

PERFORMANCE TESTING DOCUMENTATION

Metro Ticket Generating System using ServiceNow

8. PERFORMANCE TESTING

Performance Testing is a critical phase in the testing lifecycle of the **Metro Ticket Generating System using ServiceNow**. This phase evaluates how efficiently the system performs under various conditions such as normal load, peak load, and stress conditions. The objective is to ensure that the system is responsive, stable, reliable, and scalable when multiple users access the ticket generation service simultaneously.

In a metro environment, ticket generation is a high-demand activity, especially during peak travel hours. Hence, performance testing ensures that the ServiceNow-based solution can handle real-world usage without degradation in service quality.

8.1 Objectives of Performance Testing

The main objectives of performance testing for the Metro Ticket Generating System are:

- To verify system response time during ticket booking
 - To evaluate system behavior under normal and peak load conditions
 - To ensure system stability during continuous usage
 - To identify performance bottlenecks
 - To validate scalability of the ServiceNow application
 - To ensure optimal utilization of system resources
-

8.2 Scope of Performance Testing

In-Scope Components

- Service Catalog ticket booking form
- Custom tables for ticket and route data
- Flow Designer automation
- Notification services
- Reports and dashboards

Out-of-Scope Components

- External payment gateways
- Hardware performance (printers, kiosks)

- Network-level performance issues
-

8.3 Performance Testing Types

8.3.1 Load Testing

Load testing evaluates system performance under expected user load.

- Simulates multiple users booking tickets simultaneously
- Validates acceptable response time
- Ensures workflows execute correctly under load

Example Scenario: - 100 users submit ticket requests within a short time interval

8.3.2 Stress Testing

Stress testing evaluates system behavior beyond normal load limits.

- Identifies breaking point of the system
 - Observes system recovery after overload
 - Ensures no data loss during high stress
-

8.3.3 Spike Testing

Spike testing checks system response to sudden increases in load.

- Sudden surge in ticket booking during peak hours
 - Evaluates workflow and database response
-

8.3.4 Endurance (Soak) Testing

Endurance testing measures system stability over extended periods.

- Continuous ticket booking over long duration
 - Detects memory leaks or performance degradation
-

8.4 Performance Metrics

The following metrics are used to evaluate performance:

Metric	Description
Response Time	Time taken to generate ticket
Throughput	Number of tickets generated per unit time
Error Rate	Failed ticket requests
CPU Utilization	Resource usage
Database Response	Query execution time

8.5 Test Environment

The performance testing environment is configured as follows:

- Platform: ServiceNow Personal Developer Instance (PDI)
- Browser: Google Chrome
- Network: Stable internet connection
- Test Data: Sample metro stations and routes
- Users: Simulated user accounts

The environment closely resembles the production-like setup for accurate results.

8.6 Test Scenarios

Scenario 1: Single User Ticket Booking

- User submits a ticket request
- System calculates fare
- Ticket generated successfully

Scenario 2: Multiple Concurrent Users

- Multiple users submit requests simultaneously
- System processes all requests without delay

Scenario 3: Peak Load Scenario

- High number of requests within a short duration
 - System maintains acceptable response time
-

8.7 Test Execution Strategy

- Identify critical workflows

- Prepare test data
 - Simulate load using ServiceNow testing tools
 - Monitor system logs and performance dashboards
 - Capture response times and error rates
-

8.8 Performance Test Results (Sample)

Test Type	Users	Avg Response Time	Status
Load Test	50	Acceptable	Pass
Stress Test	150	Slower	Pass
Spike Test	Sudden 200	Stable	Pass

8.9 Bottleneck Analysis

Potential bottlenecks identified:

- Complex Flow Designer logic
- Inefficient database queries
- Excessive notification triggers

Optimization actions: - Simplified workflow logic - Indexed frequently used fields - Reduced unnecessary notifications

8.10 Risk and Mitigation

Risk	Impact	Mitigation
High traffic during peak hours	Performance drop	Workflow optimization
Database overload	Data delay	Indexing & caching
System timeout	Request failure	Load balancing

8.11 Acceptance Criteria

The system is considered performance-ready if:

- Ticket generation time is within acceptable limits
- No critical errors under peak load

- System recovers gracefully from stress conditions
 - Data integrity is maintained
-

Conclusion of Performance Testing

Performance testing confirms that the **Metro Ticket Generating System using ServiceNow** meets expected performance standards under various load conditions. The system demonstrates stability, scalability, and responsiveness suitable for metro ticketing operations.

Successful completion of this phase ensures readiness for deployment and provides confidence in system reliability during real-time usage.
