12-2011

Enhancing iNetTest by Improving the Programming Question and Group Grading

Sushil Dosi
Utah State University

ENHANCING iNETTEST BY IMPROVING THE

PROGRAMMING QUESTION AND GROUP GRADING

by

SushilDosi

A report submitted in partial fulfillment
of the requirements for the degree
of

MASTER OF SCIENCE

in

Computer Science

Approved:

Donald H. Cooley Major Professor	Curtis Dyreson Committee Member

Stephen J. Allan
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah
2011

ABSTRACT

Enhancing iNetTest by Improving

the Programming Question and Group Grading

by

SushilDosi

Utah State University 2011

Major Professor: Dr. Donald H. Cooley
Department: Computer Science

This report describes an improvement to the Utah State University iNetTest testing system. The iNetTest system allows instructors and/or students to:

- Create/take tests with rich sets of question types (multiple choice, essay, true/false, computational programming question, etc.);
- Monitor the test takers for cheating;
- Auto-grade for many types of questions, as well as group grade for all question types; and
- Send scores to students via either email or SMS.

Specifically, this report discusses the design and development of an improved computational programming question for the iNetTest system. For programming questions, iNetTest allows for the use of various programming languages including some scripting languages. The improved system makes grading faster and more straightforward by assessing all students' answers automatically. All enhancements described herein improve iNetTest's

functionality and implement new security layers that protect against the misuse of features and/or functionality.

This report also describes the layered architecture used to build the iNetTest system, including several new technologies, such as Ajax[4] and JavaScript Frameworks[5]. MVC frameworks[1] and socket programming[10] are also discussed and compared. Finally, this report discusses how the system was tested and projects future enhancements to the system.

CONTENTS

Page

Abstract……. **i**

Acknowledgments.....

List of Tables…… **iv**

List of Figures..... **v**

1. Introduction to iNetTest……. **1**
1.1. Enhancement……... 2
1.1.1. Running Programs Through iNetTest……. 2
1.1.2. Improving the Group Grading of Programming Questions…….. 3
1.1.3. Support to Run Scripting Languages……... 4
1.2. Enhancements 4
Summary……...
5

2. Functions……. 5
2.1. Running Programs in Programming Question Mode…… 5
2.1.1. Instructor View..……... 14
2.1.2. Student View…….. 15
2.1.3. Preview View……. 16
2.2. Grading 17
View……
2.3. Group Grading View…… **19**
19

3. Implementation and Technology Used…… 19
3.1. Layered Architecture…… 20
3.2. MVC Web Frameworks…… 21
3.2.1. Struts MVC Architecture……... 23
3.2.2. Struts MVC Workflow..…… 25
3.3. Comparison of JavaScript Frameworks…… 26
3.4. Ajax 27
…..……..... 27
3.5. JSON …...…… 28
3.6. PostgreSQL ……..... 30
3.7. Socket 30
Programming……. 30
3.7.1. Workings of a Programming Server……...
3.8. Version Control System…… **31**
3.8.1. Subversion…….……..... 31
3.8.2. Subclipse …...……..... 32

4. Testing.....
4.1. Unit Testing.....
4.2. System Integration Testing..... **34**
34
35
36

5. Enhancement Results and Future Work.....
5.1. Overall Improvement..... 37
5.2. Conclusion .....
5.3. Future Work .....

References.....

List of Tables

Table Page

1. Comparison of JavaScript Frameworks 23

List of Figures

Figure Page

1. Programming languages available to instructor....5
2. Question text and code included in question....6
3. Instructor program for questions....8
4. Options of functionality available to students....10
5. Student view of the programming test....11
6. Preview view of a test....12
7. Grading view of a programming test....13
8. Group grading view of a programming question....14
9. Struts MVC....15
10. Struts MVC workflow....18
11. Ajax sequence diagram....22
12. Example of a JSON object....23
13. Socket programming....24
14. Socket server workflow....25

CHAPTER 1

INTRODUCTION TO INETTEST

The Utah State University iNetTest system is a web-based application that offers instructors a variety of automated tools to test students' knowledge. Further, the iNetTest system provides flexibility to students, while reducing instructors' grading efforts and providing a wide variety of question types.

Utah State University (USU) instructors have used iNetTest for more than 12 years. During that time, Dr. Donald Cooley has managed the development of the software. Until CIL was abandoned as a requirement by USU, iNetTest was used to administer the general education Computer and Information Literacy (CIL) tests to measure students' ability to use computers to access and present information. Both graduate and undergraduate students carried out software development for iNetTest, as well as maintenance and updates of the system.

Using iNetTest, instructors can create tests customized to meet their course objectives for assessment. There is wide range of question types available in iNetTest, including:

- Calculated questions
- Essay questions
- Fill in the blank questions
- Matching questions
- Multiple choice questions
- Multiple question questions
- Programming questions
- Sequence questions
- True or false questions

An iNetTest test can be unlocked (made available) either for a specific student(s) or a group of students with specified start and end times for the test to remain unlocked. This provides students flexibility, allowing them to choose the time at which they can take a test, using either a computer in a USU computer lab or at some other location. To prevent honor code violations, iNetTest has a unique anti-cheating feature. When the feature is enabled, if another browser window is opened during the taking of a test, the test closes. An e-mail containing a (be consistent, it is either email or e-mail, not both) screenshot of the student's computer screen at the time of the infraction is then sent to the instructor. This is a unique iNetTest feature that reduces the need to monitor students while they are taking tests. If the computer is not available or the instructor wants to conduct an in-class test, the system is capable of exporting a test in either PDF or Word format. Grading in iNetTest is designed to minimize the instructor's workload. The system supports auto-grading for many types of questions. In cases where manual grading is required, such as for an essay question, iNetTest allows the grader to grade the same question for each student answering the question, thus allowing for more consistency in grading.

Another feature of iNetTest is the option to send students their scores via e-mail and/or through SMS.

1.1 Enhancements

1.1.1 Running Programs through iNetTest

The run feature for programming questions brings various advantages to the system. Previously, while creating programming tests, computer science instructors had only the option of compiling the program. The same holds true for students taking the test and for instructors grading such tests: the only option available was compiling the program. If an instructor wished to run a student's program as part of the grading, she/he had to save it onto his/her system and

install the respective compiler to get results, which was tedious and time consuming. The run feature in iNetTest, allows instructors to compile and run programming question answers, while staying within iNetTest itself.

The system has the capability of allowing students and instructors to both write and execute programs from within iNetTest. Rather than simply checking for syntax errors, they can check their programs for output or run time errors. For students to have these capabilities, the test owner must grant permission before they are allowed to view both compiler and run time results.

1.1.2 Improving the Group Grading of Programming Questions

As stated above, in the first implementation of iNetTest, the instructor had to compile the program written by each student, one at a time. The group grading enhancement offers a way to compile all student programs as part of a single command. The group grading enhancement also allows the instructor to run all student programs with a single command. The result of either of these commands is a scrollable window showing the results, in sequence, for each of the students who answered that particular question.

1.1.3 Support to Run Scripting Languages

As another enhancement is that iNetTest now supports additional languages, specifically, scripting languages such as Perl, Python, PHP, and Ruby. This feature allows an instructor to create tests using a variety of programming languages, and students to answer questions in a variety of programming languages.

1.2 Enhancements Summary

The enhancements made to iNetTest as part of this project (report) introduced time and resource-saving functionalities. Each enhancement of the iNetTest system solves specific instructor/student problems. Object-oriented programming practices and advanced frameworks

were used to implement each enhancement. Ajax is becoming the standard for advanced web applications, and this project used Ajax to meet this standard.

CHAPTER 2

FUNCTIONS

Enhancements to iNetTest are visible throughout the iNetTest system. This chapter discusses these functions and feature enhancements, as viewed in the following modes:

- Instructor view
- Student view
- Preview view
- Group grading view

2.1 Running Programs in Programming Question Mode

2.1.1 Instructor View

The instructor view is used to create programming tests. It now has customizable options that instructors can use to modify the tests to meet their own course objectives.

When an instructor creates a test containing a programming question(s), the first task is to select the desired programming language. iNetTest gives the instructor the option of selecting a different language for each question. The selected language becomes the language the student must use to answer that particular question. The instructor has a wide variety of language choices, including C++, C#, Java, Perl, Python, Ruby, and PHP. Figure 1 shows a screenshot from the Create Question page. The figure shows a drop-down menu populated with the currently available programming languages. An asterisk (*) over the name of a programming language denotes that it as a scripting language. A negation (~) over the name of a programming language denotes that the compiler for that language is not available; JBabu and JLokesh are dummy names inserted to show this functionality. An instructor could develop a question for

which a compiler is not available. In such a case, the instructor/student would not be able to compile and/or execute the answer for grading purposes.

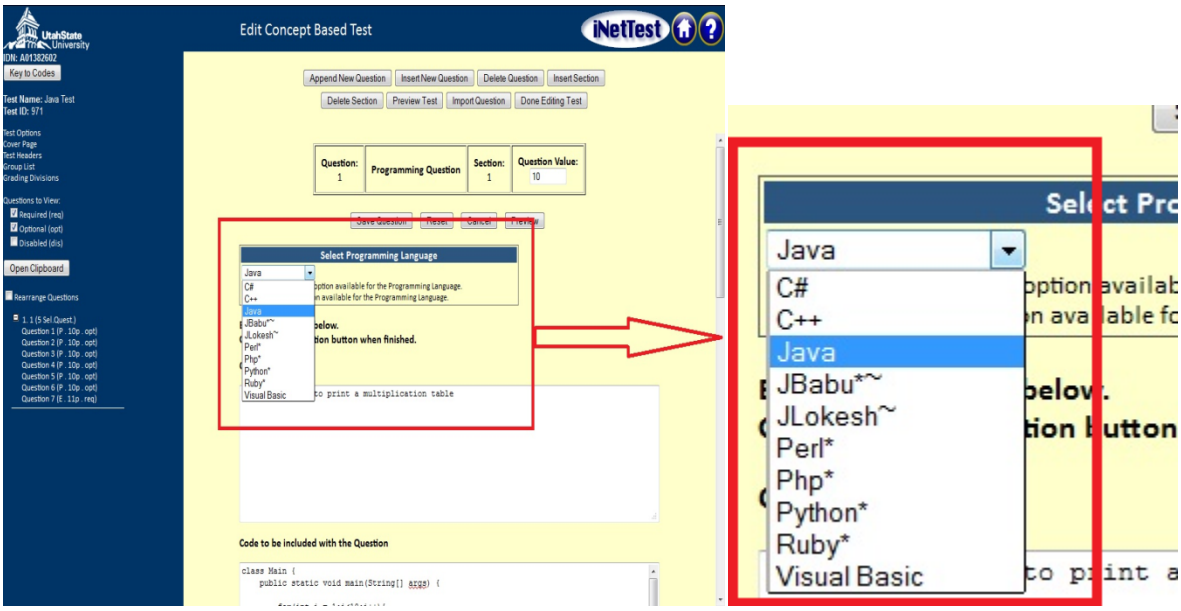

Figure 1.Available programming languages in instructorview.

After selecting the programming language and writing the text for the question, the instructor can choose from three different versions of the programming question:

1) “Write the complete program” mode. See Figure 2.
2) “Fill in the blank or blanks” mode. See Figure 3.
3) “Fill in a text field in the code” mode. See Figure 4.

In ”Write the complete program” mode, the instructor does not provide any code to the student. Instead, the “Code to be included with the question” window, as shown in Figure 2, is left blank. While answering this type of programming question, the student is required to write

all of the code needed for the program to compile and execute. This would include options such as needed imports/includes, etc. When taking the test, a student will only see a text area window in which she/he must write the correct code for the entire program. When student compiles and possibly executes the code, it is only the contents of this window that are sent to the compiler/server.

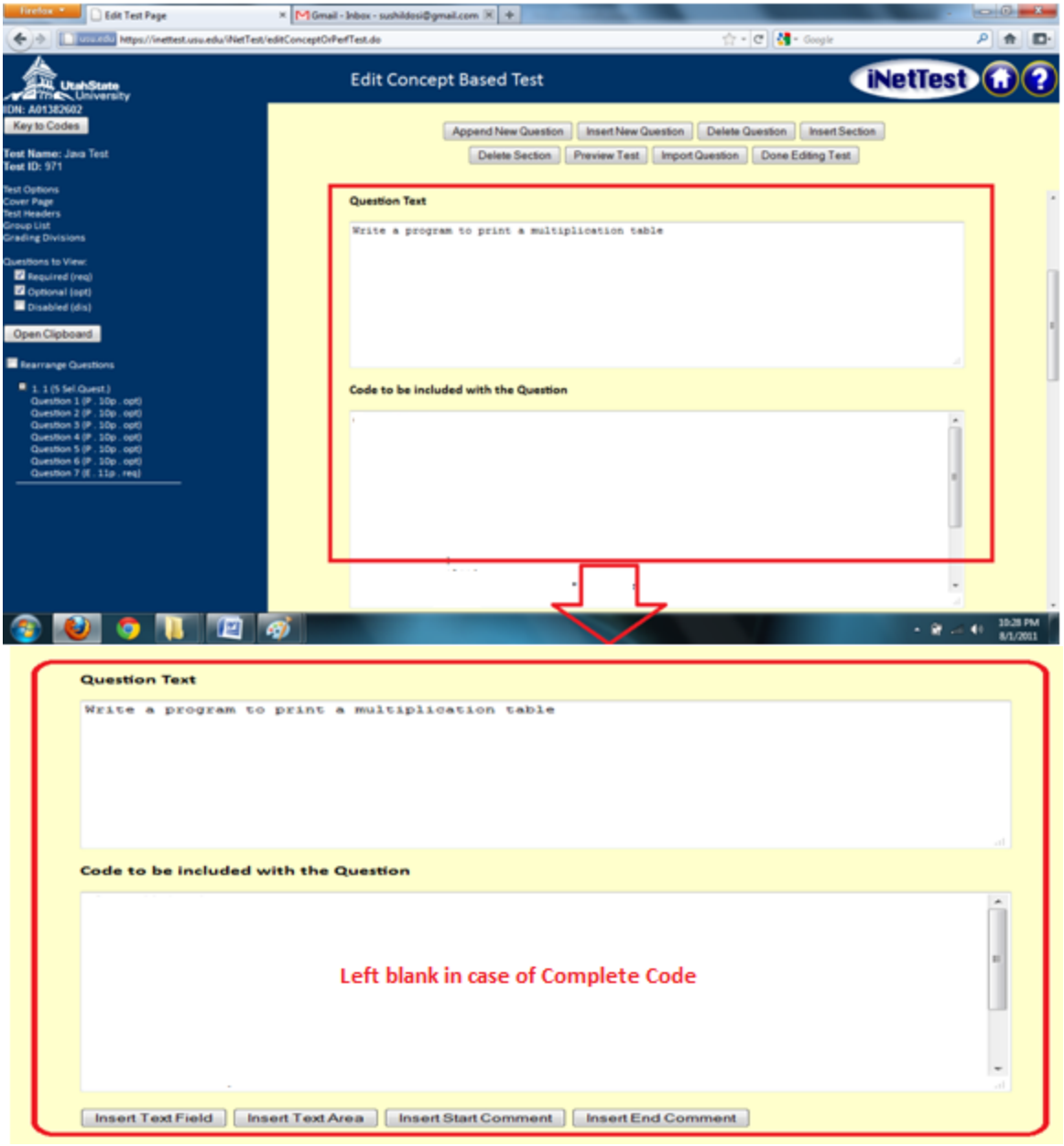

Figure 2.“Write the complete program” mode.

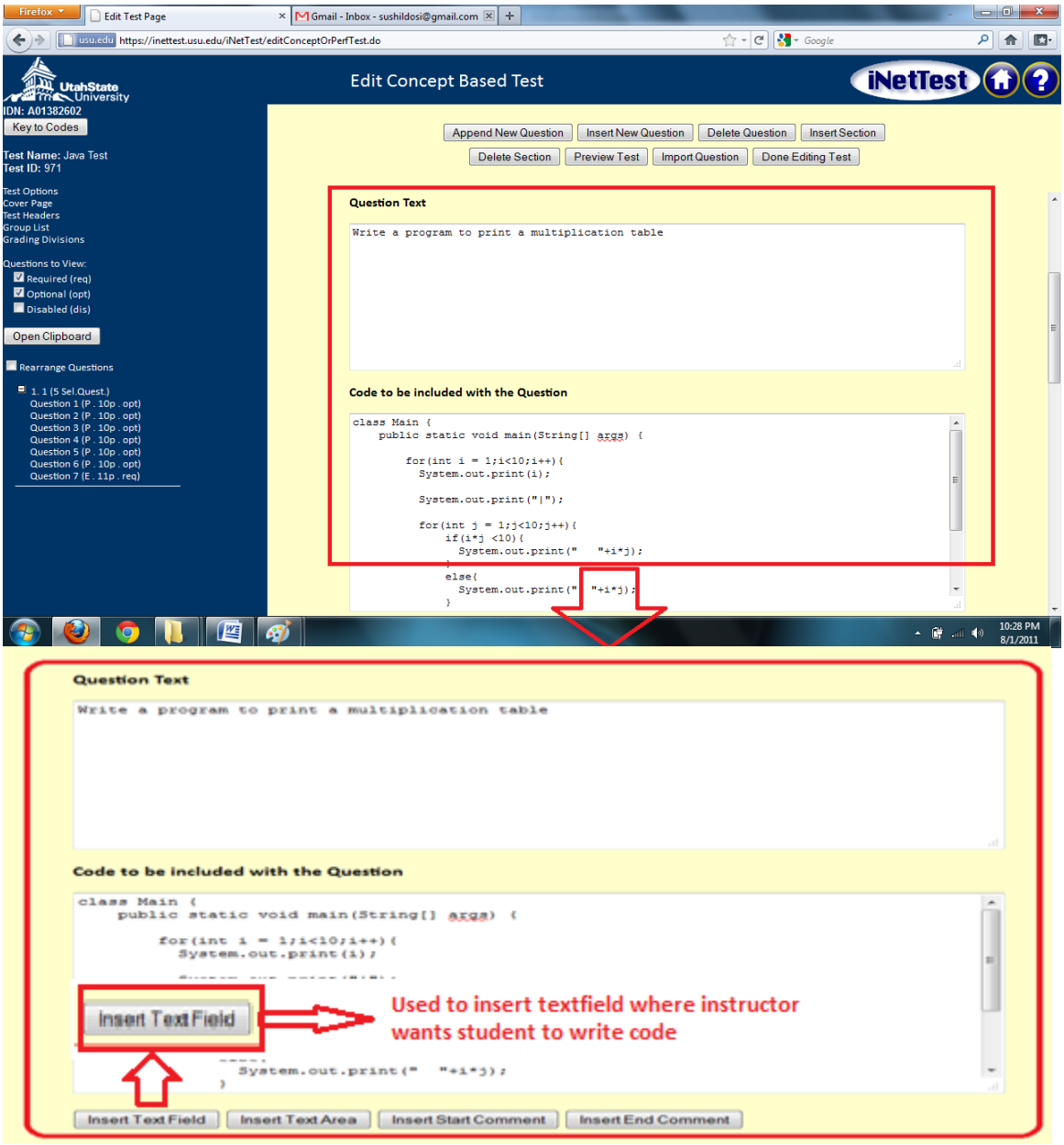

Figure 3.Question text and partial code for "Fill in the blanks" mode.

In "Fill in the blanks" mode, the instructor provides a portion of the code in the "Code to be included with the Question" window. Within this window, the instructor inserts a text field(s) in those places where in the student is to add the needed code (Figure 3).

In “Fill in a text field in the code” mode, the instructor provides partially completed code in the “Code to be included with the Question” section and inserts a text area where she/he wants students to write the code (Figure 4). The difference between “Fill in the blank” and “Fill in the field” questions is that the latter is used when an instructor wants students to write more than one short segment or line of code. Also the instructor can insert multiple text areas within the test question. While answering such a question, students sees one or more blank text areas within the partially completed code.

As noted previously, instructors can write, compile, and run their own code for each question, which they can then use as a grading key. The key is not visible to the student. In order to prevent an infinite loop from “hanging” the grading process, there is an option to set the maximum running time for a program. This option terminates the student’s program after a specific time, preventing the iNetTest server from being trapped in a deadlock, locked in an infinite loop, or eventually experiencing out-of-memory errors. If an instructor wants to test his/her code with various inputs, space is also provided to pass command line arguments. Figure 5 shows the instructor’s code in the “Program Grading Rubric” section. Along with Compile and Run buttons, the text area for providing command line arguments is shown.

Finally, instructors are able to choose the functionality they want available to their students during the test. There are three options: no compile and no run, compile, and compile and run. Figure 6 shows how the above three options are available to the instructor.

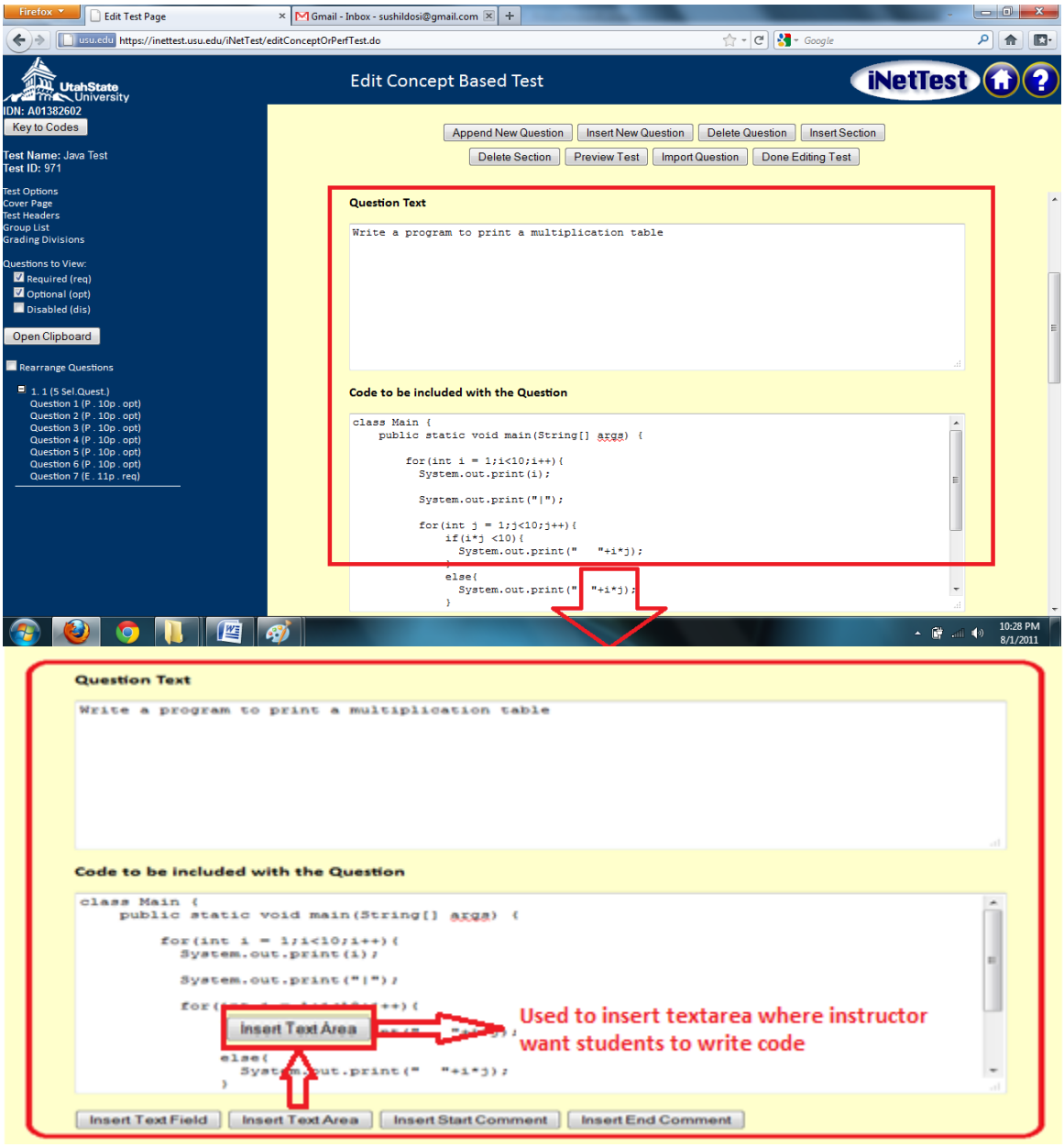

Figure 4.Partial code with InsertText Area function.

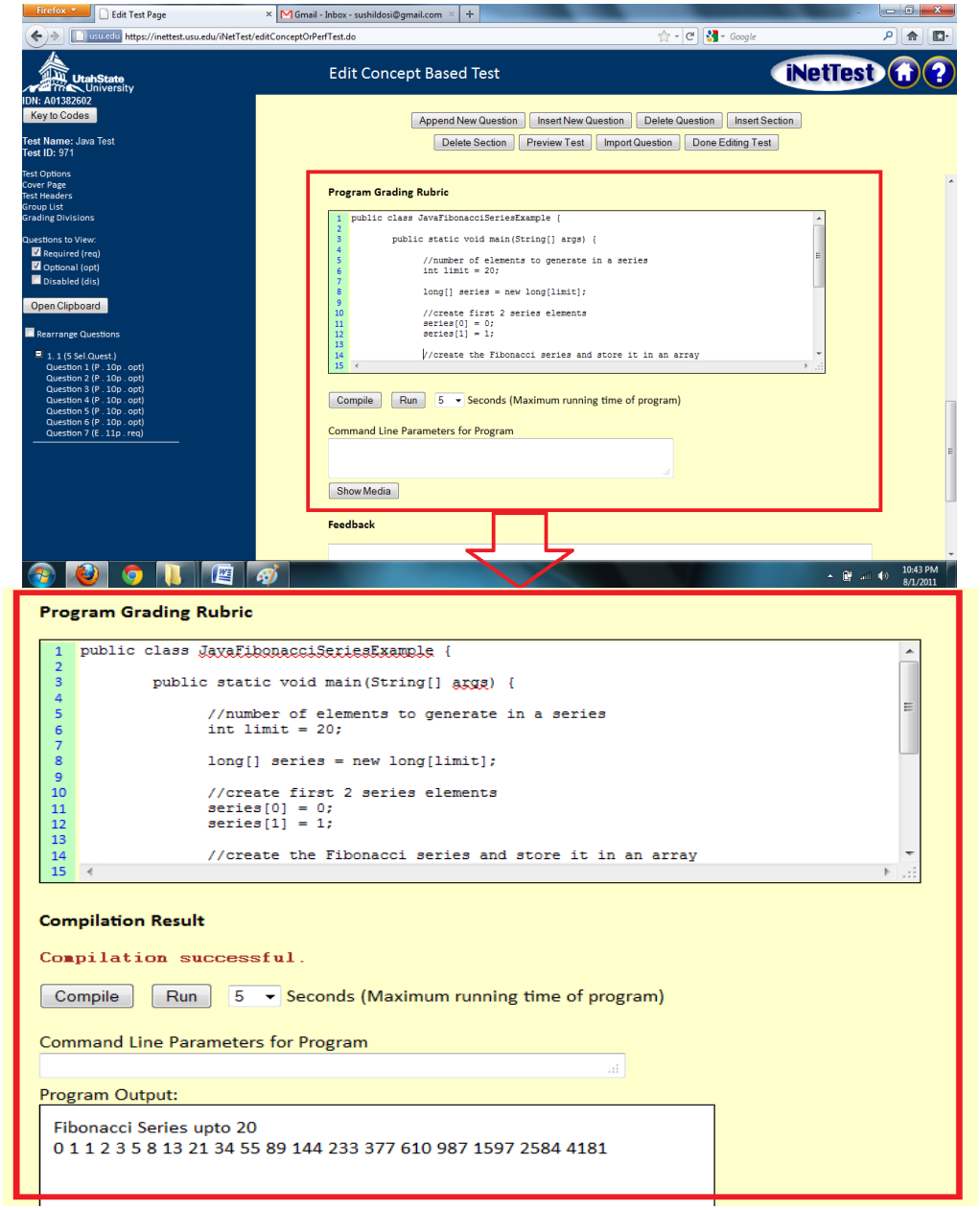

Figure 5.Instructor's answer for a question.

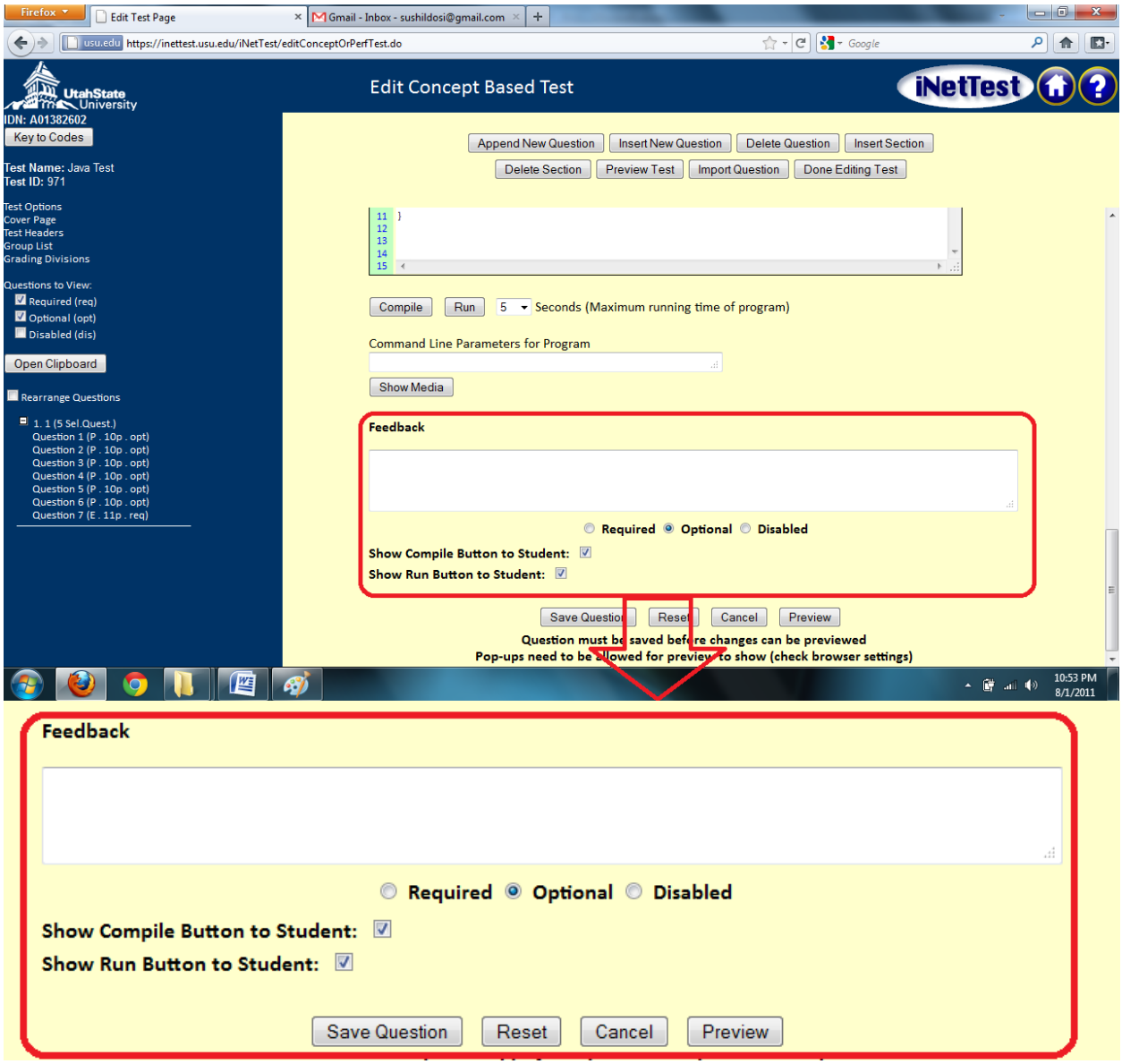

Figure 6. Functionality choices for what students will see.

2.1.2 *Student View*

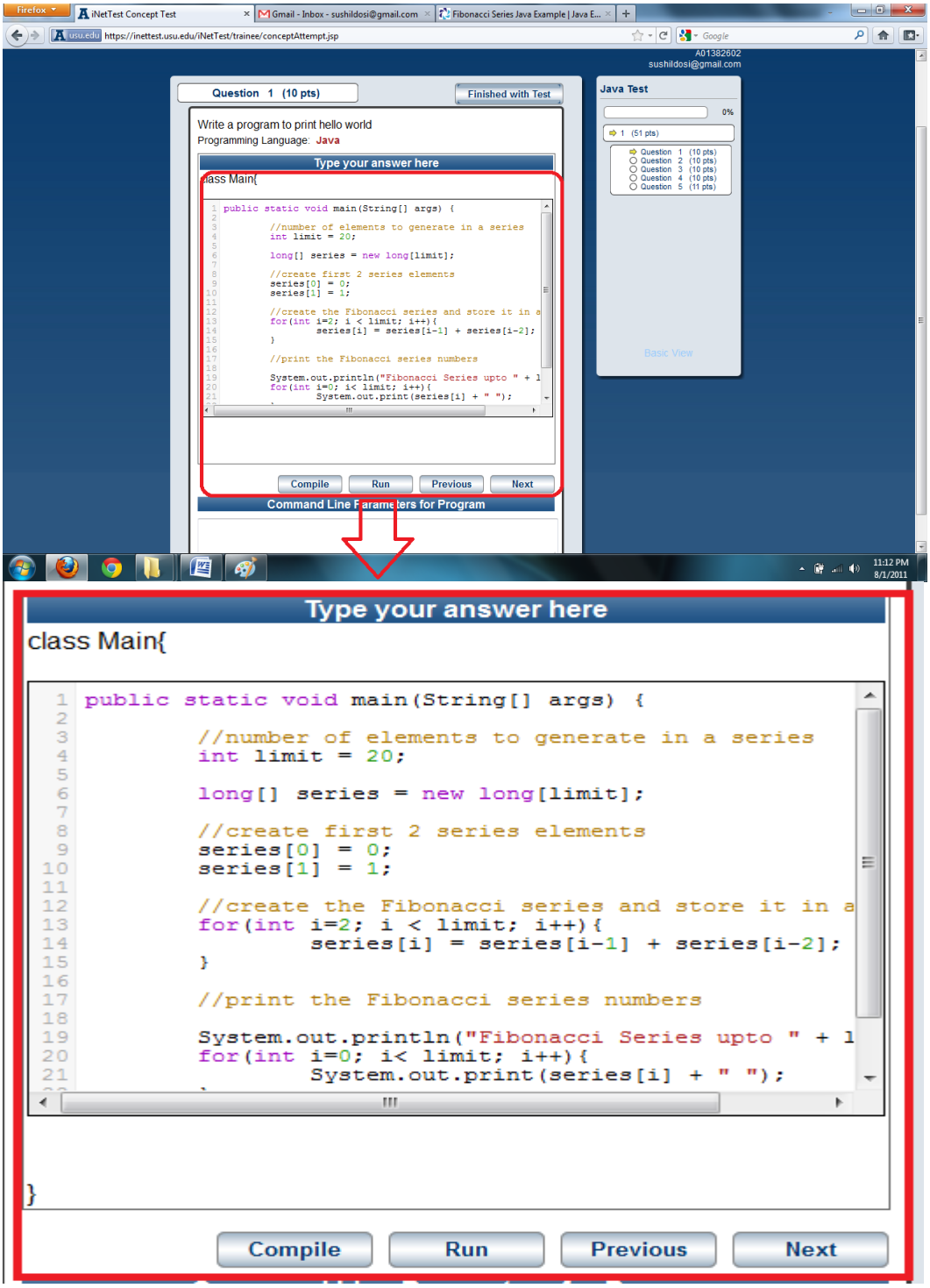

Figure 7.Student view of the programming question.

The design of this view (Figure 7) was kept as simple as possible in order to make the learning curve for the user interface to be as flat as possible. If an iNetTest test is unlocked (made available) for a student, that student is presented with the view shown in Figure 7. In an effort to improve students' performance in answering these types of test questions, syntax highlighting has been incorporated into the editor. Depending on the functionality chosen by the instructor while creating the Test, Compile, and Run buttons may be shown or hidden. The textbox space to pass command line parameters is also provided. The maximum running time of the program is preset to a default setting of 20 seconds by the system. Figure 7 shows an editor with syntax highlighting, as well as other buttons for navigation; this is what is shown to students when they take any programming test.

2.1.3 Preview View

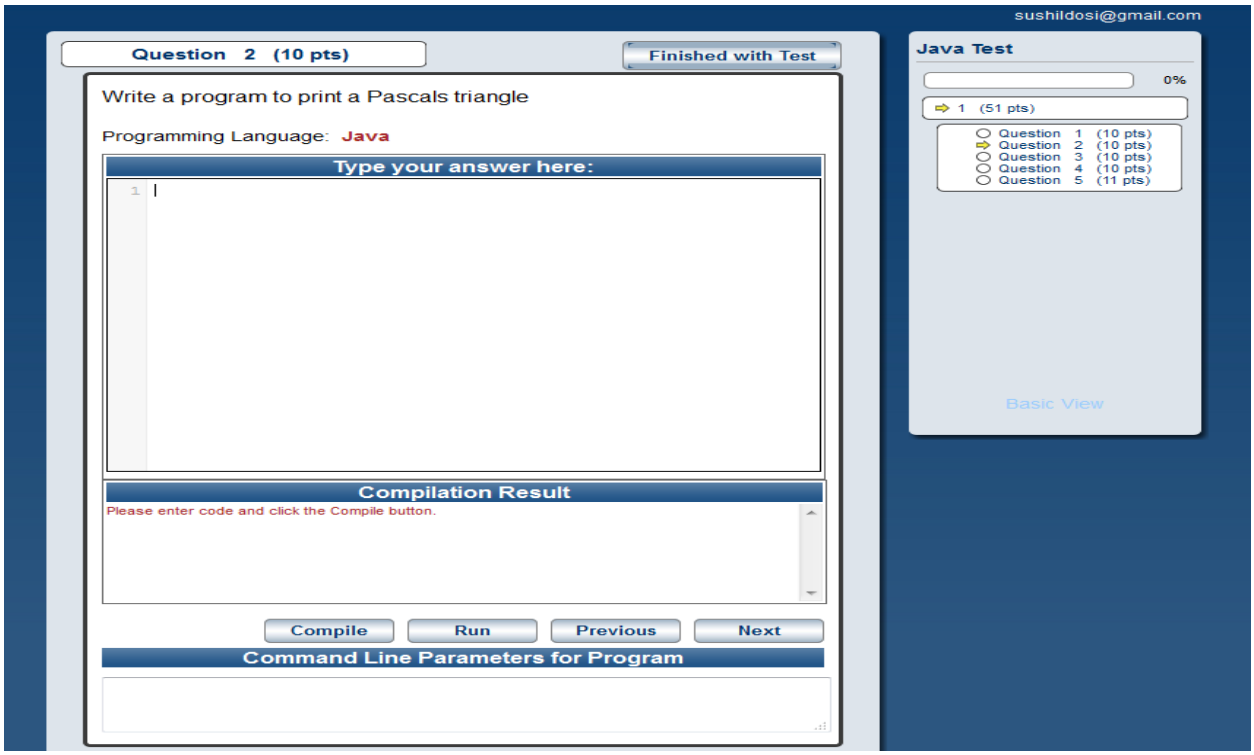

Figure 8. Preview view of a test.

A preview of the functionality of a test is available to the instructor only. The preview shows the instructor what the student sees, so that the instructor can check for any errors in the questions or their formatting. When the preview is active, the instructor can also write, compile, and run code like any other student. See Figure 8.

2.2 **Grading View**

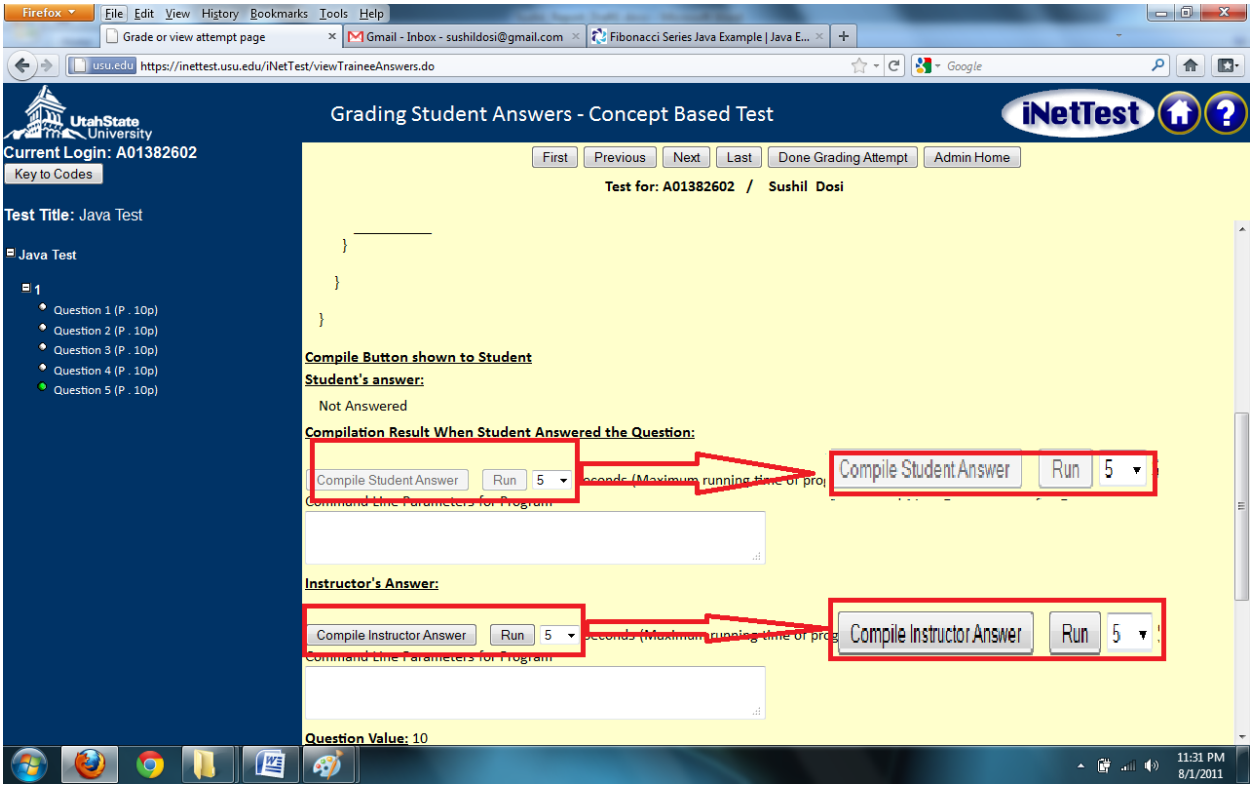

Figure 9.Grading View for programming test.

After a student has taken a test, the assigned grader must grade the test. The Grading View allows the instructor to view the students' program output directly in iNetTest. Consequently, grading individual students' work is easier for the instructor or grader. Along with each student's program, iNetTest shows the instructor's answer, so that it is easy to compare the

grading key with the student's entry. Finally, the instructor can run both programs and check the output in the same window.

Instructors who grade tests individually, see both the student answer and their answer at the same time. Even if students do not have access to the compiler while they take the test, the instructor can still compile and run the student's program in iNetTest itself. Figure 9 shows the Grading view for a programming test.

2.3 Group Grading View

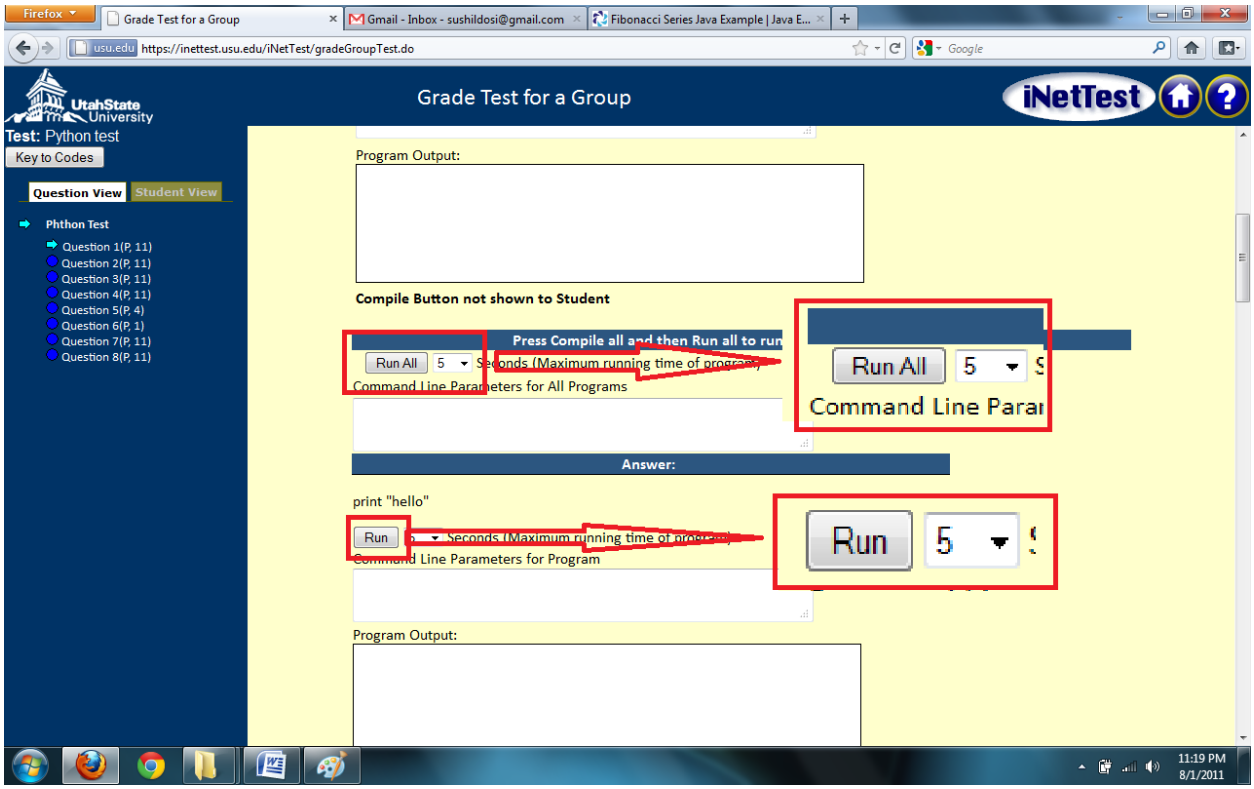

Figure 10. Group Grading view for a programming question.

The Group Grading function saves even more time while grading. Group Grading uses a Compile All and Run All functionality to compile and run all the students' programs at the same time.

It is time consuming to grade a large group of students' work by going through each program individually, so the Group Grading functionality provided to the grader shows the programs and if needed their outputs of all students for a particular question. The functionality also includes a unique feature of Compile All and Run All that allows the grader to compile and run all programs at once. Note that this functionality is always available for the grader, whether or not the students are provided with Compile or Run functionality. Figure 10 shows the Group Grading view of a programming test with Compile All and Run All functionalities. The Group Grading enhancement offers a way to compile and run all student programs using a single command. The result of either of these commands is a scrollable window showing the results, in sequence, for each of the students who answered that particular question. This saves the instructor considerable time while grading the work of a large number of students.

CHAPTER 3

IMPLEMENTATION AND TECHNOLOGY USED

3.1 Layered Architecture

This system was designed using a three-layer architecture consisting of a data layer, business layer, and presentation layer. As such, each layer is independent and separate from the others. The layers are connected by an interface. Any changes in one layer do not necessarily necessitate changes in the other layers.

We used open source applications whenever possible. The team used Struts, an open source model-view-controller (MVC) framework to handle the presentation layer and logic flow.

3.2 MVC Web Frameworks

The MVC design pattern was originally introduced by Reenkaugand when he applied it to the SmallTalk-80 environment. Its purpose is to implement a dynamic software design that simplifies the maintenance and extension of the software application. MVC breaks an application into individual components and defines the interaction between each of them, thereby limiting the coupling between them and allowing each one to focus on its responsibilities without worrying about the others. MVC consists of three categories of components: model, view, and controller. This means that it separates the input, processing, and output, for the applications and molds them into a model–view-controller three-tier structure.

The model portion of the software contains the business workflow application. It is a “black box” that receives the user input and broadcasts the corresponding result. The view represents the user interface. The controller is a dispatcher. It is mainly responsible for

interpreting user input and updating the model in response. Secondly, it registers the view to receive the notifications of changes to the domain model so that the view can refresh itself with the updated data.

Apache Struts[1] is an open source web application framework for developing Java EE applications. It uses and extends the Java Servlets API to encourage developers to adopt the MVC architecture. Struts utilizes a centralized servlet controller, multiple business logic adapters, JSP pages, and a Struts-config.xml configurable file to establish the basic MVC structure of this framework.

Struts is a mature framework with considerable documentation and an active user community. Furthermore, Struts can be easily integrated with the Spring framework. Numerous IDEs have integrated plug-ins for Struts to speed development. For these reasons, we chose Struts as the Web framework for this project.

3.2.1 Struts MVC Architecture

As noted, the Model component employs business logic and interacts with persistence storage to store, retrieve and manipulate data. The view is responsible for displaying the results back to the user. In Struts, the view layer is implemented using JSP. The controller handles all requests from the user and selects the appropriate view to return. In Struts, the ActionServlet does the controller's job. Figure 11 shows a typical Struts MVC architecture.

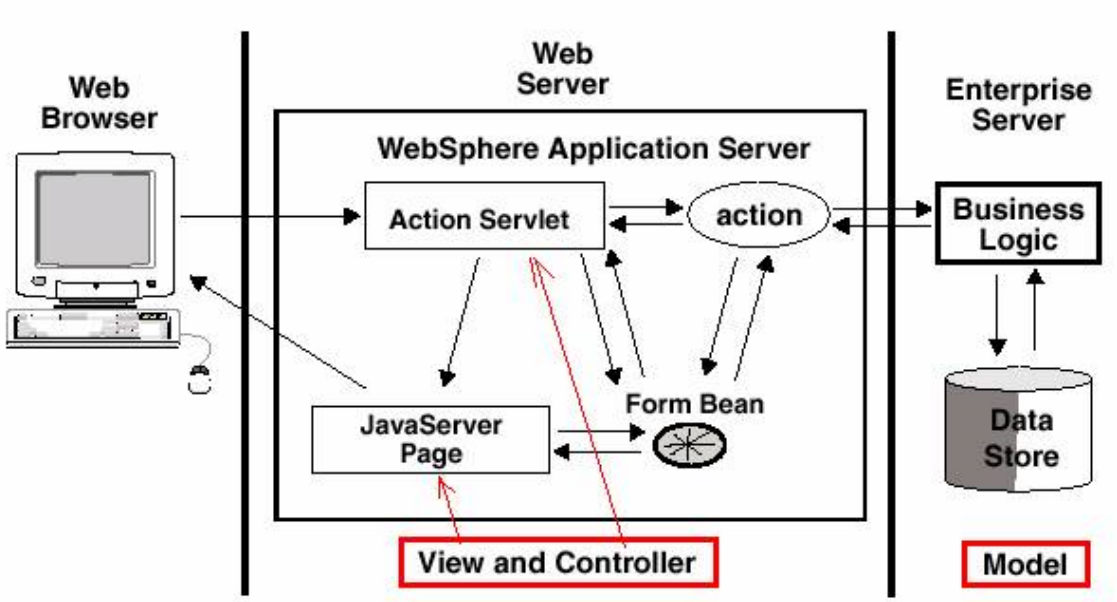

Figure 11.Struts MVC architecture.

3.2.2Struts MVCworkflow

The following events occur when the client browser issues an HTTP request (See Figure 12):

- The ActionServlet receives the request.
- The struts-config.xml file contains the details regarding

  the Actions, ActionForms,ActionMappings and ActionForwards.
- While processing the request, the ActionServletmakes decisions by referring to the

  configuration objects present in the database. The ActionServlet creates this database

  during startup after reading the struts-config.xml file.

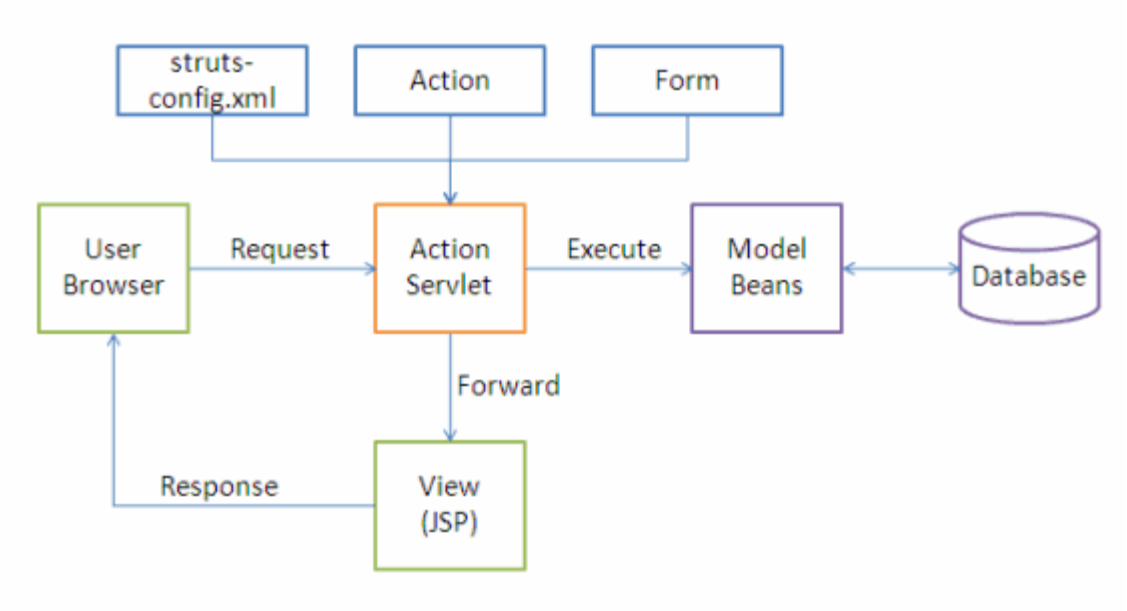

Figure 12.Struts MVC workflow.

When the ActionServlet receives the request, the following tasks are performed:

- The ActionServlet bundles all the request values into a JavaBean class, which extends the StrutsActionForm class.
- The ActionServlet decides which action class to invoke to process the request and validates the data entered by the user.
- The Action class processes the request with the help of the model component. The model interacts with the database and processes the request.
- After processing the request, the Action class returns an ActionForward command to the controller.
- Based on the ActionForward response, the controller invokes the appropriate view.
- The HTTP response is rendered back to the user by the view component.

3.3 Comparison of JavaScript Frameworks

With the concept of asynchronous JavaScript and XML (Ajax[4]), JavaScript has evolved to become far more useful, bringing a whole new level of interactivity to Web-based programming. It is currently one of the most popular programming languages, due to its role as the scripting language of the Worldwide Web. JavaScript is used in millions of Web pages to add functionality, validate forms, and detect browsers, etc. There are many benefits derived from using JavaScript in web applications, including:

- *Use of cache*: The browser can cache the JavaScript file, so the browser does not need to download the same JS again and again.
- *Improved user experience*: JavaScript works client-side as opposed to server-side, so one can do things like validation without sending a request to the server, which indirectly improves the performance of the server.

Due to the complications that arise when trying to provide support for multiple Web browsers, it can be challenging to ensure that JavaScript code is cross-browser compatible. A JavaScript framework is a set of utilities and functions that makes it much easier to write cross-browser compatible JavaScript code. Hundreds of JavaScript frameworks have emerged to allow developers to write powerful, flexible, cross-browser code. Each one of these frameworks has its own pros and cons. The following is a comparison of some of well-known JavaScript frameworks: Ext JS, jQuery, MooTools, Prototype, and YUI. See Table 2 for a summary comparison.

Table 1.Comparison of JavaScript Frameworks.

	jQuery	**Prototype**	**YUI**	**ExtJS**	**MooTools**
Library Size(KB)	10 – 50	40 – 140	30 – 300	150 – 500	65 – 101
Learning Curve	Quick	Slow	Slow	Slow	Slow
Ajax Support	YES	YES	YES	YES	YES
Dom Manipulation	YES	YES	YES	YES	YES
Dom Traversal	YES	YES	YES	YES	YES
Event Handling	YES	YES	YES	YES	YES
JSON	YES	YES	YES	YES	YES
Selectors	YES	YES	YES	YES	YES
Plug-ins	YES	YES	YES	YES	YES

After exploring some of the JavaScript frameworks, it was determined that the jQuery[5] library struck the right balance between size, feature set, and ease of use. It is also an open source framework. Unlike other frameworks, jQuery is easily learned. jQuery has a very large and active community, and it is well documented. jQuery allows developers to write more concise and compact code than other JavaScript frameworks. It is also currently the most widely used framework. There are numerous Internet plug-ins that add features to the framework. Finally, jQuery allows developers to create plug-ins on top of the JavaScript library. Thus, we chose jQuery to perform low-level interaction, animation, and Ajax calls in this project.

3.4 Ajax

Ajax[4]combines several browser-supported technologies, using JavaScript to pass asynchronous requests to a server. The backbone of Ajax is the XMLHttpRequest (XHR) object. The XHR object is a JavaScript object that is used to make asynchronous background requests to the server while a user is viewing a single webpage. These requests can be triggered by a user action or executed automatically by a background JavaScriptobject. Originally, Ajax responses from the server were in XML. Now, with the implementations made in this project, multiple forms of textual responses can be returned including JavaScript object notation (JSON) objects, HTML, JavaScript, and XML.

Using the XHR object across multiple browsers can be tricky and complex. Fortunately, JavaScript libraries have encapsulated the browser compatibility complications and have provided easy-to-use Ajax interfaces. These libraries greatly reduce the complexity of using Ajax in Web applications. jQuery is a JavaScript framework that has an easy-to-use, robust Ajax interface. The jQuery framework lets a programmer provide success and error callback methods. These methods let the programmer choose what to do with server responses.

A complete sequence diagram of an Ajax request is shown in Figure 13. In the sequence diagram, one can see that the same webpage is present and active while the Ajax server request is being made. The XMLHttpRequest object does its work asynchronously. Callback functions are invoked when the asynchronous methods are complete.

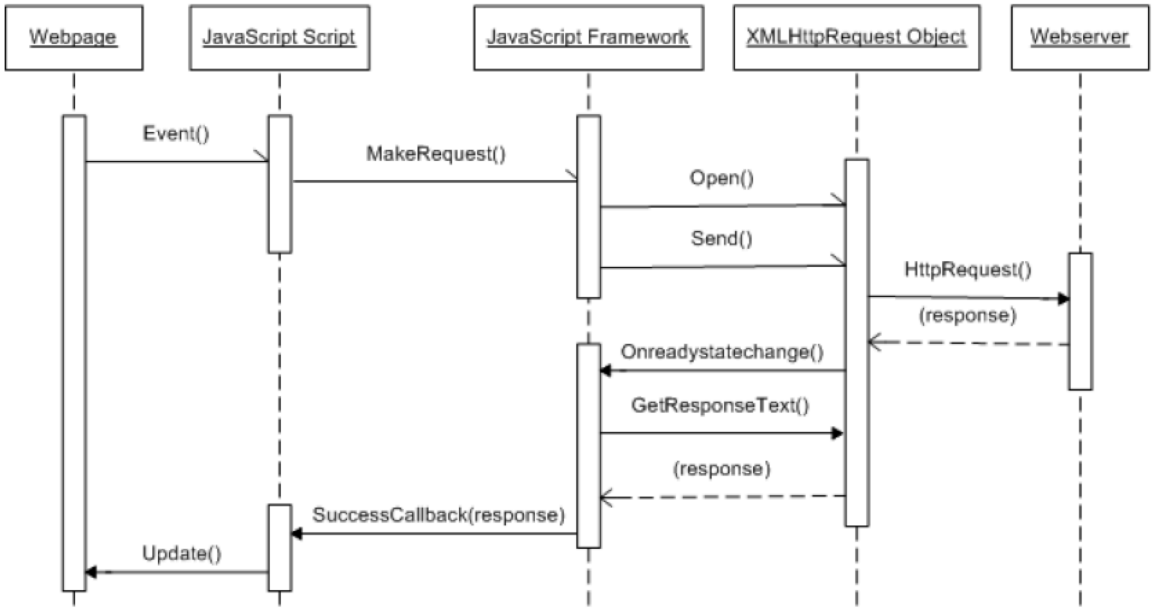

Figure 13.Ajax sequence diagram.

3.5 JavaScript Object Notation

JSON[11] is primarily used to transmit data between a server and Web application, providing an alternative to XML**.** It has two primary advantages over XML when used for Ajax responses:

- JSON can be naturally processed by JavaScript and does not require complicated parsing like XML requires.

- The JSON format is more compact than XML and can make server responses shorter and thus faster.

JSON objects consist of a list of name-value pairs. An example of a serialized JSON object is shown in Figure 14.

```
{
    "userId":    2322,
    "userIdn":   "Smith",
    "courseIds": [255,453,2094,1234],
    "phone": {
                "home": "555-5555",
                "work": "555-5555"
             }
}
```

Figure 14.Example of a JSON object.

3.6 PostgreSQL

PostgresSQL[6] is an object-relational database. It is an open source alternative to commercial databases like Oracle and SQL server. PostgresSQL is now comparable with commercial databases in terms of features, performance, log shipping, clustering, and reliability. It has transactions, views, stored procedures, and referential integrity constraints. It offers more security, reliability, and data integrity than its primary open-sourced competitor, MySQL. Though it is not as fast as some commercial database tools, its speed is sufficient to handle a project with a relatively small database such as this one.

3.7 Socket Programming

A socket[10] is a software endpoint that establishes bidirectional communication between a server program and one or more client programs. The socket associates the server program with a specific hardware port on the machine where it runs so that any client program anywhere in the network with a socket associated with that same port can communicate with the server program.

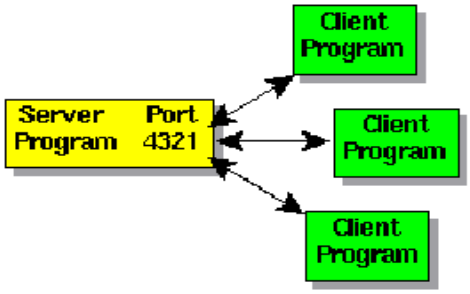

Figure 13.Socket programming.

A server program typically provides resources to a network of client programs. Client programs send requests to the server program, and the server program responds to the requests. One way to handle requests from more than one client is to make the server program multi-threaded. A multi-threaded server creates a thread for each communication it accepts from a client. A thread is a sequence of instructions that run independently of the program and of any other threads. Using threads, a multi-threaded server program can accept a connection from a client, start a thread for that communication, and continue listening for requests from other clients.

3.7.1 Workings of a Programming Server

The main components in the programming question type are:

- Web page (displays the Compile button).

- Action class.
- Programming server.

The steps that take place during compilation are illustrated in Figure 14.

1. The program is submitted to the Web server by clicking the Compile button in the webpage (for example,ProgrammingQuestionAttempt.jsp).
2. The Action class (for example, ProgrammingQuestionAttemptAction.java) in the web server receives the program and sends it to the ProgrammingSocketutil.java class. This class then establishes a connection with the programming server and forwards the answer to it.
3. The programming server, named as SocketServer.jar, then compiles the program.
4. The results of the compilation are sent back to the ProgrammingSocketutil.java class.
5. The ProgrammingSocketutil.java class then sends it to the Action class.
6. The Action class then returns the compilation result to the webpage.

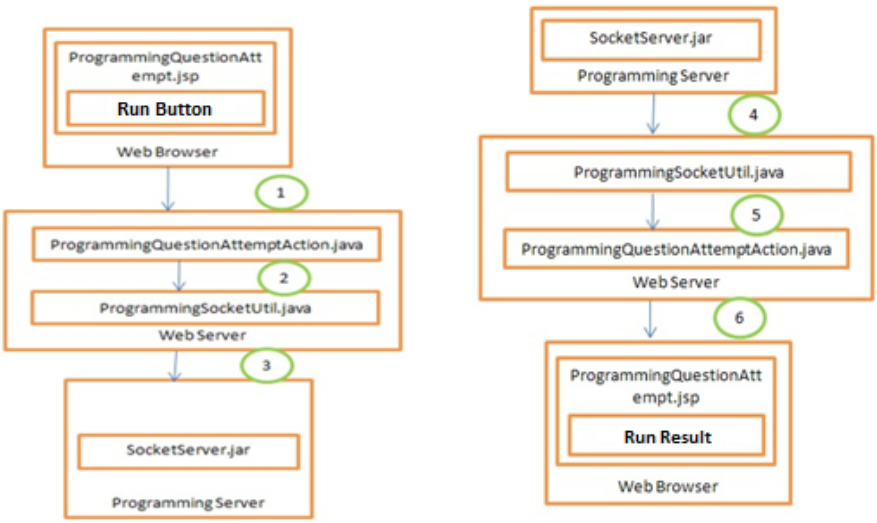

Figure 14.Socket server workflow.

3.8 Version Control System

3.8.1 Subversion

Subversion [12], a version control system, was used in this project. Subversion is a configuration management system that allows multiple users to develop concurrently on a project. One can check out the latest version of the code from the repository, make changes to it, and then commit the files back to Subversion. Subversion keeps track of the changes and then integrates them back into the main code base, or it lets the programmer know if other modifications were made between the last download and subsequent upload.

Subversion repositories can be accessed by the following means:

- Local file system or network file system
- Over HTTP or HTTPS by using the mod_dav_svn module for Apache 2
- Custom SVN protocol, using plain text or over SSH

We adopted the Apache method to access files on the server through HTTP. We found this method to be easier to use than the other means, mainly because HTTP has fewer problems getting through firewalls.

3.8.2 Subclipse

We also used Subclipse, an Eclipse plug-in that provides the ability to interact with a Subversion server, to manipulate the project on a Subversion server in the Eclipse environment. This saved time that would otherwise have been spent going back and forth between Eclipse and Subversion.

CHAPTER 4

TESTING

Testing is involved in every stage of the software life cycle, but testing done at the software development level has its own set of objectives. At this stage, testing becomes the most vital part of the system that runs and compiles the developed programs. Note that in this case, the system is isolated from iNetTest and communication between them is established via sockets. The vulnerabilities involved in socket programming are:

- Sockets should always bind to a specified port.
- Releasing the socket is important once communication is completed.
- Threads should be both created and aborted at specified points.
- Deadlock conditions need to be addressed appropriately.
- If multiple requests are encountered, they should be queued. Otherwise, a new socket should be created for them.
- It is necessary to check the correct functioning of all security-related features of the application.

Methods of testing this add-on system are explained in the following sections.

4.1 Unit Testing

Unit testing [8] is a method by which individual units of source code are tested to determine if they are fit for use. It tests the basic unit of software, which is the smallest testable piece of software and is called “unit,” “module,” or “component.” Unit tests are typically written and run by developers to ensure that the code meets design parameters and behaves as intended.

In this project, every class had its own unit test case for determining if each method within that class worked as expected. JUnit[8], a unit-testing framework for the Java programming language, was used to automate the process.

Sometimes, classes may have references to other classes; thus, testing a class can spill over into testing another class. A common example in a J2EE project is the action class that depends on httpRequest. To test the class, I used the mock object StrutsTestCase, an extension of the standard JUnitTestCase class that provides facilitates for testing code based on the Struts framework, thereby allowing one to test Struts code with or without a running Servlet engine. In this way, one does not need to start the server container for each test of a single action class, which makes the unit test more efficient.

4.2 System Integration Testing

Unit testing by definition only tests the functionality of units themselves. Therefore, it does not catch integration errors or broader system level errors. So, system integration testing [13] must also be performed against the application. System testing tends to affirm the end-to-end quality of the entire system. A system test is often based on the functional requirement specification of the system. Non-functional quality attributes, such as reliability, security, and maintainability, are also checked.

While integrating the socket server with iNetTest, it was necessary to ensure two things: first, that none of the existing features collapsed due to the additional features, and second, to ensure that after integration, all the add-ons worked as expected. These tests were performed by manually checking all question types.

Due to the small scale of this project, we employed an exploratory testing method to test the application for correct behavior prior to release to the end user. In this method, we invited

end users to access and explore the user interface of the application, using as many of its features as possible.

CHAPTER 5

ENHANCEMENT RESULTS AND FUTURE WORK

The enhancements made to the iNetTest system were designed and implemented over a period of several months. Features were added one by one to the iNetTest system, as required. All security issues that were introduced to the system were resolved and tested successfully. The enhancements have made many improvements to the user interface and steered the overall direction for the development of the iNetTest system.

5.1 Overall Improvement

We added three main features to the iNetTest system: run programs through iNetTest, improve the group grading of programming questions, and add support to run the scripting languages. Adding these features introduced many vulnerabilities to the system, which were also resolved successfully. The enhancements made the interface more functional and interactive.

The capability to run programs through iNetTest provides a new way to test computer programming, and removes the need to install any compiler or IDE on students' individual computers. With the new interface, related test answers and code are easily accessible at the same time as other instructor data. Students can save time when working with this kind of interface, given that they do not need to store any files or code on their computers.

Our use of Ajax and JSON makes grading of individual assignments easier, and it greatly enhances group grading. The new interface for group grading saves a lot of time by providing Run All and Compile All features. These two features facilitate accessing answers of all students at once on the same page. The iNetTest system puts scores on the same page; this makes grading much more convenient. In addition, iNetTest is designed to kill any program with infinite loops

and protect the server against bad code. As well as grading for individual students, iNetTest has been modified to run the student's program and show the instructor's test code in the same place.

The idea was to provide the instructor with as many options as possible for computer programming tests. In addition, the number of programming languages supported was increased. Finally, iNetTest was enhanced in a way to support scripting languages(Ruby, Python, Perl, and PHP).

5.2 Conclusion

This report discusses the design, development and implementation of three iNetTest features. This is a Web-based application that can be used by students, instructors, staff, and the general public. We designed it with generality and extensibility in mind. Therefore, with a few modifications, it could be used for any operation. It is a fully featured system that allows instructors to easily create and grade tests, as well as to allow students to take tests.

We used an object-oriented development approach in building this system. This approach facilitated code reuse and simplified the development of a manageable system. The OO approach has also made the code less intrusive and less coupled, while making the entire system more flexible and easier to extend.

We built this project using a three-layer architecture. This structure separates the presentation layer, business layer, and data layer from each other. Any change in one layer does not necessarily cause changes to the other layers.

While building this system, we used extreme programming (EP)[9], one of the most popular agile software development processes. As such, EP enables developers to confidently respond to changes in customer requirements, even late in the life cycle. Managers, customers, and developers are all equal partners in a collaborative team. EP implements a simple and

effective environment that enables teams to be highly productive. Using EP, programmers constantly communicate with customers. They keep their design simple and clean, while delivering the system to the customer as quickly as possible with the changes implemented.

We used Eclipse to develop this project. Eclipse is one of the most popular open source integrated development environments (IDEs). It has numerous features, such as syntax checking, code completion, code generation, and code folding. In addition, it supports Ant, CVS, and JUnit—all of which we used in the development of this project. One of the most useful features of Eclipse is automated refactoring. Also, Eclipse is faster and more stable than other IDEs. All of these factors combine to help developers write clean and efficient code easily and quickly.

The team managed version control with a system called Subversion (SVN). We configured an SVN server and integrated it with Apache so that we could access the server through HTTP. SVN allows one to keep track of the change history of the source code, and it allows one to easily roll back to a previous version if needed. It also facilitates development when two or more developers work on the same project.

5.3 Future Work

The programming questions of iNetTest can be enhanced in many ways in the future. These future enhancements and extensions could include:

- Auto code completion or suggestions could be added to the online editor.
- Auto code generation is also a suggestive feature in all IDE; it could be implemented in the iNetTest editor.

Both of these enhancements would provide students with an interactive and friendly user interface.

REFERENCES

1. The Apache Software Foundation. Struts Framework.http://struts.apache.org/. Accessed month year.
2. Wikipedia. Inversion of control. http://en.wikipedia.org/wiki/Inversion_of_control. Accessed September2011.
3. Wikipedia. Java Persistence API. http://en.wikipedia.org/wiki/Java_Persistence_API. Accessed September2011.
4. Wikipedia. Ajax. http://en.Wikipedia.org/wiki/Ajax_%28programming%29.Accessed September 2011
5. The jQuery Project. jQuery. http://jquery.com/. Accessed September2011.
6. PostgreSQL.PostgreSQL 9.0 http://www.postgresql.org/. Accessed September2011.
7. Wikipedia. Unit testing. http://en.wikipedia.org/wiki/Unit_Testing. Accessed September 2011.
8. JUnit.org. JUnit. http://www.junit.org/. Accessed October2010.
9. Wells D. Extreme programming: A gentle introduction. http://www.extremeprogramming.org/.Accessed September2011.
10. Oracle. Socket Programming.http://www.oracle.com/technetwork/java/socket-140484.html. Accessed month year.
11. Wikipedia. JSON. http://en.wikipedia.org/wiki/JSON/. Accessed month year.
12. Wikipedia. Subversion.http://en.wikipedia.org/wiki/JSON. Accessed month year.
13. Wikipedia. System integration testing. http://en.wikipedia.org/wiki/System_integration_testing. Accessed month year.