Requirements SmartInventory

Riad BENBRAHIM & Youssef KAYA

October 2025

Contents

SmartInventory System is an application for easy management of stock for businesses or warehouses. Tracks items, monitor quantities, and sends alerts when stock is low. SmartInventory helps you optimise and grow your business by focussing on the winning products minimising costs, and channeling efforts.

1 Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the **SmartInventory System**, an intelligent stock management application intended for small and medium retail businesses, warehouses, and distributors. This document specifies the functional and non-functional requirements that the system must satisfy during design, implementation, and testing phases.

1.2 Scope

The **SmartInventory System** is a web application designed to help businesses efficiently manage their inventory. The system will automate repetitive stock management tasks such as item tracking, supplier management, sales and purchase recording, alert generation, and performance reporting. It will also provide analytical insights and recommendations to improve profitability and efficiency. The system will:

- Allow authorized users to create, update, and delete inventory items and supplier information.
- Track all stock movements including purchases, sales, damages, and returns...
- Generate automated alerts and customizable analytical reports.
- Support multiple users with role-based access (Manager, Worker).
- Provide a responsive, intuitive web interface.

The system will not:

- Handle online payment processing or external financial transactions directly.
- Provide advanced machine learning–based demand forecasting in the initial version.

User Classes and Benefits:

- **Manager:** Manages system configuration, users, and access permissions. Monitors stock levels, reviews reports, and makes strategic decisions.
- **Worker:** Records daily sales ,returns, and updates .

1.3 Product Functions

The major functional capabilities include:

- Managing inventory items, suppliers, and clients.
- Recording purchases, sales, and returns.
- Generating low-stock alerts and periodic reports.

- Producing analytics based on ROI, turnover rate, and best-selling products...
- Supporting user authentication and role-based permissions.
- Enabling multi-store management and automation in scalable deployments.

2 Feasibility Study

2.1 Technical Feasibility

Category	**Tools / Technologies**	**Our Level**	**Area Of Improvement**
Programming Language	Python	Basic to intermediate	Need practice with OOP and structuring a large project
Database	SQLite (via sqlite3 and SQLAlchemy)	Some SQL knowledge	Learn DB schema design and migrations
Backend / API	Flask (REST API)	Beginner knowledge	Learn routing, request handling, JSON responses
Frontend	HTML, CSS, JavaScript	Basic HTML/CSS	Learn JavaScript for frontend features and data exchange.
Reports / Analysis	Pandas, Matplotlib	Limited experience	Need practice with data visualization
Version Control	Git + GitHub	Basic git usage	Define clear workflow (branches, pull requests)
Testing	Pytest	No experience	Must learn from scratch
hardware	Any laptop with the technologies in **requirements.txt** downloaded	Basic terminal commands	No major issue

Table 1: Technical feasibility analysis.

Conclusion: The project is technically feasible. The team already has Python and SQL basics. The main challenges are learning Flask for APIs, structuring the project with OOP, acquiring remaining tools for frontend, handling data exchange properly and ensuring efficient collaboration with GitHub.

2.2 Market and User Feasibility

Target Users:

- Small and medium businesses (shops, warehouses, resellers).
- Independent entrepreneurs currently still relying on Excel and notebooks.

Competitive Analysis:

Existing Tool	Features	Weakness/Gaps
Microsoft Excel / Google Sheets	Flexible, simple to start	Not automated, no alerts, no API integration
Odoo Inventory	Professional Enterprise Resource Planning with stock management	Too complex and expensive for small shops
Zoho Inventory	Cloud-based stock and order management	Subscription cost, may be overkill for small businesses
SmartInventory (Our Project)	Intuitive stock management tools, alerts, reports, performance analysis, and recommendations	For small/medium business, local storage use.

Table 2: Competitive analysis.

Survey Results:

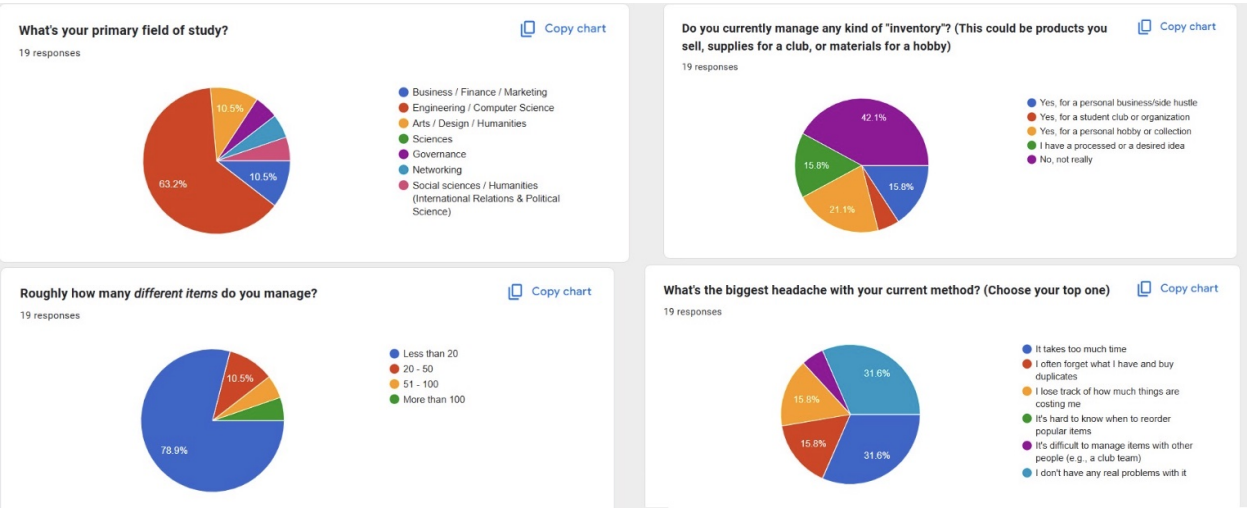

Figure 1: Survey Results

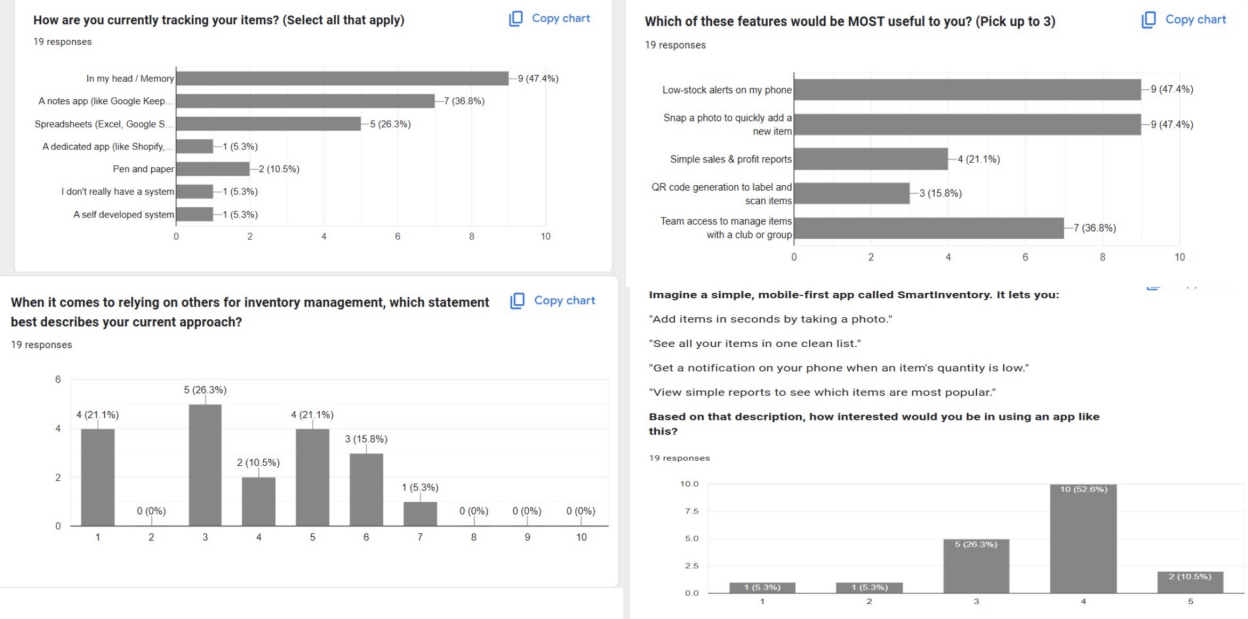

Figure 2: Survey Results

Figure 3: Suggestions

Conclusion: There is a clear need for a lightweight, intuitive, and smart stock manager. It fills the gap between simple spreadsheets and heavy Enterprise Resource Planning solutions. It also reduces risks involved in managing inventory and trusting the team, SmartInventory is a safer and more reliable system to ensure transparency and protect both parties interests.

2.3 Schedule Feasibility

The schedule consider 10 week duration for the project, we included main functionalities, and left the scaling and secondary functionalities aside for the moment.

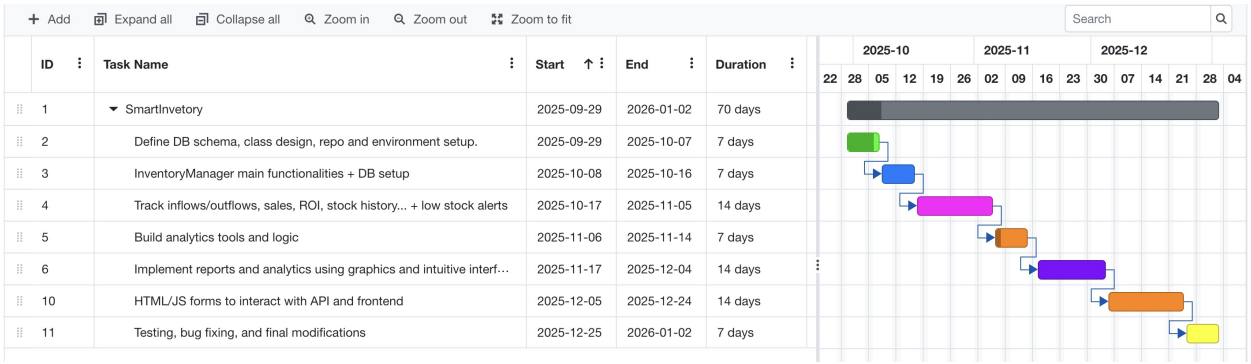

Figure 4: Project Schedule

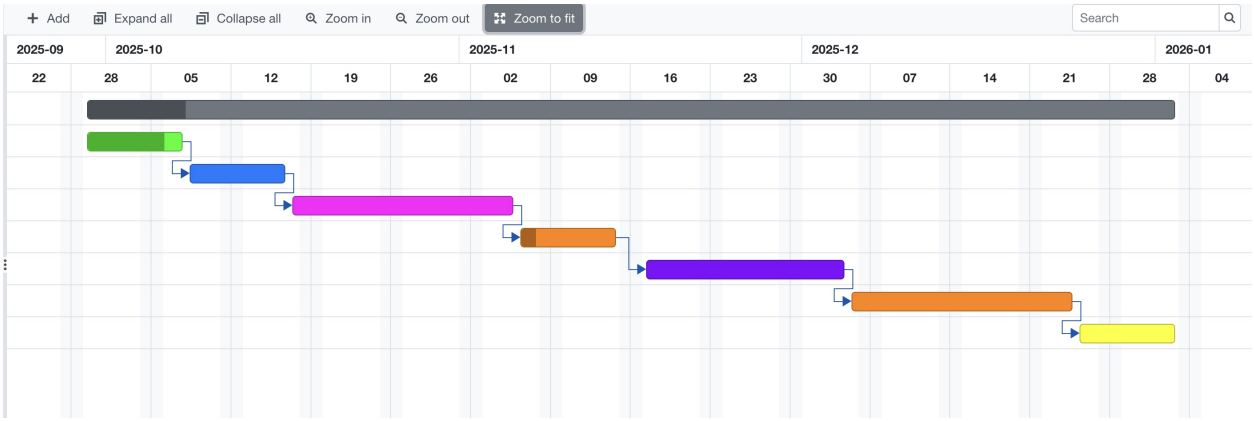

Figure 5: Project Schedule

Conclusion: The project is feasible within the semester if tasks are divided among the two team members. Flask + frontend add complexity, but remain achievable with proper planning.

3 Functional Requirements

3.1 Item Management

ID	Requirement	State	Priority	Effort
FR1	The system shall allow an authorized user to add a new item to the inventory with fields such as SKU, name, supplier, quantity, threshold, cost price, selling price, quantity, and category.	Draft	Must Have	Medium
FR2	The system shall allow an authorized user to modify existing item details.	Draft	Must Have	Medium
FR3	The system shall allow an authorized user to delete an item or archive it if stock history exists.	Draft	Should Have	Medium

FR4	The system shall display item details and stock history (current quantity, total sold, location, etc.).	Draft	Must Have	Medium
FR5	The system shall provide search and filter options by name, category, supplier, margin, discount, stock level, etc.	Draft	Should Have	Medium

3.2 Stock Control & Transactions

ID	Requirement	State	Priority	Effort
FR1	The system shall allow users to record stock additions (purchases), including item, quantity, cost price, date, and supplier.	Draft	Must Have	Medium
FR2	The system shall allow users to record sales, including items, quantities, selling price, discounts, payment method, and status.	Draft	Must Have	High
FR3	The system shall allow users to process customer returns, increasing the stock quantity accordingly.	Draft	Should Have	Medium
FR4	The system shall allow users to record stock removals due to damages, expiration, waste, internal use, or warranty repair.	Draft	Should Have	Medium
FR5	The system shall maintain an immutable, auditable history of all stock transactions.	Draft	Must Have	High

3.3 Alerts & Reporting

ID	Requirement	State	Priority	Effort
FR1	The system shall automatically generate low-stock alerts when quantities fall below the threshold.	Draft	Must Have	Medium
FR2	The system shall automatically generate and send bi-weekly or monthly reports summarizing key metrics.	Draft	Must Have	High
FR3	The system shall allow users to generate custom on-demand reports with filters, analytics and recommendations.	Draft	Must Have	High
FR4	The system shall include report content such as stock percentage, ROI, profits, turnover, and revenue contribution.	Draft	Must Have	High
FR5	The system shall provide analytics using graphics for clear visualization and easy understanding.	Draft	Must Have	High

3.4 Supplier Management

ID	Requirement	State	Priority	Effort
FR1	The system shall allow users to add, modify, or remove suppliers with details such as name, address, and contact information.	Draft	Should Have	Medium
FR2	The system shall link items to one or more suppliers.	Draft	Must Have	Medium
FR3	The system shall track payment status for suppliers, including "Paid" or "In debt."	Draft	Must Have	Medium

3.5 Data Analysis & Recommendations

ID	Requirement	State	Priority	Effort
FR1	The system shall analyze sales data to identify the most profitable and least profitable products.	Draft	Must Have	High
FR2	The system shall allow tracking and comparing sales performance across different resellers or selling sites.	Draft	Should Have	High
FR3	The system shall provide intelligent recommendations such as optimal reorder quantities, focus products, and best suppliers.	Draft	Must Have	High

3.6 System & User Management

ID	Requirement	State	Priority	Effort
FR1	The system shall provide a modern, user-friendly web interface.	Draft	Must Have	High
FR2	The system shall support inventory management across multiple physical store locations.	Draft	Should Have	High
FR3	The system shall require user authentication with username and password.	Draft	Must Have	Medium
FR4	The system shall adapt functionnalities on a role-based access control with different permissions for managers and workers.	Draft	Must Have	High

4 Non-Functional Requirements

4.1 Performance

ID	Requirement	State	Priority	Effort
NFR1	The system shall load main pages (dashboard, item list) in under 2 seconds under normal load.	Draft	Should Have	Medium

NFR2	The system shall support at least 10 concurrent users without significant performance degradation.	Draft	Should Have	Medium

4.2 Security

ID	Requirement	State	Priority	Effort
NFR1	The system shall encrypt sensitive data at rest and in transit using HTTPS.	Draft	Must Have	High
NFR2	The system shall enforce access control based on user roles.	Draft	Must Have	Medium
NFR3	The system shall be protected against SQL Injection and XSS.	Draft	Must Have	High

4.3 Usability

ID	Requirement	State	Priority	Effort
NFR1	The system shall provide an intuitive and easy-to-use interface suitable for non-technical users.	Draft	Must Have	Medium
NFR2	The system shall have a responsive design compatible with desktop, tablet, and mobile.	Draft	Should Have	Medium
NFR3	The system shall provide clear user documentation and a help section.	Draft	Should Have	Medium

4.4 Reliability & Availability

ID	Requirement	State	Priority	Effort
NFR1	The system shall perform daily backups to prevent data loss.	Draft	Must Have	Medium
NFR2	The system shall be implemented with maintainable, and well-documented code, to simplify scaling process later.	Draft	Must Have	Medium