A FLEXIBLE TEST INTERFACE AND GRADING SERVICE FOR AN
AJAX BASED COURSE MANAGEMENT SYSTEM

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Yunita
Fall 2012
The Undersigned Faculty Committee Approves the

Thesis of Yunita:

A Flexible Test Interface and Grading Service for an AJAX Based Course

Management System

Subrata Bhattacharjee, Chair
Department of Mechanical Engineering

Kris Stewart
Department of Computer Science

Carl Eckberg
Department of Computer Science

Approval Date
Sept 27, 2012
Copyright © 2012

by

Yunita

All Rights Reserved
DEDICATION

This thesis is dedicated to my parents and family for their love and support during my studies. I also dedicated this to my husband for his endless support, company and encouragement to help me strive and reach my goals.
Whether you think you can or think you can’t, you’re right.

- Henry Ford
ABSTRACT OF THE THESIS

A Flexible Test Interface and Grading Service for an AJAX Based Course Management System
by
Yunita
Master of Science in Computer Science
San Diego State University, 2012

The aim of this project is to build a flexible test interface and grading service module for ClassTA Course Management System (CMS). Most CMS has similar features in their test interface and grading module. However, most of them also shared common issues such as how to design a responsive yet easy to use test interface and grading service, how to minimizing the possibility of cheating, and how to enhance students’ learning experience. The newly test interface and grading service are built with consideration to tackle the known issues above.

AJAX is used as a based technology for this project. It is chosen because of its capability to communicate with the server asynchronously and allowing part of the web page to be updated without having to reload the entire page, thus creating a fast and responsive test interface and grading services. Moreover, to make it easy to use, the basic interface uses icons and simple drag and drop feature.

New test interface is developed for students to take the different types of available tests such as In-Class Test, Online Test, and Self-Graded Test. In-Class test is unique feature developed to minimize cheating possibility. It is a type of hybrid test where students take the test in the class and later on submit their answers online. New test interface has options for teacher to display one question in one page or display all questions in a page. To provide a better learning experience for students, it also has features that allowing students to check their grades, review test result and solution, and submit difficulty rating of a question in the test.

The grading service is built for faculties to provide an advance and flexible grading system. With the new grading service, faculties can perform manual grading or automatic re-grading process in many different ways, including options to use or ignore manual changes in re-grading process. New statistics, histogram, and comments for survey and regular test also provided for faculties to track the class performance.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT ................................................................. vi</td>
</tr>
<tr>
<td>LIST OF TABLES ....................................................... x</td>
</tr>
<tr>
<td>LIST OF FIGURES ................................................... xi</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS ............................................... xiv</td>
</tr>
<tr>
<td>CHAPTER</td>
</tr>
<tr>
<td>1 INTRODUCTION ..................................................... 1</td>
</tr>
<tr>
<td>1.1 Course Management System ............................... 1</td>
</tr>
<tr>
<td>1.2 Blackboard and Moodle Comparison .................. 2</td>
</tr>
<tr>
<td>1.3 What is ClassTA ................................................ 5</td>
</tr>
<tr>
<td>1.4 Test Interface and Grading Service in ClassTA .... 6</td>
</tr>
<tr>
<td>2 TECHNOLOGIES USED ............................................. 7</td>
</tr>
<tr>
<td>2.1 AJAX ................................................................. 7</td>
</tr>
<tr>
<td>2.1.1 JavaScript ....................................................... 8</td>
</tr>
<tr>
<td>2.1.2 XHTML ............................................................ 9</td>
</tr>
<tr>
<td>2.1.3 CSS ............................................................... 10</td>
</tr>
<tr>
<td>2.1.4 XMLHttpRequest ............................................. 10</td>
</tr>
<tr>
<td>2.1.5 JavaScript Object Notation (JSON) .................. 11</td>
</tr>
<tr>
<td>2.1.6 Document Object Model (DOM) ....................... 11</td>
</tr>
<tr>
<td>2.2 MySQL ............................................................... 12</td>
</tr>
<tr>
<td>2.3 phpMyAdmin ....................................................... 13</td>
</tr>
<tr>
<td>2.4 Netbeans ............................................................. 13</td>
</tr>
<tr>
<td>2.5 Java Based Web Services ................................. 13</td>
</tr>
<tr>
<td>3 THE CLASSTA PROJECT ........................................ 15</td>
</tr>
<tr>
<td>3.1 ClassTA Desktop ............................................... 15</td>
</tr>
<tr>
<td>3.2 ClassTA Class .................................................. 16</td>
</tr>
<tr>
<td>3.3 ClassTA Help Service ....................................... 16</td>
</tr>
<tr>
<td>3.4 ClassTA Calendar ............................................ 18</td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.5 ClassTA Friend Finder</td>
</tr>
<tr>
<td>3.6 ClassTA Roster</td>
</tr>
<tr>
<td>3.7 ClassTA Quiz</td>
</tr>
<tr>
<td>4  DESIGN AND IMPLEMENTATION OF CLASSTA TEST INTERFACE AND GRADING SERVICE</td>
</tr>
<tr>
<td>4.1 System Architecture</td>
</tr>
<tr>
<td>4.2 Database Architecture</td>
</tr>
<tr>
<td>4.3 Test Interface</td>
</tr>
<tr>
<td>4.3.1 Online Test Interface</td>
</tr>
<tr>
<td>4.3.2 In-Class Test Interface</td>
</tr>
<tr>
<td>4.3.3 Survey Test Interface</td>
</tr>
<tr>
<td>4.3.4 Self Graded Test Interface</td>
</tr>
<tr>
<td>4.4 Grading Service</td>
</tr>
<tr>
<td>4.4.1 Faculty Grading Service</td>
</tr>
<tr>
<td>4.4.2 Student Grading Service</td>
</tr>
<tr>
<td>4.5 Test Statistics</td>
</tr>
<tr>
<td>4.6 Comparative Study of Test Interface and Grading Service in CMS</td>
</tr>
<tr>
<td>5  TESTING</td>
</tr>
<tr>
<td>5.1 Faculty Interface</td>
</tr>
<tr>
<td>5.1.1 Online Test</td>
</tr>
<tr>
<td>5.1.1.1 Creation of an Online Test</td>
</tr>
<tr>
<td>5.1.1.2 Updating an Online Test</td>
</tr>
<tr>
<td>5.1.1.3 Checking Online Test Progress</td>
</tr>
<tr>
<td>5.1.1.4 Re-Grading Online Test</td>
</tr>
<tr>
<td>5.1.1.5 Updating Online Test Score in Roster</td>
</tr>
<tr>
<td>5.1.1.6 Updating Online Test Question’s Points in Result Interface</td>
</tr>
<tr>
<td>5.1.1.7 Reviewing Online Test Statistics and Rating</td>
</tr>
<tr>
<td>5.1.2 In-Class Test</td>
</tr>
<tr>
<td>5.1.2.1 Creation of an In-Class Test</td>
</tr>
<tr>
<td>5.1.2.2 Updating, Re-Grading, and Reviewing Result of an In-Class Test</td>
</tr>
<tr>
<td>5.1.3 Self-Graded Test</td>
</tr>
<tr>
<td>5.1.3.1 Creation of a Self-Graded Test</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1.1. Student Overall Preference .................................................................4
Table 2.1. Result of Traditional VS AJAX ..........................................................7
Table A.1. Structure of quiz_live Table ..............................................................63
Table A.2. Structure of quiz_result Table ..........................................................64
Table A.3. Structure of question_statistics Table .............................................65
LIST OF FIGURES

PAGE

Figure 1.1. Comparison between the top CMS for 2011/2012 provided by EduTools. ............2
Figure 1.2. Top 15 LMS products used in 2009. .................................................................3
Figure 1.3. Faculty overall preference. .................................................................................4
Figure 1.4. Faculty assessment and grading tools preference .................................................5
Figure 2.1. How AJAX works .............................................................................................8
Figure 2.2. XHTML basic structure ....................................................................................9
Figure 2.3. JSON object representation .............................................................................11
Figure 2.4. JSON array representation ..............................................................................11
Figure 2.5. A table in XHTML .........................................................................................12
Figure 2.6. DOM graphical representation of a table .........................................................12
Figure 2.7. NetBeans IDE ...............................................................................................14
Figure 3.1. ClassTA login page. .......................................................................................15
Figure 3.2. Faculty desktop in ClassTA ...............................................................................16
Figure 3.3. Student desktop in ClassTA .............................................................................16
Figure 3.4. Faculty account - view of a class desktop. .........................................................17
Figure 3.5. Student adding a new class ..............................................................................17
Figure 3.6. ClassTA help page ...........................................................................................17
Figure 3.7. ClassTA calendar ............................................................................................18
Figure 3.8. ClassTA Friend Finder ....................................................................................18
Figure 3.9. Faculty roster ..................................................................................................19
Figure 3.10. Gradable editor – compose/edit questions step. ............................................20
Figure 3.11. Question bank ...............................................................................................21
Figure 3.12. Gradable editor – set/update rules step ..........................................................21
Figure 3.13. Gradable editor – deploy/un-deploy step ......................................................21
Figure 4.1. Test interface and grading service system architecture .....................................23
Figure 4.2. Student enters gradable code .........................................................................24
Figure 4.3. Header of an online test interface. .................................................................25
Figure 4.4. Submission confirmation with warning message upon clicking “Submit All and Finish” .................................................................25
Figure 4.5. Online test - display all questions in one page .................................................................26
Figure 4.6. Confirmation window from “Submit” ..............................................................................26
Figure 4.7. Online test - display one question in one page ...............................................................27
Figure 4.8. Sample of pdf questions paper for in-class test ............................................................28
Figure 4.9. Bubble sheet for in-class test .........................................................................................28
Figure 4.10. In-Class test interface ..................................................................................................29
Figure 4.11. Survey test interface ....................................................................................................29
Figure 4.12. Setting up weight for survey choices ............................................................................30
Figure 4.13. Self-graded test interface ............................................................................................31
Figure 4.14. Override score from roster ..........................................................................................32
Figure 4.15. Override score from test result .....................................................................................32
Figure 4.16. Student account – "My Grade" icon ............................................................................33
Figure 4.17. Student roster view in "My Grade" .............................................................................33
Figure 4.18. Student view online test result ....................................................................................34
Figure 4.19. Question difficulty and success statistics display .......................................................35
Figure 4.20. Survey result and comments display .........................................................................36
Figure 4.21. Survey statistics display .............................................................................................36
Figure 4.22. Student review test submission page in Blackboard – "My Grade" ............................38
Figure 4.23. Student view of a test result in Moodle .........................................................................38
Figure 4.24. Student view test result in ClassTA ............................................................................39
Figure 5.1. Roster view with the newly deployed test column ..........................................................41
Figure 5.2. Warning message when updating rules of a live online test .........................................41
Figure 5.3. Alert when trying to add question to a live test ...............................................................42
Figure 5.4. Warning when withdrawing live test with students .......................................................42
Figure 5.5. Roster contains latest updates from all tests .................................................................43
Figure 5.6. Question detail with the original points .........................................................................43
Figure 5.7. Question detail with the updated points ........................................................................43
Figure 5.8. Updated roster after re-grade is performed ....................................................................44
Figure 5.9. Roster view after re-grade ignore manual is performed ................................................44
Figure 5.10. Updated roster after the manual overrides ..................................................................44
Figure 5.11. Roster view before single question point is overridden. .....................................45
Figure 5.12. Test result display with updated final points. ..................................................45
Figure 5.13. Roster view after single question point is overridden. ........................................45
Figure 5.14. Statistics and difficulty rating display in test editor. ..........................................46
Figure 5.15. Sample of in-class question paper. .................................................................47
Figure 5.16. Statistics and survey comments display in test editor. ......................................49
Figure 5.17. Error message when gradable code is invalid. ..................................................50
Figure 5.18. Error message when the gradable is already closed. .........................................50
Figure 5.19. Error message when the gradable is not open yet. .............................................50
Figure 5.20. Error message when student already completed and submitted the test. ............51
Figure 5.21. Warning message when student is late but still allowed taking the test. ..........51
Figure 5.22. Test interface display when gradable code is valid. ..........................................51
Figure 5.23. Online test first attempts. ................................................................................52
Figure 5.24. Online test second attempts. ...........................................................................52
Figure 5.25. Student grade view with “Immediate” access in roster. .....................................53
Figure 5.26. Student grade view with “Delayed” access in roster ........................................53
Figure 5.27. Student grade view with “Overall Stats Only” access in roster .........................54
Figure 5.28. Student grade view with “Disabled” access in roster. ......................................54
Figure 5.29. Test result and difficulty rating display. ...........................................................54
Figure 5.30. In-class test interface. ....................................................................................55
Figure 5.31. Self-graded test interface. ................................................................................56
Figure 5.32. Survey interface. ..........................................................................................58
ACKNOWLEDGEMENTS

I want to express my sincere thanks and gratitude and to my thesis committee member: Professor Subrata Bhattacharjee, for his time, help, and guidance throughout this project, Professor Carl Eckberg and Professor Kris Stewart for their support and willingness to take the time to become my thesis committee. Also to Professor Alan Riggins and Professor Roger Whitney who taught me important courses which are very useful for the completion of this project.
CHAPTER 1

INTRODUCTION

The birth of the Internet in the early 60s has transformed our world like never before. In modern times like nowadays, the needs to be able to access information at anytime from anywhere has become one of the necessities in life. Along with this, the technology has evolved dynamically over the years to create a faster, user friendly, and secure web based application.

Education is one of the fields that used technology to enhance the traditional classroom experience. It used to be students who must physically go to school. Now, school can virtually come to students in the form of online classes. With the increase popularity of online classes, the needs for better online course management system begin to emerge.

1.1 COURSE MANAGEMENT SYSTEM

At its simplest, a course management system (CMS) is a system used by instructor to post information on the web without having to know or understand any programming language. CMS provides instructors with a set of tools and framework that allows the relatively easy creation of online course content. Subsequently it supports teaching and management of that course including various interactions with students taking the course [1].

The needs to provide access for students to be able to learn by them self anytime, anywhere, not limited to the traditional teacher-student classroom interaction, has made the CMS an integral part of the new learning experience. CMS help made it easier for school to offer online courses to anyone regardless where they live.

According to Gilfus Education Group, an independent education consulting company, some of the top CMS for 2011/2012 [2] are Adrenna Academic [3], Blackboard [4], Desire2Learn [5], Moodle [6], and Sakai Project [7]. The comparison between these CMS as shown in Figure 1.1 [8] can be generate from EduTools [8], a website that provides independent reviews, side-by-side comparisons, and consulting services to assist decision-making in the e-learning community.
Another research conducted by eLearning Guild shows that Moodle, with 20.1%, is the top CMS product used in 2009. Meanwhile Blackboard is in the second place with 13.1% followed by Total LMS with 10.9% [9]. More detail on the top 15 CMS products can be seen in Figure 1.2 [9].

### 1.2 Blackboard and Moodle Comparison

Blackboard is a CMS comprises of the following modules [10]:

- A learning system that provides online course delivery and management for institutions.
- A community and portal system for use in creating online campus communities.
A content management system for centralized control over course content.

A system to record and analyze student assessment results.

Moodle (Modular Object-Oriented Dynamic Learning Environment) is an open source CMS that has many free third-party plug-ins. Moodle users can use PHP to author and contribute new modules [11]. The cost and flexibility are some of the advantages of Moodle compares to Blackboard. However on the other side, Blackboard has a more polished appearance and better grade book than Moodle.

In spring 2009, Gilfus Education Group performed an evaluation for University of North Carolina – Charlotte to compare Moodle Learning Management System with the University’s current Blackboard system (Vista) [12]. The evaluation conducted a survey
among 313 students and 39 faculties (30 with prior Blackboard experience, 9 without) to compare 12 different aspects such as: Assessment and Grading Tools, Communications Tools, Interface, Organizational Tools, Ease of Use, Compatibility, and Reliability.

The overall result from students, as shown in the Table 1.1 [13], is 80.5% prefer to take courses using Moodle rather than Blackboard Vista. As for the faculties, overall result shown on Figure 1.3 [14] concludes they also strongly prefer Moodle than Blackboard. For the Assessment and Grading Tools feature, more faculties also prefer Moodle than Blackboard (see Figure 1.4) [14].

Table 1.1. Student Overall Preference

<table>
<thead>
<tr>
<th>If you were given the option to take courses using either Moodle or Blackboard Vista, which would you choose?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Moodle</td>
<td>80.5%</td>
</tr>
<tr>
<td>Blackboard Vista</td>
<td>19.5%</td>
</tr>
</tbody>
</table>


Q16: Overall Preference

### 1.3 What is ClassTA

Most CMS has similar features and also shared common problems too. It is not easy to choose and implement a CMS. Often times, it takes a great amount of time and funds. One of the important issues faced by any CMS is how to design a responsive yet easy to use system. Other problem raised in the use of CMS is how to minimize cheating when students taking an online test. ClassTA was originally designed and built with consideration to improve and solve the known problems above.

ClassTA [15] is a Course Management System built with AJAX technologies. AJAX (Asynchronous JavaScript and XML) is a way to use standard web technologies to allow asynchronous request without having to reload the whole web page. Thus, makes ClassTA a responsive application. The basic appearance of ClassTA is designed as a desktop with an easy to use drag and drop interface. ClassTA has many modules such as Quiz Builder, Grading Service, Calendar, and Help Service. These modules will be explained briefly in Chapter 3.
1.4 Test Interface and Grading Service in ClassTA

One of the important functions of ClassTA is to let students take the test and review the result. Test interface is built to serve this function. However, the old version of test interface in ClassTA has limitations such as:

- All questions in the test will be shown in one page. There is no option to select whether to display one question in one page or display all in one page.
- Questions must be submitted all at once, there is no option to submit a single question and/or submit all at once.
- Instruction is placed in the front page where students enter the submission code, once the test is started, students have no way to view the instruction again.
- If permitted, after test is submitted students can only view their scores but they cannot review the questions and answers of the test. It makes it difficult to reconcile if students thought there is any discrepancy with their test scores.
- No difficulty rating system to know what students think about the difficulty level of a question.

The current version of ClassTA has a grading service which basically performs the automatic grading based on the correct answers and points set by faculty. It allows faculty to override total grades directly from the roster. If there is any mistake in the test, faculty does not have an option to fix the test and re-grade the test result. To improve the grading service and make it more flexible, it needs to have the following function:

- If there is any mistake in the test, faculty should be able to correct the mistake and perform re-grade.
- Ability to override the total scores and the score for a single question, especially for a descriptive type question.
- Faculty needs a place to enter comments or explanation of any manual scores.
- Option to ignore or include manual score in re-grading process.
- Need a view to show the test statistics and difficulty rating.

The purpose of this thesis is to improve and enhance the current version of test interface and grading service in ClassTA. More detail of this project will be discussed in Chapter 4.
CHAPTER 2

TECHNOLOGIES USED

ClassTA is built with AJAX technologies. It uses JavaScript on the client side and JAVA on the server side. For database, ClassTA uses MySQL RDBMS along with phpMyAdmin for the administration tool.

As an AJAX application, ClassTA is browser and platform independent. AJAX is chosen for the development because of its state-of-the-art way to communicate with the server and allowing part of the web page to be updated asynchronously without having to reload the entire page, thus increasing the performance and responsiveness.

In 2005, Alexei White from Developer.com conducted a study to compare the performance of application built using a traditional web model and AJAX web model. Based on his test, he concluded that AJAX application has a better performance in term of number of bytes transferred and times. The detail result can be seen in Table 2.1 [16].

### Table 2.1. Result of Traditional VS AJAX

<table>
<thead>
<tr>
<th></th>
<th>Traditional (Average)</th>
<th>AJAX (Average)</th>
<th>Performance Increase</th>
<th>Performance Increase (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes Transferred:</td>
<td>1,737,607</td>
<td>460,799</td>
<td>1,276,809</td>
<td>73%</td>
</tr>
<tr>
<td>Time (seconds):</td>
<td>115</td>
<td>78</td>
<td>36</td>
<td>32%</td>
</tr>
<tr>
<td>Estimated Transmission time to US West Coast (56k) (seconds):</td>
<td>293.45</td>
<td>94.44</td>
<td>199.01</td>
<td>68%</td>
</tr>
</tbody>
</table>


2.1 AJAX

AJAX (asynchronous JavaScript and XML) is a modern approach of using various existing technologies to create a faster web application without having to refresh the whole page. In general, AJAX incorporates a combination of: JavaScript to make a call to the server
side asynchronously using XMLHttpRequest object, Document Object Model (DOM) to interact and display the information received, XHTML and CSS to style the content of the web page, and JSON/XML or any other format for transferring data. Figure 2.1 [17] show the process of how AJAX is working.


### 2.1.1 JavaScript

JavaScript is a scripting language designed to create an interactive web page. According to World Wide Web Consortium (W3C), a script is a program code that does not need pre-processing (e.g. compiling) before being run [18]. Originally invented by Brendan Eich at Netscape, JavaScript has appeared in all browsers since 1996 [19]. The lightweight, dynamic, and cross browser feature has made JavaScript widely use in web development.

Even though the prefix name is JAVA, JavaScript is not related to JAVA. The two are similar in some ways but completely different in other ways. JavaScript is a weakly-typed language, unlike JAVA, JavaScript allow implicit conversion between different data type. JavaScript does not use a common class-based object oriented model like JAVA, instead it uses a prototype-based model and provides dynamic inheritance where properties and methods can be added dynamically to an object at any time.

JavaScript also fully compliant with ECMAScript language specification, which is a standardized scripting language establish by European Computer Manufacturing Association.
(ECMA) International, an international standards association for information and communications technologies.

2.1.2 XHTML

XHTML is a well-formed markup language used to format the content of web pages. It is basically a modified, stricter HTML that conforms to the XML syntax standard. The most important differences between XHTML and HTML are [20]:

- XHTML elements must be properly nested.
- XHTML elements must always be closed.
- XHTML elements must be in lowercase.
- XHTML documents must have one root element.

Figure 2.2 [21] shows an example of a basic XHTML document. The basic structure of XHTML document consists of:

- DOCTYPE (Document Type) Definition: DOCTYPE specifies the format of XHTML used for the document. Three types of DOCTYPE are Strict, Transitional, and Frameset. Strict emphasizes more on structure than presentation and not allowing deprecated elements and presentational attributes. Transitional includes all elements in Strict plus presentation and deprecated elements. Frameset is a variant of Transitional that allows the use of frames.

- Head Section: The head section encloses meta-data information such as title, author, style sheets, and scripts.

- Body Section: The body section contains the content of the web page such as texts, images, tables, and paragraphs.

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    </body>
</html>
```

2.1.3 CSS

CSS (Cascading Style Sheet) is a language used to define the style of a web page. The concept of CSS is to separates content from the presentation. Many web pages mix both content and presentation together. As a result of this, it becomes difficult to modify and maintain the web pages. With CSS, style can be edited and applied easily to different pages as needed.

The fundamental syntax of a CSS consists of a list of rules defined by selector and declaration block where each declaration contains pair of property and value as followed:

```
selector1 [, selector2, selector3, …]  {
    property1: value1;
    [ property2: value2; …]
}
```

CSS can be added to XHTML page in various ways, some of the most commonly used are:

1. Inline CSS: CSS is added directly to an XHTML tag using a “style” attribute. Using inline CSS will mix the content with presentation and making it hard to maintain.
2. Embedded CSS: CSS rules added to the head section of an XHTML page. The embedded CSS only applied to a single page where the CSS added.
3. External CSS: CSS is saved to its own external file and linked to a page using `<link>` tag in head section. The external CSS will be applied to a whole website, makes it easier to change a style for the entire website.
4. Imported CSS: CSS file is attached within the `<style>` tag or in another CSS file using the `@import` statement. Import CSS works similar like external CSS but it allows importing a CSS into another CSS. By using import, a page can use multiple style sheets.

2.1.4 XMLHttpRequest

XMLHttpRequest is an object use to communicate between client and server. The advantage of using this object is the capability to submit and retrieve information without having to reload the entire web page. The basic steps to use XMLHttpRequest are: create an instance of the object, send the request to server, and process the response from server.

The request made using XMLHttpRequest can be asynchronous or synchronous. In best practice, the call should be asynchronous to let the browser and user continues working while waiting for the callback response.
2.1.5 JavaScript Object Notation (JSON)

JSON is a simple, light-weight format to exchange data between client and server. It uses JavaScript syntax for describing data objects. JSON parsers and JSON libraries exist for many different programming languages [22].

The structure of a JSON object is start and end with a curly bracket where each data is expressed in a name/value pair enclosed in a double quote and separated by colon. For example: {“name”: “value”}. Figure 2.3 [23] describe the structure of a JSON object.

![JSON object representation](http://www.json.org/)


JSON Array is a collection of JSON object separated by a comma. Its structure begins with a left bracket and ends with a right bracket. See Figure 2.4 [23].

![JSON array representation](http://www.json.org/)


2.1.6 Document Object Model (DOM)

DOM is a standard representation of XHTML elements and XML documents that provides an interface for modifying and accessing the elements using a scripting language. DOM converts elements of a web page into an object. These objects build a hierarchy or tree that reflects the structure of the web page itself. Figure 2.5 [24] show a sample of a table in XHTML and Figure 2.6 [24] show the DOM graphical representation of the example table.
2.2 MySQL

MySQL is an open source relational database management system (RDBMS) owned by Oracle. Its open source and multi-platform characteristic has established MySQL status as one of the most commonly used database server in existence. Many of the world's largest and fastest-growing organizations including Facebook, Google, Adobe, Alcatel Lucent and
Zappos rely on MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software [25].

2.3 **phpMyAdmin**

phpMyAdmin is a cross platform, open source database tool for administering MySQL via web browser. Some of the features that phpMyAdmin provides are [26]:

- Intuitive web interface.
- Support for most MySQL features.
- Import data from CSV and SQL.
- Export data to various formats.
- Administering multiple servers.
- Creating PDF graphics of your database layout.
- Creating complex queries using Query-by-example (QBE).
- Searching globally in a database or a subset of it.
- Transforming store data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link.

2.4 **NetBeans**

NetBeans is a Java based open source integrated development environment (IDE) for software development. Netbeans supports several languages including Java, C/C++, XML, HTML, PHP, Groovy, Javadoc, JavaScript, and JSP. It can be extended to support other languages [27]. The base IDE provides functionality such as source code editor, database integration, source version control, and builds automation tools. Figure 2.7 shows how the NetBeans IDE screen looks like.

ClassTA test interface and grading services use Java and JavaScript module in NetBeans as the source code editor. It also uses GlassFish module for web service deployment purpose. Another handy feature is the Subversion module which is used as ClassTA source code version control.

2.5 **Java Based Web Services**

According to W3C, Web Services are application components that communicate using open protocols and designed to support interoperable machine to machine interaction over a network. ClassTA uses Java API for XML Web Services (JAX-WS) to create a
Figure 2.7. NetBeans IDE.

SOAP-based Web Service. SOAP (Simple Object Access Protocol) provides a way to communicate between applications via HTTP. SOAP was designed in 1998 by Dave Winer, Don Box, Bob Atkinson, and Mohsen Al-Ghosein for Microsoft. SOAP version 1.2 became a W3C recommendation on June 24, 2003 [28].
CHAPTER 3

THE CLASSTTA PROJECT

ClassTA is an AJAX based application for Course Management System. This chapter provides an overview of ClassTA project by describing all the different main modules within ClassTA applications.

3.1 CLASSTTA DESKTOP

In order to use ClassTA, every user must have an account. There are two types of account in ClassTA: student account and faculty account. New user can simply use the “Sign Up” button to create a new account. Figure 3.1 shows the ClassTA login page.

Figure 3.1. ClassTA login page.

Upon a successful login, user will be presented with a student or faculty desktop view. The desktop consists of a set of icons that can easily managed by drag and drop function. As a faculty, user can creates a new class, archives old classes, sends email, and deletes unused items from the desktop. Meanwhile as a student, user can only add a new
class and deletes items. The desktop for faculty account and student account is shown in Figure 3.2 and Figure 3.3.

![Faculty desktop in ClassTA](image)

**Figure 3.2. Faculty desktop in ClassTA.**

![Student desktop in ClassTA](image)

**Figure 3.3. Student desktop in ClassTA.**

### 3.2 ClassTA Class

In ClassTA, faculty has access to create and administer class. Each class is represented with an icon in faculty desktop. The display of class desktop for faculty is shown in Figure 3.4. After creating a new class, faculty can add students’ information to the class roster. To join a class, student selects the “New Class” icon on ClassTA desktop (see Figure 3.5) and enters the add code given by Faculty.

### 3.3 ClassTA Help Service

ClassTA includes a help service (see Figure 3.6) to assist and familiarize user with its features. Different topics such as how to get started, how to use the desktop, and information on ClassTA security are provided in the help menu along with videos and demos.
Figure 3.4. Faculty account - view of a class desktop.

Figure 3.5. Student adding a new class.

Figure 3.6. ClassTA help page.
3.4 **ClassTA Calendar**

ClassTA has an integrated calendar feature, as shown in Figure 3.7, which organizes all events and important announcements in one place. The calendar is designed to provide an interactive and easy to use interface for both faculties and students. It also has a color code function to distinguish important events or days.

![ClassTA Calendar](image)

**Figure 3.7. ClassTA calendar.**

3.5 **ClassTA Friend Finder**

Friend Finder function is inspired by the rise of social networking in today’s world. It builds to provide interface for students to search friends and study partners based on certain criteria. Basic search criteria on Friend Finder can be seen in Figure 3.8.

![ClassTA Friend Finder](image)

**Figure 3.8. ClassTA Friend Finder.**
3.6 CLASSTTA ROSTER

Roster has a very important function in ClassTA class. In roster, faculty can do various tasks such as: view and track student’s performance, manage mailing list and send email, get the current statistics and histogram, and perform manual grading. When a new resource is deployed, a new column representing the resource will be added to roster. An example of roster in ClassTA is shown in Figure 3.9.

![Faculty roster](image)

Figure 3.9. Faculty roster.

3.7 CLASSTTA QUIZ

ClassTA quiz builder is a module where faculty creates a test and sets all the rules and attributes. It also integrated with a quiz bank management system that allows faculty to search and export questions from one test to another.

Quiz builder is related with the test interface and grading service. Data from a deployed gradable is used by test interfaces to render and display the test for students. Later on, the data also used by grading services to calculate scores and re-grade if necessary.

The first main component of ClassTA quiz module is a question. A question is a basic object that builds a quiz. Question object contains all information needed to represent different type of question. Each question may have attributes such as answers, body, penalty, points, schematic, solution, tags, and tolerance. Based on the answer types, there are seven types of question in ClassTA:

1. Fixed Answer, the answer to this question is a single fixed number with tolerance.
2. Multiple Choices Single Answer, there is only one correct answer from all the
provided options.
3. Multiple Choices Multiple Answers, there are more than one options to make up the
correct answers.
4. Descriptive Answer, the answer is a descriptive or essay.
5. Self-graded, the answer is a percentage of completion chooses by student themselves.
6. Multi Part Parent, a parent for multiple part question that contains question body but
has no answer.
7. Survey Question, similar with multiple choices where students are given a set of
options to select from but there is no right or wrong answer.

The second component of quiz builder module is a gradable. A gradable is an object
consists of one or more questions and binds together by a set of rules. ClassTA support four
types of gradable: Online, In-Class, Survey, and Self-Graded. More detail about these
different types of gradable will be covered in the next chapter.

In ClassTA, a new gradable is created in three steps:
1. First step is to compose/edit questions (see Figure 3.10). ClassTA quiz builder
module is integrated with a quiz bank. Faculty can create a new question from scratch
or export it from quiz bank (see Figure 3.11).
2. The second step is to set or updates gradable rules. In this step, faculty sets the quiz
attributes such as quiz type, test duration, submission dates, and set quiz
randomization rules (see Figure 3.12).
3. The last step is to deploy the quiz (see Figure 3.13). Once quiz is deployed, a column
represent the quiz name will be added to roster.

![Figure 3.10. Gradable editor – compose/edit questions step.](image)
Figure 3.11. Question bank.

Figure 3.12. Gradable editor – set/update rules step.

Figure 3.13. Gradable editor – deploy/un-deploy step.
CHAPTER 4

DESIGN AND IMPLEMENTATION OF CLASS TA
TEST INTERFACE AND GRADING SERVICE

The test interface and grading service is built to enhance the existing function in ClassTA. The old version of test interface basically only serves as an interface that allows students to take a test and always displays all questions in one page. Moreover, there is no feature provided to let students review their answers. These limitations initiate the need to develop a new and better test interface. As for the old grading service, what it does is perform automatic grading based on the correct answers and points. No re-grade function is provided and there is no way for faculties to override score of a single question. Faculties also need a display to easily view statistics and overall class performance. That is why ClassTA needs a new grading service to improve and solve the issues above. The concept of the new test interface and grading service is based on how to create a more functional, flexible and interactive module while still holding on to a principle of a simple and user friendly application.

4.1 SYSTEM ARCHITECTURE

The architecture of test interface and grading Service in ClassTA is shown in Figure 4.1. It follows the basic architecture of an AJAX application adapted from Iron Speed website [29]. The client scripting on presentation layer of test and result interface is done by BubbleQuizS.js and BubbleResultS.js. Web service calls and information exchanged is done by Gradable.js which contains XMLHttpRequest object. Server side programming is handle by Grader.java which responsible for all transactions on the database.

4.2 DATABASE ARCHITECTURE

ClassTA test interface and grading service use the following tables to store all the information needed:

- quiz_live: This table is used by online test as a placeholder to temporarily store the progress for each student before they completed the test. When the test is completed,
data from this table will be transferred to quiz_result table. By the time an online test is finalized, data for that particular test will be deleted from this table.

- quiz_result: This table is used to store the student result detail for all types of test.
- question_statistics: This table is used to store question statistics and difficulty ratings.

The detail structure of these tables can be seen in the Appendix, Table A.1 - A.3.

### 4.3 Test Interface

Test interface is an interface used by students to take different types of test. The new test interface is designed to be a simple and self-explanatory interface, yet it will provide students with important information needed.

Faculty creates and deploys test from gradable editor. Each test has a unique submission code. Students will log in to their ClassTA account and use the “Gradable” icon on the desktop to access the test interface.
To take the test or survey, students must enter the correct submission code given by faculty (see Figure 4.2). If the submission code is valid and students have not completed the test, a question display panel contains all the questions in the test will be presented. On the other hand if the code is invalid, error message will be shown. The error message will inform students about what might cause the problem. Once students start working on a test, the roster’s cell for that particular students and test will be updated with a message “quiz is in progress” to acknowledge faculty about students’ attempt.

![Figure 4.2. Student enters gradable code.](image)

ClassTA support four different types of test interface: Online, In-Class, Survey, and Self-Graded. Each type will be explained more detail in each section below.

### 4.3.1 Online Test Interface

Online test interface is designed for students to take an online test. Each online test has a time window that determines when students are allowed to take the test. Once a question is submitted, it will be removed from the display and students are not allowed to modify their answer anymore.

In a case where students have to quit before the test is done, they can try login again and continue the test. However, the remaining time will be calculated from their first attempt and they can only work on the remaining un-submitted questions. Faculty may allow students to take the test after the time is up but their late duration will be recorded. If time is up and late submission is not allowed, the test will automatically be submitted.
The display panel for online test consists of two parts, the header and the detail. 

Figure 4.3 shows the header of an online test. Some of the important elements on the header are:

- **Hide/Show Instructions**: used to hide/show the test instructions row. In the new test interface, the instruction is viewable for the whole period of test. Students can use this option to show or hide it as needed.

- **Hide/Show All Details**: used to hide/show details such as answer table and schematic in all questions. This option is useful in particular when students need to focus only on the questions and hide all the supporting details first.

- **Hide/Show Timer**: used to hide/show the timer display.

- **Submit All and Finish**: used to submit all the questions and finished the test. Upon clicking this button, a confirmation window as shown in Figure 4.4 will be displayed. The confirmation window will warn students if there is any blank or invalid answer. Students can go back and fix their answer if they still have time.

- **Total Question**: used to inform students about total number of questions in a test.

- **Total Submitted**: used to inform students about total number of questions have been submitted so far.

- **Total Points**: used to inform students about the total maximum points they can get in the test.

![Figure 4.3. Header of an online test interface.](image)

![Figure 4.4. Submission confirmation with warning message upon clicking “Submit All and Finish”.](image)
All the questions in the test will be shown in the detail section of the test interface.

ClassTA supports two options to display the questions for online test:

1. **Display all questions in a single page**: With this option, the entire questions in the test will be shown in one page (see Figure 4.5). Each question has a submit button in which students can submit a single question. When clicking the submit button, a confirmation window with students’ answer will be displayed (see Figure 4.6). It also has a Hide/Show Details link to hide/show answer table, notes, schematic, and submit button for each question. A timer counting down the test duration is located on the bottom right corner.

2. **Display one question in one page**: When choosing this option, there is only one question displayed in each page. Each page will has navigation tools (such as previous and next link) to move between each question. Similar to display all questions at one page option, it also has Submit button, Hide/Show Detail, and timer on the bottom right corner. Each time student submit their answer the same confirmation windows (see Figure 4.7) is shown. Once a question is successfully submitted, it will be removed from the display and the page automatically goes to the next available question.

![Figure 4.5. Online test - display all questions in one page.](image)

![Figure 4.6. Confirmation window from “Submit”](image)
In-Class Test Interface

In-class test is one of the unique features provided in ClassTA. It develop with a consideration to minimize a cheating attempt when students taking a test online. In-class test is a combination of paper based test and online test. It is a slight variation from traditional in class test where students usually take a paper based test. The different is, along with the questions paper, students are given bubble sheet with a unique submission code to record their answers and submit it later online into ClassTA.

When faculty creates an in-class test, based on the randomization rules, ClassTA will generate different sets of questions paper in PDF format as shown in Figure 4.8. Each student will get different submission code and might get different sets of questions and option in multiple choice type questions. They take the test in the class and later on submit their answers online. When submitting the answers online, students can only see the answer sheet similar to their bubble sheet. The test’s questions will not be displayed again.

In-Class test interface is used by students to submit the answers of an in-class test they have already took in the class. The header for in-class test is similar with online test but it does not have “Hide/Show Details” link and “Submit All and Finish” button. The details part is also a simplified version of online test interface. No question is shown in the in-class test interface. It will only show the bubble sheet similar to what students get in the class. Students then enter their answers and submit it to ClassTA. Figure 4.9 show bubble sheet for in-class test and Figure 4.10 show the in-class test interface with similar layout.
Name: __________________________  First Letter of Your Last Name: ___

ME-520, Quiz-1; Duration 45 min
Submission Code: 3Fk7d

Instruction:

1. Select one correct answer for the questions about HTML below. (Points: 20 – 10 + 10)
   a. What tag use to create list with no order? (Points: 10)
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>dl</td>
<td>ol</td>
<td>ul</td>
<td>li</td>
</tr>
</tbody>
</table>

   b. What tag use to create item in the list? (Points: 10)
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>dl</td>
<td>ol</td>
<td>ul</td>
<td>li</td>
</tr>
</tbody>
</table>

2. Select one correct answer for the questions about Javascript below. (Points: 20 – 10 + 10)
   a. \( x = 1 + "1" \) document.write (x) (Points: 10)
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>2</td>
<td>11</td>
<td>undefined</td>
</tr>
</tbody>
</table>

   b. Inside which HTML element do we put the JavaScript? (Points: 10)
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>scripting</td>
<td>script</td>
<td>javascript</td>
<td>js</td>
</tr>
</tbody>
</table>

Figure 4.8. Sample of pdf questions paper for in-class test.

Figure 4.9. Bubble sheet for in-class test.
4.3.3 Survey Test Interface

As its name suggest, a survey test interface is used to perform a survey to gather data from users. The display is similar to online class but more simplified. All survey questions always shown in one page (see Figure 4.11). There is no option to only show a single question in each page. In addition to that, unlike any other type of test interface, the survey interface has a comment box on the bottom of the page where students may enter the overall comments regarding the survey.
When designing a survey questions, faculty may put weight for each available choices. To add weight for a particular survey option, simply enters it inside a bracket behind the option text (see Figure 4.12). This weight later on will be used to calculate survey result and displaying the survey statistics diagram.

![Figure 4.12. Setting up weight for survey choices.](image)

Survey has a confidential characteristic. Upon submitting a survey, faculty can only see the total results and overall comment. No detail for each student result will be provided by ClassTA.

### 4.3.4 Self Graded Test Interface

Self-graded interface is used to take a self-graded test which is similar to a take home test or homework. The interface is a simplified version of an online test interface (See Figure 4.13). In self-graded test, students will select the percentage of completion for each question. Later on in grading process, the percentage will be used to calculate their grade. All questions in this test have to be a self-graded questions type.

### 4.4 Grading Service

New grading service is designed to create a flexible and easy to use feature for faculty to perform a grading or re-grading process. When students completed a test, ClassTA grading service will automatically grade it based on the rules set for each question. However if it is a survey, no grading process needs to be done. Grading service only collects the result for statistical purpose which later on will be presented in a form of histogram.
Figure 4.13. Self-graded test interface.

Grades are posted to roster for faculty to see. Faculty has control to set the permission which allowed student to view their grade from “My Grade” menu. Faculty also has options to manually override the grades.

4.4.1 Faculty Grading Service

Grades will be posted to the appropriate roster. In roster, faculty has option to calculate grades by combining multiple columns and set the weight for each column. ClassTA has a flexible grading service which allow faculty to perform re-grade or override grade in four different ways below:

1. **Auto re-grade from roster**: Should a mistake happen in the test, such as wrong answer attached to a question or wrong points given for a question, faculty can correct this mistake by modifying the test from Gradable Editor. After that, simply select “Regrade…” option in the roster that will trigger ClassTA to do a re-grade based on the corrected value. The new grades will be updated to roster automatically. Any manual grading performed before, will be calculated and updated back to roster.

2. **Auto re-grade (ignore manual score) from roster**: This re-grade option is similar to auto re-grade. The difference is, this option only performs re-grade based on the points and correct answer set in the gradable editor, all the manual grades changes will be ignored.

3. **Manual override from roster**: In roster, faculty can manually override the test score. To do this faculty may select the student row by clicking the checkbox next to student’s name. The next step is to go to the appropriate cell and enter the new grade. Lastly, faculty has to save the roster to ensure the change is stored properly. Total
score will be recalculated using the new override grade. Grade that has been manually edited is denoted with a green color text (see Figure 4.14).

4. **Manual override from test result**: Another option to override grade can be done from test result display. After students completed the test, faculty can view the test result by clicking the score in roster. Figure 4.15 show the example of a test result display. From test result, faculty can override grade for each questions and adds comments to it. This is particularly useful to grade a descriptive question type. When done with grading, faculty clicks on the “Submit Manual Grades” to trigger a re-grade process using the newly entered grades. Manually overridden grades will be marked with a green background color. In a case where students’ attempt needs to be reset, test result display has a handy feature to do so simply by clicking the “Clear Attempt” button.

![Figure 4.14. Override score from roster.](image)

![Figure 4.15. Override score from test result.](image)

### 4.4.2 Student Grading Service

ClassTA provides a tool for students to view grades information. Students can access this feature by selecting “My Grade” icon in the class desktop as shown in Figure 4.16. The display of “My Grade” is basically a simplified version of faculty roster which only show a
Figure 4.16. Student account – “My Grade” icon.

row for a particular student (see Figure 4.17). Students can view their grades and detail for each completed test only if permitted by faculty. Faculty can control this setting from faculty roster.

Figure 4.17. Student roster view in “My Grade”.

Detail for a test result can be accessed by clicking a cell containing students’ grades. Test result layout is similar to the original test interface but it has more features and information such as: correct answer, solution display, comments, and difficulty rating submission. Figure 4.18 show an example of online test result interface from a student’s point of view.
For each question in test result display, students can view the following information:

- **Solution**: If available and permitted, after the test deadline is over, students able to see the solution for each question. This extremely helpful to help students study and review their test.

- **Correct Answer**: Except for descriptive type question, the correct answer will be displayed here.

- **Student’s Points**: Points that student gets for this question.

- **Points Comment**: Comments from faculty.

- **Percentage of Correct Answer**: The number of students who answered the question correctly (in percentage).

Test result display has a rating system feature in which students will rate and submit the difficulty rating for each question. There are five different ratings to select from: 1-Very Easy, 2-Easy, 3-Normal, 4-Difficult, 5-Very Difficult. Faculty can view the overall statistics of difficulty ratings from Grable Editor.

**4.5 Test Statistics**

Test statistics is designed to provide faculty with a view of an overall result and statistics in a test. As the test progressed, question statistics and difficulty level can be viewed from Grable Editor. In this display faculty can get information about the average
difficulty level submitted by students. A histogram is presented with data of total number of students getting the particular questions and how many students answered it correctly. Difficulty level submitted by students also displayed in the same histogram (see Figure 4.19).

![Difficulty & Success Stats](image)

**Figure 4.19. Question difficulty and success statistics display.**

This view is really useful for faculty to see the following information like which question students think difficult or easy, which question most students failed to answer correctly, and how the comparison between the difficulty rating sets by faculty and students. All these information will be helpful to analyze class performance and create a better test in the future.

For survey, the Gradable Editor will show a similar display as above, however it also has the overall survey comments on the bottom (see Figure 4.20). As shown in Figure 4.21, each question detail has a histogram containing information regarding the survey such as: survey result, total persons answered the question, and the average points for each question. Average points for each survey question are calculated based on the number of students selecting each available option multiplied by the weight of each option. For example in Figure 4.21, the average is a result from \( ((1 \times 5) + (0 \times 4) + (2 \times 3) + (0 \times 2) + (0 \times 1)) / 3 = 3.67 \).
Figure 4.20. Survey result and comments display.

Survey Comments:

- I think the class is really good and useful
- the course is not explained so clear by instructor

Figure 4.21. Survey statistics display.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Persons</td>
<td>3</td>
</tr>
<tr>
<td>Very Good (5)</td>
<td>1</td>
</tr>
<tr>
<td>Good (4)</td>
<td>0</td>
</tr>
<tr>
<td>Average (3)</td>
<td>2</td>
</tr>
<tr>
<td>Poor (2)</td>
<td>0</td>
</tr>
<tr>
<td>Very Poor (1)</td>
<td>0</td>
</tr>
<tr>
<td>Average</td>
<td>3.67</td>
</tr>
</tbody>
</table>
4.6 COMPARATIVE STUDY OF TEST INTERFACE AND GRADING SERVICE IN CMS

Most CMS has similar test interface and grades function, however ClassTA has the following features that distinguish it from other CMS such as Moodle and Blackboard:

- ClassTA has an in-class test feature. In-Class test enable faculty to create a hybrid test where students take the test in the class and submit the answers online. This will minimize the cheating attempt and ensure that student take the test by themselves.

- The ability for faculty to attach solution to a question. If permitted, students can view their test result online along with the solution, comments, and statistics for each question. The solution display is helpful for students to study and review their past tests.

- Difficulty rating system where students can submit rating for each question. This function will help faculty gets information on class performance and level of understanding.

- ClassTA has a very flexible randomization rules and supports questions variation. Most CMS provides one randomization rule in questions level and randomization rule for multiple choices question. In ClassTA, faculty can determine more than one set of custom rules to randomize questions in a test.

- Option to include or ignore manual changes in re-grading process. With this option, faculties have flexibility to select which option needed when perform a re-grade in a test.

Figure 4.22 [30] show the display of student’s test result in Blackboard. Meanwhile, the test result viewer in Moodle is shown in Figure 4.23 [31]. Both result viewers lack the solution and rating system which are present in ClassTA (see Figure 4.24).

Figure 4.24. Student view test result in ClassTA.
CHAPTER 5

TESTING

This chapter will discussed various testing cases for ClassTA test interface and grading service module. Testing is done from student and faculty point of view to show how the system will handle each given case. In addition to this testing, on spring 2012, the new test interface and grading services module has been successfully piloted in the ME 350 Thermodynamics I class by Prof. Bhattacharjee.

5.1 FACULTY INTERFACE

In general, test cases for the faculty interface are involving the following process: creates the new test, updates the test, grades the test, and views test result and statistics. Different conditions and options for faculty to perform the entire tasks above will be explained in the following section.

5.1.1 Online Test

An online test is, as its name suggest, is a test taken by student online. This section will explain the testing process for online test from a faculty point of view.

5.1.1.1 CREATION OF AN ONLINE TEST

Online test is created from gradable editor interface. In ClassTA, there are three steps to create an online test. Those three steps are:

1. Compose/Edit Questions: In this step, faculty creates different types of questions for the online test. ClassTA also provides faculty with an option to search and import questions from questions bank.
2. Set/Update Rules: The next step is to set the test attributes and rules such as: submission windows, test duration, test display, late duration, solution display, and randomization rules.
3. Deploy the Test: If faculty is done with composing and setting rules for online test, the last step is to deploy the test. Once deployed, students can take the test within the given submission period time. A column representing the newly deployed online test will be added to roster (see Figure 5.1).
5.1.1.2 UPDATING AN ONLINE TEST

Online test which is already deployed can be updated to a certain points. If it is not considered as significant changes, faculty can update the test and then save it. There is no need to re-deploy the test. The types of changes allowed without re-deploying the test are:

- Cosmetic changes in a question, such as: change question type, update text in question body, change the answer, change the unit, change the tolerance, change the points, upload new schematic, upload new solution, and update the notes.

- Change the attributes of online test, such as: base points, roster, test duration, submission windows, maximum late submission, solution release, and basic instructions.

- Change the randomization rules.

Even though changes to the test attributes and rules above are permitted, ClassTA will show a message (see Figure 5.2) to warn faculty about the effect it might cause to the live test.

![Figure 5.2. Warning message when updating rules of a live online test.](image)
If new questions need to be added to a live test, faculty must withdraw the test. An alert, as shown in Figure 5.3, will appear to inform faculty about this. To be able to withdraw a live test, faculty must delete any existing instruction files (usually in pdf or xml format). The instruction files can be found in the “Locked” folder of faculty class desktop. ClassTA also checks if any students have already taken the test. If yes, it will show the warning message as in Figure 5.4 to inform faculty.

![Alert when trying to add question to a live test.](image1)

![Warning when withdrawing live test with students.](image2)

5.1.1.3 CHECKING ONLINE TEST PROGRESS

Faculty can check the latest progress on a test by viewing the roster. Roster contains information about students who are currently taking the test. The tool tips in roster’s cell will inform faculty which student still taking the test and which one already completed the test. Figure 5.5 show an example of test progress display in a roster.
5.1.1.4 RE-GRADING ONLINE TEST

If there is any mistake, faculty can update the online test from gradable editor. For example, faculty can modify points for a particular question by selecting the question’s detail, as shown in Figure 5.6. Faculty then enters the new points and saves it. Figure 5.7 shows the question detail with the new updated points. In case students already took the test, faculty can go to roster and performs automatic re-grading in roster.

There are two types of re-grade option in roster, the first one is “Regrade”. With this option, all the previous change to students’ grades will be included in the re-grading process. The scores in roster will be updated accordingly after re-grade is done (see Figure 5.8).
The second option is “Regrade (Ignore Manual Score)”. This option is provided in case faculty needs to re-grade the online test by ignoring all the previously overrides scores. It will trigger the grading services to reset all the manually overrides scores and perform a re-grade. The original score is updated back to roster (see Figure 5.9).

5.1.1.5 UPDATING ONLINE TEST SCORE IN ROSTER

Faculty has the ability to override online test score in roster. To do this, select a student by putting a check mark next to student’s name and then change the score in the selected row. After the roster is saved and reloads, the color of the overridden score is changed to green (See Figure 5.10).
5.1.1.6 **UPDATING ONLINE TEST QUESTION’S POINTS IN RESULT INTERFACE**

ClassTA has a feature to allow faculty to override points for any single question in the online test. This can be done in the test result interface. To access test result interface, go to roster, as shown in Figure 5.11, and clicks a cell containing student’s score. From here, faculty can override the machine points by entering new points in the “Final Points” textbox (see Figure 5.12). The color of the updated scores in roster, as shown in Figure 5.13, will be changed to purple.

![Figure 5.11. Roster view before single question point is overridden.](image)

![Figure 5.12. Test result display with updated final points.](image)

![Figure 5.13. Roster view after single question point is overridden.](image)
5.1.1.7 Reviewing Online Test Statistics and Rating

To review test statistics and rating, faculty can go to the gradable editor interface and clicks “Show Details” link in a question. As seen in the Figure 5.14, the statistics histogram and difficulty rating will be shown below each question body.

![Figure 5.14. Statistics and difficulty rating display in test editor.](image)

5.1.2 In-Class Test

In-class test is a hybrid test which combines an online test with paper based test. This section will discuss the test cases applied for in-class test from faculty’s perspective.

5.1.2.1 Creation of an In-Class Test

In-Class test is created by faculty with similar steps as creating an online test. One important setting that makes a difference for in-class test is the number of paper sets to be generated. When an in-class test is deployed, ClassTA will generate a pdf file which contains the questions and bubble sheet for answering the test. Each paper sets have its own unique submission code (see Figure 5.15). Faculty then prints out this file and distributes it to students in the class.
5.1.2.2 UPDATING, RE-GRADING, AND REVIEWING RESULT OF AN IN-CLASS TEST

The rules to update an in-class test are similar to updating an online test as explained in the Section 5.1.1.2. The re-grading process and reviewing in-class test result also the same as in online test.

5.1.3 Self-Graded Test

Self-graded test is similar to a take home test or homework where students keep track of their own progress. Test cases for self-graded are similar to online test where faculty as the main user will create, update, and review the test result.

5.1.3.1 CREATION OF A SELF-GRADED TEST

Creation of a self-graded test has the same steps as online and in-class test. However, self-graded test can only contains self-graded question type in which students select the completion percentage for each given task.

5.1.3.2 UPDATING, RE-GRADING, AND REVIEWING RESULT OF A SELF-GRADED TEST

Some settings in the self-graded test can be updated the same as in updating an online test. Self-graded test also has the same re-grade options as online test. The process to review
result and statistics of a self-graded test are as well no different than online test. Please refer to online test section above for more details.

5.1.4 Survey

Survey is a type of gradable used to collect data from students. The testing process from faculty’s side will be explained more detail in this section.

5.1.4.1 Creation of a Survey

Survey is a special type of gradable. Although it is created from gradable editor similar to the other gradable type, a survey does not have a right or wrong answer. Thus, it usually has no grades to it. If needed, faculty may give student a participation point for taking the survey. This point can be set in the “Base Point” field in Set/Update Rules steps.

5.1.4.2 Updating and Re-Grading a Survey

Survey can be updated similar to any other test. The same rules for updating online test also applied to survey. Survey initially has no grades to it, thus commonly no re-grading process needs to be done to survey. However, if faculty makes changes to survey’s base point, the same re-grading process explained for the online test can be done.

5.1.4.3 Reviewing Survey Statistics and Comments

Survey is confidential. No detail will be available for each submitted survey. However, faculty has access to view the overall comments, result, and statistics. Survey has a different statistics and result view than any other test. Faculty can view result of a survey by opening the gradable editor for the selected survey. Clicking each question in the editor will brings up the survey statistics and histogram for that particular question. As shown in Figure 5.16, survey overall comments are displayed at the end of the editor.

5.2 Student Interface

Testing for student interface is designed to cover all the processes done by students to take and completed a test. It starts from entering the gradable code to viewing the test result.
5.2.1 Online Test

This section explains the test cases for online test from student’s point of view from entering the submission code to reviewing online test result.

5.2.1.1 VALIDATING ONLINE TEST SUBMISSION CODE

The submission code for online test consists of three characters. When students enter the submission code, system will check the code and show the appropriate message. The different types of messages shown when validating the submission code are:

- Code is invalid: Figure 5.17 shows the message when a submission code is invalid. This can happen when a code is in the wrong format or not found in the database.
- Test is closed or not yet opens: If test is already closed, students will get error message as shown in Figure 5.18. Meanwhile if it is too early, a message as in Figure 5.19 will be displayed.
- Student has completed the test: Figure 5.20 show a message when student already completed the test.
- Late submission with penalty: A warning message as in Figure 5.21 showed when student is late but still allowed to take the test where a penalty may applied.
- Code is valid: If the code is valid, student will be taken straight to the online test page as in Figure 5.22.
Figure 5.17. Error message when gradable code is invalid.

Figure 5.18. Error message when the gradable is already closed.

Figure 5.19. Error message when the gradable is not open yet.
Figure 5.20. Error message when student already completed and submitted the test.

Figure 5.21. Warning message when student is late but still allowed taking the test.

Figure 5.22. Test interface display when gradable code is valid.
5.2.1.2 **Taking an Online Test**

Online test interface (Figure 5.23) will display the test based on settings and rules set in test editor. In the case where students quit in the middle of the online test and try to take the test again, the duration of the test will be calculated from the last access time. In addition to that, only the previously un-submitted questions will be displayed (Figure 5.24). Students are not able to work on any questions that already been submitted in the previous attempt.

**Figure 5.23. Online test first attempts.**

**Figure 5.24. Online test second attempts.**
5.2.1.3 Viewing Online Test Grade

If permitted, after completed an online test, students can view their grades by clicking “My Grade” icons on the class desktop. Students’ view of “My Grade” page is depends on the level access given by faculty. In roster, faculty can set the type the access granted to student. Different types of access to view students’ grade are:

- **Immediate**: If the student access in roster is set to “Immediate”, after a test is submitted students can go to “My Grades” and view their test score and detail result (see Figure 5.25).

- **Delayed**: If the access in roster is set to “Delayed”, students cannot view the score and result until the test is closed (see Figure 5.26).

- **Overall Stats Only**: If the access in roster is set to “Overall Stats Only”, no scores will be shown in “My Grade”. Students only able to see the overall statistics (see Figure 5.27).

- **Disabled**: If the access in roster is set to “Disabled”, students have no access to view the grade page at all (see Figure 5.28).

![Figure 5.25. Student grade view with “Immediate” access in roster.](image)

![Figure 5.26. Student grade view with “Delayed” access in roster.](image)
5.2.1.4 VIEWING ONLINE TEST RESULT DETAIL AND SOLUTION

To be able to view the online test result and solution, first of all faculty must granted permission for students to access “My Grades”. From there, students can view test detail by clicking on their score in the roster. If available, once the test deadline is over, students can also see the test solution and submit difficulty level (see Figure 5.29).
5.2.2 In-Class Test

In-class test has a unique submission code for each student. This section will describe the testing process to validate in-class test submission code and other test cases from student’s point of view.

5.2.2.1 VALIDATING IN-CLASS TEST SUBMISSION CODE

The process to validate in-class test is similar to the online test validation. The only difference is the code format. Since each student got different test paper, they also got a different submission code linked to the each test paper. Submission code for in-class test consists of two sets of three digit characters separated by a semicolon, for example 3bc:h4g. The first part of the code tells the system which in-class test students are taking. The second part is needed to check which test paper they are getting.

5.2.2.2 TAKING AN IN-CLASS TEST

In-class test is a type of hybrid test where students take the test in the class and later on submit the answers online. In the class, students answer the questions in a given bubble sheet. The interface for in-class test will be shown similar as in the bubble sheet (see Figure 5.30) and students are not allowed to see the questions again.

![Question Display Panel - Click to Expand/Contract](image)

**Figure 5.30.** In-class test interface.
5.2.2.3 Viewing In-Class Test Grades, Result Detail, and Solution

The process to check grades, result detail, and solution of an online test are applied to the in-class test as well. Please refer to Online Test section above for more details.

5.2.3 Self-Graded Test

In self-graded test, students will enter the percentage of completion for each given task. This section will discuss test cases for self-graded test from a student’s perspective.

5.2.3.1 Validating Self-Graded Test Submission Code

Self-graded has the same submission code format as online test. The rules to validate self-graded submission code are also the same as online test in Section 5.2.1.1.

5.2.3.2 Taking a Self-Graded Test

In self-graded test, students will answer the test questions by selecting the completion percentage for each given task. The interface for self-graded will always shows all questions in one page and only have one submit button in the bottom of the page (see Figure 5.31).

![Self-graded test interface](image-url)

Figure 5.31. Self-graded test interface.
5.2.3.3 VIEWING SELF-GRADED TEST GRADES, RESULT DETAIL, AND SOLUTION

The rules applied to check grades, result detail, and solution of a self-graded test are the same as in online test. More detailed explanation about these processes can be found in the Section 5.2.1.

5.2.4 Survey

This section describes the testing process from how students take a survey and what type of survey result is available for them.

5.2.4.1 VALIDATING SURVEY SUBMISSION CODE

Validation process for survey submission code is similar as in online test. The format of a survey submission code is also the same as online test code. Please refer to Section 5.2.1.1 for more details.

5.2.4.2 TAKING A SURVEY

When students are taking a survey, all questions in the survey will be displayed in one page. As shown in Figure 5.32, similar to self-graded test, the interface for survey only has one submit button. The unique thing about survey is it has a textbox at the end of the page where students can enter their comments.

5.2.4.3 VIEWING SURVEY RESULT DETAIL

Privacy and confidentiality is bedrock of a survey. If permitted, students can only see the overall statistics of a survey from “My Grade”. No detail will be available for both students and faculties.
Figure 5.32. Survey interface.
CHAPTER 6

FUTURE ENHANCEMENTS AND CONCLUSION

6.1 FUTURE ENHANCEMENTS

AJAX technology has made the test interface and grading service in ClassTA as a fast and responsive module. In the future, some enhancements below can be added to make it better for faculties and students:

1. Enable function to upload media files, such as video or sound clip, to a question. This will be useful particularly for a language test. Currently when designing a test, only text or pictures can be attached to a question.

2. Add a function to allow student to review all their answers before submitting the test.

3. Enhance the difficulty rating system by adding a more flexible rating interface such as star rating with full and half star rating.

6.2 CONCLUSION

The new test interface and grading service in ClassTA provides a more advanced function compare to the older version. The grading service offers lots of flexibility for faculties to perform grading and re-grading process. Test interface has more functions yet still emphasized on a simple design to make it easy to use. The new test result display is developed to enabling users to review the test and can also be used as a proof of student’s work if there is any discrepancy with their grades.
BIBLIOGRAPHY


APPENDIX

DATABASE TABLES
### Table A.1. Structure of quiz_live Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>quizLiveID</td>
<td>Integer</td>
<td>Primary key.</td>
</tr>
<tr>
<td>quizID</td>
<td>Integer</td>
<td>Reference to the quiz in progress.</td>
</tr>
<tr>
<td>userID</td>
<td>Integer</td>
<td>Reference to the student’s user id.</td>
</tr>
<tr>
<td>email</td>
<td>Varchar</td>
<td>User email address.</td>
</tr>
<tr>
<td>rosterIRL</td>
<td>Integer</td>
<td>Reference to roster where student belongs.</td>
</tr>
<tr>
<td>questionID</td>
<td>Integer</td>
<td>Reference to question in the quiz.</td>
</tr>
<tr>
<td>choices</td>
<td>Text</td>
<td>Stores the options that student gets for a multiple choice question.</td>
</tr>
<tr>
<td>sequence</td>
<td>Text</td>
<td>Stores the sequence of a question.</td>
</tr>
<tr>
<td>studentAnswers</td>
<td>Text</td>
<td>Stores student’s answer.</td>
</tr>
<tr>
<td>startTime</td>
<td>Datetime</td>
<td>Stores the time student start the test.</td>
</tr>
<tr>
<td>timeSubmitted</td>
<td>Datetime</td>
<td>Stores the time student submitted a question.</td>
</tr>
<tr>
<td>offset</td>
<td>Integer</td>
<td>Stores the offset for in-class test to determine the randomization rules.</td>
</tr>
<tr>
<td>isSubmitted</td>
<td>Integer</td>
<td>Flag to check if a question is submitted or not.</td>
</tr>
<tr>
<td>secondCode</td>
<td>Varchar</td>
<td>Stores the second part or test code, used for in-class test.</td>
</tr>
</tbody>
</table>
Table A.2. Structure of quiz_result Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resultID</td>
<td>Integer</td>
<td>Primary key.</td>
</tr>
<tr>
<td>quizID</td>
<td>Integer</td>
<td>Reference to the quiz table.</td>
</tr>
<tr>
<td>userID</td>
<td>Integer</td>
<td>Reference to the student’s user id.</td>
</tr>
<tr>
<td>email</td>
<td>Varchar</td>
<td>User email address.</td>
</tr>
<tr>
<td>rosterID</td>
<td>Integer</td>
<td>Reference to roster where student belongs.</td>
</tr>
<tr>
<td>resultJSON</td>
<td>Text</td>
<td>Stores in result detail in JSON format.</td>
</tr>
<tr>
<td>totalScore</td>
<td>Decimal</td>
<td>Stores total score from automatic grading.</td>
</tr>
<tr>
<td>scoreInRoster</td>
<td>Decimal</td>
<td>Stores the score in roster, to record if there is any manual grading by faculty.</td>
</tr>
<tr>
<td>timeStart</td>
<td>Datetime</td>
<td>Stores the time student start the test.</td>
</tr>
<tr>
<td>timeEnd</td>
<td>Datetime</td>
<td>Stores the time student end the test.</td>
</tr>
<tr>
<td>secondCode</td>
<td>Varchar</td>
<td>Stores the second part or test code, used for in-class test.</td>
</tr>
<tr>
<td>offset</td>
<td>Integer</td>
<td>Stores the offset for in-class test to determine the randomization rules.</td>
</tr>
<tr>
<td>scoreDetail</td>
<td>Text</td>
<td>Stores the score detail for roster tool tips.</td>
</tr>
<tr>
<td>isUpdateToRoster</td>
<td>Bit</td>
<td>Flag to check if the result is updated to roster or not.</td>
</tr>
<tr>
<td>studentDuration</td>
<td>Varchar</td>
<td>Stores the duration student needs to completed the test.</td>
</tr>
<tr>
<td>lateOrEarly</td>
<td>Varchar</td>
<td>Stores how late or early student completed the test.</td>
</tr>
<tr>
<td>hasSubmitRating</td>
<td>Integer</td>
<td>Flag to check if student already submit difficulty rating or not.</td>
</tr>
</tbody>
</table>
### Table A.3. Structure of question_statistics Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Integer</td>
<td>Primary key.</td>
</tr>
<tr>
<td>questionID</td>
<td>Integer</td>
<td>Reference to question table.</td>
</tr>
<tr>
<td>json</td>
<td>Text</td>
<td>Stores the question statistics, difficulty rating, and survey result in JSON format.</td>
</tr>
</tbody>
</table>