Garage Management System

# A Salesforce Implementation

## Author: ANYAM MOUNIKA LAKSHMI

**Date:** [29-06-2025]

**Table of contents**

# 1. INTRODUCTION

## 1.1 Project Overview

This report details the design, development, and implementation of a custom Garage Management System built on the Salesforce platform. The project aims to address critical inefficiencies and risks prevalent in traditional, often manual, inventory management processes within healthcare facilities such as hospitals, clinics, and emergency care centers.

The solution leverages the power of Salesforce's declarative and programmatic capabilities to create a centralized, real-time, and automated system for tracking medical supplies. This includes everything from basic consumables like gloves and syringes to high-value pharmaceuticals and surgical equipment.

The system provides a 360-degree view of the entire inventory lifecycle: procurement from suppliers, stock management across multiple locations (e.g., central warehouse, operating rooms, emergency wards), real-time consumption tracking, automated reordering based on predefined thresholds, and robust reporting for strategic decision-making. By digitizing and automating these processes, the project delivers a scalable, secure, and compliant solution designed to reduce operational costs, minimize waste due to expired stock, and ultimately enhance patient care by preventing stockouts of critical items.

## 1.2 Purpose

The primary purpose of this project is to develop a robust technological solution that solves the core challenges of medical supply chain management. The key objectives are:

- **To Improve Operational Efficiency:** Replace manual, paper-based, or spreadsheet-reliant tracking with a fully automated system, reducing human error and freeing up staff to focus on patient-facing activities.
- **To Ensure Supply Availability:** Implement intelligent alerts and automated reordering workflows to prevent stockouts of life-saving medical supplies, thereby improving patient safety and quality of care.
- **To Reduce Waste and Control Costs:** Enable precise tracking of inventory items, including batch numbers and expiration dates, to facilitate a First-In, First-Out (FIFO) or First-Expired, First-Out (FEFO) consumption strategy, significantly reducing spoilage and waste.
- **To Provide Real-Time Visibility and Analytics:** Offer comprehensive dashboards and reports that give administrators and stakeholders immediate insights into inventory levels, consumption patterns, supplier performance, and budget adherence.
- **To Enhance Compliance and Traceability:** Create an auditable trail for every inventory item, crucial for regulatory compliance (e.g., FDA, HIPAA) and for managing product recalls efficiently by quickly identifying affected batches and their locations.

# 2. IDEATION PHASE

## 2.1 Problem Statement

Healthcare facilities consistently face significant operational and clinical challenges due to inadequate inventory management systems. The core problems can be summarized as follows:

1. **High Risk of Stockouts:** Manual tracking methods are prone to error and lag, leading to unforeseen shortages of critical medical supplies, which can directly compromise patient treatment and safety.
2. **Excessive Waste and Financial Loss:** The inability to effectively track expiration dates results in the disposal of expired stock. Overstocking due to poor demand forecasting ties up capital and increases carrying costs.
3. **Lack of Centralized Visibility:** Inventory is often siloed in different departments or locations (e.g., pharmacy, OR, ER) with no unified view, making it impossible for administrators to get an accurate, real-time picture of enterprise-wide stock levels.
4. **Inefficient and Error-Prone Procurement:** The reordering process is typically manual, reactive, and time-consuming, involving phone calls, emails, and paperwork, which increases administrative overhead and the risk of ordering mistakes.
5. **Compliance and Recall Management Burdens:** In the event of a product recall, manually tracing specific lot numbers across thousands of items in various locations is a daunting, slow, and often inaccurate process, posing a significant compliance risk.

This project directly addresses these issues by creating a single source of truth on the Salesforce platform.

### 2.2 Empathy Map Canvas - Hospital Inventory Manager

To better understand our primary user, we created an empathy map for the persona of "Maria, a Hospital Inventory Manager."

[Image: A visually designed Empathy Map Canvas graphic. The following text would be arranged within that graphic.]

**Persona:** Maria, Hospital Inventory Manager

| **SAYS** | **THINKS** |
|---|---|
| "Did anyone check the stock in Ward 5 today?" | "I hope we don't run out of surgical kits during the night shift." |
| "I have to call three different suppliers to get a price comparison." | "This paperwork is taking all my time. I barely have time to walk the floor." |
| "Which batch of catheters was recalled? We have so many." | "Why can't I just see everything on one screen?" |
| "Another urgent order. The budget is going to be a mess this month." | "I feel like I'm always putting out fires instead of planning ahead." |
| **DOES** | **FEELS** |
| Manually counts inventory on clipboards. | **PAINS:** |
| Spends hours on the phone and email with suppliers. | - Stressed about potential stockouts. |
| Updates complex, multi-tabbed Excel spreadsheets. | - Frustrated with repetitive, manual tasks. |
| Rushes to place emergency orders. | - Anxious about compliance and audit risks. |

| | |
|---|---|
| Tries to reconcile invoices with delivery notes. | - Overwhelmed by the lack of data visibility.<br>**GAINS:**<br>- Peace of mind from knowing stock levels are accurate.<br>- More time for strategic tasks like supplier negotiation.<br>- Confidence in meeting compliance requirements.<br>- Empowerment through data-driven decision-making. |

## 2.3 Brainstorming & Mind Mapping

A brainstorming session was conducted to explore all potential features and functionalities. The results were organized into a mind map to visualize the system's components.

[Image: A mind map diagram with "Medical Inventory System" at the center. The main branches should be the points listed below.]

Central Idea: A Unified Garage Management System on Salesforce

- **Core Modules:**
    - Inventory/Product Management (Catalog, Batch/Lot #, Expiration Date)
    - Supplier Management (Supplier Info, Contracts, Performance)
    - Purchase Order Management (Creation, Approval, Tracking)
    - Stock Location Management (Warehouse, Departments, Carts)

    - Consumption Tracking
- **Automation:**
    - Low-Stock Alerts (Email, Chatter, Tasks)
    - Automated Reorder Point (ROP) Triggers
    - Purchase Order Approval Flows
    - Expiration Date Notifications
- **User Experience & Interface:**
    - Central Dashboard for KPIs
    - Mobile-friendly Interface (Salesforce Mobile App)
    - Barcode/QR Code Scanning Integration
    - Customizable List Views and Reports
- **Reporting & Analytics:**
    - Inventory Valuation Reports
    - Consumption Trend Analysis
    - Supplier Lead Time Reports
    - Expired/Near-Expiry Stock Reports
    - Budget vs. Actual Spend Dashboards
- **Integration & Security:**
    - HIPAA Compliance
    - Role-Based Access Control
    - Potential API for ERP/Financial System Integration
    - Supplier Community Portal (Experience Cloud)

# 3. REQUIREMENT ANALYSIS

## 3.1 Customer Journey Map

This map illustrates the process an inventory manager like Maria goes through to restock an item, highlighting pain points and opportunities for our Salesforce solution.

**Journey:** Restocking Surgical Gloves

| Stage | Actions | Touchpoints | Pain Points | Salesforce Solution Opportunity |
|---|---|---|---|---|
| **1. Need Awareness** | A nurse reports low stock. Maria manually checks the supply closet. | Physical Stock, Word of Mouth, Spreadsheet | - Delayed awareness. <br>- Inaccurate manual count. | - **Automated Alert:** System automatically flags stock below reorder level. <br>- **Real-time Dashboard:** Shows current stock at a glance. |
| **2. Order Creation** | Maria finds the supplier's contact info. Creates a PO in a Word template. Emails it. | Phone, Email, Excel/Word | - Time-consuming. <br>- Risk of typos/errors in PO. <br>- No centralized record. | - **Automated PO Generation:** Create PO from a template with one click. <br>- **Supplier Object:** All |

| | | | | |
|---|---|---|---|---|
| | | | | supplier data is stored in Salesforce. |
| **3. Approval** | Maria prints the PO and walks it to the department head for a signature. | Paper, In-person | - Slow physical process. <br>- Lack of tracking for approval status. | - **Approval Process:** Submit PO for approval digitally. Approvers are notified automatically via email/mobile. |
| **4. Order Tracking** | Maria calls the supplier to check on the delivery status. | Phone Call | - No real-time visibility. <br>- Manual follow-up required. | - **Status Field:** PO record is updated with status (e.g., Sent, Acknowledged, Shipped). <br>- **Supplier Portal:** Suppliers can update status directly. |
| **5. Receiving Stock** | Delivery arrives. Maria manually checks items | Paper PO, Delivery Note | - Prone to error in verifying quantities. <br>- Difficult to | - **Barcode Scanning:** Scan items on arrival to update quantities |

| | | | | |
|---|---|---|---|---|
| | against the paper PO. | | record batch/expiry info. | automatically. <br>- **Record Creation:** Log batch and expiry dates in the system. |
| **6. Updating System** | Maria updates her master Excel spreadsheet with the new quantities. | Excel Spreadsheet | - Redundant data entry. <br>- High chance of error. <br>- Not a real-time system. | - **Automated Update:** Receiving the stock automatically updates the Inventory Item record's quantity. No manual entry needed. |

### 3.2 Solution Requirements

| ID | Category | Requirement Description | Priority |
|---|---|---|---|
| FR-01 | Inventory Mgt. | The system must allow users to create and manage a master catalog of all medical inventory items. | Must-Have |
| FR-02 | Inventory Mgt. | Each inventory item record must track: Name, SKU, Category, Supplier, Quantity on Hand, Reorder Level, and Location. | Must-Have |
| FR-03 | Inventory Mgt. | The system must support tracking of Batch/Lot Number and Expiration Date for relevant items. | Must-Have |

|  |  |  |  |
|---|---|---|---|
| FR-04 | Procurement | Users must be able to create, edit, and track Purchase Orders (POs) linked to suppliers and inventory items. | Must-Have |
| FR-05 | Automation | The system must automatically generate a notification (Task or Email) when an item's quantity falls below its reorder level. | Must-Have |
| FR-06 | Automation | The system must have a configurable approval process for Purchase Orders exceeding a certain value. | Must-Have |
| FR-07 | Reporting | The system must provide a dashboard showing key KPIs: Items below reorder level, value of inventory, pending POs. | Must-Have |
| FR-08 | Reporting | Users must be able to generate a report of all items expiring within a specified timeframe (e.g., next 90 days). | Must-Have |
| FR-09 | User Interface | The system must support barcode/QR code scanning via the Salesforce Mobile App to update or consume inventory. | Should-Have |
| FR-10 | Supplier Mgt. | The system must store and manage supplier information, including contacts, addresses, and associated items. | Must-Have |
| FR-11 | Location Mgt. | The system must allow inventory to be assigned to and tracked across multiple physical stock locations. | Should-Have |

| ID | Category | Requirement Description |
|---|---|---|
| NFR-01 | Performance | All primary record pages (e.g., Inventory Item, Purchase Order) must load in under 3 seconds on a standard internet connection. |
| NFR-02 | Security | The system must be HIPAA compliant. All data must be encrypted at rest and in transit. |
| NFR-03 | Security | User access must be role-based. For example, a nurse can only view and request stock, while a manager can approve POs. |
| NFR-04 | Scalability | The system must be able to handle up to 50,000 inventory item records and 5,000 purchase orders per year without degradation. |
| NFR-05 | Usability | The system must be accessible and fully functional on the Salesforce Mobile App for on-the-go inventory checks. |
| NFR-06 | Reliability | The system must have an uptime of 99.9% or higher, consistent with Salesforce platform SLAs. |

### 3.3 Data Flow Diagram (DFD)

[Image: A formal Data Flow Diagram (Level 1) showing the main processes: "Manage Inventory," "Process Orders," "Manage Suppliers," and "Generate Reports," with data stores like "Inventory Store," "PO Store," and "Supplier Store," and external entities like "Inventory Manager," "Supplier," and "Department Head."]

**Description of DFD:**

- **External Entities:**
  - **Inventory Manager:** Initiates POs, receives stock, views reports.
  - **Department Head:** Approves/rejects POs.
  - **Supplier:** Receives POs, provides shipment information.
- **Main Processes:**
  1. **Process 1.0 - Manage Inventory:** Handles the creation and updating of inventory items, including quantity adjustments upon consumption or receipt of new stock.
  2. **Process 2.0 - Process Orders:** Manages the entire lifecycle of a Purchase Order, from creation and submission to approval and fulfillment.
  3. **Process 3.0 - Generate Reports:** Queries the data stores to produce dashboards and reports on inventory levels, expiry dates, and procurement analytics.
- **Data Stores:**
  - **D1 - Inventory Items:** Stores all data related to medical supplies (SKU, quantity, location, expiry).
  - **D2 - Purchase Orders:** Stores all PO records, including line items, status, and approval history.
  - **D3 - Suppliers:** Stores information about all approved vendors.

### 3.4 Technology Stack

- **Platform:** Salesforce Lightning Platform (Salesforce Health Cloud or Sales/Service Cloud).
- **Core Components:**
    - **Custom Objects:** For modeling inventory, purchase orders, suppliers, and stock locations.
    - **Salesforce Flow:** For all declarative automation, including low-stock alerts and record-triggered updates.
    - **Approval Processes:** For managing the multi-step approval of Purchase Orders.
    - **Lightning App Builder:** For creating the custom application, home pages, and record pages.
    - **Reports and Dashboards:** For all analytics and business intelligence.
- **Code-Based Components:**
    - **Apex:** For complex business logic that cannot be handled by Flow (e.g., complex validation rules, future integration callouts).
    - **Lightning Web Components (LWC):** For any highly custom user interface components, such as a specialized barcode scanning interface or a multi-item order form.
- **Mobile:** Salesforce Mobile App.
- **External Services (Future Scope):** REST/SOAP APIs for potential integration with external financial ERPs or supplier systems.

# 4. PROJECT DESIGN

## 4.1 Problem-Solution Fit Matrix

This matrix clearly maps the identified problems to the specific features designed in our solution.

| Problem Statement | Proposed Solution Feature(s) |
|---|---|
| **High Risk of Stockouts:** | - Automated Low-Stock Alerts via Email/Tasks. <br>- Real-time Inventory Dashboard. |
| **Excessive Waste & Financial Loss:** | - Expiration Date field on Inventory records. <br>- "Items Nearing Expiry" Report. |
| **Lack of Centralized Visibility:** | - Centralized Inventory object model. <br>- Reports/Dashboards showing enterprise-wide stock. |
| **Inefficient and Error-Prone Procurement:** | - Templated Purchase Order generation. <br>- Digital Approval Processes. |
| **Compliance and Recall Management Burdens:** | - Batch/Lot Number tracking on all items. <br>- Global search to quickly locate specific batches. |

## 4.2 Proposed Solution

The proposed solution is a custom Salesforce Lightning App named "GarageCare Command".

The application consists of the following key components:

1. **Custom Objects:** A relational data model forms the backbone of the system.
   - Inventory Item: The master record for each unique medical supply.

   - Stock Location: Represents physical locations (e.g., Central Pharmacy, OR-3). An Inventory Item can exist in multiple Stock Locations.
   - Purchase Order: The header record for a procurement request.
   - PO Line Item: The junction object linking a Purchase Order to the Inventory Items being ordered.
   - Supplier: An account-like object to store all vendor information.
2. **Automation:**
   - A **record-triggered Flow** on the Inventory Item object fires whenever the Quantity_on_Hand__c field is updated. If the new quantity is below the Reorder_Level__c, a Task is created for the Inventory Manager and an email alert is sent.
   - An **Approval Process** is initiated when a Purchase Order is submitted. POs with a total value over $5,000 require two-level approval (Inventory Manager -> Department Head).
3. **User Interface:**
   - **Home Page Dashboard:** The default landing page for the app, featuring key reports like "Items Below Reorder Level," "Pending POs," and "Stock Nearing Expiry."
   - **Lightning Record Pages:** Highly customized pages for Inventory Item and Purchase Order to show related information and guide users through processes. For example, the PO page will have a "Submit for Approval" button that is visible only at the correct stage.

### 4.3 Solution Architecture

[Image: A detailed ERD showing the custom objects and their relationships. For example:]
[Supplier] --< [Purchase Order] --< [PO Line Item] >-- [Inventory Item]
[Stock Location] --< [Inventory Stock Junction] >-- [Inventory Item]

**Key Objects and Relationships:**

- **Supplier__c:** Master object for vendors.
- **Purchase_Order__c:**
  - Lookup to Supplier__c.
- **Inventory_Item__c:**
  - Master record for products.
- **PO_Line_Item__c:**
  - Master-Detail to Purchase_Order__c.
  - Lookup to Inventory_Item__c.
  - Fields: Quantity__c, Unit_Price__c, Total_Price__c (formula).
- **Stock_Location__c:**
  - Represents physical storage areas.
- **Inventory_Location_Junction__c:** (To handle many-to-many relationship between Items and Locations)
  - Master-Detail to Inventory_Item__c.
  - Master-Detail to Stock_Location__c.
  - Fields: Quantity_in_Location__c, Batch_Number__c, Expiration_Date__c.

Access is controlled via Profiles and Permission Sets to ensure users only see and do what is necessary for their role.

| User Role | Profile | OWD (Inventory Item) | Key Permissions |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **System Administrator** | System Admin | Public Read/Write | Full CRUD access on all objects. Customize Application. |
| **Inventory Manager** | Custom: "Inv. Manager" | Public Read/Write | Create/Edit/Delete on all inventory objects. Submit POs for approval. |
| **Department Head** | Custom: "Approver" | Public Read-Only | Read-only on all inventory objects. Can only "Approve/Reject" records they are assigned. |
| **Nurse / End User** | Custom: "End User" | Public Read-Only | Read-only access to view stock levels. May have ability to create a "Stock Request" record. |

[Image: A flowchart or business process model diagram illustrating the "Low Stock to Reorder" process. It should show the trigger (quantity update), the decision diamond (is quantity < reorder level?), and the resulting actions (create task, send email).]

**Process Flow for Low-Stock Alert:**

1. **Trigger:** An Inventory_Location_Junction__c record is updated (e.g., stock is consumed).
2. **Flow Start:** A record-triggered Flow initiates.
3. **Decision:** The Flow checks if Quantity_in_Location__c < Reorder_Level__c on the parent Inventory_Item__c.

4. **Action 1 (If True):** Create a new Task record assigned to the Inventory Manager with the subject "Low Stock Alert for [Item Name]".
5. **Action 2 (If True):** Send an Email Alert using a predefined template to the Inventory Manager's email address.
6. **End:** The Flow finishes.

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Plan (Gantt Chart)

The project was executed over 10 weeks, following an agile-like methodology with clear phases.

[Image: A Gantt chart graphic. The table below represents the data for that chart.]

| Phase | Task | Duration (Days) | Start Date | End Date |
|---|---|---|---|---|
| **1. Discovery & Planning** | | **10** | **Week 1** | **Week 2** |
| | Requirement Gathering Workshops | 5 | Day 1 | Day 5 |
| | Finalize Project Scope & Plan | 3 | Day 6 | Day 8 |
| | Setup Salesforce Sandbox | 2 | Day 9 | Day 10 |
| **2. Design & Configuration** | | **15** | **Week 3** | **Week 5** |

| | | | | |
|---|---|---|---|---|
| | Design Data Model & Build Custom Objects | 5 | Day 11 | Day 15 |
| | Configure Profiles & Permission Sets | 3 | Day 16 | Day 18 |
| | Build Core Automation (Flows, Approvals) | 5 | Day 19 | Day 23 |
| | Design Lightning App & Record Pages | 2 | Day 24 | Day 25 |
| **3. Development & Testing** | | **20** | **Week 6** | **Week 9** |
| | Build Reports and Dashboards | 5 | Day 26 | Day 30 |
| | Develop Custom LWC (if needed) | 5 | Day 31 | Day 35 |
| | System Integration Testing (SIT) | 5 | Day 36 | Day 40 |
| | User Acceptance Testing (UAT) | 5 | Day 41 | Day 45 |
| **4. Deployment & Go-Live** | | **5** | **Week 10** | **Week 10** |
| | Prepare & Execute Deployment to Production | 2 | Day 46 | Day 47 |
| | Data Migration (Initial Upload) | 2 | Day 48 | Day 49 |

| | | | | |
|---|---|---|---|---|
| | Post Go-Live Support & Handover | 1 | Day 50 | Day 50 |

**(Page Break)**

# 6. FUNCTIONAL AND PERFORMANCE TESTING

## 6.1 Performance Testing Strategy

Performance was a key consideration to ensure a responsive user experience. Our strategy included:

- **Salesforce Optimizer:** Ran the Salesforce Optimizer tool before and after development to identify and address any potential performance bottlenecks in our configuration.
- **Page Load Times:** Used browser developer tools to measure the load times of key Lightning Record Pages (Inventory Item, Purchase Order) and the Home Page Dashboard. The target was to keep Core Web Vitals within acceptable limits (< 3 seconds).
- **Apex Code Analysis:** All Apex code was written following best practices to avoid hitting governor limits. This includes bulkifying code and writing efficient SOQL queries.
- **Load Testing (Simulated):** A data load of 50,000 Inventory_Item__c records was performed in a full-copy sandbox to test the performance of list views, reports, and search functionality under a realistic data volume.

## 6.2 Functional Testing (Sample Test Cases)

Rigorous functional testing was performed to validate all requirements.

| Test Case ID | Test Case Description | Test Steps | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| TC-001 | Verify Low-Stock Alert fires correctly. | 1. Create an Inventory Item with Reorder Level = 10. <br>2. Set Quantity on Hand = 11. <br>3. Update Quantity on Hand to 9. | A new Task should be created and assigned to the Inventory Manager. An email alert should be received. | As expected. | Pass |
| TC-002 | Verify PO Approval Process for >$5k. | 1. Create a PO with Total Value = $6,000. <br>2. Submit for Approval. <br>3. | The PO status should be "Pending Approval". The approver should see an "Approve/Reject" button. | As expected. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | Log in as designated approver. | | | |
| TC-003 | Verify Expiry Report functionality. | 1. Create 3 items with expiry dates in 30, 60, and 120 days. <br>2. Run the "Items Expiring in Next 90 Days" report. | The report should show the first two items but not the third. | As expected. | Pass |
| TC-004 | Verify Role-Based Security for End User. | 1. Log in as a user with the "End User" profile. <br>2. Navigate to an Inventory Item record. | The user should see all fields but should not see an "Edit" or "Delete" button. | As expected. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| TC-005 | Verify Lot Number traceability. | 1. Create an item with Lot Number "LOT-ABC-123". <br>2. Use the global search bar to search for "LOT-ABC-123". | The search results should return the correct Inventory Item record. | As expected. | Pass |

# 7. RESULTS

## 7.1 Output Screenshots

The following screenshots demonstrate the final implemented solution.

Screenshot 1: The GarageCare Command Home Page Dashboard
[Image: A screenshot of the Salesforce Home Page, showing a dashboard with charts and KPIs. Key components to show: "Items Below Reorder Level," "Inventory Value by Category" (pie chart), and "Open Purchase Orders" list.]
Description: This dashboard provides an at-a-glance view of the most critical inventory metrics, enabling managers to make quick, informed decisions.

**Screenshot 2: The Inventory Item Record Page**
[Image: A screenshot of a Lightning Record Page for a single Inventory Item. It should clearly show fields like "Quantity on Hand," "Reorder Level," "Batch Number," and "Expiration Date." A related list for "Stock Locations" should be visible.]

**Description:** This is the central record for any medical supply, providing a 360-degree view of its status, location, and procurement details.

**Screenshot 3: Purchase Order Approval Process**
[Image: A screenshot of a Purchase Order record page, showing the "Approval History" related list. The status should be "Pending Approval," and an "Approve / Reject" button should be visible at the top.]
**Description:** This screenshot shows a PO that has been submitted for approval, demonstrating the digital workflow that replaces manual, paper-based processes.

**Screenshot 4: Low-Stock Alert Email**
[Image: A screenshot of an email received by the inventory manager. The subject line should be "Low Stock Alert: [Item Name]" and the body should contain details about the item and its current quantity.]
**Description:** An example of the automated alerts that form the core of the system's proactive inventory management capabilities.

**Screenshot 5: "Items Nearing Expiry" Report**
[Image: A screenshot of a Salesforce report. The report should be a table of inventory items, filtered to show only those expiring in the next 90 days, sorted by expiration date.]
**Description:** This critical report helps managers prioritize the use of older stock, significantly reducing waste from expired products.

# 8. ADVANTAGES & DISADVANTAGES

## 8.1 Advantages of the Salesforce Solution

- **Centralized Single Source of Truth:** All inventory data resides in one system, eliminating data silos and ensuring everyone works from the same real-time information.
- **Automation and Efficiency:** Automating alerts, reorders, and approvals drastically reduces manual workload, minimizes human error, and speeds up the entire procurement lifecycle.
- **Scalability and Flexibility:** The Salesforce platform can easily scale as the hospital or clinic grows. The system can be easily modified with new fields, objects, or automation rules as business needs evolve.
- **Superior Reporting and Analytics:** Salesforce's native reporting engine provides powerful, easy-to-create reports and dashboards, turning raw data into actionable business intelligence.
- **Enhanced Security and Compliance:** Leveraging Salesforce's robust security architecture, including role-based access and detailed audit trails, helps meet strict healthcare compliance standards like HIPAA.
- **Mobile Accessibility:** The Salesforce Mobile App allows staff to check stock levels, receive alerts, and even update inventory from anywhere in the facility, using their mobile devices.

## 8.2 Disadvantages and Mitigations

| Disadvantage / Challenge | Mitigation Strategy |
|---|---|
| **Licensing Costs:** Salesforce licenses represent a significant | A detailed ROI analysis was performed, demonstrating that cost savings from reduced waste and |

| | |
|---|---|
| ongoing operational expense. | improved efficiency outweigh the licensing fees. Using a mix of license types (e.g., full licenses for managers, cheaper Platform licenses for some users) can also optimize cost. |
| **Dependency on a Single Vendor:** The solution is entirely dependent on the Salesforce platform. | This is a strategic decision. The benefits of the integrated platform (security, reliability, innovation) are deemed to outweigh the risks of vendor lock-in. Salesforce's strong market position and uptime record provide confidence. |
| **User Adoption and Training:** Staff accustomed to manual processes may resist a new system. | A comprehensive training plan was developed. The UI was designed to be as intuitive as possible. "Super Users" or champions were identified in each department to encourage adoption and provide peer support. |
| **Complexity of Customization:** Over-customization can lead to technical debt and maintenance challenges. | We adhered to a "clicks-not-code" philosophy wherever possible, using standard declarative tools like Flow before resorting to Apex. All customizations were thoroughly documented. |

## 9. CONCLUSION

The Garage Management System project successfully meets its core objectives of transforming a traditionally chaotic and reactive process into a streamlined, proactive, and data-driven operation. By leveraging the Salesforce platform, we have created a robust solution that directly addresses the critical pain points of healthcare facilities: preventing stockouts, reducing waste, ensuring compliance, and improving overall operational efficiency.

The system's real-time dashboards provide unprecedented visibility into the entire supply chain, while its intelligent automation frees up valuable staff time from mundane administrative tasks. The successful implementation of features like batch/lot number tracking and automated expiry alerts not only leads to significant cost savings but also enhances patient safety—the ultimate goal of any healthcare initiative.

This project serves as a powerful testament to how a modern, cloud-based platform like Salesforce can be tailored to solve complex, industry-specific challenges, delivering tangible value and a strong return on investment.

## 10. FUTURE SCOPE

While the current implementation provides a comprehensive solution, there are several avenues for future enhancement that could deliver even greater value:

1. **AI-Powered Demand Forecasting:** Integrate Salesforce Einstein AI to analyze historical consumption data and predict future demand for inventory items. This would move the system from a reactive reorder-point model to a proactive, predictive one.
2. **Supplier Community Portal:** Build a portal using Salesforce Experience Cloud where suppliers can log in

to view their Purchase Orders, update shipment statuses, and communicate directly with the inventory team, further automating the procurement process.

3. **IoT Integration:** Place IoT-enabled weight sensors on shelves for high-volume items. These sensors could automatically trigger a stock update in Salesforce when the weight drops below a certain threshold, eliminating the need for any manual counting or scanning.
4. **Integration with Financial Systems:** Develop an API-based integration with the hospital's primary ERP or accounting software to automatically sync Purchase Order and invoicing data, streamlining the procure-to-pay cycle.
5. **Enhanced Mobile Functionality:** Develop a custom Lightning Web Component for the mobile app that allows for "cycle counting"—where staff can quickly scan a shelf or cart and the app will automatically reconcile the scanned items against the system's records.

## 11. APPENDIX

### 11.1 Appendix A: Sample Apex Code

Below is a sample Apex trigger that could be used to prevent a user from setting the quantity of an item to a negative number. While a validation rule could also do this, a trigger is shown for illustrative purposes.

Generated apex

```
      /**
 * @description Trigger on
Inventory_Location_Junction__c to prevent negative
inventory.
 */
trigger InventoryLocationJunctionTrigger on
Inventory_Location_Junction__c (before update) {
    for (Inventory_Location_Junction__c invJunction : 
Trigger.new) {

        // Check if the quantity is being updated to 
a negative value
        if (invJunction.Quantity_in_Location__c < 0) 
{

            // Add an error to the record, preventing 
the save
            invJunction.addError('Inventory quantity 
cannot be less than zero.');
        }
    }
}
```

### 11.2 Appendix B: Dataset Link

A sample CSV file containing 100 medical inventory items for data migration testing can be found at the following location:

[Link to Google Drive, Dropbox, or other repository containing a sample Inventory_Items.csv file]

### 11.3 Appendix C: GitHub & Project Demo Link

The metadata for this project, including all custom objects, flows, and layouts, is stored in a public GitHub repository for review.

- **GitHub Repository:** [Link to your GitHub repository]

A recorded video demonstration of the system's key features is available at the link below.

- **Project Demo Video:** [Link to a YouTube, Loom, or Vimeo video]

THANK YOU