Performance Test Plan

1. Introduction
   1. Purpose of Test Plan
      1. The Purpose of this document is to specify performance requirements and conditions for BlogAngine application. The document will outline test scenarios, test cases, parameters and data used in performance testing process.
   2. Test Plan Identification
      1. Application name – BlogEngine
      2. Phase of testing – 1
      3. Test Cases – Anonymous script, Editor script, Admin script which include specific load model
2. Anonymous Script description
   1. A pattern of user behavior, who enters the site without authenticating, interested in posts on the pages, and may leave comments
   2. Since these users are the most numerous compared to others, the load will mainly be generated through this user
   3. User behavior’s model and components tested are described on next page
   4. Needs several posts already generated
   5. Probabilities usage:
      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%
      7. Open Random page (yes/no):

50% / 50%

* + 1. Open post (yes/no):

80% / 20%

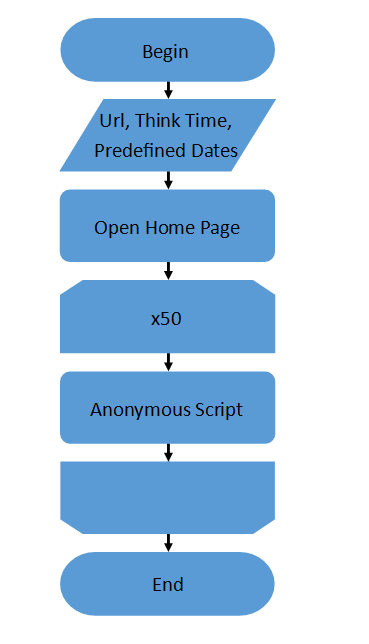
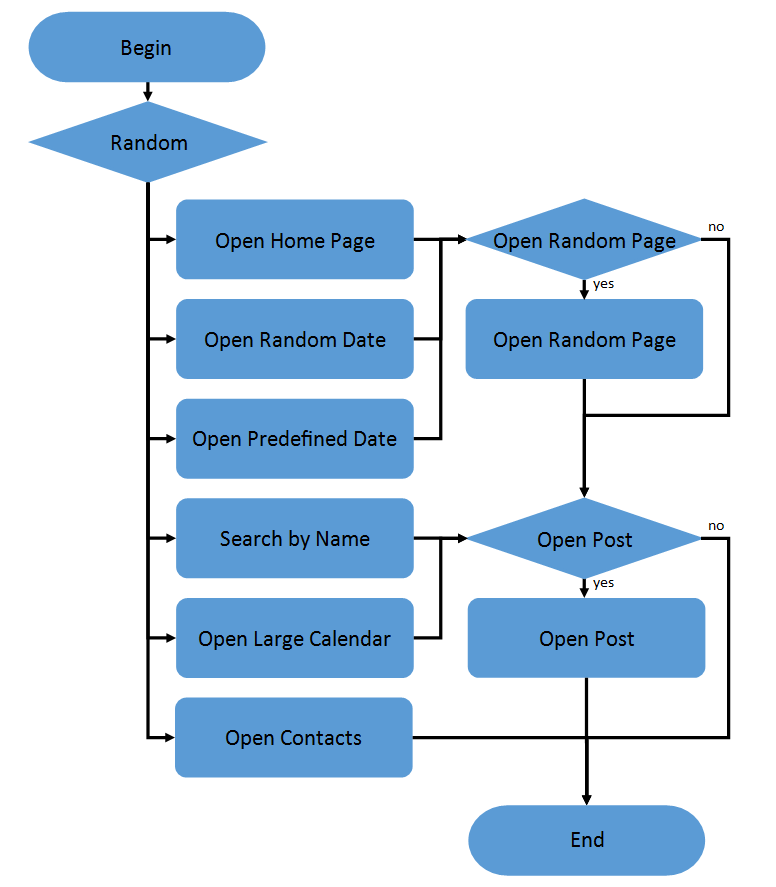
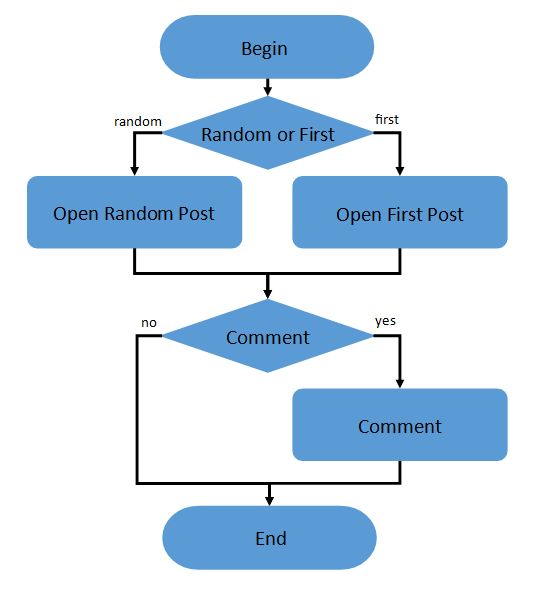
* + 1. Random or First (yes/no):

65% / 35%

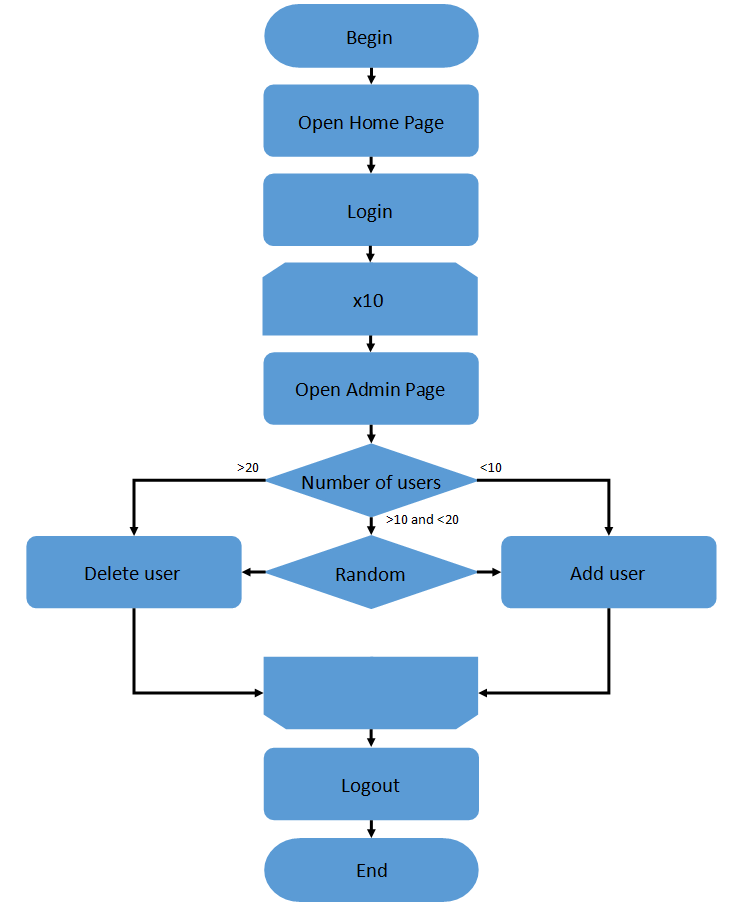
* + 1. Comment (yes/no):

20% / 80%

Anonymous script

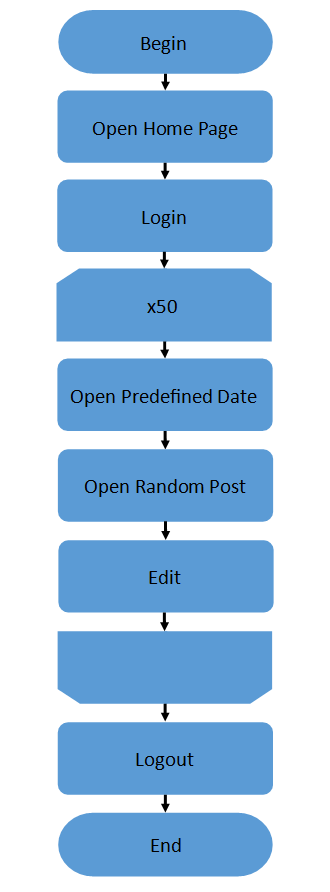
****

1. **Admin Script description**
   1. **A pattern of user behavior, who uses authorization, creates and deletes users.**
   2. **User behavior’s model and components tested:**

****

* 1. **Needs users with Admin’s role created and application configured to “Multi-user” mode**

1. **Editor Script description**
   1. **A pattern of user behavior, who edits posts.**
   2. **User behavior’s model and components tested:**



* 1. **Needs several posts generated and users with Editor’s role created.**

1. **Features not to be tested**
   1. **Functionality of the whole application**
   2. **Features which are not implemented by the start of the performance testing**
2. **Test Data**
   1. **Test data should be generated by Load Generation Tool or Manually and should include:**
      1. **Two sets of Admin and Editor users**
      2. **5000 posts files generated by Load Generation Tool with Random amount of text and unique id.**
      3. **1000 posts files generated by Load Generation Tool with Random amount of text, unique id and 1Mb photo attached.**
      4. **1000 1Mb photos**
      5. **SQLite Database with 5000 posts files generated by Load Generation Tool with Random amount of text and unique id and 1000 posts files generated by Load Generation Tool with Random amount of text, unique id and 1Mb photo attached.**
   2. The lists of users, posts files and databases should be submitted to performance testing group for running load tests
3. Test users’ roles
   1. The only two roles of Admin and Editor users to be used in performance testing.
4. Suspension criteria and resumption requirements
   1. Suspension criteria
      1. Test environment issues (not ready, any trouble with accessing to it etc.)
      2. Not stable version of the application
      3. Test data issues (no test user or wrong credentials, incomplete data)
      4. Significant changes in workflow of functionality of the application which require updates in the test plan or scripts/scenarios
      5. Testing tools issues (load generation, monitoring)
   2. Resumption criteria
      1. Test plan is complete/updated and approved by Company and the client.
      2. Correct version is installed in performance testing environment, i.e. the version previously functionally tested and fixed if needed
      3. Test data is complete and in the performance testing environment in sufficient time to allow test scripts to be completed.
      4. Test accounts have been created in the performance testing environment in sufficient time to allow test scripts to be completed.
      5. Test scripts complete.
      6. All assigned resources are available to monitor the test.
      7. All parameter sets used in the script are generated based on the Database values.
5. Testing tasks
   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
   7. Running set of performance testing:
      1. Smoke
      2. Capacity
      3. Regular load test
      4. Stress
      5. Scalability
      6. Volume
      7. Stability