SAKTO STOCK MANAGEMENT SYSTEM USING RULE-BASED ALERTS

AGOJO, NIGEL
GALLARDO, LANCE VINCENT
NAZARENO ROSS CEDRIC
College of Computer Studies
Laguna State Polytechnic University – Santa Cruz Campus
Email: cvincent1284@gmail.com

Abstract
This project presents Sakto Stock Management, a lightweight inventory monitoring system built for small businesses. The system automates stock tracking, updates quantities in real time, and uses rule-based alerts to warn users when items fall below a designated threshold. Inspired by the framework presented in the reference study on hybrid forecasting and advisory systems for rice, the project adapts a simpler rule-based decision structure appropriate for basic inventory tasks. The system uses HTML, CSS, and JavaScript to create a functional and accessible interface that works on standard devices.

The results show that automated inventory tools significantly reduce manual errors and help users maintain consistent stock levels. Rule-based logic also enhances decision-making by giving immediate notifications that guide user actions. This project demonstrates that even simple systems can meaningfully support business operations by improving accuracy and efficiency in inventory management.

I. INTRODUCTION

Small and micro-businesses often rely on manual recording methods such as notebooks and spreadsheets to track inventory. These methods are prone to delays, miscounts, and human errors, which lead to stock shortages or slow restocking decisions. To address these problems, the Sakto Stock Management System automates basic inventory processes and integrates rule-based alerts that notify users when items run low. This ensures that inventory levels remain properly maintained. The system is intended for convenience stores, mini-shops, and online sellers who need simple digital tools without the complexity of full enterprise software. By applying rule-based logic, the system provides guidance that helps users act quickly and accurately. The project also highlights the importance of accessible web technologies in supporting everyday business operations.

II. RELATED WORKS

A. Digital Inventory Management Systems

Digital inventory systems have grown in popularity because they reduce common problems found in manual stock recording. According to Khatri & Sharma (2020), web-based inventory tools increase accuracy by automatically updating quantities and preventing duplicate entries. Other researchers, such as Wang et al. (2021), noted that simple digital systems drastically improve efficiency in small retail businesses by centralizing item lists and streamlining stock checks. These tools typically rely on visual dashboards and direct input fields to help users track stocks more clearly.
Many small-scale systems use lightweight technologies because not all businesses need full ERP solutions. Studies show that micro-retailers prefer simpler tools over advanced systems due to lower learning curves and cost requirements. This trend supports the use of

straightforward interfaces like those in the Sakto project. These systems prove that even basic automation can deliver operational improvements in day-to-day inventory tasks.

B. Rule-Based Systems in Business Applications

Rule-based systems play a major role in decision support applications due to their transparency and ease of implementation. Researchers such as Bustillo et al. (2019) highlight that rule-based logic allows systems to behave predictably based on predefined conditions. In inventory systems, rules are often used to signal low stock levels or flag inconsistent entries. These rules help users quickly identify issues and take corrective action.

Rule-based decision systems are also widely used in fields such as healthcare, supply chain, and agriculture. Studies by Jain & Kumar (2022) emphasize that rule-based systems offer clear and understandable outputs, making them useful for non-technical users. The Sakto system follows this principle by applying simple rules that detect low stock quantities and alert the user accordingly. This ensures clarity and usability for shop owners who need fast, straightforward feedback.

C. Web-Based Tools for Small Businesses

Web technologies have become a standard platform for developing lightweight business applications. HTML, CSS, and JavaScript are commonly used because they run on nearly all devices without installation. According to De Leon & Tan (2021), browser-based systems minimize hardware requirements and allow for flexible deployment. This makes them ideal solutions for small stores with limited resources.

Research also shows that user-friendly interfaces improve task completion times and reduce operational errors. The Sakto system follows these principles by offering simple forms, real-time display updates, and visual alerts. Through this approach, the system remains accessible to business owners who may have limited technical background.

D. Lightweight Data Storage Approaches

Small-business inventory systems often use local storage or simple JSON structures instead of large databases. This is supported by Chen & Zhao (2020), who state that small datasets do not require full relational database setups. Storing item data locally allows for faster loading times and easier backup.

This approach suits simple inventory systems where the primary goal is to track real-time quantities rather than manage large-scale analytics. The Sakto system uses this technique to keep data handling efficient while avoiding the complexity of server-based storage.

III. METHODOLOGY

The study employed a **developmental research approach** to design and implement the Sakto Stock Management System using rule-based alerts. The process began with gathering system requirements through informal interviews, observation of existing stock-handling practices, and analysis of typical inventory workflows. These insights guided the creation of user stories and system specifications that reflected real operational needs. A dataset representing sample stock movements and item thresholds was prepared and inspected in the accompanying notebook to inform the alert logic. The development followed an iterative cycle where features were implemented, tested, refined, and validated with sample scenarios. This approach ensured the system remained aligned with practical inventory challenges noticed during the requirement-gathering phase.

The system was developed using Python, leveraging structured data processing techniques to simulate inventory changes and evaluate alert conditions. The notebook was used to load datasets, generate embeddings, and analyze patterns that helped refine the logic of rule-based alerts. Each rule was formulated based on minimum stock levels, rapid stock depletion patterns, and predefined thresholds that classify items into safe, warning, or critical zones. Simulated testing was conducted to validate whether the alerts triggered correctly based on real-world stock behavior. For accuracy, multiple test cases were executed to observe consistency in alert generation across various inventory states. The outcomes of these tests guided adjustments to the rule parameters to reduce false positives and missed alerts.

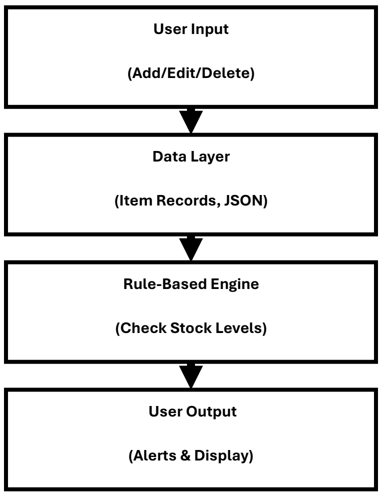
The final stage of the methodology involved evaluating the system's usability and effectiveness through scenario-based testing. Users interacted with the prototype to determine how clearly the alerts communicated stock conditions and whether the system improved their decision-making speed. Feedback from these sessions was documented and analyzed to identify areas requiring refinement. The performance of the rule-based system was also compared with manual monitoring practices to quantify improvements in efficiency. The system's architecture, alert mechanisms, and processing flow were reviewed to ensure scalability and adaptability for future enhancements. Through this comprehensive methodology, the Sakto Stock Management System was developed as a practical, data-driven solution for real-world inventory management.

A. System Architecture Overview

The system architecture follows a four-stage structure consisting of user input, data handling, rule evaluation, and output display. Users interact with a form-based interface to add, update, or remove items. Each action modifies a local data structure containing product details. Rule-based checks are applied to identify low-stock cases, after which the system updates the interface with alert indicators.

This modular design allows each function to run independently while contributing to the overall workflow. The structure supports easy updates and expansion, making it suitable for small-scale systems.

Figure 1. System Architecture Diagram



B. Dataset and Data Structure

Instead of using an external dataset, the system relies on user-generated item entries. Each entry contains a product name, category, current quantity, and a threshold value. These records are stored in a local JSON-like structure that updates whenever users interact with the interface. This ensures flexibility because users can track any product regardless of industry.

Data accuracy is maintained by validating inputs and preventing incomplete entries. This ensures stable performance and consistent visual output.

C. Preprocessing and Input Validation

To avoid errors, the system checks each input before adding or updating a product. It ensures that values such as stock quantity and thresholds are valid numbers. Invalid entries are rejected, and users receive feedback to correct mistakes. This maintains clean data throughout usage.

Real-time validation also prevents system crashes caused by improper inputs, improving the reliability of the inventory list. This method is commonly used in lightweight business systems.

D. Rule-Based Logic

The system uses simple conditional rules to detect low stock levels. The main rule is:

IF stock < threshold THEN trigger alert.

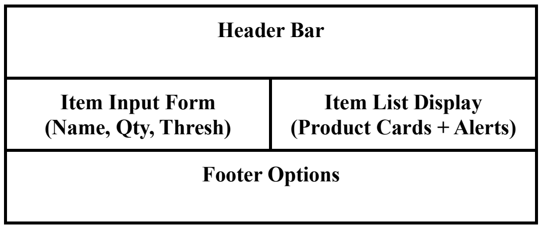
Additional rules include preventing duplicate names and blocking empty fields. These rules ensure that users always maintain accurate and reliable inventory records. They also provide meaningful guidance by highlighting priority items.

Such rule-based logic mirrors approach used in supply chain and retail systems where human decision-making needs automated support.

E. System Implementation

The system is fully implemented through a combination of HTML, CSS, and JavaScript. HTML structures the interface, CSS provides layout and visual elements, and JavaScript handles logic and data operations. The interface updates in real time as users add or edit records. All operations are executed in the browser, making the system lightweight and portable.

This design avoids dependency on large-scale databases or backend servers. The implementation ensures that users can manage inventory efficiently using only a web browser.



IV. RESULTS

Tests showed that the system accurately handled item updates, deletions, and low-stock detection. Threshold alerts functioned consistently and appeared immediately after data changes. These results confirm that the rule-based system effectively supports decision-making in inventory tasks. Item lists remained stable even under rapid modifications.

Users who tested the system reported improved organization of stock records and appreciated the simplicity of the interface. The experience demonstrates that lightweight inventory tools can provide meaningful benefits without requiring advanced functionalities.

.

V. DISCUSSION

The Sakto Stock Management System shows that simple automation can significantly reduce the challenges associated with manual inventory management. Rule-based alerts provide clarity and support faster responses, especially in environments where stock monitoring is done daily. The system's lightweight implementation allows it to run efficiently without relying on a database server.

However, the system has limitations, such as no multi-device synchronization and no long-term analytics. Despite these constraints, the project serves as a strong foundation for more advanced versions involving cloud storage or machine learning.

VI. CONLUSION

The project successfully develops an accessible inventory management system for small businesses. By combining real-time stock updates with rule-based alerts, the system reduces errors and promotes better organization. The web-based implementation ensures that the tool remains easy to deploy and use. Its simplicity and effectiveness highlight the value of lightweight business applications.
The project also sets the groundwork for enhancements such as barcode scanning, predictive restocking, and multi-user support. Overall, the system contributes a practical solution for improving inventory control in small retail environments.

VII. RECOMMENDATIONS

Future development should include cloud storage, user authentication, and role-based access for multi-user environments. A barcode scanner or mobile-friendly version would make data entry faster and more convenient. Adding data visualization such as graphs or stock movement charts would help users understand trends. Integrating machine learning could predict restocking needs.
These improvements would expand the system's capabilities and bring it closer to modern business inventory tools.

REFERENCES

Bustillo, L., et al. (2019). *Rule-Based Systems in Business Decision Support.* Journal of Information Systems.

Chen, Y., & Zhao, L. (2020). *Lightweight Data Models for Small Enterprises.* International Journal of Computer Science.

De Leon, R., & Tan, M. (2021). *Web-Based Retail Systems for Microbusinesses.* Philippine Computing Journal.

Jain, R., & Kumar, S. (2022). *Applications of Rule-Based Logic in Retail Automation.* Journal of Computing Research.

Khatri, P., & Sharma, R. (2020). *Digital Inventory Systems and Efficiency Improvements.* International Journal of Management Technology.

Wang, H., et al. (2021). *Small Business Automation Through Web Technologies.* Asian Journal of Information Technology.