Computer Science ~
Software Engineering

Celebrating 40 Years
of Excellence!

College of Engineering
California Polytechnic
State University

# An Evaluation of Potential Technologies for a Web Based Development Environment for CS1 Students

Prepared for: David Janzen, Computer Science Professor

Prepared by: Manuel Garcia, Computer Science Student

December 11, 2009

Computer Science ~
Software Engineering

*Celebrating 40 Years of Excellence!*

College of Engineering
California Polytechnic
State University

# Table of Contents

# 1. Abstract

The goal of this senior project was to design a web-based integrated development environment that would incorporate Web-CAT. Web-CAT is an automated testing tool designed by Stephen Edwards at Virginia Tech University, to help beginning students learn test-driven development. This tool runs on a server where a professor assigns usernames and passwords to students so they can submit code. This program can assist professors in grading, and students in showing errors in code. The development environment design was kept simple to create less confusion for beginning programming students. This development environment will be online to give all students universal access and remove any operating system dependencies. The development environment will be developed using Adobe Flex and run on Adobe Flash through any browser that supports Flash.

## 2. List of Figures

**Figures**

## 3. Introduction

This project was developed to provide beginning students an environment that is simple to develop code and accessible to beginning users.  Creating a development environment in the web browser is accessible for everyone and can be designed for beginning students.  This environment would use new web technologies to provide a user-interface specific for beginning students.  This project would test the feasibility of creating a development environment through the web.  Professor Janzen recommended that Web-CAT be incorporated because Web-CAT provides a system to allow test driven development to be incorporated in projects.  The tool that will be used to develop the web environment is Adobe Flex using the Flex Builder development environment.  A server dedicated to Web-CAT will be necessary for students to submit their work and for professors to grade them.  To begin the development, we studied the development environments designed for beginning students.

## 4. Comparison of Existing Systems

### 4.1 BlueJ

BlueJ is an integrated development environment designed for beginning students.  BlueJ emphasizes object oriented programming and the use of Java.  It provides a visualization of the class structure to assist students in object oriented programming. BlueJ includes a compiler, editor, debugger and virtual machine. The user-interface is very simple.  It has one row of commands to manipulate text and compile the program. The coding area provides syntax highlighting and diagrams are present to show how classes interact with one another.  There is an area to see documentation that is generated from the code. The development environment allows objects to be created interactively and methods to be called interactively. This keeps professors from showing beginning students' concepts that cannot be explained earlier in their programming development.

## 4.2 GreenFoot

Greenfoot is an integrated development environment with a framework for creating two-dimensional grid assignments in Java. The development environment is designed for novice programmers. Greenfoot focuses on assisting students whose programming exercises involve a visual element and controlling the execution of a loop. Greenfoot has a window to view the code and another window to view code being executed. This development environment gives you further control to run a single method. Greenfoot provides syntax highlighting and diagrams that show classes.

## 4.3 JGrasp

JGrasp is a development environment designed to provide software visualizations. These visualizations are generated automatically and are to help the user understand the software being developed. The development environment provides tabbed browsing so multiple files can be opened. The interface provides a separate window to analyze errors that are found in executed code. There is a pane that shows a directory and the files that are contained in this directory. JGrasp includes many settings with a row of buttons to provide different functionality. This program is more advanced and may not be suitable for beginning programming development.

## 4.4 TuringsCraft

TuringsCraft is a web-based interactive programming exercise system for introductory students. This system provides questions for students to answer in a web-based text environment. This environment provides no ability to create code. The coding provides no syntax highlighting and is designed for small pieces of code, not a whole program. The system provides feedback by checking your answer and comparing it to a list of possible answers.

### 4.5 JavaBat

JavaBat is a free website that provides Java coding problems to build coding skills. Students select a problem and a webpage is provided with the problem and an area to place the solution code. When the solution code is executed by the click of a button, the coding problem provides immediate feedback on a screen next to the coding area. The coding area provides no syntax highlighting and is more of an HTML text area. This website was designed to practice programming concepts. The problems provided in JavaBat are short problem statements usually found in exams. This interface provides no way of developing code other than to solve the coding problems.

## 5. Requirements

### 5.1 Web-CAT Requirements

Web-CAT is a plug-in-based web application that supports electronic submission and automated grading of programming assignments. Web-CAT is developed in four subsystems known as the Core, Grader, Admin, and Example. The Core subsystem provides the tab-based navigation system to navigate through the web-site. The core data model provides the information for professors to create courses, course sections, create TAs and create a roster of students. The Admin subsystem provides a very direct way to view the database to search, view, or update database entities. The Grader subsystem supports automated program grading and supports assignments where students provide their own testing code. This encourages test-driven development where students write small unit tests for each piece of code they add. The Grader subsystem tests validity, completeness, and code correctness when a student submits their test cases and solutions. The Example subsystem provides examples to help users become familiar with all the features that Web-CAT provides.  Web-CAT needs to be installed on a server with an operating system that supports Java 1.5 or above, Apache Tomcat 5.5 or above, MySQL, and a version of Perl with HTML and XML parser modules.

## 5.2 User-Interface Requirements

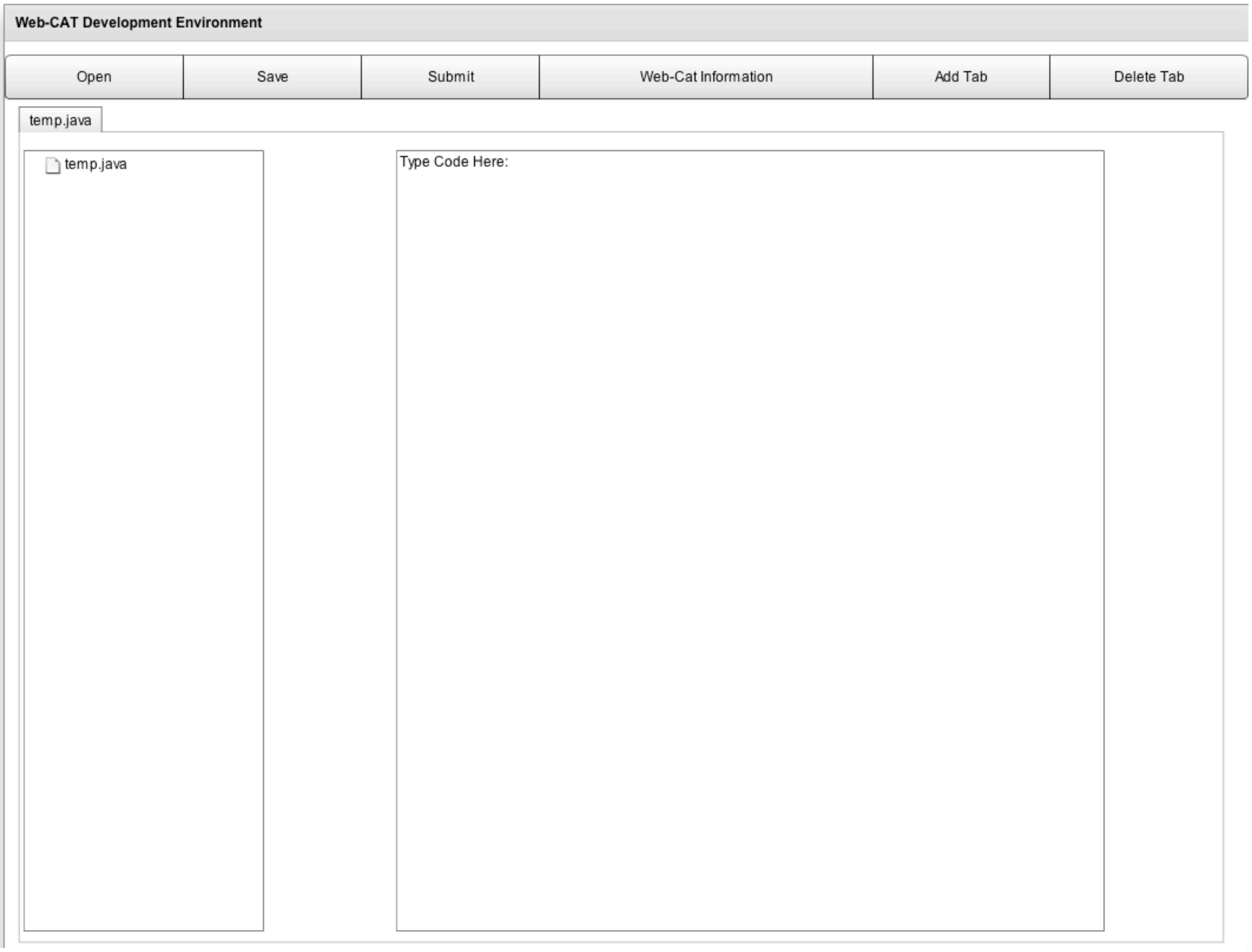
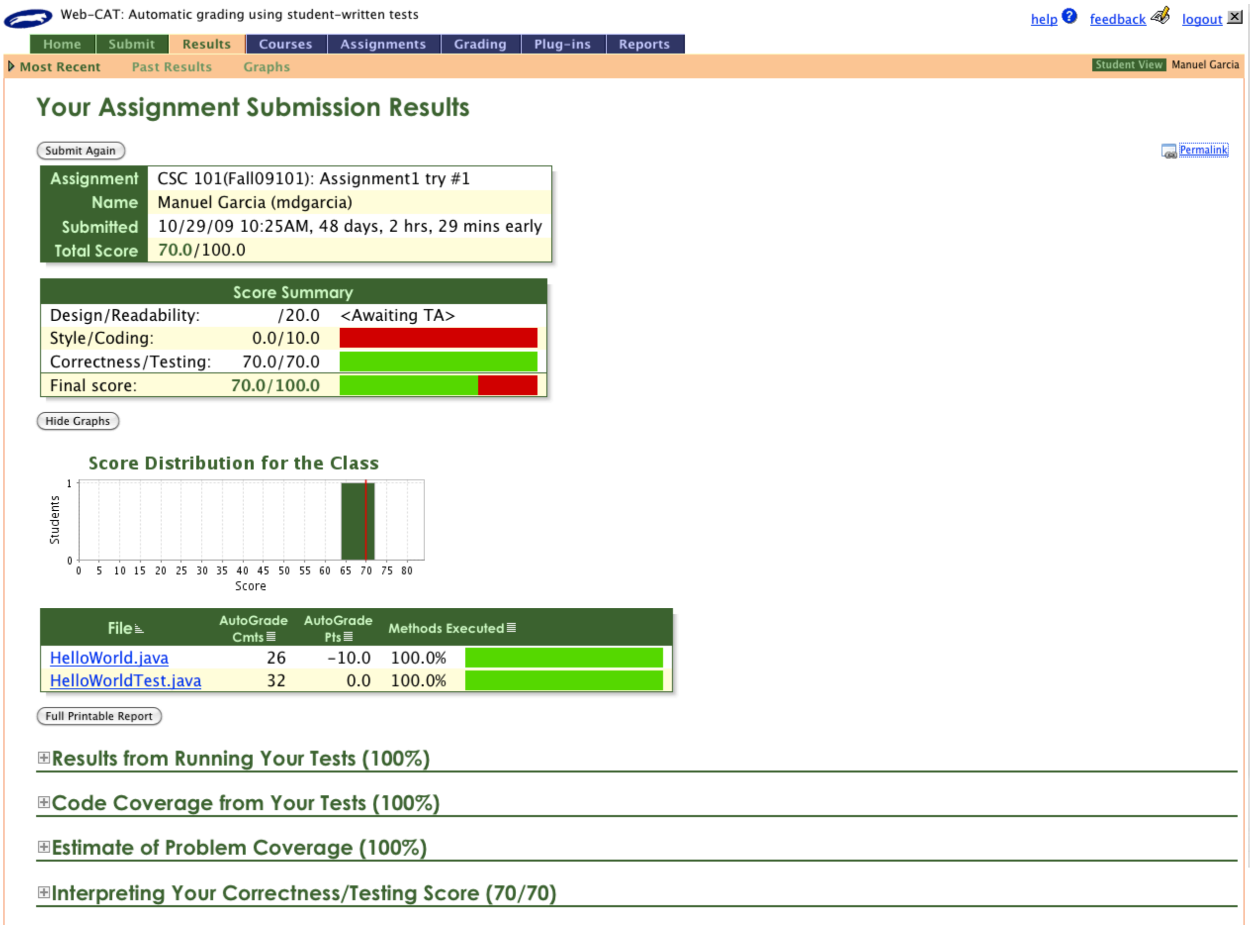


*Figure 1*

Figure 1 is the user-interface for the Web-Based Development Environment.  The layout will include a tool bar with six buttons.  The Open and Save button will provide dialog boxes to open and save files on a user's computer.  The Submit button will provide a dialog box to select files that the user would like to zip.  After the files are selected, the information provided from the Web-CAT Information button is used to submit the files. A browser window is opened to show the results of the submission. The Web-CAT Information button will hold the credentials of the user that are needed when the code is submitted. The user-interface is the front-end application and Web-CAT is the back-end application. Web-CAT takes the submitted files from the user-interface and runs them against their test code and the professor's test code. Next Web-CAT will provide test results of the submitted code and a grade. Students are graded in three areas: Design/Readability, Style/Coding, Correctness/Testing. A final grade is given based on the how each three are weighted. The Web-CAT information is displayed in Figure 2.



Your Assignment Submission Results

| | |
|---|---|
| Assignment | CSC 101(Fall09101): Assignment1 try #1 |
| Name | Manuel Garcia (mdgarcia) |
| Submitted | 10/29/09 10:25AM, 48 days, 2 hrs, 29 mins early |
| Total Score | 70.0/100.0 |

| Score Summary | | |
|---|---|---|
| Design/Readability: | /20.0 | <Awaiting TA> |
| Style/Coding: | 0.0/10.0 | 🟥 |
| Correctness/Testing: | 70.0/70.0 | 🟩 |
| Final score: | 70.0/100.0 | 🟩🟥 |

| File | AutoGrade Cmts | AutoGrade Pts | Methods Executed | |
|---|---|---|---|---|
| HelloWorld.java | 26 | -10.0 | 100.0% | 🟩 |
| HelloWorldTest.java | 32 | 0.0 | 100.0% | 🟩 |

⊞Results from Running Your Tests (100%)

⊞Code Coverage from Your Tests (100%)

⊞Estimate of Problem Coverage (100%)

⊞Interpreting Your Correctness/Testing Score (70/70)

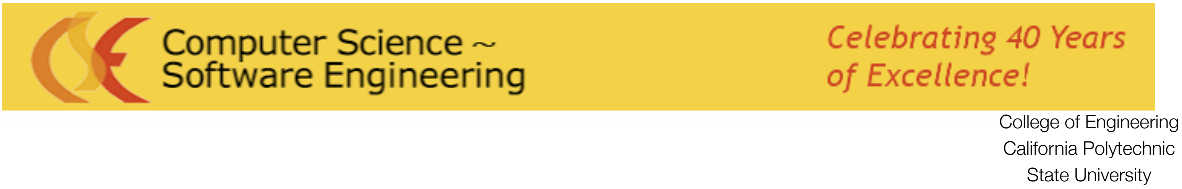


*Figure 2*

There will be two buttons to support tab functionality. The Add Tab button will add a tab to the development environment and the Delete Tab deletes the current tab in focus. These tabs will allow multiple development environment instances to be open to allow beginning programmers to work on their test code and project code. This will reinforce the methodology of test-driven development. Figure 3 shows the dialog box that appears with the necessary information that Web-CAT requires. The form requires a User Name, Password, Course ID, Department ID, and Assignment ID. The Web-Based Development Environment was developed on Adobe Flash using Adobe Flex 4.

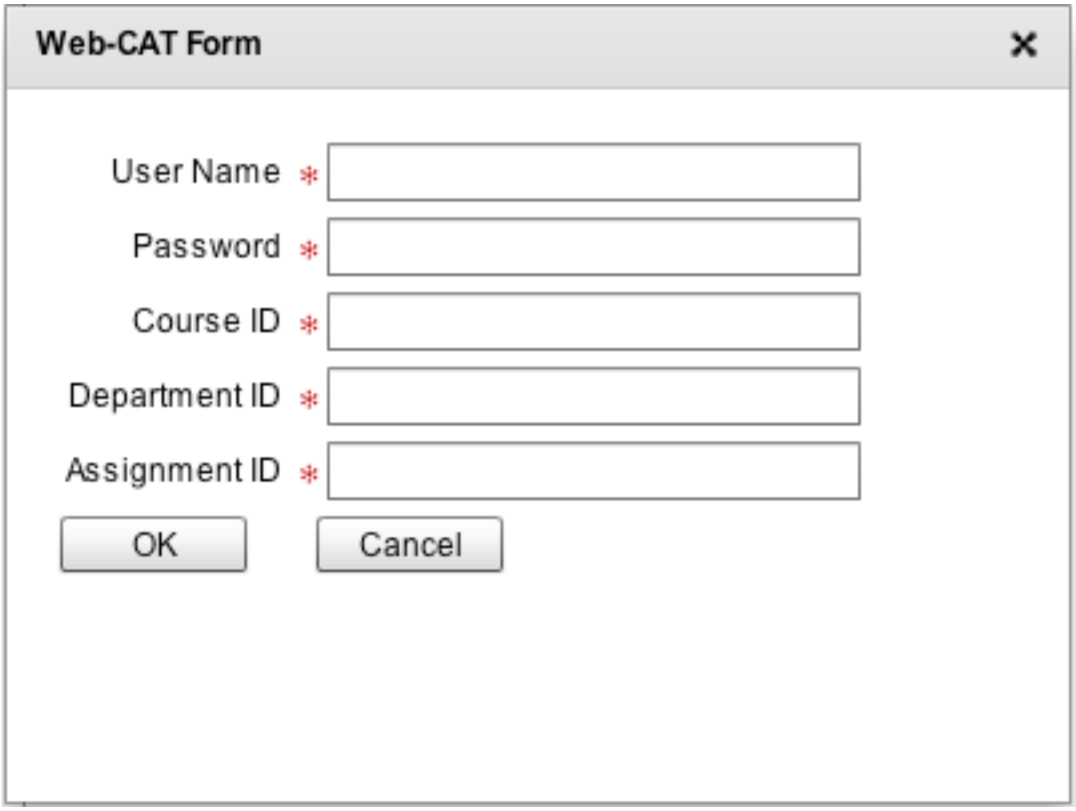


*Figure 3*

# 6. Results

## 6.1 Product Completed

### *6.1.1 Introduction*

A year of development into the project returned a web software package. This package included: a web-based development environment, a submission webpage, and a Web-CAT Server. These three components come together to form the first web-based development environment. The project provides a framework towards developing a basic user-interface for beginning students and how to expand further on the idea of developing a web-based development environment.

### *6.1.2 Communication Design*

The design involved the development environment and the Web-CAT server communicating with each other. Figure 4 shows the original diagram of how the user-interface and Web-CAT server communicate with each other.

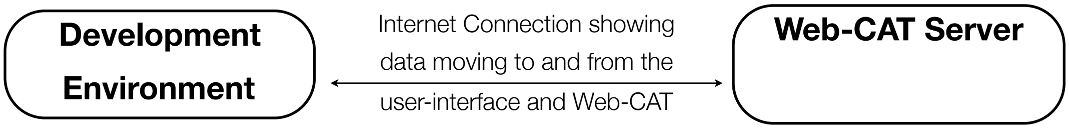

Development Environment
Internet Connection showing data moving to and from the user-interface and Web-CAT
Web-CAT Server

*Figure 4*

The development environment could send code and receive results from the Web-CAT server. Figure 5 shows how the development environment and Web-CAT communicate with each other now. The development environment no longer communicates with the Web-CAT server. The submission web-page sends submitted code to the Web-CAT server and receives nothing back. The Web-CAT server doesn’t send any data back, but displays the results in its own web-page. This method still allows us to communicate with the Web-CAT server directly.

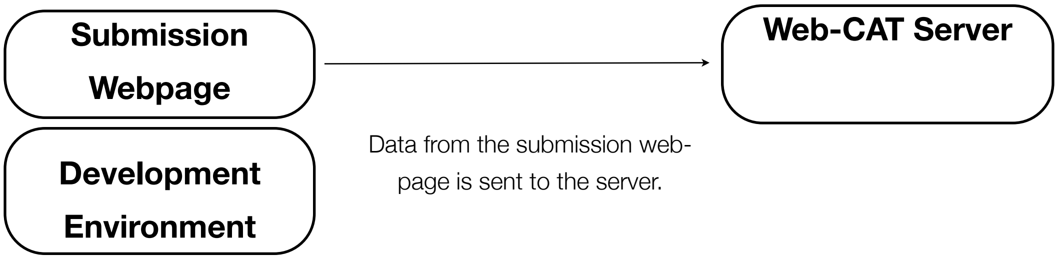


*Figure 5*

### *6.1.3 Installing Web-CAT Results*

The Computer Science Technology staff provides a senior project student that requires a server, a virtual machine with Centos 5 included. I installed Java, Apache Tomcat, Perl, and MySQL onto the server. Then I loaded the Web-CAT web application on Tomcat and followed the procedure to setup Web-CAT. I created an institution, added courses, and assignments and tested to make sure the Web-CAT installation could handle student submissions. As of today, it is being used by professors to take advantage of its automated grading capabilities.

### *6.1.4 User-Interface Results*

Figure 6 is the user-interface for the Web-Based Development Environment. The first difference between figure 1 and figure 6 is there are only five buttons in the new user-interface. The first button labeled Open, provides a dialog box to open a file from the user's computer. The second button labeled Save, provides a dialog box to save code onto the users computer. The third button labeled Submit, opens up a web browser to a web-page that allows user's to remotely submit their work to Web-CAT. The Web-CAT information button was removed from figure 1 because the information is no longer needed to be stored in the development environment. This information is placed in the web-page that submits work to Web-CAT. The fifth button labeled Add Tab, adds a tab to the development environment so multiple files can be opened.

The sixth button labeled Delete Tab, deletes the current tab in focus by the development environment. The code that was used to create the Web-Based Development Environment can be found at Appendix 8.2.

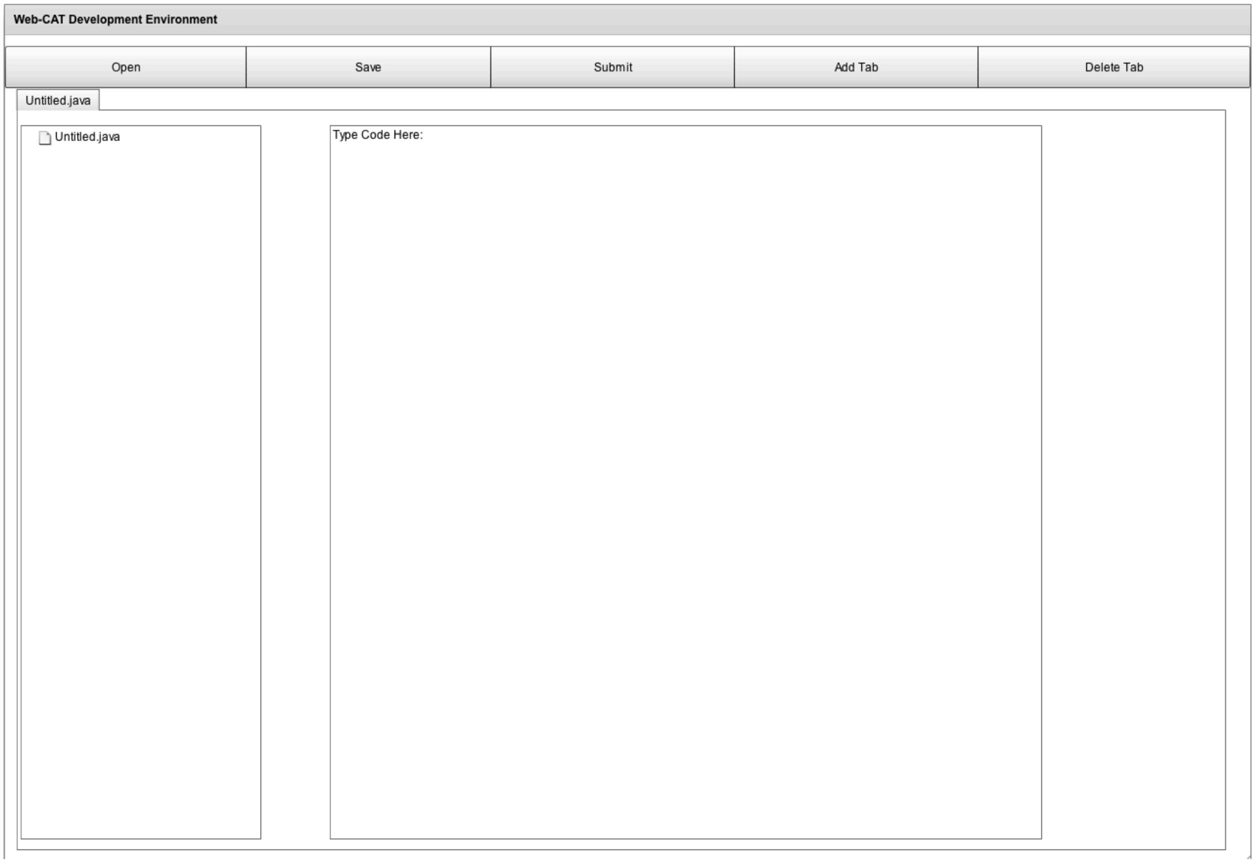
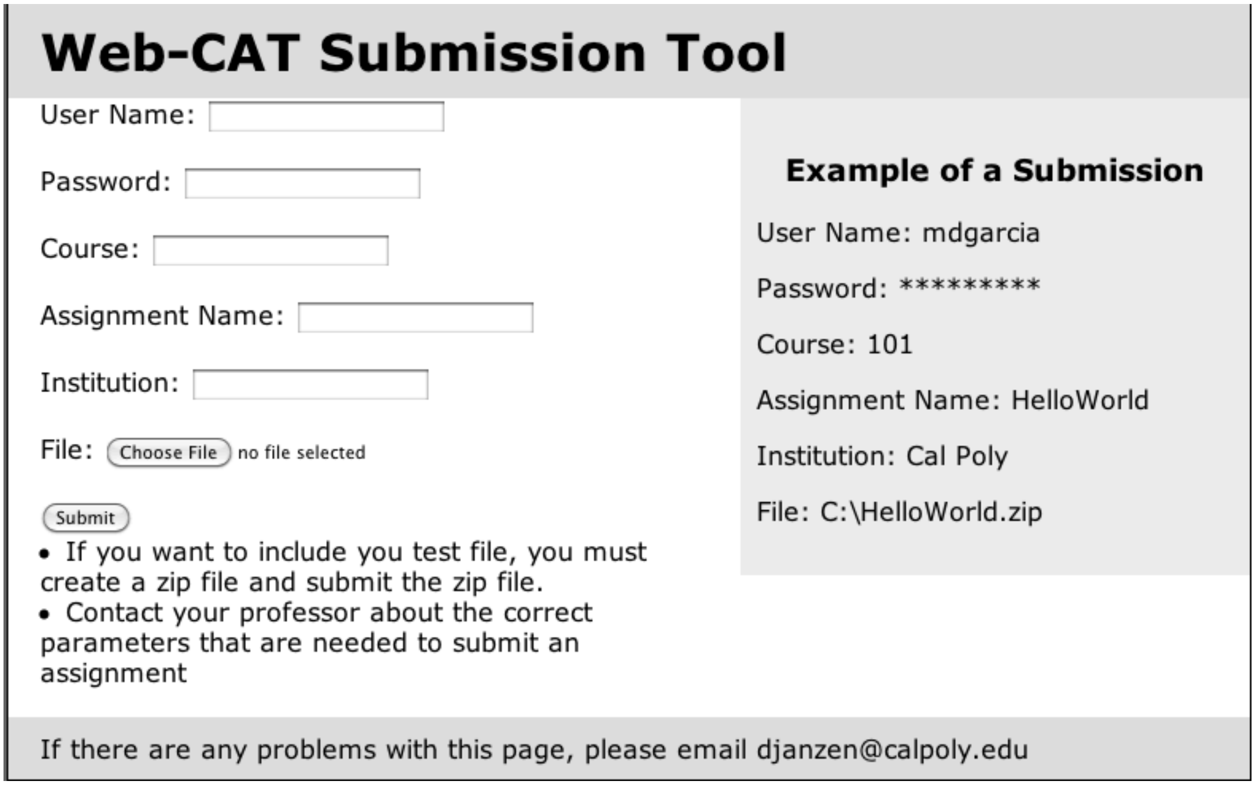


*Figure 6*

### *6.1.5 The Submission Webpage*

Figure 7 shows the Web-CAT Submission Tool. This tool has six fields. The first five fields are information necessary to submit remotely to Web-CAT. These fields are the User Name, Password, Course, Assignment Name, and Institution. The last field allows you to upload a zip or jar file. These two file types are necessary to hold the test and submission code a student might submit. Once the Submit button is clicked, the file is sent and the webpage directs you to the results of your submission. The Submission Webpage is located at http://cslvm156.csc.calpoly.edu/Submitter/submitter.html. The code is given at Appendix 8.3.



## Web-CAT Submission Tool

User Name:

Password:

Course:

Assignment Name:

Institution:

File: Choose File no file selected

Submit

- If you want to include you test file, you must create a zip file and submit the zip file.
- Contact your professor about the correct parameters that are needed to submit an assignment

**Example of a Submission**

User Name: mdgarcia

Password: *********

Course: 101

Assignment Name: HelloWorld

Institution: Cal Poly

File: C:\HelloWorld.zip

If there are any problems with this page, please email djanzen@calpoly.edu

*Figure 7*

## 6.2 Things That Went Wrong

### *6.2.1 Web-CAT installation issues*

In the beginning, there were problems with setting up the server to support Web-CAT. Centos 5 is a free enterprise class operating system derived from Red Hat. The first task before installing Web-CAT was to update the GNU GCJ version of Java from 1.42 to 1.5. After the upgrade, I continued to install Web-CAT. With this version of Java, I would continually receive errors and the system would crash from the installation of Web-CAT. The Web-CAT website noted that this version and Sun's OpenJDK project are not supported by Web-CAT. Web-CAT requires the Official Sun Java Development Kit. This version of Java is not a normal dependency of Centos 5 and required to be installed manually. After following instructions that can be found on Appendix 8.1, I finished installing the Official Sun Java Development Kit, Apache Tomcat, and MySQL. I began the installation of Web-CAT and had it up and running to begin accepting submissions.

### *6.2.2 The New Submission Process*

The submission process in the original requirements could not be fulfilled. To submit a project remotely to Web-CAT requires a multipart post request. A multipart post method follows two steps, setting the request data and retrieving the response data. Adobe Flex can send a post request. Adobe Flex couldn't send a multipart post request through either XHTML or Action Script 3. This means that the development environment couldn't submit the files to the Web-CAT server. To overcome this limitation, a webpage called the Web-CAT Submission Tool was created to allow users the ability to submit to Web-CAT without having to login to the system.

### *6.2.3 Technical Limitations of Action Script 3*

With a new submission process, I searched for a way to automate the process. The webpage required a zip file for users that had to submit more than one file to Web-CAT. I searched for Action Script 3 classes that could zip files. During my search I found that none of these classes could develop a zip file consistently. I found null pointer errors every time a file was zipped. I saw no other possible way that the submission process could be automated without the user having to zip the files on their own computer.

# 7. Future Work

## 7.1 HTML 5

This project in the future can be developed using HTML 5. JavaBat is a great example of a template that can be used to create a web-based development environment. You have two text areas side by side. One side is where you type the code and the other side provides a continuous update of your code with syntax highlighting. The syntax highlighting is done in Javascript with a program like SHJS (Syntax Highlighting in JavaScript). I think this is important for CS1 students because syntax highlighting provides a better view of code to find errors. Most browsers provide tabbed browsing; each web-page is an instance of the development environment so opening multiple tabs will provide you with multiple development environments. The ability to run code inside the environment is important to make it a viable development environment. Test data needs to be allowed to run to incorporate Test-Driven development. The environment needs to load the test data or have the user point to the location of the test data to be run against the code in the environment. The environment requires the ability to open or save work from the web-page. Last, the environment needs to work offline if the user loses internet connection for some reason.

### 7.2 Google Gears

Google Gears is an open source browser extension that allows applications that web developers create to run offline. This provides an interface with the computer and the ability to save or load files from a computer. Also, Google Gears API allows work to be done if the web-page goes offline. Google may drop Gears in the future if HTML 5 standards allow web applications to run offline. This is important because Google Gears doesn't work in Mac OSX. Google Gears would keep the environment from working in any operating system.

## 8. Conclusion

This project was a great experience for me in the arena of web programming. This is a field that is changing and evolving from the products that people want to place on the web. The web will transform the landscape by removing the need to have applications installed on our computers. With wireless networks becoming cheaper and faster, web development only has room to grow from its limitations to endless possibilities. At the moment we are not there yet from the troubles that were endured in this project.  I am excited about the future of web programming and happy to have participated in the senior project.  Last I would like to thank Professor Janzen for pushing me to find the solutions that would help bring this project to life.  His dedication to the field of computer science is evident from the reality of this project.

# 9. Appendix

## 9.1 Web-CAT Server Installation Instructions

How to install Java on a Centos 5 Machine ¶
Download the latest jdk binary from http://java.sun.com/javase/downloads/index.jsp
Choose Java SE Development Kit (JDK) Update xx then select platform, please select linux or linux x64 depending on your centos version you‘ve install in your machine. then select [jdk-xxx-linux-xxx.bin] please do not select the rpm one.
To Download through the command line, type:

```
[server]# wget http://xxxxx.jdk-xxx-linux-xxx.bin
```

Change the file permissions to be executed:

```
[server]# chmod +x jdk-xxx-linux-xxx.bin
```

Than extract the file:

```
[server]# ./jdk-xxx-linux-xxx.bin
```

In the screen appear EULA just press spacebar ’till it ask yes or no, just type yes then move the extracted folder to /usr/lib/jdk1.6.0_xx

```
[server]# mv jdk1.6.0_xx /usr/lib/
```

Link the JRE directory to recreate the JAVA_HOME

```
[server]# ln -s /usr/lib/jdk1.6.0_xx/jre /usr/lib/jre
```

Define Java home to that path

```
[server]# export JAVA_HOME=/usr/lib/jre
```

Then type this command to add it to the sbin list: [server]# /usr/sbin/alternatives --install /usr/bin/java java /usr/lib/jdk1.6.0_xx/jre/bin/java 2
Next you enter this command to enter the sbin list

```
[server]# /usr/sbin/alternatives --config java
```

You'll be asked to choose which Java software you want for your system. Press the 2 key and hit enter.
Type this command to see if everything looks good and your system has a new Java Environment:

```
[server]# /usr/sbin/alternatives --display java
```

You can type this command to see the version of your Java Runtime Environment: java -version
Mine looks like this:

```
java version "1.6.0_12"
Java(TM) SE Runtime Environment (build 1.6.0_12-b04)
Java Hotspot(TM) 64-Bit Server VM (build 11.2-b01, mixed mode)
```

Java Installation is complete

How to install Tomcat 6 on a Centos 5 Machine ¶
Download tomcat 6 from http://tomcat.apache.org/download-60.cgi, choose binary distribution core with tar.gz,type:

```
[server]# wget http://xxx/bin/apache-tomcat-6.0.xx.tar.gz
```

Extract it with: [server]# tar -xzf apache-tomcat-6.0.xx.tar.gz

Move it to /opt/tomcat-6 [server]# mv apache-tomcat-6.0.xx /opt/tomcat-6
Create tomcat user so the server wont run it with root priveleges:

```
[server]# adduser -s /sbin/nologin tomcat
[server]# chown -R tomcat. /opt/tomcat-6
[server]# chown -R tomcat. /usr/lib/jre
[server]# export CATALINA_HOME=/opt/tomcat-6
[server]# export CATALINA_BASE=/opt/tomcat-6
[server]# export CATALINA_OPTS=”-Xms512m -Xmx512m”
```

Open up the /etc/profile so you can save the variables in case you reboot the machine:

```
[server]# vi /etc/profile
```

Scrolldown to the end of file and type this export JAVA_HOME=/usr/lib/jre

```
export CATALINA_HOME=/opt/tomcat-6
export CATALINA_BASE=/opt/tomcat-6
export CATALINA_OPTS=”-Xms512m -Xmx512m”
```

How to install Mysql 5 on centos 5 Machine ¶
Open up a command window and type:

```
[server]# yum install php-mysql mysql mysql-server
```

Next, change the run levels by typing:

```
[server]# /sbin/chkconfig --levels 235 mysqld on
```

To turn on Mysql type this command:

```
[server]# /etc/init.d/mysqld start
```

## 9.2 Web-CAT Environment Source Code

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
                              xmlns:s="library://ns.adobe.com/flex/spark"
                              xmlns:mx="library://ns.adobe.com/flex/halo" minWidth="1024" minHeight="768"
creationComplete="init()" >
        <fx:Declarations>
                <!-- Place non-visual elements (e.g., services, value objects) here -->
        </fx:Declarations>
        <fx:Script>
                <![CDATA[
                        import mx.controls.TextInput;
                        import mx.containers.Form;
                        import mx.controls.Alert;
                        import mx.rpc.http.HTTPService;
                        import mx.rpc.events.ResultEvent;
                        import mx.rpc.events.FaultEvent;
                        import mx.core.UIComponent;
                        import mx.core.mx_internal;
                        import mx.core.IFlexDisplayObject;
                        import mx.managers.PopUpManager;
                        import mx.containers.TitleWindow;
                        import mx.containers.FormItem;
                        import mx.events.IndexChangedEvent;
                        import mx.rpc.mxml.Concurrency;
                        import mx.core.Container;
                        import mx.collections.ArrayCollection;
                        import mx.controls.ButtonBar;
                        import mx.containers.TabNavigator;
                        import mx.controls.Tree;
                        import mx.containers.Panel;
                        import mx.controls.Spacer;
                        import mx.controls.TextArea;
                        import mx.containers.HBox;
                        import mx.containers.Canvas;
                        import mx.controls.Button;
                        import mx.controls.Label;
                        import mx.containers.VBox;
                        import mx.events.ItemClickEvent;

                        private var fileReference:FileReference = new FileReference();
                        private var treeName:ArrayCollection;
                        private var tabListLength:int = 0;
                        private var menuBar:ButtonBar;
                        private var currentCanvas:Canvas;
                        private var currentCanvasIndex:int;
                        private var canvasBox:Canvas;
                        private var treeFile:Tree;
                        private var midSpacer:Spacer;
                        private var codeArea:TextArea;
                        private var tabLabel:Label;
                        private var inputForm:TitleWindow;
                        private var ev:ItemClickEvent;
                        private var tcc:Event;
                        private var service:HTTPService;
                        private var buildPopUp:TitleWindow;
                        private var buildForm:Form;

                        private function init():void
                        {
                                inputForm = new TitleWindow();
                                inputForm.width = 400;
                                inputForm.height = 300;
                                inputForm.title = "Web-CAT Form";
                                inputForm.showCloseButton = true;
                                menuBar = new ButtonBar();
                                menuBar.dataProvider = new Array("Open","Save","Submit","Add Tab","Delete
Tab");

                                menuBar.addEventListener(ItemClickEvent.ITEM_CLICK,menuBarHandler);
                                menuBar.y= 11;
                                menuBar.percentWidth=100;
                                menuBar.percentHeight=5;
                                treeName = new ArrayCollection();
                                treeName.addItem("Untitled.java");
                                currentCanvas = buildCanvas(0);
                                currentCanvasIndex = 0;
                                tabNav.addChild(currentCanvas);
                                tabNav.addEventListener(Event.TAB_INDEX_CHANGE,findNewTab);
                                panelBox.addElement(menuBar);
                        }

                        private function buildCanvas(i:int):Canvas
                        {
                                canvasBox = new Canvas();
                                canvasBox.percentWidth=100;
                                canvasBox.percentHeight=100;
                                canvasBox.label=treeName[i].toString();
                                treeFile = new Tree();
                                treeFile.top = 5;
                                treeFile.left = 3;
                                treeFile.percentHeight=98;
                                treeFile.percentWidth=20;
                                treeFile.dataProvider=treeName;
                                codeArea = new TextArea();
                                codeArea.percentWidth = 59;
                                codeArea.percentHeight= 98;
                                codeArea.left = 319;
                                codeArea.top = 5;
                                codeArea.editable=true;
                                codeArea.enabled=true;
                                codeArea.wordWrap=true;
                                codeArea.text="Type Code Here:";
                                codeArea.name="inputField";
                                midSpacer = new Spacer();
                                midSpacer.percentHeight = 98;
                                midSpacer.percentWidth = 10;
                                midSpacer.left = 210;
                                midSpacer.top = 5;
                                canvasBox.addChild(treeFile);
                                canvasBox.addChild(codeArea);
                                canvasBox.addChild(midSpacer);
                                return canvasBox;
                        }

                        private function menuBarHandler(evt:ItemClickEvent):void
                        {
                                switch(evt.label)
                                {
                                        case "Open":
                                                onLoad();
                                                break;
                                        case "Save":
                                                onSave();
                                                break;
```

```
            case "Submit":
                loadURL();
                break;
            case "Add Tab":
                addTab();
                break;
            case "Delete Tab":
                deleteTab();
                break;
        }
}

private function addTab():void
{
        tabListLength++;
        treeName.addItem("Untitled"+tabListLength.toString()+".java");
        tabNav.addChild(buildCanvas(tabListLength));
}

private function deleteTab():void
{
        if(currentCanvasIndex == 0)
        {

        }
        //tabNav.removeElementAt(currentCanvasIndex);
        else if(currentCanvasIndex > 0)
        {
            tabNav.removeChild(currentCanvas);
            //tabNav.removeElement(currentCanvas);
            treeName.removeItemAt(currentCanvasIndex);
            tabListLength--;
        }
}

private function onLoad():void
{
        //Instantiate on loading
        fileReference = new FileReference();
        //create the filter which will be just uploading the txt
        var myFilter:FileFilter = new FileFilter("Text","*.*");
        fileReference.browse([myFilter]);
        fileReference.addEventListener(Event.SELECT,onFileSelect);
        fileReference.addEventListener(Event.COMPLETE,onFileComplete);
}

private function onSave():void
{
        //Instantiate on downloading otherwise you will get an error
        fileReference = new FileReference();
```

```
                fileReference.save(codeArea.text,"myExample.java");
            }

            private function onFileSelect(event:Event):void
            {
                fileReference.load();
            }

            private function onFileComplete(event:Event):void
            {
                var tempCodeArea:TextArea = currentCanvas.getChildByName("inputField") as
TextArea;
                tempCodeArea.text = fileReference.data.readUTFBytes(fileReference.data.length);
                treeName[currentCanvasIndex] = fileReference.name;
                currentCanvas.label = treeName
[currentCanvasIndex];
            }

            private function onViewChange(evt:IndexChangedEvent):void
            {
                var tempCanvas:Canvas = tabNav.getChildAt(evt.newIndex) as Canvas;
                currentCanvas = tempCanvas;
                currentCanvasIndex = evt.newIndex;
                var tempCodeArea:TextArea = currentCanvas.getChildByName("inputField") as
TextArea;
                tempCodeArea.text = currentCanvasIndex.toString();
            }

            private function findNewTab(evt:Event):void
            {
                tabNav.validateDisplayList();
            }


            private function loadURL():void
            {
                //Instantiate on loading
                var url:String = "http://cslvm156.csc.calpoly.edu/Submitter/Submitter.html";
                var request:URLRequest = new URLRequest(url);
                navigateToURL(request);
            }
        ]]>
    </fx:Script>
    <s:Panel left="5" top="5" height="100%" width="100%" id="panelBox" title="Web-CAT Development
Environment">
        <mx:TabNavigator left="11.7" top="55" height="98%" width="98%" id="tabNav"
change="onViewChange(event)">
        </mx:TabNavigator>
    </s:Panel>
</s:Application>
```

## 9.3 Web-CAT Submission Webpage

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body  {
	font: 100% Verdana, Arial, Helvetica, sans-serif;
	background: #666666;
	margin: 0; /* it's good practice to zero the margin and padding of the body element to account for differing browser defaults */
	padding: 0;
	text-align: center; /* this centers the container in IE 5* browsers. The text is then set to the left aligned default in the #container selector */
	color: #000000;
}
.twoColFixRtHdr #container {
	width: 780px;  /* using 20px less than a full 800px width allows for browser chrome and avoids a horizontal scroll bar */
	background: #FFFFFF;
	margin: 0 auto; /* the auto margins (in conjunction with a width) center the page */
	border: 1px solid #000000;
	text-align: left; /* this overrides the text-align: center on the body element. */
}
.twoColFixRtHdr #header {
	background: #DDDDDD;
	padding: 0 10px 0 20px;  /* this padding matches the left alignment of the elements in the divs that appear beneath it. If an image is used in the #header instead of text, you may want to remove the padding. */
}
.twoColFixRtHdr #header h1 {
	margin: 0; /* zeroing the margin of the last element in the #header div will avoid margin collapse - an unexplainable space between divs. If the div has a border around it, this is not necessary as that also avoids the margin collapse */
	padding: 10px 0; /* using padding instead of margin will allow you to keep the element away from the edges of the div */
}
.twoColFixRtHdr #sidebar1 {
	float: right; /* since this element is floated, a width must be given */
	width: 300px; /* the actual width of this div, in standards-compliant browsers, or standards mode in Internet Explorer will include the padding and border in addition to the width */
	background: #EBEBEB; /* the background color will be displayed for the length of the content in the column, but no further */
	padding: 15px 10px;
}
.twoColFixRtHdr #mainContent {
	margin: 0 350px 0 0; /* the right margin on this div element creates the column down the right side of the page
- no matter how much content the sidebar1 div contains, the column space will remain. You can remove this margin if
you want the #mainContent div's text to fill the #sidebar1 space when the content in #sidebar1 ends. */
	padding: 0 20px; /* remember that padding is the space inside the div box and margin is the space outside the
div box */
}
.twoColFixRtHdr #footer {
	padding: 0 10px 0 20px; /* this padding matches the left alignment of the elements in the divs that appear
above it. */
	background:#DDDDDD;
}
.twoColFixRtHdr #footer p {
	margin: 0; /* zeroing the margins of the first element in the footer will avoid the possibility of margin collapse - a
space between divs */
	padding: 10px 0; /* padding on this element will create space, just as the the margin would have, without the
margin collapse issue */
}
.fltrt { /* this class can be used to float an element right in your page. The floated element must precede the element it
should be next to on the page. */
	float: right;
	margin-left: 8px;
}
.fltlft { /* this class can be used to float an element left in your page */
	float: left;
	margin-right: 8px;
}
.clearfloat { /* this class should be placed on a div or break element and should be the final element before the close of a
container that should fully contain a float */
	clear:both;
    height:0;
    font-size: 1px;
    line-height: 0px;
}
-->
</style><!--[if IE 5]>
<style type="text/css">
/* place css box model fixes for IE 5* in this conditional comment */
.twoColFixRtHdr #sidebar1 { width: 220px; }
</style>
<![endif]--><!--[if IE]>
<style type="text/css">
/* place css fixes for all versions of IE in this conditional comment */
.twoColFixRtHdr #sidebar1 { padding-top: 30px; }
.twoColFixRtHdr #mainContent { zoom: 1; }
/* the above proprietary zoom property gives IE the hasLayout it needs to avoid several bugs */
</style>
<![endif]--></head>

<body class="twoColFixRtHdr">
```

```
<div id="container">
  <div id="header">
    <h1>Web-CAT Submission Tool</h1>
  <!-- end #header --></div>
  <div id="sidebar1">
    <h3><center>Example of a Submission</center></h3>
    <p>User Name: mdgarcia</p>
    <p>Password: *********</p>
    <p>Course: 101</p>
    <p>Assignment Name: HelloWorld</p>
    <p>Institution: Cal Poly</p>
    <p>File: C:\HelloWorld.zip</p>
  <!-- end #sidebar1 --></div>
  <div id="mainContent">
    <form action="http://cslvm156.csc.calpoly.edu:8080/Web-CAT/WebObjects/Web-CAT.woa/wa/submit"
method="post" enctype="multipart/form-data">
User Name:  <input type="text" name="u"> <br /><br />
Password: <input type="text" name="p"> <br /><br />
Course: <input type="text" name="course"> <br  /><br />
Assignment Name: <input type="text" name="a"> <br /> <br /><!-- Assignment2 -->
Institution: <input type="text" name="d"> <br /><br />
File: <input type="file" name="file1"> <br /><br />
<input type="submit" value="Submit"/><br />
</form>
<li>If you want to include you test file, you must create a zip file and submit the zip file.</li>
<li>Contact your professor about the correct parameters that are needed to submit an assignment</li>
                <!-- end #mainContent --></div>
                <!-- This clearing element should immediately follow the #mainContent div in order to force the #container div to
contain all child floats --><br class="clearfloat" />
  <div id="footer">
    <p>If there are any problems with this page, please email djanzen@calpoly.edu</p>
  <!-- end #footer --></div>
<!-- end #container --></div>
</body>
</html>
```