Project Design Phase

Proposed Solution

Date: 10 November 2025
Team ID: NM2025TMID02283
Project Name: Medical Inventory Management in Salesforce (HealthStock)
Maximum Marks: 2 Marks

Proposed Solution Template

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Hospitals, clinics and pharmacies often face stockouts, expired medicines, over-stocking, and lack of real-time visibility across sites. Manual recordkeeping and siloed systems cause delays in patient care, financial losses, and regulatory non-compliance.
2.	**Idea / Solution description**	Build a Medical Inventory Management solution on Salesforce that centralizes inventory across sites using custom objects (Inventory_Item, Lot, Stock_Location, Transaction), automates replenishment, enforces expiry & batch tracking, and provides real-time dashboards and mobile scanning support. Use declarative automation (Flows, Process Builder where needed) plus minimal Apex for complex validations and integrations (e.g., barcode scanners, ERP, procurement systems).
3.	**Novelty / Uniqueness**	Combines full lot/expiry traceability, role-based mobile scanning, and Salesforce-native approvals/workflows for medical inventory — eliminating the need for separate legacy inventory apps. Leverages platform features (Sales Cloud & Health Cloud concepts) for patient-centric allocation and compliance reporting.

S.No.	Parameter	Description
4.	**Social Impact / Customer Satisfaction**	Reduces medication stockouts and wastage from expired stock, improves patient safety and treatment continuity, speeds up clinical workflows for nurses/pharmacists, and increases trust through auditable chain-of-custody for controlled substances.
5.	**Business Model (Revenue Model)**	SaaS or managed package offered to hospitals/clinics: subscription tiers by number of locations and features (basic tracking, lot/expiry, integrations, analytics). Saves operational costs by optimizing stock levels and reducing emergency procurement.
6.	**Scalability of the Solution**	Designed on Salesforce multi-tenant architecture: supports scaling from single clinic to multi-hospital networks. Modular design allows adding modules (procurement automation, supplier onboarding, cold-chain telemetry, RFID/IoT) and role-based restrictions for large organizations.

Solution Description (Detailed)

Overview:
HealthStock is a Salesforce-native inventory management capability tailored for medical supplies and pharmaceuticals. It centralizes items, lots, locations and transactions in Salesforce, enforces expiry checks, and integrates with barcode scanners and external procurement/ERP systems.

Key Data Model (custom objects & fields):

- **Inventory_Item** (SKU, name, category, unit_of_measure, reorder_point, reorder_qty)
- **Lot** (Inventory_Item lookup, lot_number, expiry_date, manufacture_date, quantity_on_hand, location lookup)
- **Stock_Location** (warehouse/clinic/pharmacy, address, contact)
- **Inventory_Transaction** (type: issue/receive/transfer/adjustment, date, quantity, related_lot, performed_by)
- **Allocation / Reservation** (optional: link to Patient or Order to reserve stock)

Business Logic & Automation:

- **Record-level validation**: Prevent issuing an item if lot is expired or quantity insufficient — implemented with Flow + Apex validation when needed.

- **Expiry prevention rule**: Before saving a transaction that will consume stock, check lot.expiry_date >= today; block and display reason if expired.
- **Reorder automation**: Scheduled Flow runs daily to evaluate Inventory_Item across locations; generates Purchase Requests or Notifies procurement when quantity ≤ reorder_point.
- **Batch/lot tracking**: All issues consume specific Lot records; audit trail kept via Inventory_Transaction object for traceability.
- **Approvals**: High-value adjustments or controlled substance transfers trigger Salesforce approval processes.
- **Mobile scanning**: Use Salesforce Mobile + scanner SDK (or integrate with Zebra/Scandit) to scan barcodes/QRs on lots for fast receiving/issuing. Scanned data calls a Flow that creates Inventory_Transaction.
- **Integrations**:
    - **ERP / Purchasing**: Outbound integration (REST/APEX callouts or middleware like MuleSoft) to submit POs and receive confirmations.
    - **Cold-chain telemetry (optional)**: Ingest IoT temperature logs to mark lots as quarantined if thresholds exceeded.
- **Reporting & Dashboards**: Real-time dashboards for stock levels, soon-to-expire lots, usage trends, stock valuation, and audit logs. Exportable reports for compliance and regulatory inspections.

Security & Compliance:

- **Profiles/Permission Sets**: Restrict who can adjust stock, create purchase requests, or delete records.
- **Field-level encryption or Shield Platform Encryption** (if required) for sensitive data.
- **Audit Trail**: Standard field history tracking and Inventory_Transaction audit table to satisfy regulatory audits.
- **Sharing**: Location-based sharing model to limit access to only relevant clinics/warehouses.

Implementation Approach:

1. **Discovery & Data Migration** — map legacy inventory & supplier data to new objects; migrate using Data Loader.
2. **Core Build (Sprint 1–2)** — implement data model, basic Flows for receive/issue, mobile scanning integration PoC.
3. **Advanced Automation (Sprint 3–4)** — reorder engine, approvals, ERP integration.
4. **Testing & Validation** — unit tests for Apex, Flow testing, UAT with pharmacists/nurses.
5. **Go-live & Training** — phased rollout; in-app guidance and short training sessions for users.

6. **Monitoring & Support —** scheduled health checks, automation logs, and performance tuning.

Fallbacks & Edge Cases:

- Negative inventory prevented by transactional checks; adjustments require reason and approval.
- Quarantined lots: prevent consumption until cleared by quality user.
- Offline mobile scanning: queue scans locally and sync when network available (via mobile SDK).

Minimal Technical Snippets (illustrative)

- **Flow**: A before-save record-triggered Flow on Inventory_Transaction that checks Lot.quantity_on_hand and Lot.expiry_date; throws error to block transaction if invalid.
- **Apex (only if necessary)**: A small trigger handler to enforce cross-object checks in complex race conditions, with unit tests covering expiry and concurrent consumption.

Conclusion

HealthStock — the proposed Salesforce Medical Inventory Management solution — addresses critical gaps in visibility, compliance, and operational efficiency for healthcare providers. By using Salesforce native objects, declarative automation, selective Apex, and mobile scanning, the solution reduces stockouts and waste, improves patient safety, and delivers auditable, scalable inventory controls that easily extend to procurement and IoT monitoring.

Reference: Designed as a Salesforce-native managed package approach; mobile scanning integrations and ERP connectors are optional add-ons.
