MEDICAL INVENTORY MANAGEMENT

Performance Testing Report

TEAM ID	NM2025TMID07179
PROJECT NAME	MEDICAL INVENTORY MANAGEMENT
TEAM MEMBER NAMES	M. SUDHARSAN (912422104043), R. SUDHARSAN (912422104045), P. SIBIKUMAR (912422104039), PL. SURYA (902422104047).

1. Introduction

Performance Testing is a crucial phase in the software testing life cycle where the system's **speed, stability, scalability, and responsiveness** are evaluated under various conditions. For the **Medical Inventory Management System (MIMS)** built on **Salesforce**, performance testing ensures that the application can handle real-time medical data transactions, multiple user requests, and complex automation processes efficiently — especially in a healthcare environment where reliability and quick response are vital.

2. Objective of Performance Testing

The main objectives of performance testing in this project are:

- To verify that the Salesforce-based system performs efficiently under expected and peak workloads.
- To identify potential **bottlenecks**, **latency issues**, and **resource usage problems**.
- To ensure smooth user experience for all roles (Admin, Pharmacist, Supplier Manager).
- To validate that automated processes like stock updates, alerts, and report generation execute within acceptable time limits.

3. Importance in Medical Inventory Management

In a medical environment, **delay or lag** in accessing inventory data or processing stock information can lead to:

- Shortages in critical medical supplies.
- Inaccurate stock records.
- Poor decision-making for reordering.

Hence, performance testing ensures that the system delivers **reliable, fast, and consistent** responses to support real-time medical operations.

4. Types of Performance Testing Conducted

The following types of tests are typically performed on the Salesforce-based Medical Inventory Management System:

1. **Load Testing:**
    - Checks system performance under normal and peak loads (e.g., multiple pharmacists updating stock records simultaneously).
2. **Stress Testing:**
    - Determines how the system behaves under extreme load conditions beyond its expected capacity (e.g., during large data imports or simultaneous order processing).
3. **Scalability Testing:**
    - Evaluates the system's ability to scale when new users, records, or hospital branches are added.
4. **Endurance (Soak) Testing:**
    - Tests system stability over an extended period of continuous use.
5. **Spike Testing:**
    - Examines system behavior when the load suddenly increases (e.g., multiple restock alerts triggered at once).

5. Performance Metrics Measured

Key metrics observed during performance testing include:

- **Response Time:** The time taken by Salesforce to load records or dashboards.
- **Throughput:** Number of transactions processed per second.
- **CPU and Memory Usage:** Resource utilization by Salesforce processes.
- **Error Rate:** Frequency of transaction failures under load.
- **Latency:** Delay between request initiation and system response.
- **Concurrent User Handling:** Ability of the system to handle multiple users simultaneously.

6. Tools and Methods Used

Since Salesforce is a cloud-based platform, performance testing is carried out using both **Salesforce monitoring tools** and **third-party test platforms** such as:

- **Salesforce Lightning Usage App:** Monitors page load times and user performance trends.

- **Apex Test Execution:** Validates performance of triggers, classes, and batch jobs.
- **JMeter / LoadRunner (Optional):** Simulates concurrent user interactions for stress and load testing.
- **Developer Console & Debug Logs:** Used to measure query execution time and governor limit usage.

7. Test Scenarios

Some practical test scenarios include:

- Creating and updating multiple medicine records at once.
- Generating dashboards and reports with thousands of entries.
- Executing automatic stock update flows and triggers under heavy data volume.
- Uploading large supplier or purchase order lists.
- Sending multiple alert notifications for low stock or expired medicines simultaneously.

8. Expected Outcomes

After performance testing, the system should:

- Maintain fast response times (<2 seconds for standard transactions).
- Handle at least the expected number of concurrent users without lag.
- Execute automated triggers (e.g., expiry alerts, stock updates) within defined time limits.
- Show stable operation without performance degradation under prolonged use.
- Ensure no data loss or corruption during high-load conditions.

9. Result Analysis and Optimization

If performance bottlenecks are detected, corrective actions include:

- Optimizing Apex code (SOQL queries, trigger logic).
- Reducing unnecessary automation or batch processing frequency.
- Using Salesforce **indexing** and **data archiving** for large datasets.
- Streamlining **Flows** and **Validation Rules** to reduce execution time.
- Enhancing system configuration to improve record retrieval speed.

10. Conclusion

The **Performance Testing Phase** ensures that the **Medical Inventory Management System in Salesforce** meets high standards of efficiency, reliability, and speed required in healthcare environments.
By validating performance across various load conditions, this phase guarantees that the system can handle real-world medical inventory operations — ensuring uninterrupted service, accurate stock management, and quick access to vital medical information.