

INVENTORY MANAGEMENT SYSTEM

5/12/2025

ITC 315 Final Project

Ahmad Wali Hemat, Alia Rafi, Zainab Qasemi,
Alisina Nazari, Hamid Sultani
AMERICAN UNIVERSITY OF AFGHANISTAN

Group C - Inventory Management System for Small Shops

ROLE	MEMBER	RESPONSIBILITY
PRODUCT OWNER	Hamid Sultani	<ul style="list-style-type: none"> • maintains product backlog • hosts sprint planning/review meetings • records minutes
SCRUM MASTER	Zainab Qasemi	<ul style="list-style-type: none"> • Daily sprint meetings • Manages sprint timing with the development team • Motivates the development team • Help create user stories

DEVELOPER	Alisina Nazari, Ahmad Wali Hemat Alia Rafi	<ul style="list-style-type: none"> • System Modelling • Development • Requirements and Testing Documentation
------------------	---	---

Instructor:

Prof. Nooria Rezazada

Date:

December 5, 2025

Table of Contents

1. Introduction	4
2. System Overview	4
2.1 User Profiles	6
Admin	6
Staff	6
Manager	6
3. User Needs	7
4. Scrum Methodology	7
5.1 User Requirements Specification (URS)	8
5.2 Functional Requirements	8
5.3 Non-Functional Requirements	9
7. Entity Relationship Diagram (ERD)	9
8. UML Diagrams	10
8.1 Use Case Diagram	10
8.2 Sequence Diagram (Stock-In)	11
8.3 User Interaction Workflows	12
Stock-In Workflow	12
Low-Stock Monitoring Workflow	12
9. Implementation Overview	12
The system was implemented using:	12
10. Testing	13
10.1 User Acceptance Criteria (UAC)	13
12. Conclusion	14

1. Introduction

The Inventory Management System (IMS) is designed to help small shops manage their products easily and accurately. Many small businesses still use paper or basic spreadsheets to track stock, which often leads to mistakes, missing information, and delays. This system solves those problems by digitizing inventory tasks such as adding products, updating stock levels, searching items, and generating reports. The goal is to make inventory work faster, more accurate, and more organized for shop owners and their staff.

2. System Overview

The IMS is a simple web-based application that allows users to manage products and stock activities in one place. The system includes modules for product management, stock-in and stock-out transactions, user account management, dashboards, and reporting tools.

Main Modules:

- **Product Management:** The Product Management module is responsible for handling all essential information about the products in the system. Admin users can add new products by entering details such as the product name, SKU, category, quantity, and minimum stock level. If any of this information changes over time. For example, the product name or category admins can easily edit the product. Products that are no longer sold can also be removed from the system. Additionally, this module allows users to view a complete list of all products, giving them clear visibility of the current inventory at any time.
- **Stock Management:** The Stock Management module updates product quantities based on daily inventory activities. When new items arrive, staff record a **stock-in**, which increases the quantity of that product. When items are sold, damaged, expired, or removed for any reason, staff record a **stock-out**, which

decreases the quantity. The system automatically adjusts the product quantity after each transaction to maintain accuracy. Staff members can also add notes during stock-out like, “damaged items” or “sold in bulk” to keep a clear history of why stock was reduced. This module ensures the inventory remains up to date and all stock movements are properly recorded.

- **User Management:** The User Management module controls who can access the system and what actions they can perform. Only the admin has permission to use this module. Admins can create new user accounts for staff or managers and assign a role to each user. These roles determine which features the user can access. For example, staff can record stock changes, while managers can view reports. Admins can also deactivate user accounts if someone leaves the organization or no longer needs access. This module keeps the system secure by ensuring that only authorized users can manage inventory.
- **Dashboard:** The Dashboard module provides a quick summary of the shop’s inventory status. It displays the most important information in a simple and clear format, allowing managers and admins to understand the condition of their inventory at a glance. The dashboard typically shows the total number of products, a list of low-stock items, and sometimes the total stock-in and stock-out activity for the day. This helps users immediately identify products that need attention without searching through long lists. Overall, the dashboard acts as a snapshot of the overall health of the inventory.
- **Reporting:** The Reporting module gives users detailed information about inventory performance and movement. Managers and admins can view a **low-stock report**, which lists all items that are close to running out. They can also access a **stock movement report** that shows all stock-in and stock-out actions over time. These reports help managers identify patterns, such as which products sell quickly or which ones are not moving. By analyzing these trends, users can make better decisions about restocking and product management. The reporting module is essential for long-term planning and inventory control.
- **Search:** The Search module allows users to quickly find products without having to scroll through long lists. Users can type in the product name or SKU, and the

system instantly displays all matching results. This feature is especially helpful for shops with large numbers of products, as it saves time and reduces the chances of selecting the wrong item by mistake. The search function ensures that staff can work faster and more accurately during stock updates.

2.1 User Profiles

The system supports three types of users:

Admin: The Admin has the highest permissions. They can add and update products, delete products, create user accounts, assign roles, and access all reports. Admin ensures the system is correctly used and updated.

Staff: Staff members perform daily inventory activities. They record stock-in and stock-out transactions and view product information. They help maintain accurate inventory levels.

Manager: The Manager reviews inventory performance. They check the dashboard, look at low-stock alerts, and review stock movement to decide when items should be reordered.

3. User Needs

Small shop owners and their employees need a system that is simple, fast, and reliable. They may not have technical experience, so the IMS must be straightforward.

Key User Needs:

- **Ease of Use:** Users want simple forms and clear buttons.
- **Accuracy:** Stock levels must always be correct.
- **Speed:** Search and stock operations must be quick.
- **Security:** Only authorized users should access the system.
- **Visibility:** Managers want a clear view of low-stock items and trends.

These needs guide the design of system features and workflows.

4. Scrum Methodology

The project used Scrum as the development methodology. Work was divided into sprints, each producing a working part of the system. Daily stand-up meetings helped track progress, and sprint reviews showed completed features.

Sprint Summary:

- **Sprint 1:** Core product and stock features.
- **Sprint 2:** Search, delete product, low-stock alerts.
- **Sprint 3:** Dashboard and reports.
- **Sprint 4:** Testing, documentation, final improvements.

This approach ensured continuous progress and flexibility in team work.

5. Requirements

5.1 User Requirements Specification (URS)

These requirements describe what the user expects from the system:

1. The user must be able to log in securely.
2. Users should see a clear list of all products.
3. Staff should easily record stock-in and stock-out.
4. Admin should manage product information.
5. Managers should view reports and dashboards.
6. Users should be notified of errors or successful actions.
7. Low-stock items must be highlighted.
8. Users should search by product name or SKU.

5.2 Functional Requirements

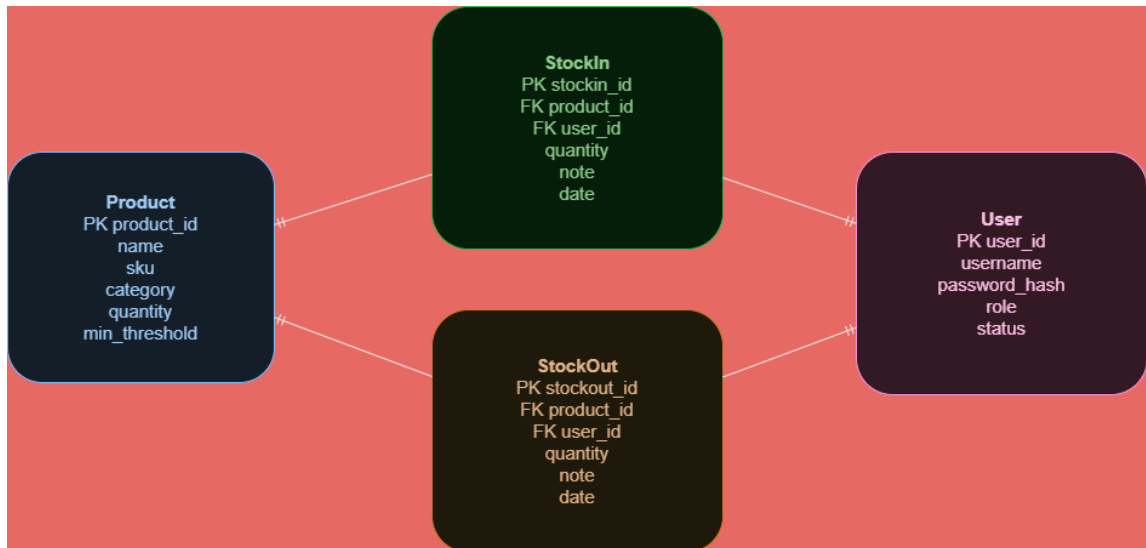
- Add, edit, delete products.
- Record stock-in.
- Record stock-out with optional notes.
- Auto-update stock quantities.
- Search products by name or SKU.
- Login and logout.
- View dashboard with low-stock items.
- Generate stock movement and low-stock reports.

5.3 Non-Functional Requirements

- **Usability:** Simple interface.
- **Performance:** Fast loading.

- **Security:** Password protection and role-based access.
- **Maintainability:** Clean and modular code.

7. Entity Relationship Diagram (ERD)



Main Entities:

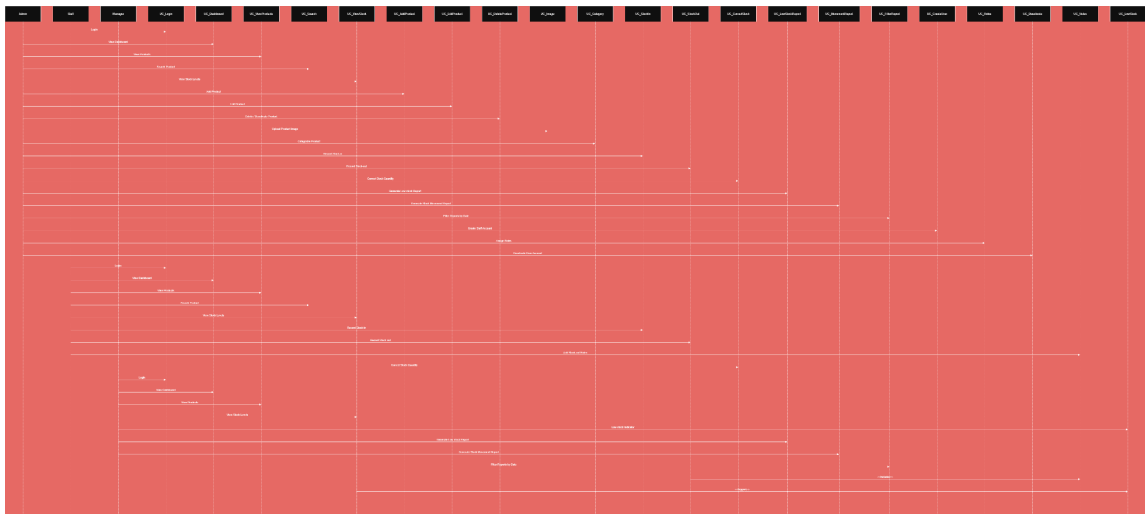
- **Product:** PKproduct_id, name, SKU, category, quantity, min_threshold.
- **StockIn:** PK stockin_id, FK product_id, FK user_id, quantity, note, date.
- **StockOut:** PK stockout_id, FK product_id, FK user, id, quantity, note, date.
- **User:** PK user_id, username, password_hash, role, status.

8. UML Diagrams

Using UML diagrams we will explain how LMS works both from a user perspective and a technical workflow perspective.

8.1 Use Case Diagram

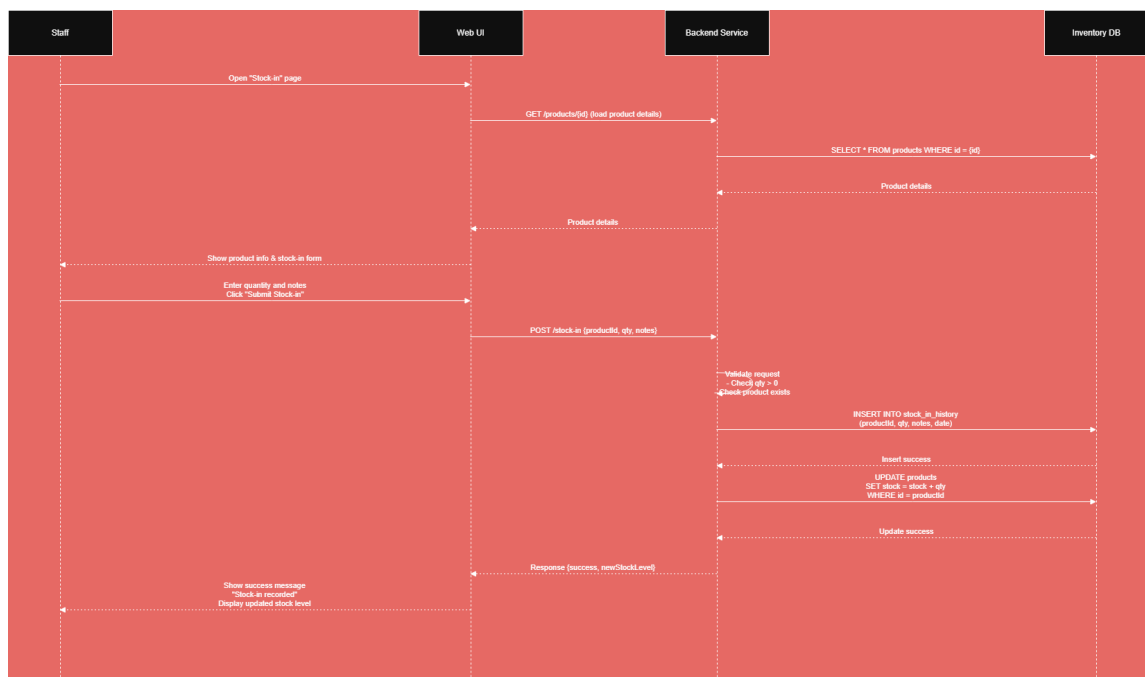
The Use Case Diagram shows three main users of the system Admin, Staff, and Manager and the actions each user can perform. It provides a high-level view of all the features available in the system and who is responsible for each action. For example, the **Admin** has the most permissions. They can add new products, edit product information, delete products, create user accounts, assign roles, and manage the system settings. The **Staff** user focuses on daily inventory operations, such as viewing products, searching for items, and recording stock-in and stock-out transactions. The **Manager** mainly reviews the system's information. They can view the dashboard, check low-stock items, and generate reports to understand inventory movement.



8.2 Sequence Diagram (Stock-In)

The sequence diagram explains how step by step the stock_in process works which is a detailed interaction between the Staff, the Web UI (front-end), the Backend Service, and the Inventory Database. The process begins when the **Staff** opens the Stock-In page. The Web UI requests the product details from the backend, and the backend retrieves the

information from the database. Once the product information is displayed, the Staff enters the quantity and any notes, then submits the Stock-In form. The Web UI sends this data to the backend, which validates the request. If everything is valid, the backend inserts a new record into the **stock_in_history** table and then updates the product's total stock quantity in the database. After the update succeeds, the backend sends a success message back to the Web UI.



8.3 User Interaction Workflows

Stock-In Workflow

1. Staff logs in
2. Staff opens the stock-in page
3. Selects a product
4. Enter quantity and note
5. Submit the form

6. The system updates stock and shows success

Low-Stock Monitoring Workflow

1. The manager opens the dashboard
2. The system compares current stock levels with minimum thresholds
3. Low-stock products are highlighted
4. The manager decides which items to reorder

9. Implementation Overview

The system was implemented using:

- **React** for the UI(frontend), which provides a fast and responsive user interface for tasks such as viewing products, recording stock changes, and navigating through different pages.
- **Node.js + Express** for API routes, where all business logic is handled. This includes validating user actions, processing stock updates, and communicating with the database.
- **MongoDB** for database, which stores information about products, users, and stock-in and stock-out transactions in a flexible document format. In the backend we use REST API endpoints that allow the frontend to interact with the system by sending and receiving data for operations such as managing products, updating stock, and handling user authentication. Together, these technologies create a reliable, efficient, and easy-to-maintain system that supports all core inventory management functions.

10. Testing

Testing for the Inventory Management System was carried out manually throughout the development process to ensure that all core features functioned correctly. Each module was tested using real sample data to confirm that the system behaves as expected in typical shop operations. In testing modules we went through the the following steps:

- Adding and editing products
- Recording stock-in and stock-out
- Checking low-stock alerts
- Searching products
- Logging in and out
- Viewing reports and dashboards

10.1 User Acceptance Criteria (UAC)

From a user perspective the Inventory Management System is considered acceptable and ready for use when the following conditions are met:

- Admin can manage products without errors.
- Stock-in and stock-out correctly update quantity.
- Low-stock alerts work automatically.
- Unauthorized users cannot access admin actions.
- Dashboard loads correctly with accurate numbers.

12. Conclusion

The Inventory Management System provides a practical and efficient solution for small shops that struggle with manual inventory tracking. By digitizing daily tasks such as adding products, recording stock-in and stock-out, and monitoring low-stock levels, the system significantly reduces human errors and saves valuable time for staff. Managers gain better visibility and control over inventory through clear dashboards and accurate reports, making it easier to make informed decisions about restocking and product performance.

Throughout the project, the Scrum methodology helped our team work in organized stages. Each sprint delivered a functional part of the system, allowing us to test early, gather feedback, and improve continuously. By the end of development, all core features product management, stock tracking, user roles, search, dashboard, and reporting were fully implemented and tested.

The system successfully meets the functional and non-functional requirements we defined at the beginning of the project. It is simple to use, secure, and flexible enough to be expanded in the future. Overall, the Inventory Management System provides a strong foundation for improving inventory operations in small retail environments.