PHASE III: REQUIREMENT ANALYSIS

Field	Details
Date	05 November 2025
Team ID	NM2025TMID07730
Project Name	Medical Inventory Management System
Maximum Marks	4 Marks

Title: Requirement Analysis for **"Medical Inventory Management System"**

1. Objective

The Requirement Analysis Phase focuses on identifying, defining, and validating the functional and technical requirements needed to develop the Medical Inventory Management application on the Salesforce platform.

This document ensures clarity of scope, system behaviour, data structure, automation needs, and performance expectations providing a strong foundation for the effective design, development, and implementation of the Medical Inventory Management System.

2. Scope of the System

The Medical Inventory Management System is designed to digitize and automate end-to-end processes involved in managing medical supplies, equipment, and stock levels within a healthcare organization.

The scope includes:

- Digital record-keeping of medical items, suppliers, and purchase orders
- Automated stock level tracking and low-inventory alerts
- Purchase order generation and vendor management
- Dashboard and report-based analytics for real-time monitoring
- Secure, role-based access for Admin, Staff, and Inventory Managers

3. Functional Requirements

3.1 Core Functional Modules

Module	Functionality	Salesforce Feature Used
Item Catalog	Maintain master data for medical items (name, category, UOM, min/max levels)	Custom Object, Validation Rules, Record Types
Supplier/Vendor Management	Store vendor profiles, contacts, terms, and ratings	Custom Object, Related Contacts, Reports
Purchase Orders (PO)	Create/approve POs; auto-generate PO IDs; track statuses (Draft→Approved→Received/Closed)	Custom Object, Flow, Approval Process, Email Alert

Goods Receipt (GRN)	Receive items against PO; capture batch/lot, MRP, expiry, quantity received/short	Custom Object, Flow (Screen Flow), Validation Rules
Inventory & Stock Ledger	Real-time stock updates by location/store; FIFO/LIFO issue basis	Custom Object, Roll-Up (DLRS if needed), Apex (if required)
Batch/Lot & Expiry Tracking	Track batch numbers and expiry dates; prevent issuing expired items	Custom Object, Validation Rules, Scheduled Flow
Reorder & Low-Stock Alerts	Auto-alert when stock falls below reorder level; suggest reorder qty	Flow, Email Alert, In-App Notification
Issue/Consumption	Issue items to departments/wards/patients; capture usage	Custom Object, Screen Flow, QR/Barcode field
Returns & Adjustments	Handle returns, damages, and cycle count adjustments with reason codes	Custom Object, Flow, Validation Rules
Barcoding/Scanning (Optional)	Scan barcodes for fast receiving/issuing	Mobile App, QR/Barcode fields, Screen Flow
Dashboard & Analytics	Visibility into stock value, near-expiry items, top-moving items, PO cycle time	Reports, Dashboards
Security & Access	Role-based access for Admin, Inventory Manager, Store Staff, Finance	Profiles, Permission Sets, Sharing Rules
Audit & Compliance	Field history for stock changes, approvals, and adjustments	Field History Tracking, Audit Trail

3.2 System Workflow (Process Flow)

1. Setup: Admin creates Item Catalog and Supplier records, defining UOM and min/max/reorder levels.
2. Procurement: Inventory Manager creates a Purchase Order; Approval Process routes it to approvers.
3. Receiving (GRN): Store receives items against the PO; user records batch/lot, expiry, quantity, and MRP via Screen Flow—stock updates automatically.
4. Quality/Quarantine (Optional): Items can be placed in a quarantine status until QA approval; only then are they added to available stock.
5. Issuance/Consumption: Departments/wards request items; Store issues stock (prefer near-expiry first). Consumption is logged for traceability.
6. Reorder & Alerts: Scheduled Flows check stock levels and expiry dates; low-stock and near-expiry alerts go to Inventory Manager.
7. Adjustments/Returns: Damaged/expired returns and cycle count adjustments are recorded with reasons; approvals applied if needed.
8. Analytics: Dashboards update in real time to show stock valuation, PO status, top movers, near-expiry items, and monthly consumption trends.

4. Technical Requirements

Category	Requirement	Description
Platform	Salesforce Developer Edition	CRM cloud for automation
Programming	Apex (if needed)	Auto constraints & validation
Automation	Salesforce Flow	Auto rent reminder & renewal alerts
Database	Salesforce Objects	Property, Tenant, Lease, Payment
Visualization	Dashboard Builder	Rent status, lease expiry chart
Security	Profiles & Roles	Owner, Tenant, Admin access control

4.1 Software Tools Used

- Salesforce Lightning Experience
- Flow Builder
- Schema Builder
- Reports & Dashboards
- Developer Console
- Data Import Wizard (for bulk item/vendor uploads)
- Setup Audit Trail (for compliance tracking)

4.2 Hardware Requirements

Component	Minimum Requirement
Processor	Intel i3 or above
RAM	4 GB+
Internet	Min 2 Mbps stable connection
Storage	500 MB browser cache
Browser	Chrome / Edge latest

5. Non-Functional Requirements

Category	Requirement	Description
Performance	System must support 500+ active stock items and 100+ daily transactions	Handled efficiently through Salesforce's multi-object architecture and optimized Flows
Security	Role-based access control	Users (Admin, Store Staff, Finance) can access only authorized modules and data; patient-related info secured
Usability	Simple UI for non-technical users	Designed using Salesforce Lightning App Builder with easy navigation and guided Screen Flows

Scalability	Support for multi-location inventory	Dynamic object relationships allow tracking by multiple stores, wards, or hospitals
Reliability	99.9% uptime	Ensured by Salesforce's secure and redundant cloud infrastructure
Maintainability	Low-code, easily modifiable system	Flow-based automation and configuration-driven logic reduce Apex dependency
Auditability	Full change tracking	Field History Tracking and Audit Trail enabled for stock, purchase, and adjustment records

6. Data Model Design

Object	**Key Fields (examples)**	**Relationship Type**
Item (Product Master)	Name, Category, UOM, Reorder_Level__c, Min_Level__c, Max_Level__c, Active__c	—
Supplier (Vendor)	Name, Contact_Person__c, Phone, Email, Payment_Terms__c, Rating__c	—
Inventory Location	Name, Type (Main Store/Ward/Pharmacy), Address__c, Active__c	—
Purchase Order (PO)	AutoNumber PO-{0000}, Supplier__c, PO_Date__c, Status__c (Draft/Approved/Received/Closed)	Lookup to Supplier
PO Line Item	Item__c, Quantity__c, Unit_Price__c, Tax__c, Amount__c (formula)	Master-Detail to Purchase Order, Lookup to Item
Goods Receipt (GRN)	GRN_Number__c, PO__c, Received_Date__c, Status__c (Partial/Complete)	Lookup to Purchase Order
GR Line (Batch/Lot)	Item__c, Batch_No__c, Expiry_Date__c, Quantity_Received__c, MRP__c	Master-Detail to Goods Receipt, Lookup to Item
Stock Ledger (Transactions)	Item__c, Location__c, Txn_Type__c (GRN/ISSUE/ADJ/RETURN), Qty_In__c, Qty_Out__c, Reference__c	Lookup to Item, Lookup to Inventory Location
Issue (Header)	Issue_No__c, Location__c, Department__c / Patient__c, Issue_Date__c	Lookup to Inventory Location
Issue Line	Item__c, Quantity__c, Batch_No__c (optional), Remarks__c	Master-Detail to Issue, Lookup to Item
Return/Adjustment	Type__c (Return/Damage/CycleCount), Item__c, Location__c, Quantity__c, Reason__c	Lookup to Item, Lookup to Inventory Location

Example Formula Fields

PO Line Amount (on PO Line Item)

Amount__c = Quantity__c * Unit_Price__c

PO Total (on Purchase Order, Roll-Up Summary)

Sum over child PO Line Item → Amount__c

Days to Expiry (on GR Line)

Days_to_Expiry__c = Expiry_Date__c - TODAY()

Near Expiry (Checkbox) (on GR Line)

Near_Expiry__c = (Expiry_Date__c - TODAY()) <= 30

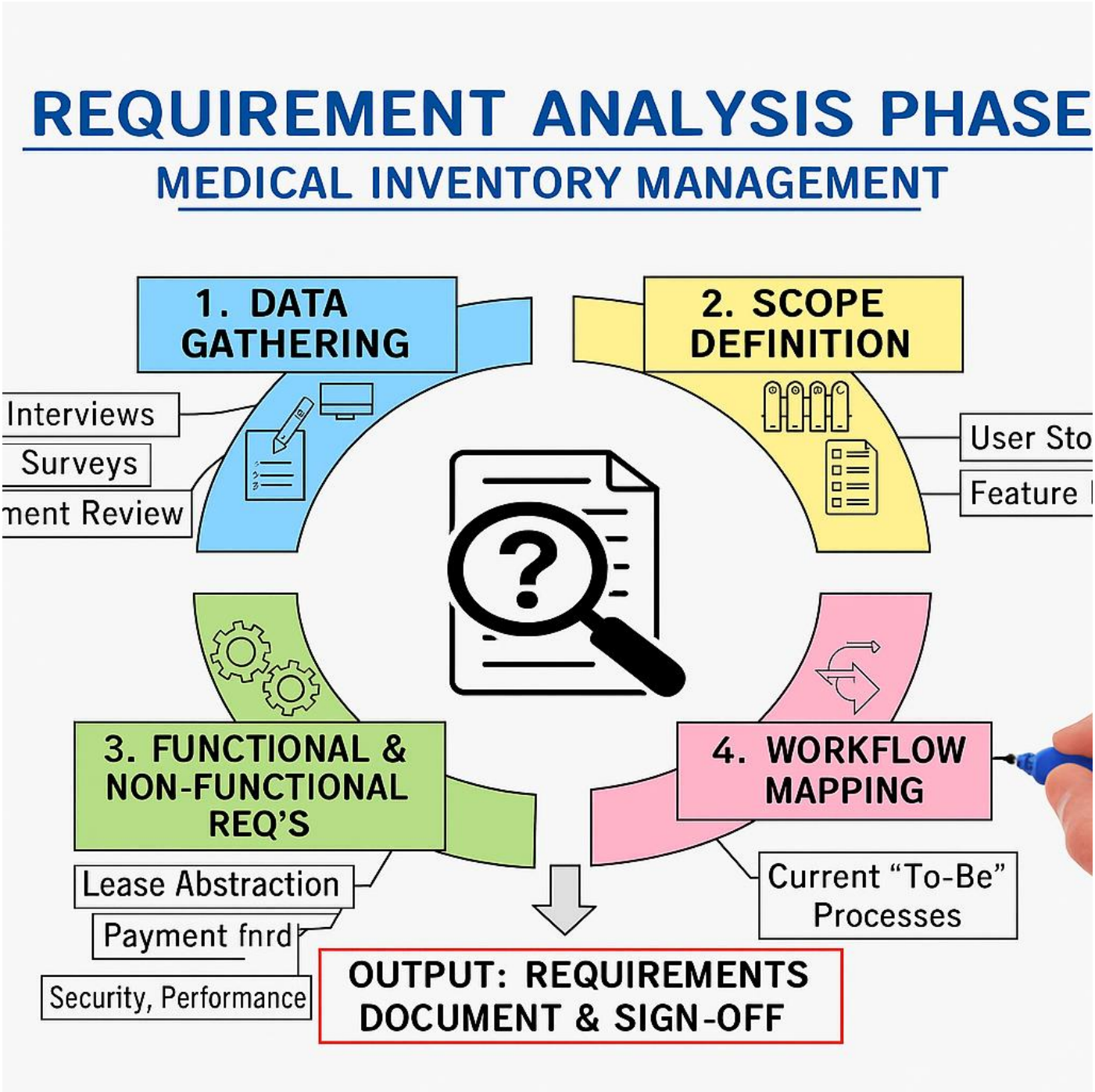
Stock on Hand (per Item per Location)

If you keep Stock Ledger totals on a helper object or Item–Location junction:

On_Hand__c = Qty_In_Total__c - Qty_Out_Total__c (via roll-ups / DLRS)

Suggested Reorder Quantity (on Item, informational)

Suggested_Reorder_Qty__c = MAX(0, Reorder_Level__c - Current_Stock__c)



7. User Interface (UI) Requirements

Navigation tabs (Salesforce Lightning App):

1. Items (Catalog)
2. Suppliers
3. Purchase Orders
4. Goods Receipts (GRN)
5. Inventory (Stock Ledger)
6. Issues (Department/Patient)
7. Returns & Adjustments
8. Locations (Stores/Wards)
9. Reports & Dashboards

10. Alerts (Low-Stock / Near-Expiry)

7.1 UI Expectations

- One-click Create PO from Item or Supplier; auto-calculate Line Amount, Taxes, PO Total.
- Guided Screen Flows for GRN and Issue with optional barcode/QR input and quick actions (Receive, Issue, Adjust).
- Status indicators/badges:
- PO: Draft / Approved / Received / Closed
- Stock: In-Stock / Low / Critical (based on thresholds)
- Expiry: Days to Expiry, Near-Expiry flag
- Issue: Pending / Completed
- Mobile-friendly layouts (Salesforce mobile): compact forms, pinned list views, and quick actions on record headers.
- Home page dashboard embed with KPIs: On-hand Value, Low-Stock Count, Near-Expiry Items, POs Pending Approval, Top Movers.
- List view enhancements: conditional highlighting (e.g., red for ≤0 stock / expired), inline edit, filters by Location, Category, Status.
- Record page design: Dynamic Forms, key related lists (Lines, Batches, Movements) above the fold; utility bar for Alerts and Scanner.
- Defaults & validation in UI: default Location/UOM, prevent issuing expired or negative stock, show remaining quantity against PO.
- Accessibility: clear labels/tooltips, keyboard navigation, high-contrast icons for color-blind friendliness.

8. System Validation Requirements

Test Area	Description	Expected Result
Flow Execution	Automatic low-stock and expiry alerts triggered daily via Scheduled Flow	Email / In-App notification sent to Inventory Manager
Reorder Alert Validation	When item stock falls below reorder level	Alert generated and logged in “Alerts” object
Purchase Order Automation	Auto PO number generation and approval routing	PO number auto-generated and status updates from Draft → Approved
Goods Receipt Validation	GRN records items correctly against PO and updates stock quantity	Stock ledger updated; status changes to Received / Closed
Expiry Validation	Prevent issuing expired items	Validation rule blocks transaction and shows error message
Dashboard Validation	Stock levels, PO summary, and near-expiry chart	Dashboards auto-refresh with latest data
Permission Test	Role-based visibility across modules	Store staff sees only assigned location; admin sees all data

9. Risk Identification & Mitigation

Risk	Description	Mitigation
Data Loss	Records or files (e.g., invoices, GRN docs) deleted accidentally	Enable Salesforce Recycle Bin; schedule weekly data export/backup
Wrong Access	Unauthorized users view/edit restricted data	Enforce Profiles, Permission Sets, and Sharing Rules
Duplicate Records	Same item or supplier added multiple times	Add duplicate rules and validation logic
Stock Inaccuracy	Manual entry errors during GRN or Issue	Use validation rules and Screen Flows with guided input
Automation Failure	Flow or approval errors stop critical operations	Add error handling, debug logs, and Apex test coverage
Compliance Risk	Expired or untraceable batch issued	Implement batch-level expiry validation and audit reports
Network or Connectivity Issues	Users unable to access cloud app	Ensure offline documentation and retry logic for imports/exports

10. Summary

The Requirement Analysis Phase establishes a complete blueprint for the Medical Inventory Management System.It documents the functional, technical, data, and validation requirements, ensuring a smooth transition into the system design and development phases.

The Key achievements of this phase:

- Defined all core modules such as Item Management, Purchase Orders, Inventory Tracking, and Alerts
- Mapped the complete data model and object relationships across all Salesforce objects
- Established user interface layouts, dashboards, and automation flows for business efficiency
- Outlined validation, security, and performance standards for reliability
- Identified potential risks and documented mitigation strategies

This phase ensures that the Medical Inventory Management project is clear, well-structured, scalable, and implementation-ready within the Salesforce environment.