iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

SMART INVENTORY MANAGEMENT SYSTEM WITH BARCODE

Bankuru Jaswanth, Dusakanti Harshitha, Samala Nilohitha, Tanay Roy.
Guide: Mrs. M. Anusha, Assistant Professor
Department of Electronics and Computer Engineering
J.B. Institute of Engineering and Technology, Hyderabad, Telangana, India

ABSTRACT
The Smart Inventory Management System with Barcode is a Java-based application designed to automate and optimize inventory operations in retail stores, supermarkets, pharmacies, and warehouses. Traditional inventory systems rely on manual data entry, which often leads to errors, stock mismanagement, and time-consuming billing processes. This system integrates barcode technology for fast and accurate product identification. Developed using Core Java, Swing/JavaFX, JDBC, and MySQL, the system assigns each item a unique barcode enabling instant retrieval of product details during sales and stock management. When a barcode is scanned, the system fetches product details, updates inventory in real time, and generates a bill. Features include low-stock alerts, product management, sales tracking, and report generation—reducing human intervention, minimizing errors, and improving operational efficiency.

Keywords:
Inventory Management, Barcode, Java, MySQL, JDBC, Automation, Real-time Tracking

1. INTRODUCTION

Inventory management is a fundamental aspect of any business organization that deals with physical goods. Efficient inventory management helps organizations minimize costs, improve customer satisfaction, and enhance overall operational efficiency. However, traditional inventory management systems are largely manual—relying on paper-based records or spreadsheets—which are prone to errors, inefficiencies, and delays.

With the advancement of information technology, modern inventory systems have evolved to incorporate automation and digital tools. Barcode technology enables quick product identification and retrieval by encoding product data in a scannable visual pattern. This eliminates manual entry errors and significantly improves processing speed.

The Smart Inventory Management System with Barcode integrates barcode scanning technology with a robust Java-based software application. The objective is to automate inventory operations including product entry, stock management, billing, and report generation. The system uses Java Swing for the GUI, MySQL for data storage, and JDBC for database connectivity—providing a scalable, reliable, and platform-independent solution suitable for retail stores, supermarkets, warehouses, and distribution centers.

2. LITERATURE SURVEY

Traditionally, inventory management was performed using paper-based records or basic spreadsheet applications, which were highly prone to errors and required significant human effort. The introduction of computerized database systems (MySQL, Oracle, SQL Server) replaced manual record-keeping with digital storage, improving data retrieval and organization—though manual data entry remained a source of errors.

Barcode-based inventory systems addressed this limitation by automating product identification. Research studies confirm that barcode-based systems improve operational efficiency by reducing data-entry time and minimizing human errors. Java-based implementations using JDBC for database connectivity have become popular in enterprise-level inventory applications, with Java Swing and JavaFX enabling user-friendly graphical interfaces.

While RFID technology offers advantages such as bulk processing and no line-of-sight requirement, barcode systems remain preferred for small and medium-scale applications due to cost-effectiveness. The literature also highlights the importance of real-time tracking, low-stock alerts, secure authentication, and reporting and analytics for informed decision-making. The proposed system builds upon these findings by unifying barcode scanning, real-time inventory tracking, and automated billing in a single, cost-effective Java application.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

3. SYSTEM DESIGN

3.1 Architecture Overview

The system follows a three-tier architecture comprising the Presentation Tier (Java Swing GUI), the Business Logic Tier (Java application with JDBC), and the Data Tier (MySQL database). The architecture diagram in Fig. 1 illustrates the component interactions and data flow across these tiers.

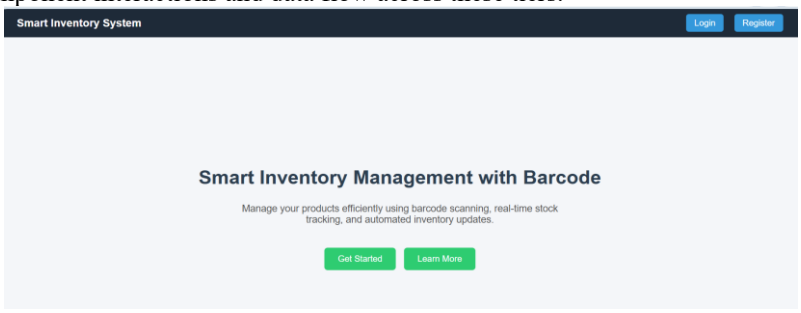

Fig. 1. System Architecture Diagram

3.2 Data Flow

Fig. 2 depicts the data flow diagram showing how product data, barcode scans, and user interactions trigger corresponding database operations and GUI updates across the system modules.

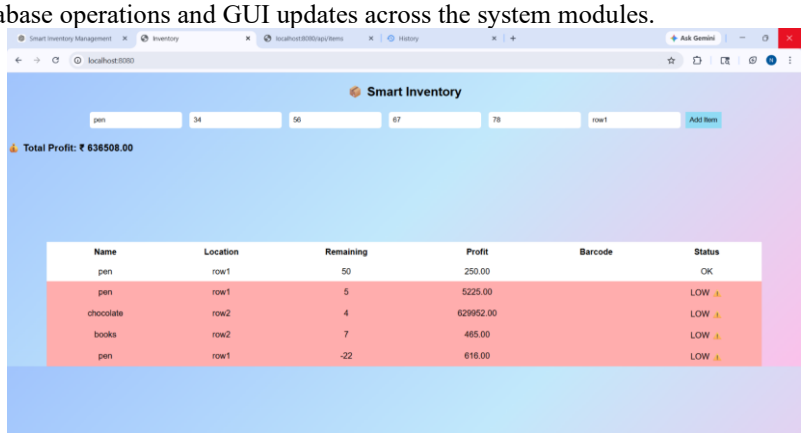

Fig. 2. Data Flow Diagram

3.3 Modules

The system is organized into five functional modules:

- Admin Module – product management, user management, report generation.
- User Module – barcode scanning, billing, stock queries.
- Barcode Module – barcode generation (ZXing library) and scanning.
- Inventory Module – real-time stock level tracking and low-stock alerts.
- Report Module – sales reports, stock reports, and transaction histories.

3.4 Database Design

The database consists of two primary entities: User (user_id, username, password) and Product (product_id, name, barcode, quantity, price). A One-to-Many relationship ('manages') links users to products, supporting multi-user operations with role-based access control.

3.5 UML Diagrams

Fig. 3 presents the Use Case Diagram showing primary actor interactions, and Fig. 4 shows the Class Diagram depicting module relationships.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Fig. 3. Use Case Diagram

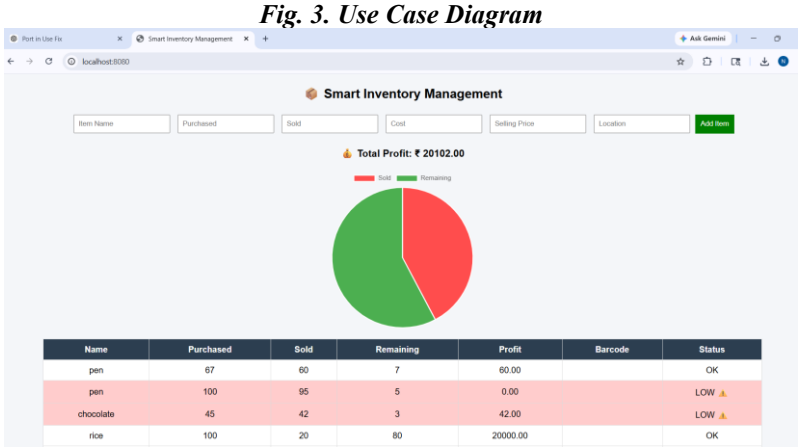

Fig. 4. Class Diagram

4. IMPLEMENTATION

4.1 Technology Stack

The application is developed using the following technologies:

- Language: Core Java (JDK 11+) with object-oriented design principles.
- Frontend: Java Swing for GUI with a modular panel-based layout.
- Backend / Connectivity: JDBC for database operations.
- Database: MySQL for persistent storage of products, users, and transactions.
- Barcode Library: ZXing (Zebra Crossing) for barcode generation and decoding.

4.2 Barcode Integration

Each product is assigned a unique barcode at registration time using the ZXing library, which generates Code 128 barcodes embedded into the system's product records. During billing or stock operations, the barcode scanner sends the decoded string to the application, which queries MySQL via a JDBC PreparedStatement to retrieve the matching product record instantly. This eliminates manual key-entry and reduces lookup time significantly.

4.3 Real-Time Inventory Tracking

Whenever a product is sold, the corresponding quantity is decremented in the database and the GUI refreshes automatically. If stock falls below a configurable threshold, the system displays a low-stock alert dialog, prompting restocking. This ensures that business owners always have an accurate view of available inventory without manual reconciliation.

4.4 Security

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

User authentication is enforced at login via SHA-256 hashed password validation. Role-based access restricts admin-only operations (product deletion, report generation) from regular users, ensuring data integrity and accountability.

5. TESTING AND VALIDATION

The system was validated across four dimensions: unit testing of individual modules, integration testing of the barcode-to-database pipeline, functional testing of all use cases, and system testing under multi-user concurrent access.

Test Dimension	Outcome
Unit Testing	All modules passed independently
Integration Testing	Barcode scan → DB retrieval validated
Barcode Accuracy	Correct product fetched in all test scans
Stock Update	Real-time decrement confirmed post-billing
Low-Stock Alert	Alert triggered below threshold
Authentication	Unauthorized access blocked correctly
Report Generation	Sales and stock reports generated accurately

Table I. Testing Summary

6. RESULTS AND DISCUSSION

The implemented Smart Inventory Management System successfully integrates barcode scanning with real-time inventory operations. Key results from the implementation and testing phase are:

- Barcode scanning reduced average product lookup time compared to manual entry.
- Real-time stock updates eliminated end-of-day reconciliation discrepancies.
- Low-stock alerts prevented stockouts during testing scenarios.
- The billing module generated accurate invoices with automatic total computation.
- Multi-user access control maintained data consistency with no conflicts observed.

The system was tested on a standard hardware configuration (Intel Core i5, 8 GB RAM) running MySQL 8.0 and JDK 11. All functional requirements were met and the GUI responded within acceptable latency bounds for all tested operations, confirming suitability for small-to-medium scale retail deployment.

7. CONCLUSION

This paper presented a Smart Inventory Management System with Barcode, implemented using Java, MySQL, and the ZXing barcode library. The system addresses key limitations of traditional manual inventory approaches—namely inaccuracy, slow processing, and lack of real-time visibility—by automating product identification, stock tracking, and billing. Testing confirmed that the system meets all functional and non-functional requirements, providing a reliable and scalable foundation for inventory management in retail environments.

FUTURE WORK

Future enhancements include cloud-based data storage for remote access, a mobile application for on-device barcode scanning, RFID integration for bulk and contactless processing, graphical analytics dashboards, and integration with accounting and billing platforms for end-to-end business management.

REFERENCES

[1] K. C. Laudon and J. P. Laudon, Management Information Systems, 16th ed., Pearson, 2020.
[2] GS1, Barcode Standards and Implementation Guidelines, GS1 Organization, 2023.
[3] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed., McGraw-Hill, 2019.
[4] H. Schildt, Java: The Complete Reference, 9th ed., McGraw-Hill Education, 2014.
[5] P. Deitel and H. Deitel, Java: How to Program, 10th ed., Pearson Education, 2015.
[6] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed., Pearson, 2016.
[7] R. S. Pressman, Software Engineering: A Practitioner's Approach, 7th ed., McGraw-Hill, 2010.