



Service Information System at Kiyowo Vet Jember Animal Clinic

Angga Ramadana^{1*}, Moh. Dasuki¹, Qurrota A'yun¹, Ulya Anisatur Rosyidah¹, Habibatul Azizah Al Faruq¹

¹ Universitas Muhammadiyah Jember, Jawa Timur, 68135, Indonesia

* Corresponding author: anggaramadhana29@gmail.com

Submitted: 01/02/2025 | Revision Accepted: 02/03/2025 | Online: 05/03/2025 | doi: xxxx.xxxx.xxxx/xxxxxx

Abstract:

Purpose: The increasing concern for animal welfare has led to a rise in pet ownership and a growing demand for proper healthcare, encouraging clinics like Kiyowo Vet Jember to provide essential services such as treatment and health check-ups. However, the clinic still relies on a manual system for managing patient records, resulting in operational inefficiencies and highlighting the need for an integrated system to improve service quality. Therefore, the purpose of this research is to develop and evaluate an integrated digital system for managing patient records at Kiyowo Vet Jember, with the goal of enhancing operational efficiency and improving service quality in veterinary healthcare.

Methods: The study used the Waterfall method, which includes six phases: Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. A Use Case Diagram was developed to illustrate system interactions. An Entity-Relationship Diagram (ERD) was created to define data structure and relationships.

Results: A web-based information system was implemented, allowing the clinic to manage data such as customer records, animal types, available services, and reservations. Black Box testing was conducted to verify system functionality, showing that all key features (customer records, reservation management, and service details) functioned correctly.

Conclusions: The web-based system improves operational efficiency by enabling online data management. The clinic can now efficiently handle customer and pet administration through the system's various features. The new system enhances service management at Kiyowo Vet Jember Animal Clinic.

Keywords:

software development, veterinary information system, waterfall method, web-based system

1. Introduction

Human concern for animals is increasing, as evidenced by the growing number of people who choose to keep and care for various types of pets, including cats, dogs, hamsters, and others (Amiot, Gagné, & Bastian, 2023). When selecting a pet, owners typically acquire them through adoption, breeding, or rescuing stray animals that have been abandoned or are without owners (Ly, Gordon, & Protopopova, 2021). Just like humans, animals require proper care beyond food and shelter; their health also needs to be a primary consideration (Hoy-Gerlach & Townsend, 2023).

When caring for animals, pet owners take various measures to maintain their pets' health (Forrest, Pearson, & Awawdeh, 2023). One common practice is regularly checking their health at animal healthcare facilities, including pet shops, clinics, and animal hospitals. The growing demand for pet care and boarding services has created a need for a system that helps clinics improve service efficiency (Ouedraogo, Weinstein, & Lefebvre, 2023). Kiyowo Vet Jember is one such animal clinic located in Jember Regency, providing pet care, treatment, and health check-up services. However, despite handling a large number of patients with varying needs, the clinic still relies on a manual system for medical records. This poses a significant challenge in establishing a well-integrated data processing system, which is essential for improving service quality and operational efficiency (Kalid, et al., 2024).

In providing services, the Kiyowo Vet Jember clinic still relies on a manual system. Patient registration is recorded manually in books, and each time a patient arrives, the nurse must search for their data to determine whether they have been previously registered. This process is repeated for every incoming patient, making it time-consuming and inefficient for clinic operations. Given these challenges, there is a need for a system that can help Kiyowo Vet Jember manage its operations more effectively (Siyam, et al., 2021).

2. Methods

The method used in this study is waterfall, this is because the user requirements of the Kiyowo Vet Jember Animal Clinic have been conveyed clearly and are on a small scale. The stages in the waterfall method are Requirement Analysis, System Design, Implementation, Testing, Deployment, Maintenance (Maylawati & Ramdhani, 2022).



Figure 1: Waterfall Method

Requirement analysis is a crucial phase in software development that involves gathering and documenting all necessary software requirements before any development work begins (Umar & Lano, 2024). This process ensures that all stakeholders, including clients, developers, and project managers, have a clear understanding of the project's scope, objectives, and constraints. During this phase, detailed discussions, interviews, and research are conducted to identify functional and non-functional requirements, which are then thoroughly documented for reference throughout the project. It is essential to finalize all requirements before proceeding with development to avoid costly revisions, scope creep, and misalignment between stakeholder expectations and the final product. By establishing a solid foundation through comprehensive requirement analysis, teams can enhance efficiency, minimize risks, and ensure that the software meets user needs and business goals effectively.

System design is a critical stage in software development where the overall architecture and structure of the system are created based on the finalized requirements (Iovescu & Tudose, 2024). This phase involves defining the system's hardware, software, and



database structures to ensure seamless integration and optimal performance. By carefully designing the system architecture, developers establish a clear framework that guides implementation, ensuring scalability, reliability, and maintainability. Hardware specifications are determined to support the system's processing needs, while software components and frameworks are selected to align with functional and non-functional requirements. Additionally, database structures are designed to store and manage data efficiently, ensuring security, accessibility, and performance. A well-planned system design not only streamlines development but also helps prevent technical challenges and inefficiencies in later stages of the project.

The implementation phase, also known as coding, is where developers translate the system design into actual code based on the predefined specifications (Al-Sarayreh, Meridji, Alenezi, Zarour, & Al-Majali, 2021). During this phase, the software is developed in modular components, allowing for better organization, maintainability, and testing. Each module is carefully coded, ensuring it meets functional and performance requirements while adhering to best coding practices and industry standards. Once individual modules are completed, they are systematically integrated to form a fully functional system, ensuring seamless interaction between different components. This modular approach not only simplifies debugging and troubleshooting but also enhances scalability and future updates. A well-executed implementation phase ensures that the software functions as intended, bringing the project closer to deployment.

The testing phase is a crucial step in software development where the system is thoroughly examined for bugs, defects, and inconsistencies to ensure it functions as expected (Ebert, Bajaj, & Weyrich, 2022). This process includes multiple levels of testing, starting with unit testing, where individual components are tested in isolation to verify their correctness. Integration testing follows, ensuring that different modules work seamlessly together. System testing is then conducted to evaluate the software as a whole, checking for performance, security, and overall functionality. Finally, user acceptance testing (UAT) is performed, allowing end-users to validate that the system meets their needs and expectations before deployment. By systematically identifying and resolving issues, the testing phase enhances software quality, reliability, and user satisfaction, ultimately leading to a stable and efficient final product.

The deployment phase marks the final stage of software development, where the completed and thoroughly tested software is released to users (Lăcătușu, Ionita, Anton, & Lăcătușu, 2022). Depending on the project requirements, the software may be installed on a client's system for on-premises use or made available online through cloud-based or web-based platforms. This phase requires careful planning to ensure a smooth transition from development to a live environment, minimizing downtime and potential disruptions. Deployment may also involve data migration, configuration, and initial user training to facilitate seamless adoption. Additionally, monitoring and support are often provided post-deployment to address any unexpected issues and ensure optimal performance. A well-executed deployment phase ensures that users can effectively access and utilize the software, marking the successful completion of the development process.

The maintenance phase is an essential part of the software development lifecycle, ensuring that the system remains functional, secure, and up-to-date after deployment

(Domagała, Grobler-Dębska, Waś, & Kucharska, 2021). This phase involves ongoing support, bug fixes, and performance improvements based on user feedback and real-world usage. As users interact with the software, issues may arise that require prompt resolution to maintain efficiency and reliability. Additionally, enhancements may be introduced to improve functionality, address evolving user needs, and keep the system compatible with new technologies. Regular updates, security patches, and optimizations help extend the software's lifespan while ensuring a seamless user experience. Effective maintenance ensures that the software continues to perform optimally, adapting to changes and improvements over time.

3. Results and Discussion

A Use Case Diagram is created to describe the interaction between actors and the main functionality of the system. This use case diagram provides a more detailed view of the system usage scenario.

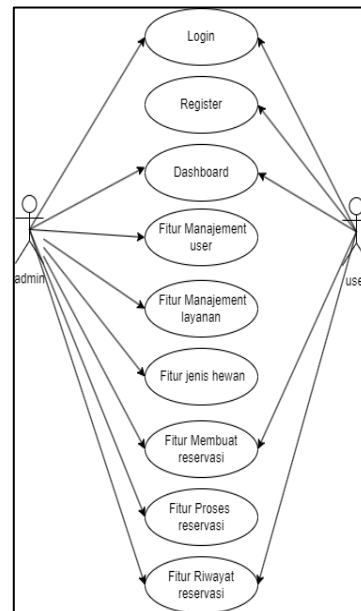


Figure 2: Use Case Diagram

After designing the use case diagram, the next step is to create an Entity-Relationship Diagram (ERD). ERD is used to describe the relationship between entities in the system and how they interact with each other. This diagram provides a deeper picture of the structure and data relationships in the system database. ERD can be seen in Figure 3.

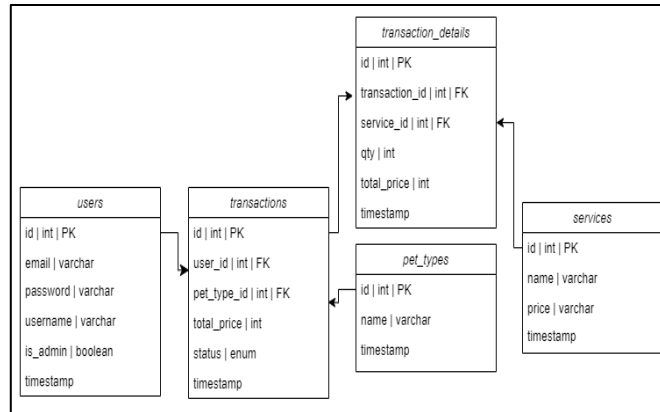


Figure 3: Entity Relationship Diagram

The following is a description of the program developed based on the information system overview:

1. Home

The home page contains a summary of data starting from the number of users, types of incoming services to the number of incoming and completed reservations

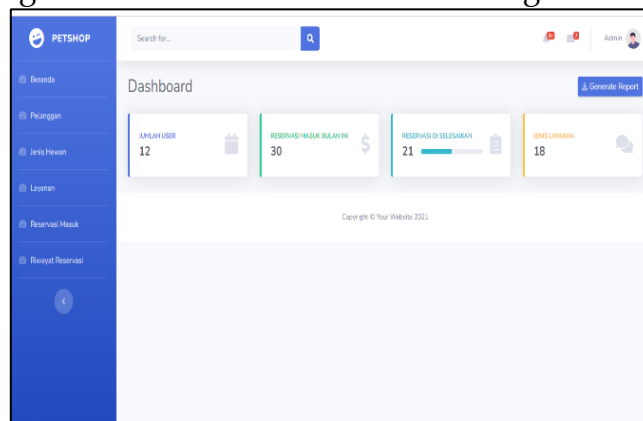


Figure 4: Home Section

2. Customer

The customer menu contains a list of customer data who have made reservations.

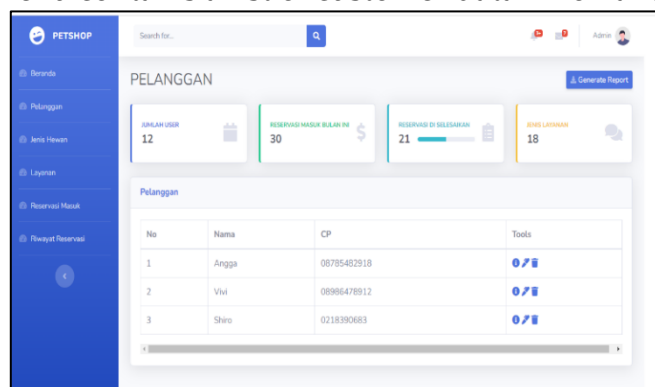


Figure 5: Customer Section



3. Types of Animals

The animal type menu contains a list of animals that can be reserved.

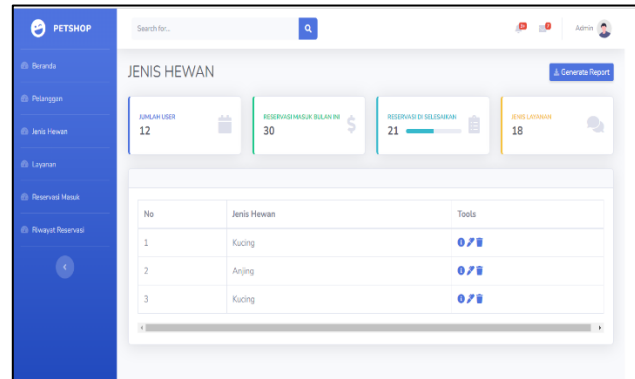


Figure 6: Animals Section

4. Service

This menu contains a list of services provided by the Kiyowo Vet Jember Animal Clinic.

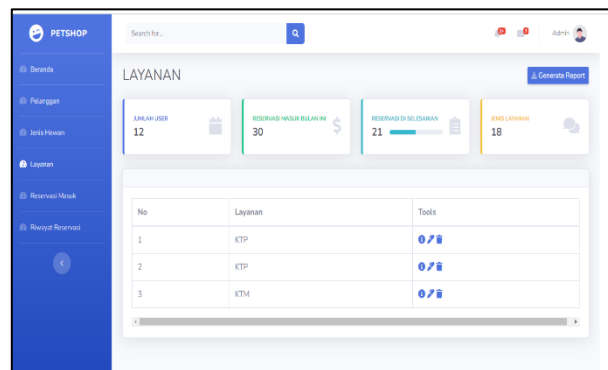


Figure 7: Service Section

5. Reservation Entry

Incoming reservations is a menu that contains a list of reservations that have been received or are currently running

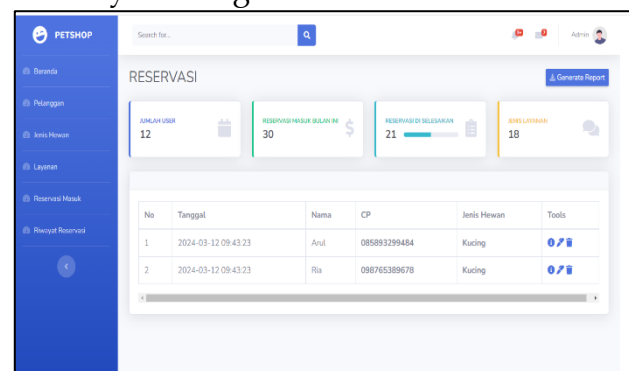


Figure 8: Reservation Section

6. Reservation History

In this menu, the admin can see the history of reservations that have been made.

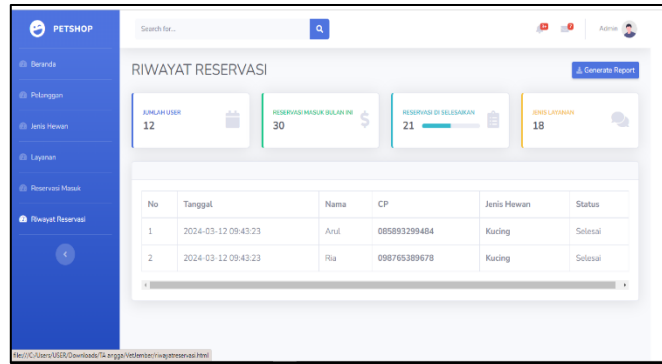


Figure 9: Reservation History

Black Box testing focuses on testing the functionality of the system by using an observation system on the data to be tested.

Table 1. Black Box Testing Results

| Tester | Condition | Test Results |
|--------------------|---|--|
| Admin Dashboard | Input data then submit | Displays the menu bar available in the clinic information system |
| Customer | Click on the customer menu in the admin information system | Displays a list of customers who made reservations |
| Types of Animals | Click on the animal type menu in the admin information system | Displays the types of animals that can be served at the clinic |
| Service input | Click on the login service in the admin information system | Displays details of available services |
| Entry reservation | Click on the reservation menu to enter the admin information system | Displaying reservation data from customers |
| Make a reservation | Input data then submit | Data was successfully saved to the database, and will be displayed on the admin information system |

4. Conclusions

The conclusions from this research on the Service Information System at Kiyowo Vet Jember Animal Clinic are as follows:

1. This information system website facilitates online data management.
2. The animal clinic can efficiently manage customer and pet administration data through a web-based platform, which includes various supporting functions such as customer name input and more.
3. With this system, customer service management at Kiyowo Vet Jember Animal Clinic becomes more efficient.



Acknowledgement

Insert acknowledgment, if any. The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Sponsor and financial support acknowledgments are also placed here.

References

- Al-Sarayreh, K. T., Meridji, K., Alenezi, M., Zarour, M., & Al-Majali, M. D. (2021). A sustainable procedural method of software design process improvements. *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)* Vol.21 No.1.
- Amiot, C. E., Gagné, C., & Bastian, B. (2023). Exploring the role of our contacts with pets in broadening concerns for animals, nature, and fellow humans: a representative study. *Scientific Reports* 13.
- Domagała, A., Grobler-Dębska, K., Wąs, J., & Kucharska, E. (2021). Post-Implementation ERP Software Development: Upgrade or Reimplementation. *Applied Sciences* Vol.11 No.11.
- Ebert, C., Bajaj, D., & Weyrich, M. (2022). Testing Software Systems. *IEEE Software* Vol.39 No.4.
- Forrest, R., Pearson, M., & Awawdeh, L. (2023). Pet Owners' Attitudes and Opinions towards Cat and Dog Care Practices in Aotearoa New Zealand. *Veterinary Sciences*, 606. doi:<https://doi.org/10.3390/vetsci10100606>
- Hoy-Gerlach, J., & Townsend, L. (2023). Reimagining Healthcare: Human-Animal Bond Support as a Primary, Secondary, and Tertiary Public Health Intervention. *Int. J. Environ. Res. Public Health*, 5272.
- Iovescu, D., & Tudose, C. (2024). Real-Time Document Collaboration—System Architecture and Design. *Applied Sciences* Vol.14 No.18.
- Kalid, S. N., Ho, S.-B., Ibrahim, A., Azman, J. H., Tan, S.-J., & Cheong, J.-E. (2024). Enhancing Personalized Healthcare with an Effective E-Healthcare Management System. *Journal of System and Management Sciences* Vol.14 No.2.
- Lăcătușu, M., Ionita, A. D., Anton, F. D., & Lăcătușu, F. (2022). Analysis of Complexity and Performance for Automated Deployment of a Software Environment into the Cloud. *Applied Sciences* Vol.12 No.9.
- Ly, L. H., Gordon, E., & Protopopova, A. (2021). Inequitable Flow of Animals in and Out of Shelters: Comparison of Community-Level Vulnerability for Owner-Surrendered and Subsequently Adopted Animals. *Frontiers in Veterinary Science* Vol.8.
- Maylawati, D. S., & Ramdhani, M. A. (2022). Logical Framework of Information Technology: Systematization of Software Development Research. *Telfor Journal* Vol.14 No.1.
- Ouedraogo, F. B., Weinstein, P., & Lefebvre, S. L. (2023). Increased efficiency could lessen the need for more staff in companion animal practice. *Journal of the American Veterinary Medical Association*, 261(9), 1357-1362.



- Siyam, A., Ir, P., York, D., Antwi, J., Amponsah, F., Rambique, O., . . . O'Neill, K. (2021). The burden of recording and reporting health data in primary health care facilities in five low- and lower-middle income countries. *BMC Health Service Research* 21 (Suppl 1).
- Umar, M. A., & Lano, K. (2024). Advances in automated support for requirements engineering: a systematic literature review. *Requirements Engineering* 29, 177–207.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of publisher: UCMM Konsortium Sdn. Bhd. and/or the editor(s). The publisher: UCMM Konsortium Sdn. Bhd. and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.