

# Garbage Management System Using Salesforce

## Phase 5 – Performance Testing

Performance Testing ensures that the Salesforce-based Garbage Management System operates efficiently under expected load conditions and meets the required speed, reliability, and scalability standards. This phase validates that the system can handle real-time operations, citizen complaints, IoT integrations, and field service tasks without delays or failures.

### 5.1 Objectives of Performance Testing

1. Assess system responsiveness under normal and peak loads.
2. Identify performance bottlenecks in Salesforce objects, automation, and integrations.
3. Ensure data accuracy and reliability during high usage.
4. Validate IoT sensor data handling and updates to bin objects in real-time.
5. Guarantee consistent user experience for citizens, field staff, and administrators.
6. Ensure scalability to support growth in users and locations.

### 5.2 Key Performance Metrics

| Metric           | Description   | Target                         |
|------------------|---|--------------------------------|
| Response Time    | Time taken to load pages, dashboards, and forms     | < 3 seconds for standard pages |
| Transaction Time | Time to create/update records, run automation flows | < 5 seconds per transaction    |
| Concurrent Users | Number of simultaneous users system can support     | ≥ 500 users                    |

|                   |   |                          |
|-------------------|---|--------------------------|
| System Throughput | Number of transactions per hour (complaints, bin updates) | ≥ 1000 transactions/hour |
| Data Accuracy     | Correct updates from IoT devices and field entries        | 99%+ accuracy            |
| Uptime            | Availability of Salesforce system                         | ≥ 99.5%                  |

## 5.3 Types of Performance Testing

### 1. Load Testing

- Simulate multiple users (citizens, field staff, administrators) performing daily operations.
- Validate that system handles expected loads without slowdowns.

### 2. Stress Testing

- Test the system under extreme conditions, e.g., surge in complaints or IoT data spikes.
- Determine breaking points and recovery capability.

### 3. Scalability Testing

- Verify that the system can scale to additional cities, bins, and users without degradation.

### 4. Endurance Testing

- Evaluate system performance over extended periods to detect memory leaks or slowdowns.

### 5. Integration Performance Testing

- Test performance of IoT sensor data feeds, APIs, and mobile portal updates.
- Ensure real-time updates and notifications are delivered efficiently.

## 5.4 Test Environment Setup

### 1. Salesforce Sandbox:

- Use a full-copy or partial-copy sandbox for realistic testing.

### 2. Test Data:

- Populate Garbage Bin, Collection Schedule, Citizen Complaints, and Recycling Record objects with simulated data.
- Include thousands of records to simulate city-scale operations.

### 3. Simulated Users:

- Citizens submitting complaints via portal/mobile app.
- Field staff updating collection status.
- Admin users generating reports and dashboards.

### 4. IoT Integration:

- Simulate real-time bin sensor data updates at varying intervals.

## 5.5 Test Cases and Scenarios

| Test Case             | Scenario  | Expected Result                                     |
|-----------------------|---|---|
| TC1 – Page Load       | Citizen logs into portal and views complaint form | Page loads within 3 seconds                         |
| TC2 – Record Creation | Field staff creates Collection Schedule           | Record saved, notifications sent, <5 seconds        |
| TC3 – Automation Flow | Bin fill level exceeds threshold                  | Task auto-assigned to nearest truck, staff notified |
| TC4 – Bulk Updates    | 1000 IoT bin updates in 1 hour                    | All updates processed accurately without errors     |

|                         |  |  |
|-------------------------|--|--|
| TC5 – Dashboard Refresh | Admin views operational dashboard          | Data refreshes within 5 seconds                          |
| TC6 – Stress Test       | 600 simultaneous users submit complaints   | System handles load gracefully or queues excess users    |
| TC7 – Integration       | IoT sends real-time data for multiple bins | Garbage Bin object updates correctly, notifications sent |

## 5.6 Performance Testing Tools

### Salesforce Performance Tools:

- Salesforce Lightning Inspector, Debug Logs, and Governor Limits monitoring.

### External Tools:

- JMeter or LoadRunner for simulating concurrent users.
- Postman for API performance testing.

### Monitoring Dashboards:

- Use Salesforce dashboards to track real-time transactions, error rates, and response times.

## 5.7 Performance Testing Process

1. Plan Testing: Define objectives, metrics, scenarios, and tools.
2. Prepare Test Data: Populate objects with realistic records and simulate IoT data.
3. Execute Tests: Run load, stress, and endurance scenarios.
4. Monitor Performance: Track response times, throughput, and errors.
5. Analyze Results: Identify bottlenecks, automation delays, or integration lags.

6. Optimize System: Adjust flows, batch sizes, API calls, or indexes as needed.
7. Retest: Validate improvements and ensure performance meets targets.

## **5.8 Success Criteria**

1. System supports 500+ concurrent users without slowdown.
2. Page load and transaction times meet targets (<5 seconds).
3. IoT sensor updates are processed accurately in real-time.
4. Dashboards refresh within acceptable time limits (<5 seconds).
5. Automation flows execute reliably without errors or delays.
6. System demonstrates scalability and resilience under stress conditions.

## **5.9 Conclusion**

- The Performance Testing Phase ensures that the Garbage Management System built on Salesforce is fast, reliable, and scalable, even under heavy load conditions. By testing page performance, automation flows, IoT integrations, and dashboards, the system guarantees smooth operations for citizens, field staff, and administrators.
- After successful performance testing, the project can move to User Acceptance Testing (UAT) and Deployment, ensuring the system is ready for production use.