|  |
| --- |
| **Performance Test Strategy**  **v.1.0** |

March 17.03.2019

**Table of Contents**

1. Introduction
2. Performance test objectives
3. Application Overview
4. Entry criteria
5. Metrics
6. System Configuration
7. Test environment
8. Testing tools
9. Test scripts
10. Type of Performance Testing to be Conducted
11. Testing schedule
12. Risk
13. Exit criteria
14. Reports

**Introduction**

This Performance Engineering Strategy document defines the approach to testing the system. It briefly describes the methods and tools which are used.

The purpose of this document is to outline the approach that the Performance Engineering team will take to ensure that the Performance Acceptance Criteria is met. Specifically, this document details the:

• Performance Acceptance Criteria

• Workload Distribution to be used to exercise and gather measurements from the application

• Testing schedule

• Tests to be performed

• Metrics to be collected

• Data and data management issues

**Performance test objectives**

1. To carry out root cause analysis of performance related common uncommon problems plans to tackle them.
2. To identify the problems causing the malfunctioning of the system & fix them well before the production run. Problems remedied during later
3. To ensure that the new system conforms to the specified performance criteria.

**Application Overview**

A brief description will provide the context for the measurement of the performance test parameters. The overview should include a high-level description of the functionality being tested under load.

This project is a set of simple services, which partially represent common logic elements of e-commerce systems. It contains 3 Dockerized python apps:

**Authentication service**

For tracking user activities, the system uses token, which could be generated with GET /api/auth/generate\_token. Requests to all other services without token would fail.

##### **Methods**

* GET /api/auth/generate\_token
* GET /api/auth/validate\_token/<string:token\_id>
* Authentication service port: 7778

#### **Product service**

The service is contain list of products and information about them.

##### **Methods**

* GET /api/products/get\_all
* GET /api/products/get\_product/<string:product\_id>
* Product service port: 7777

#### 

#### **Cart service**

Using this service authenticated user can add products to cart and then get,clear all products which were added.

##### **Methods**

* GET /api/cart/get\_items
* GET /api/cart/add\_item/<string:product\_id>
* GET /api/cart/checkout
* Cart service port: 7779

**Entry criteria**

Before QA start testing should be implement next points:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Responsible | When should be provided | Comment | Status |
| Stable environment | Andrii Sezko |  |  | DONE |
| Test scripts | Oleksandr Golubishko |  |  | In progress |
| Test data | Oleksandr Golubishko |  |  |  |
| Test metrics | Oleksandr Golubishko | At beginning SDLC |  | DONE |
| No blockers/criticals bugs | Oleksandr Golubishko | 2 weeks before PQA started write scripts | Before doing performance testing, application should be tested manually. | DONE |
| Development cycle |  |  | Must be completed and all code should be freezed. Latest code should be successfully deployed on UAT server. | DONE |

**Metrics**

|  |  |  |
| --- | --- | --- |
| **Performance Metrics** | | |
| Name | Description | Comments |
| Response time | * Minimum * Maximum * Average |  |
| Error Rate | * Passed% * Failed% |  |
| Throughput | Bits/Seconds   * Minimum * Maximum * Average * Total |  |
| Average Latency | Wait Time. The time it takes from when a request is sent until the first byte is received. |  |
| Connect time | Time when our request connect to the server |  |
| Memory Use | % of available memory used |  |
| CPU Usage | % of available CPU used |  |

**6. System configuration**

This section should contain technical information:

* A description of the specific system architecture, including servers (e.g., web, database)
* Specific details of computing hardware (e.g., CPU cores, RAM, Solid State
* Disks (SSD), Hard Drive Disks (HDD) and software)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OS | CPU | RAM | SSD | HDD |
| Windows | 4 cores | 16 G | 512 | 1 T |

**7. Testing environment**

This section includes a description of which testing environments will be used by performance testing team.

|  |  |  |
| --- | --- | --- |
| Environment | Purpose | Status |
| 192.168.99.100 | For development and first tests | Done |
| stress.stage | Similar to production | In progress |

**8. Testing tools**

This section includes a description of which test tools (and versions) will be used in scripting, executing and monitoring the performance tests.

* Gatling - for performance testing
* Postman - to investigate how API work

**Test scripts**

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, a Performance Test script needs to be developed.

|  |  |  |
| --- | --- | --- |
| **End2End** | | |
| Step | Description | Comment |
| 1 | Get token |  |
| 2 | Validate token |  |
| 3 | Get all product |  |
| 4 | Get product by Id |  |
| 5 | Add several product to cart |  |
| 6 | See your cart items |  |
| 7 | Clear cart |  |

Full scope will be provided after check this strategy by lecturer!!!!!!!!!!!!

**Type of Performance Testing to be Conducted**

**Smoke** - before run all tests scripts need to run smoke suite to make sure that environment is ready for testing. Also this suite will be rub first after fix or changes on environment.

**Capacity -** to determine allowable and under allowable boundary.

**Load testing** - PQA team will investigate API up to allowable boundary.

**Stress testing** - PQA team will investigate API under allowable.

**Endurance testing** - according to business needs, need to provide good quality 24/7 hours. Testing will focused on the stability of the system over a time frame specific to the system’s operational context.

**Schedule**

QA activity will be doing according next schedule.

|  |  |  |
| --- | --- | --- |
| **Activites** | **Milestone** | **Comments** |
| Determine KPI |  |  |
| Create test scenario |  |  |
| Create test scripts |  |  |
| Test execution |  |  |
| Test Evaluation and Reporting |  |  |

**Risk**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Impact** | **Possible mitigation** |
| Critical issue found during testing | Blocking testing for an indefinite period | TBD |
| Testing environment is not ready | Blocking testing for an indefinite period | TBD |
| Requirement changes | Wrong metrics and test strategy | TBD |
| Manual testing is not done | Possible to find critical bug. | TBD |

**Exit criteria**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Value** | **Priority** |
| System performance has been assessed according to the goals of the testing. |  | High |
| Number of P0/P1 | 0 | High |
| Number of open P2/P3 | 10% of total | Medium |
| Test reports write and passed to relevant people |  | High |