./

Demoblaze Application Performance Test Plan



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **To be Reviewed By** | **Remarks/Revision Details** |
| 1. | 18-12-2020 | Milind Mohapatra,  Y Sushank | Veeresh Tonne |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Details**

# Contents

[Contents 3](#_Toc59543638)

[Table of Figures 4](#_Toc59543639)

[Table of Tables 4](#_Toc59543640)

[Objectives of the test 5](#_Toc59543641)

[Application description 5](#_Toc59543642)

[Demoblaze performance testing architecture 5](#_Toc59543643)

[Software details 6](#_Toc59543644)

[Critical business transactions (CBT’s) 6](#_Toc59543645)

[Scope of the test 6](#_Toc59543646)

[Items out of scope 7](#_Toc59543647)

[Methodology 8](#_Toc59543648)

[Performance testing Approach/Procedure 9](#_Toc59543649)

[Entry and Exit criteria (Assumption) 10](#_Toc59543650)

[Suspension Criteria 10](#_Toc59543651)

[Environment 10](#_Toc59543652)

[Tools used in the project 11](#_Toc59543653)

[Types of testing 11](#_Toc59543654)

[Test data 12](#_Toc59543655)

[Monitoring 12](#_Toc59543656)

# Table of Figures

[Figure 1 Demoblaze performance testing architecture 5](#_Toc59543635)

[Figure 2 Performance test methodology 8](#_Toc59543636)

[Figure 3 Performance test approach 9](#_Toc59543637)

# Table of Tables

[Table 1 Software details 6](#_Toc59543625)

[Table 2 Items out of scope 7](#_Toc59543626)

[Table 3 Entry and exit criteria 10](#_Toc59543627)

[Table 4 Tools used for this project 11](#_Toc59543628)

[Table 5 Types of testing 11](#_Toc59543629)

[Table 6 Client side monitoring metrics 12](#_Toc59543630)

[Table 7 Server side monitoring metrics 12](#_Toc59543631)

[Table 8 Time schedules 13](#_Toc59543632)

[Table 9 Risks and mitigations 13](#_Toc59543633)

[Table 10 Performance test deliverables 14](#_Toc59543634)

# Objectives of the test

The purpose of this document is to outline the environment and performance test plan for benchmarking **Demoblaze Application** and tovalidate the core **Demoblaze Application** such that certain transactions meet the minimum performance standards established for this project. The transactions and flows that are going to be performance tested will be from the below workstream.

* Demoblaze Web Application

The document is intended for the below audiences:

* The business owners to confirm the functional scope selected for performance testing exercise.
* LTTS performance testing interns to prepare scripts and execute the performance testing exercise as outlined in this document

# Application description

The goal of the application is to make an e-Commerce application for electronic products platform with intuitive user interfaces with best user experience. The potential users of the system are the general public.

# Demoblaze performance testing architecture

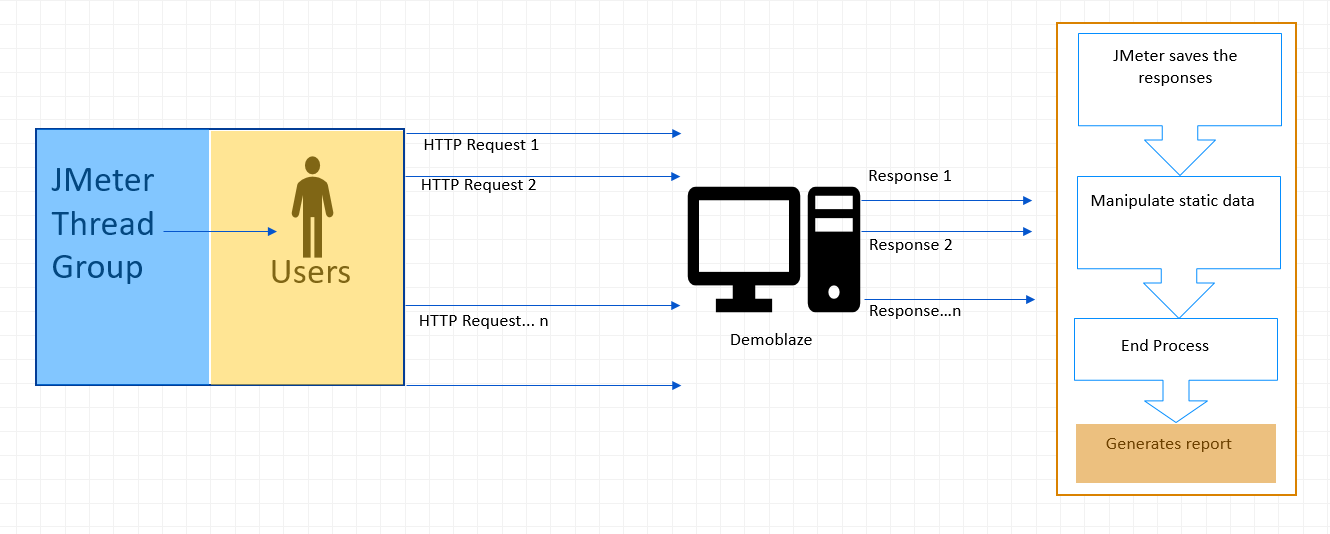


Figure Demoblaze performance testing architecture

# Software details

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Component** | **Performance Environment** | **Production Environment** |
| 1 | Operating System | Win 10 | Unknown |
| 2 | Web Server | Unknown | Unknown |
| 3 | Browsers | Chrome, Mozilla Firefox, Edge | Unknown |
| 4 | Performance testing tool | JMeter | Unknown |
| 5 | Performance monitoring tools | JProfiler | Unknown |
| 6 | Protocols | HTTPs | Unknown |

Table Software details

# Critical business transactions (CBT’s)

* Add cart
* Browse items by category
* Make payment

# Scope of the test

This following are considered as in scope for performance testing,

* The different modules that are being considered for performance testing are in scope.
* Baseline, Load testing, Stress testing, Endurance testing.
* Identification of test objectives, test requirement and test goals.
* Development of test designs, test scripts to meet the test requirement and goals.
* Execution of Performance tests and report test results for further analysis.
* Generate Test Report for the Test and publish the Test Report.

The following are the of Business Scenarios / Transactions / Events that are agreed to be included in scope for performance testing.



# Items out of scope

The following are out of scope for performance testing.

* Functional testing.
* Changes in business transactions or any new business transaction.
* Changes in user load.
* Data setup of the application.
* Monitoring of performance of other component apart from agreed parameters.
* Scalability or Sizing of hardware

This section details the lists the Business Scenarios / Transactions / Events that will not be covered under the scope of performance testing.

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Scenario/Flow/Module** | **Out of scope -Reason** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

Table Items out of scope

# Methodology

LTTS’s performance testing methodology which is depicted in the below diagram will be used for performance testing. This will be customized based on the needs.

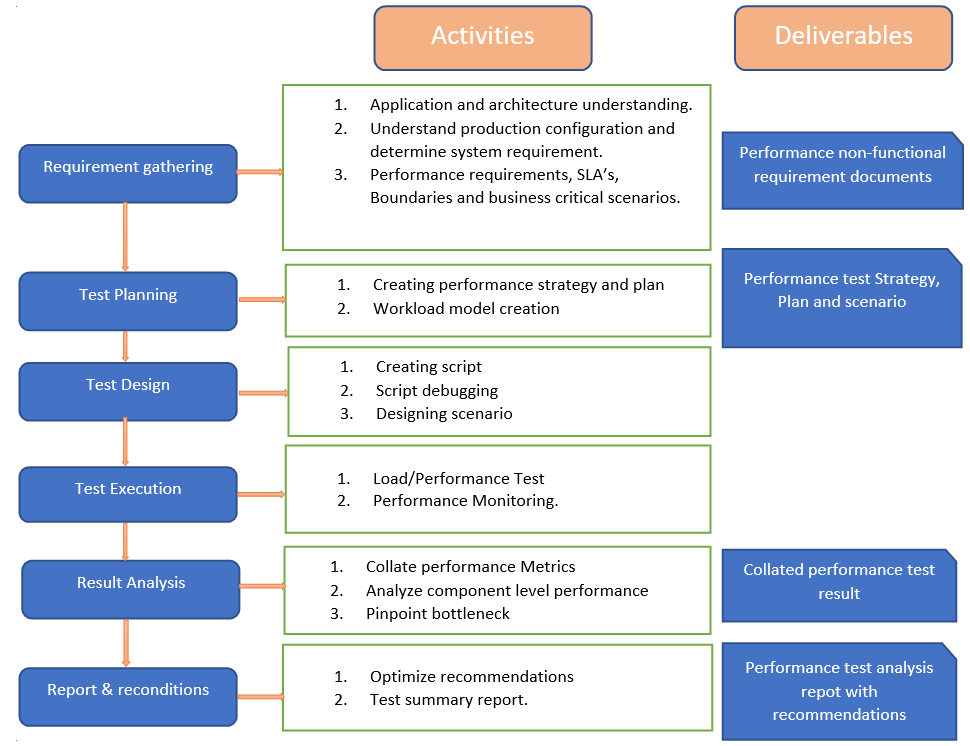
****

Figure Performance test methodology

# Performance testing Approach/Procedure

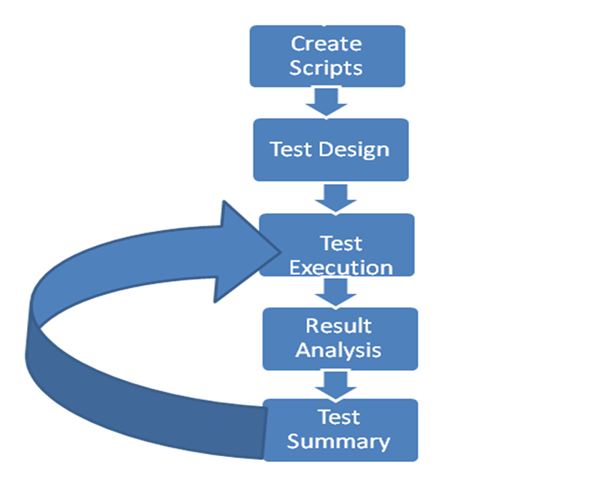


Figure Performance test approach

* Dry run with Single user, then with multiple user (say 5) users to validate the script.
* Run with 5 users for 5 minutes to warm up the load balancers and servers.
* Test Execution & Monitoring – Execute the configured load test and monitor & collect the required metrics/result for analysis.
* Result Analysis – Analyze the test result and identify the bottlenecks if any (Taken care by Dev team)
* Re-Execution – Re execution for same load scenario is required for validating bottleneck fixed or getting two consistent result
* Report preparation and reporting

# Entry and Exit criteria (Assumption)

|  |  |  |
| --- | --- | --- |
| **Activity** | **Enter Criteria** | **Exit Criteria** |
| Test Planning | * Well defined objective * Knowledge transfer on the application * Completed project plan / schedule | * Completed Test plan * Test data requirement documented * Test plan walk-through * Approved Test plan |
| Test Preparation | * Functionally stable application * Stable test environment | * Completed Test Scripts * Dry Run of scripts |
| Test Execution | * Completed Test Environment setup * Access to Load balancer, Web server, application server and DB server should be available to Performance team/Dev Team for performance monitoring. * Completed Scripts * Test data for execution of the scripts * QA Certified Build deployed for performance testing. | * All scripts executed based on the scope of testing * Execute the scripts to verify the performance SLA is met. * Test results and analysis for each round of execution. |
| Test Closure | * Test results for each round of execution | * Completed Test Report shared with stakeholders |

Table Entry and exit criteria

# Suspension Criteria

This section details the various Test Suspension criteria such as Showstoppers, % of Severity defects, % of Transactions failed etc. against which the test shall be suspended temporarily and will resume once the defects/ issues are fixed.

If the application does not meet the performance SLA defined, then it will be considered as performance defect.

The following issues would be treated as show stopper issues

* Any functional failure in load test
* Application hanging
* Database exceptions etc.,
* Out-of-memory related issues

# Environment

Test execution will happen in the performance environment which is same as production environment.

[ Performance environment Architecture]

**TBD**

# Tools used in the project

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Tool** | **Version** | **Purpose** |
| 1 | JMeter | 5.3 | Load Testing |
| 2 | JProfiler | 12 | Monitoring |

Table Tools used for this project

# Types of testing

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Test** | **Description** |
| 1 | **Performance Baseline Test** | This test is carried out as a first stage to find out the optimal performance/response time of the AUT (Application Under Test). The objective of this test is to determine the response times and system resource utilization (CPU, Memory, IO, Network) for single user. |
| 2 | **Load Test** | This test is conducted with gradual increase in user load and identifies the optimum allowed user load, during when the application response time, throughput and system resource utilization are under optimum or desired SLA. The objective is to determine the optimum number of users allowed on the system, during when the application is under optimum resource utilization. To measure response times, throughput rates, and resource-utilization levels at the optimum allowed user’s load.  This test can also be carried out with pre-determined user load to measure and validate application response time, throughput and system resource utilization against desired SLA |
| 3 | **Stress Test** | This test would be conducted to assess the stability of the application under high load:  Load Test ++  With/without Think time  > 90% CPU Usage on App Server  The main purpose of the test is to determine load related issues during stress. |
| 4 | **Endurance Test** | This test is conducted to know the availability of the application under continuous load. The load against application is kept constant for long hours ex: 8 to 10 hours |

Table Types of testing

# Test data

Data set will be created in the .CSV file format for all the required fields that need to be entered in each of the scenarios. This will act as data input for the created performance test scripts.

# Monitoring

The following metrics would be collected as part of the test execution and analyzed for identifying resource level bottlenecks.

* **Client side metrics**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Metrics** | **Description** |
| 1 | Concurrent Users | Number of concurrent users supported by the application on the given infrastructure. |
| 2 | Average response time | Average time taken to service the request/transaction reported in milliseconds/seconds |
| 3 | Minimum Response time | The minimum time taken among all the requests of a transaction reported in milliseconds/seconds |
| 4 | Maximum response time | The maximum time taken among all the requests of a transaction reported in milliseconds/seconds |
| 5 | % of Errors | Percent of requests with errors during the test run |
| 6 | Throughput | It is measured in requests/transactions served by the system per second/minute/hour. |

Table Client side monitoring metrics

* **Server side metrics**

The resource utilization in load balancer, webserver, application server and database servers will be monitored during the test runs as specified below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Metrics** | **Web Server** | **App Server** | **DB server** |
| 1 | CPU |  |  |  |
| 2 | Memory |  |  |  |
| 3 | Disk I/O |  |  |  |
| 4 | Network |  |  |  |

Table Server side monitoring metrics

* **Time schedules**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Activity** | **Start Date** | **End Date** | **Responsibility** | **Status** |
| 1 | Nonfunctional requirements elicitation and analysis | 19-11-2020 | 20-11-2020 | Milind Mohapatra and Y Sushank | Completed |
| 2 | Test planning | 21-11-2020 | 22-11-2020 | Milind Mohapatra and Y Sushank | Completed |
| 3 | Scripting | 23-11-2020 | 23-11-2020 | Milind Mohapatra and Y Sushank | Completed |
| 4 | Test Data Setup | 24-11-2020 | 24-11-2020 | Milind Mohapatra and Y Sushank | Completed |
| 5 | Script validation & Dry Run & Environment Validation | 24-11-2020 | 30-11-2020 | Milind Mohapatra and Y Sushank | Completed |
| 6 | One Round of Test Execution for All User Load Models: Pre-Execution Setup Execution & Monitoring Result Collection Post Test Analysis | 01-12-2020 | 14-11-2020 | Milind Mohapatra and Y Sushank | Completed |
| 7 | Report Preparation | 15-12-2020 | 18-12-2020 | Milind Mohapatra and Y Sushank | Completed |

Table Time schedules

* **Risks and Mitigations (Assumptions)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Risk/Issue** | **Priority** | **Mitigation** |
| 1 | Scope creep (change in requirement after the commencement of work) | High | TBD |
| 2 | Non-Availability of systems and environments during the testing | High | Timelines in the Test Window are to be communicated to the team. |
| 3 | Unavailability of Test data might impact the execution of the Test data dependent scenarios | High | Testing activity will be continued with the available test data |
| 4 | Unavailability of the application will impact the execution schedules. | High | The performance testing is carried out for the high priority scenarios. |
| 5 | Server not available | High | TBD |

Table Risks and mitigations

* **Assumptions and Dependencies**
* **Data creation**
  + The data volume required for load testing will be provided on a clean production sized database before the performance test execution activity.
  + The test data that needs to be input in the test scripts for unique fields will be provided by the functional test team before the test execution.
* **Environment**
* Performance test execution will be conducted on a production-size environment that will be made available exclusively to performance test team for the duration of performance tests.
* **Performance script Automation**
* Script automation will be done on a functionally stable application and environment.
* Any changes to the application or the environment, the scripts will be redone or modified according to the level of changes.
* **Time Schedule**
* Time schedule is based on environment availability and tests on a stable system solution. If tested product quality has issues impacting the robustness or severe functionality it needs to be solved before test execution. The schedule will be extended according to rerun of appropriate test cases.
* **Performance test deliverables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Test Plan** | **Test Script** | **Interim Report** | **Final Report** |
| **Create** | Milind Mohapatra and Y Sushank | Milind Mohapatra and Y Sushank | Milind Mohapatra and Y Sushank | Milind Mohapatra and Y Sushank |
| **Review** | Veeresh Tonne | Veeresh Tonne | Veeresh Tonne | Veeresh Tonne |
| **Sign Off** | XXX | XXX | XXX | XXX |

Table Performance test deliverables

* **Reporting**

Performance Test Interim Report will be shared at the end of each test and Performance Final Report will be shared consolidating the performance tests conducted throughout the performance testing cycle