12-07-2022 Task 13

Test Purpose

* Prepare and present Test strategy, Test plan, Test cases for the demoblaze.com

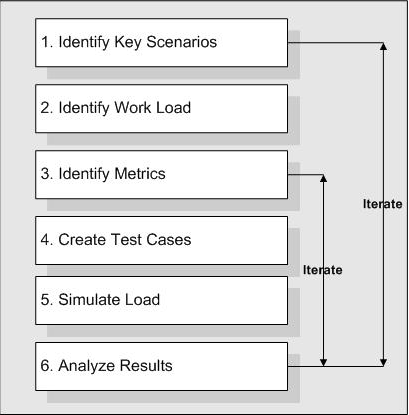
Test plans

|  |  |  |
| --- | --- | --- |
| **Start date:** 05.05.2022  **End date:** 05.07.2022 | **Introduction:** BlogEngine.NET 3.2 | |
| **In Scope**   * Smoke testing * Capacity testing * Load testing * Regression testing * Scalability testing * Volume testing * Stability testing * Stress testing * Web Service Testing | **Entry Conditions**   * Installed JMeter * Installed Grafana * Installed InfluxDB * Installed Telegraf | **People**  QA: Nehrii Stanislav |
| **Exit Conditions** | **Test Environment**  Load testing tool: Apache JMeter 5.4.3  Software/hardware configuration: I7 3.0 GHz, 32 GB, Disk 96.3 GB, Windows 10 64bit |
| **Out of Scope** | **Risks**   * unstable internet connection * unstable electrical supply * We need to use dev environment | **Timescales**  1. Planning and estimating - 3 days  2. Вevelopment of test scripts - 2 weeks  3. Running Tests - 2 weeks  4. Reporting - 1 week |

Test strategy

Performance Testing Process

The following steps are involved in performance testing process



1.1 Identify Key Scenarios

It is important to identify application scenarios that are critical for performance rather than testing all the scenarios that are defined by the functional test team. Here the scenarios refer to the web methods that need to be Performance tested.

- Main script

- Anonymous script

- User script

- Admin script

1.2 Identify Work Load

It is important to identify the work loads for the various scenarios mentioned above. In the real world several Performance Test / web methods which will be accessed simultaneously and the load distribution among them are to be identified and simulated in the test process.

**- Need to get more performance baselines from client**

1.3 Identify Metrics

Identify the metrics that need to be collected during the performance testing. This will help us to come up with the appropriate reports to analyze the test results. The primary metrics that needs to be identified are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Throughput** | It is the sum of the data rates that are delivered to all terminals in a network | **Server metrics** | We need to monitor each individual system resources like Memory, CPU, Disk I/O, network bandwidth |
| **Response Time** | We need to check the web services and UI response time | **Client side metrics** | FCP (First Contentful Paint)  LCP (Largest Contentful Paint)  SI (Speed Index)  TTI (Time to Interactive)  TBT (Total Blocking Time)  CLS (Cumulative Layout Shift) |
| **Maximum concurrent and simultaneous users** | We have to check how many concurrent user can sustain in the environment |

1.4 Create Test Cases

Create comprehensive test cases where the scenarios will be simulated. The test scripts need to be generated for all the test cases and also need to make sure all variants of the Performance Test are tested for all the scenarios. The variants which will be tested are:

- UI Scenarios to select and order Items with logging off

- UI Scenarios to select and order Items with logging on

- UI Scenarios to add and\or remove Items by Admin

1.5 Simulate Load

Concurrent user load will be simulated in various patterns using Jmeter. Following Load Testing types will be conducted to simulate the demoblaze.com application load with the production load.

1.5.1 Performance Testing Types

|  |  |
| --- | --- |
| Testing Type | Purpose of the testing |
| Load Testing | By doing the load testing we need to verify demoblaze.com UI, server behavior under normal and peak load conditions  Objectives:- To Check the response times, throughput rates, resource utilization levels |
| Stress Testing | By doing the stress testing we need to evaluate our demoblaze.com UI, server behavior when it is pushed beyond the normal or peak load conditions.  Objectives:- synchronization issues and memory leaks. Stress testing enables you to identify your application's weak points, and how it behaves under extreme load conditions |
| Capacity Testing | By doing the capacity, we can determine the server's ultimate failure point. We can use capacity planning to plan for future growth of the application, such as an increased user base or increased volume of data. For example, to accommodate future loads we need to know how many additional resources (such as CPU, RAM, disk space, or network bandwidth) are necessary to support future usage levels. Capacity testing helps you identify a scaling strategy to determine whether you should scale up or scale out. |
| Volume Test | By doing volume testing is to evaluate the performance of a system while increasing data flow. During testing, an increase in the intensity of system operations with a simultaneous increase in database volumes is emulated. |
| Scalability testing | By doing scalability testing analyzes how well our system scales in response to different load levels |
| Web Service Testing | Need to test the each individual web-service and need to find out the system behavior , when we apply the load  Objective:- To check the response time and the server utilization |

1.5.2 Performance Test Activates and deliverables

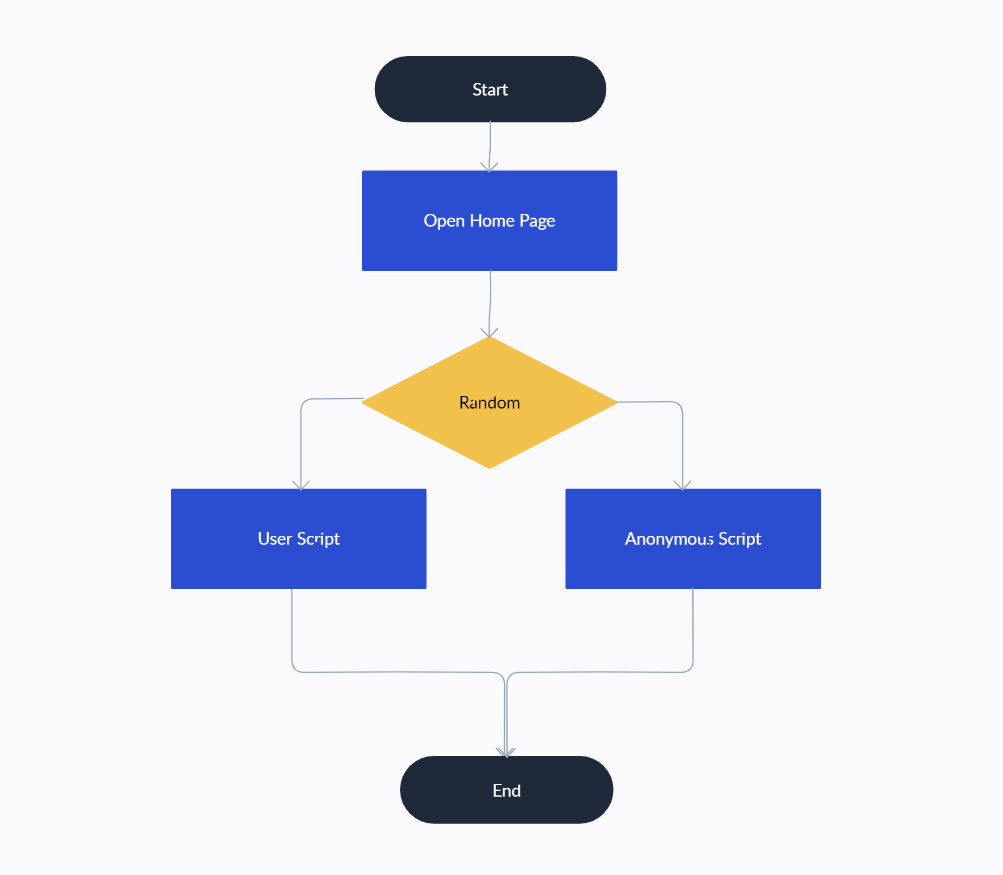
|  |  |  |  |
| --- | --- | --- | --- |
| **Activities** | **Responsible Teams** | **Deliverables** | **Responsible Teams** |
| Set-up of test environment | Client Team has to provide the stage environment to do the performance testing. | Test Strategy | Performance Team |
| Set-up of test data | Performance team needs to coordinate with manual and dev teams to get the required test data. | Test Plan | Performance TEAM |
| Performance Test Execution | The Performance Team is responsible for doing this testing. | Test Scripts | Performance TEAM |
| Monitoring and collating results | Performance team needs to monitor the servers. Performance team is responsible for gathering the results. | Test Executions | Performance TEAM |
| Obtaining user acceptance | Need to get customer approval for the performance testing results. | Results | Performance TEAM |

1.6 Analyze the Results

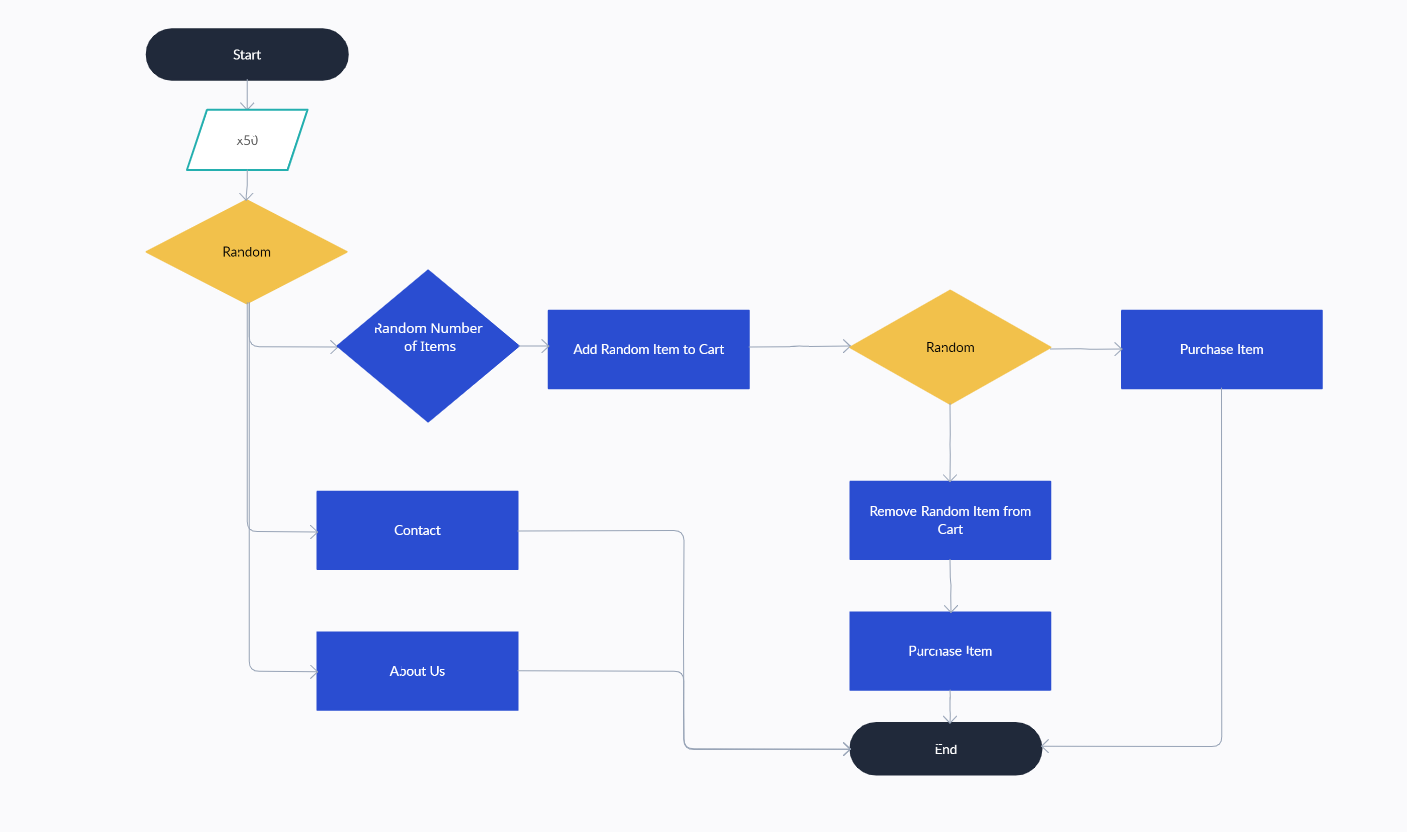
Analyze the metric data captured during the test against the predefined expectations. If the expectations are not met, necessary modifications are made and testing is repeated until the desired results are achieved.

Test cases for testing demoblaze.com

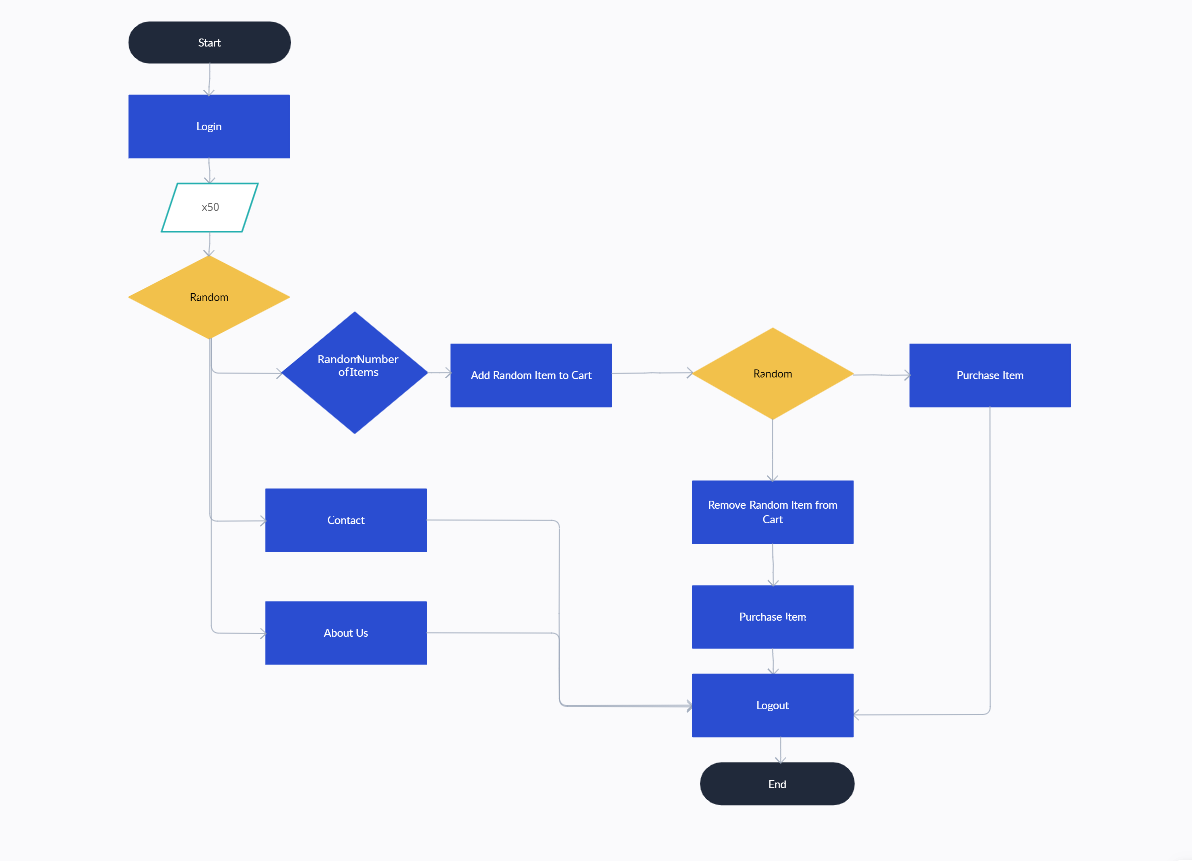
* Main script



* Anonymous script



* User script



* Admin script

