

GradeWatch – the Software Package Displaying on Web Pages Students Grades

Bogdan M. Wilamowski, Aleksander Malinowski
University of Wyoming, Laramie, WY / Bradley University, Peoria, IL

Abstract

This paper describes a web-based database allowing students to check their progress in courses they are taking using a web browser front end. The database interface is written in PERL and is interfaced to the Internet by a web server. It can be tested and downloaded for non-profit use from <http://sant.bradley.edu/~olekmali/grades/>.

Introduction

Providing up to date and accurate feedback to students is very important in the process of learning. This is usually achieved by returning graded homework and projects. However, in case of assigning many various types of work, which contribute with different weights to the final grade, it is difficult to keep the student informed about his or her current standing. While keeping track of grades is usually easy to perform using a simple spreadsheet, the process of informing students requires more effort and time. Therefore, the need for a database that would be easy to use by both an instructor and students was determined.

There are many database systems available. Each university registrar's office has one. However, these systems are not available for everyday use. For security reasons such systems may only be updated by a limited number of authorized people, and cannot always be accessed by students at any time by phone or from the Internet. Therefore, a small database system was developed and interfaced with a web browser. Internet based information distribution was chosen because of already existing network infrastructure freely available on campuses. Web based service was selected because of the ability to make a non-platform specific front-end which does not require students to have any specific computer system other than being capable of running a web browser¹.

To make the system easy to use by instructors, who usually already incorporate some kind of spreadsheet to keep his or her students' grades, the database is designed to work with a spreadsheet. Because of many existing and evolving data formats a decision was made that the database will work with a generic tab-delimited format that could be produced by most software packages by exporting data as a tab-delimited text file. This approach provides only a minimal inconvenience in forcing the instructor to save the database and then to export it once again to another format.

GradeWatch Architecture

During software development, it is important to justify which part of the software should run on the client machine and which part should run on the server. Applets are transferred through the network when requested and execution is performed entirely on the client machine that has made the request. The applet would have to query a database server located at the same machine where the web site is established. With the CGI approach, approximately the same amount of information has to be passed to the server. However, this approach eliminates the necessity to transfer a database front-end applet each time². If data security is an issue, a web server with a secure http can be utilized, thus making it unnecessary to custom encrypt passed data. The server executes instructions based on the given information and sends the results back to the local machine that made the request^{3,4}.

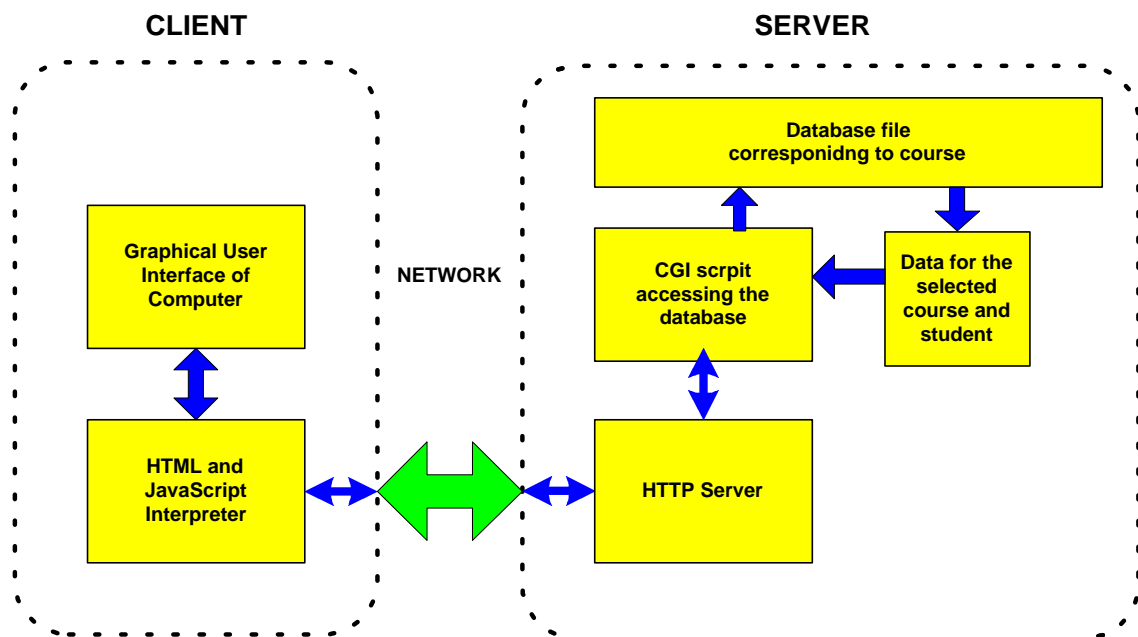


Fig. 1. Data flow in the GradeWatch.

Fig. 1 shows the program component division and data flow in the application. The user interface is programmed in HTML enhanced with JavaScript. A PERL script handles the data flow on the server. The PERL script accesses databases, verifies access authorization, and generates the report sent back to the client in the form of a web page⁵.

To use the database, one can either use an interfacing web page or call a CGI script which is responsible for the dynamic generation of an HTML front end with list of courses for which databases are currently available. A student needs to select the course number, enter his or her family name, and the student ID number as a password. The name and password are verified in the database corresponding to the selected course. If access is granted, the student receives a report on his or her current standing in the class. Otherwise an access denied message is returned.

The GradeWatch has several other features. Each instructor is assigned a password, which gives him or her more access to the database. By using that password the instructor may inspect grades

for all students and send grades to all students registered for the course, provided that students' e-mails was previously entered into the spreadsheet. This message may be accompanied either by an individual message or by a memo to everybody. The other feature is an automatic building of a mailing list, which can be easily transferred to any e-mail client program.

GradeWatch User Interface

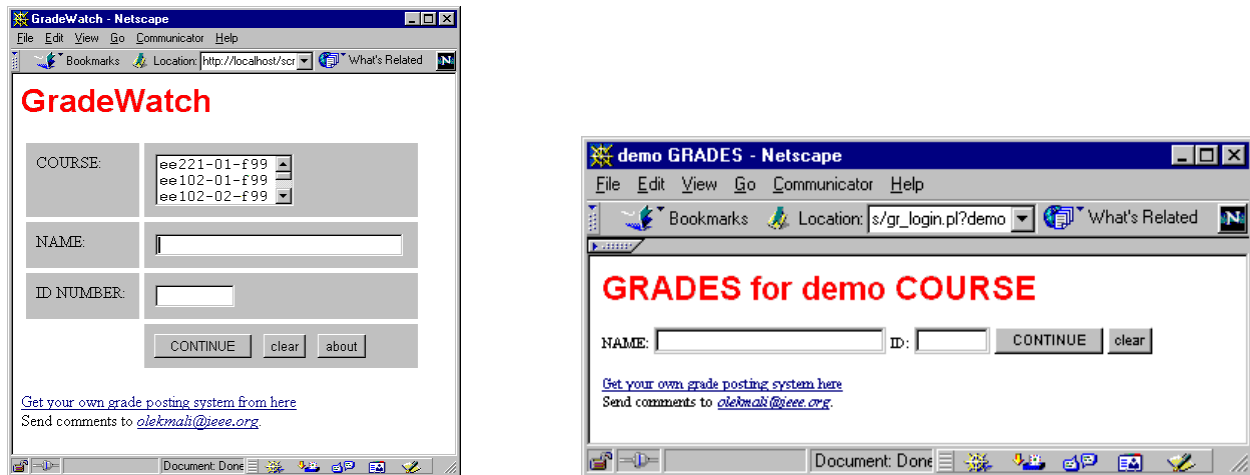


Fig. 2. GradeWatch login pages: with course menu (left) and for a particular course (right).

In order to access the database person needs to provide the course number, family name, and student ID as a password. The database front-end is a simple web page with a CGI form as shown in Fig. 2. After providing the required information the data is sent to the server, which runs a CGI scripts and passes data to it as input. The CGI script is written in PERL5, one of the languages commonly used for server side CGI programming. The program first checks the information about a course in the courses and instructors database, it then scans the database for the appropriate course and collects the information necessary to generate a progress report. The report is generated as a web page that is passed back to the web server, which redirects it to the client - a web browser. The actual amount of information shown depends on the organization of the grade database file for a particular course. A sample report for a demo course is shown in Fig. 3 right.

If an instructor wishes to access the database more functionality is provided. An instructor can access the grades of any student by providing the student's family name and the instructor's password, or can access all grades at the same time by providing his own name in the name field. In the sample database used for both, an illustration in this paper and as a demo on the Internet, the instructor's name and the password are both set to admin. After inspecting the report web page the instructor has the option to send a grade report by e-mail to all students, to a particular student, or to a selected group. Optionally, a few lines of additional memo may be appended. A sample instructor's report for all students is shown in Fig. 3 left.

The instructor can also have access to two additional features of the GradeWatch: the access log for his or hers course, and the course mailing list. The access log proved to be very useful to monitor problems the students might have while accessing the system, and the interest of students in their grades. Incorrect and off campus logins are displayed in different colors and are

easy to pick out of the listing. Fig. 4 illustrates a fragment of the actual log for the demo course.

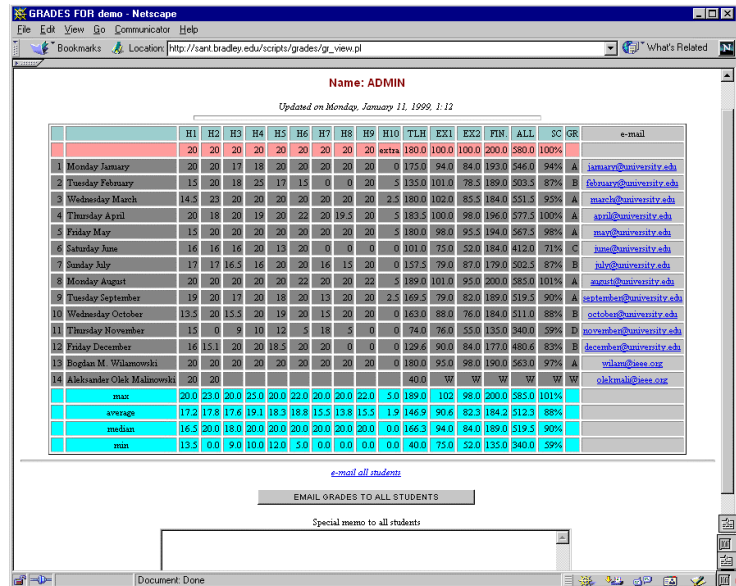
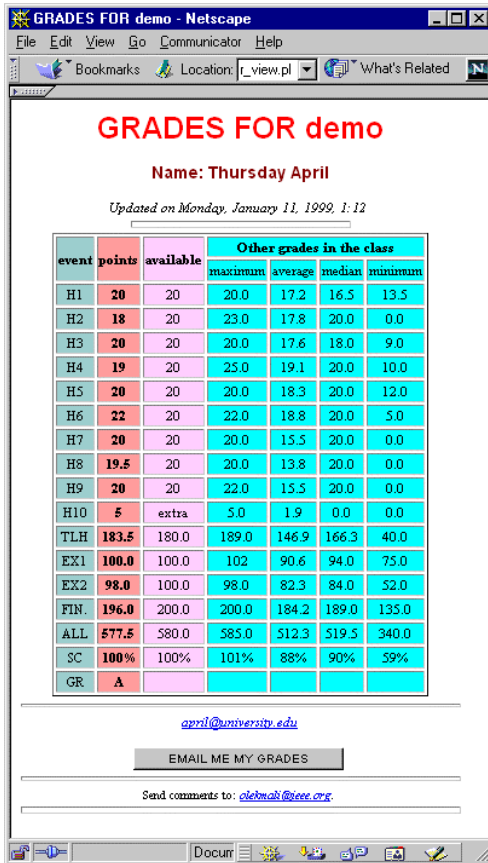


Fig. 3. Course progress report for a student (left) and for the instructor (right)

GradeWatch Database Structure

GradeWatch utilizes at least two database files. The first database consists of information about courses, instructors, instructors' passwords and database file location for each course. Only a web server administrator, or one dedicated person, should manage the file for reasons of information security. Each course has its own database file created and updated by the instructor. The latter database file may be produced by the instructor through exporting data from a spreadsheet as a tab-delimited ASCII file. The format is very flexible with only a few restrictions. Fig. 5 illustrates the format used.

Data for each student is placed in one row of a spreadsheet. The first column contains the student's password. The second column contains the student's name starting with the family name. Data for each student should be followed by ~!~ and optionally an e-mail address in the last column. There may be several lines containing information that should be available to every student. These lines have the first column blank, and the second column in the format of tilde followed immediately by the line identifier. The recognized identifiers are ~header, ~points, ~average, ~median, ~minimum, and ~maximum. The information from these lines may be identified in examples provided in Fig. 3 as listed in columns or rows with appropriate labels. All other lines not matching the criteria given above are ignored. That gives the instructor a great deal of flexibility and freedom in utilizing the spreadsheet as a database.

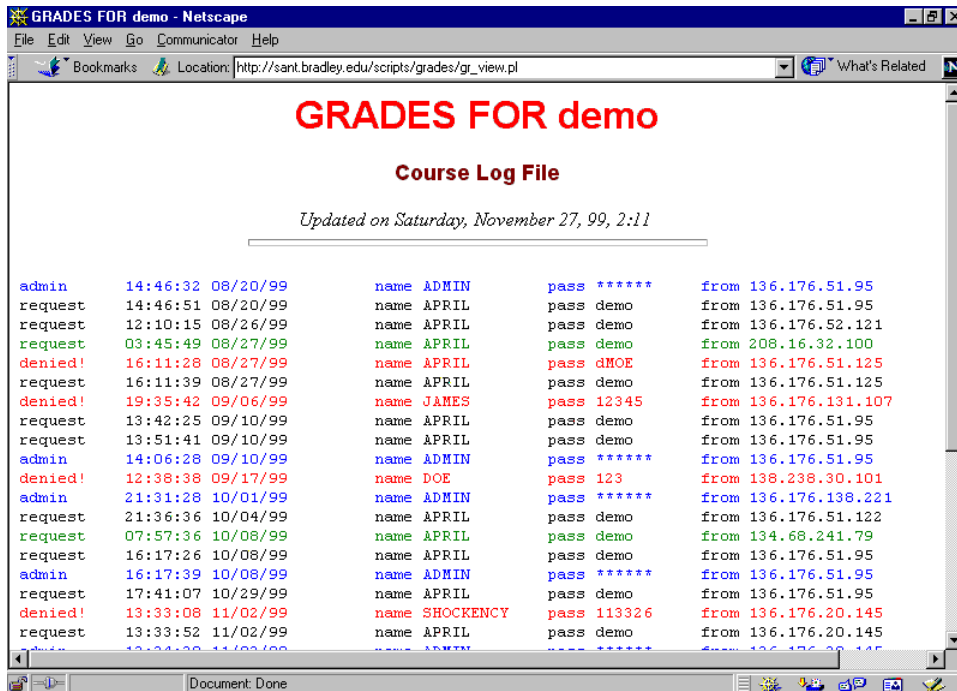


Fig. 4. Access log for the demo course. (Fragment of the actual access log)

	~header	H1	H2	TLH	TLH	EX1	PROJ.	ALL	SC	GR
	~points	115		115	200.0			200.0	100%	
123456789	Family First Middle	115		115	200.0			200.0	100.0	~!~
987654321	Malinowski Aleksander	114		114	198.3			198.3	99.1	~!~
123546789	Family First Middle	90		90	156.5			156.5	78.3	~!~
111223333	Smith John A.	115		115	200.0			200.0	100.0	~!~
	~average	108.5		108.5	188.7			188.7	94.3	
	~median	114.5		114.5	199.1			199.1	99.6	
	~min	90.0		90.0	156.5			156.5	78.3	
	~max	115.0		115.0	200.0			200.0	100.0	

Fig. 5. The course database format.

GradeWatch described in this paper can be inspected by going to <http://sant.bradley.edu/~olekmali/grades/>. Its current version can be freely downloaded and redistributed under the GNU license. The detailed help and installation manual is also available on-line.

Conclusion

The paper shows how to use the new opportunity created by Internet technologies for the efficient and platform independent grade posting system. The system has been tested for more than two academic years at three universities: University of Wyoming, Bradley University, and California Polytechnic by several instructors. The response from the students was usually enthusiastic, although some students prefer not to receive e-mail with their grades too often. To fix this problem, the instructor may move their e-mail address in the database so that it is not recognized as e-mail.

recognized as e-mail.

During the first year of its use at Bradley University, approximately 75 students accessed the system 1790 times during fall of 1998. In 1999, 55 students accessed the system 1470 times in spring of 1999 and 100 students 2100 times so far in fall of 1999. The utilization of the system is approximately 23 times per student per semester and appears to be similar each semester. The peaks of its use are observed just around midterms and final exams due to students waiting for their grades to be posted before they receive their tests back, or final grades in the mail.

Bibliography

1. Sweet, W. and Geppert, L., "http:// It has changed everything, especially our engineering thinking," *IEEE Spectrum*, January 1997, pp. 23-37.
2. Camposano, R.; Deering, S.; DeMicheli, G.; Markov, L.; Mastellone, M.; Newton, A.R.; Rabaey, J.; Rowson, J.; "What's ahead for Design on the Web", *IEEE Spectrum*, September 1998, pp. 53-63.
3. Gundavaram, S.. *CGI Programming on the World Wide Web*. O'Reilly & Associates, Inc., 1996
4. Jamsa, K.; Lalani, S.; Weakley, S.; *Web Programming*, Jamsa Press, Las Vegas, NV, 1996.
5. Wall, L., Christiansen, T., Schwartz, R.L. *Programming Perl*, O'Reilly & Associates, Inc., 1996.

BOGDAN M. WILAMOWSKI

Bogdan M. Wilamowski (IEEE Fellow) is a Professor of Electrical Engineering at the University of Wyoming. He is the treasurer of IEEE Industrial Electronics Society, a member of the IEEE Neural Network Council, an associate editor of IEEE Trans. on Neural Networks and IEEE Trans. on Education. Dr. Wilamowski is the author of 4 textbooks, more than 200 refereed publications, and 27 patents. He received his MS in computer engineering in 1966, Ph.D. in neural computing in 1970, and D.Sc. in integrated circuit design in 1977, all from the Technical University of Gdansk, Poland.

ALEKSANDER MALINOWSKI

Aleksander Malinowski is an Assistant Professor at the Electrical and Computer Engineering Department of the Bradley University. He is also an associate editor of the IEEE Industrial Electronics Society Newsletter. He received his MS in electronics in 1989 from the Technical University of Gdansk, Poland and Ph.D. in computer science and engineering in 1996 from the University of Louisville, KY. Dr. Malinowski is the author of 4 journal papers and 31 other refereed publications. The areas of his main interests are Web programming, network computing, computer simulation and modeling, computational intelligence and computer architecture.