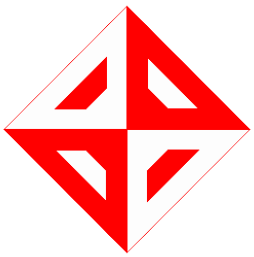

METU - Department of Computer Engineering

SOFTWARE TEST DOCUMENT

WebCat

3 June 2016

Group Name : **Overcode**

Group Members:

Mert Basmacı - 1819119

Onur Ozan Yüksel - 1881663

İzzet Barış Öztürk – 1881796

Özge Donmaz - 1819234

1. Introduction

1.1. DOCUMENT IDENTIFIER

This software test document for Webcat project, which is being developed by group Overcode, is prepared according to the IEEE 829- 2008 Standard for Software and System Test Documentation. Since it is the first test document that describes all the tests performed on Webcat, its version is "1.0". The features of the system are verified and validated through these tests.

1.2. SCOPE

The objectives, possible scenarios, procedural requirements and the outcomes of each and every test case is defined in this document. Whether outcomes of the test cases meet the functional requirements and use-cases predefined in Software Requirements Specification report ascertains the reliability of the system. Consequently,validation and verification procedures are applied and demonstrated here. The WebCat project has a user interface which is defined in SRS and designed in SDD. The scope of this document is the tests of this user interface features.

1.3. REFERENCES

[1] Software Requirements Specification of the project Webcat

[2] Software Design Descriptions of the project Webcat

[3] IEEE. (2008). IEEE Std. 829-2008 IEEE Standard for Software and System Test Documentation. IEEE Computer Society.

1.4. LEVEL IN THE OVERALL SEQUENCE

Three types of tests are applied in overall sequence; iteration test, integration test and full-pass test.

1.5. TEST CLASSES AND OVERALL TEST CONDITIONS

All features that are explained in in Software Requirement Specifications and Software Design Descriptions are tested throughly. For more information you can check section 4.

2. DETAILS FOR SYSTEM PLAN

The following sections describes test items and their identifiers, test traceability matrix, features to be tested, feature not to be tested, approach

to the testing phase, pass/fail criteria for the items and test deliverables.

2.1. TEST ITEMS AND THEIR IDENTIFIERS

The functionalities defined in Software Requirements Specifications and Software Design Descriptions are tested. The test cases are shown below according to these functionalities together with their identifiers.

Test case 1 - Already Categorized

Test case 2 - Not Categorized

Test case 3 - Standard Classification

Test case 4 - Nested Classification

Test case 5 - Repeating Just Categorized

Test case 6 - Crawler Initialization

Test Case 7 - Categorizer Initialization

2.2. TEST TRACEABILITY MATRIX

The test traceability matrix ,that shows the dependencies between the predefined use-cases and the test cases, is shown below. Use cases identifiers are abbreviated as UC-i and those of test cases are done so as TC-i.

	UC1	UC2	UC3	UC4
TC1	✓			
TC2		✓		
TC3		✓		
TC4		✓		
TC5		✓		
TC6			✓	
TC7				✓

2.3. FEATURES TO BE TESTED

The features to be tested are comprised of functionalities in Software Requirements Specifications and Software Design Descriptions documents,which associated to the use-cases above in test traceability matrix.

2.4. FEATURES NOT TO BE TESTED

While implementing the WebCat project, some features, not used in final project, are also implemented. These features are;

i) Gathering URLs which contains any category name using the web crawler,

ii) Creating a doc2vec model using a large corpus(1GB wikipedia pages),

iii) Using created doc2vec model, creating infer vectors instead of bag of words implementation,

iv) Creation of the training model.

Also language detection of a document, Zemberek morphological analyzer class, creating test vector.arff from a URL are not tested.

2.5. APPROACH

Since all members of our team are informed about the process of th project thaks to weekly meetings, we all know about the internal structure and the progress of the project.This enabled us to create and test our own cases. We've also done testing all members together to minimize errors.

2.6. ITEM PASS/FAIL CRITERIA

Both of the success scenarios of use cases and the expected outcome of the test cases determine the outcomes of the tests. The Applied test passes if and only if the outcomes are identical to the expected ones, otherwise, it is a failure.

2.7. TEST DELIVERABLES

This document, test cases, level test cases, reports of passing and failing actions constitute test deliverables.

3. TEST MANAGEMENT

3.1. PLANNED ACTIVITIES AND TASKS/TEST PROGRESSION

In order to check test cases of WebCat Project, all the components described in SDD are implemented and integrated. The database is set and required tables stated in SDD are created before starting test cases. All the test cases is applied sequentially. The classifier needs trained models. Therefore our trained models(standard model, master model, slave model) is provided initially. GUI is started to verify test cases. Test cases described in part 4 is applied initially. In order to pass the Test Case #4, initial seed list of the web crawler should be also provided. Seed list can be added to the database table manually(insert urls into seedList table).

All of the team members are responsible for test management.

Thrown Java exceptions are failure of the any tests.

The returned category may not be the expected category. However that does not indicates the system failure.

Testing resources and requirements are mentioned in 3.2 section.

3.2. Environment / Infrastructure

All test cases have the following environmental needs:

Hardware Needs:

Any computer having at least 10gb free disk space and internet connection.

Software Needs:

Windows Operating Systems (Windows XP or above), Linux Operating System (Specifically centOS), JAVA, Java SE Development Kid.

Recommended Software:

Eclipse or Netbeans

4. Test Case Details

4.1 USE CASE 1: Already Categorized

Test Case Identifier	Test Case #1
Objective	Testing whether the system works with a web page that is already categorized and in the database.
Scenario	User enters a URL that exists in the database table and clicks “Categorize” button
Expected Outcome	GUI shows the page category found from the database without any classification
Special Procedural Requirements	The url's existence should be checked beforehand to be sure the typed url is really in the database.

4.2 USE CASE 2: Needs to be Categorized

Test Case Identifier	Test Case #2
Objective	Testing whether the system asks for classification due to url not being exist in the database.4.2 USE CASE 2: Needs to be Categorized
Scenario	User enter a URL that doesn't exist in the database table and clicks “Categorize” button
Expected Outcome	GUI asks classification method to the user
Special Procedural Requirements	The url's existence should be checked beforehand to be sure the typed url is not in the database.

Test Case Identifier	Test Case #3
Objective	Testing standard classification is done right when user chooses “Standard Classification” choice after TC2's expected outcome
Scenario	When user is asked “Which method would you like to use?” after TC2's expected outcome, the user chooses “Standard Classification”
Expected Outcome	The system starts standard classification function with the url entered as its input. Then returns the result of the classification to user through showing it on gui and lastly adds it to database for future searchs
Special Procedural Requirements	The url's existence should be checked beforehand to be sure the typed url is not in the database. Special print lines in code to be sure the right function is called.

Test Case Identifier	Test Case #4
Objective	Testing nested classification is done right when user chooses “Nested Classification” choice after TC2's expected outcome
Scenario	When user is asked “Which method would you like to use?” after TC2's expected outcome, the user chooses “Nested Classification”
Expected Outcome	The system starts nested classification function with the url entered as its input. Then returns the result of the classification to user through showing it on gui and lastly adds it to database for future searchs
Special Procedural Requirements	The url's existence should be checked beforehand to be sure the typed url is not in the database. Special print lines in code to be sure the right function is called.

Test Case Identifier	Test Case #5
Objective	Testing the system really adds newly found categories to the database for future searchs
Scenario	User enters the same urls used in TC3 and TC4 after they are categorized.
Expected Outcome	GUI shows the page category found from the database without any classification
Special Procedural Requirements	TC3 and TC4 test cases should be tested and resulted in success beforehand.

4.3 USE CASE 3: Crawler Initial Start

Test Case Identifier	Test Case #6
Objective	Testing whether the Crawler component works as it suppose to be or not
Scenario	Starting the crawler, then periodically checking database tables to observe the changes.
Expected Outcome	Crawler starts visiting web pages from seed list and fills the frontier table with new found urls
Special Procedural Requirements	Making sure the frontier table is empty initially and seed list table is filled with known values.

4.4 USE CASE 4: Categorizer

Test Case Identifier	Test Case #7
Objective	Testing whether the Categorizer component works as it suppose to be or not
Scenario	Starting the categorizer, then periodically checking database tables to observe the changes.
Expected Outcome	Categorizer starts categorizing web pages in frontier list and stores them on main table with their newly found categories. As the process goes on the frontier list should get shorter while the main table getting longer
Special Procedural Requirements	- The main table should be filled with known values (just a few rows) - Crawler should be stopped after a working while (before the test starts). This is to observe changes easier.

5. SYSTEM TEST REPORT DETAILS

5.1. OVERVIEW OF TEST RESULTS

All required tests are done and the details are documented in Section 5.2. All tests are18 resulted in success.

5.2. DETAILED TEST RESULTS

Test Case #1	Passed with flying colors!
Test Case #2	Passed with flying colors!
Test Case #3	Passed with flying colors!
Test Case #4	Passed with flying colors!
Test Case #5	Passed with flying colors!
Test Case #6	Passed with flying colors!
Test Case #7	Passed with flying colors!

5.3. RATIONALE FOR DECISIONS

We made sure that the tests we've chosen test all the components and their integrity. The tests covers all possible routes that a user can take. The tests of the parts that doesn't requires a user to function are kept simple but covers the part throughly.

5.4. CONCLUSIONS AND RECOMMENDATIONS

All tests are prepared, done, and observed with all four members of the team to minimize errors. As shown in Section 5.2 all the tests finished with a “Passed” mark. The tests shown in this part are only technical tests. The tests for categorization models and their true classifying ratios are constantly being done while new training urls arrive.