Chapter 6

Software testing of classification of unstructured data from online course web pages

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

6.1 Testing Environment

Testing was done on a DELL laptop, connected to a power supply (not on battery power), with the following specifications:

Hardware Specifications:

- Intel ® Core ™ i5 CPU
- 4.00 GB RAM
- 32 bit OS
- 500GB HDD

Software Specifications:

- Windows 7 OS
- Netbeans IDE

6.2 Unit Testing of Main Modules

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming a unit could be an entire module but is more commonly an individual function or procedure. In object-oriented programming a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

6.2.1 Unit Testing of Crawler Module

The Crawler unit is tested and test cases are tabulated in the tables. Two tests were performed to check whether basic crawler and multiple crawler were working. Basic crawler filters images and other non required information from a web page. It filters out non relevant web pages like search, account, login, rss etc. It checks whether the given url should be crawled be based on a crawler logic. Multiple crawler works with multiple crawlers crawling simultaneously .Both tests were found to have been successful. Tables 6.1 and 6.2 reference these two tests.

Table 6.1: Unit Test Case 1: Checks if Basic crawler works

Sl.no of the test case	Unit test case-1
Name of the test	Test to check if basic crawler works
Featured being tested	Crawler module
Description	Basic crawler filters out irrelevant information from a web page
Sample input	URL
Expected output	.The relevant url
Actual output	Same as above
Remarks	Working

The unit test case 1 is to check whether basic crawler filters images and other non required information from a web page correctly. It checks whether the filtering of irrelevant web pages is happening. This ensures that the crawler is able to bring out only

pages which can be featurized correctly in the required feature string format for the classification process to take place.

Table 6.2: Unit Test Case 2: Checks if multiple crawler works

Sl.no of the test case	Unit Test case-2
Name of the test	Test to check if multiple crawler works
Featured being tested	Crawler module
Description	This class decides which URLs should be crawled and handles the downloaded page
Sample input	A web URL.
Expected output	relevant web page
Actual output	Same as above
Remarks	Working

6.2.2 Unit Test of Featurizer module

The featurizer unit is tested and the test case is tabulated in the following table 6.3. The test case checks whether the feature string is extracted correctly. The features involved in the feature string are URL, the title of the webpage, meta-description, course name, keywords, social tags, provider, price, duration, certificate, instructor, description and level.

The above mentioned test case fails because the featurizer encounters a page of unrecognized file type which may be a multimedia file, document or stylesheet. Therefore modifications are made to the crawler in order to filter out such pages and include only pure HTML pages and remove unnecessary ones. The test is repeated after revision of code below in Table 6.4.

Table 6.3: Unit Test Case 3: Check if featurizer works

Sl.no of the test case	Unit Test case-3
Name of the test	Test to check if featurizer works
Featured being tested	Featurizer module
Description	The featurizer has to return the feature string containing all the HTML page features, social features, URL features, content features.
Sample input	A web URL.
Expected output	The feature string with url, title, description, course name, keywords, social tags, provider, price, duration, certificate, instructor, description, instructor
Actual output	The featurizer returns a java invalid file type exception and does not print the required feature string
Remarks	Not Working

Table 6.4: Unit Test Case 4: Check if featurizer works

Sl.no of the test case	Unit Test case-4
Name of the test	Test to check if featurizer works
Featured being tested	Featurizer module
Description	The featurizer returns the feature string containing all the HTML page features, social features, URL features, content features.
Sample input	A web URL.
Expected output	The feature string with url, title, description, course name, keywords, social tags, provider, price, duration, certificate, instructor, description, instructor
Actual output	Same as above
Remarks	Working

6.2.3 Unit Test of Classifier module

The classifier unit is tested and the test case is tabulated in the following table 6.5. The test case for this module checks if the datasets given as input to the classifier is compatible for the classifier to run.

Table 6.5: Unit Test Case 5: Check if classifier works

Sl.no of the test case	Unit test case-5
Name of the test	Test to check if classifier works properly
Featured being tested	Classifier module
Description	To test whether the the classifier model is trained with appropriate trained dataset. To compare the results of various classification algorithms.
Sample input	A trained dataset.
Expected output	Correctly classified output with good and efficient results
Actual output	Same as above
Remarks	Working

6.2.4 Unit Test of Database module

The database unit is tested and the test case is tabulated in the following table 6.6. The test case for this module checks if relevant results are returned as per the user query. The database here uses a Comma Separated Value (CSV) file as input and allows the user to perform Structured Query Language (SQL) queries on the data in this file and treat it as a conventional database. Commands are issued from the java code interface using an Application Programming Interface (API) similar to Java Data Base Connectivity (JDBC). This open source driver is called CSVJDBC and is linked to the front end by the java code that accepts user query requirements to display results fetched from the data source.

Table 6.6: Unit Test Case 6: Checks if Database module works

Sl.no of the test case	Unit test case-6
Name of the test	Test to check if database module works properly. A csvjdbc driver is used to read the data.
Featured being tested	Database module
Description	CsvJdbc is a read-only JDBC driver that uses Comma Separated Value (CSV) files or DBF files as database tables.
Sample input	A user query
Expected output	Return of relevant results
Actual output	Same as above
Remarks	Working

6.2.5 Unit Test of User Interface module

The user interface unit is tested and the test case is recorded in the following table 6.7. The test case for this module checks if the front end works properly with proper loading. The front end involves options such as search, update and bookmark. The search option enables the user to search for online course by the course name, course provider, instructor, price, level etc. The update option delivers the updated information about the latest online courses available. The bookmark option lets the user to bookmark the courses for later reference.

Table 6.7: Unit Test Case 6: Checks if User Interface module works

Sl.no of the test case	Unit test case-6
Name of the test	Tests to check if user interface module works with all the functionality such as search, update and bookmark.
Featured being tested	User interface module, Database module
Description	To check the functioning of search, update and bookmark function.
Sample input	A user input.
Expected output	Appropriate display of the results.
Actual output	Same as above
Remarks	Working

6.3 Integration Testing Of Modules

In integrated testing, the modules are joined to form sub-systems which each perform a specific function. The performance of the product is determined by its holistic performance. 3 integration tests performed are at the front end for search, update and bookmarks options. table 6.8, 6.9 and 6.10 illustrate the test cases.

Table 6.8: Integrated test case 1: Test case to see if search option in the user interface works

Sl.no of the test case	Integrated test case-1
Name of the test	Tests to check if the search function works fine.
Featured being tested	Search option.
Description	There are 6 types of searches that can be made by the user. The user can search by course, provider, level, instructor, price and certification.
Sample input	A user input according to the required search option.
Expected output	Appropriate display of the results. The display includes the course, provider instructor, level, price, certification and a brief description about the course
Actual output	Same as above
Remarks	Working

Table 6.9: Integrated test case 2: Test case to see if update option in the user interface works

Sl.no of the test case	Integrated test case-2
Name of the test	Tests to check if the update function works fine.
Featured being tested	User interface module, crawler module, featurizer module & classifier module
Description	The update option is to bring the all the new courses introduced in the recent past to the user in a consolidated manner. Upon the selection of the update option the crawler is set off in action to crawl for all the new course webpage urls. Once crawled these pages are featurized which is then given to the trained classifier to classify the webpage.
Sample input	Selection of the update option by the user.
Expected output	Appropriate display of the results. The display includes the course, provider instructor, level, price, certification and a brief description about the course.
Actual output	Same as above
Remarks	Working

Table 6.10: Integrated test case 3: Test case to see if bookmark option in the user interface works

Sl.no of the test case	Integrated test case-3
Name of the test	Tests to check if the bookmark function works fine.
Featured being tested	bookmark option in the user interface module.
Description	The bookmark option lets the user bookmark any course for future reference
Sample input	Selecting the bookmark option for a course by the user.
Expected output	The bookmarked courses get stored in the database and the list of selected course gets displayed once the user chooses to see them. Further the user can view the details of the bookmarked courses individually
Actual output	Same as above
Remarks	Working

6.4 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing fails within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. The entire course portal is tested from start to finish, including testing of all conditions and permutations queries. The test was found to be successful. The Test Case is tabulated in table 6.11.

Table 6.11: System testing

S.no of the test case	System testing
Name of the test	Holistic system test
Featured being tested	User Interface module
Description	The search, update and bookmark option will be tested. Also the system testing checks the display of the results in the appropriate pattern.
Sample input	Selecting search, update bookmark, show all bookmark and clear all bookmarks options.
Expected output	If there is match to the user query in the search option the results are displayed, if there is no match found then a message is displayed intimating about the absence of the course the user is looking for. If the update option is chosen by the user the appropriate results are displayed. If the bookmark option is displayed the selected courses by the user gets stored in the database for future reference. The clear all bookmarks option clears the list of previously bookmarked courses.
Actual output	Same as above
Remarks	Working

The system test is used to ensure the compliance of all modules of the system. It tests to see if the various modules work in synchronization with each other to provide the expected results. If the test is a success, it indicates the application is ready for use.

6.5 Summary

This chapter has detailed, listed and evaluated the testing process of various components in an integrated manner to show successful working of the system under multiple constraints and in various stages of working, as well from the user's point of view to fulfil the characteristics mentioned in the requirements specification.