Performance Test Plan for
PROJECT A

General information	
Customer	<Project name>
Created by (Author)	
Preparation date	
Version	
Status	

Revision History					
Version	**Description**	**Author**	**Date**	**Approved by**	
				Author	**Date**

Summary

1. Introduction **4**

1.1. Purpose of the document 4

1.2. Objective 4

2. Scope of project **4**

2.1. The components and functions to be tested 4

2.2. The components and functions not to be tested 4

3. Quality criteria **4**

4. The decisive factors of the project success 5

5. Limitations, assumptions and risks **5**

5.1. The risks of the project 5

5.2. Plan to reduce the risks 6

5.3. Assumptions 6

6. Resources **6**

6.1. The team of external testing 6

6.2. Tools and services for testing **6**

7. Deliverables **7**

7.1. Testing Documentation and Reports **7**

8. Strategy of testing **7**

8.1. Testing phases 7

8.2. Acceptance criteria **8**

8.3. Completion criteria **8**

8.4. Reporting **8**

9. Requirements for the application for performance testing **9**

10. Test iterations **9**

10.1. Main test run **9**

11. Performance test scenario **10**

12. Load infrastructure **11**

1. Introduction

1.1. Purpose of the document

This document describes a test plan for the project "A" and approaches, which the test team will use to verify the quality of the product. The document also lists the different resources that are needed for a successful performance testing of the project.

1.2. Objective

The purpose of the test plan is to formalize the testing process, plans and approaches to testing, interfacing process with the development team and the project team to achieve the high quality of the software product. The plan takes into account the specifics of the functionality of the project "A"

2. Scope of project

2.1. The components and functions to be tested

ID	Components/ Applications name	Functions	Link
1	Front end	- Product search and navigation through E-commerce website. - Purchasing actions	https://www.example.com

2.2. The components and functions not to be tested

ID	Components/ Applications name	Functions	Comment
1	Back end		Purpose of performance testing is testing web application under load generated by the certain amount of users on the front end.
2	Connected 3rd party services	Services intended for metrics collection, performance monitoring and infrastructure maintenance.	These services are connected for additional needs and are not related to the performance test scenario.

3. Quality criteria

The delivered product must work in accordance with the requirements and the functional specification listed in sections “Scope of Work“.

The delivered product must not contain any known defects with critical and high priority in the final version.

4. The decisive factors of the project success

- The application should not include known defects with critical and high priority at the time of the final version.
- The application correctly handles required amount of load, without any errors or performance issues.

5. Limitations, assumptions and risks

- The late submission of information or delays in document approval by the Customer.
- Changes in the requirements for performance testing during the testing process.
- Ambiguous requirements can increase the risk of insufficient coverage of functionality by performance testing or risks when input test data or test scenario does not reflect actual product requirements or usage.
- The narrow time frame increases the risk of bugs appearance during performance script development and testing. If the timing of development and environment preparation phases are not met, it will directly affect the timing of testing.
- Unformed or not formed enough team from the Customer's side, which is responsible for monitoring the infrastructure of the application during performance testing, may lead to incorrect performance testing results and application breakdown.

5.1. The risks of the project

ID	Risk description	Probability (High / Medium /Low)	Influence (High / Average / Low)	Effects on Cost / Schedule / Quality
1	The late submission of information, delays in document approval by the Customer	Medium	High	Schedule
2	Incorrect or incomplete stated requirements for testing	High	High	Cost, Schedule
3	Changes in the requirements during testing	High	High	Cost, Schedule
4	Problems with application infrastructure configuration, unavailability of servers.	High	Medium	Schedule
5	Errors in the 3d party performance monitoring tools of the software	Low	High	Schedule, Quality

6	The narrow time frames. If the timing of development and environment preparation phases are not met, it will directly affect the timing of testing	Medium	High	Cost, Schedule, Quality
7	Unformed or not formed enough team from the Customer side, which is responsible for monitoring the infrastructure of the application during performance testing	Medium	High	Schedule, Quality
8	Insufficient amount or incorrect amount of statistics from the application that may lead to incorrect or incomplete performance testing	Medium	High	Quality

5.2. Plan to reduce the risks

ID	Actions to reduce the risk
1	Compliance with the rules of planning and organizing meetings. Timely information about the unavailability of employees (including due to vacation, illness, etc.). The schedule of meetings and the provision of necessary information in advance
2	Splitting testing into several iterations. Frequent testing results discussions
3	Fixing the basic list of requirements in the contract
4	Getting further details on installing the product from the Customer's IT department as soon as possible
5	The provision of an initial stage of development for defining and studying architecture
6	Follow the development schedule. Timely notification of potential problems or shifts in the schedule
7	Pre form a team of developers, system administrators and testers from the Customer’s side before performance testing
8	Provide detailed statistics from application and infrastructure about application usage for a long period of time (a few months at least)

5.3. Assumptions

All requirements for performance testing are not yet defined in detail. Estimates made on the basis of how the NexoQA sees the system at the time of the analysis requirements. Estimates may change (increase or decrease) depending on the appearance of new requirements for the system

6. Resources

6.1. The team of external testing

Company	Name	Role	Contact Information
NexoQA	Firstname Lastname	Program Manager	Skype: live:skype E-mail: mail@example.com
NexoQA	Firstname Lastname	QA Lead	Skype: live:skype E-mail: mail@example.com

6.2. Tools and services for testing

#	Tool	Comment
1	Apache JMeter	Performance testing tool for performance scripts development and execution
2	AWS EC2	Cloud hosting service for load infrastructure setup

7. Deliverables

7.1. Testing Documentation and Reports

#	Title	Responsible person	Frequency (delivery time)	Delivery method
1	Test Plan	QA Lead	One time before testing	e-mail
2	Scenario performance testing	QA Lead QA Team	Upon receipt of the final version of specification	e-mail
3	Bug reports	QA Lead QA Team	After bug detection	e-mail
4	Reports on the results of testing	QA Lead QA Team	After every test / deliveries	e-mail
5	Source code of testing scripts	QA Lead	After all tests	e-mail

8. Strategy of testing

8.1. Testing phases

Main stages of work of the testing team:

1. The testing team gets information about the application (access to the application, testing data) and check what can be tested in case of performance testing.
2. Collect initial statistics information from the application that can be used for performance test plan preparation and performance scenario development.
3. Prepare performance test scenario and confirm it with the Client. Make time estimates needed for testing script development and give the approximate time needed to perform these tests for the desired amount of virtual users.

4. Record and correct testing scripts.
5. Execute the script using low amount of virtual users and generate sample report.
6. Update performance testing scripts if needed.
7. Find the suitable time to provide the main part of the testing. The Client organizes a team from his side: system administrator, programmers and testers, everyone who will monitor the health status of the application and servers and can tweak or reboot infrastructure in case of any problems.
8. The testing team prepares load infrastructure before the testing depending on the statistics of usage from step 2.
9. Run testing script using specified amount of virtual users according to actual statistical information from the application using information from Test iterations section. The Client's team is monitoring the application.
10. After the testing is done, generate execution report and send it to the Client.

8.2. Acceptance criteria

1. Requirements for performance testing are received and confirmed.
2. Testing team has access to the application, has all required test data (test accounts, input data).
3. The system is fully configured and ready for performance testing. In the case of “development” or “testing” environment, it is configured in the same way as “product” environment.
4. Test data is loaded into the database of the application in the amount enough for performance testing.
5. The client assigns the task to the testing team.

Test team can partially or completely suspend work, if the following occurs:

1. There is an error in functionality, which does not allow continuing testing.
2. There is a serious problem that prevents the continuation of testing (non-working or damaged test environment, force majeure, such as turning off the Internet or electricity).
3. The developers have not corrected the problem that blocked the testing.

8.3. Completion criteria

1. All test scenarios of the plan for performance testing were performed, performance testing is conducted.
2. Performance testing reports are prepared and sent to the Client.
3. The source code of performance scripts is sent to the Client.

8.4. Reporting

The tools described in Tools and services for testing section will be used to collect the results. Metrics and statistics will be included in the reports, including:

1. Statistics summary:
    - Maximum running concurrent users
    - Total throughput
    - Average throughput
    - Average hits per second
    - HTTP responses summary
2. Transactions summary:
    - Total passed transactions
    - Total failed transactions
3. HTTP responses summary:
    - Total amount of HTTP 2XX responses
    - Total amount of HTTP 4XX responses
    - Total amount of HTTP 5XX responses
4. Running concurrent users graph
5. Response times graph

The reports contain metrics and statistics described above, a list of issues (with description and links to statistics section) that occurred during tests execution, general conclusion about the performance of the application.

The reports are prepared by the testing team after each iteration of performance scripts execution and sent to the Customer.

9. Requirements for the application for performance testing

The following requirements for the application and load amount values for the testing are under consideration and may change later.

The application must meet the following requirements:

1. The application must respond without errors.
2. The application is required to be available 24 hours per day every day.
3. All user transactions must respond to the user within 60 seconds.

10. Test iterations

10.1. Main test run

#	Operation description	Time (minutes)
1	Initialize first 1 concurrent thread	1
2	Increase the load by 50 concurrent threads per 60 seconds till the number of 250 concurrent threads is reached	5
3	Keep the load using 250 concurrent threads	10
4	Increase the load by 50 concurrent threads per 60 seconds till the number of 500 concurrent threads is reached	5
5	Keep the load using 500 concurrent threads	10
6	Increase the load by 50 concurrent threads per 60 seconds till the number of 750 concurrent threads is reached	5
7	Keep the load using 750 concurrent threads	10
8	Increase the load by 50 concurrent threads per 60 seconds till the number of 1000 concurrent threads is reached	5
9	Keep the load using 1000 concurrent threads	20
10	Finish test execution, gradually stop concurrent threads	3

11. Performance test scenario

Performance testing includes one test scenario that will be conducted for all test iterations. The test scenario includes the following actions

#	Action name	% of total users	Links
1	Open home page	100	<link>
2	Sign in	30	
3	Select a subcategory from main menu	50	
4	Search for a product using global search functionality	40	
5	Open a product page	30	<link>
6	Select product options	30	
7	Add a product to Cart	10	
8	Navigate to checkout	7	
9	Place the order	5	

Actual values of % of total users can be discussed and updated before the performance script execution

12. Load infrastructure

The testing team prepares load infrastructure before performance scripts execution. The infrastructure is consists of a few components:

- Load controller. This station is used by the automation team to manage scripts execution, adjust the number of virtual users (concurrent threads) during tests execution, analyze results after the testing and generate execution report.
- A set of load generators (load servers). Server stations for the required amount of time to provide load testing. These servers are located across the world in different datacenters and used by load controller during load testing to generate virtual users (send requests to the application, process responses and collect statistics).

#	Region	Instance type	Amount of servers
1	US West (Oregon)	c4.large	6-9

All servers are provided by AWS EC2 cloud hosting service

The web application under test is configured with a load balancer and 3 VMs. Thereby performing the testing using 6-9 different AWS instances with different IPs should grant optimal load generation for the application.