

AC 2007-1839: MEASURING STUDENT PARTICIPATION IN GROUP PROJECTS THROUGH AN ONLINE PEER EVALUATION SYSTEM

Phil Rawles, Purdue University

Phil Rawles is an Associate Professor in the Department of Computer Information Technology at Purdue University, West Lafayette, IN. Professor Rawles specializes in teaching network administration and information security in the network engineering technology program.

Measuring Student Participation in Group Projects Through An On-Line Peer Evaluation System

Background

Group work is an important facet of a technical undergraduate degree. Through group projects students learn to work cooperatively with others, manage resources and deadlines, and resolve interpersonal conflicts. These skills are essential for workplace success in many disciplines where group projects are common.

Unfortunately, whenever students work in groups the amount of work done by individual students is rarely split evenly. Some students do the majority of the work while others “coast on their peer’s coat tails.” Attempting to reward credit appropriately to the industrious students often proves problematic as the majority of the work is often done outside of scheduled class times when the instructor is not present.

In an effort to award credit appropriately instructors often use a peer evaluation process whereby students evaluate the effort and results turned in by their peers on the project. In this manner students learn important assessment techniques and budgeting approaches that mirror those used for merit increases in a professional setting. This approach has two key benefits: students are more aware of the quality and effort of their peers’ work than the instructor and can more accurately evaluate each other and students are empowered to affect each other’s grades directly which usually results in higher quality work. Despite these benefits, many instructors choose not to employ a peer evaluation process because it is time consuming and cumbersome to implement and is all too often not embraced by students.

Over the course of several years the author has experimented with many methods of peer evaluation where students provide input to award credit among group members appropriate to their level of effort and effectiveness. This work has led to the development of a robust peer evaluation process that allows students to reward or penalize their peers based on their work while insulating students from reprisal and ensuring system integrity. The system has been refined over several years us by multiple faculty members in several courses.

To enable the data and computationally intense process to be efficiently implemented, an on-line data collection and scoring process has been developed. Using a web site, students evaluate their peers. This on-line data collection technique not only adds error checking and simplifies data collection, but also increases student participation. Students are far more likely to fill out an on-line form than to complete a paper based document and submit it manually.

This paper will provide background on peer evaluation processes and introduce an automated, web based system that can be used by instructors to add peer evaluation to their courses with minimal required administration. Implementation information along with a link to a sample of the system and production code is provided. The code can be freely used in other courses in either its native form or altered as required to meet the needs of the environment.

The system detailed in the paper is designed for small groups of two to six students. For groups of more than six students a different peer evaluation processes may be required. If a project is

highly integrated with all students working together closely then the peer evaluation scheme presented here may remain appropriate. However, projects that have many students involved are typically of large enough scope that the project is broken into multiple teams. Due to this segmentation students are not familiar with the work done by any particular peer outside of their team, making it difficult for them to effectively evaluate other's performance. For projects of this nature a project manager or task force of team leaders typically coordinates the work of the teams to ensure the overall project is successful. In these cases it is logical to have this leadership evaluate the performance of the team members. This type of evaluation is outside the scope of the automated system described in this paper.

Guidelines

While peer evaluation appears to be an area where there are relatively few widely held truths, informal research, anecdotal evidence from discussions with colleagues, and personal experience show that there are some key guidelines that should always be observed when implementing a peer evaluation system:

- Students must be mature enough to objectively evaluate their and other's work. The percentage of final project grade and several other choices are depended on the maturity level of the students participating in the process.
- The purpose and procedures used in the system should be clearly explained to the students in advance. There should be no ambiguity as to what is being evaluated and how it is being evaluated.
- The peer evaluation should be represented to the students as a key part of their project grade rather than a punitive measure at the end of the project. The author defines the peer evaluation scores as "performance points" and lists it along with function, presentation, and documentation as the four components of the project grade. In this manner students are aware from day one that their performance is being evaluated and they must earn their performance points just as they earn the rest of their grade.
- Students should be thoroughly briefed on the need to honestly evaluate everyone. Often students are unwilling to penalize a peer only to learn that by not being honest with them they are rewarding the very behavior they wish to stop. In our early courses when students are initially exposed to peer evaluation relatively few low grades are given. In higher level courses where students have more experience working in groups and doing peer evaluation they are much more likely to address problems early and score their peers accurately. The range of scores tends to increase in direct correlation with student maturity and exposure to the system.
- Steps must be taken to ensure that everyone participates in the process. If half of the students do not submit an evaluation the resultant data is of little real value. A common approach is to require a student to submit an evaluation or forfeit any points they receive from their peers. In this manner students are highly incented to participate.
- Results from a peer evaluation should only be reported to the students as an aggregate number: no scores from an individual evaluator should be shared. This is critical to

preserve the integrity of the system and to limit retaliatory actions.

- Results should be shared with students as soon as practical. This ensures that the students are aware of the perception of their performance and can make any required changes in their work or communication with their group members to resolve issues on subsequent projects.

Failure to meet any of these guidelines can greatly limit the effectiveness and success of any peer evaluation system. However, there are a multitude of methods to conduct a peer evaluation within the guidelines for a successful system.

Options

Despite general agreement on overarching guidelines, there is limited consensus as to how best to conduct a peer evaluation. Informal research and discussions with colleagues indicate that there are a few key options used in peer evaluation systems. These options include whether the student evaluates themselves, the weighting of the peer evaluation in terms of the overall project grade, whether students can earn bonus points, and whether comments are required along with scores.

A successful system can be built regardless of which set of options are chosen. Therefore the decision on which options to use remains one of personal preference based on the types of projects being evaluated and the make up of the student groups. Each of these options is available in the automated system described later in the paper.

A common point of contention among educators regarding peer evaluation systems is whether students should evaluate themselves in addition to their peers. Like many other aspects of the peer evaluation process the student's maturity level should be a key question asked when making this decision. At the author's institution, upper division undergraduate students have proven in most cases to have the ability to objectively review their own performance. The experience of the author's colleagues indicates that lower division undergraduate students may lack the maturity for such objective self-analysis.

Another factor in choosing whether to have the students evaluate themselves is the size of the group. For groups of two, self-evaluation is a key metric as there is relatively little data upon which to base an evaluation. For groups of four or more there is sufficient external data to reduce the need for self-evaluation. Groups of three normally will work with either approach.

The percentage of the final grade affected by peer evaluation is another key question. If too small, the incentive of increased participation is not met. If too large, students can be victimized by unjust peers. In the author's experience twenty percent is a good compromise for most undergraduate applications: It provides enough leverage to incent students to perform, yet limits their peer's influence to around two letter grades on a project. Depending on the maturity of the students this percentage can be adjusted. More mature students typically show the ability to fairly evaluate their peers and can be given more leverage in the form of a larger percentage. Conversely, less mature students tend to be more petty and capricious and should be given less leverage via a lower percentage.

As has already been discussed, work in a student group is rarely evenly split among the membership; one or two students will often be responsible for the majority of the group's success. This can be reflected in a peer evaluation system by allowing students to award extra points beyond the base percentage to an outstanding peer. This is best achieved by viewing the peer evaluation as a zero sum operation. The sum of points awarded cannot exceed the number of points available (x points for each member of the group). In this approach students can award extra points to outstanding peers if they are willing to award fewer points to underachieving peers.

In the author's experience it is best to cap the number of points that can be awarded to any individual to one hundred and fifty percent of the base peer evaluation points. If the peer evaluation is worth twenty points and there are three members being evaluated there would be a total of sixty points available. Any student could be given up to thirty points (150% of 20), but the number of points awarded to the other group members must be reduced to limit the total number of points awarded to sixty.

This approach forces students to take a pragmatic viewpoint of their peers. To reward someone who excelled they must penalize someone who performed poorly. This is an excellent method to help break down the wall that most students have against negatively reviewing their peers. Often students will complain incessantly to the instructor about a peer only to give them full points on the peer evaluation because they do not want to face the discomfort of presenting honest criticism; it is easier to ignore the poor performance rather than proactively deal with it. Fortunately the desire to reward those deserving of bonus points usually triumphs over the aversion of penalizing those deserving of fewer points and the peer evaluation process promotes honest communication among the group members.

Another option that some instructors like to exercise is requiring students to provide free form feedback to the student being reviewed in addition to a numeric score. This approach can be useful in helping the students justify their scores. It is the author's opinion that comments should be required for any students that is given either bonus points or less than fifty percent of the available points. In this manner students learn to document exceptional and poor performance and it provides better evidence of over and underperforming student's work that may be useful in a grade appeal situation.

The author normally uses peer evaluation in the laboratory section of upper division undergraduate courses. Teams of two to four students work to complete each project and its documentation. In this environment the peer evaluation base is 20 points with up to 150% of the base available to any individual student. Students who award less than 50% or more than 100% of the base points to an individual must justify their score with a written comment. Students self-evaluate if there are fewer than four members in a group, but cannot give themselves bonus points. They can however, give themselves fewer than the base points to enable them to give others bonus points. The number of points awarded to each student is averaged to calculate their final score.

System History

The author went through several iterations in building an effective automated peer evaluation system. Originally students submitted their evaluations on paper, but implementation proved to be problematic in a paper domain. Students would consistently submit invalid scores (too many points awarded) and it took hours to calculate final evaluation scores. Many students simply did not turn in evaluations. To increase participation and the accuracy of the system while reducing the time required to calculate final scores an automated system was created. The automated system consists of an on-line application that accepts and error-checks data from the students and a client application where data is initially configured and final scores are calculated.

The system has been used in the author's home department for over three years and has been expanded over time to offer support for each the options previously presented. Different instructors routinely use different options so each option set is well tested. It is the author's hope that this work can be used in the wider domain to improve instructional effectiveness at other institutions.

The system was developed for ease of implementation. The on-line portions are written using Active Server Pages (ASP), a Visual Basic type web programming environment that can be implemented on Windows XP and above. The client application is written in Visual BASIC for Applications (VBA) and is contained within the Microsoft Access .mdb file accessed by the on-line application. These programming environments and languages were chosen to make the system as accessible as possible. Although limited, the relative simplicity of the environment makes it easier for instructors with minimal programming skills to understand and modify the code. Instructors with more substantial programming skills might find the environment cumbersome, but can still work within it or are free to port the system approach and logic to other environments.

Implementation

The system is designed to run under Microsoft Internet Information Server (IIS). It has been tested under IIS on Windows 2000, Windows 2003 Server, and Windows XP. When running IIS be sure that your system is up to date on patches and protected by a firewall. Ideally the system should be implemented using HTTPS rather than HTTP to protect student data while in transmission.

To implement the system an IIS site is created that points to the directory where the .asp files are placed. Edit the ASP files to reference the location of the database (.mdb) file which should be placed in a directory not accessible from the web. More complete installation instructions are available at the system download page <http://netpeereval.tech.purdue.edu/download>.

Configuration

The base system is designed to be implemented in a directory for a single course. In this manner it can be installed by an instructor rather than requiring implementation by a system administrator. If you wish to use it for multiple courses copies can be placed in multiple directories.

To begin using the system two items must be configured: student groups and projects. To configure student groups open the “students” table in the database file as shown in figure 1.

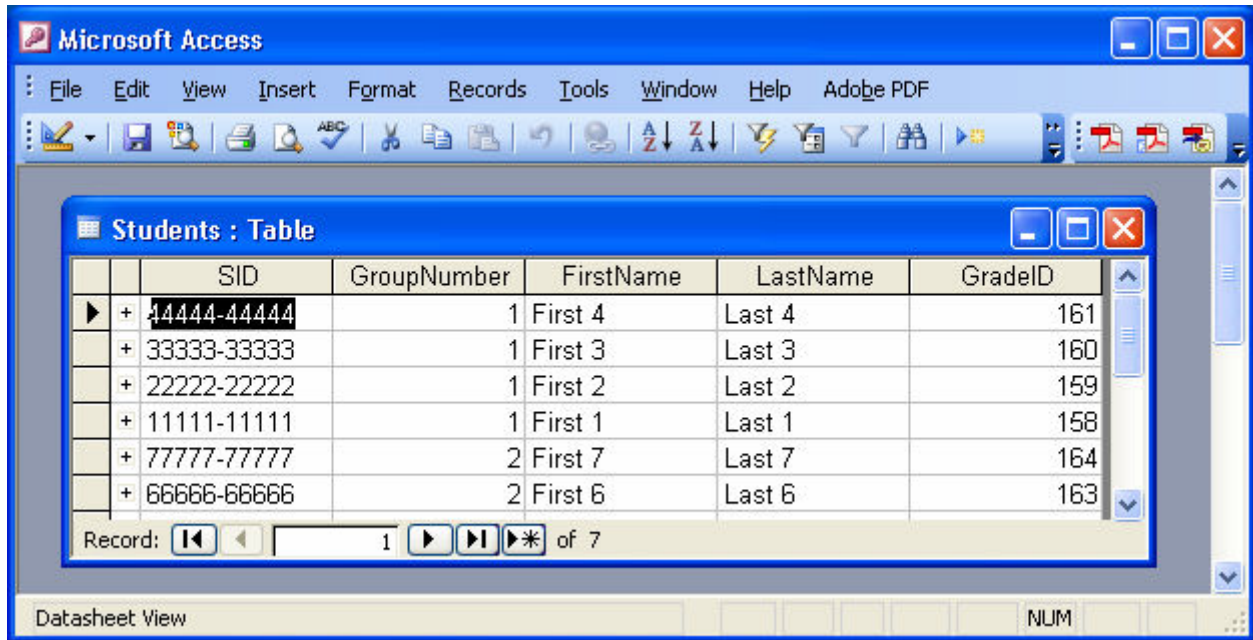


Figure 1 – Students Table

In the students table you provide the following information for each student:

- SID – This is the authentication mechanism the student must enter to log into the system. In a larger system students would log in via pre-existing accounts, but in this small implementation their student ID number will suffice.
- GroupNumber – The number of the student’s group
- FirstName – The student’s first name
- LastName – The student’s last name
- GradeID – This is the key for the table and will automatically be entered when you add the other information

Entering data directly into the table allows for copy and pasting of student data from a spreadsheet or grade book application. Alternatively a filter can be used to import a file from a spreadsheet or other delimited file.

Once the student information is loaded into the system each project must be configured. To configure projects, open the StudentEval.mdb file in Microsoft Access and go to the project configuration form shown in figure 2.

In the project configuration form you provide key information for the project:

- Project number – the number of the project to be displayed on the web form
- Project name – the name of the project to be displayed on the web form
- Start date – the system will start accepting data at 12:01 AM on this date

- End date – the system will stop accepting data at 11:59 PM on this date
- Self Evaluation – whether the student can evaluate themselves
- Base Points – the base number of points per person
- Max Points – the maximum number of points a person can receive
- Self Max Points – the maximum number of points a student can award to themselves
- Require Comment –requires a comment for every student (a comment is always required if a student is given more than base points or less than 50% of base points).

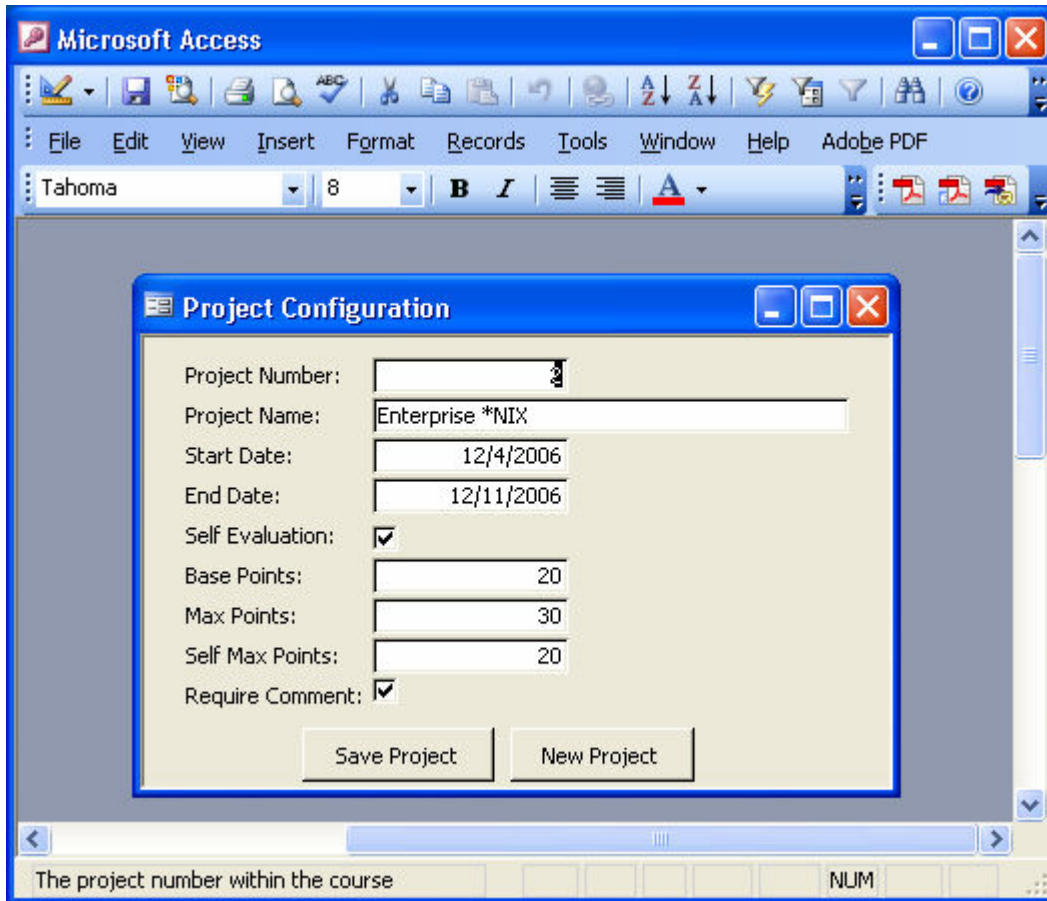


Figure 2 – Project Configuration Form

Once all projects have been configured the system will automatically allow students to enter evaluation data between the defined start date and end date. This allows the instructor to set-up the system for the entire term at one time. The author typically has the peer evaluation run for a period of eight days. Since laboratory reports are due the day before lab, the peer evaluation period starts and ends on a laboratory day which gives a chance to remind the students that it is due. This approach has worked well, but should be adapted to fit local course/department culture.

Student Data Entry

As illustrated in figure 3, students enter their data into the system using a web interface. Students authenticate to the system via their student ID or other unique, private value assigned in the SID field of the student table. In a larger deployment the authentication and authorization function

could be configured to use student's existing college credentials, but that requires cooperation from the IT area and is beyond the scope of this simple solution.

Because students are using their student ID to directly access the system it is highly recommended/required that the web site be HTTPS enabled to encrypt the data during transmission. Enabling a site for HTTPS is not difficult and there are a multitude of HOWTOs available on the web that detail specific steps. Links to this supporting documentation are included in the distribution package.

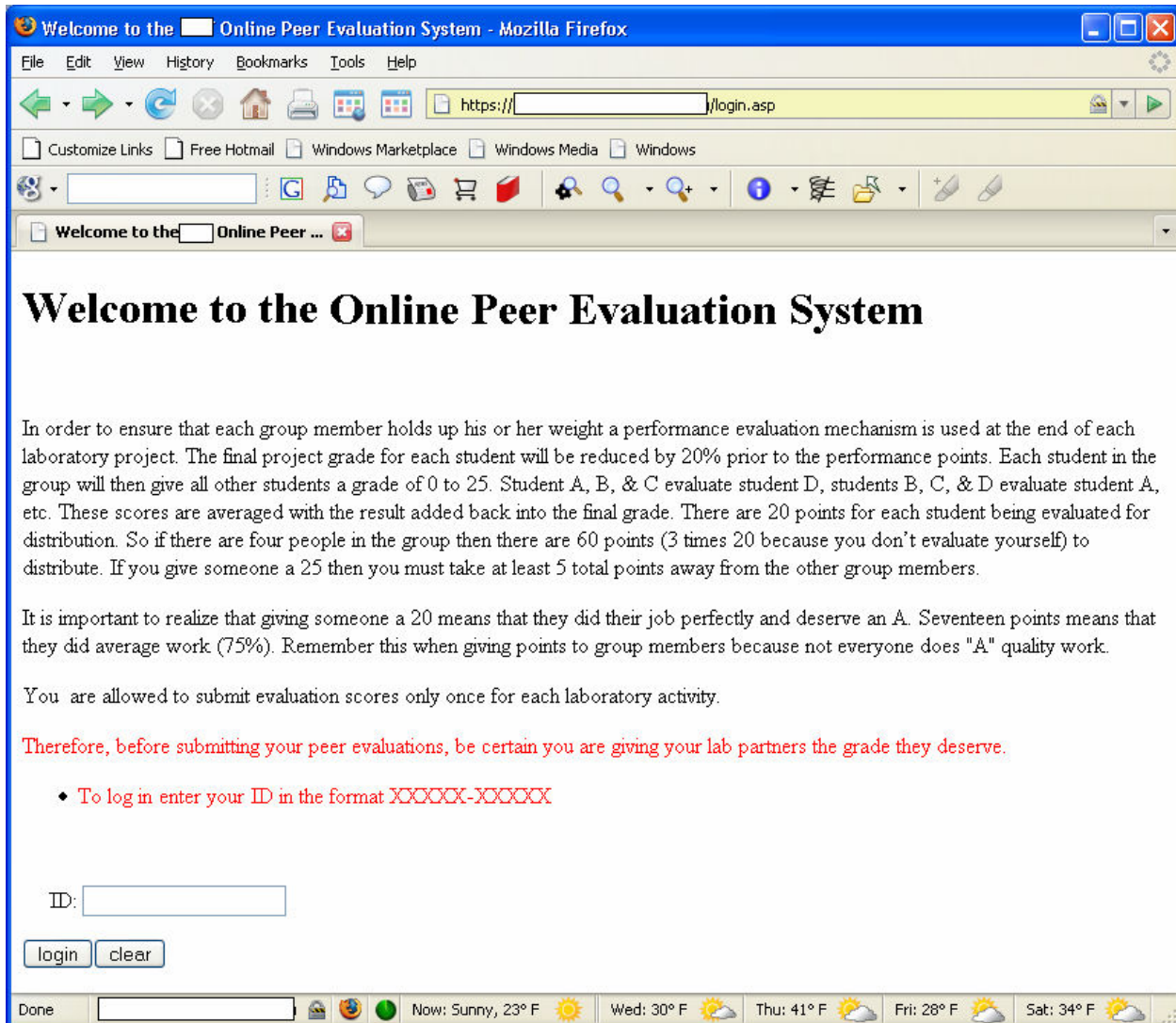


Figure 3 – Student Login Form

After they are authenticated, the student will be presented with a list of projects for which they are expected to complete a peer evaluation along with the dates each project is open for evaluation as illustrated in figure 4. If a student selects a project that is not open an error page is displayed.

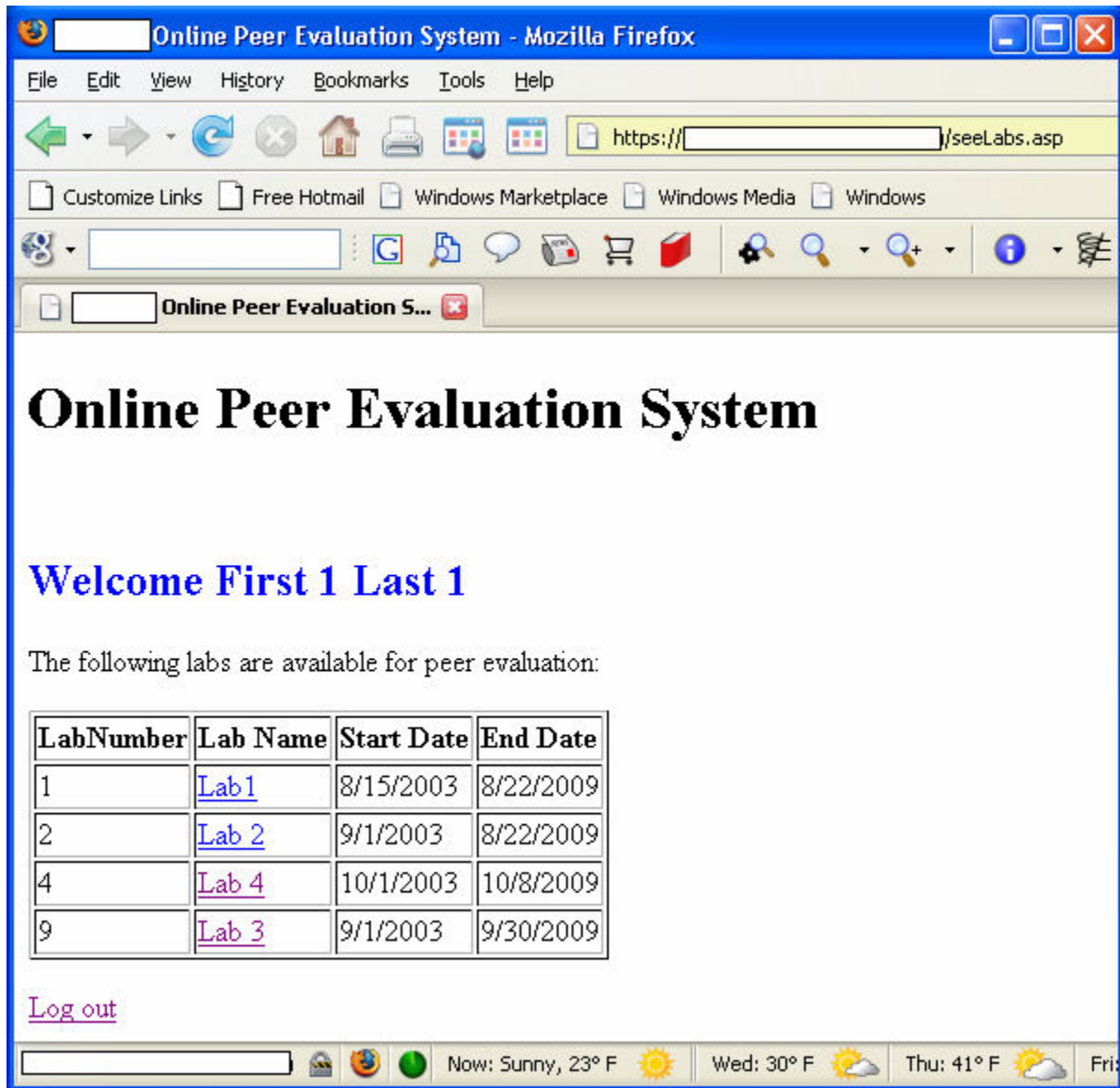


Figure 4 – Project List Page

Once the student selects a project that is open for evaluation they are shown a page to enter their evaluations. As shown in figure 5, this screen contains a list of students to evaluate and a pull down menu of allowable scores. The student list will contain an entry for each of the student's peers and if *self evaluation* is specified in the project configuration, their name as well. The score entry pull down contains scores in the range of zero to the value set as *max points* in the project configuration.

Once scores have been entered the system will compare the scores against the rules for acceptable scores. It will verify that no score entered is higher than *max points* and that the total number of points awarded does not exceed the number of allowed points as calculated by multiplying the number of students to be evaluated by the value entered as *base points* in the project configuration screen. The system automatically adjusts the total number of points if

students are self-evaluating. Should the number of points awarded exceed the allowable amount an error message will be displayed and the student will be re-directed back to the score entry page to re-enter their scores.

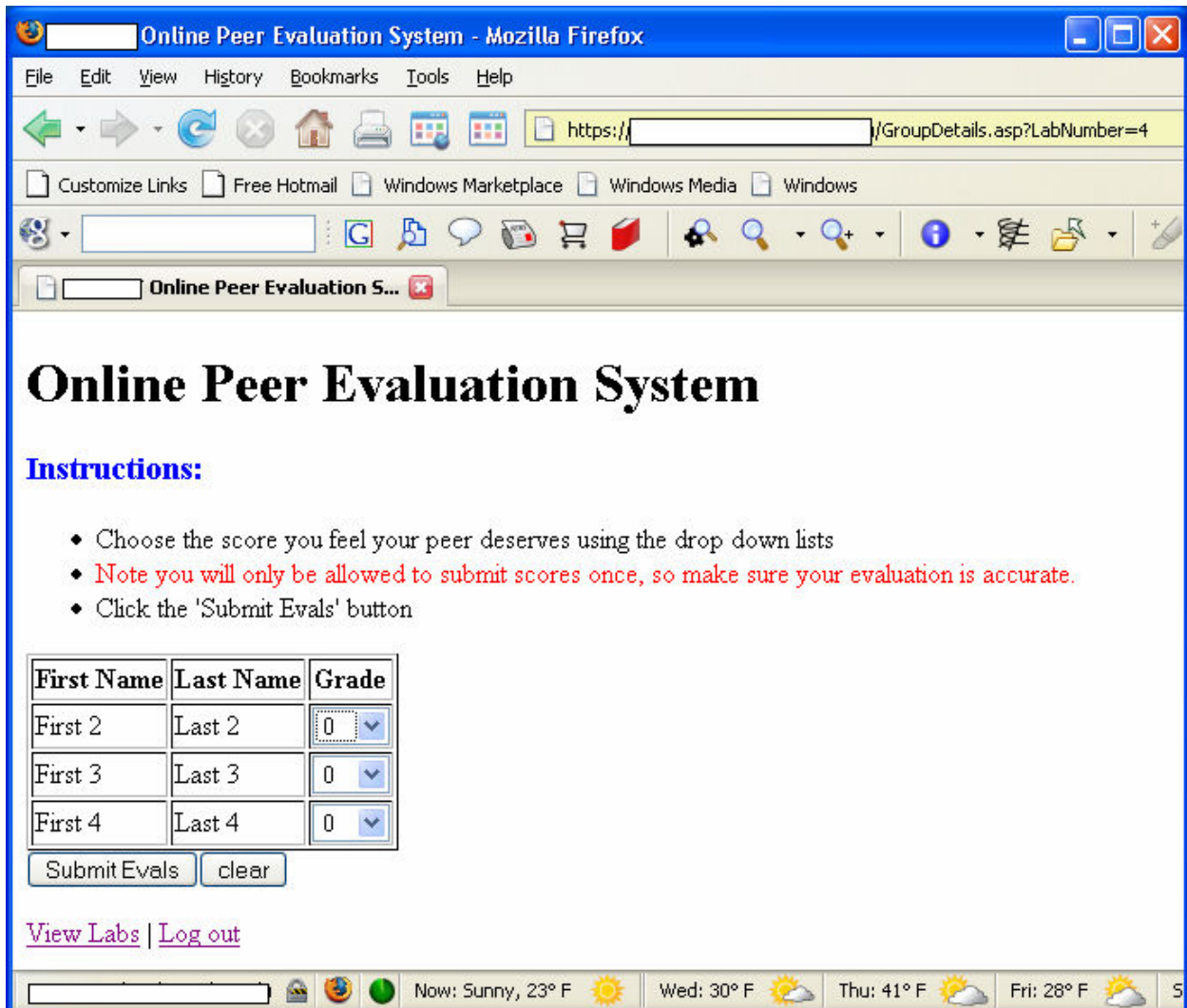


Figure 5 – Score Entry

Once a valid set of scores are entered the system will display a page containing the scores. The score review page is shown in figure 6. If an error has been made the student needs to contact the instructor to make changes as the system only allows them to enter scores once. This limitation is designed to ensure students take the time to think through their evaluations thoroughly before entering their data.

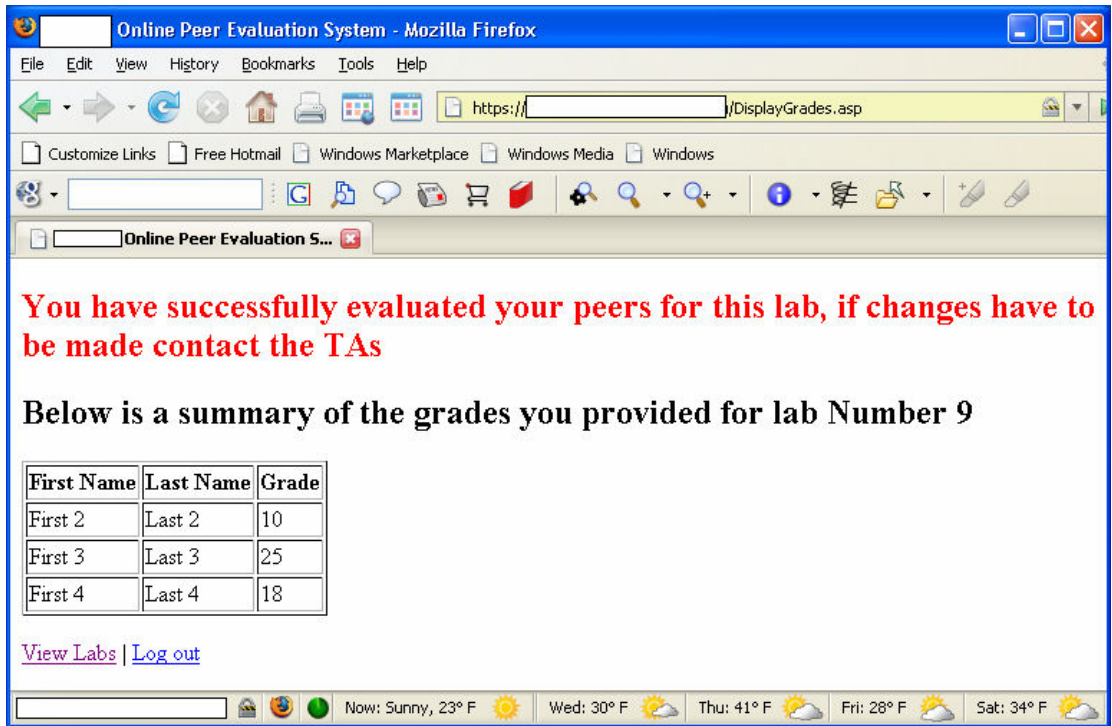


Figure 6 – Score Review

Result Calculation

After the peer evaluation period has closed open the database file in Microsoft Access and go to the Score Evaluations form shown in figure 7 and select the project you wish to score. After you hit the “Create Score Table for Selected Project” button the system will calculate the scores and display the results.

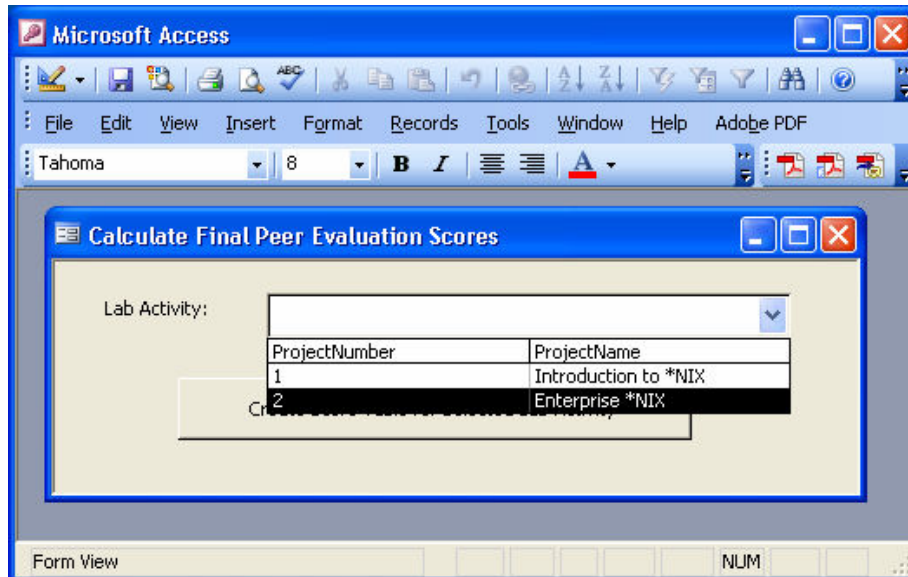


Figure 7 – Score Evaluations

The score calculation takes several cases into consideration in determining each student's final score. If a student did not complete the peer evaluation by entering scores for their peers they automatically receive a "0" for their score. In the case where students are not self-evaluating and they entered scores, but none of their peers entered scores for them, they automatically receive the base number of points for their final score. Otherwise each student receives the average of the scores assigned them by their peers.

Once the calculation is complete the system will display a table of results as shown in figure 8. By displaying the results in a table it is easy to cut and paste the data into a spreadsheet or grade book file. Alternately the results table can be exported in a delimited or fixed column format and imported into another grade application.

	LastName	FirstName	SID	FinalScore
▶	Last 1	First 1	11111-11111	20
	Last 2	First 2	22222-22222	0
	Last 3	First 3	33333-33333	0
	Last 4	First 4	44444-44444	0
	Last 5	First 5	55555-55555	0
	Last 6	First 6	66666-66666	0
	Last 7	First 7	77777-77777	0
*				

Figure 8 – Final Score Table

Example System and Code Availability

A working example of the system is available at <http://netpeereval.tech.purdue.edu/demo>. The example contains example projects for several options and provides authentication information for multiple students on each project. There is a button available that resets the entered data to allow users to start over and experiment with different data sets. The supporting database file can be downloaded via a link on the score review page to allow users to run the score calculation for their experimental data.

This page also contains a link to download an archive of the system and installation instructions to support implementation at other institutions.

Conclusions

The system detailed in this paper provides an automated method of collecting and analyzing peer evaluation data for group projects. The system is intentionally small in scope, supporting only a single class in a given data file. As each course is implemented and maintained separately, privacy and FERPA concerns are minimized while flexibility is maximized.

Although the system is small in scope, it is quite flexible and contains several options commonly used in peer evaluation systems. By selecting options when setting up a project an instructor can easily tailor the system to their liking. Data collection and analysis is completely automated, reducing the time required to calculate evaluation scores and reducing the likelihood of error.

The system is available for download and can be used as-is or can be freely tailored to meet the needs of individual instructors.