# Software Requirements Specification

## for

# Online Inventory Control System

**Version 1.2 approved**

**Prepared by Anirban Bandyopadhyay**

**Techno Main Salt Lake**

**01.08.2025**

# Table of Contents

# 1 Introduction

**“Inventra**”, The Online Inventory Control System is being developed to manage and track inventory efficiently in a digital environment. This system is intended for use in retail businesses, college events, startups, and warehouses where monitoring stock levels, item entries, and reordering needs to be automated and accessible via the web. By replacing traditional manual methods, the system will provide a streamlined solution to minimize errors, reduce overstock or stockouts, and improve overall operational visibility.

## 1.1 Purpose/Objective

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for the development of **Inventra**, the Online Inventory Control System. The objective of this system is to provide a user-friendly, secure, and efficient platform that allows users to add, update, delete, and monitor inventory items in real time. It will support essential features such as low-stock alerts, role-based access, stock reports, and accessibility.

## 1.2 Document Conventions (Definition, Acronyms, Abbreviations)

### 1.2.1 **Alignment**

The entire document is in justified alignment.

### 1.2.2 Convention for the SRS

1.2.2.1 Font Face: Times New Roman

1.2.2.2 Font Style: None

1.2.2.3 Font Size: 12

### 1.2.3 Definition

1.2.3.1 Inventory - A collection of goods, raw materials, or finished products that are stored, tracked, and managed by a business or organization.

1.2.3.2 Stock Keeping Unit (SKU) - A unique identifier assigned to each distinct product or item in inventory. SKUs are used to track quantities, manage stock movement, and differentiate between product variations such as size, color, or model.

1.2.3.3 Inventory Management System - A software solution that automates the process of tracking inventory levels, managing reorders, monitoring item availability, and generating reports related to stock movement and sales..

### 1.2.4 Acronyms

1.2.4.1 OICS: Online Inventory Control System (Inventra)

1.2.4.2 SRS: Software Requirements Specification

1.2.4.3 SKU: Stock Keeping Unit

1.2.4.4 DBMS: Database Management System

1.2.4.5 API: Application Programming Interface

## 1.3 Scope

This document applies to the development of Inventra, Online Inventory Control System (OICS) – Version 1.0. The software is designed to help users efficiently manage and monitor inventory items through a secure, web-based interface. It aims to replace manual stock tracking methods with automated tools that allow businesses, student startups, and project teams to maintain real-time visibility of their inventory..

### 1.4 References

1.4.1 "Inventory Management System Documentation and Use Cases" – GeeksforGeeks - https://www.geeksforgeeks.org/inventory-management-system

1.4.2 "Inventory Management Best Practices" – Zoho Inventory - https://www.zoho.com/inventory/guides/inventory-management.html

1.4.3 "Wikipedia Article on Inventory Management - https://en.wikipedia.org/wiki/Inventory_management.

1.4.4 "Inventory Control Explained" – Investopedia - https://www.investopedia.com/terms/i/inventory-control.asp**.**

# 2 History/Background Study (Sources of Domain Knowledge)

## 2.1 Technical Literature

The term Inventory Control System refers to a software-based solution designed to monitor, manage, and optimize stock levels within an organization. To develop an effective Online Inventory Control System, extensive research was conducted on inventory management principles, digital stock tracking tools, and automation technologies used in small to medium-sized businesses.

The Technical Literature section provides a consolidated review of academic publications, whitepapers, industry case studies, and software engineering books related to inventory systems. It includes insights on real-time inventory tracking, barcode-based item identification, role-based access control, and cloud-based storage techniques.

## 2.2 Existing Applications

2.2.1 **Zoho Inventory -** Zoho Inventory is a popular cloud-based inventory management solution aimed at small to medium-sized businesses. It offers multi-channel inventory tracking, automated reordering, and detailed analytics. Zoho stands out for its seamless integration with platforms like Amazon, eBay, Shopify, and various accounting tools. However, advanced features are locked behind paid tiers, making it less accessible for very small businesses or student-led startups.

2.2.2 **Odoo Inventory -** Odoo is an open-source ERP suite with a powerful inventory module. It provides warehouse management, automation via rules, and real-time stock operations. While feature-rich, it has a steep learning curve and requires technical setup, making it more suitable for large organizations..

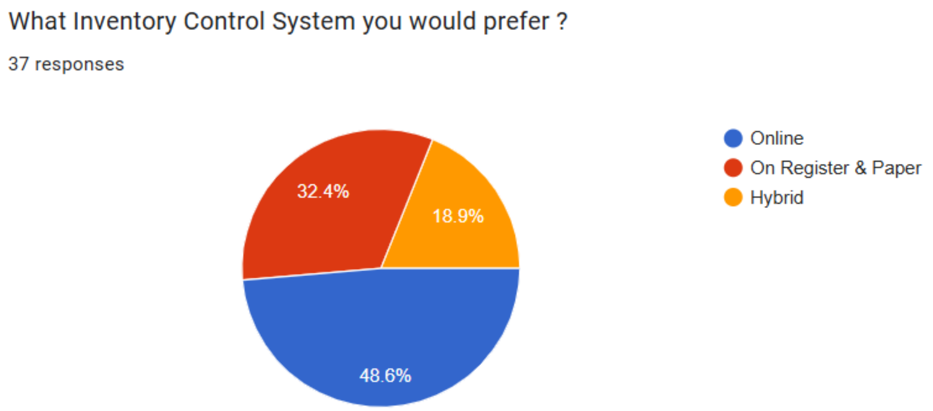
2.2.3 **Inflow Inventory -** Inflow is a desktop-first inventory tool designed for easy product and order tracking. It supports barcode scanning, stock adjustments, and order fulfillment. While its simplicity is appreciated by new users, it lacks a strong mobile interface and real-time collaboration capabilities.

2.2.4 **Google Sheets/Excel -** Many startups and college teams still use spreadsheets for inventory tracking. While flexible and easy to use initially, spreadsheets become difficult to manage as the stock grows, leading to errors, version control issues, and lack of real-time collaboration or alerts.

## 2.3 Customer Surveys

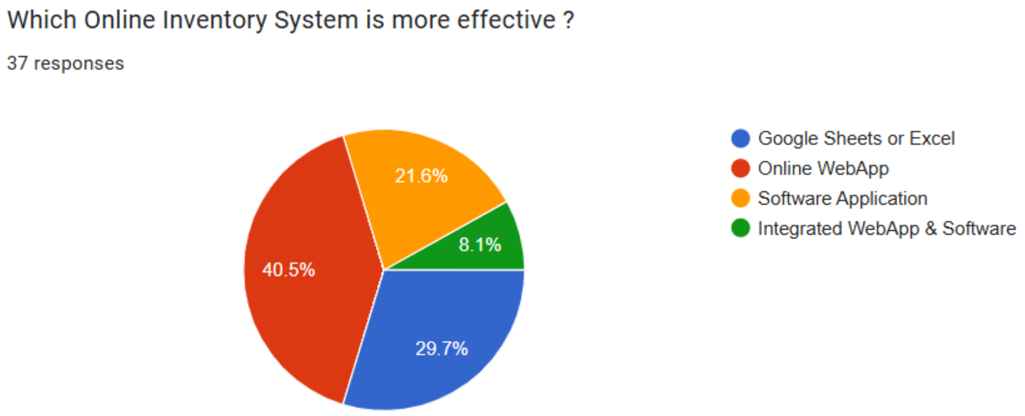
Survey Form ( Click to view )
Response Sheet ( Click to view )

**Response Statistics:**



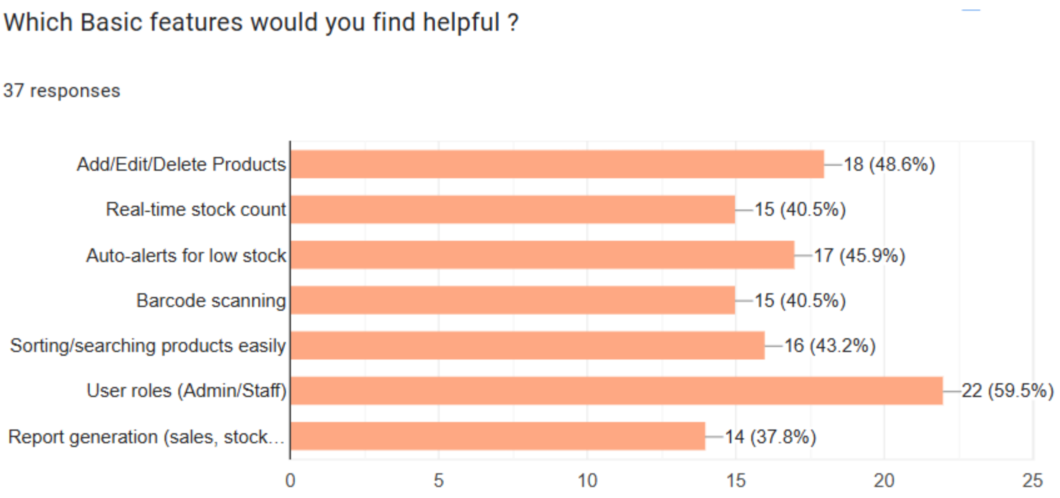
What Inventory Control System you would prefer ?

37 responses



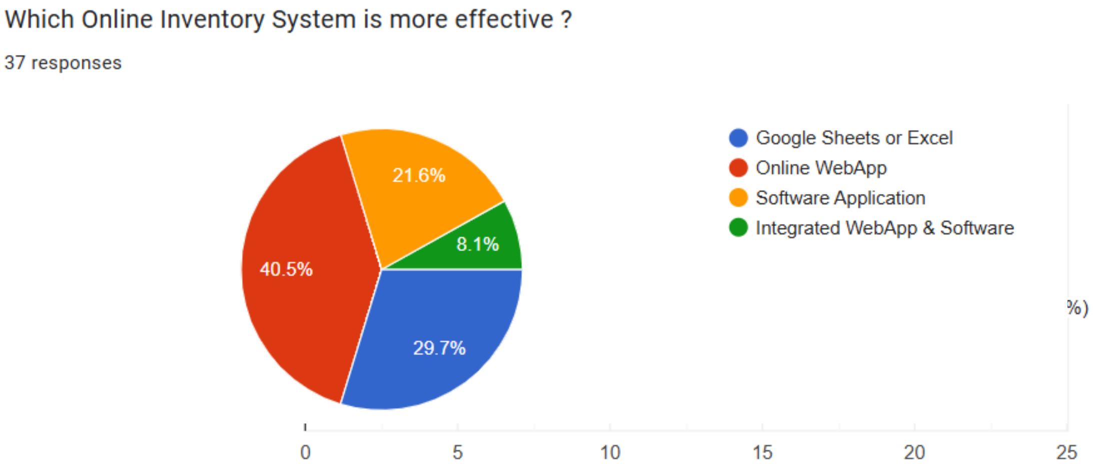
Which Online Inventory System is more effective ?

37 responses



Which Basic features would you find helpful ?

37 responses

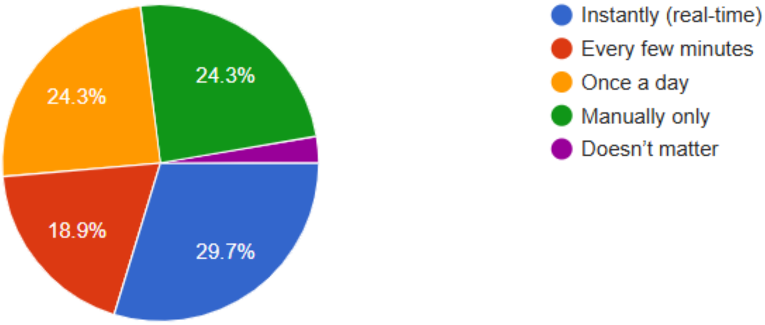


Which Online Inventory System is more effective ?

37 responses

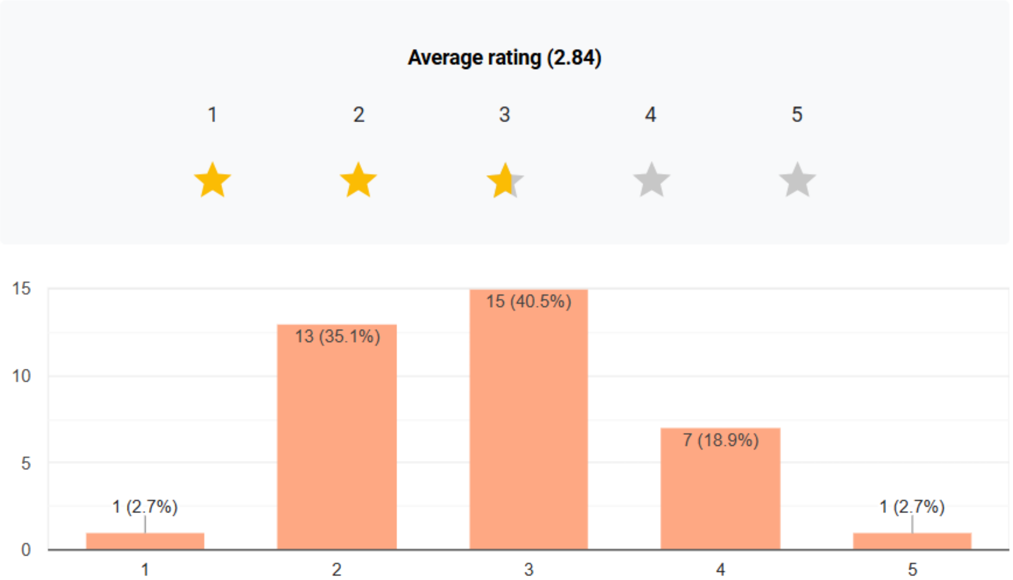
How often would you ideally like the system to update stock status?

37 responses



How Satisfied are you with existing Inventory Control Systems ?

37 responses



## 2.4 Expert Advice

2.4.1 Inventory management is a core function in retail, manufacturing, and e-commerce operations, and industry experts unanimously emphasize the importance of automation and digital visibility. According to supply chain analysts and ERP consultants, systems like Inventra can greatly reduce operational inefficiencies and human errors by enabling real-time tracking, automated alerts, and structured inventory workflows.

2.4.2 Experts from platforms like Zoho Inventory, Oracle NetSuite, and SAP highlight that businesses—especially small to medium-sized enterprises—struggle with stockouts, overstocking, and manual entry errors. A centralized, web-based inventory solution like OICS (Inventra) ensures synchronized inventory views across locations and devices, which is critical in today's fast-moving sales environments.

2.4.3 Ease of use, mobile accessibility, and actionable insights (through analytics and reports) are considered essential by industry leaders. Inventory systems must be intuitive enough for non-technical users while providing robust control for admins. Advisors also stress the need for scalability and integration with POS, CRM, and e-commerce systems, to enable a seamless operational flow from purchase to delivery.

## 2.5 Current/Future requirements

2.5.1 All functionalities in Inventra—including product management, stock tracking, user roles, and low-stock alerts—must operate reliably and without disruption across devices and platforms.

2.5.2 The system should be designed with scalability and modularity in mind, allowing future features such as barcode/QR integration, sales analytics, multi-location inventory.

2.5.3 Future versions of Inventra may include optional integration with payment modules (e.g., UPI, Razorpay, Stripe) for order-based inventory or billing purposes.

# 3 Overall Description

## 3.1 Product Functions

### 3.1.1 Hardware Requirement

#### 3.1.1.1 Server-Side Requirements

- Processor: Multi-core processor (Intel i5/i7 or AMD equivalent)
- RAM: Minimum 8 GB (16 GB recommended)
- Storage: SSD with at least 500 GB capacity
- Network: High-speed internet connection with stable uptime
- Backup Device: External hard drive or cloud-based backup service

#### 3.1.1.2 Client-Side Requirements

- Device: Desktop, Laptop, Tablet, or Smartphone

- Display: Minimum 13" screen with 1366x768 resolution
- Input: Keyboard/Touchscreen for interaction
- Browser: Modern browser (Chrome, Firefox, Edge, Safari)
- Barcode Scanner (Optional): USB/handheld scanner for product input

### 3.1.2 Software Requirement

#### 3.1.2.1 Operating System

- Server: Linux (Ubuntu/CentOS) or Windows Server
- Client: Compatible with Windows, macOS, Android, iOS

#### 3.1.2.2 User Interface

- Built using modern web frameworks (NextJS)
- Responsive design to support multiple screen sizes
- Dashboard with filters, search, and statistics

#### 3.1.2.3 Security

- Role-based authentication system (Admin/Staff/User)
- Password encryption and secure login
- Token/session-based access control

#### 3.1.2.4 Database

- Automatic database backups
- No-SQL database ( MongoDB )

## 3.2 Functional Requirements

### 3.2.1 New User

**Description:**

New users (such as inventory managers or staff) can register an account on **Inventra** by providing necessary credentials. Registration ensures secure access and enables role-based functionality within the system..

**Input**

- Full Name
- Email Address
- Phone Number
- Password (with confirmation)
- Role selection (Admin/Staff)

**Output**

- Confirmation of account creation

**Error**

- Invalid or duplicate email/phone number
- Weak password

### 3.2.2 Existing User Login

**Description:**

Registered users can securely log in to **Inventra** using their credentials. The system authenticates the user and grants access based on assigned roles and permissions.

**Input**

- Email/Username
- Password

**Output**

- Successful login and redirection to the user dashboard

**Error**

- Invalid login credentials
- Account locked after multiple failed attempts

### 3.2.3 Product Management

**Description:**

Admins or authorized users can add, update, or delete products in the inventory. Each product includes fields such as name, SKU, quantity, category, price, and description.

**Inputs:**

- Product Name
- SKU (Stock Keeping Unit)
- Category
- Quantity
- Price
- Description
- Image (optional)

**Output:**

- Confirmation message after product operation (add/update/delete)

**Errors:**

- Duplicate SKU
- Invalid input fields

### 3.2.4 Stock Adjustment

**Description:**

Users can manually update inventory quantity when stock is received, sold, returned, or damaged. The system ensures logs are maintained for audit and traceability.

**Inputs:**

- Product selection
- Quantity to increase/decrease
- Reason (e.g., Purchase, Sale, Damage)

**Output:**

- Updated inventory and transaction log

**Errors:**

- Negative quantity not allowed unless backorder is enabled

### 3.2.5 Low Stock Alert System

**Description:**

Inventra will automatically trigger low-stock alerts for products falling below their threshold value. Admins or staff will receive notifications via dashboard or email.

**Inputs:**

- Predefined stock threshold value per product

**Output:**

- Alert/notification on low stock

**Errors:**

- None (system-generated)

### 3.2.6 User Role Management

**Description:**

Admins can create, edit, and manage user roles. Roles determine access levels (e.g., Admin has full access; Staff can only view and update stock).

**Inputs:**

- Select user
- Assign or modify role

**Output:**

- Role updated successfully

**Errors:**

- Unauthorized access attempt by non-admins

### 3.2.7 Inventory Search & Filter

**Description:**

Users can search for products or filter them by name, category, SKU, stock level, or date added.

**Inputs:**

- Search query or selected filter parameters

**Output:**

- Filtered and sorted product list

**Errors:**

- No matching results

### 3.2.8 Report Generation

**Description:**

Authorized users can generate downloadable reports for inventory status, stock changes, low-stock items, and transaction logs.

**Inputs:**

- Report type (Stock, Transaction, Alerts)
- Date range
- Export format (PDF, Excel, CSV)

**Output:**

- Generated report ready for download

**Errors:**

- None (validation for empty data range handled)

### 3.2.9 System Notifications

**Description:**

The system generates notifications for low stock, product expiry, successful stock transactions, or failed operations.

**Inputs:**

- Trigger event (e.g., stock < threshold)

**Output:**

- Notification pop-up, email, or dashboard alert

**Errors:**

- Notification delivery failure (e.g., email config issues)

#### 3.2.10 Session Security & Timeout

**Description:**

For security, the system automatically logs out users after a set period of inactivity. Sensitive operations may require re-authentication.

**Inputs:**

- User inactivity timeout duration

**Output:**

- Session timeout warning and logout

**Errors:**

- Loss of unsaved data if session expires without action

#### 3.2.11 AI Chatbot

**Description:**

It provided all information and customer support. It provides an end to end guide about the platform and helps smoothen tasks.

**Inputs:**

- User asks necessary questions

**Output:**

- Customized answers accordingly

**Errors:**

- None

## 3.3 Non-Functional Requirements

### 3.3.1 Correctness Requirement

- To ensure the delivery of a reliable and high-quality inventory management system, Inventra must meet the following correctness criteria:

- The system must validate all user inputs and prevent the entry of incorrect or invalid data (e.g., negative quantity, duplicate SKU).
- All core functionalities (e.g., product creation, stock updates, report generation) must be thoroughly tested across roles and usage scenarios.
- Inventory quantities must be accurately updated in real time based on stock movements such as purchases, sales, and adjustments.
- Only authenticated users with appropriate roles should be allowed to perform restricted operations like deleting records or modifying inventory.
- All calculations (e.g., total stock value, quantity alerts) must reflect accurate and up-to-date values from the database.
- The system should handle edge cases (e.g., server failure, invalid operations) gracefully using robust error handling and user feedback mechanisms.
- Data integrity and confidentiality must be ensured using encryption protocols and secure authentication.

### 3.3.2 **Portability requirement**

- Inventra should be compatible across multiple platforms including Windows, Linux, macOS, Android, and iOS through browser-based or app-based access.
- The system must run on various screen sizes including desktop, tablet, and mobile with responsive UI.
- The software should not rely on hardware-specific configurations and must be easily deployable on different server environments (cloud/on-premise).

### 3.3.3 **Efficiency Requirement**

- The system should handle database operations and stock updates efficiently, ensuring minimal latency (within 2 seconds).
- Dashboard and report load times must remain under acceptable thresholds, even with large volumes of data.
- Inventory calculations and filtering should be optimized for performance under concurrent user activity.

### 3.3.4 **Usability Requirement**

- Inventra's interface must be intuitive and easy to navigate for both technical and non-technical users.
- UI components like buttons, tables, and alerts should be clearly labeled, color-coded, and responsive.
- User onboarding, tooltips, and help sections must be available for first-time users.

- Special care should be taken to prevent navigation issues or confusing workflows during critical operations like stock edits or report generation.

### 3.3.5 **Reusability Requirement**

- Inventra must ensure data consistency across modules, with real-time syncing and transaction integrity.
- System performance must remain stable during high-traffic usage, such as bulk uploads or multi-user operations.
- A fallback mechanism must exist to handle data recovery in case of failure, using regular backups and redundant storage.

### 3.3.6 **Reliability Requirement**

- Inventra must ensure data consistency across modules, with real-time syncing and transaction integrity.
- System performance must remain stable during high-traffic usage, such as bulk uploads or multi-user operations.
- A fallback mechanism must exist to handle data recovery in case of failure, using regular backups and redundant storage.

### 3.3.7 **Maintainability Requirement**

- The system must be built using well-structured and documented code to enable quick debugging, upgrades, and scalability.
- Admins must have tools to monitor logs, update settings, and configure alerts without needing deep technical expertise.
- Updates to the application (bug fixes, new features) should be deployable without affecting existing functionality or data integrity

### 3.3.8 **User Characteristics**

- New users with little or no experience in inventory systems (e.g., student project leads, first-time entrepreneurs).
- Experienced users who frequently manage stock across departments or platforms.
- Admins or technical users responsible for system maintenance, settings configuration, and report oversight.

## 3.4 Design & Implementation Constraints

3.4.1 The system must have an intuitive and user-friendly interface to ensure accessibility for both technical and non-technical users.

3.4.2 Inventra should follow a modular and scalable architecture to support future enhancements without disrupting existing functionalities.

3.4.3 The system should use open-source or freely available technologies where possible to reduce development and deployment costs.

3.4.4 The inventory database must be normalized and optimized to handle a large volume of products efficiently.

3.4.5 Browser-based accessibility must be ensured, with compatibility across commonly used browsers (Chrome, Firefox, Edge, Safari).

## 3.5 Assumptions & Dependencies

3.5.1 The server hosting the Inventra system will have reliable uptime, sufficient processing power, and storage capacity.

3.5.2 Users will access the system through supported devices with a stable internet connection.

3.5.3 The product and stock data entered by users is assumed to be accurate and complete.

3.5.4 The software will run in an environment where routine backups and updates are regularly maintained by the system administrator.

3.5.5 The system will not be responsible for third-party hardware failures (e.g., barcode scanners, printers) and assumes they function as expected.

## 3.6 Effort Estimation – COCOMO Model

- **Estimation Model**
  - The project follows the Intermediate COCOMO (Constructive Cost Model) to estimate the effort and development time required for the successful delivery of Inventra (OICS).
- **Assumptions**
  - **Estimated Size:** 8 KLOC (thousands of lines of code)
  - **Project Type:** Semi-Detached (medium complexity and experience)
  - **Effort Adjustment Factor (EAF):** 1.12 (based on nominal cost drivers)
- **Calculated Results**
  - **Using the Intermediate COCOMO formula:**

    Effort (E) = 3.0 × (KLOC)^1.12 × EAF = 3.0 × (8)^1.12 × 1.12 ≈ 37.92 person-months

Development Time (D) = 2.5 × (Effort)^0.35 ≈ 7.93 months

- **Summary**

| **Metric** | **Value** |
|---|---|
| Estimated Effort | 37.92 person-months |
| Estimated Time | 7.93 months |
| Assumed Team Size | ~5 developers |

# 4 Interface Requirements

## 4.1 User Interfaces

The user interface (UI) of Inventra is designed to be clean, responsive, and user-friendly, enabling smooth navigation for both technical and non-technical users. The interface will guide users through various inventory functionalities using a clear, menu-driven structure with tooltips, icons, and inline guidance wherever necessary.

4.1.1 The application begins with a secure login page that prompts users to enter their registered email/username and password. Passwords are masked and login attempts are limited to enhance security.

4.1.2 Upon successful login, users are redirected to a dynamic dashboard that provides an overview of key metrics such as total products, low-stock alerts, recent transactions, and inventory value.

4.1.3 To A side navigation menu will allow users to access modules such as:

- Product Management
- Inventory Logs
- User Management (Admin only)
- Reports
- Settings

4.1.4 The UI will be fully responsive to support desktop, tablet, and mobile screens. It will also include dark/light mode toggles and support for basic accessibility features such as keyboard navigation and screen reader compatibility.

## 4.2 Hardware Interfaces

As a web-based solution, Inventra interacts minimally with hardware, but the following components may be required or optionally supported:

- A stable power supply and internet connection for hosting and accessing the application.
- Barcode scanners (USB or wireless) for scanning product SKUs, compatible with browser input fields.
- Printers (optional) for printing inventory reports or stock receipts in PDF format

### 4.3 Software Interfaces

In Inventra

interacts with various software components and services to function effectively:

- Database: The system will use MySQL, MongoDB, or PostgreSQL to manage product records, transaction logs, and user data.
- Backend Services: Built using Node.js, Express.js, or Django for managing API requests and authentication.
- Frontend: Developed using modern frameworks such as React.js or Vue.js for a fast and responsive UI.
- Authentication: Integration with JWT-based (JSON Web Tokens) or OAuth-based user authentication for security.
- Export & Report Tools: Integration with libraries for exporting data in PDF, Excel, or CSV formats

### 4.4 Communication Interfaces

Inventra relies on secure and reliable communication protocols to ensure data integrity during user interaction and backend operations:

- Communication between the frontend and backend servers will use RESTful APIs over HTTPS (SSL encryption).
- Real-time data synchronization (if required) can be enabled using WebSockets or MQTT protocol.
- Email notifications (e.g., low stock alerts) will be handled via SMTP or third-party email APIs (e.g., SendGrid or Mailgun)

## 5 Conclusion

This Software Requirements Specification (SRS) outlines the complete functional and non-functional requirements for **Inventra**, the Online Inventory Control System. The system is designed to offer a secure, user-friendly, and efficient platform for managing stock, monitoring inventory levels, generating reports, and ensuring real-time updates across devices and users.

**Inventra** aims to simplify inventory tracking for startups, small businesses, and project teams by providing key features such as product management, stock alerts, role-based access, and reporting tools. The system will be accessible through modern web browsers and optimized for both desktop and mobile interfaces.

Proper care must be taken during implementation to ensure that the software, user interfaces, database interactions, and communication protocols operate smoothly without glitches. The system should also be easy to maintain, scalable for future enhancements, and secure to protect sensitive business data.

.