Performance Testing Mentoring Program 2022-2023

Task 12 Part 2

Created by: Serhii Diedov

Date: May 2023

BlogEngine.NET 3.2
PERFORMANCE TESTING TEST PLAN

1 Introduction

1.1. Purpose of this Test plan

The purpose of this document is to specify performance requirements and conditions for BlogEngine.NET 3.2 application. The document will outline the test scenarios, test cases, parameters and data used in evaluating the capacity of the included features.

1.2. Test plan identification

Application Name	BlogEngine .NET 3.2
Phase of the testing	1
Test Cases	Scenario for a user with the "Admin" role Scenario for a user with the "Editor" role Scenario for a user with the role "Anonymous"

2 Test description

2.1. Test objectives

2.1.1.Check test scripts workability
2.1.2.Check application’s availability
2.1.3.Get basic response time for each tested item with no load
2.1.4.Run the tests below
2.1.5.Get the required metrics

2.2. Sequence of test execution

First, tests will be carried out for the BlogEngine .NET 3.2 (web) for which the data source will be the file system. After that, data source from file system to DB will be switched and check the impact of changes on the system. The results for both data sources will be presented for comparison.

2.2.1. Test types assumed for conducting:

Before each test:

Smoke testing should be performed every time when functionality of the application and the script needs to be checked. Also, if needed can be used as a warming-up test before the main testing step.

Tests to be carried out for both types of system data sources:

2.2.1.1 Perform capacity testing (for 1000 text posts) and compare results with both file system and DB data source.

2.2.1.2 Perform scalability testing (for 1000 text posts) and compare results with both file system and DB data source using the following approach:

2.2.1.2.1 Approach to determine the impact of the number of CPUs on system performance:

a) Set the size of the memory as bigger as possible for the virtual machine (6Gb).

b) CPU scaling and perform load testing under regular load (~70% of the system capacity) for different numbers of CPUs: 1, 2, 3, 4, 6.

2.2.1.2.2 Approach to determine the impact of the RAM on system performance:

a) Set the number of CPUs as big as possible (6).

b) Scaling RAM and perform load testing under regular load (~70% of the system capacity) for different sizes of RAM: 2Gb, 3Gb, 4Gb (if possible), 6Gb (if possible).

2.2.1.3 Perform volume testing and compare results with both file system and DB data source using the following approach:

a) Testing under regular load (~70% of the system capacity) with different number of the text posts 100, 1000, 2000, 5000.

b) Testing under regular load (~70% of the system capacity)with media information and compare results for 1000 text posts and 1000 posts with a text and attached 1Mb photo

2.2.1.4 To ensure system stability under long-term load, **additionally perform long-time testing** under low load (~30% of the system capacity)

!!!TBD!!! Given the time constraints, these tests can only be carried out on a system with a data source that has shown the best results in terms of stability and potential for scaling:

2.2.1.5 Perform regular load testing (~70% of the system capacity) with the parameters based on the results of capacity testing (for 1000 text posts) **and perform stress testing** with a load of 120%, 200%, 300% and 500% system capacity. The goal is to determine the ability of the system to cope with the stress load and to identify the consequences of such a load on the system.

2.3. Test user’s roles

In the application, users have 3 main roles "Admin", "Editor" and "Anonymous" user.

During testing, three scripts will be executed simultaneously:

- Scenario for 2 users (according to NFR received from customer) with the "Admin" role;
- Scenario for 2 users (according to NFR received from customer) with the "Editor" role;
- Scenario for user with the role "Anonymous" (the number of users is indicated for each test in paragraph 2.2. of this plan).

2.4. Items to be tested

Below is a list of functionality with an indication of the roles under which this functionality will be used in test scenarios.

#	Page Name/Feature	Roles
1.	Open Home page	Anonymous
2.	Open Contacts	Anonymous
3.	Open Large Calendar	Anonymous
4.	Open Predefined Date	Anonymous
5.	Open Random Date	Anonymous
6.	Open Random page =1	Anonymous
7.	Open Random page >1	Anonymous
8.	Search by Name	Anonymous
9.	leave a comment	Anonymous
10.	Open first post	Anonymous
11.	Open Random post	Anonymous
12.	Open Home page	Anonymous
13.	Open login page	Admin
14.	Login	Admin
15.	Open Admin Page	Admin
16.	Open USERS menu	Admin
17.	Open Users page	Admin
18.	Add user	Admin
19.	Delete user	Admin
20.	LogOff	Admin
21.	Login	Editor
22.	Open editpost page	Editor
23.	Open Predefined Date	Editor
24.	Open Random post	Editor
25.	Editpost	Editor
26.	LogOff	Editor

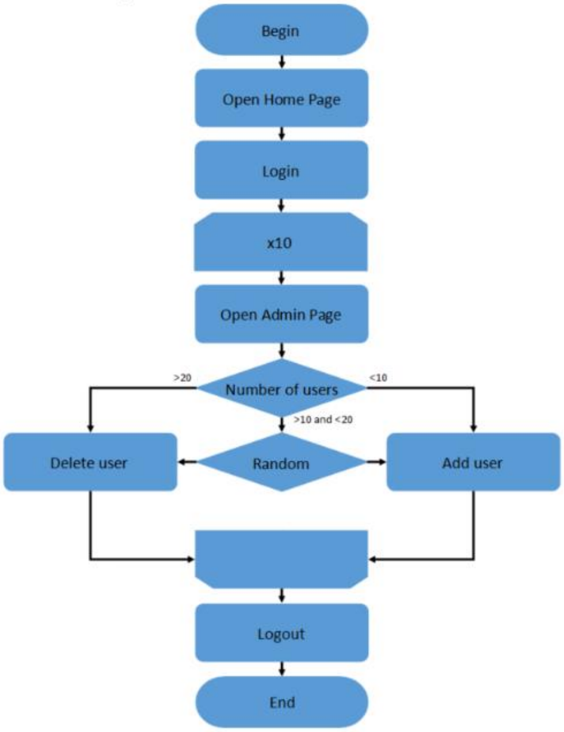
2.5. Items not to be tested

Functionality not listed above will not be used in the implementation of test scenarios

2.6. Detailed description of test scenarios (scripts)

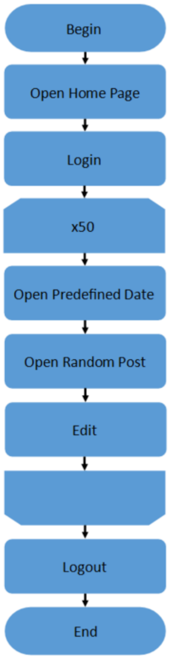
2.6.1.Admin script

First, open the home page then log in as an Admin user, after that 10 times check number of Anonymous users and decide to add or delete users, and finally log out.



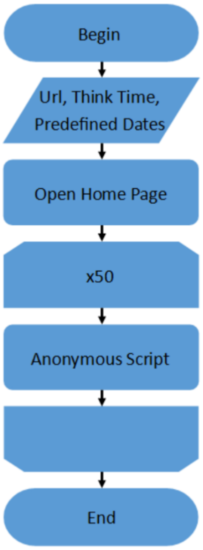
2.6.2. Editor script

First, Open home page, log in as an Editor user, open a predefined date, open a random post and edit it, rapid it 50 times, and finally Log out.



2.6.3. Anonymous Script

First, open the home page then 50 times runs Anonymous Script.



In the Anonymous Script:
Step 1. The following pages open with the following probabilities:

1. Home Page: 15%

2. Open Random Date: 10%

3. Open Predefined Date: 30%

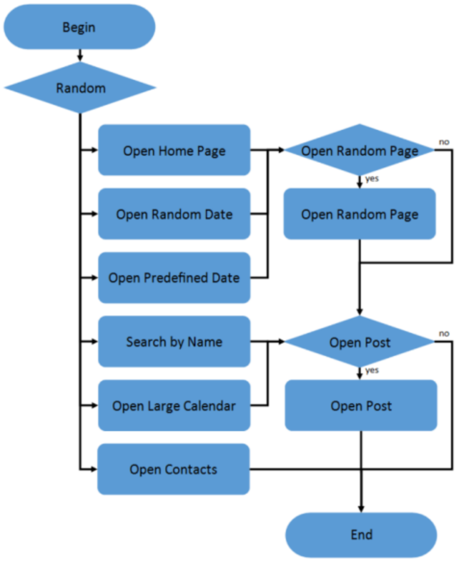
4. Search by Name: 30%

5. Open Large Calendar: 10%

6. Open Contacts: 5%

Step 2. If Step 1 is opened one of the first three pages (home page, page of random date, page of predefined date from csv file) then make a random decision to open or not to open a random page (if there is more than one page with posts).

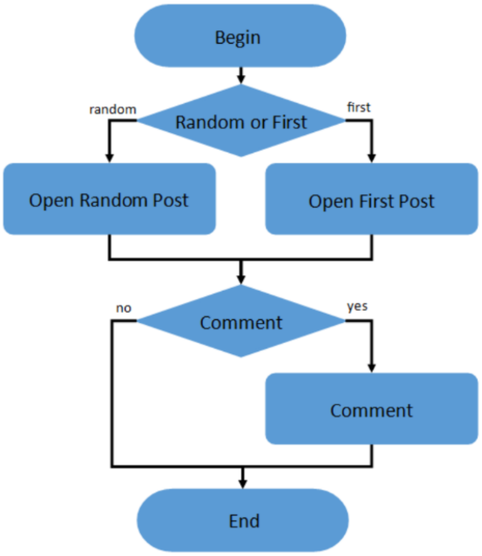
Step 3. After Step 2 (or after Step 1 if there was no opened one of the first three pages) make a decision to open or not to open a post with 80% / 20% probabilities.



If "Open post script" is selected

Step 1. Make a decision to open a random post of the first post with 65% / 35% probabilities.

Step 2. Make a decision to add a comment or not with 20% / 80% probabilities.



2.7. Test data

Test data must be generated by a separate script.

List of required test data:

- Test users at least 2 with the role of "Admin" and at least 2 with the role of "Editor"
- Text posts in the amount of 100, 1000, 2000, 5000 posts (after generating posts, you need to make copies of files / database dumps to speed up testing)
- 1000 posts with a text and attached 1Mb photo (after generating posts, you need to make copies of files / database dumps to speed up testing)

3 Non-Functional requirements

With regard to absence of defined NFR at the start of the performance testing as preliminary the following indicators may be the following:

NRF	Value	Notes
Number of **virtual users** with the role of "Admin" that work at the same time	2	
Number of **virtual users** with the role of "Editor" that work at the same time	2	
Expected average number of posts in the system	TBD (1000?)	
Average response time for requests on pages interacted with by a user with the "Anonymous" role	< 3 sec	

4 Suspension criteria and resumption requirements

4.1. Suspension criteria
- 4.1.1. Test environment issues (not ready, any trouble with application setup)
- 4.1.2. Test data issues (data generation script not ready)
- 4.1.3. Testing tools issues (load generation, monitoring)
- 4.1.4. Setting up system parameters after switching from the file system to DB (if necessary)
- 4.1.5. Making changes to the test script after switching from the file system to DB (if necessary)

4.2. Resumption criteria
- 4.2.1. Test plan is complete/updated and approved by client.
- 4.2.2. Test data is complete and in the performance testing environment in sufficient time to allow test scripts to be completed.
- 4.2.3. Test accounts have been created in the performance testing environment in sufficient time to allow test scripts to be completed.
- 4.2.4. Test scripts complete.
- 4.2.5. All assigned resources are available to monitor the test.

5 Test deliverables

Main expected test deliverables are:

1. Test plan
2. Basic test scripts and scenarios
3. Baseline of main metrics for deployed functionality: response times for the pages, system resources consuming
4. Performance indicators needed for to determine with which data source the server side of the system is more productive and stable, including with an increase in data volume, and also has the best potential for scaling.

6 Testing tasks

6.1. Basic scripting
6.2. Basic scenarios creation
6.3. Setting up load generation tools
6.4. Setting up monitoring and collecting tools

6.5. Setting up test environment
6.6. Smoke testing
6.7. Test data preparation
6.8. Saving test data
6.9. Automation test running from CI/CD
6.10. Running set of performance testing (performing a smoke test before each run):
  6.10.1. Perform capacity testing
  6.10.2. Perform scalability testing
  6.10.3. Perform volume testing
  6.10.4. Perform long-time testing
  6.10.5. Perform regular load and stress testing
6.11. Switch data source from file system to DB
6.12. Configuring system settings and permissions for roles after data source switchover
6.13. Updating scripts/scenarios
6.14. Test data preparation
6.15. Saving test data
  6.15.1. Perform capacity testing
  6.15.2. Perform scalability testing
  6.15.3. Perform volume testing
  6.15.4. Perform long-time testing
  6.15.5. Perform regular load and stress testing
6.16. Test results analysis
6.17. Prepare complex report on entire performance testing process

7 Test environment

Test environment should be described in BlogEngine.NET 3.2 strategy document.

8 Risks

Main risks are described in BlogEngine.NET 3.2 strategy document.