



Smart  
Internz



# ***SB8096-Salesforce Developer***

## ***(Nan Mudhalvan)***

**Project Title : MEDICAL INVENTORY MANAGEMENT**

**Team ID : NM2025TMID02861**

*Team Size : 4*

*Team Leader : Subbhashini G (813022205045)*

*Team Member : Nisha S (813022205030)*

*Team Member : Nikitha R (813022205029)*

*Team Member : Gokul K (813022205016)*





# Medical Inventory Management System (MIMS)

---



## Table of Contents

1. Abstract
  2. Introduction
  3. Problem Statement
  4. Objectives
  5. Scope of the Project
  6. System Requirements
  7. Phase 1: Requirement Analysis
  8. Phase 2: System Design
  9. Phase 3: Development and Implementation
  10. Phase 4: Testing
  11. Phase 5: Deployment & Maintenance and Evaluation
  12. Results and Discussion
  13. Future Scope
  14. Conclusion
  15. References
-

## 1. Abstract

The **Medical Inventory Management System (MIMS)** is a web-based application designed to streamline the management of medical supplies in hospitals, pharmacies, and healthcare institutions. The system automates stock tracking, supplier management, expiry monitoring, and purchase orders. Built using **Django (Python)** and **MySQL**, it reduces manual errors, ensures timely replenishment, and prevents medicine wastage. This report presents a detailed, phase-wise development lifecycle of the project, from requirements to deployment and maintenance.

---

## 2. Introduction

Efficient management of medical inventory is vital to ensure patient safety and smooth hospital operations. Manual tracking methods are prone to errors, leading to shortages or overstocking of critical medicines. The **Medical Inventory Management System** provides a centralized platform to record, track, and analyze inventory data, offering automated alerts for low stock and expired items. The system enhances decision-making and operational efficiency.

---

### **3. Problem Statement**

Hospitals and pharmacies often face challenges such as:

- Lack of real-time inventory tracking
- Expiry of unused medicines
- Inefficient ordering and supplier coordination
- Manual errors in stock recording

These challenges result in wastage, financial loss, and potential risks to patients.

There is a need for a digital solution to monitor, manage, and automate the inventory workflow.

---

#### **4. Objectives**

- To design a web-based application to manage medical inventory effectively.
  - To automate the process of stock updates, purchase orders, and supplier management.
  - To minimize expired and overstocked items.
  - To generate analytical reports for better decision-making.
  - To provide secure role-based access to users (admin, pharmacist, staff).
-

## **5. Scope of the Project**

The system is designed for hospitals, clinics, and pharmacies to manage medical supplies. It can be extended to include multiple branches, integrate with barcode scanners, or connect with hospital management systems. The system ensures efficient supply chain management and reduces administrative workload.

---



# Medical Inventory Management System — Detailed Phase-Wise Plan

## PHASE 1: Project Initiation & Requirement Analysis

### Objective:

Understand the healthcare organization's inventory needs, current challenges, and define system goals and scope.

### Duration: 2–3 weeks

#### Tasks:

##### 1. Stakeholder Identification

- Identify key users: pharmacists, store managers, procurement officers, hospital admin.
- Assign project manager and development team.

##### 2. Current Process Study

- Document how inventory is currently managed (manual registers, Excel sheets, legacy systems).
- Identify bottlenecks: stockouts, overstocking, expired items, duplicate entries.

##### 3. Requirement Gathering

- Conduct interviews, surveys, and workflow observations.
- Identify system requirements:
  - Real-time stock updates
  - Auto alerts for low stock/expiry

- Supplier management
- Multi-location inventory control
- Reporting and analytics

#### **4. Requirement Documentation**

- Create **Software Requirement Specification (SRS)** document.
- Define **Functional Requirements** (what system should do).
- Define **Non-Functional Requirements** (security, performance, scalability).

#### **Roles Involved:**

- Business Analyst
- Project Manager
- Pharmacist/Inventory Officer
- Technical Lead

#### **Deliverables:**

- Project Charter
- Feasibility Study Report
- Requirement Specification Document (SRS)
- Preliminary Risk Assessment

## **PHASE 2: System Design**

### **Objective:**

Design the architecture, data model, and interface structure for the system.

### **Duration: 3–4 weeks**

#### **Tasks:**

##### **1. System Architecture Design**

- Choose architecture: 3-tier (Presentation, Logic, Data) or Cloud-based.
- Define technology stack:
  - **Frontend:** React.js / Angular / HTML-CSS-JS
  - **Backend:** Node.js / Python (Django or Flask) / Java (Spring Boot)
  - **Database:** MySQL / PostgreSQL / MongoDB
  - **Deployment:** AWS / Azure / On-premise

##### **2. Database Design**

- Identify entities:
  - Medicine, Supplier, Purchase Order, Stock, User, Transaction, Expiry Logs.
- Create **Entity Relationship Diagram (ERD)** and **Data Dictionary**.

##### **3. Interface Design**

- Create **UI wireframes** for:
  - Login & Dashboard
  - Add/Update Inventory
  - Supplier Management

- Reports Page
- Alerts/Notifications

#### **4. System Design Document (SDD)**

- Define module interactions and API endpoints.
- Include DFDs (Level 0, 1, 2) and UML diagrams (Class, Sequence, Use-case).

#### **Involved:**

- System Architect
- Database Designer
- UI/UX Designer
- Security Analyst

#### **Deliverables:**

- System Architecture Document
- ERD and DFD diagrams
- UI/UX prototypes
- API specifications
- SDD (System Design Document)

- Reduces human error,
- Improves operational efficiency,
- Provides instant access to stock information,
- Enhances decision-making for procurement and budgeting, and
- Ensures timely availability of critical medical supplies.

The developed system also promotes **data transparency, regulatory compliance, and cost optimization**. Furthermore, the modular architecture allows for **easy scalability** and **future integration** with technologies such as **AI-driven demand forecasting, IoT-based stock monitoring, and mobile health applications**.

In conclusion, this project not only fulfils its immediate purpose of streamlining medical inventory management but also lays a strong foundation for **digital transformation in healthcare logistics**. It demonstrates the potential of technology to significantly improve hospital management systems and ultimately enhance patient care delivery.

## **PHASE 3: System Development**

### **Objective:**

Build the system according to approved designs and specifications.

### **Duration: 6–8 weeks**

#### **Tasks:**

##### **1. Development Setup**

- Configure development environments and version control (Git, GitHub).
- Set up backend server, frontend framework, and database connections.

##### **2. Module Implementation**

- **Module 1:** User Authentication & Roles (Admin, Pharmacist, Auditor)
- **Module 2:** Inventory Management (Add/Edit/Delete items, batch & expiry tracking)
- **Module 3:** Supplier & Purchase Management
- **Module 4:** Sales & Usage Tracking (Issue to wards, consumption logs)
- **Module 5:** Reports & Analytics (Stock summary, expiry, cost reports)
- **Module 6:** Notification System (Email/SMS alerts for low stock, expiries)

##### **3. Integrations**

- Barcode/RFID integration for item tracking (optional).
- Payment or procurement system integration (if needed).

##### **4. Testing During Development**

- Conduct **unit tests** for each module.
- Maintain code review and bug tracking using Jira or Trello.

#### **Roles Involved:**

- Backend Developers
- Frontend Developers
- Database Engineer
- QA Testers

**Deliverables:**

- Working prototype
- Source code repository
- Unit test documentation
- Module-wise progress report

## **PHASE 4: Testing & Quality Assurance**

### **Objective:**

Ensure the system meets functional, performance, and security standards.

### **Duration: 3–4 weeks**

### **Tasks:**

#### **1. Testing Plan Creation**

- Prepare detailed **Test Plan** and **Test Cases**.

#### **2. Testing Types**

- **Functional Testing** – validate each feature against requirements.
- **Integration Testing** – ensure modules communicate correctly.
- **Performance Testing** – test load, response time, and concurrency.
- **Security Testing** – check login, access control, and data encryption.
- **Usability Testing** – ensure intuitive and error-free UI.

#### **3. Bug Reporting & Fixing**

- Document bugs in test reports.
- Retest after fixes.

#### **4. User Acceptance Testing (UAT)**

- Real hospital staff validate system performance.
- Collect approval for deployment.

### **Roles Involved:**

- QA Testers
- Developers
- End Users (for UAT)
- Project Manager

### **Deliverables:**

- Test Plan & Test Case Documents
- Bug Report & Resolution Sheet
- UAT Sign-off Document & Final Quality Report

- **PHASE 5: Deployment & User Training**

**Objective:**

Deploy the system in a live environment and train users.

**Duration: 2–3 weeks**

**Tasks:**

**1. Deployment Preparation**

- Set up production server.
- Perform data migration from old systems.
- Final configuration and security audit.

**2. Go-Live Execution**

- Deploy the system and monitor performance for initial 48–72 hours.
- Create backups and recovery plans.

**3. User Training**

- Conduct workshops for hospital staff.
- Prepare user manuals and video tutorials.

**4. Post-Deployment Monitoring**

- Track system stability and resolve post-launch issues.

**Roles Involved:**

- DevOps Engineer
- System Administrator
- Project Manager
- Trainers

**Deliverables:**

- Deployed Live System
- Training Materials (Manuals, Tutorials)
- Deployment Report

## **Maintenance & Continuous Improvement**

### **Objective:**

Provide ongoing support, updates, and enhancements.

### **Duration: Continuous (Post-deployment phase)**

### **Tasks:**

#### **1. Monitoring and Support**

- Monitor logs and performance metrics.
- Address bug reports or user feedback.

#### **2. System Updates**

- Add new modules (e.g., predictive stock analytics, mobile app).
- Update according to regulatory or organizational changes.

#### **3. Performance Optimization**

- Improve database queries, reduce response time.
- Implement data archiving for old records.

### **Roles Involved:**

- Support Engineer
- Maintenance Team
- Product Manager

### **Deliverables:**

- Maintenance Logs
- Updated Documentation
- Enhancement Reports



## Suggested Timeline (Example for 6–8 Month Project)

Phase	Duration	Key Output
1. Initiation & Analysis	3 weeks	SRS Document
2. System Design	4 weeks	Architecture, ERD, UI Design
3. Development	8 weeks	Working Prototype
4. Testing & QA	4 weeks	Test Reports, UAT Signoff
5. Deployment & Training	3 weeks	Live System, Training Completed
6. Maintenance	Continuous	System Updates



## Optional Advanced Features

- **AI-based Demand Forecasting:** Predict medicine usage based on past data.
- **IoT Integration:** Automatic stock update via smart shelves or RFID.
- **Mobile Application:** Real-time inventory visibility for staff.
- **Audit Logs:** Track all user activities for compliance.
- **Multi-Branch Inventory:** Centralized control for multi-location hospitals.

## Conclusion:

The **Medical Inventory Management System** project successfully addresses the crucial need for effective and efficient management of medical supplies in healthcare organizations. Through the implementation of this system, hospitals and clinics can maintain **accurate, real-time records** of their medicines, consumables, and equipment, thereby minimizing losses due to expiry, theft, or overstocking.

The project systematically progressed through well-defined phases — **requirement analysis, system design, development, testing, deployment, and maintenance** — ensuring a robust and reliable software solution.

By automating manual inventory processes, the system: