BlogEngine

Test plan

# Introduction

## Purpose of this Test plan

The purpose of this document is to specify performance requirements and conditions for BlogEngine application. The document will outline scenarios, test cases, parameters and data used in evaluating the capacity of the included features.

## Test plan identification

|  |  |
| --- | --- |
| Application Name | BlogEngine |
| Test Cases (scripts) | Anonymous user flow  Admin user flow  Editor user flow |

# Test description

## Test objectives

* + 1. Check application’s availability on selected environment
    2. Get basic response time for each tested item with no load
    3. Define capacity in term of number of concurrent virtual users
    4. Check system counters from involved servers

## Items to be tested

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Action** | **Test case (script)** | **Notes** |
| Open Home Page | Open Home Page | Anonymous, Admin, Editor |  |
| Open Random Date | Open Date | Anonymous |  |
| Open Predefined Date | Open Date | Anonymous, Editor |  |
| Search by Name | Search | Anonymous |  |
| Open Large Calendar | Open Large Calendar | Anonymous |  |
| Open Contacts | Open Contacts | Anonymous |  |
| Open Random Page | Open Random Page | Anonymous |  |
| Open First Post | Open Post | Anonymous |  |
| Open Random Post | Open Post | Anonymous, Editor |  |
| Send Comment | Send Comment | Anonymous |  |
| Login | Login | Admin, Editor |  |
| Logout | Logout | Admin, Editor |  |
| Edit Post | Edit Post | Editor |  |
| Open Admin Page | Open Admin Page | Admin |  |
| Create new user | Create User | Admin | Depends on number of existing users |
| Delete existing user | Delete User | Admin | Depends on number of existing users |

1. Home Page

This is the first page opened after entering the application. Contains list of newest posts.

Main operations:

* Open Posts
* Check posts on small calendar and open large calendar
* Search
* Choose posts by Category lists, Tags, Months, Authors.
* Log In

1. Search

Page with Search results. Search happens in post name or text. Each post can be opened.

1. Large Calendar

Contains calendar of current month with posts linked to days they were created. Month can be switched to previous or next.

1. Admin page

Can be opened by logged in users.

Main operations:

* 1. Admin:
  + Create new users
  + Edit or delete posts
  + Delete comments
  + Change settings
  1. Editor:
  + Create new posts
  + Edit own posts

## Items not to be tested

All other features not listed in previous point.

## Test data

1. Pre-created test users: 2 admin users and 2 editor users.
2. Different number of posts needed for different test types (check Non-Functional Requirements).

## Test user roles

1. Anonymous user – can surf in the BlogEngine, open separate Posts, send comments.
2. Editor user – all rights of Anonymous + can create new posts, edit or delete posts created by that user.
3. Admin user – all rights of Anonymous + can create new users with different level of access, edit or delete any posts, moderate comments, set up settings of whole BlogEngine.

# Non-Functional Requirements

Initial Non-Functional Requirements may be the next:

|  |  |
| --- | --- |
| **NFR** | **Value** |
| Number of anonymous/admin/editor users | 500/2/10 |
| Number of text posts | 100 or 1000 (2000 and 5000 for volume tests) |
| Number of media posts (volume tests only) | 1000 |
| Image size in Media posts | 1 MB |
| Average response time for pages with posts (Home, Search, open date, separate post) | <0.5s |

After set of tests NFRs can be updated according to obtained test results.

# Monitored metrics

## Client metrics

* Throughput
* Average response time
* Median response time
* 90 percentile response time
* 95 percentile response time
* Error rate

## Hardware metrics

* CPU load
* Memory usage
* Cache and Garbage Collector
* Disk I/O (databases)
* Network usage

# Suspension criteria and resumption requirements

## Suspension criteria

* **Not valid or not stable build** of application
* **The need for updates of test scripts** because of significant changes in functionality of application
* **Testing tools issues** - Load Generator, Metric Collector, Visualizer, CI/CD
* **Test environment issues**. Wrong configuration, access problems
* **Test data issues**. Not reached number of users, absence of certain user type accounts, not enough number of posts and stories.

## Resumption criteria

* **Test plan is completed** (updated) and approved.
* **Correct and stable build is running** on chosen test environment.
* Required **test scripts are completed and valid**. Smoke test verified all key steps.
* All **testing tools are confirmed to work** during the test.
* **Test environment has correct configuration** which is verified and approved for usage in current test.
* **Test data satisfies requirements** to run tests with different types of user and posts.

# Performance Entry, Exit, and Suspension Criteria

## Entry Criteria

* Test plan is complete and approved by the client.
* Correct version is installed in performance testing environment, i.e. the version previously functionally tested and fixed if needed
* Test data is complete and in the performance testing environment in sufficient time to allow test scripts to be completed.
* Test accounts have been created in the performance testing environment in sufficient time to allow test scripts to be completed.
* Test scripts complete.
* All assigned resources are available to monitor the test.
* All parameter sets used in the script are generated based on the Database values.

## Exit Criteria

* All test scripts completed successfully
* No critical problems encountered
* All non-critical problems are logged
* All test logs are captured
* All post-test notifications sent

## Suspension Criteria

* Not all test scripts will complete
* Critical problems are encountered and logged
* Hardware errors prevent the completion of the test

# Test deliverables

Main expected test deliverables are:

* Test strategy and plan
* Basic test scripts and scenarios
* Baseline of main metrics for deployed functionality: response times for the pages, throughput (requests per second), system resources consuming
* Test reports and analysis

# Testing tasks

To run each test case next steps are required:

1. Basic scripting
2. Basic scenarios creation
3. Setting up load generation tools
4. Setting up monitoring and collecting tools
5. Deployment stable version to environment
6. **Smoke** testing. Is used for general check of application availability on selected environment and confirms that test script is valid.
7. Running set of performance testing:
   * 1. **Capacity**. First testing to determine load model for other types of testing.
     2. **Regular load** test. Is completed regularly for each change of application or configuration. Runs with ~50% of capacity.
     3. **Volume.** Checks system and application behavior with different levels of load: low - ~30% of capacity, middle - ~60% of capacity, and high - ~90% of capacity. Is used to find possible issues with data (new photos/videos are not added properly, existing posts have problems with loading or photo/video quality)
     4. **Longevity.** Low load test to check system and application behavior under constant load and to find possible memory leaks. Runs with ~20% of capacity
     5. **Scalability**. Checks application behavior with different configurations.

All test cases (Anonymous, Admin, Editor) will be used for testing types described in p.6-7 as they cover different parts of application and create different type of load for application. See “Schedule” section to check what is the flow to have different types of tests for different test cases.

# Test environments

For performance testing a dedicated environment is needed. The configurations of the servers should be as much as close to production’s ones.

Expected servers:

* Server with deployed .NET application of BlogEngine.

To perform partial and full end-to-end performance testing it’s recommended to have a separated database and other services, i.e. they should not be located on the same server where related services are being tested by another team.

**Test environment specifications:** Close to Production environment as much as possible.

# Schedule and estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Stage** | **Period (estimation)** | **Notes** |
| 1 | Project onboarding | 10-15 days |  |
| 2 | Test strategy design | 4-5 days |  |
| 3 | Test plan creation | 4-5 days |  |
| 4 | Draft of NFR definition | 4-5 days |  |
| 5 | Setting up test environment | 15-20 days |  |
| 6 | Test data preparation | 2-3 days |  |
| 7 | Script/Scenarios design | 13-15 days |  |
| 8 | Automation test running from CI/CD | 1-2 day |  |
| 9.1 | Smoke and Capacity tests for Anonymous test case | 2-3 days |  |
| 9.2 | Anonymous test results analysis | 1-2 days |  |
| 9.3 | Anonymous test reporting | 1 day |  |
| 10.1 | Smoke and Load tests for Admin test case | 1-2 days |  |
| 10.2 | Admin test results analysis | 1-2 days |  |
| 10.3 | Admin test reporting | 1 day |  |
| 11.1 | Smoke and Load tests for Editor test case | 1-2 days |  |
| 11.2 | Editor test results analysis | 1-2 days |  |
| 11.3 | Editor test reporting | 1 day |  |
| 12 | Update of anonymous test script and test data | 1 day |  |
| 13.1 | Smoke and Capacity tests for Anonymous test case with updated probabilities | 2-3 days |  |
| 13.2 | Anonymous test results analysis | 1-2 days |  |
| 13.3 | Anonymous test reporting | 1 day |  |
| 14 | Combining of test scripts | 2 days |  |
| 15.1 | Smoke and Capacity tests for general test case (combined test script) | 2-3 days |  |
| 15.2 | Combined test results analysis | 1-2 days |  |
| 15.3 | Combined test reporting | 1 day |  |
| 16.1 | Scalability test for Combined test case | 4-5 days |  |
| 16.2 | Scalability test results analysis | 1-2 days |  |
| 16.3 | Scalability test reporting | 1 day |  |
| 17.1 | Volume test for Combined test case. Includes additional test data preparation | 4-5 days |  |
| 17.2 | Volume test results analysis | 1-2 days |  |
| 17.3 | Volume test reporting | 1 day |  |
| 18.1 | Longevity test for Combined test case | 4-5 days |  |
| 18.2 | Longevity test results analysis | 1-2 days |  |
| 18.3 | Longevity test reporting | 1 day |  |
| 19 | NFR update, load level definition | 1 day |  |
| 20 | Updating scripts/scenarios | 2-3 days |  |
| 21 | Starting regular load test running (Combined test script) | 1-2 days |  |

# Risks

* A significant difference in configuration from the production environment
* Performance testing results can be essentially different even in case of minor difference in think times, arrival rate and test duration
* During the execution of the tests, some major performance or functional problems that may require code changes, creation of a new build may be discovered and in that case it may be necessary to repeat the load test from the beginning
* Load test should be performed against a build that is solid enough, and that has been functionally tested, after code is complete. Failure to follow this rule may result on rework to update test scripts for every new build, plus the load test may need to be repeated from the beginning. This will affect the schedule
* Performance testing tool is not capable of identically reproducing real life scenarios - so results could only be trusted as having limited reliability level
* Network/systems latency issues
* Environment’s unavailability

# Test plan versions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Creator (editor)** | **Reviewer** |
| 1.0 | 30.08.2022 | Oleksandr Maksymenko | Mykhailo Kurshakov |
|  |  |  |  |
|  |  |  |  |