

Original Research

Web-Based Inventory Information System for Asset and Consumable Management at Ratumbuysang Mental Hospital

Faith Nathaniel Alexander ^{a*}, Chriestie E. J. C. Montolalu ^b, Wisard Kalengkongan ^c, Christian Alderij Soewoeh ^d, Mahardika Inra Takaendengan ^e, Stephano C.W. Ngangi ^f

^{a,b,c,d,e,f}Department of Information Systems, Faculty of Mathematics and Natural Sciences, Sam Ratulangi University, Manado, Indonesia

*Corresponding Author: faith911416@gmail.com

ABSTRACT

This study aimed to design and develop a web-based Inventory Information System for Ratumbuysang Mental Hospital to address inventory management that was previously performed manually. Manual recording made inventory searching, asset tracking, and report preparation less efficient and increased the risk of recording errors. The system was developed using the Waterfall method through requirements analysis, system design, implementation, testing, and maintenance. The application uses React.js for the frontend, Node.js and Express.js for the backend, PostgreSQL for data management, and Prisma ORM as the database interface. The implemented functions include item master-data management, incoming assets, outgoing assets, consumable items, user management, report generation, and QR Code-based asset identification. Black Box Testing was conducted using 27 functional scenarios and showed that all tested functions operated according to the expected results. User Acceptance Testing (UAT) involved 24 statements using a five-point Likert scale. The overall mean score was 4.68, equivalent to 93.50%, and was categorized as Very Good in the original evaluation. These findings indicate that the developed system was accepted by users and can support structured inventory management at Ratumbuysang Mental Hospital.

Copyright © 2026 by Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0).

How to Cite:

F. N. Alexander, C. Montolalu, W. Kalengkongan, C. A. J. Soewoeh, M. I. Takaendengan, and S. C. W. Ngangi, "Web-Based Inventory Information System for Asset and Consumable Management at Ratumbuysang Mental Hospital," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 5, no. 3, pp. 852–861, 2026.

Doi: <https://doi.org/10.69916/jkbt.v5i3.613>

KEYWORDS

Inventory Information System
Web Application
React.js
Waterfall
QR Code

ARTICLE INFO

Received: July 31, 2026

Accepted: September 27, 2026

Available Online: September 30, 2026

1 INTRODUCTION

Information technology has become an important component of administrative and operational management in healthcare organizations. Inventory is one of the operational resources that requires accurate records because hospitals need information about item quantities, conditions, locations, and usage histories to support daily activities [1]. A web-based information system can integrate recording, storage, retrieval, and reporting so that inventory information can be accessed more efficiently.

Ratumbuysang Mental Hospital is a government-owned healthcare institution in North Sulawesi that manages various inventories, including medical equipment, non-medical equipment, furniture, operational vehicles, and consumable items. Based on communication with inventory personnel, the existing process included recording received items, incoming assets, asset usage, consumable items, and outgoing assets. The records were still managed manually, which made data retrieval and report preparation time-consuming and increased the risk of recording errors.

Previous studies have reported the use of web-based inventory systems in retail and institutional environments.

Octavianus developed a web-based inventory system for a retail case, while Fauzi reported an inventory information system for a company environment [4, 5]. Other studies have applied web-based information systems and the Waterfall approach in institutional and healthcare-related contexts [6–9]. These studies demonstrate the usefulness of information systems for inventory and administrative activities, but their process configurations are not specifically aligned with the inventory workflow of Ratumbusang Mental Hospital.

The research gap addressed in this study is therefore the need for an inventory system that reflects the hospital’s operational workflow while integrating asset-level identification through QR Code. The QR Code standard provides a mechanism for automatic identification and data capture that can be used to associate a physical asset with its digital record [12]. The proposed system also separates access rights between administrators and warehouse officers so that inventory activities can be performed according to user responsibilities.

The novelty of this work lies in the integration of hospital-specific inventory processes, role-based access, asset-level QR Code identification, report export, and systematic Waterfall development within a single web-based application. React.js was selected for the frontend because it supports interactive web interfaces, while the backend uses Node.js and Express.js to process application requests [1, 10]. PostgreSQL was used as the database management system [11].

Accordingly, this study aims to design and implement a web-based Inventory Information System for Ratumbusang Mental Hospital using React.js and the Waterfall method. The study evaluates the resulting system using Black Box Testing for functional verification and User Acceptance Testing (UAT) to measure user acceptance.

2 RESEARCH METHODS

2.1 Research Design and Data Collection

The study used a descriptive research approach combined with information-system development. The research object was inventory management at Ratumbusang Mental Hospital. Data were obtained through interviews with inventory personnel, document review, and literature study. The interviews focused on the existing recording process, operational problems, user requirements, and desired system functions. The thesis documentation states that direct field observation had not yet been performed at the initial stage; therefore, understanding of the workflow was obtained through communication and explanations from hospital personnel [13–17].

The research process began with problem identification and literature study, followed by data collection, requirements analysis, system design, implementation, system testing, and report preparation. The original research workflow is presented in Figure 1.

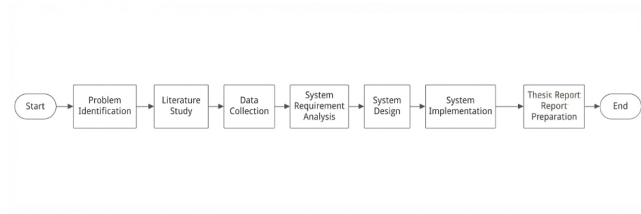


Figure 1. Research framework used to develop and evaluate the inventory information system.

2.2 Waterfall Development Method

The system was developed using the Waterfall model because its sequential stages provide a structured development path. The implementation followed five stages: requirement analysis, system design, implementation, testing, and maintenance. Requirement analysis identified functional and non-functional needs. System design used UML and related process models. Implementation translated the design into a web application [18–20]. Testing verified functional behavior and user acceptance, while maintenance was defined as the stage for correcting errors and supporting future improvements [3].

Table 1. Waterfall stages and activities.

Stage	Main activity
Requirement analysis	Identify functional and non-functional requirements from interviews, documents, and literature.
System design	Develop Use Case, Activity, Class, ERD, BPMN, and interface designs.
Implementation	Develop the application using React.js, Node.js/Express.js, PostgreSQL, and Prisma ORM.
Testing	Conduct Black Box Testing and User Acceptance Testing.
Maintenance	Support corrections and future feature improvements after deployment.

2.3 System Requirements and Architecture

The system has two user roles: Admin and Warehouse Officer. Both roles can authenticate, view the dashboard, manage inventory-related transactions, use QR Code functions, update profiles, and log out. The Admin role additionally manages user accounts and inventory reports. The application architecture consists of React.js at the frontend, Node.js and Express.js at the backend, PostgreSQL as the database, and Prisma ORM as the data-access layer. QR Code is generated for individual assets so that each unit can be identified and retrieved through scanning.

The Use Case Diagram in Figure 2 summarizes the interactions between the two user roles and the main system functions. The model shows that administrative functions extend the operational functions available to warehouse personnel.



Figure 2. Use Case Diagram of the inventory information system.

2.4 Evaluation Procedure

Two evaluation methods were used. First, Black Box Testing verified system behavior from the perspective of inputs and expected outputs without examining the internal program code. Twenty-seven scenarios covered login, master-data management, incoming and outgoing assets, QR Code generation and scanning, consumable items, reports, user management, profile updates, and logout. The success rate of Black Box Testing was calculated based on the number of successfully executed scenarios compared with the total number of tested scenarios using the following formula:

$$\text{Black Box Success Rate} = \left(\frac{\text{Number of Successful Scenarios}}{\text{Total Number of Tested Scenarios}} \right) \times 100\% \quad (1)$$

Second, User Acceptance Testing (UAT) was performed using 24 statements rated on a five-point Likert scale: 1 (strongly disagree) to 5 (strongly agree). The mean score for each statement and the overall percentage were calculated using the following formulas:

$$\text{Mean Score} = \frac{\sum \text{Score}}{\text{Number of Respondents}} \quad (2)$$

$$\text{Overall Mean} = \frac{\sum \text{Mean Score for All Questions}}{\text{Number of Questions}} \quad (3)$$

$$\text{Overall Percentage} = \left(\frac{\text{Overall Mean}}{5} \right) \times 100\% \quad (4)$$

These evaluation measures were used to assess both the functional performance of the developed system and its

acceptance by the intended users.

3 RESULTS AND DISCUSSION

3.1 Implementation of the Inventory Information System

The developed application translates the requirements identified during the analysis stage into an integrated inventory-management workflow. The system does not only provide a digital form for recording items, but connects master data, individual asset records, consumable stock, user accounts, QR Code identification, and reporting within one application. This structure addresses the central problem identified in the preliminary analysis, namely that inventory records were previously distributed across manual documents and repeated spreadsheet recording, making retrieval and reporting less efficient.

The implementation separates the functions according to user responsibility. The Admin role is responsible for broader administrative control, including inventory management, reporting, and user-account management, while the Warehouse Officer focuses on operational inventory activities. This separation is reflected in the system navigation and access rights. From a system-design perspective, role separation is important because it limits administrative functions to authorized users and organizes operational work according to responsibility.

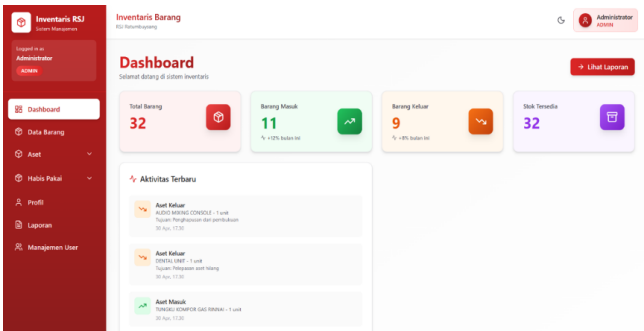


Figure 3. Admin dashboard of the implemented inventory information system.

Figure 3 shows the Admin dashboard as the central interface for accessing the application’s main functions. The dashboard provides a consolidated starting point for inventory activities and reduces the need to move between separate manual records. Although the present study did not measure task completion time before and after implementation, the availability of a single interface for these functions directly addresses the previously identified difficulty of searching and compiling inventory information.

3.2 Master Data and Asset Transaction Management

The system provides an item data management module for recording and managing inventory information. This module displays essential item attributes, including item codes, item names, item types, categories, and units, and provides functions for searching, filtering, adding, editing, and deleting item records. This feature enables inventory staff to maintain centralized and structured item data within the system. The item data management interface is presented in Figure 4.

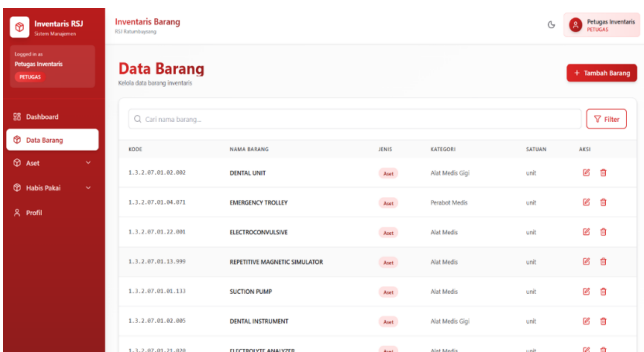


Figure 4. Item master-data management interface.

The item master-data module forms the foundation of the inventory transaction process. The implemented functions include adding, editing, deleting, and searching item records. Separating master data from transaction data allows an item type to be maintained as a reference while individual incoming assets can subsequently be recorded as specific units. This is particularly relevant for hospital inventory because the same type of item can occur in more than one physical unit and each unit may have a different condition or transaction history.

The incoming-asset function records newly received assets and supports generation of a QR Code for each asset. The outgoing-asset function records the movement or removal of an asset and includes QR Code scanning and condition updating. The consumable-item function, meanwhile, focuses on stock-oriented records in which quantities can be updated as items are received or consumed. These distinctions demonstrate that the system models different inventory behaviors rather than treating all inventory records as a single undifferentiated process.

From an operational perspective, the integration of these modules can reduce repeated data entry because subsequent activities can refer to records that have already been stored in the system. However, the present study establishes this conclusion from functional implementation and user evaluation; it does not provide a controlled before-and-after measurement of data-entry time or error frequency. Therefore, the result should be interpreted as evidence that the required functions have been implemented and accepted, rather than as a quantified efficiency improvement.

3.3 QR Code-Based Asset Identification

The system provides a QR Code scanning feature to facilitate the identification of assets. Through this feature, inventory staff can scan the QR Code attached to an asset using the device camera. The scanning process is intended to provide quick access to the corresponding asset information and support more efficient asset identification and management. The QR Code scanning interface is presented in Figure 5.

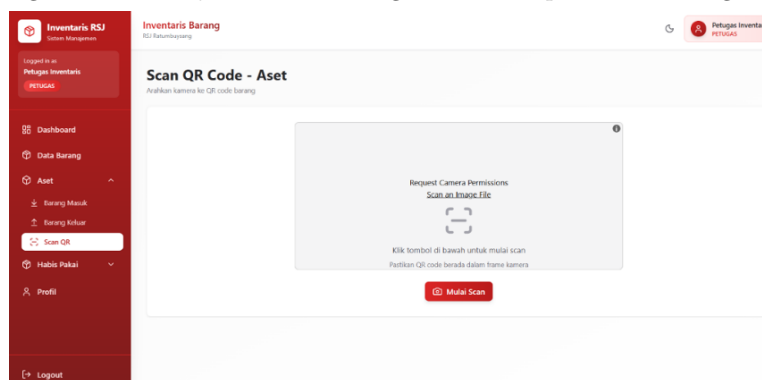


Figure 5. QR Code scanning interface for asset identification.

The QR Code feature provides a direct connection between a physical asset and its digital inventory record. During the incoming-asset process, the system generates a QR Code associated with the recorded asset. During the outgoing-asset process, the QR Code can be scanned so that the corresponding asset information can be retrieved before the transaction is recorded.

This mechanism changes the identification process from a record-searching activity into an identification-and-retrieval activity. For an operational user, the QR Code can reduce dependence on remembering item names or searching manually through a long inventory list. The Black Box test also specifically verified QR Code generation and QR Code scanning, and both scenarios were reported as successful. Consequently, the QR Code component is not merely a visual addition to the interface; it is connected to the transaction workflow and was included in functional testing.

Nevertheless, QR Code identification remains dependent on the quality and availability of the physical label and the scanning environment. The current evaluation did not quantify scanning success under different lighting conditions, camera qualities, label damage levels, or network conditions. Such measurements would be appropriate in future work if QR Code performance is to be evaluated as an independent technical variable.

3.4 Reporting and Administrative Support

The system also provides a reporting feature to support inventory monitoring and administrative activities. The reporting module allows users to generate stock reports, asset transaction reports, and used-item reports in PDF and Excel formats. Users can also specify a date range to filter the required transaction data, making the reporting process more structured and facilitating the preparation of inventory records. The reporting interface is presented in Figure 6.

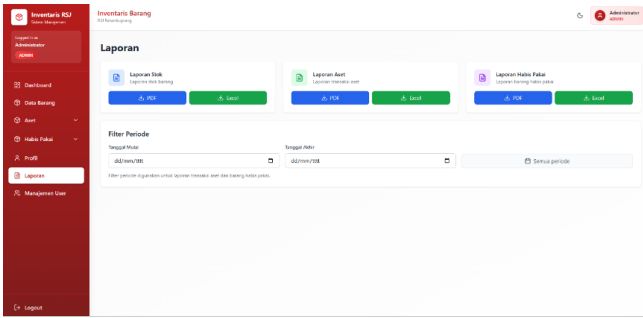


Figure 6. Inventory reporting interface.

The reporting module converts stored inventory records into administrative outputs that can be used for documentation and reporting. The system supports report display and export in PDF and Microsoft Excel formats. This function is important because inventory management does not end with recording transactions; the resulting information must also be communicated and documented.

Compared with the previous manual workflow, the reporting function provides a direct path from transaction records to report output. Users therefore do not need to reconstruct the report by manually combining separate records before exporting the information. At the same time, the study does not establish a numerical reduction in report-preparation time because no pre-implementation time study was conducted. The contribution demonstrated by the research is therefore functional integration: the application provides reporting capabilities using the inventory data already maintained in the system.

3.5 Black Box Testing Results

Black Box Testing was performed after implementation to verify whether the application produced the expected outputs for predefined input scenarios. The test matrix in the thesis covers authentication, item management, incoming assets, outgoing assets, consumable items, reporting, user management, profile management, and logout. The 27 documented scenarios all produced the expected result and were classified as successful.

Table 2. Functional coverage of the 27 Black Box Testing scenarios.

Functional area	No. of scenarios	Main scenarios	Result
Login	3	Valid login; wrong password; empty credentials	Successful
Item data	4	Add; edit; delete; search	Successful
Incoming assets	3	Add asset; generate QR; view detail	Successful
Outgoing assets	3	Scan QR; add transaction; update condition	Successful
Consumables	3	Add; update stock; delete	Successful
Reports	3	Display; export PDF; export Excel	Successful
User management	4	Add; edit; delete; reset password	Successful
Profile and logout	4	Edit profile; change password; logout	Successful

The result demonstrates consistency between the functional requirements and the implemented behavior. Importantly, the test includes both normal transactions and basic validation conditions. For example, the login module was tested using valid credentials, an incorrect password, and empty input; the asset module was tested for QR generation and scanning; and the reporting module was tested for PDF and Excel export. This breadth of functional coverage supports the conclusion that the main workflows identified in the requirements stage were operational at the time of testing.

However, a 100% successful Black Box result should not be interpreted as proof that the software contains no defects. Black Box Testing in this study is scenario-based and limited to the cases documented in the test matrix. It does not cover every possible combination of input, concurrency condition, browser environment, network failure, security vulnerability, or long-term data-integrity condition. Thus, the result provides evidence of functional conformance for the tested scenarios, while further testing would be required for broader reliability and security assurance.

3.6 User Acceptance Testing Results

User Acceptance Testing was conducted after the functional testing stage so that respondents evaluated a system whose main functions had already been verified. The questionnaire contained 24 statements and used a five-point Likert scale consisting of Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree. The statements covered usability, conformity with inventory requirements, operational benefits, functionality, and overall acceptance.

Table 3. Detailed UAT results by assessment aspect.

Assessment aspect	Mean	Percentage (%)
Usability (Q1–Q4)	4.95	99.00
Requirement fit (Q5–Q6)	4.60	92.00
Operational impact (Q7–Q11)	4.52	90.40
Feature functionality (Q12–Q22)	4.67	93.45
Acceptance (Q23–Q24)	4.60	92.00
Overall	4.68	93.50

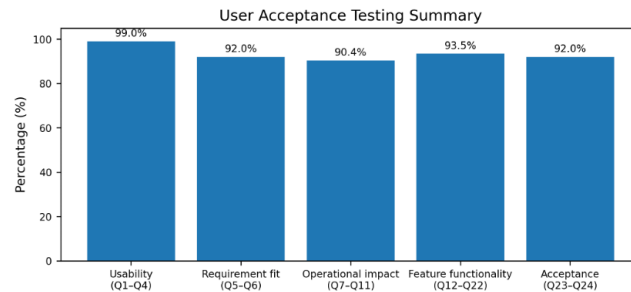


Figure 7. Percentage summary of User Acceptance Testing.

The overall mean of 4.68 corresponds to 93.50% of the maximum Likert score. The original thesis categorizes this result as Very Good and concludes that the system was accepted by the evaluated users. The distribution across aspects provides additional insight into the result. Usability achieved the highest aggregate value at 99.00%, indicating very positive responses regarding interface comprehension, menu accessibility, ease of learning, and information clarity.

The requirement-fit aspect reached 92.00%, suggesting that users considered the implemented functions consistent with the inventory-management requirements identified for the hospital. Feature functionality reached 93.45%, which is consistent with the broad set of modules implemented in the system, including inventory data, asset transactions, QR Code functions, reports, and user management.

The operational-impact aspect obtained the lowest aggregate value among the five grouped aspects, at 90.40%. More specifically, the questionnaire items concerning reduction of manual book recording and reduction of repeated Microsoft Excel recording each received a mean of 4.20, or 84.00%. This result is important for interpretation: users accepted the system strongly, but they did not assign the same level of agreement to the claim that it had already eliminated manual or spreadsheet-based work. The finding may reflect the transition between the previous workflow and the newly implemented system and indicates that organizational adoption and process standardization remain relevant after software deployment.

The UAT therefore complements the Black Box result. Black Box Testing addresses whether the functions behave according to predefined expectations, whereas UAT captures whether users perceive the application as understandable, appropriate, and useful. The combination of successful functional scenarios and a 93.50% UAT score provides evidence from two different perspectives, although neither evaluation measures long-term productivity, return on investment, or reduction in real-world inventory errors.

3.7 Relationship Between System Design and Operational Workflow

The goods receiving process was modeled using Business Process Model and Notation (BPMN) to illustrate the sequence of activities between the hospital and warehouse staff. The process includes goods requests, physical receipt, verification of goods, recording of incoming items, storage, and distribution to the hospital. The BPMN model is presented in Figure 8.

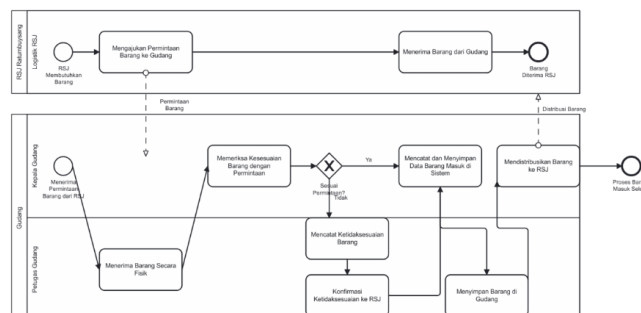


Figure 8. Business Process Model and Notation (BPMN) representing the inventory workflow.

The system design was derived from the operational workflow described during the requirements analysis. The BPMN model represents the movement of goods through requests, warehouse processing, checking, and confirmation, while the application modules provide digital records for the corresponding inventory activities. This relationship between business process and software function is important because an inventory system is useful only when its data structures and interfaces correspond to the actual sequence of organizational activities.

The separation between item master data, asset transactions, consumable stock, users, and reports also supports traceability within the application. For example, an asset can be registered as an incoming unit, associated with a QR Code, subsequently identified during an outgoing transaction, and included in reporting. Similarly, consumable items can be managed according to stock quantities rather than being treated in exactly the same way as durable assets. The implementation therefore reflects the different operational characteristics of the inventory categories identified in the research.

3.8 Comparison With the Initial Problem and Research Contribution

The initial problem identified in the research was not the absence of inventory information, but the fragmented and manual way in which inventory information was recorded and retrieved. The developed system addresses this problem by centralizing the principal inventory records in a web application and connecting them to transaction, identification, user, and reporting functions.

The contribution of the study is consequently practical and system-oriented. It demonstrates how a hospital-specific inventory workflow can be transformed into an integrated web application using React.js, Express.js, PostgreSQL, Prisma ORM, and QR Code. The research also provides a documented testing process in which functional verification and user acceptance are evaluated separately. The inclusion of both Black Box Testing and UAT gives a more complete description of the application's status at the end of development.

At the same time, the evidence should be bounded by the study design. The research focuses on one hospital and evaluates the developed system using the documented test scenarios and user questionnaire. No controlled comparison of pre-implementation and post-implementation processing time, data-entry error rate, or inventory discrepancy rate was reported. Accordingly, the contribution is best understood as a validated application for the specified operational context rather than a quantitative demonstration of efficiency across hospitals.

3.9 Limitations and Future Development

Several limitations should be considered when interpreting the findings. First, the system was designed for the inventory requirements of Ratumbusang Mental Hospital, so its data structures and workflows may require adaptation before being applied to another institution. Second, the evaluation emphasizes functional correctness and user acceptance; it does not include a longitudinal assessment of system use after deployment. Third, the system scope explicitly excludes procurement, financial management, asset accounting, and integration with other hospital information systems.

These limitations provide a direct basis for future development. The original research recommends automatic notifications when consumable stock reaches a minimum threshold or when an asset requires maintenance, integration with other hospital information systems, a mobile application, and a statistical inventory dashboard. These extensions would move the system from primarily recording and reporting inventory information toward more proactive monitoring and decision support.

3.10 Discussion

The developed Inventory Information System integrates the main inventory activities into a single web-based platform, including item management, incoming and outgoing asset recording, consumable inventory, reporting, user management, and QR Code-based asset identification. This integration addresses the limitations of the previous workflow, which relied on manual records and repeated spreadsheet processing. The use of QR Code also provides a direct relationship between physical assets and their digital records, allowing users to identify and retrieve asset information more efficiently. The implementation therefore demonstrates that the proposed system is not limited to digitizing individual inventory forms but provides an integrated workflow for inventory recording, identification, and reporting.

The functional evaluation further supports the implementation results. All 27 Black Box Testing scenarios produced the expected outputs, covering authentication, inventory management, asset transactions, consumable items, reporting, and user management. This result indicates that the principal functions defined during the requirements analysis were successfully implemented. However, the 100% functional success rate should be interpreted within the scope of the tested scenarios and does not necessarily demonstrate the absence of all software defects, particularly with respect to security, performance, network conditions, and long-term reliability.

The UAT results provide an additional perspective from the users. The overall score of 93.50% demonstrates strong acceptance, with usability achieving 99.00% and feature functionality reaching 93.45%. Nevertheless, the operational-impact aspect obtained 90.40%, while the statements concerning reduction of manual book recording and repeated Microsoft Excel recording obtained 84.00%. This finding suggests that although users perceived the system positively, the transition from established manual practices to a fully digital workflow may require further organizational adaptation. Therefore, the system can be considered functionally implemented and well accepted, while its long-term operational benefits should be verified through subsequent measurements of processing time, data-entry errors, inventory discrepancies, and reporting efficiency.

4 CONCLUSION

This study successfully developed a web-based Inventory Information System for Ratumbuysang Mental Hospital using the Waterfall method, integrating React.js, Node.js, Express.js, PostgreSQL, Prisma ORM, and QR Code technology. The system provides integrated functions for item management, incoming and outgoing assets, consumable inventory, user management, reporting, and asset identification. Black Box Testing of 27 scenarios showed that all tested functions produced the expected results, indicating that the implemented system was consistent with the defined functional requirements. User Acceptance Testing involving 24 statements produced an overall mean score of 4.68, equivalent to 93.50%, indicating strong user acceptance. Usability achieved the highest score of 99.00%, while operational impact reached 90.40%, with the items related to reducing manual and repeated Excel recording obtaining 84.00%. These findings indicate that the system was well accepted while also showing that complete elimination of previous manual practices requires further organizational adoption. Future development should focus on automatic stock and maintenance notifications, integration with other hospital information systems, mobile access, statistical dashboards, and longitudinal evaluation using measurable indicators such as processing time, inventory discrepancies, and data-entry errors.

DECLARATIONS

Author Contributions:

Faith Nathaniel Alexander: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing—original draft.

Christie Montolalu: Supervision, Methodology, Validation.

Wisard Kalengkongan: Conceptualization, Data Curation.

Christian A.J Soewoeh: Investigation, Formal Analysis.

Mahardika Inra Takaendengan: Resources, Visualization.

Stephano C.W. Ngangi: Writing—review & editing.

Funding:

The authors received no financial support for the research, authorship, and/or publication of this article.

Conflicts of Interest:

The authors declare that there are no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability:

The data that support the findings of this study are available from the corresponding author upon reasonable request.

AI Use Statement:

During the preparation of this manuscript, the authors used ChatGPT to assist with manuscript structuring and language editing. The authors reviewed and edited the generated content and take full responsibility for the final manuscript.

Additional Information:

No additional information is available for this paper.

REFERENCES

- [1] A. Banks and E. Porcello, *Learning React: Modern Patterns for Developing React Apps*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.
- [2] A. Dennis, B. H. Wixom, and D. Tegarden, *Systems Analysis and Design: An Object-Oriented Approach with UML*, 6th ed. Hoboken, NJ, USA: John Wiley & Sons, 2020.

- [3] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2020.
- [4] A. Fauzi, "Implementasi Sistem Informasi Inventory Berbasis Web (Studi Kasus: CV. Sinar Abadi Cemerlang)," *J. Sist. dan Teknol. Inf.*, vol. 5, no. 1, pp. 29–37, 2022.
- [5] S. R. Octavianus, "Rancang Bangun Sistem Informasi Inventory Barang Berbasis Web (Studi Kasus: Toko Dina Beauty Care)," *J. Teknol. Inf. dan Komp.*, vol. 6, no. 2, pp. 45–53, 2021.
- [6] R. Yuliani, "Sistem Informasi Inventaris Barang di Puskesmas Kemlagi Berbasis Web," *J. Teknol. Inf. dan Komp.*, vol. 8, no. 2, pp. 45–52, 2022.
- [7] A. Siregar, "Penerapan Metode Waterfall pada Pengembangan Sistem Informasi Berbasis Web di Lingkungan Kesehatan," *J. Inform. dan Teknol.*, vol. 10, no. 2, pp. 74–82, 2023.
- [8] R. Murdani, "Aplikasi berbasis web untuk mendukung kinerja organisasi," *J. Teknol. dan Sist. Inf.*, vol. 9, no. 1, pp. 66–74, 2023.
- [9] T. Pramudia, "Desain sistem informasi administrasi rawat inap pasien BPJS IGD di Rumah Sakit Muhammadiyah Bandung dengan metode waterfall," *JMeRS J. Med. Rec. Stud.*, vol. 2, no. 3, pp. 88–95, 2024.
- [10] A. Nugroho, *Pemrograman Web Dinamis dengan JavaScript dan Node.js*. Jakarta, Indonesia: Elex Media Komputindo, 2022.
- [11] The PostgreSQL Global Development Group, "PostgreSQL Documentation," 2025.
- [12] International Organization for Standardization, "Information Technology—Automatic Identification and Data Capture Techniques—QR Code Bar Code Symbology Specification," ISO/IEC 18004:2015, 2015.
- [13] M. Chandra, R. P. Pratama, F. A. Pradana, and A. Bonita, "Chatbot Interaksi Rumah Sakit Menggunakan FFNN," *Indones. J. Data Sci.*, vol. 3, no. 1, pp. 36–44, 2022, doi: 10.56705/ijodas.v3i1.36.
- [14] M. R. Nugroho and I. E. Hendrawan, "Penerapan Algoritma K-Means Untuk Klasterisasi Data Obat Pada Rumah Sakit," *J. Comput. Sci.*, vol. 16, no. 2, pp. 125–133, 2022.
- [15] A. Rahma and A. I. Suryani, "Analisis Penggantian Password User Id Dalam Sistem Rekam Medis Elektronik Guna Menjaga Keamanan Data Rekam Medis Di Rumah Sakit Hermina Arcamanik," *Media Bina Ilm.*, vol. 18, no. 12, pp. 3139–3146, 2024.
- [16] M. Dwi Payana, Z. Musliyana, Z. Ardhian, M. B. Wibawa, D. Ria, and Y. Tb, "Sistem Pendjadwalan Dokter dan Fasilitas Poli pada Rumah Sakit Sultan Iskandar Muda Nagan Raya Berbasis Android Menggunakan Flutter," *J. Inform. Comput. Sci.*, vol. 9, no. 1, pp. 27–33, 2023.
- [17] Z. Mutaqin, B. Imran, and S. Rosida, "Sistem Informasi Penjualan Online (E-Commerce) Berbasis Web Pada Toko Matahari Praya," *J. Comput. Technol.*, vol. 1, no. 2, pp. 29–36, 2023, doi: 10.69916/comtechno.v1i2.46.
- [18] E. Febriani, B. Imran, and R. Muslim, "Sistem Informasi E-Commerce Penjualan Kerajinan Rotan Berbasis Website Pada Desa Loang Maka Kecamatan Janapria," *J. Comput. Technol.*, vol. 1, no. 1, pp. 32–40, 2023.
- [19] M. Raihan and Tommy, "Majority-Class Collapse in E-Commerce Emotion Mining: A Diagnostic Evaluation of LSTM Under Severe Class Imbalance and Noisy Text," *J. Comput. Technol.*, vol. 4, no. 1, pp. 1–13, 2026.
- [20] T. Ramadhani, S. Nabilah, A. Abimayu, and T. Loi, "Pengembangan sistem rekomendasi produk e-commerce menggunakan algoritma collaborative filtering," *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 4, no. 2, pp. 4848–4854, 2025.