Student Grading System Using ASP.NET Framework

Jiaoyu Song

College of Technology and Innovation

ASU Polytechnic Campus

- Project Advisor: Lindquist, Timothy
- Project Committee: Gary, Kevin & Whitehouse, Richard
- Project Period: 1/19/2010 to 7/30/2010

ARIZONA STATE UNVERSITY

July 2010

Abstract: Student Grading System (SGS) is a web application developed using ASP.NET framework. SGS provides interfaces for instructors to set up coursework such as assignments, projects and exams, and the system provides grading criteria, scores and comments on graded work to each coursework submitted by a student. It also provides capability for instructors to change their personal profile, and provides capabilities for students to view coursework expectations and outcomes for their own submitted coursework. It also provides capability for students to upload their assignments according to instructor's requirements such as due date, file size and file types. SGS has been implemented on ASP.NET framework with PostgreSQL server.

Key words: Student Grading System, instructor, coursework, web application, ASP.NET, PostgreSQL

1. Introduction

A coursework submission and grading system plays a key role in a course management system at a university. But, such systems don't often relate expectations, outcomes and performance. As each student desires to achieve a good score for each assignment, exam, and/or technical report, the whole process adds heavy workload for instructors in order to make their evaluation fair, comprehensive, and accurate. From the faculty perspective, these are necessary to avoid disagreement from students. An online grading system is a highly desirable addition to the educational tool-kit, particularly when it can provide less effort and a more effective outcome.

In general, a good student grading system must have the following three criteria[1]:

- Reflect differences in student achievements accurately.
- Be clear and accessible to students so they can follow their own progress.
- Be clear and accessible to faculty for administration of coursework and student achievement

Student performance can be evaluated either in relative or absolute terms, which means comparing students with each other or ranking their achievement against a set scale by themselves. Whichever grading scheme an instructor uses, students should be able to determine how they are performing in the course at any time during the semester. Some grading schemes make it impossible for students to determine their final grades because they do not provide the assignment weight distribution, and do not allow students to access them until the end of the course. A complete description of the grading system should appear in the course syllabus, which includes the amount of credit for each assignment, and provides the weights for students to know how to calculate the final grades, and the grade equivalents for the final scores. Also, students should view the grading system as equitable, rewarding them for their performance proportionately. From the standpoint of grading methods, multiple assignments, which spread over the entire semester, provide a better assessment of student performance than one or two large tests or semester-long projects.

A web-based Student Grading System (SGS) provides a solution for recording each exam, assignment and project, which is done by college students for their course credits, and allows student access to the coursework anytime and anywhere. A final course grade should evaluate a student's performance in learning efforts lasting several months. The

final grade often considers several assignments, projects and exams, as these tasks play different roles at final course grade. The web-based Student Grading System is flexible and allows instructors to assign different weight on each task. SGS utilizes a database for recording and query, and it utilizes the current web technology to develop a better student grading solution.

Currently, there are several commercial course management solutions such as Sakai and Moodle[2]. They provide wide solution for college student leaning and coursework management. However, they lack a complete and agile grading system to allow students to access the grading system from different views. Several web-based grading systems have also been developed for automatic submission of coursework and online grading of coursework. For example, there are WebCT, Blackboard, and ClassNet, which are notable[3,4,6,7]. Blackboard[3] supplies a set of tools for coursework management, online learning communities, and an advanced architecture for Web-based integration of multiple learning systems. WebCT[7] provides support for developing course materials and many educational tools for collaboration, learning, and communication among students and instructors. ClassNet[6] supplies an interface for coursework-related activities such as creation of courses, tests and assignments, enrollment control, assignment submission and grading. It also provides tools to facilitate communication between students and instructors such as chat rooms, discussion forums and e-mail exchange. The objectives of all these systems are to provide a user-friendly interface for students to submit coursework, and a user-friendly interface for instructors to grade their coursework and give immediate comment and feedback. However, none of the systems has complete support for online grading of coursework.

This project developed a web-based application with and underlying database for students and instructors to use in performing their grading related operations, such as submission, checking, as well as assigning of coursework and grades. SGS was designed to be password-protected by providing a login feature, which authenticates the user by means of a user name and password. Users are able to login to the website, and to use the system as required. SGS can also be used to grade the coursework online, keeping track of the student's progress and producing a group statistics of the overall student performance. The primary objective is to facilitate computer science faculty in grading coursework easily, and at the same time to easily manage information related to student coursework and grades.

This technical report is organized as follows. In Section 2, I present the objectives of SGS, describe the key challenges involved in this application and then discuss my selection of development tools and database. Section 3 provides the web-based SGS design and database diagram, and illustrates SGS 4-tier architecture. Section 4 explains the key related implementation techniques, user interfaces and results. In Section 5, the conclusions and potential future work are presented.

2. Objective and Solution

2.1 Objective

The objective of this project is to develop Student Grading System, a web application implemented using ASP.NET framework and database server. This system will provides user-friendly interfaces for instructors to set up coursework including assignments, projects, and exams, and grade the coursework and give

immediate feedback and comments on each coursework. It also provides capability for instructors to change their personal profile information, such as password. It provides capabilities for students to view expectations and their own grades. It also provides capability for students to upload their assignments according to instructor's requirements such as due date, file size and file types. Also it will provide various reports for instructors such as showing the average in a class or for a particular student. Most importantly, this system assists instructors to manage each assignment grading criteria, feedback, and numeric grade (performance). Students can view the grading criteria before submission. After grading is completed by the instructor, students can view all three results along with average/mean scores.

In the project, the key technical challenges are as follows:

- Authentication with different roles (instructor and student), user account, password management and maintenance
- Coursework management including assignments, projects, exams, presentations and reports
- Score record forms which let instructors input the scores friendly
- Score tracking based on course
- Score weight input (assignment, project, exam, presentation and report) and calculation
- Score statistical reports based on courses (average, minimum and maximum score, figures) in the view of instructors
- Score modification by instructors
- Student log in, lookup and score query
- Score reports in the view of students
- The system will protect Actions on their Controller using Forms Authentication.

Based on the goal and the expected challenges, I selected ASP.NET as my development environment and PostgreSQL as database server.

2.2 Web Application

In general, web based applications may contain N-layers, with each layer having its well-defined functionality. The following discussion is a system with four layers:

- Presentation layer (PL) decodes the data from the client. Typically data entered by the user involves certain defined process. For example, consider the user name and password entered by the user. There are a number of popular tools to build PL: (1) CGI, (2) ASP.NET[8], and (3) JavaServer Pages (JSP).
- Application logic is usually implemented in the business logic layer. Business logic includes: (1) Performing all required calculations and validations; (2) Managing workflow (including keeping track of session data); (3) Managing all data access for the presentation tier.
- Data access layer (DAL) provides simplified access to data stored in persistent storage of an entity-relationship database. The DAL encapsulates the complexity of the underlying data store from the remainder of the application.
- The data layer is responsible for managing the data. In the most case, a data layer may simply be a modern relational database. However, it may include data access procedures to other data sources such as hierarchical databases, legacy flat files, or XML files.

2.3 PostgreSQL

There are two leading open source systems: MySQL and PostgreSQL. MySQL has the advantage of significant opportunities, since it actually comes with almost none of the features taught in a typical DB systems course.

It has no cost-based optimizer, no B+-trees, no fine-grained concurrency control, no recovery, and no hash joins. By contrast, PostgreSQL already has most of these features.

3. Web-based SGS Design

A web-based Student Grading System is developed in this project. The system has several components including online coursework setting and grading for instructor, online assignment submission for students, an interface for students to read, review and assess their assignments, and check their graded assignments. SGS is a Web-based application and can interact with the user through a Web browser.

Factoring a student grading application into logical parts is useful. Breaking a large piece of software into smaller pieces can make it easier to build, to reuse, and to modify. It can also be helpful in accommodating different technologies. Therefore, before giving SGS's architecture design, I first present use case analysis for the SGS.

3.1 Use Cases

In general, a coursework management system includes three use cases: Instructor, Student and Administrator. The Student Grading System mainly considers two use cases:

- Instructor – The user assigns coursework and set its requirements such as weight, due date, description, objective, uploading file type(s) and limit size etc. Instructor also assesses the grades and gives comments to each assignment and calculates class performance. Fig.1 shows the flow chart of instructor use case.
- Student – The end user who reviews the coursework from the instructor, submits the answer of coursework via uploading file through client web page, and also can check his/her grades and compares his performance with the class. Fig.2 shows the flow chart of Student use case.

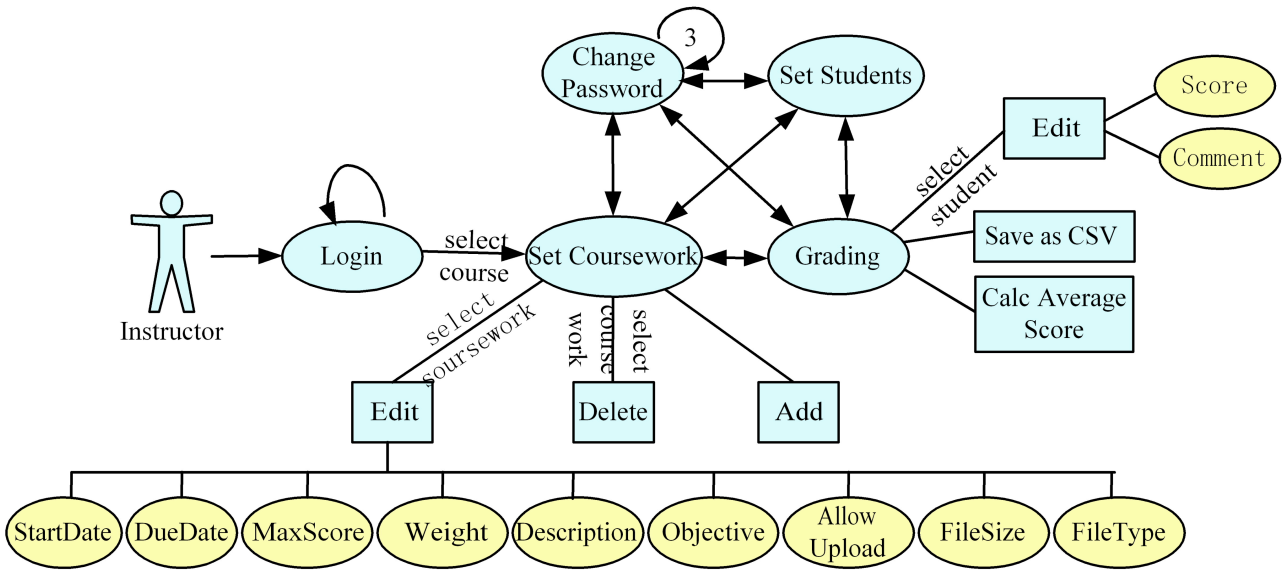

Fig.1 Instructor Use Case of SGS

The ellipsoid indicates the states and the squares represent functions of objects. The arrow shows the state translation, and the line indicates the function call. The number on arrow shows the limitation of the operation. The description on the line or arrow shows the pre-operation.

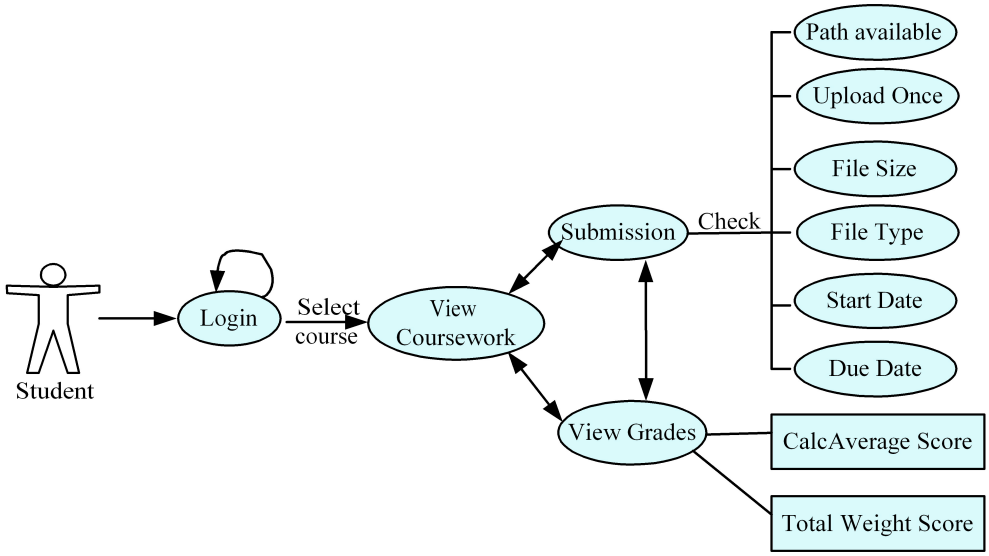

Fig.2 Student Use Case of SGS

The ellipsoid indicates the states and the squares represent functions of objects. The arrow shows the state translation, and the line indicates the function call.

Instructors may add assignments, projects and exams to their courses. Each coursework is represented by a unique assignmentNo. The number of possible points should be the maximum possible for the assignment. The weight defaults to 0. This weight represents the value of this assignment versus other assignments in the same category. The weight is used to weigh one assignment more than another without changing the total points possible. For example, an assignment worth 50 points has a weight of 20% associated with it. When the overall average is computed, that assignment will actually be worth 10 points.

Fig.1 presents a graphical overview of the functionality provided by web-based Student Grading System in terms of Instructor. The ellipsoid shows the state [1] of instructor's activity or attributes of instructor's operation (illustrated by squares). Instructor can easily switch from one state to another. There are four states for an instructor: set coursework, grading, set students and change password.

An instructor needs to login by entering username and password. After authentication, instructor can set coursework which include the following operations:

- Edit the attributes of an existing coursework; update assignment table.
- Delete coursework. Before deleting, it is necessary to check whether there are related submissions or grades. Only after cleaning all its grades, the coursework operation can be executed.
- Add new coursework. Assign a unique assignmentNo and insert this new record into assignment table.

In order to increase the security of SGS and lower risk of the web attack, an instructor only has three chances to modify his/her password at a time.

Fig.2 shows a graphical overview of the functionality provided by web-based Student Grading System from the perspective of the

[1] In this report, a state corresponds to a web page or a tab in web pages.

student. The SGS allows a student to select an available course and login by entering username and password. After authentication, student has three states: view coursework, view grades and assignment submission.

Student will be prompted with viewing coursework information. The other option is to view the scores associated with the student and his/her overall average. Also displayed in the same page, it will be the average scores of entire class for each assignment and total count. This allows students to get a more detailed picture of the performance of the class as a whole.

The submission has to pass several checks: 1) whether submission satisfies requirements on file type(s) and size; 2) whether the assignment is submitted between start date and due date; 3) whether upload path is available on the server; 4) whether the student has uploaded already.

3.2 SGS 4-Tier Architecture Design

After developing an application into several logic parts, it is necessary to find an approach to organize these parts together to create a software system. N-tier applications have become the norm for building software today. An N-tier application divides a system into discrete logical parts. The common selection is a 4-part breakdown — presentation, logic, data access and data — although other possibilities exist. N-tier applications have first emerged as a way of solving some of the problems associated with traditional client/server applications. With the arrival of the Web, this architecture has come to dominate new development. Fig.3 shows 4-tier architecture of my SGS as follows:

- **Tier I: SGS Presentation Layer**

In the presentation layer, the user interacts with the server using a web browser such as Internet Explorer or Firefox. The client requests the web server for ASP page and the server responds to it. To implement ASP.NET presentation layer using standard technique – XML master pages, code-behind, sessions, cache, etc.

- **Tier II: SGS Logic Layer**

The SGS logic layer works as a mediator to transfer the data from the presentation layer. In the 4-tier architecture, the data access layer is not made to interact with the presentation layer.

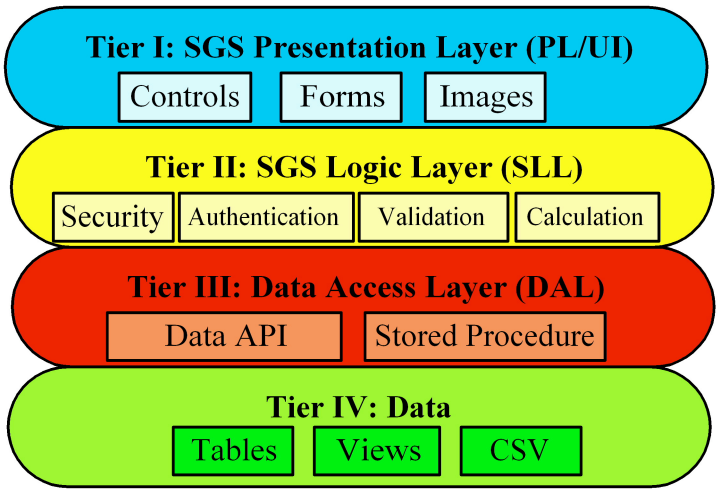

Fig.3 4-tier SGS Architecture Design

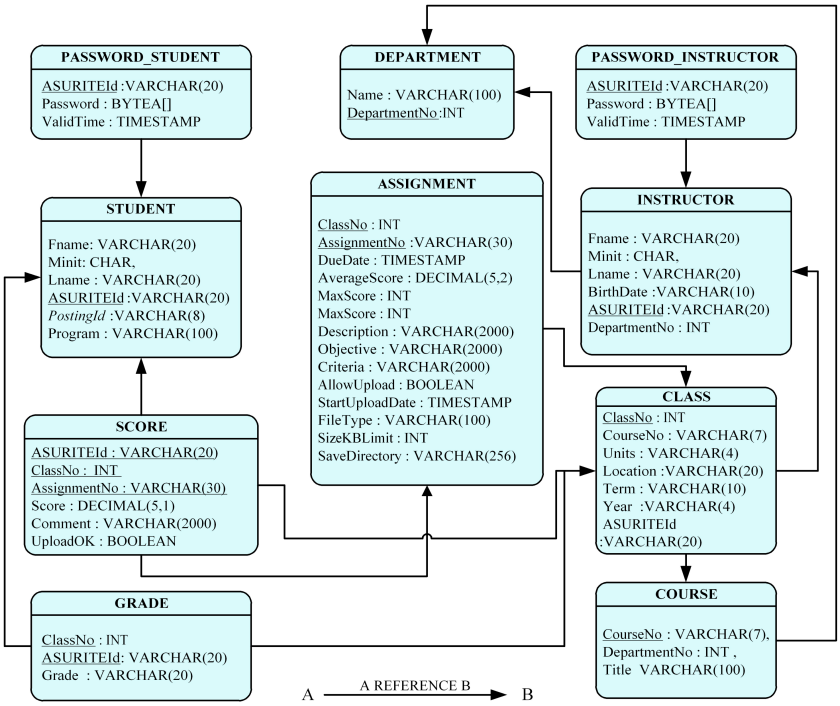

Fig.4 ER Model of SGS

- **Tier III: Data Access Layer (DAL)**

This layer is used to get or set the data to the database back and forth. This layer only interacts with the database. Database queries or stored procedures are utilized to access the data from the database or to perform any operation to the database. Npgsql is an ASP.Net data provider for PostgresSQL. It allows any program developed for ASP.NET framework to access database server. It is implemented in C# code.

- **Tier IV: Data Layer**

PostgreSQL is used as the database server. It receives requests from the Npgsql in the form of SQL queries, executes them and returns the dataset back to the Npgsql. It allows a client application (Console, WinForms, ASP.NET, Web Services...) to send and receive data with a PostgreSQL server. It is actively developed based on the guidelines specified in the ASP.NET documentation.

In an ASP.NET n-tiered architecture, web pages do not make direct calls to the database. A given layer only communicates with its neighboring layers. ASP.NET Web pages should reference custom objects defined in the business object layer. These objects provide database information in a class structure.

- **Database Design**

SGS data are stored in the PostgreSQL server. Fig.4 shows ER model of SGS. It provides the tables of SGS database and shows the foreign references among the tables.

4. Implementation and Related Technologies

In this section, implementation of SGS and related key technologies are discussed in details.

4.1 Login Page

A form login page is created to authenticate users. Users are forced to use a form login

page before the application performs any HTTP POST actions to protected database or files. This can be set in the web.config file for the application. Also display an appropriate error message when login with error. The following key technologies are used in SGS:

● Forms Authentication

Forms authentication uses an authentication ticket that is created when a user logs in to a site, and then it tracks the user throughout the site. If the user requests a page that requires authenticated access and that user has not previously logged in to the site, then the user is redirected to a configured login page. The login page prompts the user to supply credentials, typically a user name and password. These credentials are then passed to the server and validated against a user store. My project uses PostgreSQL Server database as the store way. After the user's credentials are authenticated, the user is redirected to the originally requested page[10].

In essence, Forms Authentication is a tool for wrapping web application with a thin security layer, allowing user to have his own custom login interface and verification functionality.

● MD5 cryptography

In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value.

The ComputeHash method of the MD5 class is used to wrap password by returning the hash as an array of 16 bytes.

4.2 For Instructor

4.2.1 Set Coursework

Highlighted tab in Fig.5 shows the web page for instructor to set coursework.

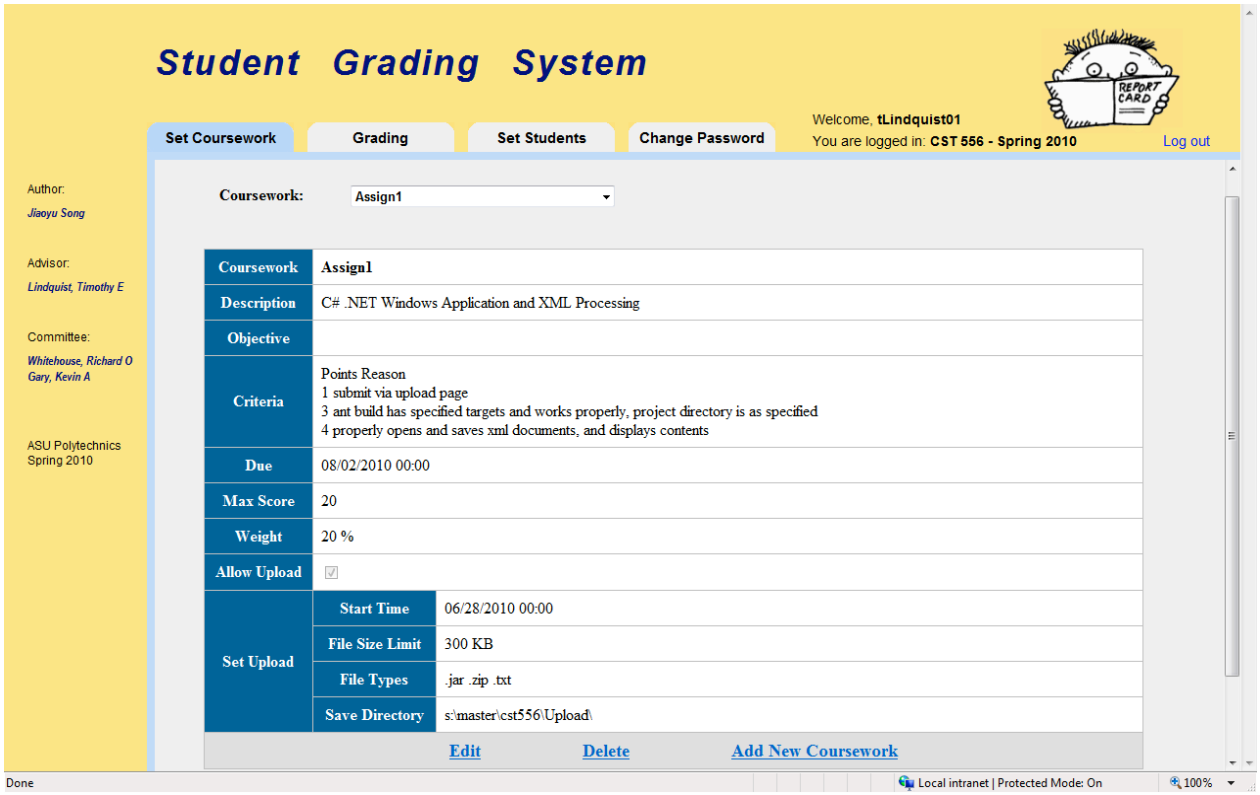

Fig.5 Instructor: Set Coursework View with Edit, Delete and Add functions

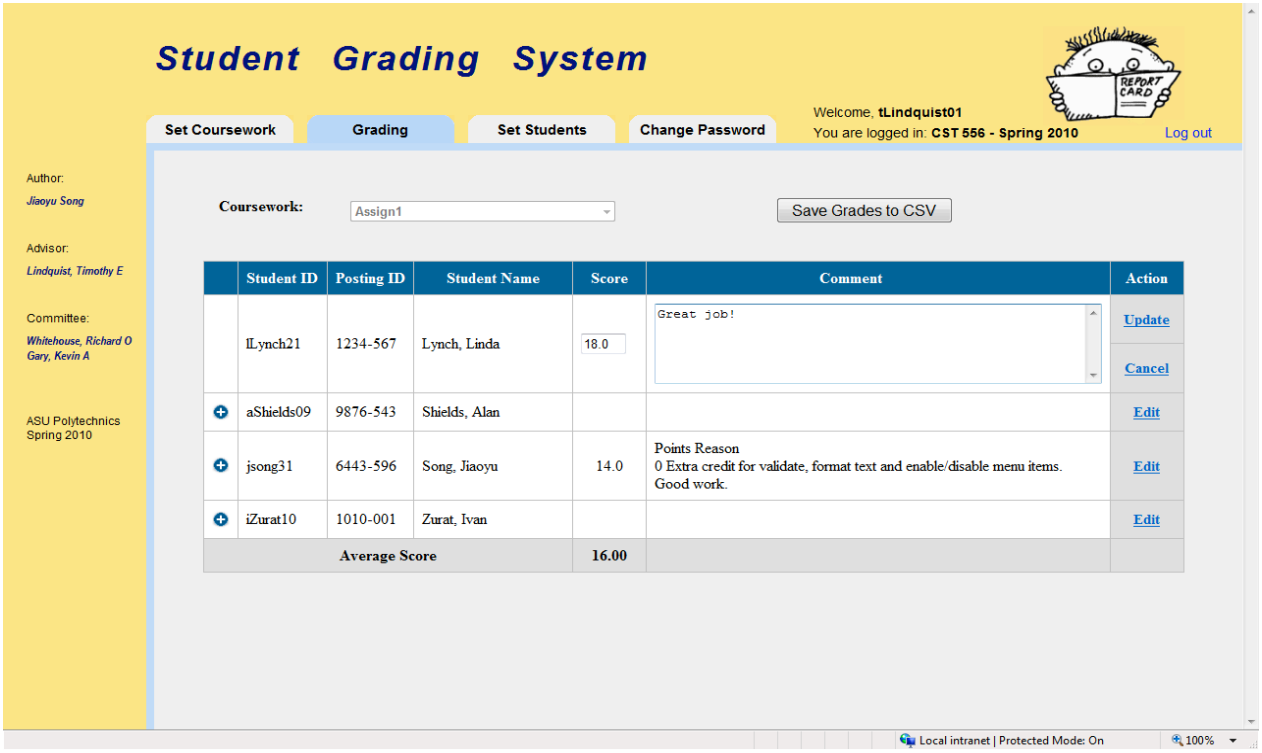

Fig.6 Instructor: Grading View with Edit and Save Grades to CSV function

The main function of Set Coursework is to accept various inputs from an instructor. The challenge is to validate all untrusted input. Weak input validation is a common vulnerability that could be exploited by a number of injection attacks such as SQL injection[11].

Since dynamic SQL queries are based on user input, an attacker could inject malicious SQL commands that can be executed by the database. To protect from injection attacks, several input validations are applied to check for known good data by validating the type, length, format, and range in both client and server sides.

On the client side, for example, the following regular expression is being utilized for multiline textbox (such as Description, Objective and Criteria) to limit length, and special characters to avoid SQL injection:

^[\w\s\|\\\(\)\[\]\{\}\+\-*\?\'\.,:;~!@#%/=]{0,1000}$

And this regular expression is being utilized for Directory to keep valid path format:

^(([a-zA-Z]:)|((([a-zA-Z]:(\\|(\\\\)|/|(//)))|(\\|(\\\\)|/|(//)))?((((\w+[\+\-\.\x20])*\w+(\\|(\\\\)|/|(//)))+((\w+[\+\-\.\x20])*\w+)?)|((\w+[\+\-\.\x20])*\w+))*))$

On the server side, for example, the Drive letter extracted from Directory is being validated for existence. Also access control is being validated to the Directory.

4.2.2 Grading

Fig.6 shows the highlight of the Grading page. The main function is to accept input for scores and comments from an instructor as well as to allow the instructor to save grades data to a CSV file.

4.2.3 Change Password

Fig.7 shows the Change Password page. From the client side, RequiredFieldValidator and RegularExpressionValidator are used to make

sure the user's input satisfies the requirements to set a password described at the top of the page. At the server side, CompareValidator is used to check whether the input password matches the stored one. At the same time, the maximum attempts are limited to change password. If the number of trying is more than three, the input will be locked for a period.

4.2.4 Set Students

Fig.8 shows the Set Students page. On this page, instructors can add new student to a specific course, or delete information of an existing student from a specific course only when the student has no related coursework.

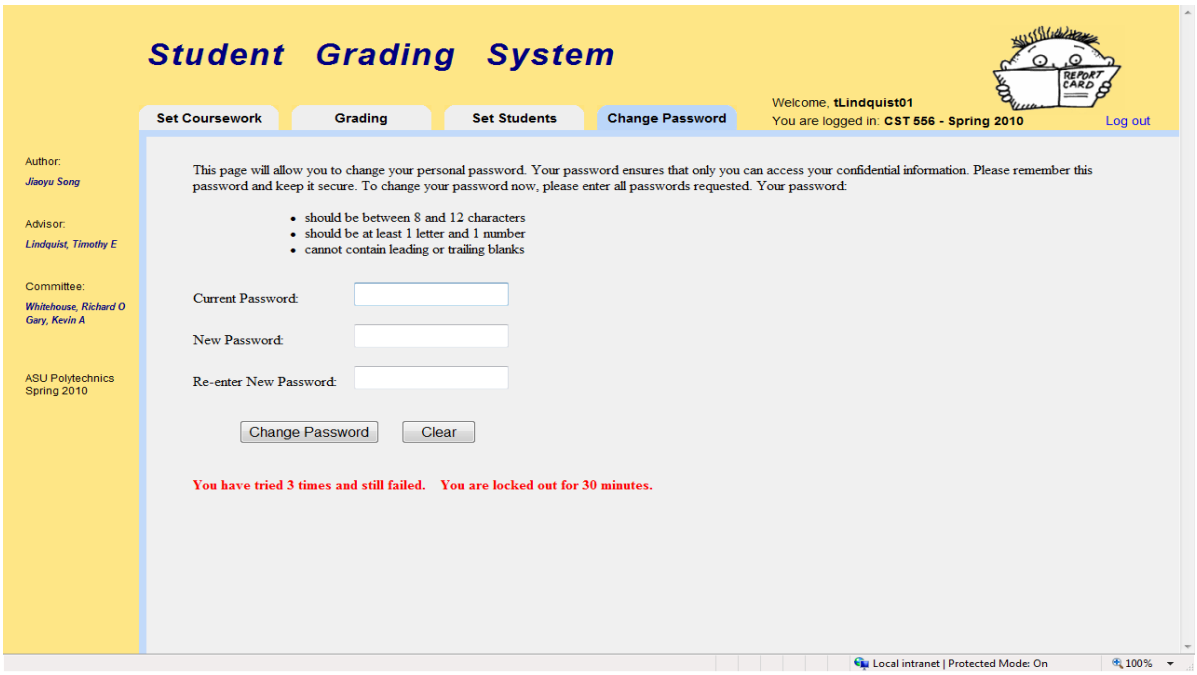

Fig.7 Instructor: Change Password

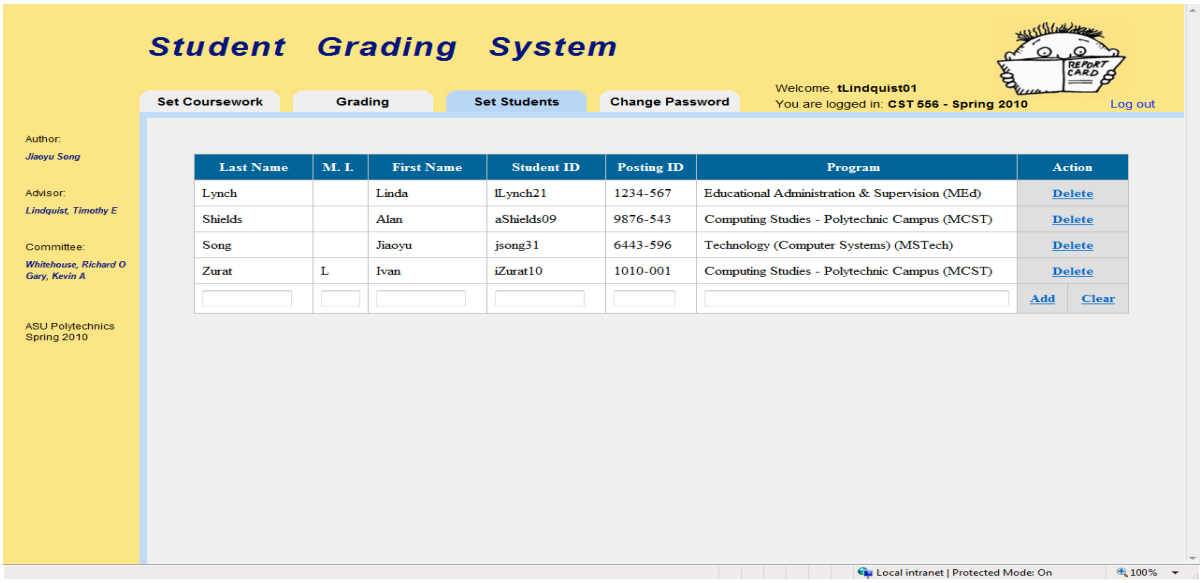

Fig.8 Instructor: Set Students

4.3 For Students

4.3.1 View Coursework and Grades

Fig.9 & fig.10 show the Coursework and Grades pages separately in the view of student. I implemented the following technologies to make the display more visibly:

- GridView Controller with custom row expands. It can give a sketchy outline of all coursework, or show detail information of these coursework when related rows were expanded.
- Custom graphic buttons. I drew two types of graphic buttons to represent two actions separately: show details and hide details.

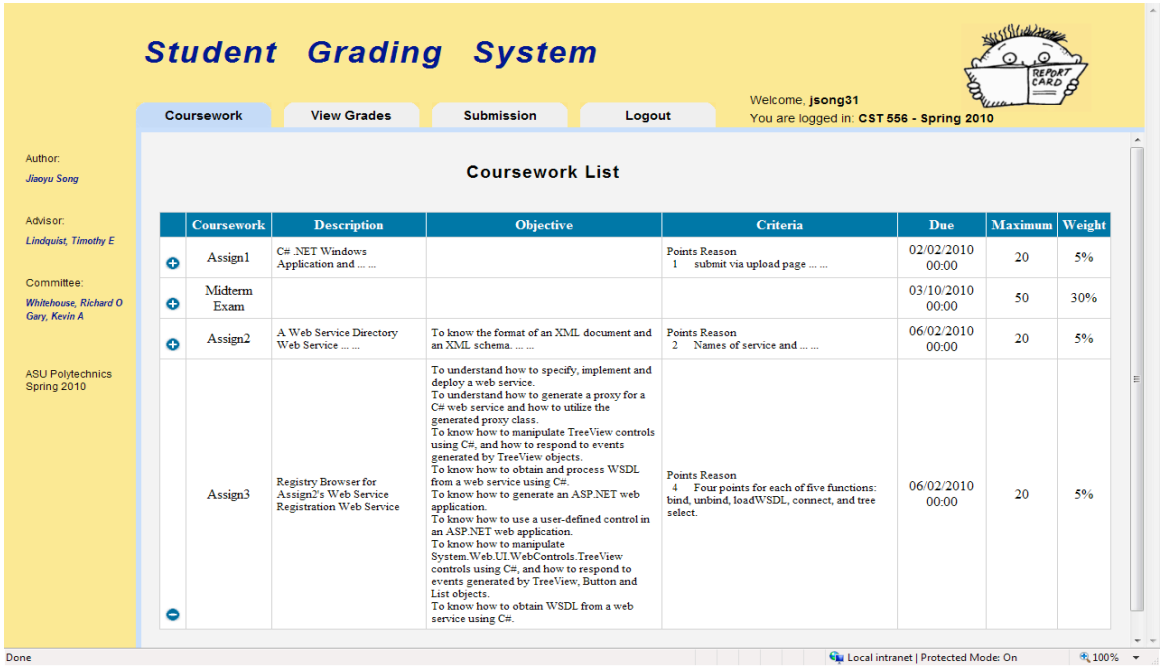

Fig.9 Student: Coursework View with custom row expand

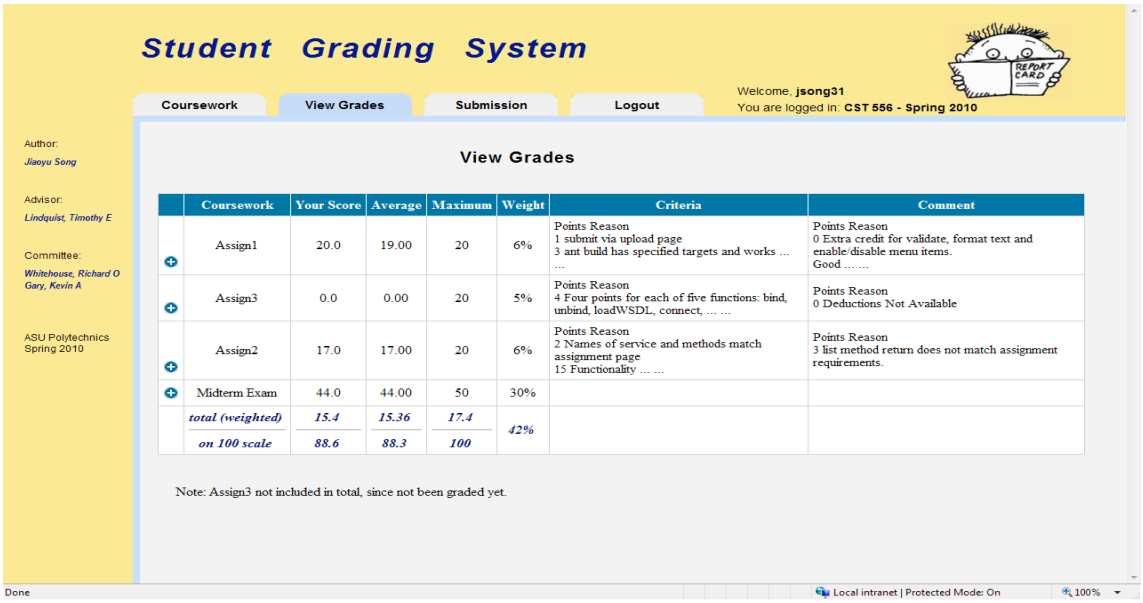

Fig.10 Student: View Grades with custom row expand

4.3.2 Coursework Submission

Fig.11 displays the Assignment Submission page for student. The interesting part is a real-time timer, which can help student estimate submission time and enrich user experience. The correspond technologies I implement are:

- AJAX Extensions
- UpdatePanel

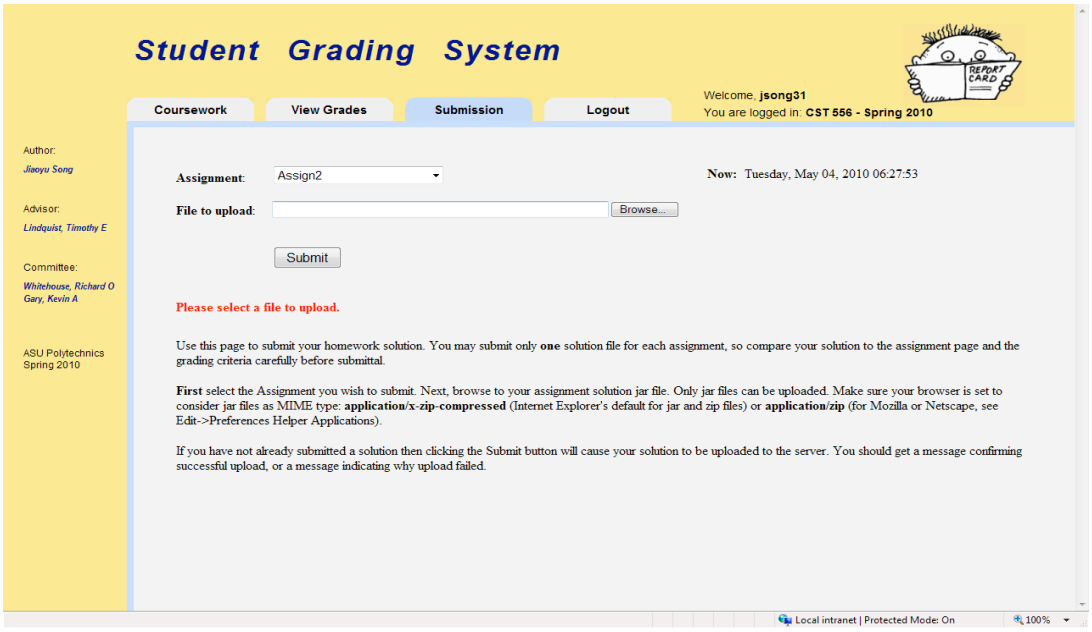

Fig.11 Student: Assignment Submission

4.4 Other Technologies

- Nested Master Page

ASP.NET master pages help to create a consistent layout for the pages in an application. A single master page defines the look and feel and standard behavior for all of the pages (or a group of pages). Then individual content pages that contain the content to display can create. When users request the content pages, they merge with the master page to produce output that combines the layout of the master page with the content from the content page.[12]

My project uses nested master pages, which with one master page referencing another as its master, to display all pages. The main master page is the colorful background with system title and information. The child master page has the image buttons as main menu and a content display area.

- Validation controllers
- Session timeout setting
- CSS
- JavaScript

5. Conclusions and Future Work

The Student Grading System assists instructors at universities to manage conveniently and flexibly for each assignment grading criteria, instructor feedback, and numeric grade (performance). It also allows that students view the grading criteria before submission. After grading completed by the instructor, students can view all three along with average/mean scores. The system has implemented using ASP.NET framework and PostgreSQL server.

Through developing this project, I grasped knowledge and skills of how to develop a web-based application using ASP.NET with interaction of PostgreSQL server thoroughly. Also, I mastered a set of technologies to implement the application such as forms authentication, input validation, listview, Ajax, and session etc.

Test has shown that ASP.NET framework with PostgreSQL server provides a satisfactory solution to SGS. Dr. Lindquist plans to use this system to support his courses.

The system features can be expanded by some further improvement. The future work may include:

- Email confirm for changing password
- Deal with forgetting password
- Provide links to the web pages of instructors' courses
- Authority for administrators

References

[1] Wright, D. L. (1987) Grading: Measuring students' achievement in a course. *Teaching at UNL*. Lincoln, NE: Teaching and Learning Center, The University of Nebraska-Lincoln. Blackboard Inc, Blackboard products, blackboard.com, 2003.

[2] Yaden B. and Blaine P., How to Wiki in Moodle or Sakai Collaboration for Online Language Learning, *The IALLT Journal*, 39:1, 2007

[3] Laws, R. Distance learning's explosion on the Internet, *Journal of Computing in Higher Education*, 7, (2), 48-64, Spring 1996.

[4] Johnson,W. L., Blake,T., Shaw, E. Automated management and delivery of distance courseware, Presented at WebNet 96 - San Francisco, CA, October 15-19, 1996.

[5] PostgreSQL Documentation, http://www.postgresql.org/docs/2009.

[6] Joy, M. S., Luck, M. The BOSS system for on-line submission and assessment, *Monitor:Journal of the CTI Center for Computing*, 10, 27-29, 1998.

[7] WebCT, WebCT software and services, www.webct.com/products, 2001.

[8] Jenq J., Alexisponniah M., Ashkenazi I. and Burus R., Online interactive home work grading system, J. *SYSTEMICS, CYBERNETICS AND INFORMATICS*, vol. 4(4) 58-62, 2010.

[9] Chappell D., Kirk S., Application design guidelines: from N-Tier to .NET, http://msdn.microsoft.com/en-us/library/ms978384.aspx, 2002

[10] Explained: Forms Authentication in ASP.NET 2.0 http://msdn.microsoft.com/en-us/library/ff647070.aspx

[11] How To: Protect From Injection Attacks in ASP.NET http://msdn.microsoft.com/en-us/library/ff647397.aspx

[12] ASP.NET Master Pages. http://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx