| <Company Logo> | <Company Name>  <Company slogan> |
| --- | --- |

<JPet Store>

Performance Test Plan

Version 1.0

10/05/2024

**Document Number:** <PT1234 >

**Contract Number:** <CN1234>

Performance Test Plan Sign-off

<List out the name of stakeholders responsible to sign-off the document>

Table 1: Sign-off Detail

| Name | Role / Designation | Signoff Date | Signature |
| --- | --- | --- | --- |
| Name | Project Manager |  |  |
| Name | Business Analyst |  |  |
| Name | Application Architect |  |  |
| Name | Lead Developer |  |  |
| Name | Test Data Manager |  |  |
| Name | Performance Test Manager |  |  |
| Name | Performance Test Environment Manager |  |  |

Record of Changes

< Provide information on how the development and distribution of the performance test plan were carried out and tracked with dates. Use the table below to provide the version number, the date of the version, the author/owner of the version, and a brief description of the reason for creating the revised version.>

Table 2: Record of Changes

| Version  Number | Date | Author/Owner | Description of Change |
| --- | --- | --- | --- |
| Draft | 01/01/2024 | vishwa | Draft version |
| 0.1 | 15/01/2024 | vishwa | Added xxx details |
| 0.2 | 30/01/2024 | vishwa | Added Environment details |
| xx | xx/xx/xxxx | xxxxxx | xxxx xxxx xxxx xxx |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Performance Test Plan Sign-off ii](#_Toc4420471)

[Record of Changes iii](#_Toc4420472)

[Table of Contents iv](#_Toc4420473)

[1. Executive Summary 1](#_Toc4420474)

[1.1 Overview: Project Background and Scope 1](#_Toc4420475)

[2. Application Architecture 2](#_Toc4420476)

[2.1 Overview: System Architecture 2](#_Toc4420477)

[2.2 Architecture Diagram 2](#_Toc4420478)

[2.3 Detailed information on each component 2](#_Toc4420479)

[3. Performance Test Requirements 3](#_Toc4420480)

[3.1 Requirements 3](#_Toc4420481)

[3.1.1 Business NFR 3](#_Toc4420482)

[3.2 NFR and NFT Matrix 3](#_Toc4420484)

[4. Performance Test Planning 5](#_Toc4420485)

[4.1 Performance Test Approach 5](#_Toc4420486)

[4.1.1 Performance Testing and Monitoring Tool Details 5](#_Toc4420487)

[4.1.2 Performance Test Script Steps 6](#_Toc4420488)

[4.1.3 Performance Test Data Planning 7](#_Toc4420489)

[5. Performance Test Execution 8](#_Toc4420490)

[5.1 Performance Test Summary 8](#_Toc4420491)

[5.2 Performance Test Details 8](#_Toc4420492)

[5.2.1 Smoke Test 8](#_Toc4420493)

[5.2.2 Load Test 8](#_Toc4420494)

[5.2.3 Stress Test 9](#_Toc4420495)

[5.2.4 Soak Test 10](#_Toc4420496)

[5.3 Performance Test Monitoring Metrics 11](#_Toc4420497)

[5.4 Performance Test Environment 12](#_Toc4420498)

[5.5 Assumptions, Constraints, Risks and Dependencies 14](#_Toc4420499)

[5.5.1 Assumptions 14](#_Toc4420500)

[5.5.2 Constraints 14](#_Toc4420501)

[5.5.3 Risks 15](#_Toc4420502)

[5.5.4 Dependencies 15](#_Toc4420503)

[6. Milestones 17](#_Toc4420504)

[6.1.1 Test Organization 17](#_Toc4420505)

[Appendix A: Acronyms 19](#_Toc4420506)

[Appendix B: Glossary 20](#_Toc4420507)

[Appendix C: Referenced Documents 21](#_Toc4420508)

# Executive Summary

<The purpose of conducting performance testing on the JPet Store e-commerce website is to evaluate its ability to handle expected user loads and transaction volumes while maintaining acceptable response times and system stability. By simulating realistic user scenarios and stress conditions, we aim to identify performance bottlenecks, optimize system resources, and ensure a seamless user experience during peak traffic periods >

## Overview: Project Background, Objective and Scope

**Project Background:**

<

"This e-commerce application aims to provide a seamless shopping experience for users across multiple devices. It focuses on offering a wide range of products, secure payment gateways, personalized recommendations, and efficient order processing to enhance customer satisfaction and drive sales."

>

**Objectives:**

1. Evaluate Response Time: Measure and analyze the response times of critical pages such as product listing, product details, cart operations, and checkout processes to ensure optimal user experience.

2. Assess Scalability: Determine the system's scalability by gradually increasing the user load to identify performance bottlenecks, server capacity limits, and resource utilization under varying user loads.

3. Validate Concurrent User Handling: Verify the system's ability to handle concurrent users by simulating realistic user scenarios, including browsing, searching, adding items to cart, and completing transactions, while maintaining acceptable response times and error rates.

4. Stress Testing: Push the system beyond its normal operational limits to identify breaking points, server failures, or performance degradation under extreme load conditions, ensuring system stability during peak traffic periods.

**Scope:**

**In Scope:**

1. Response Time: Measure and analyze the response times of critical pages such as homepage, product listing, product details, cart operations, and checkout processes.
2. Concurrency Testing: Validate the system's ability to handle concurrent users performing typical e-commerce activities simultaneously.
3. Transaction Throughput: Measure the system's ability to process a high volume of transactions (product searches, add to cart, checkout) per second.
4. Scalability Testing: Evaluate system scalability by gradually increasing the user load and monitoring server resources, response times, and error rates.
5. Performance Metrics: Capture and analyze key performance metrics including response time, throughput, error rate, server CPU/memory usage, and database performance.

6. Network Latency: Evaluate network latency impact on response times for users accessing the website from different geographic locations.

**Out of Scope:**

1. Backend System Testing: Detailed testing of backend processes such as inventory management, order processing, payment gateways, and shipment tracking are out of scope.
2. Client-Side Performance: Testing client-side performance metrics like browser rendering time, client device performance, and client network conditions are not included.
3. Security Testing: Security testing such as penetration testing, vulnerability assessments, and authentication/authorization testing are outside the scope of this performance testing.
4. Localization and Internationalization: Testing for localization (language support) and internationalization (currency, date formats) aspects are not part of this performance testing.

5. Browser Compatibility: Testing compatibility across different browsers, devices, and operating systems for rendering and functionality is not included in this scope.

# Application Architecture

<Please write here the summary of the architecture, technology used, impacted components etc.>

< Get these details from FS, BR and other reference documents provided in your organization for your project>

## Overview: System Architecture

<Please write here the detailed description of the application/system>

< Get these details from FS, BR and other reference documents provided in your organization for your project>

## Architecture Diagram

<Add architectural diagram of the application in this section>

< Get these details from FS, BR and other reference documents provided in your organization for your project>

## Detailed information on each component

<Please write here about each component which is in the system and may impact due to the changes (not for a new application). Also, write how each component will interact with others, what would be the input and what would be the output? Etc.>

< Get these details from FS, BR and other reference documents provided in your organization for your project>

# Performance Test Requirements

## Requirements

<Please write here the justification to include the performance testing for this project. Attach the Performance Score Metrics sheet or MOM in which Performance Testing of specific or all the components was agreed.>

**Justification for Including Performance Testing:**

1. **User Experience Assurance:** Performance testing helps ensure that the system meets user expectations in terms of response times, page load times, and overall system responsiveness. This directly impacts user satisfaction and retention.
2. **Scalability Testing:** By simulating load scenarios, performance testing identifies how the system behaves under varying user loads. This information is crucial for scaling the infrastructure to handle growing user bases or seasonal spikes in traffic.
3. **Risk Mitigation:** Performance issues such as slow response times, crashes under load, or resource bottlenecks can lead to poor user experiences, increased support costs, and reputational damage. Performance testing helps identify and mitigate such risks early in the development lifecycle.
4. **Optimization Opportunities:** Performance testing uncovers performance bottlenecks such as inefficient code, database queries, or server configurations. Addressing these bottlenecks improves system efficiency, reduces resource consumption, and lowers operational costs.
5. **Compliance and SLA Adherence:** Many projects have service level agreements (SLAs) or regulatory requirements related to system performance. Performance testing ensures compliance with these standards and helps in meeting contractual obligations.

### Business NFR

<This section will contain all the non-functional requirements which come from the project team. These requirements could be in the layman term or very high level. A typical example is given in the table>:

Table 3: Business NFR

| Business Transactions | User Load | SLA/response times | Transactions per hour | NFR ref# |
| --- | --- | --- | --- | --- |
| Order Fish | 50 | 5 seconds | 1500 | NFR1 |
| Order Dog | 50 | 5 seconds | 1500 | NFR2 |
| Order Cat | 50 | 5 second | 1500 | NFR3 |
| Order Reptiles | 50 | 5 seconds | 1500 | NFR4 |
| Order Birds | 50 | 5 seconds | 1500 | NFR5 |
| Login and Logout | 100 | 4 Seconds | 2500 | NFR6 |
| Add to Cart | 100 | 4 seconds | 1800 | NFR7 |
| Search item | 120 | 4 Seconds | 1600 | NFR8 |

## NFR and NFT Matrix

<This section contains the non-functional test cases (scripts) and applicable non-functional requirement>

Table 4: NFR-NFT Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **NFT1** | **NFT2** | **NFT3** | **NFT4** | **NFT5** |
| **NFR1** | √ | √ | × | × | × |
| **NFR2** | × | × | √ | × | × |
| **NFR3** | √ | × | × | √ | × |
| **NFR4** | √ | × | × | × | × |
| **NFR5** | × | √ | √ | × | √ |

# Performance Test Planning

## Performance Test Approach

<Write a high-level approach for performance testing of the application under test. >

Apprach as given in the below table:

|  |  |
| --- | --- |
| Load Test | **Load**: 100 concurrent users  P**ass/Fail criteria**: Response time < 2 seconds, Error rate < 1%  **Duration**: Ramp-up: 5 mins, Steady State: 1 hour, Ramp-down: 5 mins |
| Stress Test | **Load**: 200 concurrent users (2X peak load)  P**ass/Fail criteria**: Response time < 3 seconds, Error rate < 1%  **Duration**: Ramp-up: 8mins, Steady State: 1 hour, Ramp-down: 8 mins |
| Soak/Endurance Test | **Load**: 100 concurrent users  P**ass/Fail criteria**: Response time < 2 seconds, Error rate < 1%  **Duration**: Ramp-up: 15 mins, Steady State: 10 hour, Ramp-down: 15 mins |

Table 5: Change Requests (CRs)

| Task ID | Description | Project Affected |
| --- | --- | --- |
| CRNFT01 | Response Time too high for Login Page | XXXXXX |
| CRNFR02 | XXXXX | XXXXXX |
| CRNFT03 | XXXXX | XXXXXX |

### Performance Testing and Monitoring Tool Details

Table 6: Description of Performance Testing Tool

| Tool Name | Description | Licensed / Open-Source? | No. of licenses |
| --- | --- | --- | --- |
| Apache JMeter | Version: 5.6.2  Required Protocol: Web HTTP/HTTPS  Support Forum Link:  Support ID: | Open-Source | Not Applicable |
| xxxxxxx | Version  Support Forum Link:  Support ID: | xxxxxxx | xxxxxxx |
| xxxxxxx | xxxxxxxx | xxxxxxxx | xxxxxxx |

### Performance Test Script Steps

<In this section, the performance test scripts that need to be developed are detailed by user action step as shown in the tables below. For each key Business Process within the Application under Test which was agreed from the project; a Performance Test script needs to be developed.

The transaction flow and script details must be given like below table: Develop performance test scripts that simulate all of the actions in the Business Processes/Transactions documented in the Load Model.>

Table 7: Performance Test (Script 1 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 01\_Order\_Fish** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Click on Sign In |
| 3 | Enter user id and password and click on sign in |
| 4 | Click Fish |
| 5 | Click on Product Id |
| 6 | Click on Add to Cart |
| 7 | Click on Proceed to checkout |
| 8 | Enter payment Details |
| 9 | Click on confirm |
| 10 | Click on Sign out |

Table 8: Performance Test (Script 2 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 02\_Order\_Dog** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Click on Sign In |
| 3 | Enter user id and password and click on sign in |
| 4 | Click Reptiles |
| 5 | Click on Product Id |
| 6 | Click on Add to Cart |
| 7 | Click on Proceed to checkout |
| 8 | Enter payment Details |
| 9 | Click on confirm |
| 10 | Click on Sign out |

Table 9: Performance Test (Script 3 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 03\_Order\_Reptiles** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Click on Sign In |
| 3 | Enter user id and password and click on sign in |
| 4 | Click Dog |
| 5 | Click on Product Id |
| 6 | Click on Add to Cart |
| 7 | Click on Proceed to checkout |
| 8 | Enter payment Details |
| 9 | Click on confirm |
| 10 | Click on Sign out |

Table 10: Performance Test (Script 4 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 04\_Order\_Cats** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Click on Sign In |
| 3 | Enter user id and password and click on sign in |
| 4 | Click Cats |
| 5 | Click on Product Id |
| 6 | Click on Add to Cart |
| 7 | Click on Proceed to checkout |
| 8 | Enter payment Details |
| 9 | Click on confirm |
| 10 | Click on Sign out |

Table 11: Performance Test (Script 5 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 05\_Order\_Birds** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Click on Sign In |
| 3 | Enter user id and password and click on sign in |
| 4 | Click Birds |
| 5 | Click on Product Id |
| 6 | Click on Add to Cart |
| 7 | Click on Proceed to checkout |
| 8 | Enter payment Details |
| 9 | Click on confirm |
| 10 | Click on Sign out |

Table 12: Performance Test (Script 6 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 06\_Add\_to\_Cart** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Click on Sign In |
| 3 | Enter user id and password and click on sign in |
| 4 | Click on any available <Shopping Item> |
| 5 | Click on Product Id |
| 6 | Click on Add to Cart |

Table 13: Performance Test (Script 7 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 07\_Login\_and\_Logout** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Click on Sign In |
| 3 | Enter user id and password and click on sign in |
| 4 | Click on Sign Out |

Table 14: Performance Test (Script 8 Steps)

| Step # | Application Name: JPet Store  Business Process Name: Product Ordering  **NFT Script Name: 08\_Search** |
| --- | --- |
| 1 | Click on Enter the store URL |
| 2 | Key in Search Value in Search box |
| 3 | Click on Search button |

Table 15: Performance Test Runtime Settings (Provide if available)

| Script # | Pacing between Iterations | Think Time between transactions |
| --- | --- | --- |
| Script 1 | 6 seconds (Fixed) | 10 seconds (Fixed) |
| Script 2 | 5-10 seconds (Random) | 5-10 seconds (Random) |
| Script 3 | No Pacing | 10 seconds (Fixed) |
| Script 4 | No Pacing | 5-10 seconds (Random) |
| Script 5 | 12 seconds (Fixed) | 10 seconds (Fixed) |
| Script 6 | 12 seconds (Fixed) | 10 seconds (Fixed) |
| Script 7 | 12 seconds (Fixed) | 10 seconds (Fixed) |
| Script 8 | 12 seconds (Fixed) | 10 seconds (Fixed) |

### Performance Test Data Planning

<Please write a summary of the test data that will be needed during the Performance Test phase. Provide the details of what type of data and how much test data will be needed for all the in-scope business flows. The requirement of test data should include all the cycles as well as releases. Also, need to mention whether test data will be prepared or extracted from the production>

|  |
| --- |
| **Required Test Data** |
| User Id |
| Password |
| Search Values |
| Product name, product Id |

#### Data Preparation

<Please write the procedure that will be used to prepare the test data for Performance Test. Define the procedures needed to create the test data. If there is any third party or external tool required to prepare the test data then name the tool and procedure. If test data is being copied from production then mention the confidentiality of the data.>

Approach:

Test data can be either created by ourself or request feedback on test data from back end team/developers/product owners/business analysts to provide the test data>

Store the test data in csv file and use that data while running tests.

# Performance Test Execution

## Performance Test Summary

<The table below provides an example of a short summary of each of the Performance Test scenario runs.>

Table 16: Performance Test Scenarios

| Test Run | Date | Test Scenario Summary |
| --- | --- | --- |
| Smoke Test | To Be Determined (TBD) | To validate the performance test scripts and monitors |
| Cycle 1 - Run 1 | TBD | Load Test - 1 Hour test with peak load |
| Cycle 1 - Run 2 | TBD | Repeat Load Test - 1 Hour test with peak load |
| Cycle 1 - Run 3 | TBD | Stress Test - 1 Hour test with 200% of peak load |
| Cycle 1 - Run 4 | TBD | Repeat Stress Test - 1 Hour test with 200% of peak load |
| Cycle 1 - Run 5 | TBD | Soak Test - 24 Hour continuous Test with average load |
| Cycle 1 - Run 6 | TBD | Repeat Soak Test - 24 Hour continuous Test with average load |
| Cycle 2 - Run 1 | TBD | Load Test - 1 Hour test with peak load |
| Cycle 2 - Run 2 | TBD | Repeat Load Test - 1 Hour test with peak load |
| Cycle 2 - Run 3 | TBD | Stress Test - 1 Hour test with 2000% of peak load |
| Cycle 2 - Run 4 | TBD | Repeat Stress Test - 1 Hour test with 2000% of peak load |
| Cycle 2 - Run 5 | TBD | Soak Test - 24 Hour continuous Test with average load |
| Cycle 2 - Run 6 | TBD | Repeat Soak Test - 24 Hour continuous Test with average load |

## Performance Test Details

### Smoke Test

The smoke test is designed to ensure that the performance test scripts are working in the Performance Test Environment. The smoke test is also used for making sure the Performance Monitors that are configured for metrics collection are operating as expected. The smoke test can also be used to run with 1 to 10 users test to determine how long it takes for transaction steps to complete. This method is valuable for the runtime settings pacing of the test.

### Load Test

Table 17: Load Test Scenarios Detail

|  | Test Details |
| --- | --- |
| Test ID | NFT01 (Cycle 1-Run1, Cycle 1-Run2, Cycle 2-Run1 and Cycle 2 - Run 2) |
| Purpose | Peak hour transaction processing will be under examination to determine if the system can maintain response times under the highest anticipated load. This test is designed to collect performance metrics on transaction throughput, response times, and system resource utilization, in comparison to Performance requirements. |
| No. of Tests | 4 (2 tests per cycle) |
| Duration | Ramp-up: 5 mins  Steady State: 1 hour  Ramp-down: 5 mins |
| Scripts | Script 1 , Script 2, Script 3, Script 4, Script 5, Script 6, Script 7, Script 8 |
| Scenario Name | Load Test Scenario |
| Covered NFR | NFR1, NFR2, NFR3, NFR4, NFR5, NFR6, NFR7, NFR8 |
| User Load / Volume | 100 users (Threads) Load |
| Entry Criteria | 1. The code should be stable and functionally verified 2. Test Environment should be stable and ready to use 3. Test Data should be available 4. All the NFRs should be agreed with the project 5. Test scripts should be ready to use 6. XXXXXX |
| Exit Criteria | 1. All the NFR must be met 2. The error rate of transactions must not be more than 1% of total transaction count 3. CPU utilization must not be more than 70% |

### Stress Test

Table 18: Stress Test Scenarios Detail

|  | Test Details |
| --- | --- |
| Test ID | NFT02 (Cycle 1-Run 3, Cycle 1-Run 4, Cycle 2-Run 3 and Cycle 2 Run 4) |
| Purpose | Stressing the system to view if the workload increases in the future then how the application and infrastructure scales. This test will be conducted to determine if response times can be maintained. This test is designed to collect performance metrics on transaction throughput, response times, and system resource utilization, in comparison to Performance requirements. |
| No. of Tests | 4 (2 tests per cycle) |
| Duration | Ramp-up: 8 mins  Steady State: 1 hour  Ramp-down: 8 mins |
| Scripts | Script 1 , Script 2, Script 3, Script 4, Script 5, Script 6, Script 7, Script 8 |
| Scenario Name | Stress Test Scenario |
| Covered NFR | NFR02, NFR04 and NFR05 |
| User Load / Volume | 200 users (Threads) Load |
| Entry Criteria | 1. The code should be stable and functionally verified 2. Test Environment should be stable and ready to use 3. Test Data should be available 4. All the NFRs should be agreed with the project 5. Test scripts should be ready to use 6. XXXXXX |
| Exit Criteria | 1. All the NFR must be met 2. The error rate of transactions must not be more than 1% of total transaction count 3. CPU utilization must not be more than 70% |

### Soak/Endurance Test

Table 19: Endurance/Soak Test Scenarios Detail

|  | Test Details |
| --- | --- |
| Test ID | NFT03 (Cycle 1-Run 5, Cycle 1-Run 6, Cycle 2-Run 5 and Cycle 2 Run 6) |
| Purpose | This soak test will determine if the system resources are recycled for re-use while processing transactions over long periods. Proper recycling of memory, CPU, and other system utilization resources is healthy for performance. This test is designed to collect performance metrics on transaction throughput, response times, and system resource utilization, in comparison to Performance requirements with o memory leakage. |
| No. of Tests | 4 (2 tests per cycle) |
| Duration | Ramp-up: 15 mins  Steady State: 24 hours continuous  Ramp-down: 15 mins |
| Scripts | Script 1 , Script 2, Script 3, Script 4, Script 5, Script 6, Script 7, Script 8 |
| Scenario Name | Soak Test Scenario |
| Covered NFR | NFR02, NFR03 and NFR06 |
| User Load / Volume | 100 users (Threads) Load |
| Entry Criteria | 1. The code should be stable and functionally verified 2. Test Environment should be stable and ready to use 3. Test Data should be available 4. All the NFRs should be agreed with the project 5. Test scripts should be ready to use 6. XXXXXX |
| Exit Criteria | 1. All the NFR must be met 2. The error rate of transactions must not be more than 1% of total transaction count 3. CPU utilization must not be more than 70% 4. No Memory leakage 5. XXXXXX |

## Performance Test Monitoring Metrics

<The two tables below describe examples of the various performance metrics that can be captured during the Performance Test stage to view resource usage trends.>

Table 20: Metrics

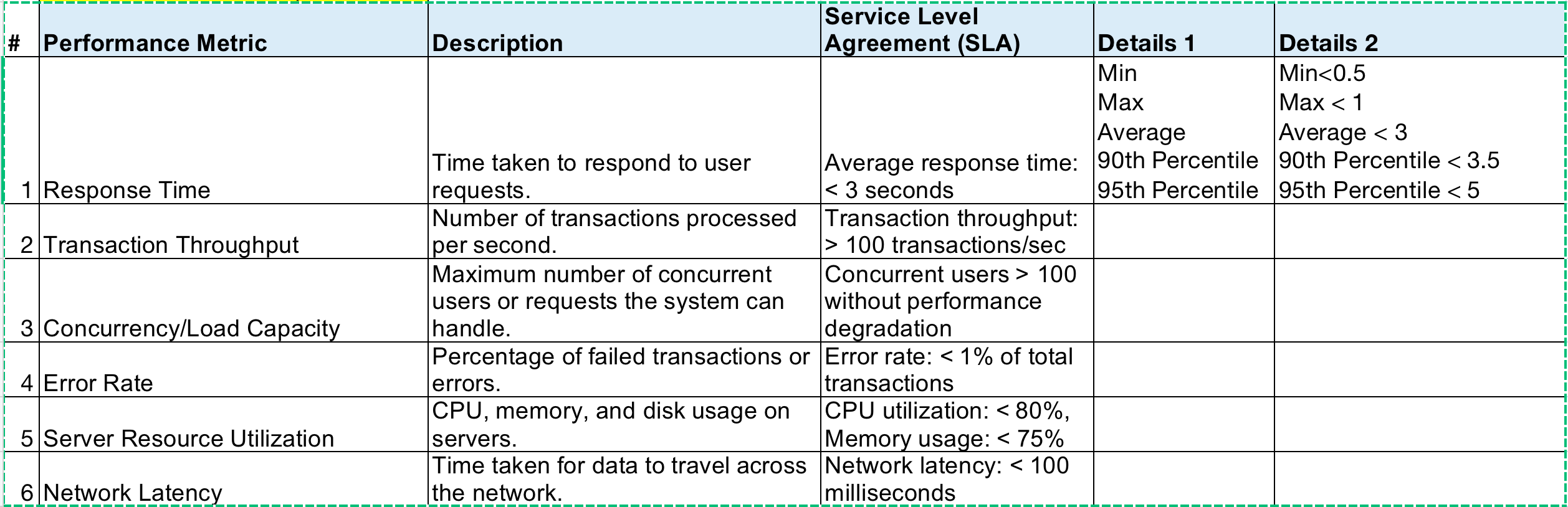


Table 21: Server Side (App server, DB Server, Web Server )

| Metrics | Value Measured |
| --- | --- |
| CPU utilization | CPU utilization |
| Physical Memory Percentage used | Physical Memory Percentage used |
| Memory | Memory utilization  Processes in the run queue (Procs r), User Time (CPU US), System time(CPU SV), Idle time (CPU ID), Context Switching (cs), Interrupts |
| Disk I/O | Disk I/O utilization  Read/Write per sec (r/s, w/s), Percentage busy (%b), Service Time (svc\_t) |
| Network | Collisions (Collis), Output Packets (Opkts), Input errors (Ierrs), Input Packets (Ipkts) |

## Performance Test Environment

The Performance Test environment is XX% of the production environment. Hence user load has been scaled down to XX%. Post-execution, the test result will be extrapolated with the same percentage.

*< As listed below, describe what the Scaling factor between the Production environment that will support the Application under Test, and the Performance Test environment that will support the Application under Test. >*

*The Scaling factors are as follows:*

1. *Number of CPUs (processors)?*
2. *Memory*
3. *Disk Space*
4. *Load Balancer and its configuration like algorithm*
5. *Environment configuration files – It should be the same in both the environment*
6. *Test Data – It should be populated in the Database to the same level as in Production? If not, what is the ratio?*

Table 22: Performance Test Environment Details

| Server Name | Environment Tier | Hardware Version | OS | Memory (GB) | CPU count | Total Disk Space |
| --- | --- | --- | --- | --- | --- | --- |
| xxx | Web Service | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Web Service | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Middleware | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Middleware | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Middleware | M820 | Linux | 32 GB | 16 cores | 1 TB |
| xxx | Database | M820 | Linux | 32 GB | 16 cores | 1 TB |
| xxx | xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx | xxx |

Table 17: Production Environment Details

| Server Name | Environment Tier | Hardware Version | OS | Memory (GB) | CPU count | Total Disk Space |
| --- | --- | --- | --- | --- | --- | --- |
| xxx | Web Service | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Web Service | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Middleware | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Middleware | M620 | Linux | 32 GB | 8 cores | 512 GB |
| xxx | Middleware | M820 | Linux | 32 GB | 16 cores | 1 TB |
| xxx | Database | M820 | Linux | 32 GB | 16 cores | 1 TB |
| xxx | xxx | xxx | xxx | xxx | xxx | xxx |
| xxx | xxx | xxx | xxx | xxx | xxx | xxx |

## Assumptions, Constraints, Risks and Dependencies

### Assumptions

<Assumptions should be documented concerning the available release software, test environment, dependencies, tools, and test schedule associated with the performance test. Examples are shown below.>

Table 18: Assumptions

| No. | Assumption |
| --- | --- |
| 1 | The code version vx.x is stable and passed in functional testing before deploying in the Performance Testing environment. |
| 2 | The required license must be available in the Performance Center to run the test. |
| 3 | The fully deployed, installed and configured Web tier, middleware tier, and database servers must be operational in order for performance testing shake-out to begin. |
| 4 | Test Data must be provided to the performance testing team before testing starts |
| 5 | xxxxxxxxxx |

### Constraints

<Constraints should be documented concerning the available release software, test environment, dependencies, tools, test schedule, and other items pertaining to the performance test. Examples are shown below.>

Table 19: Constraints

|  |  |  |
| --- | --- | --- |
| No. | Constraint | Impact |
| 1 | The Performance Test environment has 50% of the servers that Production has. | The scaling factor of the Performance Test to Production is 50%. All Production Load Models that are executed in the Performance Test should be run at 50% of the full Production load Model to represent a 100% Load Test in the Test environment. |
| 2 | The Performance Test environment does not have some of the older data that Production has, which limits some of the data scenarios that can be simulated. | The data in Production has not been purged since xxxx; searches in Production intermingle with older data than Performance Test can. This could limit the capability of reproducing some Production issues. |
| 3 | The Performance Test team does not have a commercial tool or an approved Wire Shark-like tool that allows for measuring network response times using packet captures. | The impact of network response times will not be measurable as we determine what areas within the Architecture are responsible for transaction response time cost. This constraint will leave network response time cost-related questions unanswered. |
| 4 | xxxx | xxxx |

### Risks

<Risks should be documented concerning the test schedule, release software, dependencies, tools, test approach test environment and other items pertaining to the performance test. Examples are shown below.>

Table 20: Risks

| No. | Risk | Impact | Action/Mitigation | Assigned To |
| --- | --- | --- | --- | --- |
| 1 | If functional errors from validation testing occur and prevent the creation of performance test scripts or performance test execution, execution of performance test project tasks will be delayed until functional errors can be addressed. | HIGH | The team will start Performance Test execution once environment certification, test script validation, and data staging efforts are completed. | Project Manager |
| 2 | If a performance-tuning effort is conducted in the middle of the performance test execution schedule and as a result configuration or code changes are made to the environment, any tests executed prior to the performance-tuning changes should be re-executed. | HIGH | It is recommended that any tests that were executed before the performance tuning changes should be re-executed after the performance-tuning changes. | Project Manager, Performance Engineering |
| 3 | xxxx | xxxx | xxxx | xxxx |

### Dependencies

<Dependencies should be documented concerning the latest build, test data, schedule, required tools’ installation, test environment and other items pertaining to the performance test. Examples are shown below.>

Table 21: Risks

| No. | Dependencies | Impact | Action/Mitigation | Assigned To |
| --- | --- | --- | --- | --- |
| 1 | The latest build should be available in the environment before NFT start date | HIGH | The team will start Performance Test execution once the environment has the latest and functionally tested code. | Developer |
| 2 | Test data should be provided by the test data team | HIGH | Test data team will fetch the production data and provide to the performance testing team. | Test Data Team |
| 3 | xxxx | xxxx | xxxx | xxxx |

# Milestones

Key milestones are listed in the table below. Each of the milestones represents a group of tasks on which completion of Performance Testing is dependent. If any of the milestones are listed as “At Risk”, the milestones that follow it will most likely be delayed as well.

Table 22: Schedule of Milestones

| ID | % Done | At Risk | Task | Due Date | Interface |
| --- | --- | --- | --- | --- | --- |
| 1 | 0-100 | Yes or No | Preliminary Project Plan submitted | xx/xx/xxxx | Project Management |
| 2 | 0-100 | Yes or No | Final Project Plan submitted | xx/xx/xxxx | Project Management |
| 3 | 0-100 | Yes or No | Performance Requirements and Production Load Model reviewed and verified | xx/xx/xxxx | Requirements Management and Performance Engineer |
| 4 | 0-100 | Yes or No | Environment Planning | xx/xx/xxxx | Environment Team and Project Management |
| 5 | 0-100 | Yes or No | Test Plan | xx/xx/xxxx | Performance Engineer |
| 6 | 0-100 | Yes or No | Script Development and Test Data Planning | xx/xx/xxxx | Performance Engineer and Vendor Project Team |
| 7 | 0-100 | Yes or No | Environment Certification and Test Script Validation | xx/xx/xxxx | Project Management and Environment Team |
| 8 | 0-100 | Yes or No | Data Staging and Setup | xx/xx/xxxx | Performance Engineer and Vendor Project Team |
| 9 | 0-100 | Yes or No | Performance Monitoring Configuration | xx/xx/xxxx | Environment Team and Performance Engineer |
| 10 | 0-100 | Yes or No | Test Execution and Analysis | xx/xx/xxxx | Performance Engineer, Monitoring Tool administrators, and Development |

### Test Organization

<Document the test organization and any other departments that will be supporting the Performance Test Phase.>

Table 23: Test Organization

| Name | Functional Role | Responsibilities |
| --- | --- | --- |
| Name | Project Manager | Facilitating and coordinating all schedules related to SDLC phases and infrastructure |
| Name | Performance Engineering Lead | Manages schedules and activities related to Performance Testing projects |
| Name | Performance Engineer | Prepares for performance test execution, executes performance tests, analyzes performance tests, and tracks problem reports |
| Name | Performance Engineer | Prepares for performance test execution, executes performance tests, analyzes performance tests, and tracks problem reports. |
| Name | Monitoring Support | Monitors performance tests using Performance monitors |
| Name | Application Support | Supports performance test execution as configuration or application issues are found |
| Name | Performance Test Environment Support | Supports and maintains the Performance Test environment |

1. Acronyms

<List out all the acronyms and associated literal translations used within the document. List the acronyms in alphabetical order using a tabular format as depicted below.

Table 24: Acronyms

| Acronym | Literal Translation |
| --- | --- |
| NFR | Non-functional Requirement |
| PT | Performance Testing |
|  |  |
|  |  |
|  |  |

1. Glossary

<Write down the clear and concise definitions for terms used in this document that may be unfamiliar to readers of the document. Terms are to be listed in alphabetical order.>

Table 25: Glossary

| Term | Definition |
| --- | --- |
| Pacing | The delay between two iterations |
| Think Time | The delay between two transactions |
|  |  |
|  |  |

1. Referenced Documents

<List out the documents which were referred during the preparation of Performance Test plan. Also, provide who and when the reference document was prepared along with version>

Table 26: Referenced Documents

| Document Name | Document Location and/or URL | Issuance Date |
| --- | --- | --- |
| AO (Architecture Overview)  Version: 2.2 | <https://xxxxxx.xxxxx.com/project_document/architecture/ao.doc> | 30/10/2022 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |