



Design and Construction of Web-Based and Mobile Cattle Feed Management System and Cattle Health Monitoring System Using the Prototyping Method

Hernawati Susanti Samosir^{1,*}, Cynthia Deborah Nababan¹, Bella Wahmilyana Asril¹, Tantri Harianti Silaen¹, Oktavia Letisya Simatupang¹, Keren Simanjuntak¹, Christian Jhon Pranata Panjaitan¹, Hasan Sinaga¹, Nita Herlinda Kurnyawati Simangunsong¹, & Sahat Pandapotan Nainggolan²

¹Faculty of Vocational Studies, Del Institute of Technology, Indonesia

²Division of Mathematics and Physical Sciences, Kanazawa University, Japan

ARTICLE INFO

Article History:

Received: 23 July 2026

Accepted: 11 September 2026

Published: 14 October 2026

Keywords:

Information system; Cattle feed management; Cattle health monitoring; Prototyping methodology; Web and mobile information system.

*Corresponding Author:

hernawati@del.ac.id

ABSTRACT

Livestock management operations are often inefficient and error-prone because agricultural institutions frequently rely on manual recording of cattle feed and health data. This study aims to develop an integrated web and mobile information system for cattle feed management and health monitoring, implemented for the Herbal and Horticultural Science and Technology Park (TSTH2), Pollung, Humbang Hasundutan Regency, North Sumatra, Indonesia. The system was designed using a prototyping methodology through three iterative cycles, each involving stakeholder communication, rapid planning, design, modeling, prototype construction, and user evaluation. The resulting system includes 68 functional components: 24 for feed management and 44 for health monitoring. The web platform was implemented using React.js for the front end, with Express.js serving as the back end for the feed management subsystem and Django REST Framework for the health monitoring subsystem. In contrast, Flutter was used for the mobile platform of both subsystems. Functional testing used black-box testing with a 100% success rate on both devices, and user acceptance testing had a satisfaction score of 76.96% (good/acceptable). This system successfully replaced previously manual steps in rural livestock management, especially in feed scheduling, consumption tracking, and health monitoring. To improve the system's intelligence, future research should add flexible scheduling algorithms and AI-based decision support.

To cite this article:

Samosir, H. S., Nababan, C. D., Asril, B. W., Silaen, T. H., Simatupang, O. L., Simanjuntak, K., Panjaitan, C. J. P., Sinaga, H., Simangunsong, N. H. K., & Nainggolan, S. P. (2026). Design and Construction of Web-Based and Mobile Cattle Feed Management System and Cattle Health Monitoring System Using the Prototyping Method. *Research in Education, Technology, and Multiculture*, 5(2), 93-108. doi: <https://doi.org/10.61436/rietm/v5i4.pp394-406>

This is an open access article under the CC BY SA licence (<https://creativecommons.org/licenses/by-sa/4.0>).

■ INTRODUCTION

The Herbal and Horticultural Science and Technology Park (TSTH2) in Pollung, Humbang Hasundutan Regency, North Sumatra, has recently expanded its scope to include dairy cattle farming as part of a broader effort to support increased national milk production. According to data from the Central Statistics Agency (BPS), Indonesia's milk production reached 946,912.81 tons in 2020, then rose sharply to 1,892,776.34 tons in 2021. However, production declined sharply by 56.45% to 824,273.18 tons in 2022 and continued to fall to 787,374.38 tons in 2023 (Badan Pusat Statistik, 2023). This decline in the national dairy cattle population shows that the dairy sector faces serious challenges. The

national dairy cattle population is affected by poor feed management and insufficient health monitoring. These factors are essential for dairy cattle to produce high milk yields. The dairy sector must address these issues to support growth in the dairy cattle population and increase milk production.

To assess the operational workflows currently employed at TSTH2, qualitative inquiries were conducted in February 2025 with the farm management team, specifically Mr. Agustavo Pacheco, Mr. Yeppi, and Mr. Irvandi Sihombing. These interviews provided empirical evidence regarding the systemic vulnerabilities inherent in manual paper-based record-keeping. The management team corroborated that the absence of a structured

digital system leads to three critical operational bottlenecks: first, inconsistent data entry and redundant manual logs which result in inventory inaccuracies and nutritional waste; second, fragmented health records that cause delayed reporting of cattle symptoms, thereby hindering timely medical interventions; and third, cumulative delays in data aggregation that impair real-time decision-making for farm productivity. The informants emphasized that these issues are endemic to their current informal recording practices. Consequently, the lack of a formal digital error logbook is not a mere oversight but a direct consequence of the paper-based system itself, which heightens the urgency of implementing a digitized platform to ensure data integrity and real-time operational oversight. TSTH2 farm plans to raise Girolando cattle, a crossbred variety noted for its tolerance to hot climates and milk production of 12 to 18 liters per day (Pasaribu et al., 2015). Achieving this production target therefore requires a structured, digitalized system for feed scheduling and health monitoring before the cattle arrive at TSTH2.

Meeting this target therefore requires TSTH2 to implement a structured, digitalized feed scheduling and health monitoring system before the cattle arrive. The digitalization of cattle feed management has been examined in several prior research works. Previous research on cattle farming technology falls into four main approaches. First, the conceptual/strategic group includes literature reviews, proposed policy directions, or field-based interventions without system implementation, such as Khairi et al. (2025) and Putra (2025), who reviewed feed automation and health monitoring. Besides that, Zaharieva & Dunchev (2026) and Vigneswari et al. (2021) examined the concept of Precision Livestock Farming. Moreover, Tanzil et al. (2025) and Pakpahan et al. (2025) proposed farmer digitalization strategies without software artifacts. Zullaikah et al. (2022) improved feed quality through aerobic fermentation of corn and rice waste. Furthermore, Qisthon et al. (2024), Christi et al. (2025), and Gifari et al. (2025) conducted counseling and training programs on silage-making and cattle breeding management that significantly improved farmers' knowledge scores.

Second, the IoT-based group, which predominantly addresses livestock health and environment through temperature sensors, RFID, GPS, and fuzzy logic, such as Kuldharan et al. (2023), Ariff et al. (2014), Okokpuijie et al. (2023), Arora et al. (2024), Sasongko (2026), Rahmawati et al. (2024) and Althafullayya & Aryani (2024) reviewed the

use of nutrition sensors, RFID, IoT, and big data analytics for real-time individual livestock monitoring. Mubarok et al. (2025) focused on cattle feeding. Third, the web-based development group consistently addresses only feeding and data management, such as Nosa & Sulistiani (2025), Hassim et al. (2011), and Alawiyah & Pamungkas (2026). Moreover, Chaniago et al. (2019) developed a PHP/CodeIgniter/MySQL-based system to record dairy cattle care, milk yield, and farmer finances. Kumala et al. (2018) built a PHP/MySQL-based system to replace manual recording of cattle growth data. However, none addressed cattle health monitoring. Fourth, only a small percentage of the mobile-based groups, such as Astrawijaya & Asriningtias (2025), Thannithi et al. (2024), and Sugiharti et al. (2019), address a single aspect, such as feeding by Setiawan et al. (2025) or health by Darvesh et al. (2023), but not both concurrently within a single system.

Based on the existing literature, a common limitation emerges: current systems consistently handle only isolated fragments of dairy farm management rather than the process as a whole. In general, mobile-based platforms cover only broad administrative tasks or a single element, web-based platforms focus mainly on feed management, and IoT-based platforms are usually limited to health monitoring functions. Therefore, a significant gap remains in the current body of research: an integrated platform that combines feed scheduling, consumption monitoring, and health tracking across both web and mobile environments, and that has been validated through real-world deployment on a dairy cattle farm, has yet to be developed.

Based on this gap, the research question guiding this study is: How can a web- and mobile-based cattle health monitoring system and cattle feed management system be developed and implemented using the prototyping method? Therefore, this study aims to answer this question by designing and implementing an integrated information system that unifies feed management and health monitoring functions across web and mobile platforms, tailored to TSTH2's operational needs. This research used a prototyping methodology. This method supports iterative refinement through direct, continuous engagement with farm stakeholders. It also ensures the resulting system accurately reflects real field conditions rather than assumptions made in isolation from actual farm operations.

Through this approach, the study seeks to demonstrate not only the technical feasibility of an integrated web-mobile system for dairy

cattle management, but also its practical applicability in addressing the inefficiencies inherent in manual recording practices, ultimately supporting more structured and reliable decision-making in feed scheduling, consumption tracking, and cattle health monitoring. This unresolved gap forms the basis of this study's contribution.

METHOD

This research used prototyping to develop software. Other projects have also found prototyping an effective approach to software development (Samsumar et al., 2023; Cicimby et al., 2021; Primada et al., 2021). These examples show that prototyping suits system development. It works well when user needs are practical, changing, and field-dependent. The prototyping method supports this approach.

Since the Girolando cattle had not yet arrived at TSTH2 during the testing period, this was addressed by testing the system on an existing dairy cattle farm, PT. Cifa Indonesia, located in Simorangkir Julu, Siatas Barita Subdistrict, North Tapanuli Regency. Real-farm testing confirms that feeding scheduling,

consumption tracking, and health monitoring workflows work correctly with actual livestock data. This shows the system's potential for successful operation at TSTH2 when the cattle arrive.

System Architecture

Figure 1 illustrates the architecture of a web and mobile-based cattle health monitoring and feed management system, comprising three main layers: client, server, and database. On the client side, the application used React.js for the web version and Flutter for the mobile version, communicating with the server via HTTP requests and responses. The system integrates two subsystems, each with its own backend service. The cattle feed management subsystem uses Express.js as its backend, while the cattle health monitoring subsystem uses Django REST Framework as its backend. Both subsystems use React.js for the web platform and Flutter for the mobile platform, and MySQL as the database. The backend services handle application logic, process data requests, and provide RESTful APIs for communication with the client applications.

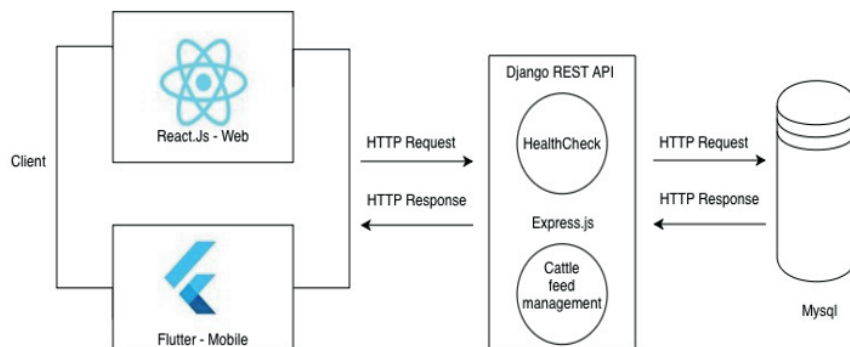


Figure 1. System architecture of a web and mobile-based cattle health monitoring and feed management system

Prototyping Methodology

Prototyping is a software engineering approach used when software requirements have not been defined in detail or remain vague. In such situations, the prototyping approach enables developers and stakeholders to collaboratively explore and understand the system to be built by creating an initial, interim, or partial version of the software. The process begins with communication between developers and stakeholders to establish the software's general objectives, followed by a "quick design" phase that focuses on visual aspects, such as the user interface. This initial version is then built and presented to gather feedback. The process is iterative, with feedback used to refine requirements and incrementally improve the prototype

(Pressman, 2010).

Key characteristics of the prototyping approach include accommodating changing requirements during development, giving users an early understanding of the system, and accelerating the identification of essential features through hands-on demonstrations. However, this approach also carries risks, such as stakeholders mistaking the prototype for the final product or developers becoming locked into technologies or temporary solutions that are suboptimal in the long run. Therefore, it is crucial to emphasize that a prototype is not the final version of the software, but a tool to help clarify and define requirements.

In software projects, prototyping is particularly valuable when development time is limited, but user input is critical, such as in

developing user interfaces or systems that require intensive interaction. Thus, prototyping effectively addresses uncertainty and the dynamic nature of user requirements while maintaining flexibility throughout the software engineering process. This approach also supports an evolutionary software development process, allowing the system to grow and be refined incrementally over time.

Research Methodology

The research methodology consisted of problem identification, literature review, communication, quick planning, quick design modeling, prototype construction, deployment and delivery, feedback, system implementation, testing, and deployment to the server.

Problem Identification

TSTH2 was first studied to understand the major pain points in identifying cattle health status and the feeding schedule, which was manually recorded in paper-based records. The process was slow and inconsistent, leading to frequent human error and limited visibility into real-time feeding activity and upcoming animal health concerns. Early health problems were especially difficult to detect because information was not shared. These limitations highlighted the need for a computer-based system with simple, accurate identification that allows farm staff to record and monitor livestock well-being and feed data.

Literature Review

A literature review guided system development by confirming the practical effectiveness of the chosen methods, focusing on three areas: livestock health assessment (body temperature, appetite, behavior), feed management (scheduling and consumption tracking), and non-AI/machine-learning-based livestock management systems. This foundation provided the present study with its significance, as the system's design reflects real-world practices identified through the review.

Communication

During a visit to TSTH2, farm managers Mr. Agustavo Pacheco, Mr. Yeppi, and Mr. Irvandi Sihombing, along with IT staff, were consulted by the development team to learn about the daily challenges experienced by farm personnel. The system's functional requirements and specifications were informed by these findings, which subsequently guided the planning and design phases.

Quick Plan

With requirements gathered, the team drafted a work plan covering requirements analysis, risk identification, resource allocation, scheduling, and stage objectives, which guided three development cycles for the feed management and health monitoring subsystems and ensured both were completed successfully.

Quick Design Modelling

After starting with an early version of the system, the user interface was designed in Figma. Case diagrams and scenario illustrations were also created to demonstrate how the system would function. Initially, BPMN diagrams and ERDs were developed to map out the system's functionality, along with a broader diagram illustrating the overall architecture. The initial version was relatively simple: users could log health data and schedule meals. However, it was treated as a work in progress; the diagrams were revised several times as user feedback was received, and features were refined to ensure health data logging and meal scheduling were effective and easy to use.

The workflow in Figure 2 begins with key feeding parameters: the schedule, feed type, and daily quantity, being entered into the system by the user. Once submitted, this information serves as the daily feed input and triggers an automatic reduction in feed stock. The system then evaluates the current stock level to determine whether it remains within a safe range. When stock is sufficient, the system displays the corresponding stock data on the feed dashboard for real-time tracking. Conversely, if stock drops below the minimum threshold, the system automatically generates and sends a warning notification to prompt timely replenishment. Regardless of the outcome, both branches converge at the end of the process.

Figure 3 shows how the system handles health data, beginning with the user entering information, including temperature, feed consumption, and observed symptoms, which is then recorded and stored as a health log. The system then assesses whether the temperature reading is within normal limits: a normal result appears on the dashboard. In contrast, a reading above the threshold triggers a warning notification. Once either path is completed, both branches merge to conclude the workflow.

Prototype Construction

A version of the program was developed to include essential features such as recording cattle health information, scheduling feedings, .

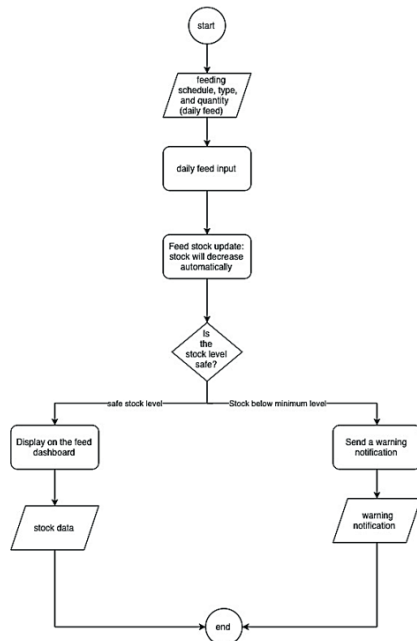


Figure 2. The TSTH2 cattle feed management system workflow

tracking livestock, and sending notifications. The system adopted a three-tier architecture consisting of client, server, and database layers. Within the client layer, a web application was developed using React.js, and a mobile application was developed using Flutter. For the backend implementation, Express.js is utilized by the cattle feed management subsystem, while the Django REST Framework is employed by the cattle health monitoring subsystem. RESTful APIs are provided by both backend services to support communication between the web and mobile applications and the database. MySQL is used as the relational database management system. The initial version was evaluated to ensure proper functionality and to verify that user requirements were met. Subsequent versions were improved based on the feedback provided by users. This information was utilized to enhance both the web and mobile applications.

Deployment Delivery and Feedback

During the Deployment Delivery and Feedback phase of each iteration, the prototypes were evaluated with the farm managers, TSTH2 IT staff, and students. To gather objective and reproducible data, structured evaluation instruments consisting of a semi-structured interview guide and a user feedback evaluation sheet based on a five-point Likert scale were utilized. Functional usability and performance were measured by the evaluation sheets, while qualitative insights,



Figure 3. The cattle health monitoring system workflow

including operational constraints and workflow suggestions, were captured through the semi-structured interviews. Feedback from these structured instruments was systematically logged and analyzed to prioritize refinements in the subsequent planning phases. This approach ensured that critical enhancements, such as the 'Cow Has Not Eaten' status and health notifications, were driven by formal, empirical user feedback rather than subjective interpretations.

System Implementation

After finalizing and approving all prototypes for both web and mobile platforms, the system was launched. Subsequent refinements prioritized speed, stability, and usability to ensure continued accessibility for rural users.

Testing

Before the system was deployed, it was thoroughly tested to confirm proper functionality and to ensure that user needs were effectively met. Two testing methods were utilized for this purpose. Black Box testing was conducted first, in which all 68 functional components across the web and mobile platforms were evaluated to verify that each produced the expected output. Subsequently, User Acceptance Testing (UAT) was performed, during which farm managers and IT staff at TSTH2, along with students, completed a Likert-scale questionnaire; this objectively

includes 24 functional components, and the health monitoring subsystem includes 44 functional components, for a combined total of 68 functions across the web mobile platforms.

Prototyping Iteration Results: Cattle Feed Management System & Cattle Health Monitoring System

In the cattle feed management system, we applied three prototyping iterations to develop the system incrementally. Each iteration followed a series of processes that began with communication, rapid planning, design modeling, and prototype building while collecting feedback. The first iteration is the foundation: CRUD feed, stock schedule, and web prototype history. The second iteration is the daily nutrition version (Flutter), and the third iteration adds uneaten status, graphs, and a date filter.

The cattle health monitoring subsystem comprises three iterations and includes 44 functional components. The system monitors core health indicators such as temperature, appetite, physical symptoms, and treatment history, capturing them in real time on both web and mobile applications. Table 1 shows the health monitoring system that includes three versions. The first iteration includes a foundational CRUD cow profile, daily health record, and web prototype. The second iteration includes per-cow history and a mobile version (Flutter), and the third iteration includes automatic alerts and health graph trends. Each iteration adhered to the core stages of the prototyping model: communication, quick planning, quick design modeling, prototype construction, and deployment with delivery and feedback.

Table 1. Prototyping iteration result

No.	Feature Category	Iteration 1	Iteration 2	Iteration 3
1.	Cattle Feeding	CRUD Feed, stock schedule, and history of web prototype	Daily nutrition version (Flutter)	Uneaten status, graph, and date filter
2.	Cattle Health Monitoring	CRUD cow profile, daily health record, web prototype	History per cow, mobile version (Flutter)	Automatic alerts, health graph trends

Black Box Testing Results

Functional verification was conducted using black-box testing to ensure the reliability and correctness of all 68 functional components across both web and mobile platforms. The

results are visualized in Figure 5 as a 100% stacked bar chart, illustrating the success ratio of the web and mobile platforms per module (cattle feed management and cattle health monitoring).

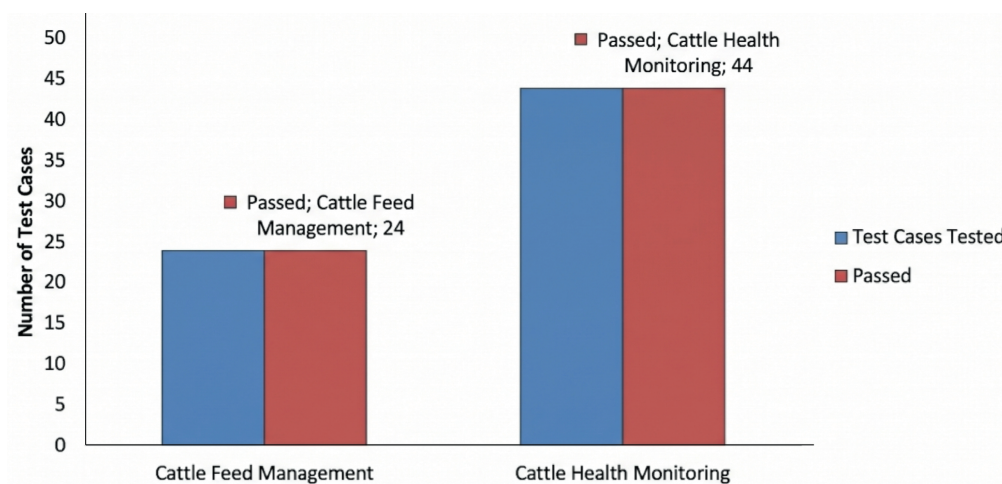


Figure 5. Black box testing results

In the cattle feed management subsystem, all 24 components, implemented across the

React.js web platform and Flutter mobile platform, were tested, achieving a 100%

success rate. Similarly, the cattle health monitoring subsystem, which comprises 44 components, achieved a 100% success rate on both devices. As depicted in the 100% stacked bar chart (Figure 5), both the web and mobile interfaces for both modules achieved a perfect 100% 'Passed' status with 0% failures. This visual representation highlights the consistent cross-platform data alignment and the functional readiness of both subsystems.

However, the Black Box testing primarily focused on functional verification of the system's 68 functional components. Operational aspects related to slow network conditions and offline synchronization were not specifically measured in this testing. These aspects are therefore identified as limitations of the current evaluation and should be considered in future testing, particularly for deployment in rural environments where network connectivity may be limited.

User Acceptance Testing (UAT) Results

User Acceptance Testing (UAT) evaluated the system's usability. Respondents were divided into two groups: target users and students. The target-user group included four farmers and one TSTH2 staff member (n = 5),

while the student group included 23 IT Del students (n = 23). The students were included as additional respondents to provide a usability perspective. The SUS adjective rating scale (Brooke, 2013) was used to determine acceptability ranges, which consist of four categories: acceptable (≥ 70), marginal high (62-69), marginal low (50-61), and not acceptable (< 50). Besides that, the grade scale consists of 5 categories (A=90-100, B=80-89, C=70-79, D=60-69, F= < 60).

Moreover, adjective ratings comprised 6 categories: best imaginable (90-100), excellent (80-89), good (70-79), ok (50-69), poor (35-49), and worst imaginable (0-34). The System Usability Scale (SUS) questionnaire consists of 10 standard statements rated on a five-point Likert scale. Participants indicated their level of agreement with each statement on a scale from 1 (Strongly Disagree) to 5 (Strongly Agree). Figure 6 presents the distribution of SUS scores for the two respondent groups: Target Users and Students. The boxplots show the distribution of SUS scores for each group, while the individual points represent respondents' scores. The System Usability Scale (SUS) questionnaire assessed the system's usability in terms of effectiveness,

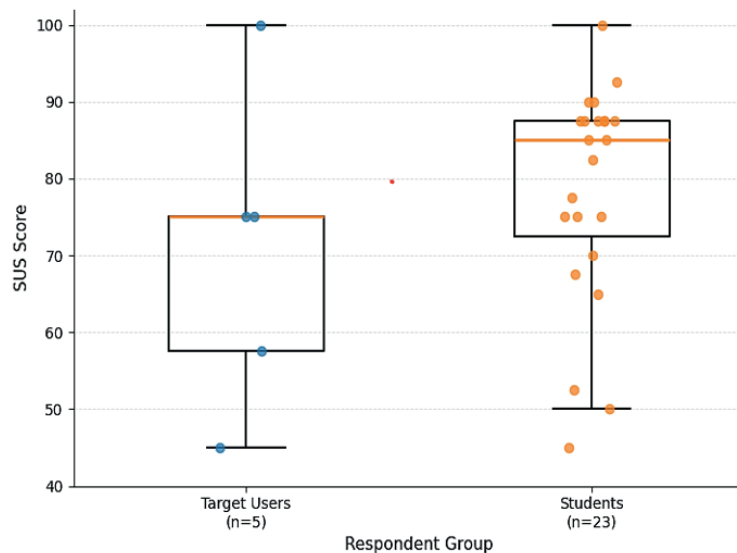


Figure 6. Distribution of SUS scores between target users and students using boxplots

efficiency, and user satisfaction.

The SUS results were analyzed by respondent group to provide a clearer comparison of perceived system usability. The

respondents were divided into two groups, namely Target Users and Students, based on their roles in the evaluation. Table 2 presents the average SUS obtained by each respondent

Table 2. The average score per respondent group

Respondent group	Number of respondents	Average SUS Score	Score category
Target Users	5	70.50	Good (acceptable)
Students	23	78.37	Good (acceptable)

group and its corresponding score category.

Table 2 shows that the Target Users group obtained an average SUS score of 70.50, while the Students group obtained a higher average score of 78.37. Both groups were categorized as 'Good' based on the SUS adjective rating. The Target Users group represents the system's intended users, consisting of four farmers and one TSTH2 staff member. In contrast, the Students group was included as an additional respondent group for usability evaluation.

To determine whether the difference in SUS scores between the two groups was statistically significant, a Mann–Whitney U test was performed. The test compared SUS scores for the Target Users group ($n = 5$) and the Students group ($n = 23$). The test resulted in a Mann–Whitney U value of 41.0 and a p-value of 0.333. Because the p-value was greater than 0.05, SUS scores did not differ significantly between the Target Users and Students groups.

However, the number of respondents in the two groups was substantially unequal, with only five Target Users compared with 23 Students. Therefore, interpret the SUS results as an initial usability evaluation. Future studies should include more actual target users to provide a more representative assessment of system usability and acceptance.

This study set out to build a web- and mobile-based system for cattle feed management and health monitoring, developing it step by step through prototyping. Testing showed that all 68 components performed correctly, and 76.96% of users said they were satisfied with the system's usability and ease of use. The system was evaluated at PT. Cifa Indonesia, and the results indicate that it performs effectively, particularly for potential future implementation in Pollung, TSTH2. The prototyping method is well-suited for developing information systems when user requirements are area-specific and may change. This is supported by Samsumar et al. (2023), Cicimby et al. (2021), and Chaniago et al. (2019).

In this study, three iterations of the prototyping method were conducted. This allowed us to get user feedback and use it to improve the system each time. The system we ended up with is very similar to how Things are actually done. This makes sense because the main idea of prototyping is that involving users makes you less likely to add unnecessary features, and more people will use the system. This aligns with research by Cicimby et al. (2021) and Primada et al. (2021). Some features, like the "Cow Has Not Eaten" status and health alert notifications, were not in the plan. These issues emerged only after thorough

prototype testing. It is a good reminder that some users' needs will not be revealed until that system is used in practice.

The core contribution of this research is integrating feed management and health monitoring into a single platform; this addresses the gap in the previous study, which implemented the system separately: feed management (Nosa & Sulistiani, 2025; Mubarak et al., 2025; Hassim et al., 2011; Setiawan et al., 2025), and health monitoring (Kuldhara et al., 2023; Ariff et al., 2014; Okokpujie et al., 2023; Putra, 2025).

These studies used systems, interfaces, data, and models. Our study fills this gap by creating a platform for feed scheduling, nutrition tracking, consumption monitoring, and health records, available on both the web and Flutter mobile apps. The platform has one backend and one user interface. This integration helps farmers manage tools and use shared data for future analysis, such as linking feed consumption to cattle health. This kind of integration has not been reported in livestock information systems yet. The platform helps farmers make decisions by combining feed management and health monitoring data. Farmers can now easily see how feed consumption affects cattle health.

This study yielded several notable results. Most significantly, the 'Cow Has Not Eaten' feature, which was introduced during Iteration 3 of the feed management subsystem, was highlighted. Through this feature, users are notified regarding which cows have not consumed their allocated feed. This functionality was not anticipated during the initial planning phase; rather, it was identified only after the Iteration 2 prototype was evaluated with prospective users. The necessity of tracking individual feed consumption was strongly emphasized by the farm managers. A similar pattern emerged in the health monitoring subsystem, where health alerts were added during Iterations 2 and 3 in response to user requests. These instances show that critical requirements, which might otherwise be overlooked, can emerge when prototypes are evaluated and user feedback is gathered. Consequently, this iterative process significantly improved both the feed management and health monitoring subsystems. Features such as the 'Cow Has Not Eaten' status and health notifications are crucial feedback-driven enhancements that refined the overall system.

This study has two scientific contributions. First, it shows with evidence that prototyping works well for building integrated livestock information systems in agricultural

institutions in Indonesia. No studies like this exist compared with those that focus on Internet of Things (IoT) approaches, for example: Kuldharan et al. (2023), Ariff et al. (2014), and Okokpuije et al. (2023). Prototyping is effective for constructing these systems. It helps to build integrated livestock information systems. Second, the study presents a framework for developing feed and health management platforms for both web and mobile devices. This framework can be used for different types of livestock and in various institutional settings. The developed system has 68 functions. It is a well-documented advancement in livestock information systems for tropical agriculture. The system is accessible via web and mobile devices. It represents an advancement in these systems. The development framework is adaptable to livestock species. It can be used in institutional environments.

The proposed system is well-suited to TSTH2 and similar institutions that manage dairy cattle farms. A digital platform is preferred over paper-based records due to its compatibility with both desktop and mobile devices. This cross-platform support significantly facilitates data entry, retrieval, and report generation. In the field, staff can record information using Flutter-based devices. Additionally, because data can be entered automatically and verified in real time, data loss and errors associated with delayed entry are effectively minimized. Thus, the system is expected to support the management of Girolando cattle and maintain their daily milk production of 12-18 liters (Pasaribu et al., 2015). Finally, integrating feed tracking and scheduling features further increases operational efficiency at TSTH2.

This system has several limitations, including that it does not integrate with IoT sensors, lacks a scheduling feature, and has not been tested in TSTH2 yet. This system requires integration with an IoT sensor to track body temperature and activity, whereas our system lacks health-monitoring capabilities at TSTH2, meaning data still has to be entered manually. This limitation may increase the risk of errors. Furthermore, it makes it harder for the system to catch problems on its own.

Based on the analysis, it was determined that the feed scheduling feature in the system is still manually configured to calculate each animal's dietary consumption. Consequently, this system still needs improvement to make the process more efficient. Another shortcoming is that it is only a pre-deployment step, since no Girolando cattle were available at TSTH2. Alternatively, the system was tested at

PT. Cifa, which provided an overview of the results for TSTH2. Therefore, for future studies, it is recommended that the system be enhanced in three key areas: the integration of IoT sensors for early and automatic detection of health issues, the application of machine learning to health data so diseases can be flagged sooner, and the incorporation of an AI-powered tool for feed scheduling to tailor nutrition for each animal. These added features can turn the system into a full-fledged smart livestock management system that supports smart agriculture through a combination of IoT, machine learning, and AI-driven feed scheduling.

■ CONCLUSION

The system for managing cattle feeding processes and monitoring health can be applied at TSTH Pollung, North Sumatra. This system was developed through three prototyping cycles. In total, the system includes 68 features: 24 for feeding management and 44 for health monitoring. These features can be accessed through the React.js web platform and the Flutter mobile platform. For the backend, Express.js is utilized by the cattle feed management subsystem, while the Django REST Framework is employed by the cattle health monitoring subsystem. Based on the evaluation, each component performed successfully across both platforms, confirming that the intended functions were executed correctly.

Furthermore, potential users found the system streamlined and effective in meeting their operational needs. Based on feedback collected in our system, we ultimately improved the system by adding feeding consumption status and health notifications. These two functions were not part of the original plan, but they appeared in this study. In the future, three aspects need to be added and evaluated: integrating temperature and activity sensors to detect anomalies in real time, applying machine learning to identify diseases earlier, and building an adaptive feed-scheduling algorithm. All these aspects support a better understanding of each animal's specific nutritional needs. These aspects add intelligence to the system, creating a complete smart livestock management platform that also fulfills TSTH's goal.

■ DECLARATION OF GENERATIVE AI USAGE IN THE WRITING PROCESS

AI assistants (claude.ai) were utilized to assist with translation and to enhance the wording of this manuscript. The authors

reviewed all AI-generated content to ensure accuracy.

■ ACKNOWLEDGMENTS

We gratefully acknowledge the financial support provided by the Research and Community Service Institute (LPPM) of Institut Teknologi Del. We are likewise grateful to the TSTH2 management team, particularly Mr. Agustavo Pacheco, Mr. Yeppi, and Mr. Irvandi Sihombing, for their cooperation and active engagement in developing and evaluating the system.

■ REFERENCES

- Alawiyah, H., & Pamungkas, B. (2026). *Perancangan Sistem Informasi Pengelolaan Pakan Ternak Berbasis Web Menggunakan Metode Waterfall* [Design of a Web-Based Animal Feed Management Information System Using the Waterfall Method]. *Bulletin of Computer Science Research*, 6(4), 1243–1248. <https://hostjournals.com/bulletincsr/article/view/1140/681>
- Althafullayya, M. R., & Aryani, N. P. (2024). *Smart Farming: Mengintegrasikan Teknologi Pemantauan Nutrisi untuk Meningkatkan Kesejahteraan dan Produktivitas Hewan* [Smart Farming: Integrating Nutrition Monitoring Technology to Improve Animal Welfare and Productivity]. *Jurnal Agro-Livestock*, 2 (1). <https://jurnal.ypkpasid.org/index.php/jal/article/view/54/54>
- Ariff, M. H., Ismarani, I., & Shamsuddin, N. (2014). *RFID-Based Systematic Livestock Health Management System*. IEEE. <https://doi.org/10.1109/SPC.2014.7086240>
- Arora, C. H., Autade, S. D., Avhad, A. M., & Deokar, R. D. (2024). Smart cattle care: An IoT-based monitoring and management system. *International Journal of Innovative Science and Research Technology*, 9(4), 1928-1931.
- Astrawijaya, A., & Asriningtias, Y. (2025). *Aplikasi Mobile Berbasis Flutter Untuk Optimasi Manajemen Peternakan Babi Skala Perorangan* [Flutter-Based Mobile Application for Optimizing Individual-Scale Pig Farm Management]. *Media Online*, 6(1), 29–40. <https://www.hostjournals.com/bulletincsr/article/view/851/533>
- Badan Pusat Statistik. (2023). *Produksi Telur Unggas dan Susu Sapi Menurut Provinsi (ton), 2021*.
- Brooke, J. (2013). SUS: A Retrospective. *JUS (Journal of Usability Studies)*, 8, 29–40.
- Chaniago, M. H., Amalia, F., & Widodo, A. W. (2019). Sistem Informasi Manajemen Ternak Sapi Perah di AKA Milk Jakarta. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(7), 6920–6929.
- Christi, R. F., Edianingsih, P., & Sudrajat, A. (2025). Knowledge of Dairy Cattle Management for Millennial Farmers in West Java. *Farmers: Journal of Community Services*, 6(1), 44–49. <https://jurnal.unpad.ac.id/fjcs/article/download/60511/24905>
- Cicimby, C. C., Rusdianto, D. S., & Wihandika, R. C. (2021). *Pembangunan Sistem Informasi Rekam Medis Ramdani Skincare & Spa Malang dengan Metode Prototyping Berbasis Web* [Development of a Medical Records Information System for Ramdani Skincare & Spa Malang Using a Web-based Prototyping Method]. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(12), 5366–5375.
- Darvesh, K., Khande, N., Avhad, S., & Khemchandani, M. (2023). IoT- and AI-based smart cattle health monitoring. *Journal of Livestock Science*, 14(3), 211–218. <https://doi.org/10.33259/jlivestsci.2023.211-218>
- Gifari, Z. Al, Andriati, R., Ashari, M., Poerwoto, H., Suhardiani, Rr. A., Amalyadi, R., Karni, I., Aminurrahman, Septian, I. G. N., Putra, R. A., & Anwar, K. (2025). *Peningkatan Kapasitas Peternak Tradisional melalui Optimalisasi Manajemen Kesehatan, Pakan, dan Pemeliharaan Sapi Semi-Intensif di Desa Teniga, Lombok Utara* [Increasing the Capacity of Traditional Cattle Breeders through Optimizing Health Management, Feed, and Semi-Intensive Cattle Maintenance in Teniga Village, North Lombok]. *Jurnal Pepadu*, 6(1), 92–100. <https://doi.org/10.29303/pepadu.v6i1.6810>
- Hassim, Y. M. M., Hassan, N., & Arbaiy, N. (2011). *Web-based Dairy Cattle Feeding Allocation System*. 2. <http://eprints.uthm.edu.my/7948/>
- Khairi, F., Haryanti, R., & Astuti, P. (2025). Precision Feed Management with Automation Technology to Optimize the Growth and Health of Beef Cattle. *The Journal of Academic Science*, 2(10), 2228–2236.
- Kulldharan, S. D., Kadake, S. S., Kamble, N. K., Bhavsar, M. M., & Honwadkar, K. N. (2023). Cattle Health Monitoring System using IoT. *International Journal of*

- Advanced Research in Science, Communication and Technology*, 450–455. <https://doi.org/10.48175/ijarsct-8855>
- Kumala, A. E., Borman, R. I., & Prasetyawan, P. (2018). *Sistem Informasi Monitoring Perkembangan Sapi di Lokasi Uji Performance (Studi Kasus: Dinas Peternakan dan Kesehatan Hewan Provinsi Lampung)* [Cattle Development Monitoring Information System at Performance Test Locations (Case Study: Lampung Province Animal Husbandry and Animal Health Service]. *Jurnal TEKNOKOMPAK*, 12(1). <https://doi.org/10.33365/jtk.v12i1.52>
- Mubarok, H., Walid, M., & Umam, B. A. (2025, October). RANCANG BANGUN SISTEM PEMBERIAN PAKAN TERNAK SAPI BERBASIS INTERNET OF THINGS (IOT). In *Seminar Nasional Humaniora dan Aplikasi Teknologi Informasi (SEHATI)* (Vol. 11, No. 1, pp. 165–172).
- Nosa, S. M., & Sulistiani, H. (2025). *Sistem Monitoring dan Manajemen Pakan Ternak Sapi Berbasis Web Pada PT XYZ Lampung Tengah* [Web-Based Cattle Feed Monitoring and Management System at PT XYZ Central Lampung]. *SMATIKA JURNAL*, 15(01), 151–166. <https://doi.org/10.32664/smatika.v15i01.1717>
- Okokpujie, K., Okokpujie, I. P., Ogundipe, A. T., Anike, C. D., Asaboro, O. B., & Vincent, A. A. (2023). Development of a Sustainable Internet of Things-Based System for Monitoring Cattle Health and Location with Web and Mobile Application Feedback. *Mathematical Modelling of Engineering Problems*, 10(3), 740–748. <https://doi.org/10.18280/mmep.100302>
- Pakpahan, B., Suprpto, B., Melvi, L., Eswanto, R. S., Dharma, S. D. M., & Jubaidah. (2025). *Inovasi teknologi pengolahan pakan dan limbah ternak sapi: solusi hemat biaya untuk keberlanjutan usaha* [Innovation in cattle feed and waste processing technology: a cost-effective solution for sustainable business]. *Community Development Journal: Jurnal Pengabdian Masyarakat*, 6, 709–714.
- Pasaribu, A., Firmansyah, & Idris, N. (2015). *Analisis Faktor-Faktor Yang Mempengaruhi Produksi Susu Sapi Perah Di Kabupaten Karo, Provinsi Sumatera Utara*. <https://online-journal.unja.ac.id/jiip/article/view/2656/7957>
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). McGraw-Hill. www.mhhe.com/pressman.
- Primada, M. M., Arwan, A., & Pramono, D. (2021). *Sistem Informasi Manajemen Peternakan Sapi Perah. Studi Kasus Dinas Pertanian Kota Batu* [Dairy Farm Management Information System: A Case Study of the Batu City Agriculture Service]. *Jurnal Pengembangan Teknologi dan Ilmu Komputer*, 5(3), 984–991.
- Putra, I. M. A. (2025). *Peran Teknologi Digital dalam Monitoring Kesehatan Ternak Ruminansia* [The Role of Digital Technology in Ruminant Livestock Health Monitoring]. *Journal of Animal Husbandry*, (1). <https://ejournal.kalibra.or.id/index.php/joah/article/download/47/81/1211>
- Qisthon, A., Sirat, M. M. P., Farda, F. T., & Wanniatie, V. (2024). *Edukasi Peternak Sapi melalui Penyuluhan Manajemen Pemeliharaan, Perkandangan, Kesehatan dan Reproduksi serta Pelatihan Fermentasi Pakan* [Educating Cattle Breeders through Extension on Maintenance Management, Housing, Health and Reproduction as well as Feed Fermentation Training]. *Jurnal Pengabdian Fakultas Pertanian Universitas Lampung*, 3(1), 1–017. <https://doi.org/10.23960/jpfp.v3i1.8332>
- Rahmawati, D., Haryanto, Rinaldi, A. R., Setiawan, H., H., & Alfita, R. (2024). *Fuzzy Smart Precision Livestock Farming System Design for Cattle Farming*. <https://doi.org/10.24912/tesla.v26i2.31446>
- Samsumar, L. D., Salman, S., Muslim, R., & Akbar, A. (2023). *Smart automatic feed: Sistem pakan otomatis pada kandang peternak ayam* [Smart automatic feed: Automatic feeding system in chicken farms]. *Jurnal Publikasi Teknik Informatika*, 2(2), 149–160.
- Sasongko, A. (2026, January). *Perancangan Sistem Monitoring Suhu Sapi dan Mengontrol Suhu Sapi Menggunakan Sensor Infrared* [Designing a Cow Temperature Monitoring System and Controlling Cow Temperature Using an Infrared Sensor]. In *Seminar Nasional Teknologi & Sains* (Vol. 5, No. 1, pp. 353–358).
- Setiawan, B. R., Pramuntadi, A., Wijaya, D. P., & Danianti, D. (2025). *Sistem Monitoring Peternakan Sapi Potong Berbasis Android* [Android-Based Beef Cattle Farming Monitoring System]. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(5), 7564–7570.

- Sugiharti, E., Putra, A. T., Suyitno, H., & Subhan. (2019). Android-based online cattle card system for recording quality cattle in Semarang Regency. *Journal of Physics: Conference Series*, 1321(3). <https://doi.org/10.1088/1742-6596/1321/3/032018>
- Tanzil, N.L., Dewi, N., Agustiyani, V.P., Irviani, R. (2025). *Strategi pengembangan usaha peternakan sapi potong Desa Wariginsari Timur di era digital* [Strategy for developing beef cattle farming business in East Wariginsari Village in the digital era]. *Jurnal PkM Pemberdayaan Masyarakat*, 6(3), 124–134.
- Thannithi, W., Intawicha, P., Phuwisanarakom, P., & Saengwong, S. (2024). Design and evaluation of a mobile application for enhancing farm management and performance assessment in fattening beef cattle. *Journal of Advanced Veterinary and Animal Research*, 11(1), 203–211. <https://doi.org/10.5455/JAVAR.2024.K766>
- Vigneswari, T., Kalaiselvi, N., Mathumitha, K., Niveditha, A., & Sowmian, A. (2021). Smart IoT Cloud-Based Livestock Monitoring System: A Survey. In *Turkish Journal of Computer and Mathematics Education* (Vol. 12, Number 10).
- Zaharieva, S., & Dunchev, D. (2026). Precision livestock farming: concepts and future perspectives. *Scientific Works*, 67(3), 189–198. <https://doi.org/10.22620/sciworks.2025.03.017>