Integrating a Research Component into a Web Development Introductory Course

Amalia Rusu and Bridget Keyes
Fairfield University, arusu@mail.fairfield.edu, keyesbr@member.fairfield.edu

Abstract - **Exposing students to research topics enhances their learning experience. It is therefore desirable for instructors to include a research component into their courses. The courses most prone for incorporating research components are the upper level undergraduate and graduate courses. However, this becomes challenging in the case of introductory courses, where learning new programming skills is the focus. In this case, the research component needs to satisfy two key conditions: to be easy enough to implement, and to have a fast learning curve. In this paper we describe the introduction of a research component into an introductory web development course (upper undergraduate and lower graduate) for software engineering students. In this context we present a web security research application that is successful at keeping unwanted malicious attacks from happening on web services. It can be implemented using basic web development skills and fast to learn and rationale. Due to the interdisciplinary nature of the research problem, the students have been exposed to techniques and research topics in several other fields. The projects outcome including student experiences conducting user studies and human surveys are discussed in the paper.**

Index Terms - Interdisciplinary project, Software engineering education, Undergraduate and Graduate, Web application (CAPTCHA).

INTRODUCTION

Exposure to real-world development or applied research projects enhances students' learning experience and greatly improves their performance in the industry after graduation. However, in the case of introductory web development courses, the focus is more on learning new programming skills rather than on research or real-world projects. Generally the purpose of these courses is teaching web site design and problem solving with various technologies that interact on the Internet. In particular, the course is an introduction to web design that covers client side web technologies such as XHTML, CSS, and Javascript.

In this paper we present a novel approach for teaching an introductory web development course, as part of a software engineering curriculum, to upper level undergraduates and lower level graduate students within a semester long course. This course incorporates the study of web development techniques, while allowing each student to work in teams and experience research and real-world projects. We also present a successful student story based on the outcome of Fall 2007 semester when, in addition to regular readings, homework, and other coursework, the class also required a semester long class project due in the final week of class that allowed the students to succeed in developing a marketable web security application.

The difficulties associated with real-world and research projects often deter the faculty in pursuing them. Such difficulties include locating appropriate projects, increased faculty workload to manage those projects in the absence of additional teaching assistants as well as the time constraints to complete the project by the end of semester. However, through a careful planning and projects selection on the instructor side, and students' coordination and commitment to the project, most of these difficulties can be overcome.

In this context we introduce CAPTCHAs as *Completely Automatic Turing Tests to Tell Computers and Humans Apart* ([1]-[2]) over the Internet that are successful at keeping unwanted malicious attacks from happening on web services. The reason CAPTCHAs were selected as the research component for the course is that they satisfy the two key conditions:

- CAPTCHA is a web security application which can be implemented using basic web development skills.
- CAPTCHAs are fast to learn and rationale (it usually takes a lecture or two to provide the background and describe the latest innovations).

Moreover, due to the interdisciplinary nature of CAPTCHA, the students have been exposed to techniques and research topics in other fields such as Pattern Recognition, Image Processing, Artificial Intelligence, Graph Drawing, Language Understanding, Computer Security, Human-Computer Interaction, and Cognitive Science.

The rest of the paper is organized as follows. We present the background on the CAPTCHA projects and then detail on our approach. Successful student stories and student reactions to our approach are included as well. Finally, we conclude with a summary of major contributions.

BACKGROUND

The purpose of CAPTCHA is to keep unwanted intruders (e.g., hackers, spammers) from accessing particular web services. CAPTCHA can virtually tell the difference between a human user and computer-generated attack. The current popular CAPTCHA tests use random distorted text images to challenge a user before entry is allowed. They can be seen on such websites as Yahoo, Gmail, and Tickemaster (see examples of CAPTCHAs in Figure 1). These random distorted text images were initially successful at allowing only humans to pass these tests. However, for certain websites the financial incentive to break past the CAPTCHA is so great that hackers will use sophisticated means to bypass it. Thus, as one CAPTCHA is created, there is someone out there trying to defeat it. Even industry giants are not immune. Recent news reports suggest that the three most popular web mail applications, Microsoft's Hotmail [3], Yahoo Mail [4], and Google's Gmail [5], have all had theirs cracked.

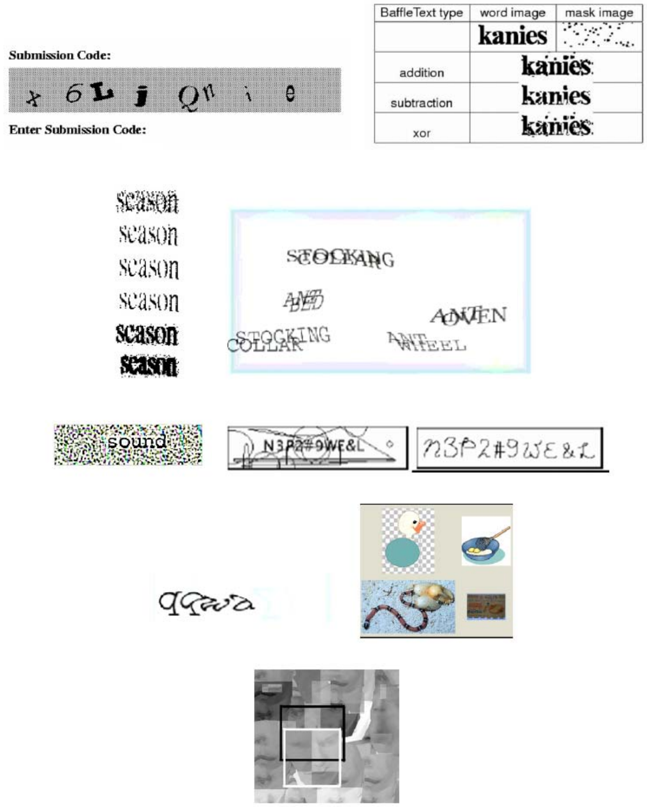


FIGURE 1
VARIOUS CAPTCHA TESTS.

CAPTCHA is considered to be a reverse Turing test. A Turing test is an experiment the famous computer scientist Alan Turing proposed back in 1950 in which a human judge presents questions to two entities: a computer and a human, and then used the answers they gave to determine which one was human. CAPTCHA is similar in that the test was to determine human intelligence versus machine intelligence but kind of the reverse in the sense that a human was the judge of the correct answers whereas in CAPTCHA a machine is the judge.

The goal for the projects was to create a new type of CAPTCHA that would be more difficult to break. To do this the students needed to create a more complex CAPTCHA than the ones employing distorted machine printed words or letters that are familiar to most web users. However, despite the added complexity it should remain easy for humans to solve. CAPTCHA cannot create an undue burden on the user or else he or she may be unable to obtain access or give up.

While the technology needed to implement the projects was not difficult to learn, it required much more conceptual thought than is typically required in most introductory level courses. The projects also had a research component with real-world impact whereas most programming courses are either more theoretical or less applicable in practice. One could argue that all web users have a vested interest in seeing CAPTCHA succeed. Being more involved in the design and spurred on by the fact that it may be used one day, the students were inspired to produce the best possible work.

PROJECTS DESCRIPTIONS

Implementations of visual challenge tests have been successful for many years at keeping unwanted attacks from happening on web services. New ideas and theories have been created to keep everybody guessing at how to solve the next best thing. Sometimes the end user, the one that they're trying to protect, is the one that cannot figure out how to pass the test. From what has been researched, these original ideas of creating visual challenges are all based on known information that can be interoperated and used to break the algorithm of augmentation.

The purpose of this study is to explore the idea of employing a random CAPTCHA that uses visual graphical images to challenge the user for an input. The challenge test will be designed and generated at random, and the question that it triggers based on the design will be at random too. With this "on the fly" type of test design, this makes the chance of a less likely computer attack and yet easily passable test by humans.

We proposed a novel challenge test called a tree-based handwriting CAPTCHA. The tree-based handwriting CAPTCHA system has been designed as a challenge-response protocol for web security (Figure 2). The authentication is performed in four steps:

- *Initialization:* the user expresses an interest in being authenticated by the server.
- *CAPTCHA Challenge:* the server generates a challenge in the form of a handwritten word image and issues it to the server.
- *User Response:* the user has to key in the right answer and return it to the server.
- *Verification:* the server verifies the user response and checks if it matches the right answer. It either grants access to the user or rejects the transaction.

The handwriting generation tool is written in MATLAB and it outputs handwritten images in the form of .png files. The tree generation code is written in the Java programming language. This code uses the generated images and randomized values to draw the tree. The final challenge is displayed on the website as a Java applet.

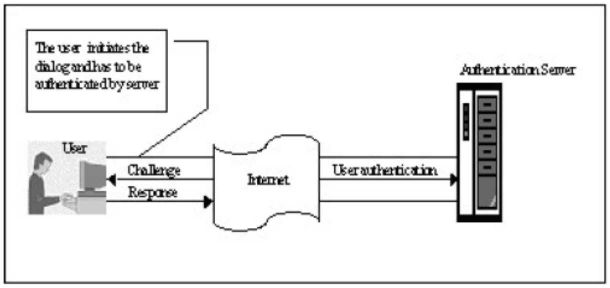


FIGURE 2
AUTOMATIC AUTHENTICATION SESSION FOR WEB SERVICES USING TREE-BASED HANDWRITTEN CAPTCHA.

With the proper implementation of this styled CAPTCHA we can ensure randomness and unpredictable results. The questions associated with this type of CAPTCHA can be automatically generated as the tree is generated. Some of the questions that can be generated instantly are; “What image is between the triangle and the rectangle?” or “How many images are there of same shape?” or “Which city is connected to another city by a line marked with a circle?” or “What shape is on the line between this word and that word?”. Since the shape of the tree and the questions are generated randomly, it poses a higher level of difficulty to computers but not to humans.

The generation algorithm uses randomness whenever possible to create the tree structure and images. The program begins with generating a random number of nodes. Once the number of nodes has been determined, the algorithm begins inserting them into a binary tree data structure. Using a randomly selected scaling and sizing algorithm that ensures that the visual representation of the tree never looks the same, the program places the drawing on the applet window. During the drawing phase, when the node is created, a random image of handwritten text will be place next to the location. Currently, this image is randomly selected from a list of possible images but in the future will be generated on the fly. For each tree branch that is created, a randomly selected symbol is placed in the middle of the branch – one of three shapes, circle, triangle or square. Finally once the tree with proper placement of images and symbols has been completed, a question will be selected at random from a list of potential questions about the tree. The challenges in the generation algorithm are creating a visual challenge test that is easily legible to read and pass by humans but unable to be cracked by computers.

To summarize, the visual challenge test is created at random and the question that is asked is based on the random generated image with the potential for generating infinitely many such challenges. These results can be seen in Figure 3.

CAPTCHA has two main requirements, that it is easy for a human to understand but difficult for computers. Therefore we conducted usability tests to validate our concept. Various tree-based handwritten CAPTCHAs have been administered on several subjects. Some of the questions we have asked when presented the challenges were:

- What is your first impression regarding the test?
- How would you rate the difficulty of the test?
- Questions or comments?

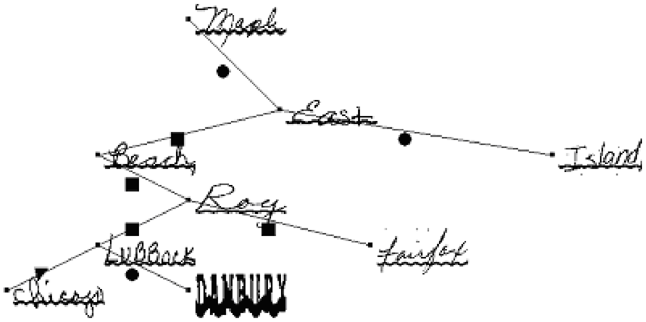
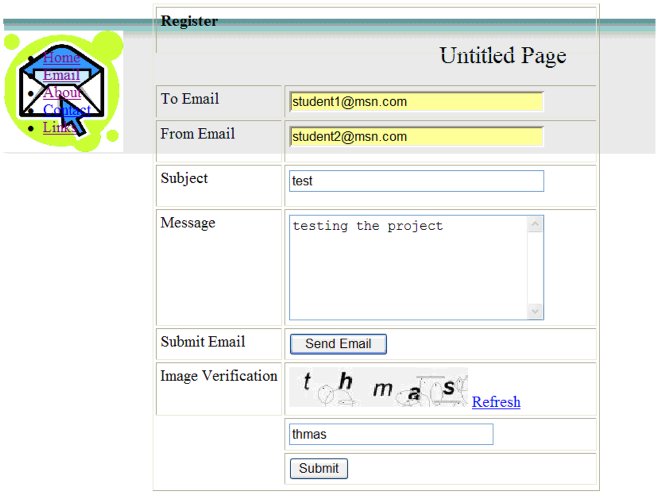


FIGURE 3
EXAMPLE OF TREE-BASED HANDWRITTEN CAPTCHA.

On the other hand, we have chosen CAPTCHA for our introductory web development course because of its many web applications. Other students have also investigated and developed another web security application that is a new authentication mechanism of e-mail addresses. Students’ solution and the concept of personalizing the email addresses are shown in Figure 4. This novel application refers to creating transformed alias e-mail addresses to make them unreadable to automatic robot senders to prevent mining by software agents. For instance, in order to contact a person by email, first receive back an alias email address to use instead of the original one CAPTCHA which our generation algorithm will render, or any CAPTCHA challenge. Since this alternative is suitable for personalizing email addresses, we investigated an anti-spam system based on sender verification. It can be created in such a way that it blocks all junk messages from being sent to real email addresses.



To Email	student1@msn.com
From Email	student2@msn.com
Subject	test
Message	testing the project
Submit Email	Send Email
Image Verification	t h m a s Refresh
	thmas
	Submit

FIGURE 4
NEW AUTHENTICATION MECHANISM OF E-MAIL ADDRESSES.

STUDENTS' EXPERIENCES

In order to realistically capture students' perspective on the approach, this section has minimal input from the instructor.

I. Project Challenges

Working in teams provided both benefits and challenges. Despite the fact that most academic coursework is done by the individual student, very few real world software projects are done singly. This provided valuable experience considering as the teammates had never met prior to working together and possessed very different backgrounds, including fields of study.

There are clearly numerous ways to implement a CAPTCHA. Being of different minds, there were some different ideas about possible solutions and ways of implementing them. However, after some discussion of the advantages and disadvantages of each, a tree-based structure was decided upon.

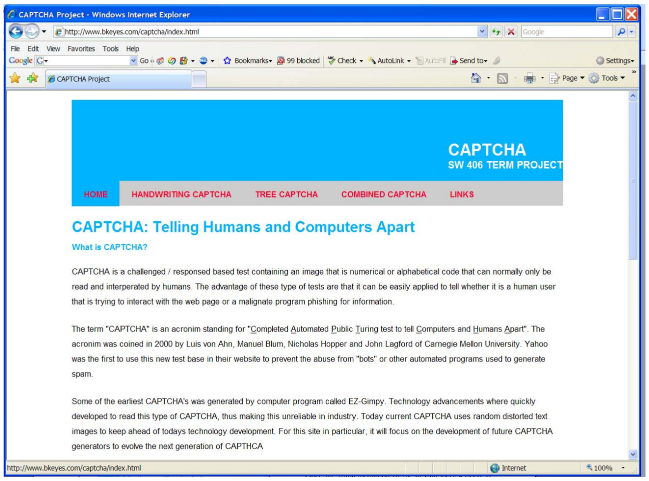
On the technical side, there were also challenges. A tool needed for generating the tests (the handwriting generator) was written in MATLAB, a tool with which teammates were not too comfortable. The output of that program is then passed to a Java applet and displayed on the website. On the other hand Java was a programming language with which not all teammates were familiar as well.

Of course, having only one semester to work, the biggest challenge faced was actually time so that a fully functional prototype that implemented all the main functionality was successfully created with no room for many enhancements. Though there are many more possible additions and improvements, we were satisfied with the amount of work that we did in the time provided.

II. Results

We began researching CAPTCHA, what it was, what had already been done and what had been successful. Learning about previous work was fascinating. There were so many different approaches. It showed how many different disciplines are involved in CAPTCHA from psychology to computer science and security. Then, we built the website where we would be posting our CAPTCHA application. The site included links to previous works and information about what we had learned.

Our CAPTCHA system has three main components: (i) the actual CAPTCHA challenge (tree-based handwritten image) that is presented to the user, (ii) user response or the answer to the challenge, and (iii) a method to validate the user response and report success or failure. The modules can be used independently to secure any online application. The process is entirely automatic so that it is easily deployable and there is no risk of image repetition, which ensures higher security. The three components have been implemented and tested online at our project URL: http://www.bkeyes.com/captcha/index.html (Figure 5).


CAPTCHA Project - Windows Internet Explorer
http://www.bkeyes.com/captcha/index.html
Google
File Edit View Favorites Tools Help
Google G Go Bookmarks 99 blocked Check AutoLink AutoFill Send to Settings
CAPTCHA Project
Page Tools
CAPTCHA
SW 406 TERM PROJECT
HOME
HANDWRITING CAPTCHA
TREE CAPTCHA
COMBINED CAPTCHA
LINKS
http://www.bkeyes.com/captcha/index.html
Internet
100%

CAPTCHA: Telling Humans and Computers Apart

What is CAPTCHA?

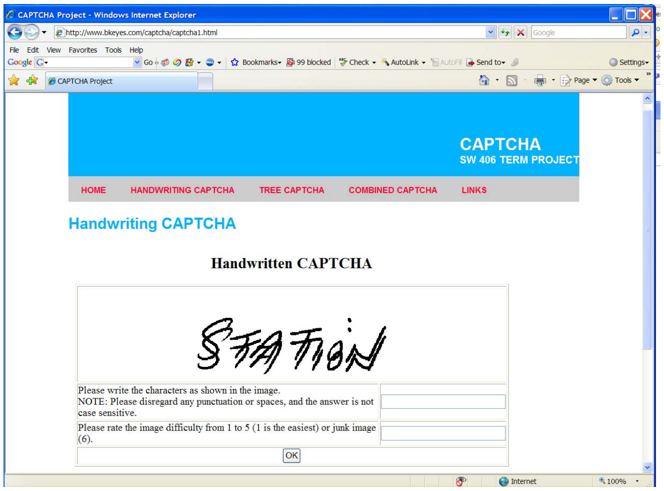
CAPTCHA is a challenged / responsed based test containing an image that is numerical or alphabetical code that can normally only be read and interperated by humans. The advantage of these type of tests are that it can be easily applied to tell whether it is a human user that is trying to interact with the web page or a malignate program phishing for information.

The term "CAPTCHA" is an acronim standing for "Completed Automated Public Turing test to tell Computers and Humans Apart". The acronim was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Lagford of Carnegie Mellon University. Yahoo was the first to use this new test base in their website to prevent the abuse from "bots" or other automated programs used to generate spam.

Some of the earliest CAPTCHA's was generated by computer program called EZ-Gimpy. Technology advancements where quickly developed to read this type of CAPTCHA, thus making this unreliable in industry. Today current CAPTCHA uses random distorted text images to keep ahead of todays technology development. For this site in particular, it will focus on the development of future CAPTCHA generators to evolve the next generation of CAPTHCA

FIGURE 5
PROJECT WEBSITE.

The handwritten tree CAPTCHA can be found listed under "Combined CAPTCHA." The website also has working examples of a CAPTCHA using just the handwriting generation (Figure 6) as well as one employing only a tree structure (Figure 7). It also includes more information and links about general CAPTCHAs.


CAPTCHA Project - Windows Internet Explorer
http://www.bkeyes.com/captcha/captcha1.html
Google
File Edit View Favorites Tools Help
Google G Go Bookmarks 99 blocked Check AutoLink AutoFill Send to Settings
CAPTCHA Project
Page Tools
CAPTCHA
SW 406 TERM PROJECT
HOME
HANDWRITING CAPTCHA
TREE CAPTCHA
COMBINED CAPTCHA
LINKS
STATION
OK
Internet
100%

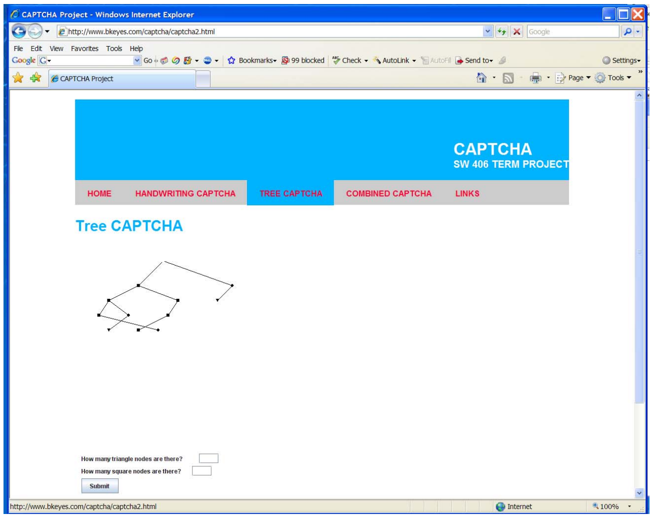
Handwriting CAPTCHA

Handwritten CAPTCHA

Please write the characters as shown in the image. NOTE: Please disregard any punctuation or spaces, and the answer is not case sensitive.	
Please rate the image difficulty from 1 to 5 (1 is the easiest) or junk image (6).	

FIGURE 6
HANDWRITING CAPTCHA.

Once the implementation of the design as described in the previous section has been done, the second step was to test whether or not our design was a functional CAPTCHA. As mentioned earlier, CAPTCHA has two main requirements, that it be easy for a human to understand but difficult for a computer. To test the former, a small number of user interaction evaluations were done.


CAPTCHA Project - Windows Internet Explorer
http://www.bkeyes.com/captcha/captcha2.html
File Edit View Favorites Tools Help
CAPTCHA Project
CAPTCHA
SW 406 TERM PROJECT
HOME
HANDWRITING CAPTCHA
TREE CAPTCHA
COMBINED CAPTCHA
LINKS
Submit
Internet
100%

Tree CAPTCHA

How many triangle nodes are there?

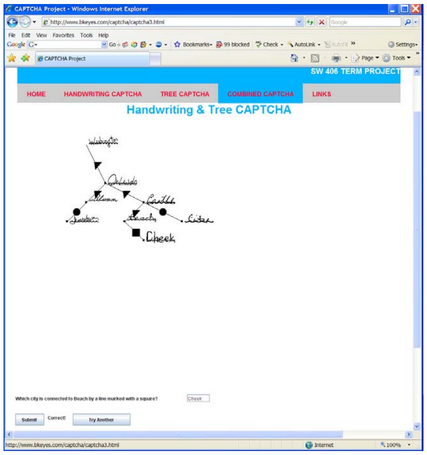
How many square nodes are there?

FIGURE 7
TREE-BASED CAPTCHA.

During user studies, many respondents thought the test were initially overwhelming or intimidating. This is probably because it looks so different than the CAPTCHAs typically seen on the web. However, a participant answered a question incorrectly only once. The majority of participants answered all of their questions correctly. The response of the users has been encouraging. One impressed tester told us that he thought our trees looked "so different each time" that it would be extremely difficult for a computer to answer.

III. Lessons Learned

Overall, the experience was a success. The teammates were able to produce a fully functional web security system of a tree-based handwritten CAPTCHA (Figure 8). Though there is still potential for further development to the current system, our prototype demonstrates that this type of CAPTCHA is viable and that users would be able to solve it correctly.



Handwriting & Tree CAPTCHA

Which city is connected to Beach by a line marked with a square? Cheek

Correct!

FIGURE 8
HANDWRITING TREE-BASED CAPTCHA.

The project also provided us the opportunity to work with an interesting, interdisciplinary tool with real world application that might not otherwise been possible. On a personal level, we felt more invested and put more work into this project than other assignments.

An additional benefit of working on the CAPTCHA project was getting fellow classmates and peers involved with the work through user studies. The feedback they provided was very valuable. The overall reaction was encouraging and gratifying. We also got more ideas for possible future additions that will further improve the design.

CONCLUSIONS

As a student in an introductory web development course, one embarks on a comprehensive study of technologies and concepts required for developing web sites. Students will learn how to efficiently create static and dynamic web pages using XHTML and client-side scripting languages. The purpose of this course is mainly teaching web site design and problem solving with various technologies that interact on the Internet.

In this paper we have presented a novel approach for teaching introductory web development courses by introducing a research component that is easy to learn and rationale. Moreover, if the research component requires learning of interdisciplinary techniques and has a real-world impact, it truly benefits all the participants. In our case, everyone involved was able to see the evolution from the initial background on CAPTCHA concept into a fully functional challenge-response protocol that uses a more sophisticated CAPTCHA, which can potentially replace the existing CAPTCHAs currently used on Internet. Nevertheless, our current web software system can continue to be expanded in numerous directions and used to secure various online applications such as suppressing spam and worms, thwarting password guessing, blocking denial-of-service attacks, preventing ballot stuffing, protecting databases, etc.

During this course, the students needed to follow two concurrent paths in order to successfully complete a task of this nature, one path being a real-world applicable project and the other being the study of relevant web development topics in textbooks [6]-[8]. Our approach anticipates the forthcoming challenges students will face in industry and help ignite their entrepreneurial spirit by presenting real-world challenges for which they need to find real solutions.

The students have also learned the importance of research by allocating time to details before embarking in a new technical domain. Generally, the results and project performance can be improved if research is completed. They also learned to make specific efforts to address user studies and ensure that the system they develop is intuitive and rational. We found that the potential of developing valuable solutions and marketable software products by spending only limited resources increases if the students are presented with applications derived from real-world problems.

Overall, our approach has overcome the difficulties associated with bringing real-world problems into the classroom by providing a source of projects of variable size and level of difficulty related to the technologies and concepts taught in class.

ACKNOWLEDGMENT

We thank students Phil Saunders and Drew Iorio for their contribution to the projects.

REFERENCES

[1] Blum, M., Von Ahn, L., Langford, J., Hopper, N. 2000. "The captcha project: Completely automatic public Turing test to tell computers and humans apart". http://www.captcha.net.

[2] Von Ahn, L., Blum, M., Langford, J. "Telling humans and computers apart (automatically) or how lazy cryptographers do AI". February 2002. *Technical Report TR CMU-CS-02-117*.

[3] "Streamlined anti-CAPTCHA operations by spammers on Microsoft Windows Live Mail". http://www.websense.com/securitylabs/blog/blog.php?BlogID=171. Accessed: 6 February 2008.

[4] "Yahoo's CAPTCHA Security Reportedly Broken". http://www.informationweek.com/news/showArticle.jhtml?articleID=205900620. Accessed: 17 January 2008.

[5] "Google's CAPTCHA busted in recent spammer tactics". http://www.websense.com/securitylabs/blog/blog.php?BlogID=174. Accessed: 22 February 2008.

[6] Felke-Morris, Terry. 2007. *Web Development & Design Foundations with XHTML*. Addison-Wesley.

[7] Callihan, Steve. *Learn HTML in a Weekend*. Prima Publishing.

[8] Lin, Forest. *QuickStart to JavaScript*.