

# A Tablet-based Paper Exam Grading System

Aaron Bloomfield  
The University of Virginia  
aaron@virginia.edu

James F. Groves  
The University of Virginia  
jgroves@virginia.edu

## ABSTRACT

We present the design and implementation of a system which allows a standard paper-based exam to be graded via tablet computers. The paper exam is given normally in a course, with a specialized footer that allows for automated recognition of each exam page. The exam pages are then scanned in via a high-speed scanner, graded by one or more people using tablet computers, and returned electronically to the students. The system provides many advantages over regular paper-based exam grading, and boasts a faster grading experience than traditional grading methods.

## Categories and Subject Descriptors

K.3.1 [Computer Uses in Education]: Computer uses in education; K.3.2 [Computer and Information Science Education]: Computer science education

## General Terms

Management

## Keywords

tablet, exam, grading, evaluation

## 1. INTRODUCTION

In large courses and distance learning courses, management of the grading of a paper-based exam becomes a time-consuming task. There can be hundreds of exams submitted (in multiple physical locations), with possibly many people helping to grade the exams. Once the exams are graded, the grade total must be calculated, the exams sorted, distributed to student locations, and then returned during class time. This uses a significant amount of instructor, teaching assistant, and administrator time during the grading process, as well as wasted lecture time to return the exam that could be better spent on other pedagogical tasks.

In an effort to alleviate the amount of time taken to grade paper exams, we have developed a system that allows the

exams to be graded completely via a tablet computer. A paper-based exam is given to the students, and is scanned in upon completion. From that point on, everything in the grading (and returning) process becomes completely digital.

There has been much research in tablet-based technology to support pedagogy. Most of this research has been in the context of using tablet computers during lecture [1, 2] or similar areas, such as providing peer-review comments in a CS1 course [3]. Research on examinations has focused on a wide variety of topics, from how to interpret exams [9], to automatic grading of programs [4], to detecting plagiarism [5, 8], to grading systems [6]. A similar grading system was presented in [7], but that system focused on grading completely online submissions, not paper-based exams.

We are not aware of any tablet-based systems for grading and management of paper-based exams.

## 2. EXAM FOOTER

The viability of this system revolves around the ability to properly recognize the details of each exam page – who it belongs to, as well as the course, exam, and page number. Modern optical character recognition algorithms are insufficient to achieve this with such a large group of people with diverse handwriting styles.

Thus, each exam has a footer in which the student must bubble in his or her userid (or other unique identification string) so that the system can recognize to whom the exam page belongs. Figure 1 shows such an exam footer. Currently, the system requires the students to bubble their information on *each* page; solving this is addressed below.

The right side of the footer contains information about the exam itself, encoded in binary. This is already encoded when the students receive their exam. The field columns are (from left to right): department, course, exam, page, and version; specific details about the meaning of the fields is beyond the scope of this article. The important aspect to note about the footer is the version field, which allows for different footers to be designed and implemented.

The left side of the exam footer is where the students bubble in their identification number or userid. All students and employees at the University of Virginia have a unique userid, which matches the regular expression  $[a-z][a-z][a-z]?([0-9][a-z][a-z])?$ . Other identification schemes, such as a 10-digit numerical identification number, could have a different footer designed. This would require a different version number (the far-right column), as well as a minor modification to the software in order to properly interpret the new footer.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITiCSE'08, June 30 - July 2, 2008, Madrid, Spain.

Copyright 2008 ACM 978-1-60558-115-6/08/06 ...\$5.00.

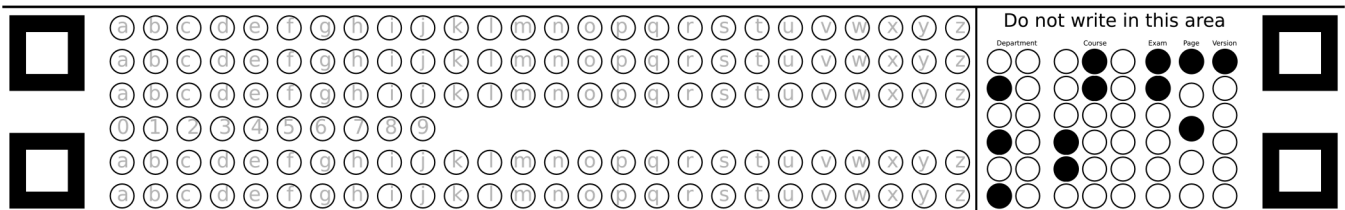


Figure 1: Exam Footer, actual size

After an exam is scanned in, the bottom 25% of the page is examined; based on the positions of the hollow squares, each row or column of information on the bubble form is interpreted. The 'darkest' bubble(s) is/are interpreted to be the value(s) for that row or column. Thus, students need not use specific writing implements, such as number 2 pencils.

Images are scanned in at 200 dpi; higher resolution images are down-sampled to 200 dpi by the system. This resolution is a balance between a high enough resolution to read and grade the exam on a tablet computer, and a low enough resolution not to require an inordinate amount of space and computation time.

A modern computer can recognize a bubbled exam in 2-3 seconds. However, this can take a significant amount of time if hundreds or thousands of exam pages have been scanned in from a single exam. Recognizing 70 10-page exams, the largest this system has been used to date, will take about a half hour to recognize. The system is designed to recognize all exam pages at once, and this process can be set to run unattended.

### 3. WEB-BASED INTERFACE

The server runs on a LAMP (Linux, Apache, MySQL, PHP) architecture, and should, in theory, be easily portable to other Unix operating systems. The majority of the system is written in PHP, and provides a web-based interface with all of the system's functionality (see Figure 6). The computationally intensive routines, such as the image recognition program, are written in C for execution speed. Javascript is the language used for client-side functionality.

The system recognizes three types of users: graders, instructors, and administrators. Each has different levels of functions that they can perform, as their names imply.

The system can operate as a stand-alone installation, or be incorporated into a larger course management system. We currently have it installed as a part of Sakai/Collab, which is the system used at the University of Virginia.

## 4. PROCEDURE

### 4.1 Exam Preparation

A paper exam is created normally, and a footer is added to each page. The easiest way to add a footer is to upload a PDF of the exam, with a 1.75 inch bottom margin. The system will then add the appropriate footer to the bottom of each page, and provide a PDF of the combined exam. The system renders each page as an image, and adds the footer to that image. Because the resulting PDF is a series of images (one per page), it is larger than the original, but not enough to make it difficult to use (a 10 page exam at 300 dpi will be approximately 2 Mb in size).

One can download individual footers as well, with or without the page numbers encoded, to add to the exam in the word processing software. This is the method employed when we used the system, as the functionality to automatically add the footer was not implemented at that time.

### 4.2 Test Day

The paper test is given with the explicit instructions for the students to enter their userid on the bottom of *each* page. As the students turn in their exams, the staples are cut off, and the stack of pages is scanned in.

We used a high-speed duplex scanner, capable of scanning in 33 gray scale double sided pages (66 images) per minute at a resolution of 200 dpi. As of the writing of this article, such a scanner can be bought for under USD \$1,000. The images scanned in were saved, in batches of 100 exam pages, to a single PDF file, with each image a separate page.

We specifically scanned our pages using gray scale, so if a student used a red pen, their markings could not be confused with the grader's markings. However, the system can also handle color scans.

The scanning of 70 10-page exams can be performed in about 10 minutes. If the scanner (which is portable) is brought to the exam room, it is estimated that 500 8-page exams could be scanned in during a 75 minute exam time, although this would probably require a second person helping to cut off the staples and feed the scanner.

### 4.3 Grading Preparation

The scanned images are uploaded into the grading system through the web-based interface. The system can accept a number of file types; we primarily used PDF, where each page is a separate exam page scan. Due to the maximum uploaded file size limits imposed by our web server, we were able to upload PDF scans of about 100 page images each at 200 dpi gray scale.

There are a number of tasks that must be done prior to grading, but not necessarily after the exams have been scanned in. Uploading the class roster, in CSV format, allows for error checking (who is missing an exam, for example) as well as customized e-mails and exam grade reports. Also, the instructor must ensure that all the graders are registered as graders for the course. This is something any instructor can do for each of his/her courses.

The exams must be run through the image recognition routines; this is automated by the system. The exam images are next reduced in resolution as a speed improvement so that the clients do not have to spend time scaling the image during grading to the lower resolution of the tablet screen.

Error checking functionality will check, among other things, that each of the students listed in the roster have the right number of exam pages, as well as the right set of exam pages.

Any such errors can be easily fixed by the instructor – typically, the problem is obvious to a human.

In practice, we found that there were three aspects that took a non-trivial amount of time.

The first part that took some time was the actual scanning in of the exams – this time was limited by our scanner model, and can be lessened by additional help or a better scanner.

The second aspect that took some time was the uploading of the exams. Because of the web server’s upload file size limit – 25 Mb, which was about 100 scanned pages – our scanned tests would consist of multiple (5-10) files. Each file needed to be uploaded individually. Although the individual uploading is easy, it does require the instructor to return after the file has been uploaded (a significant time later, as the uploaded image needs to be processed) and upload the next file. A forthcoming enhancement is to create a utility that will take a given directory or set of files and batch upload each of the file(s), thus eliminating the time taken for this step.

The third aspect that took some time was the correcting of the image recognition errors – most of these errors were a userid that the system did not recognize properly. This time varied greatly – one exam in a smaller class (35 students, 10 pages per exam) had no recognition errors, while an exam in a larger class (70 students, 10 pages per exam) had almost 10% recognition errors. The average number of exam pages with errors was under 5% – this is discussed further below.

#### 4.4 Grading

The exam is graded using tablet computers. Graders mark up the exam using digital ink, and can enter text (rendered from TrueType fonts) onto the exam, as described below. The actual writing of the marks and text on the digital copy of the exam uses the canvas HTML element, which is not available in some browsers, notably Internet Explorer. For that reason, the system only uses (and has only been tested with) Mozilla’s Firefox browser.



Figure 2: A tablet exam grading session

Figure 3 shows a screen shot of an exam that is in the process of being graded. The tablets we used had a resolution of 768x1024. The particular exam page scan is being used with the student’s permission, although his/her identification information has been changed or removed. The gray

scale reproductions of this article will not display it properly, but the rendered text on the exam page (‘good job’ and ‘right name for this algorithm’), as well as the two checkmarks on the page, are all in red; the rest of the exam page is in gray scale.

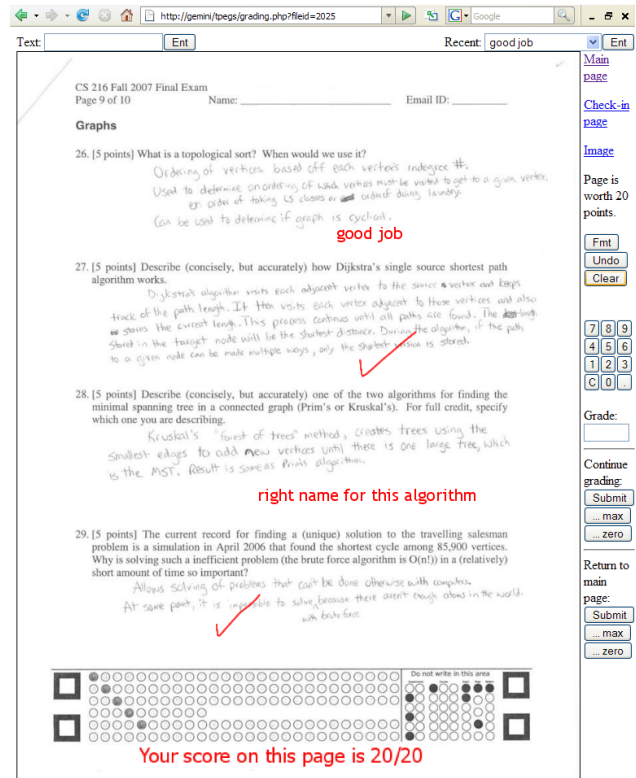


Figure 3: A partially graded exam page

As the grading in this system is designed to be performed primarily with a stylus – and with as little keyboard input as possible – the grade entry is done via the buttons on the right. The ‘image’ link above that allows for viewing of the full-sized scan, which is useful for hard to read exam pages.

In Figure 3, there is both text – rendered from the Tahoma TrueType font – and line markings that the grader entered. The toolbar at the top of the screen (below the Firefox toolbar) allows for the graders to enter text. On the left side of the toolbar, they can enter text one character at a time, either by using the physical keyboard, or through the tablet’s on-screen keyboard. The grader then uses the stylus to select where on the exam page to place the text. Once a text comment has been entered, it is saved in the drop-down box in the upper right. Thus, a grader can re-use comments that he or she entered without having to type them in each time. Figure 4(a) shows the drop-down box that is populated with recent text entries, the most recently used at the top.

The toolbar at the top of the page can be changed by pressing the ‘Fmt’ (short for ‘Format’) button on the right of the screen. The toolbar it is changed to is shown in Figure 5. This toolbar allows the grader to control two things. The vertical offset changes how far above or below the stylus the entered marks would appear, as this varied with different tablet models. The drop-down box on the right side of that toolbar allows the graders to select which pages they will see

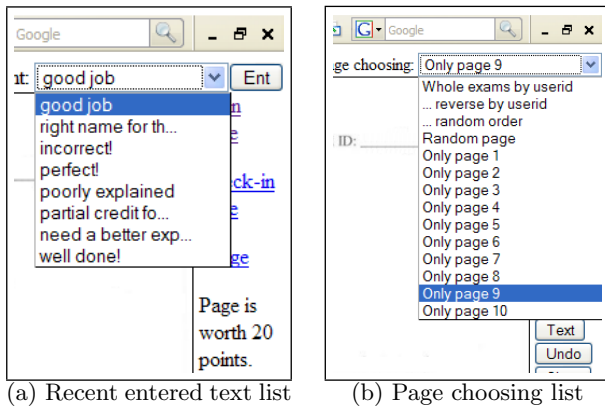


Figure 4: The expanded drop-down lists

once they finish grading this page. The options are shown in Figure 4(b). This allows one grader to only grade a given page of an exam, for example. Both of these are saved as browser cookies, and do not have to be set again unless they change.

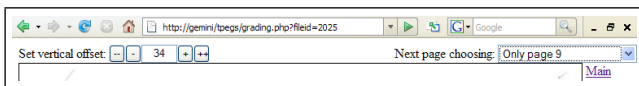


Figure 5: The formatting toolbar

The system allows for undoing of marks made during the grading, as well as complete re-grading of an already graded exam page. Graders can see which exam pages they have graded, and go back and change their marks and/or grades. Instructors can see statistics on the grading progress.

### 4.5 Post-grading Processing

After the grading, the instructor must perform a number of tasks to send out the grade results. These are easily done through the web-based interface, which is shown in Figure 6.

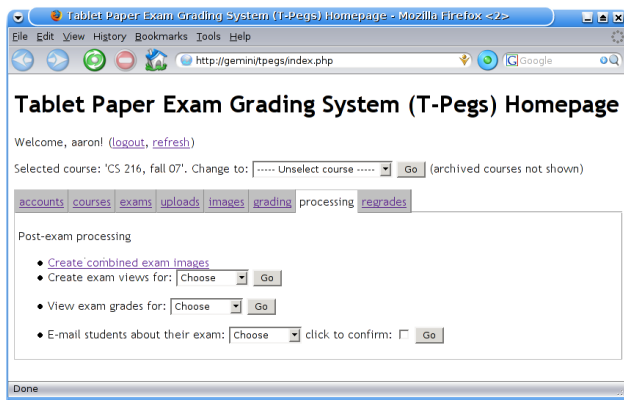


Figure 6: The web-based interface

These tasks include processing the images (combining the grader’s ink and text with the original image), formatting the images for viewing by the students, and finally e-mailing the results. The e-mail each student receives contains a link

to view their particular exam; part of the link is an MD5 hash of a semi-random string, thus it is not likely that one could guess a URL. Alternatively, students can view their exams through a course management system.

While these steps take a large amount of computing time, they are very rapid to perform – the HTML forms need to perform these (and other) steps can be seen in Figure 6. The commands are grouped together to allow for easy recall of the steps.

Note that the image post-processing is not done during the grading so as to allow for rapid loading of the next page to be graded. Once the exam e-mails have been sent out to the students, the server load is relatively light, as they are just requesting data directly from the database – no additional processing is needed. The exams are provided in two formats: a web page with embedded exam page JPEG images, and a PDF file. The vast majority of the students used only the PDF version of the graded exam.

The returned exam shows the scan of each original page with the digital ink overlaid on top of the scanned image. The exam pages were scanned in gray scale, and the digital ink is in red. Thus, it was always clear what is an original marking on the exam and what is a marking by a grader.

The system also allows students to submit a regrade request, if they feel their exam was not graded properly. This is disabled by default, and needs to be explicitly enabled by the instructor. Handling of the regrades, and thus changing the exam grades, is also supported.

A CSV of the exam grades is easily available to the instructor for loading the grades into a grade book system.

## 5. RESULTS

This system was used in two courses during the fall 2007 semester. One course had 70 students and 3 exams, the other had 35 students and 2 exams. Both had approximately 10 pages per exam, although one or two pages on each exam, such as the cover page, had no questions.

The tablets we used for this system were 1 GHz Pentium machines, with 512 Mb RAM, 768x1024 resolution, and Windows XP. While these tablets are slow by current standards, a number of optimizations on both the server side and the client side allowed the grading process to be quite rapid. The server was a 3 GHz Pentium computer with 2 Gb RAM running Debian Linux. The tablets were connected to the server via a wired network and a dedicated Ethernet switch – we used a wired network for speed and stability reasons.

Our grading sessions consisted of up to 9 individuals who used the system simultaneously, while all in the same room, as shown in Figure 2. Although the system does allow for remote grading, we had everybody in the same room to allow discussion about the grading policies for individual questions.

We found that image recognition errors ranged from 0% to 10% of the scanned exams, but averaged under 5%. The variability depended largely on how well the students followed the directions. Many students would miss entering their userids on a page or two, would miss-bubble an entry, or would not darken it enough. We required the users to also write out their name and userid at the top of each exam page, so that when an exam page is viewed, a human could tell to whom it belongs. The web-based interface allowed for easy correction of incorrectly recognized userids.

The first use of the system took some additional grading time, as the graders needed to learn the system interface and become accustomed with entering the marks on the exams. Successive grading sessions eliminated this training time.

The time taken to grade an individual page was on par with, or faster than, the time taken to grade an individual page of a paper exam. This was measured subjectively – all participants who used the system, both teaching assistants and faculty, agreed on this claim. A few pages would take longer, such as when a student wrote in very small handwriting, and the larger exam scan needed to be loaded to read the student’s marks. However, this only occurred on about 5% of the pages. The time taken to process a page when the grade was entered, and load up the next page, was on the order of 2 seconds; this would be reduced if the tablets were faster machines. When not connected to the server via a dedicated Ethernet switch, the time to load up the next exam page to grade was still rapid – in one situation, it took 3-5 seconds per page when grading from a hotel location thousands of miles away. Off-line functionality, described below, would eliminate this delay.

The time taken for other tasks during the grading of paper-based exams – finding the next exam to be graded, flipping to that page, managing the piles of graded and ungraded exams, adding up the points on each page to yield the total (and then checking that total), and entering all the grades into a grade book program – was completely eliminated.

Subjectively, the graders felt that the system was easy to use and helped the efficiency of the grading process.

The turn-around time from the student’s perspective was very high. For the exams in which we were able to schedule a grading session the evening of the day of the test (for each exam we completed the grading in a single grading session), the students were e-mailed back their exams that night.

Occurrences of cheating were reduced (although, sadly, not eliminated) with the use of this system, as students were not able to fill in their answers later, and then claim that their answer was ‘missed’ during the original grading.

## 6. FUTURE WORK

One of the limitations of the system is that the students must bubble in their userid on *each* page. The motivation behind this is because the pages might become separated or shuffled prior to scanning. This causes extra time to be spent filling out the bubble forms during the exam, although less than 30 seconds per exam page. We have a number of ideas to pursue, including only bubbling in the information on one side of the double sided pages, or using an image merge functionality to allow each exam copy to have a unique barcode, and thus only requiring the students to bubble in their information on one page. We are actively working on reducing the amount of time the students must spend entering their information into the bottom of each exam.

Another direction we plan to pursue is to add additional grading clients. Currently, the system requires the graders to be connected to the server via a network. Another concept is for the grading client to be able to “check-out” multiple pages of an exam (perhaps all), and keep them locally on the tablet. A user could then grade them when not connected to the network (such as on an airplane), thus avoiding the problem of having to bring the paper exams along on a trip, for example. We have the initial implementation of such a client written for a Windows platform.

Lastly, we are very aware that increased adoption of the system will depend on reducing the amount of administrative time needed to manage the exams. We have a number of ideas in this regard, including creating a program that will upload a series of exam scans in one pass, rather than the current version that requires the user to upload each file individually. Additional client customization options, such as switching the side that the toolbars are on, is another area we plan to pursue.

## 7. ACKNOWLEDGEMENTS

This work was funded by The Virginia Tobacco Indemnification and Community Revitalization Commission as part of an effort to extend undergraduate engineering education to rural communities in Virginia. Also, the cooperation of the Institute for Advanced Learning and Research in Danville, Virginia helped to make this work possible.

## 8. REFERENCES

- [1] R. Anderson, R. Anderson, K. M. Davis, N. Linnell, C. Prince, and V. Razmov. Supporting active learning and example based instruction with classroom technology. *SIGCSE Bull.*, 39(1):69–73, 2007.
- [2] D. Berque, T. Bonebright, and M. Whitesell. Using pen-based computers across the computer science curriculum. *SIGCSE Bull.*, 36(1):61–65, 2004.
- [3] E. F. Gehringer. Electronic peer review and peer grading in computer-science courses. *Proceedings of the 32nd SIGCSE technical symposium on Computer Science Education*, pages 139–143, February 2001.
- [4] M. T. Helmick. Interface-based programming assignments and automatic grading of java programs. In *ITiCSE '07: Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 63–67, New York, NY, USA, 2007. ACM.
- [5] J.-H. Ji, G. Woo, and H.-G. Cho. A source code linearization technique for detecting plagiarized programs. In *ITiCSE '07: Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 73–77, New York, NY, USA, 2007. ACM.
- [6] R. E. Noonan. The back end of a grading system. In *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 56–60, New York, NY, USA, 2006. ACM.
- [7] J. L. Popyack, N. Herrmann, B. Char, P. Zoski, C. Cera1, and R. Lass. Pen-based electronic grading of online student submissions. *Syllabus*, 18-20, January 2003.
- [8] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: local algorithms for document fingerprinting. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 76–85, New York, NY, USA, 2003. ACM.
- [9] P. Thomas, N. Smith, and K. Waugh. Computer assisted assessment of diagrams. In *ITiCSE '07: Proceedings of the 12th SIGCSE conference on Innovation and technology in computer science education*, pages 68–72, New York, NY, USA, 2007.