*Performance Test Execution Sheet*

*[Blazedemo](https://www.blazedemo.com/)*

# Performance Testing Program Plan

# Date: August 12th - 2025

*Created by* [*Hashem Al-Hazzaa*](https://www.linkedin.com/in/hashem-al-hazzaa-032183183/)

## Note: All ramp-up, ramp-down, hold-time, and initial delay values shown here have been reduced from their original timings to save execution time. This document is intended to resemble a business/stakeholder analysis report, but in the actual test script, the values are much lower and are used solely for application testing purposes.

## **1.0 Introduction**

### 1.1 Purpose

The purpose of this Performance Testing Program Plan is to identify the process and phased approach that will be implemented at Site Blazedemo. The purpose of the testing program at Site Blazedemo is specifically designed to evaluate the effectiveness of systems that are employed at this site. This plan defines tasks to be accomplished to ensure that performance testing is conducted as effectively and efficiently as possible.

## **2.0 Requirements**

### 2.1 System Elements

* The test is conducted using Apache Jmeter version 5.6.3 only.
* Test is done by creating a test plan that has thread groups each for a specific intended test.
* Each Thread Group will have its assertions, controllers, samplers…etc. Independently
* Test results will be compared to the expected test results, and identify any bottlenecks, bugs, anomaly behaviors and overall performance metrics.

### 2.2 Testing Concept

Performance testing is a test to evaluate the ability of an implemented and operating system element or total system to meet an established requirement. Individual performance tests for response are used to determine whether guard and response procedures are effective, whether personnel understand and follow the procedures, and whether personnel and equipment interact effectively. Performance Test Exercises are means to realistically evaluate the effectiveness of response force programs; provide application skills training for personnel; identify areas requiring system improvements; validate implemented improvements and motivate personnel to perform duties in the most efficient, effective, and safest manner. To effectively coordinate an exercise or test, the process outlined below is used to properly plan and conduct these types of exercises or tests.

## **3.0 Performance Testing**

### 3.1 Types of tests that will be conducted

The following performance test types will be executed on the Blazedemo site. Each test type will include a **Workload Model** defining user distribution, transaction mix, and test data requirements.

#### 3.1.1 Load Testing

* **Purpose**: To determine the system’s performance under expected peak load conditions.
* **Focus**: Measures response times, throughput, and resource usage at the predicted user volume.
* **Workload Modeling**: Based on average concurrent users during high-traffic periods and typical transaction patterns (e.g., browsing products, adding to cart, checkout).

#### 3.1.2 Stress Testing

* **Purpose**: To identify the system’s breaking point and behavior under extreme load.
* **Focus**: Gradually increase the load beyond expected capacity until performance degrades or failures occur.
* **Workload Modeling**: Starts from baseline load and increases in increments until system failure.

#### 3.1.3 Endurance/Soak Testing

* **Purpose**: To evaluate system stability and performance over an extended duration under expected load.
* **Focus**: Detects memory leaks, resource exhaustion, and performance degradation over time.
* **Workload Modeling**: Simulates normal usage levels continuously for several hours or days.

#### 3.1.4 Spike Testing

* **Purpose**: To observe system behavior when the load increases or decreases abruptly.
* **Focus**: Measures recovery time and stability after sudden traffic surges.
* **Workload Modeling**: Sharp user volume spikes followed by rapid drop to normal load.

### 

### 3.2 Workload Modeling & Test Design

#### 3.2.1 Load Test

* User Load Profile:
* Expected user load: 1,500
* Ramp-Up time: 30 minutes
* Duration: 1 hour
* User Behavior Scenarios
* Select Destination: 25% of users → 375 users
* Choose Flight: 50% of users → 750 users
* Input data and Purchase: 25% of users → 375 users
* Performance Metrics:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Run ID | User Action | Avg Response Time (s) | Avg Throughput (req/s) | Avg Error Rate (%) | Avg Resource Util. (%) |
| Run 1 | Select Destination | 0.9 | 45 | 0.5 | 55 |
| Choose Flight | 1.4 | 70 | 0.7 | 63 |
| Input data and Purchase | 2.1 | 35 | 1.0 | 68 |
| Run 2 | Select Destination | 1.0 | 48 | 0.6 | 57 |
| Choose Flight | 1.6 | 72 | 0.8 | 65 |
| Input data and Purchase | 2.3 | 36 | 1.2 | 70 |

#### 

#### 3.2.2 Stress Test

Note: Each phase will conclude the 4 steps of user behavior.

* Phase 1:
  + Number of users: **100**
  + Ramp up: 60 seconds
  + Ramp down: 60 seconds
  + Hold Time/Steady state: 5 mins
  + User Behavior Scenarios
  + Select Destination: 35% of users → 35 users
  + Choose Flight: 50% of users → 50 users
  + Input data and Purchase: 15% of users → 15 users
* Phase 2:
  + Number of users: **600**
  + Ramp up: 60 seconds
  + Ramp down: 60 seconds
  + Hold Time/Steady state: 5 mins
  + User Behavior Scenarios
  + Select Destination: 60% of users → 360 users
  + Choose Flight: 20% of users → 120 users
  + Input data and Purchase: 20% of users → 120 users
* Phase 3:
  + Number of users: **1700**
  + Ramp up: 60 seconds
  + Ramp down: 60 seconds
  + Hold Time/Steady state: 5 mins
  + User Behavior Scenarios
  + Select Destination: 25% of users → 375 users
  + Choose Flight: 50% of users → 750 users
  + Input data and Purchase: 25% of users → 375 users

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Phase | Threads | Ramp-Up (sec) | Hold (sec) | Ramp-Down (sec) | User Action | Resp. Time (s) | Throughput (req/s) | Error Rate (%) | Resource Util. (%) |
| Phase1 | 100 | 60 | 300 | 60 | Select Destination | 0.8 | 15 | 0.2 | 40 |
| Choose Flight | 1.2 | 25 | 0.4 | 45 |
| Input data and Purchase | 2.0 | 10 | 0.5 | 50 |
| Phase2 | 600 | 60 | 300 | 60 | Select Destination | 1.5 | 60 | 1.5 | 70 |
| Choose Flight | 2.4 | 100 | 2.0 | 75 |
| Input data and Purchase | 3.0 | 50 | 2.5 | 80 |
| Phase3 | 1700 | 60 | 300 | 60 | Select Destination | 3.5 | 120 | 5.0 | 85 |
| Choose Flight | 5.0 | 200 | 7.0 | 90 |
| Input data and Purchase | 6.2 | 90 | 10.0 | 95 |

#### 3.2.3 Spike Test

* Stage 1: Initial Load
* Number of Threads: 100
* Ramp-Up Time: 60 seconds
* Hold Load For: 10 minutes
* Shutdown Time: 60 seconds
* User Behavior Scenarios
  + Select Destination: 30% of users → 30 users
  + Choose Flight: 40% of users → 40 users
  + Input data and Purchase: 30% of users → 30 users
* Stage 2: **Spike Load (ex: starts 2 mins after initial load is steady & spike runs for 1 mins)**
* Number of Threads: 300
* Ramp-Up Time: 10 seconds
* Hold Load For: 2 minutes
* Shutdown Time: 10 seconds
* User Behavior Scenarios
  + Select Destination: 25% of users → 375 users
  + Choose Flight: 50% of users → 750 users
  + Input data and Purchase: 25% of users → 375 users
* Stage 3: **(Starts 5 mins after initial load is steady & spike runs for 1 mins)**
* Number of Threads: 500
* Ramp-Up Time: 10 seconds
* Hold Load For: 2 minutes
* Shutdown Time: 10 seconds
* User Behavior Scenarios
  + Select Destination: 25% of users → 375 users
  + Choose Flight: 50% of users → 750 users
  + Input data and Purchase: 25% of users → 375 users

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stage | Threads | Ramp-Up (sec) | Hold (sec) | Ramp-Down (sec) | User Action | Resp. Time (s) | Throughput (req/s) | Error Rate (%) | Resource Util. (%) |
| Stage1 | 100 | 60 | 600 | 60 | Select Destination | 0.9 | 18 | 0.3 | 45 |
| Choose Flight | 1.3 | 24 | 0.5 | 50 |
| Input data and Purchase | 2.2 | 15 | 0.6 | 55 |
| Stage2 | 300 | 10 | 120 | 10 | Select Destination | 1.8 | 40 | 1.0 | 65 |
| Choose Flight | 2.5 | 60 | 1.5 | 70 |
| Input data and Purchase | 3.2 | 30 | 2.0 | 75 |
| Stage3 | 500 | 10 | 120 | 10 | Select Destination | 2.8 | 70 | 3.0 | 80 |
| Choose Flight | 4.0 | 100 | 4.5 | 88 |
| Input data and Purchase | 5.5 | 45 | 6.0 | 92 |

#### 3.2.4 Endurance/Soak Test

* User Behavior Scenarios (Expecting 1200 Concurrent Users)
* Select Destination: 15% of users → 180 users
* Choose Flight: 45% of users → 540 users
* Input data and Purchase: 40% of users → 480 users

This test focuses on identifying **memory leaks**, **performance degradation**, and any other issues that arise from the long run of the application. However, due to the lack of proper tools to monitor these metrics, I will only run the test with the specified user behavior.