. 12, no. July, p. 101178, 2025, doi: https://doi.org/10.1016/j.atech.2025.101178.

[4] M. A. Al-ghouti, M. Khan, M. S. Nasser, K. Al Saad, and O. O. N. E. Heng, “Jo ur na l P re of,” *J. Clean. Prod.*, p. 125039, 2020, doi: https://doi.org/10.1016/j.csag.2025.100069.

[5] T. Ohashi, M. Saijo, K. Suzuki, and S. Arafuka, “From conservatism to innovation: The sequential and iterative process of smart livestock technology adoption in Japanese small-farm systems,” *Technol. Forecast. Soc. Change*, vol. 208, no. June, p. 123692, 2024, doi: https://doi.org/10.1016/j.techfore.2024.123692.

[6] K. D. Prasetya, Suharjito, and D. Pratama, “Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-Party Application Development,” *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 103–111, 2021, doi: https://doi.org/10.1016/j.procs.2020.12.014.

[7] T. Thesing, C. Feldmann, and M. Burchardt, “Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project,” *Procedia Comput. Sci.*, vol. 181, pp. 746–756, 2021, doi: https://doi.org/10.1016/j.procs.2021.01.227.

[8] A. A. S. Gunawan, B. Clemons, I. F. Halim, K. Anderson, and M. P. Adianti, “Development of e-butler: Introduction of robot system in hospitality with mobile application,” *Procedia Comput. Sci.*, vol. 216, no. 2019, pp. 67–76, 2022, doi: https://doi.org/10.1016/j.procs.2022.12.112.

[9] R. Dwivedi, S. Nerur, and V. Balijepally, “Exploring artificial intelligence and big data scholarship in information systems: A citation, bibliographic coupling, and co-word analysis,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 2, 2023, doi: https://doi.org/10.1016/j.jjimei.2023.100185.

[10] D. Ding *et al.*, “Data collection and information security analysis in sports teaching system based on intelligent sensor,” *Meas. Sensors*, vol. 28, no. June, 2023, doi:

https://doi.org/10.1016/j.measen.2023.100854.

[11] S. Park, J. yoon Lee, and J. Lee, "AI system architecture design methodology based on IMO (Input-AI Model-Output) structure for successful AI adoption in organizations," *Data Knowl. Eng.*, vol. 150, no. September 2023, 2024, doi: https://doi.org/10.1016/j.datak.2023.102264.

[12] W. Lu *et al.*, "Design for the Emergency Command Information System Architecture of Ocean Oil Spill," *Aquat. Procedia*, vol. 3, no. 22, pp. 41–49, 2015, doi: https://doi.org/10.1016/j.aqpro.2015.02.226.

[13] G. A. Attwell, K. E. Bennin, and B. Tekinerdogan, "Reference architecture design for computer-based speech therapy systems," *Comput. Speech Lang.*, vol. 78, no. August 2022, p. 101465, 2023, doi: https://doi.org/10.1016/j.csl.2022.101465.

[14] N. A. Bhaskaran and M. Durairaj, "Highlighting bugs in software development codes using SDPET for enhancing security," *Meas. Sensors*, vol. 30, no. October, p. 100930, 2023, doi: https://doi.org/10.1016/j.measen.2023.100930.

[15] P. Krawiec, R. Janowski, J. Mongay Batalla, E. Andrukiewicz, W. Latoszek, and C. X. Mavromoustakis, "On providing multi-level security assurance based on Common Criteria for O-RAN mobile network equipment. A test case: O-RAN Distributed Unit," *Comput. Secur.*, vol. 150, no. July 2024, 2025, doi: https://doi.org/10.1016/j.cose.2024.104271.