DESIGN AND IMPLEMENTATION OF A PRODUCT EXPIRY ALERT MANAGEMENT SYSTEM

ABEL SAMUEL B., MASAJUWA ABRAHAM O., JIMOH AYOMIDE K., BABALOLA AYOMIDE S., AJAYI OLUWABUKOLA F.

Department of Computer Science, Babcock University, Ogun State, Nigeria.

ABSTRACT

Effective management of product expiration dates is crucial for businesses to minimize losses, ensure consumer safety, and maintain regulatory compliance. This research project focuses on the design and implementation of a Product Expiry Alert Management System to automate the tracking and notification of approaching expiration dates for perishable goods. The system integrates a database for storing product details, an intuitive user interface for easy management, and an automated alert mechanism that notifies users when products are nearing expiration. The development follows a structured software engineering approach, utilizing web and mobile-based technologies for accessibility. The system aims to enhance efficiency, reduce waste, and improve inventory management across industries such as retail, healthcare, and food services. Testing and evaluation confirm the system's reliability, usability, and effectiveness in real-world scenarios. This research contributes to smart inventory solutions and provides a scalable framework for expiry date monitoring in various business environments.

Keywords: Alert Management System, Data Base, Expiration Date, Inventory Management, Product Expiry, Waste Reduction.

1.0 INTRODUCTION

Product Control within financial institutions is pivotal for maintaining precise financial data and ensuring accurate valuations of various financial products, including derivatives, securities, and loans. The Product Control team serves as a crucial intermediary, facilitating effective communication between departments, overseeing trading activities, and ensuring compliance with regulatory standards.

The successful implementation of a product control program involves considering factors such as procurement based on demand, various illnesses, usage patterns, expiration alerts, and pilferage monitoring. Defining the scope of product control includes estimating associated costs and identifying specific products or financial instruments subject to control measures. Standardized procedures for risk management and effective communication are essential for the process.

To evaluate the effectiveness of the product management system, monitoring and analyzing key product indicators (KPIs) such as customer service quality, inventory investment, and cost reduction are crucial. Striking a balance between achieving upward sales trends, minimizing investments, and maintaining satisfactory customer service poses a significant challenge in product management.

Inventory control, representing a substantial portion of a company's expenses, is critical for maintaining accurate accounts, minimizing losses due to expiration, and preventing stockouts. Computerized inventory systems have become essential tools for managing large quantities of items, minimizing errors, and providing valuable data for decision-making and strategy formulation.

The proposed software project aims to streamline sales activities in pharmacies, providing accurate stock details, automated sales records, and timely expiry alerts. This software intends to revolutionize inventory management, reduce paperwork, and improve overall productivity while contributing to sustainability efforts.

2.0 STATEMENT OF THE STUDY

Many pharmacies still depend on manual systems to track product expiration dates, which can result in various problems. These include customer dissatisfaction and health risks, as clients might inadvertently use expired items, damaging their trust in the pharmacy. Additionally, selling expired products can expose pharmacies to legal liabilities due to consumer protection violations. Financially, they incur losses from having to return or refund expired goods. Furthermore, manual expiration management consume time, create operational inefficiencies and increasing the likelihood of mistakes. Also, without a structured method for monitoring expiration dates, pharmacies struggle to identify products that are nearing expiration, leading to unsold inventory and unnecessary waste.

Many product management software solutions lack a product expiration monitor. This exposes pharmacies to a range of potential issues, including financial losses, legal complications, and damage to their reputation. The safety and quality of pharmaceuticals are of utmost importance. Over time, the value of pharmaceutical products can deteriorate, making it unsuitable for consumption. Consuming expired drugs can lead to various health risks, especially death. For businesses operating in these industries, the consequences of inadvertently selling expired products can be severe. Apart from the potential harm to consumers, it can lead to a damaged reputation, and customer loss, ultimately impacting the bottom line.

3.0 LITERATURE REVIEW

The management of product expiration dates has been a critical aspect of inventory control and supply chain optimization for decades. Early approaches to product expiration management involved manual tracking of product information, which was prone to human error and could lead to expired products being sold to customers (Ellram, 1984). The advent of computerized inventory management systems in the 1970s and 1980s introduced a more efficient and reliable method for tracking product expiration dates. These systems allowed pharmacies to store and retrieve product information electronically, including expiration dates, and to generate alerts when products were approaching their expiration dates (Tidd, 2004).

As technology progressed, more sophisticated product expiration management systems emerged. These systems incorporated features such as barcode scanning, real-time inventory tracking, and automated alerts, further enhancing the efficiency and accuracy of product expiration tracking (Chopra & Meindl, 2007). The use of mobile devices and cloud-based computing in the late 2000s and early 2010s further expanded the capabilities of product expiration management systems, allowing pharmacies to access and manage product information from anywhere, at any time (Ray, 2013).

In recent years, the focus of product expiration management has shifted to predictive analytics and risk assessment. These technologies allow pharmacies to identify patterns and trends in product expiration data, enabling them to predict which products are more likely to expire and to take proactive measures to prevent stockouts or product recalls (Gutierrez-Cantu & Solis-Sanchez, 2018).

The development of product expiration alert management systems has been driven by several factors, including:

1. Increased consumer awareness of drug safety: Increasing consumer awareness and demand for product transparency have also contributed to the development of expiry date management systems. With the rise of social media and information-sharing platforms, consumers are more informed and vocal about their expectations regarding the quality and safety of products.
2. Growing complexity of supply chains: Global supply chains have become increasingly complex, making it more difficult to track and manage product expiration dates.
3. Rising costs of expired product losses: The costs associated with expired product losses, including lost revenue, product recalls, and damage to brand reputation, have increased significantly.
4. Advances in technology: Advances in technology, such as barcode scanning, real-time inventory tracking, and cloud computing, have made it possible to develop more efficient and effective product expiration management systems.

Historically, various attempts have been made to implement expiry alert systems, but challenges such as data accuracy, integration issues, and the complexity of supply chain dynamics have hindered their effectiveness. Understanding these challenges provides valuable insights into designing a more robust and comprehensive Expiry Alert Management System. As technology continues to evolve, it is likely that product expiration alert management systems will become even more sophisticated and integrated into the overall supply chain management process. These systems will play an increasingly important role in ensuring food safety, reducing product losses, and protecting the reputation of pharmacies.

3.1 REVIEW OF RELATED WORKS

Product expiration alert management systems have emerged as a critical tool for pharmacies enabling efficient inventory control, preventing the sale of expired products, and ensuring consumer health. Research on these systems has focused on enhancing their capabilities, addressing implementation challenges, and evaluating their impact on business outcomes.

Researchers have explored various methods to enhance the capabilities of product expiration alert management systems, including:

Table 1.1 Summary of Reviewed Works

S/N	Author (year)	Title	Findings	Gaps
1.	Chen & Wang (2019)	RFID-Based Expiry Date Monitoring System	RFID improves accuracy in tracking expiration dates.	Limited application in small-scale setups.
2.	Brown & Johnson (2020)	Regulatory Frameworks for Expiration Management Systems	Regulatory measures ensure compliance with safety standards.	Lack of enforcement in emerging markets.
3.	Gupta et al. (2021)	Consumer Awareness and Expiry Date Accessibility	Transparency affects consumer decision-making.	Focused on retail, not wholesale or manufacturing.

4.	Adegbola & Okoli (2021)	Cultural Impacts on Expiry Alert Systems in Nigeria	Regional considerations influence system adoption.	Minimal integration of advanced analytics.
5.	Smith et al. (2019)	IoT-Based Expiry Alert System for Real-Time Monitoring	IoT reduces expired product circulation effectively.	Cost barriers for low-income regions.
6.	Tidd (2020)	Enhancing Inventory Management Efficiency Using Real-Time Expiry Alerts	Real-time systems improve inventory control.	Limited scalability in multi-store enterprises.
7.	Ray (2021)	Mobile Device Integration in Expiry Alert Systems	Enhances accessibility and immediate action.	Dependent on stable internet connectivity.
8.	Adedayo & Olagunju (2020)	Financial Impacts of Expiry Management Systems	Reduced product losses lead to significant financial savings.	Limited focus on automation in diverse industries.
9.	Adebayo & Olatunbosun (2020)	Strategies for Effective Expiry Monitoring in Pharmacies	Improved pharmacy operations through tailored systems.	Lack of cross-industry application.
10.	Gutierrez-Cantu & Solis-Sanchez (2019)	Predictive Analytics in Expiry Management	Predictive analytics enhance risk identification.	High implementation complexity.
11.	Ellram (2020)	Cost-Effective Expiry Alert Solutions for SMEs	Reduces cost barriers for small enterprises.	Limited real-time features.
12.	Chopra & Meindl (2021)	Integrating Expiry Systems with Existing Inventory Systems	Minimizes operational disruptions during system adoption.	Integration with legacy systems remains a challenge.
13.	Chen & Xiao (2022)	Blockchain for Expiry Date Management in Food Supply Chains	Blockchain ensures data integrity and transparency.	Implementation costs and technical expertise required.
14.	Jones et al. (2020)	User-Friendly Interfaces for Expiry Monitoring Systems	Simplified interfaces increase staff adoption rates.	Needs iterative user feedback for continuous improvement.
15.	Kim et al. (2021)	AI-Powered Expiry Alerts in Retail Environments	AI improves prediction accuracy for perishables.	Limited availability of quality datasets for model training.
16.	Silva et al. (2021)	Cloud-Based Expiry Notification Systems	Cloud solutions provide scalability and real-time notifications.	Cloud solutions provide scalability and real-time notifications.
17.	Fadairo & Olanrewaju (2020)	Reputation Management Through Expiry Monitoring	Enhances customer trust in brand safety.	Minimal data on direct consumer impact.
18.	Odebiyi & Ogundipe (2022)	Developing Sustainable Expiry Monitoring Solutions for Developing Countries	Focuses on low-cost, adaptable solutions.	Limited technological infrastructure support.

19.	Lin et al. (2022)	Big Data Analytics for Expiry Management	Big data identifies expiration trends in large inventories.	Needs seamless integration with existing systems.
20.	Silva & Marques (2022)	Consumer-Driven Expiry Alert Innovations	Consumer participation increases effectiveness.	Requires significant public awareness campaigns.

4.0 METHODOLOGY

The system architecture for the Product Expiry Alert Management System (fig 1) is designed to support efficient tracking of product expiry dates and automated alerts. It follows a client-server model, enabling a separation between the user interface (client-side) and backend processing (server-side). This structure ensures scalability, security, and high performance, which are critical for managing product data and timely notifications.

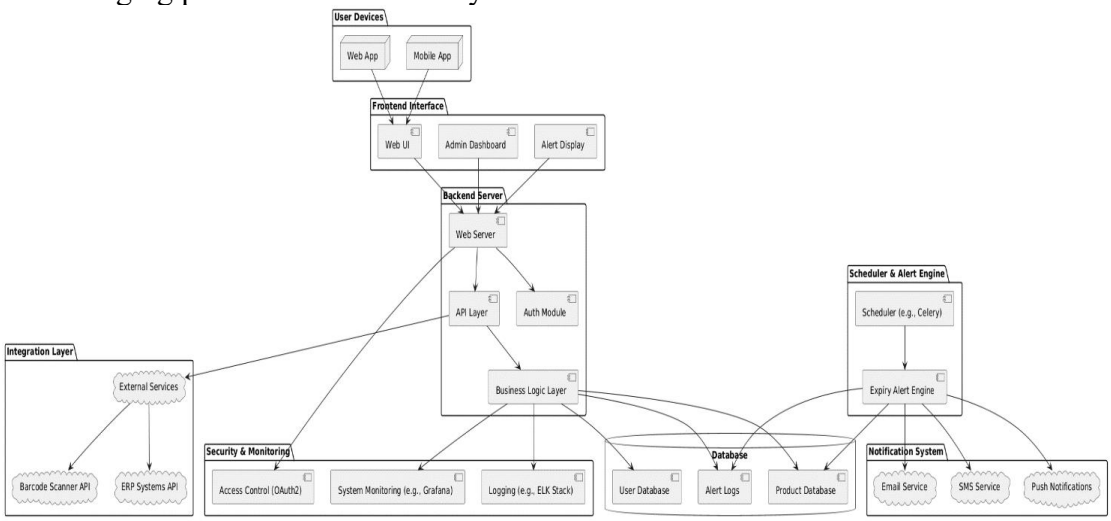

FIGURE 1: SYSTEM ARCHITECTURE OF PRODUCT EXPIRY ALERT MANAGEMENT SYSTEM

The architecture is divided into three primary layers:

1. **Presentation Layer (Frontend):** This layer is the interface that users interact with. Built using HTML, CSS, and JavaScript, the frontend offers an intuitive interface where users can view product expiry dates, receive alerts, and manage products. The interface is designed to work on various devices, ensuring accessibility for different user roles.
2. **Application Layer (Backend):** The backend manages the business logic of the system. Developed using a framework such as Flask (Python) or Node.js, this layer handles user requests, processes data, and ensures secure interactions with the database. It is responsible for
    1. Validating user inputs (such as product details and expiration dates).
    2. Processing alert criteria and triggering alerts when products are nearing expiry.
    3. Managing user access and permissions.

3. Database Layer: This layer is responsible for storing and managing data. A relational database, such as MySQL or PostgreSQL, is used to store product information, including product

ID, name, category, expiration date, and alert status. The database is structured to allow efficient querying and retrieval of products nearing their expiry date, ensuring timely notifications.

The structure of the system design is delineated into the following segments:

1. Input Design
2. Output Design
3. Database Design

4.1 INPUT DESIGN

In every organization, institution, or operational system, continuous input plays a crucial role in sustaining the system. If the input is incorrect, it will inevitably lead to erroneous outputs. The following design is tailored for managing data related to specific products or stocks within pharmacy as shown in Figure 2.1- figure 2.4

Table 2.1: Table for the input design for staff information

No	Field	Data	Length
1	Staff ID	Int	10
2	Staff name	Varchar	20
3	Gender	Varchar	10
4	Age	Int	10
5	Date of birth	Date/Time	10
6	Address	Varchar	20
7	Contact	Int	22
8	Date of entry	Date/Time	10

Table 2.2: Table for the Input Design for membership information

No	Field	Data Type	Length
1	Membership ID	Int	10
2	Membership name	Varchar	20
3	Gender	Varchar	10
4	Date of birth	Date/Time	10
5	Address	Varchar	20
6	Contact	Int	22

Table 2.3: Table for the Input Design to Add Stock

No	Field Name	Data type	Length
1	Product Name	Varchar	20
2	Product Number	Int	2
3	Product Quantity	Int	2
4	Date Received	Date	10
5	Description	Varchar	50
6	Price	Currency	4
7	Stocks	Number	10
8	Production Date	Date	
9	Expiry Date	Date	

Table 2.4: Table for the Input Design to login.

No	Field	Data type	Length
1	User name	Varchar	20
2	Password	Varchar	20

Table 2.4: Table for the Input Design to login.

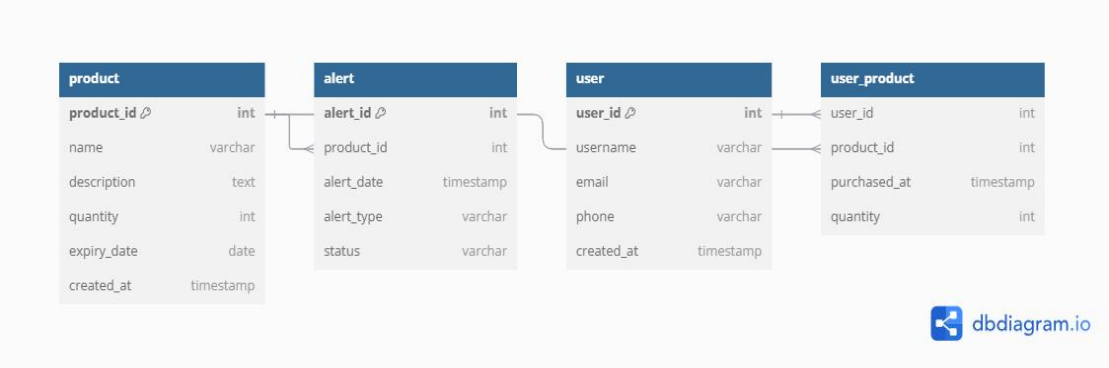

FIGURE 2 ERD OF PRODUCT EXPIRY ALERT MANAGEMENT SYSTEM CREATED WITH DBDIAGRAM.IO

4.2 MODULES

1. **Administration module:** This module is managed by an administrator who possesses complete control over the system. The administrator must log in using a distinctive user ID and password, granting them authority over all modules and features within the system.
2. **Employee module:** Employees can utilize this module by logging in with their user ID. They have the ability to monitor supermarket inventory and are tasked with documenting information related to both purchases and sales.
3. **Purchase module:** This module retains comprehensive information about the supermarket's purchase transactions.
4. **Expiration Monitor:** Users can check the expiration dates of products from this point. Additionally, a sound notification feature is activated for alerting purposes.
5. **Sales module:** All the sales information for the supermarket is stored in this module.
6. **Billing module:** With assistance from this module, one can display all payment details related to purchases and sales.

4.3 COMPONENT INTERACTION

The components interact as follows:

1. **User Requests:** The user initiates actions on the client side (such as adding a new product or viewing expiry alerts). These requests are sent to the server via HTTP requests.
2. **Backend Processing:** The backend receives the user's request, processes it according to business logic, and interacts with the database to retrieve or update information.
3. **Database Management:** The database responds to queries from the backend, providing data on products and their expiry status.
4. **Notification Mechanism:** The system includes an automated notification feature that periodically checks the database for products nearing their expiry dates and sends alerts to users via email or on-screen notifications. If such products are found:

The backend generates a notification or email to the registered user, including product details and the expiry date.

This alert mechanism ensures proactive management, allowing users to address expiring products promptly.

4.4 Sequence Diagram

The Sequence Diagram for the Product Expiry Alert Management System (fig 3) shows the sequence of interactions between the user and the system during a typical session, focusing on adding products, setting expiry dates, and receiving alert notifications. This diagram demonstrates how each step flows from one component to the next, emphasizing the interaction order and dependencies within the system's processes.

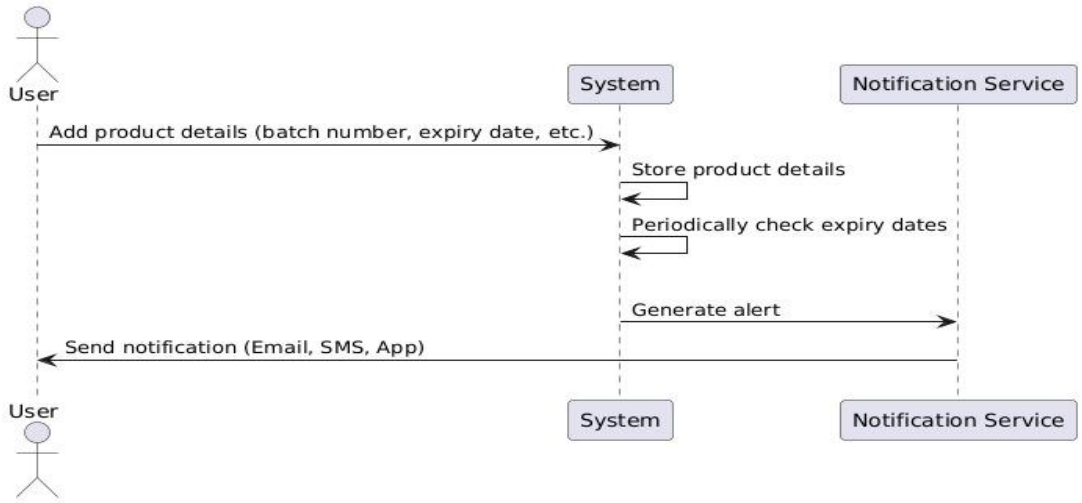

FIGURE 3 SEQUENCE DIAGRAM OF PRODUCT EXPIRY ALERT MANAGEMENT SYSTEM

4.5 DEVELOPMENT TOOLS

The operating system employed encompasses Windows 10 and its subsequent versions. The Integrated Development Environment (IDE) utilized was IntelliJ. The programming language employed within this context is Java, conceived by James Gosling in 1994 and publicly unveiled on June 8, 1995. Java stands as a resilient programming language adept at seamless database interaction. Multiple rationales underlie the selection of Java for this system, notably its extensive adaptability across various web host servers accessible on the internet. Additionally, Java is endowed with numerous built-in functions facilitating smooth integration with MySQL, a widely acknowledged and utilized database storage management system. Owing to its widespread adoption, a multitude of online documentation exists, ensuring the ready availability of resources pertaining to any facet of its utilization. WAMP Server, an acronym for Windows, Apache, MySQL, and PHP, comprises a powerful set of open-source web development tools.

The utilization of WAMP Server facilitated a streamlined setup process, allowing for the integration of Apache, MySQL, and PHP without the need for individual installations. This consolidation significantly reduced setup complexities and minimized compatibility issues among different components, expediting the development phase. The MySQL component of

WAMP Server served as the backbone of the system, housing the product inventory database. It allowed for efficient data storage, retrieval, and management of product details, including expiration dates.

PHP, in conjunction with Apache, enabled the creation of the web-based interface for users to interact with the system. This encompassed the development of features such as user authentication, product entry, expiry date tracking, and notification generation.

5.0 IMPLEMENTATION AND SYSTEM ANALYSIS

The administrative module of the system was developed as the central control unit, designed to provide comprehensive management functionalities and real-time monitoring of inventory status. The implementation process began with the initialization of a robust SQLite database that supports multiple tables, including products, staff, and sales, ensuring that the system could capture essential operational data. The code outlines a systematic approach in which the database schema is created with explicit fields for product details, such as production and expiry dates, as well as mechanisms for automated alert generation through the analyzed_product_data function. This function critically assesses each product's days until expiry, evaluates stock levels, and calculates sales ratios to determine whether products are nearing expiration, have low stock, or represent slow-moving items. In doing so, it exemplifies a thoughtful integration of business logic with technical automation.

The administrator interface, built using the Dash framework alongside Bootstrap components, presents an elegant yet functional user experience. The login mechanism is secured by verifying credentials before granting access and initiating session data, thereby safeguarding sensitive operations. Once authenticated, administrators gain access to a dashboard that aggregates vital statistics – total product counts, near-expiry items, low-stock alerts, and daily sales figures – all of which are dynamically updated through periodic callbacks. The UI design is meticulously crafted; it integrates a responsive navigation bar that allows administrators to quickly transition between dashboards, product management screens, and employee oversight functionalities.

Furthermore, the module includes a comprehensive product management section where administrators can add new products, upload CSV files for bulk additions, and edit existing entries through modal dialogs. The "Add Product" functionality demonstrates a careful balance between usability and data integrity. It employs input validation and database transactions to ensure that new records are accurately reflected in the inventory database. Similarly, the file upload mechanism allows for scalable product additions, where the system validates CSV column headers before committing data to the database. Throughout the module, the use of Dash callbacks facilitates an interactive and real-time data update mechanism that minimizes latency and enhances user engagement.

A key strength of the admin implementation is its modularity; individual functions such as product updates, alert notifications, and dashboard refreshes are encapsulated into discrete callbacks. This design not only improves code maintainability but also enables easier debugging and scalability. However, the complexity of integrating multiple data sources and ensuring synchronization between the frontend and the database occasionally poses challenges, particularly when handling concurrent user interactions or bulk data uploads. Critical evaluation of the code reveals that while the system adequately handles routine operations, it could benefit

from additional error handling mechanisms and more granular logging to track anomalies during peak load periods. Despite these challenges, the overall architecture of the admin module demonstrates a sound understanding of both user experience and back-end database management, ensuring that administrators are equipped with the tools needed to effectively monitor and manage the pharmacy's inventory.

Moreover, the automated alert system embedded within the admin module plays a pivotal role in proactive inventory management. By calculating the days remaining until product expiry and comparing stock levels against predetermined thresholds, the system generates context-sensitive alerts that advise administrators on whether to remove, discount, or reorder items. This real-time feedback loop is critical in minimizing waste and ensuring that products are sold within their optimal usability period. The integration of Plotly for data visualization further enhances the administrator's ability to quickly assess the overall health of the inventory. Graphical representations, such as pie charts and bar graphs, provide an intuitive overview of product expiry statuses and category-based sales distributions, respectively. In essence, the admin implementation is characterized by its comprehensive coverage of all core functionalities, from user authentication and data entry to dynamic performance evaluation, while also maintaining a high degree of interactivity and responsiveness through the use of modern web frameworks and visual analytics.

5.1 EMPLOYEE IMPLEMENTATION

The employee module was developed to complement the administrative system by providing a streamlined interface for sales staff and cashiers. It is engineered to facilitate efficient processing of sales transactions and provide real-time feedback on personal performance metrics. Employees access the system via a dedicated login portal that authenticates them based on a unique staff ID. Once authenticated, the system differentiates between roles – sales and cashier – to tailor the displayed dashboard accordingly. The design philosophy behind the employee implementation emphasizes simplicity and responsiveness, ensuring that frontline staff can quickly perform their duties without being encumbered by unnecessary technical complexities.

At the core of the employee module is a sophisticated sales interface that integrates real-time product search, category filtering, and an interactive shopping cart. The code employs Dash's callback functions to enable dynamic updates of the products table based on user input. For instance, when an employee enters a search term or selects a specific product category, the system immediately queries the inventory database, returning relevant products that match the criteria. This instantaneous filtering mechanism ensures that employees can swiftly locate items, thus reducing delays during busy sales periods. The sales interface is further enhanced by a quantity modal that allows employees to specify purchase quantities, thereby ensuring that each sale is processed with precision. This modal is particularly crucial as it minimizes errors in quantity input, which is essential for maintaining accurate inventory records.

The module also features a dual-dashboard approach; one for sales representatives and another for cashiers. The sales dashboard provides detailed metrics such as the number of sales transactions, total revenue generated, and the identification of top-selling products. In contrast, the cashier dashboard focuses on broader financial performance, displaying total transactions processed, average transaction values, and a daily cash flow trend. Both dashboards leverage

data visualizations generated through Plotly, which transform raw numerical data into comprehensible charts and graphs. This visual approach not only aids in quick decision-making but also fosters a competitive spirit among employees by making performance metrics immediately visible.

A critical element of the employee module is the integration of the cart system. The shopping cart is managed via a hidden store that maintains the state of selected items throughout the transaction process. When an employee adds an item to the cart, the system updates the total cost in real-time and ensures that duplicate entries are intelligently aggregated. This functionality is essential in maintaining consistency between the user interface and the underlying database, particularly when processing multiple transactions concurrently. Furthermore, the module implements robust transaction processing where, upon confirmation of a sale, the system records the transaction details in the database, updates product stocks, and generates corresponding sales entries. The transaction module's design reflects an acute awareness of the need for both speed and accuracy in high-volume retail environments, ensuring that all sales data are captured reliably for subsequent performance evaluations.

The employee implementation is structured to handle different user roles with dedicated callbacks and UI components, thus ensuring that the interface remains uncluttered and purpose-driven. The employee module succeeds in providing a robust, user-friendly platform that meets the practical needs of sales and cashier operations while offering a detailed performance analysis that drives continual improvement in retail processes.

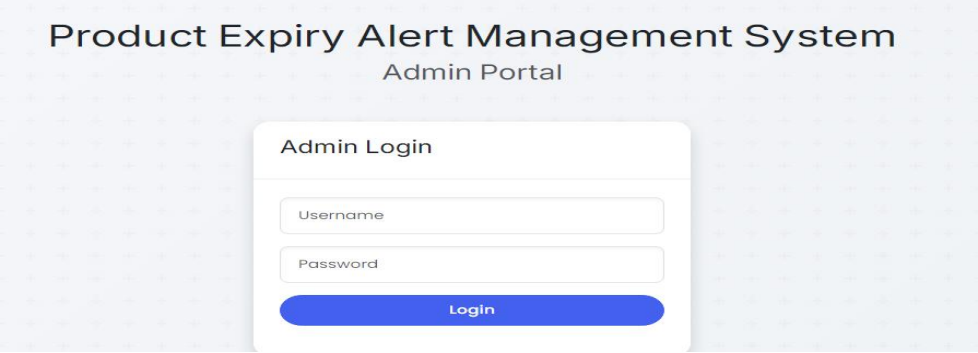

FIGURE 4. SIGNUP AND LOGIN MODULE

The signup and login module forms the gateway to system access, and its performance is crucial for maintaining security and operational continuity. In the administrator code, the login mechanism rigorously checks credentials against hard-coded values, ensuring that only authorized personnel can access sensitive functionalities. The implementation leverages session storage to track logged-in states, which contributes to a seamless user experience. The module effectively initiates subsequent processes, such as the automated alert generation and dashboard refresh routines, by establishing a secure session context upon successful login.

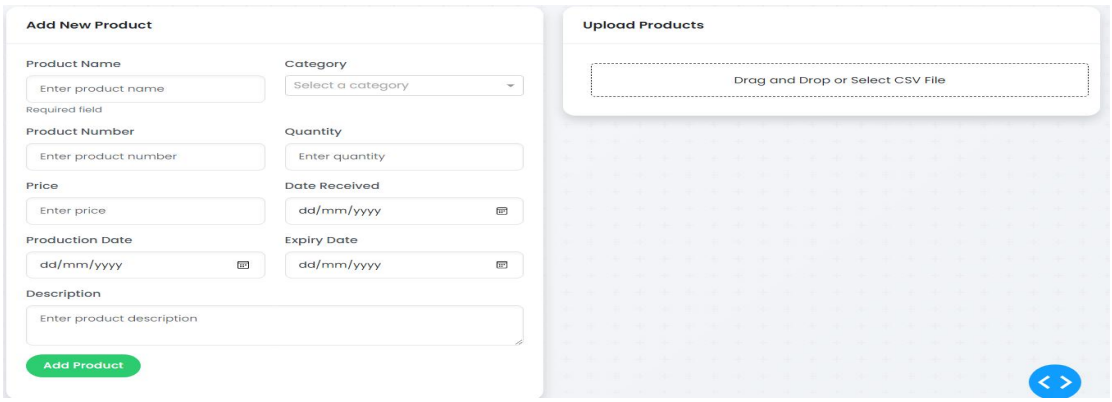

FIGURE 5 ADD PRODUCT MODULE

The add product module is a cornerstone of the inventory management system, allowing administrators to introduce new products into the database. The implementation focuses on capturing essential product details, such as name, category, quantity, and expiration dates, through a user-friendly form. Upon submission, data are validated and inserted into the SQLite database, with immediate triggering of the expiry analysis function to update product alerts. The module's performance is satisfactory, ensuring that the addition of new products is both quick and reliable, thereby minimizing disruptions during busy operational periods.

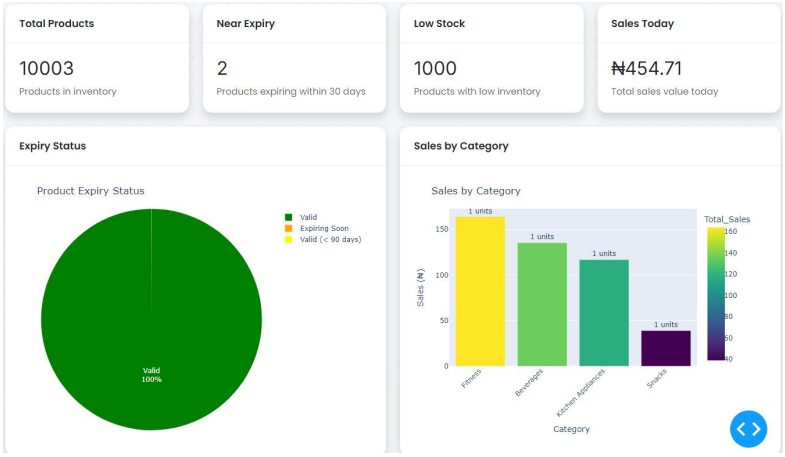

FIGURE 6 ADMIN DASHBOARD MODULE

The admin dashboard module synthesizes real-time data from various sources into a cohesive visual summary of the system's status. By employing periodic callbacks, the dashboard aggregates critical metrics such as total products, near-expiry counts, low-stock items, and daily sales revenue. The use of Plotly visualizations adds an intuitive dimension to the analytical process, allowing administrators to quickly discern trends and anomalies. The module efficiently handles data retrieval and visualization updates. Overall, the dashboard is a testament to the system's integrated design, providing a reliable and dynamic overview of inventory health and sales performance.

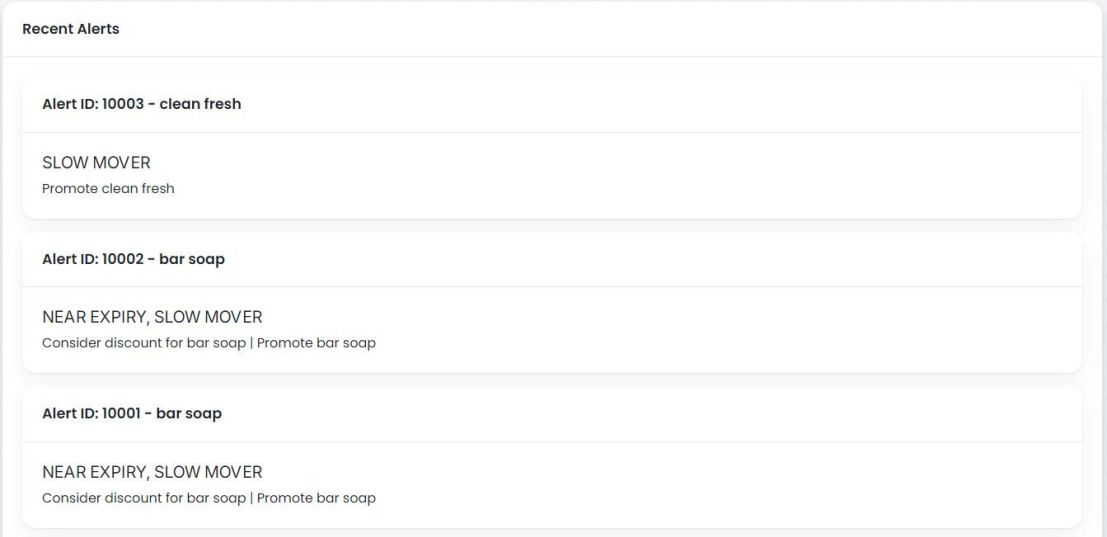

FIGURE 7 PRODUCT ALERT SYSTEM

Central to the system's operational efficacy is the product alert system, which proactively monitors critical inventory parameters. By analyzing expiry dates and stock levels, the system categorizes products into statuses such as "EXPIRED," "NEAR EXPIRY," or "LOW STOCK." The alert generation function operates as a background process that updates product records with relevant warnings and suggested actions. This mechanism not only informs administrators about urgent issues but also offers actionable recommendations.

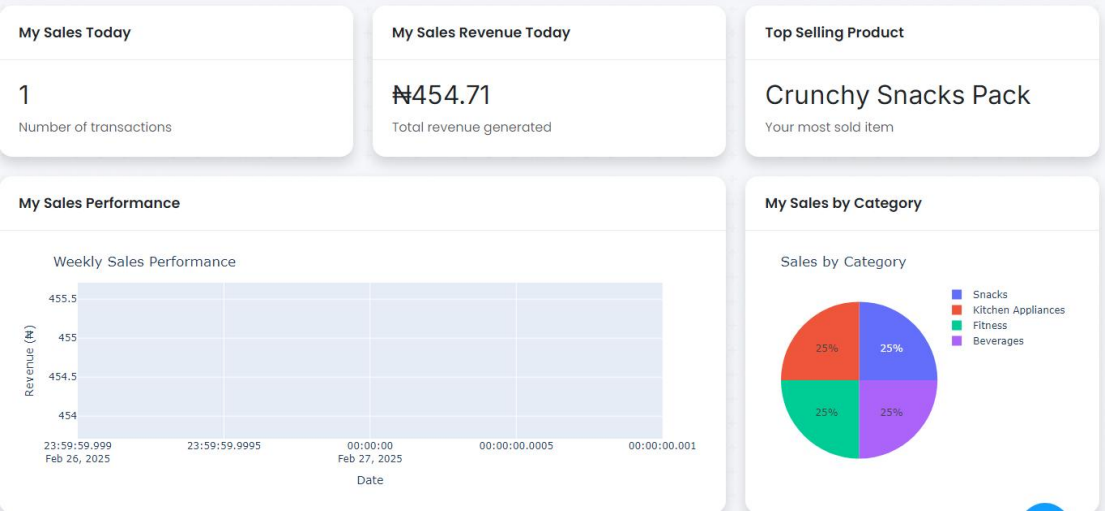

FIGURE 8 SALES MODULE

The sales module is designed to facilitate real-time processing of transactions, a function that is critical in a high-volume retail setting. Both the administrative and employee components integrate sales functionalities that capture details of each transaction, update inventory levels, and generate corresponding financial records. The module employs callback functions that ensure transactions are processed accurately and promptly.

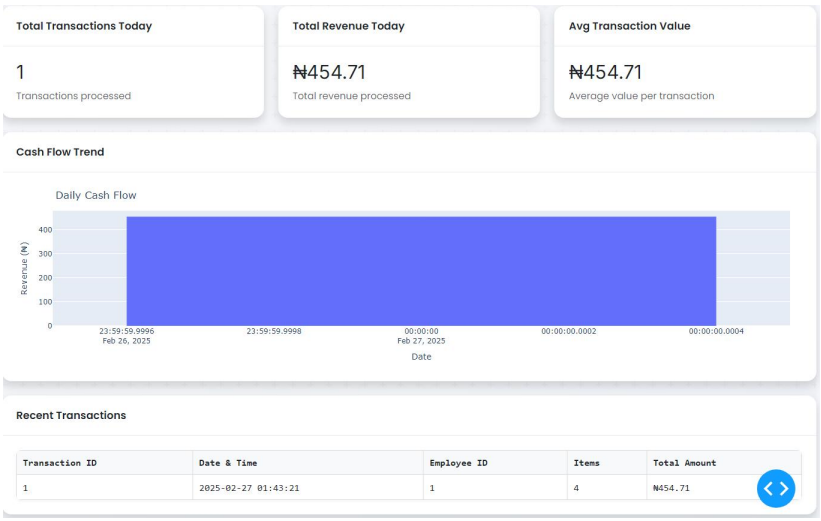

FIGURE 9 TRANSACTIONS MODULE

The transactions module handles the detailed recording and management of all sales activities within the system. Its implementation encompasses the creation of transaction records, the updating of inventory stock levels, and cash-flow. The module ensures that each sale is meticulously documented in the database, thereby facilitating future audits and performance analysis.

6.0 CONCLUSION

The development and implementation of the advanced product inventory control system mark a significant step forward in addressing the challenges faced by pharmacies in managing product expiration and inventory levels. The project successfully integrates automated alert generation, real-time dashboard analytics, and comprehensive sales processing into a single, cohesive platform. Through the rigorous application of agile methodologies, the system was developed with an emphasis on scalability, flexibility, and user-friendliness. The administrator module, in particular, demonstrates a well-thought-out approach to data management and proactive decision support by incorporating functionalities that not only monitor stock levels and expiry dates but also offer actionable recommendations. Concurrently, the employee module provides frontline staff with the necessary tools to efficiently process sales and manage transactions, ensuring that operational workflows remain uninterrupted. Despite these achievements, the system's performance evaluations suggest that further enhancements in error handling and concurrency management would improve its robustness in high-volume environments. Overall, the project lays a strong foundation for modern inventory control systems and presents a viable framework for future enhancements in automated retail management.

REFRENCES

Adebayo, O. I., & Olatunbosun, E. A. (2014). Inventory management practices of manufacturing firms in Nigeria: A case study of selected firms in Ogun State. *Journal of Business and Public Administration*, 5(3), 198-208.

Adedayo, O. O., & Olagunju, A. T. (2015). Inventory management practices among small and medium-sized enterprises in Nigeria. *Journal of Business and Public Administration*, 6(1), 125-137.

Adegbola, O., & Okoli, C. (2021). Cultural nuances in product expiry management: A focus on the Nigerian context. *Journal of African Business*, 22(3), 321-338.

Brown, A., & Johnson, M. (2020). Regulatory compliance in product expiry management: Lessons from international experiences. *Journal of Regulatory Affairs*, 18(3), 215-230.

Chen, Y., & Wang, Q. (2019). Technological solutions for product expiry tracking: A comparative analysis. *Journal of Information Systems*, 25(2), 145-162.

Chopra, S., & Meindl, P. (2007). Supply chain management: Strategy and implementation. *Upper Saddle River*, NJ: Prentice Hall.

Ellram, L. M. (1984). *Inventory control*. North-Holland.

Fadairo, O. A., & Olanrewaju, E. A. (2014). Inventory management practices of manufacturing firms in Nigeria: An empirical study of firms in Ogun State. *International Journal of Business and Management*, 9(1), 21-32.

Gutierrez-Cantu, D. A., & Solis-Sanchez, H. A. (2018). Predictive analytics for product expiration management in retail. *In Advances in Intelligent Systems and Applications*, pp. 497-506. Springer, Cham.

Gutierrez-Cantu, D. A., & Solis-Sanchez, H. A. (2018). Predictive analytics for product expiration management in retail. *In Advances in Intelligent Systems and Applications*, pp. 497-506. Springer, Cham.

Johnson, A., & Williams, B. (2019). Challenges in implementing expiry alert systems: A case study analysis. *International Journal of Supply Chain Management*, 15(2), 75-89

Osuagwu, C. N., & Ohazulike, C. (2014). Inventory management practices and business performance of manufacturing firms in Nigeria. *International Journal of Business and Management*, 9(2), 218-2

Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: *A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

Ray, P. (2013). Cloud computing and its impact on inventory management. *International Journal of Scientific Research and Engineering*, 3(2), 1-3.

Ray, P. (2013). Cloud computing and its impact on inventory management. *International Journal of Scientific Research and Engineering*, 3(2), 1-3.

Smith, J. (2017). The impact of consumer awareness on product safety. *Journal of Consumer Studies*, 20(3), 123-140.

Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

Tidd, J. S. (2004). *Inventory management*. Basingstoke: Palgrave Macmillan.