Final Report

CAPTCHA SAM

Subtraction

Addition

Multiplication

Name: Baldeep Birak

Course: CNMD 2007/08

1 Abstract

This report looks into the ECE Intranet and reveals a security vulnerability with the current system. It looks into the implementation of a CAPTCHA with a web based login page. There was a feedback questionnaire that required the participation of staff and students of the department. Questions were asked about the use of a new system and the analysis goes into detail about how the respondents would feel if there was a possible switch over.

CAPTCHAs are given a bad name because of how big companies implement them on their websites. This report reveals both good and bad examples of CAPTCHAs from some of the big companies around and explains why this is the case. It also looks into alternative methods for companies that wish to use CAPTCHAs to help pass the authentication process of a registration, system login or even a blog post.

There are some alternative methods to CAPTCHAs mentioned in the report that maybe workarounds, however there are also reasons why these are not so good in a real world example. This would raise questions to some of the articles on the subject matter from large organisations (such as W3C) and why they may wish to retract their statements.

Contents

2 Summary

This project is about making an authentication system that makes the University Intranet login more secure. The problem with the current system is that anyone can keep trying to login (human or bot using brute force methods). There is no limit on the amount of guesses a user can make. This adds a security vulnerability with the possibility of University content (such as the new forum) leaking out.

The simple way to fix this is to limit the number of attempts a user can make to login before locking them out temporarily. The more secure method is to have a web page with a login dialog box and prompting for the user to submit their account details. This would ensure the user that the website is genuine from the website URL, the padlock symbol in the browser and the use of a web page that is themed with the Intranet template. In addition to the login box a CAPTCHA can be added as extra measure of security. The reason for this is that a bot cannot read an image (unless programmed to do so) whereas a human can.

After the implementation of the CAPTCHA, the next task was to assists against brute force attacks. This was done by counting the number of attempts made by the user before locking them out of the system. The number of attempts set in the software (for this project) was set to ten. This was set to a high number to assist the respondent by not locking them out of the system. The developer required feedback from as many respondents possible and by not locking them out, he could maximise the potential number of successful logins.

To assist against denial of service, a timer needed to be implemented to allow the user back into the system. This could not be too short that it's not worth having or too long that no one comes back to the website. After careful consideration the idea was to set the timer to five minutes. When a user has been locked out by the system their user account credentials would be ignored for the set time limit. If the user stopped logging in during this time limit then the counter would reset. After a counter reset, another set of attempts were allowed and this process looped.

To prevent the user account from being locked out indefinitely, a method was needed to distinguish where the "supposed" attack was taking place. To find out this information out, a

function in PHP was used to find out the user's IP address. This information was inserted into a MySQL database. The same table was used to lookup the user's IP address, find the current number of login attempts and the time of the last attempt. By using the IP address, the block prevented the "supposed attacker" accessing the system. This way the user was not affecting genuine users from using the system.

The login attempts were updated by using the respondent's IP address. The reason for this is that in a real world scenario such as emails, companies block emails coming from certain IP addresses (Rickert 2008). Another scenario is where a company or organisation may block SSH or access to their website to and from certain IP addresses. The downside of blocking a user based on their IP address was that if a user on a shared internet connection failed to login after ten times, other people using the same internet connection will also be locked out.

3 Statement of Objectives

The aim of the project was to create a small website with the addition of a CAPTCHA. The home page required the use of a CAPTCHA to be added to a login page. When you try to login, the username, password and the answer to the CAPTCHA would be validated against. Once you successfully logged in to the system you could leave feedback and then you are logged out. The website needed to include a help section, copyright section and an about section so the user knew enough about the system and how to use it. The style of the site should be based on the Intranet home page.

The login page needed to authenticate users with the use of a MySQL database in order to see who is a member and then lookup their username and password. All passwords should be encrypted using an algorithm such as MD5. Other methods of encryption that could also be used in the system include SHA1, DES and Triple DES. As the project was about authentication, rather than password security, either SHA1 or MD5 would be the best algorithms to use as phpMyAdmin allows the encryption of passwords with both of these methods.

The CAPTCHA had to be an image that would be displayed on the login screen. It needed the use of PHP sessions to synchronise the answer from the home page to the login authentication script.

The software had to be functional as staff and students were required to assist with a survey. The aim of the survey was to find out respondents opinions of the software, the CAPTCHA and how they would feel if this system replaced the current intranet pop up dialogue box.

In order to maximise the number of feedback responses a strategy was needed to attract the audience to the website. This was made possible with the use of the departmental forums, by word of mouth and via communications (such as email). The website would have to be accessible all day, everyday so that participants could help at a time they preferred.

To maximise on the number of feedback responses all the respondents were kept anonymous. In order to do this whilst, knowing the user account was by storing the user account into a database and the name and email address had to be emailed to the surveyors email address. The name and email part of the feedback was used to answer any question or personal opinions the respondent may have about the system.

To assists with any sort of SQL injection (that could possible wipe the database contents), the responses from the question were stored in a database whilst the name, email and comments were sent via email. To keep the system online the aim was to have two websites, one stored on the University server and another by a private hosting company. If for any problem one website was down, the other was available. A backup procedure was put in place to store the email content and keep a copy of the database offline.

With the information from the feedback, the aim is to gather the information from the survey and use it to prove whether the objectives of the CAPTCHA were met. If they were successful then a literature review will be written with our findings and prove to some organisations that their comments on the subject area would need to be closely looked at and possibly reviewed. We would like to show to industry that it is possible to create CAPTCHAs that are easy to read and that it is possible to create CAPTCHAs that do not need to have any of the text distorted.

4 Introduction and Motivation

There are three main reasons that drive this project, they include: the addition of a welcome screen for the Intranet, the vulnerability that has to be removed and proving to the industry that CAPTCHAs do not need to be overly complex to serve a purpose.

Whenever a user (staff or student) goes to the Intranet they are shown a dull pop up box asking for their username and password. This is good as the system works, but as a company the aim should be to advertise itself. With the use of a welcome page the user can see for themselves that they are at a website with a nice interface showing the University logo and the user knows that they are at a secure web page with the use of a security certificate. This would be publicised in the user's web browser with the use of a padlock symbol and the start of the URL (https).

4.1 A Problem

At this moment of time there is a vulnerability with the Intranet. This is a brute force vulnerability and was found when testing how many times you could attempt a login before the system locked you out. After attempting over 100 attempts the system would still let you carry on. After the 100 attempts were made a genuine user account and password were entered and we were able to login successfully.

To see if this was an issue with just one web browser the following web browsers in the test included: Firefox, Internet Explorer, Opera and Safari all of which were on Windows XP Professional. To check that the test was reliable, the same tests were carried out a Linux distribution (Fedora 7) with the web browsers Firefox, Konqueror and Sea Monkey. This set alarm bells that it must be true and when asking Phil Trew a couple of days later, he said "there is no limit".

With this information, the aim was two fold. The first showing to others that there is a problem with the current system and an exploit should not be available. With the ECE forums now moved onto the Intranet, we don't want to have them moved again do we? Upgrading them cost the University money and the students distress without the availability of a shared resource.

4.1.1 Convincing Others

The second factor is to prove to industry and the critics of CAPTCHAs that they are useful and content in the image is easy to read and does not need to be distorted to serve a purpose (Oleg 2007). By making the text clear to read and using a puzzle (that was a mathematical operation) there was a small challenge for the respondent. The target audience when designing the CAPTCHA was staff and students of the University. They should (in theory) have the educational ability to work out the answer of an operation whether it was subtraction, addition or multiplication.

The developer made this even easier for the respondent by allowing them to generate a new image if there was a problem with the one they saw. This was possible with the use of JavaScript where the respondent could click on the image to see a new one. Whether this was because the respondent could not read the text of one image or even where the math was difficult, there was the optional choice for the respondent.

If the respondent still found this difficult there was information of how to make this even easier for them. Explanations of how to use a calculator or a magnifying tool on a Windows and Linux platform were explained to the respondent in the help section of the website with the use of screenshots. The developer even added the ability to add user accounts where an individual could bypass the CAPTCHA if they required this for any disability reasons. Even where the CAPTCHA may have been exempt for that user, the number of login attempts was still enabled so that a brute force vulnerability was not possible.

5 Background Theory

CAPTCHA stands for "Completely Automated Public Turing test to tell Computers and Humans Apart". Alan Turing first invented the Turing Test back in 1950 and this was used to see whether a person could distinguish between a computer and a human. Three staff at Carnegie Melon University and another working for IBM (Wikipedia 2004) used the idea of the Turing Test and came up with the concept of the CAPTCHA in the year 2000. This was first publically used by Yahoo.

The developers of the CAPTCHA used the idea of the Turing Test to create an image with text, images and even questions to show to a person and see if they could work out the answer. Some people would say that this is the reverse Turing Test as it is focusing on a human answering the CAPTCHA question rather than a computer (Coates, Fateman and Baird 2001).

On the left is an example of a simple CAPTCHA from a website using JSP (Conner 2007). Not all CAPTCHA images are easy to read like this one. Further examples can be found in section 14.

The way a CAPTCHA works is by having a server generate an image (this can also be an animated gif) or even audio. The image CAPTCHA could be some text or even an image(s). In the case of an image containing text, you may see a word, random characters and symbols or even a question. Below are some examples:

- word: story
- character: f s u & 2 9
- question: 7 + 8 =

For audio CAPTCHAs, the above three can still apply, but they would be spoken out to you. In image CAPTCHAs, you may see one or more images or puzzles. Some examples include:

- Click the correct image where there is a football
- What colour is the sky?
- How many people are in the image?

The user is asked to enter the answer in a text box to submit or even click on the correct image (as if it is a submit button on a form) and the correct answer is compared with the answer submitted. If the answer submitted is correct then you can proceed. It's an authentication technique to hopefully tell the difference between a real person using the system rather than a computer (bot) trying to guess the answer.

CAPTCHAs are mostly found when registering for an email address, registering on a forum and even when posting a comment on website blogs (Claburn 2008).

6 Theory directly relevant to the Project

This project required the use of PHP, MySQL, HTML, CSS and mathematics. The use of web technologies allowed the creation of a website and the use of mathematics to work out the answer to the equation of the CAPTCHA.

In order to tackle the problem of a replacement Intranet page, we needed to look into the current problems with the system. Whilst logging in with the wrong account details we found that there is no limit to the number of login attempts. The aim was to have a lockout mechanism that blocked an attacker for five minutes if there was a suspicion of brute force activity and the implementation of a timer to reset this count after the time limit was over.

Whilst researching about CAPTCHAs and alternative solutions our findings showed that the main concern about them is that the designers make them hard to read by distorting the image (W3C 2005). The aim was to make the text clear to see with no distortion and the grid is easy to read for the user. The grid references in the image were displayed in multiple colours to assist people with colour blindness. One colour was used for the text in the grid and another was used for the grid lines to make navigation between numbers easier to the end user.

6.1 User Login

When a user attempted a login there are five possible stages to process their details to determine whether they would be able to enter the system successfully. To make this easier to understand, these processes are shown in Figure 6-1.

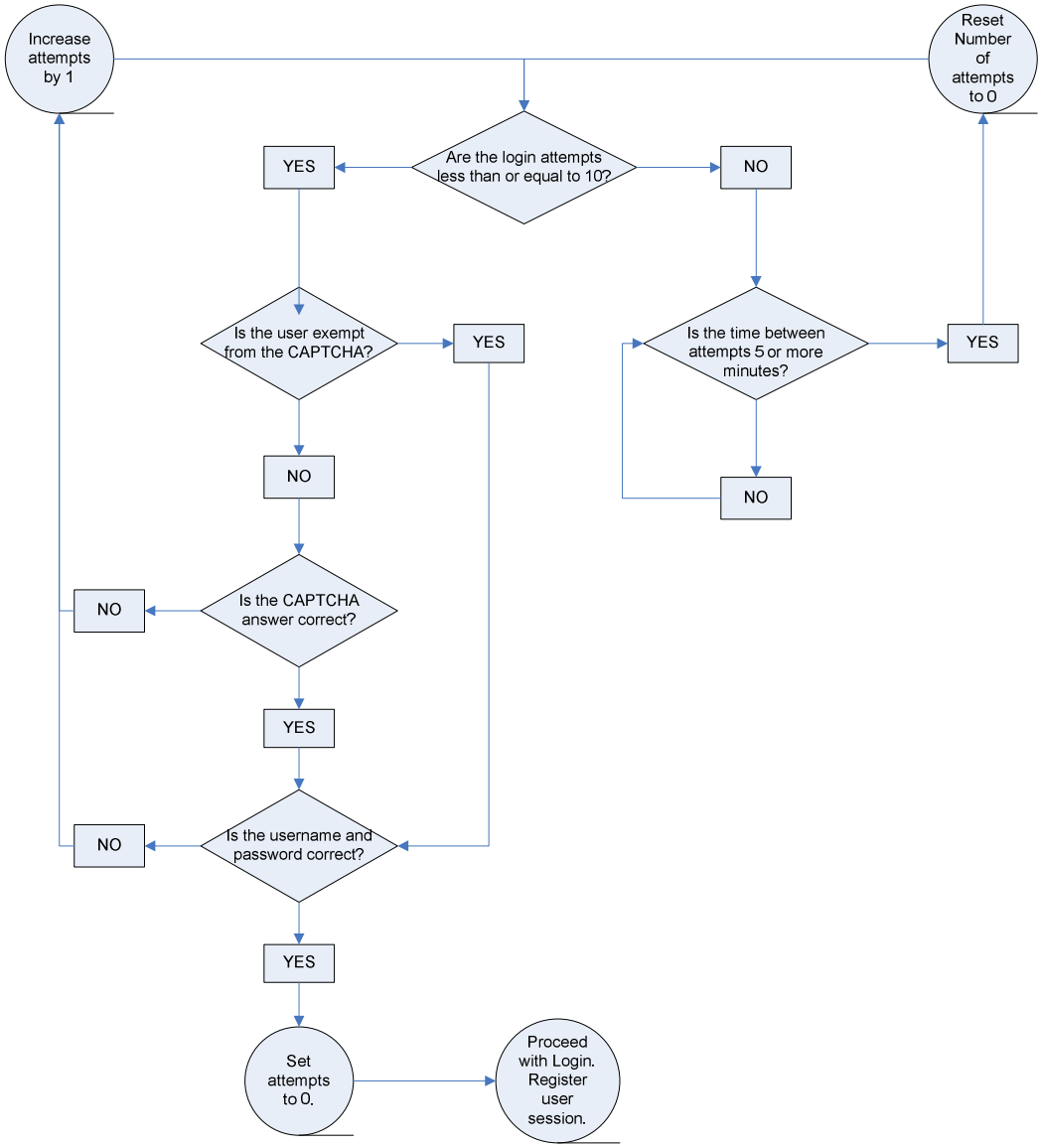


Figure 6-1 Steps involved to login to the system

When the user submitted their login details and the answer of the CAPTCHA, an initial query would look at the number of attempts a user has made before deciding whether to accept their login credentials or not. If they have made more than ten attempts, they will be locked out for five minutes. When the time limit had expired then the user's login details would be compared with the database.

6.1.1 Validation

The validation of the user's details went through a number of processes. The first was to lookup the username, password and whether the user is exempt from the CAPTCHA. If the user was exempt from the CAPTCHA then the answer of the CAPTCHA will not be checked,

however if they are not exempt, then the answer is compared. For the majority of users, if the answer is correct then the username and password is compared with the database. If the answer is incorrect, the number of attempts increments by one and the user would then see a failure message.

If the user passed the first step then their username and password would be compared against the system. This is done by looking at the database for one record that matched both username and password. A query was used to lookup the username and password, if only one record appeared this would be accepted and a session would be registered. If they did not match, the user would be shown an error message along the lines “incorrect username or password” and the number of attempts would increase by one.

In the event where more than one record appeared, an error message would appear. This should not be possible as all usernames are unique. New usernames could not match any existing ones on the database.

Once a user could login with one account, a check was made to see whether multiple users could login with only their own login credentials (username and password). Tests were used to check cross matched records (e.g. username one with password two). If the test was successful then the login page would display an error message. On the PHP side, sessions were needed to allow the user to view all pages that are viewable to that user’s group.

6.1.2 User Groups

During the implementation of the software, there were five user groups. They ranged from a basic account where a user could login and leave feedback to an admin account where the user could login to the admin section, creating and maintaining user accounts and even viewing the feedback from the survey. The user groups in the system are:

- Restricted User – can login and leave feedback data but cannot change their password.
- Users – can login, leave feedback and change their password.
- Staff – can login, leave feedback, change their password and view extra content than users.
- Supervisor – has all staff privileges and can view the admin section of the site.
- Admin – have entire system access and change anyones password.

7 Design Considerations and Implementation Options

This section looks back at the preliminary study for the design ideas and mixing those with the Intranet website styles to produce a website that the user would understand how to use.

7.1 The Current Theme

The design of the website in our opinion had to be based on the Intranet theme. This was because a welcome page should take the current website formatting. Changing the theme would only have confused the users and they may have been reluctant using the website.

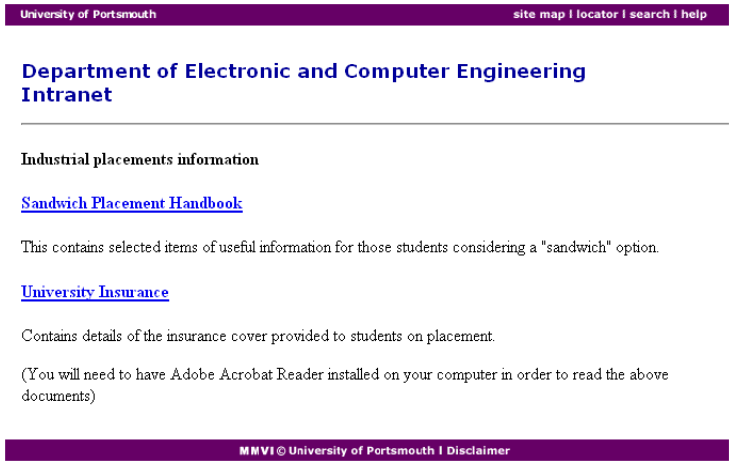


Figure 7-1 Current web page on the Intranet

The theme at the time of the website build included a purple header and footer with white text. The body of the page was white. The header text was blue and the text on the screen was in black as seen in Figure 7-1 .

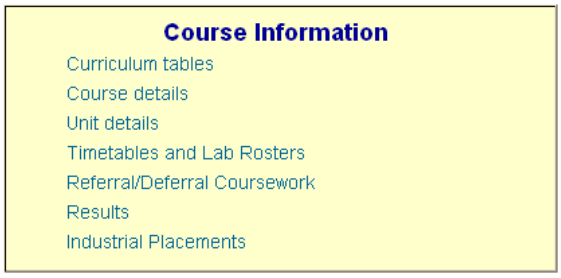


Figure 7-2 Current menu system on the Intranet

Apart from the theme throughout the website, the navigation menu originally had taken the same format as the Intranet one. The navigation menu on the Intranet uses green text and a blue header as seen in Figure 7-2.

The next problem was how to design the login box. Looking around the Intranet we needed an idea of creating our login box. This could not just be white and to our fortune we found one on the projects website login page. It used the yellow background and blue text. There was a purple login button that was made using an image.

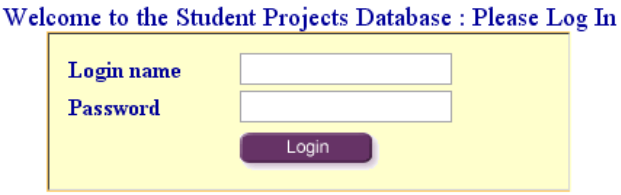


Figure 7-3 Login box on project website

7.2 The New Theme

Without using any of the images on the website (with the exception of the University logo) we managed to replace the images of text in the header, footer and the login button, by just using text.

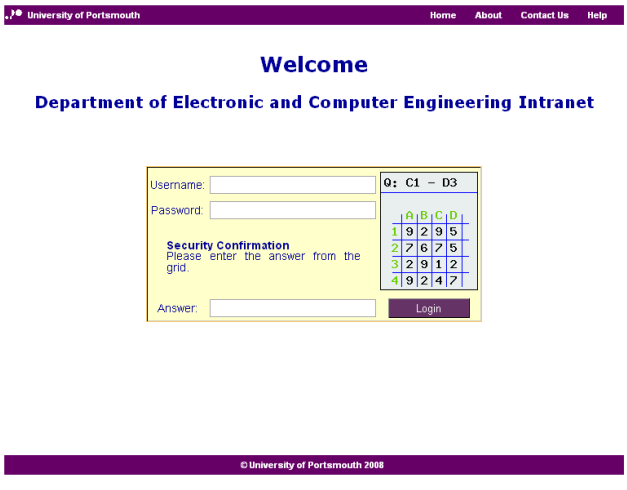


Figure 7-4 Website home page

After many of hours of work trying to replicate the style of the Intranet with a few improvements (as we like to call them) we had our home page. The page can be seen in Figure 7-4 and comparing the two between this page and the previous, there is not that much difference.

All the text in the header and footer is only text (not images like the Intranet) and this would save lots of bandwidth over the course of a year. In order to make this work in the most commonly used web browsers a script was used to detect each web browser and the second style sheet would override the first for that web browser. This was very handy as a bug in one browser did not spoil this for another.

In Figure 7-5 are the two menu styles used. Originally the first one was used as this was for the user section of the website. This was styled on the current Intranet style (as seen in section **Error! Reference source not found.**) and by keeping this simple the user was not over complicated with information.

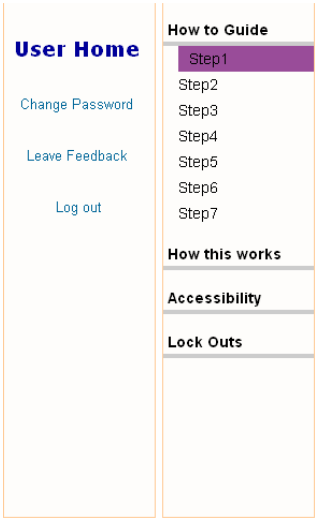


Figure 7-5 Menu navigation

The second image was used to help everyone navigate through the help section of the website. When a user viewed a certain page on the help section the navigation page of that link would highlight. The developer wanted to make this as easy as possible informing the user which step they were on in the help guide. By using PHP this was made possible.

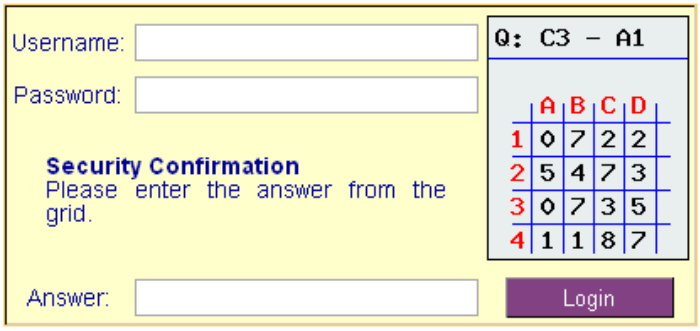


Figure 7-6 Login box

The final comparison is the login box. The styling used the same idea of the project database login page, however this had one difference. Instead of using an image for the login button, CSS was used to make the login box wider and change the colour of the submit button.

The next comparison is with the image above and the original idea in the preliminary study (shown in Figure 7-7). If you were to compare the in Figure 7-6 (above) with the one in Figure 7-7 (right) you should notice that the question is located in two different places. The question in the actual software has the question above the grid whereas in the preliminary study this was away from the grid.

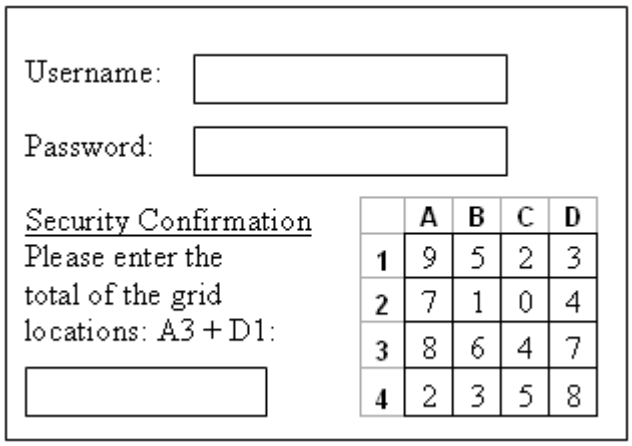


Figure 7-7 Preliminary study original login box design

The problem when trying to implement the CAPTCHA in the same way as the original design was that the values on the web page would not echo out the same grid reference or answer. By modifying the system session variables could have been used but how PHP uses cookies, the user could find out the answer without looking at the grid, so this was avoided.

In order to make the CAPTCHA easy to read, grid lines were used (as seen in the original design) and the use of different colours for grid references were also used. The reason for using different colours was to make the grid references easy to read for multiple people. If one colour was hard to read, by clicking the image another colour was displayed.

8 Preferred Solution

There were a number of issues that needed to be planned for thoroughly. They included: the number of user accounts to use, account passwords, the survey start date and the CAPTCHA.

8.1 User Accounts

When the system was developed, there were five user groups created. They include: restricted user, user, staff, supervisor and admin. By implementing these types of accounts, the developer could test the website for integrity of these user permissions and prevent standard users from interacting with the admin side of the site.

The only account not used during public testing of the system was the staff account. The idea was to allow certain members of staff to use their own account (with the possibility of bypassing the CAPTCHA) so that some members of staff knew the system better and they could see the system from a different perspective. Later on this idea was rejected by the developer as it meant that not all staff would see the system equally. This would have given different sets of feedback and by providing different staff with various permissions this would add prejudice amongst the staff community (possibly resulting in a smaller number of feedback responses).

When the testing was carried out, most of the respondents used a restricted account. This saved a lot of time since creating hundreds of accounts would have been time consuming. The usernames “user” and “staff” were both generic accounts and had restricted privileges so a large number of respondents could login to the system without changing any of the account details (such as the account password). This allowed the developer to separate student and staff feedback responses for the analysis of the feedback results.

The supervisor account was used by the project supervisor. This allowed him to login to both the user website and the admin website. The difference for the supervisor was that when he logged in to the admin website, he could not make any changes to user accounts. The account was needed to allow the supervisor to monitor the developers work and state possible improvements like what to add or remove.

8.1.1 Lockouts and Timer resets

Once login and logouts worked, the CAPTCHA was merged with the login page. The test was to authenticate both data types and this required the use of sessions to authenticate the CAPTCHA by keeping the answer from the home page to the authentication script. Once this was working, more useful features were added (including login timeouts and improving on design features).

In order to secure the system down, a user needed to be locked out of the system for a small period of time. As an administrator would not be unlocking user accounts all day, a timer was implemented along side so that a user was not prevented access to the system with denial of service. The problem for the developer was how to achieve this goal by either locking the user account that could be in the process of a brute force attack or by blocking the user that is attempting a possible attack by using their IP address.

The idea of using the IP address was used because in industry, companied block access to their website and even emails from certain IP address ranges (Rickert 2008). By using the IP address, this would help against denial of service as the account that may be brute forced would not be locked, but the attacker would be. If the system was implanted in the Intranet the other benefit here is that by leaving the user account unlocked, the user could still use other services like SSH to access their files. This was particularly an issue for the electronic submission of coursework.

There was a problem that the developer was aware of and this was the problem with shared internet connections. If one of four students were to be locked out, then the other three would not be able to login. A timer was used that lasted five minutes and as long as no attempt was made in that time limit, the user could attempt another ten attempts. The number of attempts was also reset after a successful login.

For this project, the number of attempts was set to ten so that the developer could maximise the number of successful logins without having a number set so high that it is now worth implementing. All the information was stored in a database so that the administrator could monitor the attempts and where necessary add suspecting IP addressed into a block list.

8.2 The Survey

The developer did not want to use the individual accounts as this specifically identified a respondent when analysing the feedback. The idea of the feedback in the project was to give the respondent anonymity from the system unless they were willing to put their name down publicise this fact. By keeping the majority of respondents anonymous this would help increase the number of feedback responses and encourage open and honest feedback from all the individuals, whether it is good or bad (Bergin 2008).

8.2.1 Pre Survey Feedback

Before the system went out to the public a group of students were given their own account to login and leave feedback. These students were willing to help the developer test the system before bugs were found by staff and students from the various year groups. Five students were chosen due to their knowledge of web technologies and the various hardware / software specifications of the computers they used. Between the group three web browsers were used and three operating systems. All of these five students had "user" permissions and could change their own password if they wanted.

The feedback from these people and earlier (from the sample of fourteen people) was mainly about the website help section and the CAPTCHA. The help section was "too wordy" (quote by Richard Priddy) and the CAPTCHA grid references blended with the numbers, making them "harder to find" (quote by Duncan Heffer). The final comment was that the numbers in the grid "were not random" (quote from Samuel Groves). All of these points were very useful.

The developer tried to used this information to improve on as much of this as possible. The biggest concern was the CAPTCHA and the grid references. To make the grid references clearer to see a number of colours were used to see which ones to use. After experimenting, there were reasons why all of them should be used and why all of them should not be used. As this was targeted for a large audience the choice of colour was vital and the decision was to store all the colours in an Array and use them randomly on each CAPTCHA reload.

The colours used were: red, blue, purple, orange and two different shades of green. When checking this with a fellow student who is colour blind, they confirmed that the colours were easy to read. The reason for some of the colours was down to the brightness of the users monitor. This had a big impact on all the colours making some colours easier to see than others.

8.3 Clickable CAPTCHA

To make the login easier for the user, the CAPTCHA image could be re-generated by clicking on the image. This feature was added so that if the user could not see the image clear enough or found the question difficult, then they could generate a new image to make it easier for them. The other reason for this was because the idea is not to distress a user who has already entered their username and password. A situation where we found this particularly annoying can be seen on another student project website called “incredi” (as seen in section 14.1) where the registration data with eight data entries are reset when the CAPTCHA is guessed incorrectly.

In order to generate a new CAPTCHA image when the image has been clicked, JavaScript was needed. Apart from this feature, JavaScript is not used anywhere else on the website. JavaScript allowed the update of the image without refreshing the web page so that the user could view as many different images possible before they saw one that was easy to read and had a question that was easy enough for them to work out the answer.

9 Project Plan

Initially the project was scheduled to start off from the 31st March and the finish date was on the 6th June 2008. With the Easter holiday, the project started even earlier by using the time most to create the software.

All the tasks needed to be divided up to match their weightings and over the course of the project the Gantt chart was compared to ensure that the project stayed on track. The original Gantt chart is shown in Figure 9-1.

9.1 Original Gantt Chart

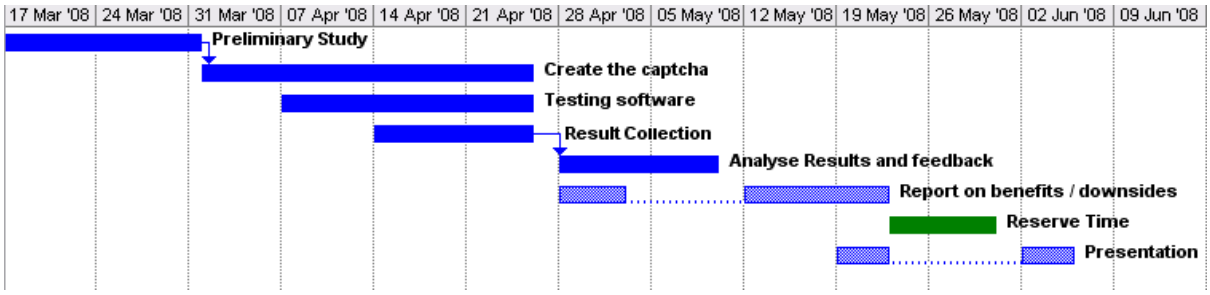


Figure 9-1 Original Gantt chart of project schedule

This Gantt chart was not as clear as it should have been due to some of the tasks being missed out of the original plan. Later on we found that some extra tasks that were required in the project. After some research (Groves, Cialdini and Couper 1992), more time was needed for the survey. This included time to come up with questions, designing a feedback form and using a small group of students for testing the feedback form and their comments, before any of the respondents (staff and students) found any problems with the software.

9.2 Project Tasks

By the end of the project there were seventeen tasks. They were divided up by: their weighting, how important the task was and a realistic time needed for completion of a specific task. All the tasks are shown in Table 9-1. All of the start weeks in the table use the University semester week as a guide. In the table 0.2 weeks represents one day.

Tasks	Start week	Total weeks	Due in date
Preliminary Study	**3**	**3**	**31st March 2008**
Create website	5	2	
Create the CAPTCHA	5	1	
Testing software	5	3	
Feedback research	5	5	
Questions	5	2	
Feedback Form	6	2	
Hypothesis	8	1	
Sample Survey	8	1	
Sample test users	8	1	
Public Survey	9	4	
Analyse of Results	8	5	
Website Help section	8	1	
Presentation Practice (for students)	**8**	**0.2**	**7th May 2008**
Presentation	**8**	**3**	**13th May 2008**
Final Report	9	4	
Reserve Time	14	1	
HAND IN ALL WORK			**4th June 2008**
Project Day	**15**	**0.4**	**6th June 2008**

Table 9-1 Tasks in the project

A week before the presentations started, the student working on this project organised a practice session for nine students to practice their presentations and improve them for the following week. In order for all the students to concentrate, the plan was for the students to be split up into two groups. With no session longer that 75 minutes, this would help ensure that all students paid attention to other students presentation whilst the presentations were underway.

In order to use two lecture theatres, the student booked two rooms through the proper University channels with a two hour booking for each room. This was to prepare for any delays on the day such as a late student, students over running or evening faulty equipment (like a projector).

Although it was practice, a few rules were added to assists all students. Not all students knew how long their presentation would be so we decided to have a time limit of fifteen minutes.

This way any student that did not plan their presentation was made to re-think their entire presentation. Luckily this was not the case with any of the students. The most time any presentation had taken was twelve minutes and the shortest was eight and a half minutes. By annoying Dr Hewitt we managed to get a copy of the marking scheme (this was new this year) so that each student had feedback on the points and would know where to improve.

When speaking to all the students, the feedback received was very useful. Everyone managed to improve their presentation for the day and over half were informed of how good their presentation was from a member of staff or fellow students.

The survey was a major part of the report and needed the most amount of time. By completing the software quicker than planed, this allowed the surveyor to start the feedback process earlier. With the extra time he managed to look at improving the quality of the questions and plan how to use the finding better. Between three and four weeks were of a research was spent reading various articles on feedback preparation, result analysis and data collection.

As the project moved on, a number of weekends were used as there were vital deadlines throughout to keep the project on schedule. The critical path required the bare minimum of: working software, a feedback form and questions, time for a public survey (for gathering results from the public), analysis of results, a presentation and a report. All of the tasks in between helped improve the strength of the questions, produce a sample (with a spread of students from various year groups) and have a larger amount of time to analyse the results (quantitatively and qualitatively).

9.2.1 The New Schedule

Throughout the project a revised Gantt chart was used to track the progress of the work and stick to the project deadline. Many projects in industry run over their allotted time and for this reason, the aim (from the start of the project) was to keep to the original plan (Gantt chart) as much as possible by having a one week window at the end of the project for any delays. This provided a nice cushion to ease the stress levels as the project progressed. Hope forbid delays would not occur, but planning a safety net was a must as presentation and project day would not be moved. The final Gantt chart can be seen in Figure 9-2.

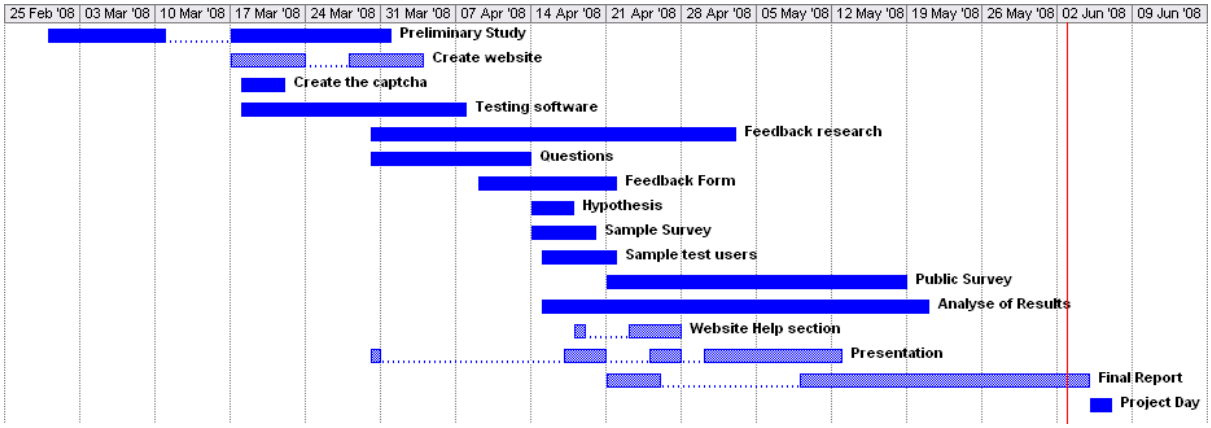


Figure 9-2 Project Gantt chart (final version)

The software was created faster than originally planned. This was because the developer practiced PHP on his own website months before the project started. This became a major advantage as the project progressed.

Later on the feedback process took over half the project allocated time. Time was needed to produce questions to ask in the public survey. Over the course of two weeks, there were multiple revisions for this. For a web designer this was out of the normal study area. At times this felt like a laborious task as it involved large amounts of time coming up with ideas and little to show (when comparing this to the website).

Once the questions were chosen, the end result was a nice website and feedback form system that would take a respondents question choices and would be emailed to the project student and the answers were added to a MySQL database. The respondent's answers were in two places in case some of the data was lost (by an SQL injection or another type of data loss) and all the feedback was backed up every couple of days.

10 Development and Test Program

This project required a number of tasks to be carried out. Initially software was created and it needed tested in depth. This required the testing of form data such as the login page, the feedback form and checking that empty fields were validated where necessary. The next stage was checking that the system displayed well across various operating system platforms and a number of website browsers.

Later on the CAPTCHA had to be readable and a comparison was needed to verify the answer between the answer of the image the seen by the user and the answer they submit. Finally the feedback form added further work. Apart from checking that all the inputs were valid, preventing mass posting from a user and checking the consistency of answers were also needed.

10.1 Display Issues

Before the software could go out to market, it needed testing in depth. The problem here is that respondents would use various operating systems and website browsers. To tackle this problem the website was tested in six different web browsers between three operating systems. This way each user could view the website with minimal display problems.

The testing was split into two parts: one for the software and one for the sample feedback. The first test required website display to work in Windows, MAC and Linux. As far as web browsers are concerned, the following were used: IE6, IE7, Firefox 2, Firefox 3, Opera, 9.26, Opera 9.50b, Safari 3, Konqueror 3.5.4 and Sea Monkey 1.13. This was going on the fact that the four most used web browsers are: Firefox, IE, Opera and Safari (W3Schools 2008) and the other two browsers Konqueror and Sea Monkey are used on the Linux OS at University.

With the various system configurations the respondents use, the feedback form needed testing in most, if not all the browsers in the above list. The tests were carried out before the form went out to the public. The following tests were carried out:

- **Home page displays the same across browsers:** Apart from IE6 the width adjusts between 770 pixels and 1020 pixels.
- **Drop downs display all answers fully:** IE6 would not display answers of fixed select boxes. This was changed, but the form looked slightly odd as a result.

10.2 Functionality

The website functionality was equally important to how it displayed across the web browsers and operating systems (see section 10.1). This is because a working website is no good if you cannot see it properly. The following tests were carried out:

- **Feedback submits to database and email address:** Works fine in all the browsers stated in this section.

- **Empty answer to a question:** This validated where one or more questions not answered.
- **Keeping answers when validating:** When the user clicked back their answers were cleared. This was a result of using sessions. The fix here was to remove the session, but use it on the send feedback script.

On the home page, the login had to be tested. Validation needed to be used to test that a missing entry would not allow a user to login, a username of a user and a password of another would not be accepted. Along with this the CAPTCHA needed checking for the correct answer.

Initially the CAPTCHA answer was printed in the image when testing the CAPTCHA on it's on. The answer was compared against the question and the answer that was shown was entered and submitted. This worked by synchronising the answer from the CAPTCHA to the login screen with the use of sessions. The only downside to this was that the username and password would be reset so the user had to re-enter the username on each failure, however this is how the current Intranet login system works (minus the use of sessions).

10.3 CAPTCHA

The CAPTCHA was a PHP file that displayed as a PNG file. A user would not notice the difference unless they were to look at the source code. By using a PHP file rather than an image file, the developer could manipulate the data between the website pages with the use of sessions.

Unlike the website that had some issues across browsers, the CAPTCHA displayed the same across platforms and website browsers. The only worry after this was that as long as the login page displayed correctly, the CAPTCHA has no problems. There was the option to the user that they could click on the image to generate a new CAPTCHA image, this relied on the use of JavaScript. If the respondent did not enable JavaScript on this one page, they could not use this useful feature.

Initially when the CAPTCHA was first created, it was tested on a single web page. The aim was to validate the CAPTCHA on its own. When it first worked, the developer set the

operation to subtraction, then addition, then multiplication and finally a random operation. The answer was printed in the image and the entered in the answer text box. From the four tests, the first 20 were entered correctly and the next 5 were entered incorrectly. At the end of the four tests, there was a 100% success rate.

When the CAPTCHA was added to the authentication script, it required the addition of the session to be copied across and this in turn allowed the account credentials along with the answer to the CAPTCHA to be verified. The respondent needed some information to let them know if they entered the correct details or not. After consideration and feedback from students and staff, the plan was to use a generic error message as information about valid usernames needed to be kept secret from the users.

10.4 Feedback Form

In this project the feedback form became more important than the actual login. Where the login page could have had the odd bug, the feedback form had to work perfectly. The form had to function properly, the questions and answers had to make sense and the spelling had to be correct.

The survey went out to the staff and students with no supervision from the surveyor. In order to maximise the feedback and the honesty of answers, error messages had to be meaningful and the anonymity was key to all of this. By not knowing which student left which feedback, this helped increase the number of feedback responses (CVent 2008).

The feedback questions had taken over two weeks to produce as a lot of research was needed in order to find out what information to gather to answers some of the questions from past articles on the subject. The developer could find out the benefits / downsides of the system, the implications of the system and the respondents views from the responses to the questions along with the comments section of the feedback form.

The form needed to be well designed so that everyone understood its purpose and knew how to use it. This was no easy task. After some research, the developer decided to add a couple of sentences to inform the respondent of how short this survey was and how long it would take (Groves et al.).

The form design requirements meant that the questions had to be clearly readable. To help distinguish between the questions, each row had a different background colour. The answers varied between questions, so the use of radio buttons was avoided. To help display all the answers, select boxes were used. Before the survey went out to the public a flaw in Internet Explorer 6 meant that two of the answer drop downs needed to be resized as the answers to questions five and seven were not fully readable. This meant that the alignment was no longer consistent, making the look of the form slightly off but improving on the form functionality.

The next problem was that a respondent was not led to a specific answer. In the form, the first answer for each question was the default option “please select”. The only way to test this was to conduct a sample survey. From the sample the respondent’s answers were varied and as a result of this, there was a belief that the survey could proceed based on the results of the fourteen respondents.

Part of the public survey brought some problems. This was due to the nature of students on the department. At first all the answers were going to be emailed to avoid the issue of SQL injection. The issue with having the answers emailed was that the results would have to be typed up and verified for inconsistencies. This brought the issue where the correct data may be modified, making the feedback questionable and this process of data entry would have been time consuming.

The workaround was to use email submission and database insertion. This allowed for a backup strategy. The database can be backed up and each email could be printed as a PDF. If there was a loss of data at either end, hopefully the other system would still be available. In the event that the email account went down, another one would have been used. In the unfortunate incident of the database going down, a duplicate website was available on another web server outside the University.

The feedback form was vulnerable to SQL injections. Students love to play with other systems and break them. To counter attack this, only the answers of the eight questions were inserted from the form. The answers were submitted as numbers between zero and five (depending on the number of answers available) to keep the size of the database down. The answers were forced to be integers (using the "int" function in PHP when requesting the form

inputs) so that any sort of bot attack would result in an answer becoming zero. Just like a missed question, any answer that is zero would be invalid and the user would see an error message. By making sure the answers were numbers, this made an SQL attack on this form, "virtually" impossible.

To deal with spam from the form input, there were a number of measures in place. The first used hidden form text box entries (Batchelder 2008). Two fields named "address" and "post code" were added after email. If any of the text boxes had any data, an error message would appear. This would only occur with a scripted attack as a genuine user would only enter data into text boxes that they see. The text boxes were hidden in two ways, one using html "type=hidden" for the text box and the other using "visibility: hidden" in the main style sheet. Unless both of these methods were picked up by an attacker, one of them should catch the attacker out.

If a user left a question, they would also see a message asking them to go back asking for them to answer one or more missing answers. One problem PHP has is that with the use of sessions, when the user would click back, all their answers would be reset. This "bug" was not diagnosed quickly and only when comparing this with the contact form was the fix found. Sessions were needed so that mass feedback could not be left by a user. To fix this problem, the feedback form no longer used sessions, but the script that the form action is set to go to was. This meant that a user that was not logged in out just be redirected to the login screen.

Once all the validation was successful and the respondent submitted their answer would their feedback be inserted into the database and emailed out. The comments from each feedback were not added to the database, but were emailed to the surveyor to help against SQL injections. At this stage the second measure to prevent a spam of feedback responses was to end the respondent's login session so that they were redirected to the home page, making this easier for the respondent as some respondents would just close the browser without bothering to logout.

The third method used to counter attack spammed responses was the use of IP address, browser details along with the time and date the feedback was posted. PHP had functions to grab this information and this was set to variables to grab this information. By analysing all the feedback, mass posting could be found and multiple entries from an IP address could be

removed at a later date. This was useful on two occasions where a respondent posted the same feedback but posted different comments and when another respondent tried to SQL inject the system.

11 Hypothesis

The aim of the project was to find out how the staff and students of the ECE department would feel if the current Intranet system was replaced with a new authentication system using a CAPTCHA. The questions in the survey are based on finding out specifically how the respondent feels using a new system, whether they care about the security of the Intranet and what sort of delay this added to the time it takes to login to the system.

CAPTCHAs have been around since the year 2000 (Ahn, Blum and Langford 2004) and as a technology department, many of the respondents should have seen the use of them somewhere on the internet, whether it's a blog, an email sign up or a forum sign up (e.g. ECE forums). With this in mind we believe that the majority of respondents would know about them.

The overall survey results should be mixed across the staff and student population. If the results are split using the answer from question one, we believe that the audience that have not used CAPTCHAs as much would show the most positive answers as they have not had as much experience with them to show the frustration of some of the difficult CAPTCHAs that are harder to read (Zarar 2008). This would also annoy students that find the maths difficult as it would increase the login delay between five and twenty seconds. However if this study was repeated a year or two later with the system implemented this would be interesting to see how this changed.

Splitting the groups between students and staff would bring another interesting result (hopefully). In theory staff should be accessing the Intranet all the time. Based on higher usage of the Intranet from the staff, there should be negative answers from staff compared to students. Not just based on this, staff see multiple cycles of students come to University who only use the system over a few years and would just put up with this over the course of their University education. With this in mind, staff would be annoyed with the extra delay for remote connections to the Intranet.

With the questions in the survey we believe that the respondents would feel some frustration after a period of time. A percentage of respondents save their username and password so that they can login to the system with ease. The addition of the CAPTCHA would mean that there is a delay reading the text in the image and to work out the answer of the question every time.

Finally, the biggest question was the last one in the survey, about Intranet usage. The belief here is that one fifth of the audience would use it less. This would be some worry to us as it proves that the system is not one worth implementing, but this is a research project and the feedback information is the achievement, not the software. If the estimation was correct, then a victory is where one or more of the respondents would prefer to use the Intranet even more with the new system implemented.

The addition of lockouts is a nice feature that should definitely be on the current system. We believe that the security controller would like the new system implemented as this would help lock down the system.

11.1 Question Choices

Question 1: Before today, I knew what a CAPTCHA is?

This question was used to find out whether the respondent had any background knowledge of what CAPTCHAs are. Some of the analysis would include a comparison of all the feedback based on the answers of this question to see if this reflected in the answer of other questions in the survey. If this did show any major difference, a comparison against other studies may be looked into using some of the articles read in this area such as W3C's view on CAPTCHAs and why not to use them.

Question 2: I found the numbers in the grid easy to read?

Whilst researching various CAPTCHAs and reading reports on them, there is a common issue about how difficult a CAPTCHA is to read (Edwards 2008). When designing the CAPTCHA the aim was to make this clear to read whilst having a purpose of assisting the University Intranet.

CAPTCHAs used by Hotmail, Yahoo and Google make their CAPTCHAs difficult to read by a computer by distorting the text in the image, making it harder for people to read (Atwood 2006).

Question 3: I found the equations easy to solve?

Subtraction, addition and multiplication are mathematical operations that University students should be expected to solve without major difficulty. Without presuming this in the analysis, this question was added to find this out this information as fact and not merely guessing the answer later in the report.

Question 4: Did you need a calculator to work out the answer?

The addition to the previous question is to find out if the respondent needed a calculator to assist with the calculation. If a large percentage of respondents used a calculator to answer the question, the presumption of simple mathematic operation was bad idea. If any future work is carried out, a consideration of dropping multiplication would have been made.

Question 5: Before you pressed login, what was the most time consuming task?

With the addition of the CAPTCHA there is an extra process used to verify a human or computer login. For a human there is going to be an additional delay for a user to login. This question is used to find out whether the CAPTCHA added the most time to login or not. One of the downsides to this question is that most, if not all of the respondents would only use the system once.

Over a period of three to six months the time should be minimal for most and many may not notice the difference in the time taken. If this project were to continue a survey of how long it would take just to answer the CAPTCHA would take place. Comparing this to the amount of time needed to login would be an interesting test.

If a large percentage of respondents were using a calculator or a magnifying glass, the design of the CAPTCHA would change, if the project continued past the ten weeks and the mathematical equations would be simpler.

Question 6: What would have helped you answer the CAPTCHA question quicker?

The previous question looked at what the possible delays were when logging in. This question was added to find out how the respondents experience could have improved.

Adding another data type will add to the login time. What would be interesting is to see what changes would speed up the time for the respondent to enter the correct answer in the answer box. The aim here is to find out a solution to the please the respondent majority if this had of been made commercially.

Q7: As a student, which of the following would you like most?

There are so many services provided by the University that the students are aware of, use daily or do not see the point of. Here the aim was to find out if the respondent cares about the security of the Intranet or whether there are other services they feel are more important to them. Would the security of the Intranet benefit them most, if not what will?

Q8: If the CAPTCHA was implemented on the ECE Intranet, would your level of usage:

This final question is to sum of the respondents experience. The aim here is to find out how the respondent would feel if they were made to use this system on a daily basis. This would be a good indicator on whether the user clearly dislikes using the CAPTCHA or not.

11.2 Result Predictions

Before carrying out the survey, predictions were made for each question. The predictions include weighted answers for both the student and staff population. Shown below are the final set of questions and what the predicted results may reveal.

Question 1: Before today, I knew what a CAPTCHA is?

	Yes	**No**
Students	60%	40%
Staff	70%	30%

University members of staff are established members of the public and the expectation is for them to know more about internet security as knowledge comes with age and experience over time.

Students would not know as much as they are still learning everything they learn in lectures, the forum and from peers. If the students were asked this question face to face, more students would own up or prove they don't knowing what a CAPTCHA is from incorrectly guessing. In the survey no one saw what answer they put so the expectation here is that 10-20% would lie in the survey.

Question 2: I found the numbers in the grid easy to read?

SA = Strongly Agree, A = Agree, N = Neither, D = Disagree, SD = Strongly Disagree

	SA	A	N	D	SD
Students	20%	40%	20%	15%	5%
Staff	15%	35%	20%	20%	10%

By using different colour text for the grid references and lines to draw the grid, this should have helped the respondent follow the grid and work out the answer. With this addition the prediction here is that half the participants should find this CAPTCHA easy to read. Out of the staff and students, the most positive answers should come from the students, since one third of the staff members wear glasses.

Question 3: I found the equations easy to solve?

	SA	A	N	D	SD
Students	20%	35%	30%	10%	5%
Staff	25%	40%	20%	10%	5%

Students on the department dislike the use of mathematics in lectures and even more so when it comes to examinations. University staff are confident with the use of mathematics and can perform complicated equations like Binary and Hex, so multiplication should be a stroll in the park.

Question 4: Did you need a calculator to work out the answer?

	Yes	No
Students	20%	80%
Staff	20%	80%

This question is used to determine how difficult the equations are and whether they should be changed in future development. Respondents who found the math difficult to answer may lie. The downside with this question is that it is focused on the respondents who answer was that they found the equations difficult (either disagree or strongly disagree) in the previous question.

Staff would generally own up to using a calculator where students would not. The end result of this question should reveal similar results for both groups.

Question 5: Before you pressed login, what was the most time consuming task?

	Calculation	Magnifier	Reading Text	Username Data
Students	25%	10%	45%	20%
Staff	30%	15%	40%	15%

Here either the mathematics or reading the CAPTCHA should take the longest amount of time for everyone taking part for the first time. Reading the grid should take between 5-10 seconds, however after multiple trials we would expect the time to decrease and the grid to become second nature.

Question 6: What would have helped you answer the CAPTCHA question quicker?

	Simpler Equations	Nothing	Larger Text	Audio
Students	40%	10%	30%	20%
Staff	35%	15%	35%	15%

Audio would add more time to the login process so we would expect a lower percentage on this. We would expect to see a higher percentage between the answers simpler equations and larger text. This would help make the CAPTCHA easier to read and answer the question

quicker. The end result will be close near the end and it will be interesting to see which one comes up the highest.

Q7: As a student, which of the following would you like most?

	Using CAPTCHA	Printing	PC Facilities	Library
Students	10%	25%	35%	30%
Staff	20%	20%	30%	30%

This question is one of the most interesting in the survey because there are no clues to what students and staff would choose in the above table. Putting the CAPTCHA side by side with three popular options shows that the emphasis was not to push the CAPTCHA system out from the wording of the question or by the answers used.

Not including the option of the CAPTCHA, the other three selections would benefit the student. A modest student may favour the developer and choose the CAPTCHA. Knowing what the student wants to assists with their degree should be helpful to the department in the next couple of years.

Q8: If the CAPTCHA was implemented on the ECE Intranet, would your level of usage:

	Increase	Stay Level	Decrease
Students	10%	70%	20%
Staff	10%	70%	20%

This question is used to determine whether the respondents usage of the system would change based on the implementation of a new system. Many students and staff use the Intranet on a daily basis and the expectation for this question is that the usage would stay about the same.
If the respondents dislike the CAPTCHA then this will reflect here with a high percentage selecting decrease.

12 Feedback

At the start of the feedback process a sample of students and a member of staff from the ECE department were needed to test the form and find out any problems with the wording of the questions and answers or even if the form data would not submit correctly.

12.1 Sample

The test sample involved thirteen students and a member of staff and students were divided to represent their year group. In order to represent the year group successfully a ratio was needed to determine how many students to sample. Unfortunately this information was not available at the time and a request was made as the administration office would not disclose that information. The project had to move on without this information.

The best way to work out the total students was to use an existing resource. This existing resource was the departmental forums. On the forums there is a total for the number of members signed up. On the forums it read at the time "We have **649** registered members" (University of Portsmouth 2008) and this consisted of both staff and students on the department. Based on this number the total was rounded up as not all students would be on there and the estimated number used was 700 students. With this in mind, the thirteen students would equate to 13 out of 700 students and this produced a two percent ratio.

The next part involved working out the amount of students to ask per year group. Still not knowing the total students in year the estimation made was that there are fewer students in each year as they progressed in their degree. The estimations are shown in Table 12-1.

Year Group	Total Number or people
Year 1	6
Year 2	4
Year 3	2
Year 4	1
Lecturer	1

Table 12-1 Sample number or students and staff

The total number of respondents in the sample was fourteen people. To calculate the ratio of students between year groups, the estimation made was that 60% of students went from the first year to the second year. From the second year the estimate was that only half of the students went into the final year. To calculate the number of masters students, the estimation was that there are less than half the year three students studying a masters degree, so only one masters students was in the sample.

The only piece of information that was available was on the ECE department notice board and the University website. All the lecturers were on the board and the two technicians. This totalled up to 32 members of staff that could be used in the survey at a later date. To add to the sample, only one member of staff was used as this equated a higher ratio than the student's sample.

In order to select a member of staff to represent the team, multiple criterions had to be considered. The lecturer would need to have a good level of knowledge in the following: have a concept of CAPTCHAs, programming, web technologies and distributed web systems as well as willing to participate in the sample survey. The staff respondent was helpful as they pointed out a few ideas and found an issue with the feedback form as one of the questions was focused on students. This was modified before the survey went public.

All the estimations may have been totally inaccurate or somehow they were actually close to the actual numbers. With the lack of information available at the time, the project could not be put on hold for a fortnight to find out this information and putting pressure on others would not have find the information any earlier. The consolation is that this did not affect how the public survey was carried out.

The sampling of this small group had taken one week to carry out this is due to the time acquiring the target audience (see Table 12-1). Just picking fourteen people would not have helped the representation of each year group. It was vital to choose the correct people as the interviewer needed to know whether or not the feedback form was at a standard that the respondent could clearly understand. On top of this the software needed to work without any bugs by checking that the submitted input was validated, entered into a MySQL database and the answers and comments were sent via email as a backup precaution.

After the sample was complete, there were some useful comments and one of the questions had a flaw. The question was focused towards students and need to be reworded or removed as it would not produce any useful data from the staff. The question was picked up by the member of staff who participated.

Question seven was originally worded “Which of the following would you like most?” To make this question applicable to the staff (without adding “not applicable” to the optional answers), the question was reworded to make staff think from a students perspective. After the sample it was changed to “As a student, which of the following would you like most?” This allowed the question to be kept and would provide some useful information afterwards.

The other issue that arose was the help section. During the testing phase a student mentioned that it would be useful to have one. Whilst the testing was carried out, the interviewer explained everything to the sample audience and no one looked at the navigation bar at the top of the web page. To emphasis the fact, a link to the help section was added above the login box. The help section was further modified to explain how to use the system in a step by step process.

Unfortunately with any survey you will not achieve a 100% participation rate. In the ECE departmental survey not all staff and students will take part so this will not reflect everyone opinions (Iarossi 2006). The aim was to have between 50 and 120 responses so that the total respondents in the survey represent around 10% of the total student population in the department and around half to two thirds of the staff population. This would help make the sample of respondents represent the department.

12.2 Public Survey

At the start of the fourth week, the survey was posted on the University forums. The feedback had been coming in for four weeks after a forum post was published and this was targeted at students of the ECE department. At the end of the second week staff were contacted by email to assist with the survey. Staff were provided details of a different account they should use to login to the system and leave feedback. By using multiple accounts the answers students left and the answers staff left could be distinguished and the findings could be placed into two groups.

Since the start of the survey there have been nearly 100 feedback responses. The ratio of respondents was approximately 22% staff and 78% students (when the number of responses reached 94). Any feedback left in the last couple of weeks was ignored due to two reasons: one due to the amount of feedback to analyse and two was down to the time this would waste to update the statistical results in this report. This could have made the data in this report inconsistent and looked badly for the student. The only information that was used near the end of the project were the comments the respondent left (if any) as the views left by late responses could be just as useful as the earlier responses.

13 Analysis and Results

In this section are the results of the survey and the respondent's reactions that came from their comments. The survey had statistical data (quantitative data) and the respondent's opinions / comments of the system (qualitative data). Along with all this information, all the feedback needed to be "cleaned" (Iarossi 2006) to ensure that the feedback each respondent left was consistent between questions.

13.1 Quantitative

Shown below are the questions and answers from the respondents, grouped into staff and students. All the predicted answers are in brackets and in blue font.

Question 1: Before today, I knew what a CAPTCHA is?

	Yes	**No**
Students	49% (60%)	51% (40%)
Staff	38% (70%)	62% (30%)

The answer to this question was a surprise to see. Believing that staff knew more about the CAPTCHAs than students was an over statement. When thinking about this in detail, students know more about the latest news in technology and the internet and the mature generation are just trying to keep up with it all.

When the sample group participated, at least a couple of students answered the question incorrectly, this is because the answer was given to them by the surveyor when he asked the respondent the question at the start of the demonstration to make the respondent aware of the subject area. Another comment that came up here was that the respondents in the demonstration knew what a CAPTCHA was, but when asked with only the word "CAPTCHA" half of respondents were confused.

Question 2: I found the numbers in the grid easy to read?

SA = Strongly Agree, A = Agree, N = Neither, D = Disagree, SD = Strongly Disagree

	SA	A	N	D	SD
Students	34% (20%)	55% (40%)	7% (20%)	3% (15%)	1% (5%)
Staff	38% (15%)	48% (35%)	9% (20%)	5% (20%)	0% (10%)

The answers here were pleasing to see and were positive compared to the predictions made. Between the staff and students only four of the respondents had a problem with the image. Looking at both groups, there was little to distinguish between the two. This was surprising to see as the prediction is was based on the fact that with maturity, text on a screen becomes harder to read. The distribution in this question is skewed towards the higher results (positive).

CAPTCHAs are generally thought of as being difficult to see. The aim for this project was to make them as easy as possible and the results here show this. With no distortion and the range of colours used to separate the grid references, grid lines and the numbers in the grid the developer has manager to keep all of these objects separate from each other. There was no overlapping in the CAPTCHA and this has certainly showed with the positive result.

As this was the first time most of the respondents used this system, this shows that the respondents could quickly read the text in the grid and select the numbers in the grid references.

Question 3: I found the equations easy to solve?

	SA	A	N	D	SD
Students	47% (20%)	49% (35%)	0% (30%)	4% (10%)	0% (5%)

Staff	81% (25%)	14% (40%)	5% (20%)	0% (10%)	0% (5%)

The questions in the CAPTCHA were based on three operations: subtraction, addition and multiplication. The answers here were fantastic to see as nearly all the respondents (95% of staff and 96% of students) agreed or strongly agreed that the mathematics operations were easy to work out. Out of all the feedback only three of the students disagreed and based on the large sample this is the small minority (1 in 25 students).

The aim when creating the CAPTCHA was to make it easy to read and to work out the answer so that it created a minimal delay to the amount of time to login. By using three simple operations the respondents were able to calculate the answer with little difficulty. If multiplication was not used, it would be interesting to see how easy people would have found the CAPTCHA.

Question 4: Did you need a calculator to work out the answer?

	Yes	**No**
Students	0% (20%)	100% (80%)
Staff	0% (20%)	100% (80%)

The answers from this question exceeded all expectations. This is a dream result that any designer of a mathematical based CAPTCHA would want to see. None of the respondents found the questions so difficult in the CAPTCHA that they would need a calculator to work out the answer from the grid. There was no separation between staff and students here and this is pleasing to see.

With the simple operations used in the CAPTCHA, the audience were not troubled by the difficulty of the equations. Had the number gone past ten, this would have been interesting to see.

The goal for the mathematical difficulty has been achieved (based on the results). A badly designed system was a major worry at the start of the project. Had remainder (of a division) been added to the list of operations, we may have seen different results here.

Question 5: Before you pressed login, what was the most time consuming task?

	Calculation	Magnifier	Reading Text	Username Data
Students	3% (25%)	0% (10%)	61% (45%)	36% (20%)
Staff	0% (30%)	0% (15%)	52% (40%)	48% (15%)

By analysing all the feedback using the method of cleaning, the results of this question revealed some false answers due to the fact that no respondent used a calculator (according to question 4). The three percent of students that used a calculator were part of the sample students. None of them used a calculator and the answers were not altered in the cleaning process to show that there were some inconsistencies in the answers from these two participants.

Tracking down the respondents would have gone against the goal of keeping the respondents anonymous. Once finding them, asking them to redo the questionnaire would result in biased answers, this is because it will not be the first time they used the software and that alone would make the survey inconsistent. The other method of asking those two respondents to answer only two questions would not have been good idea as their experiences of the system was back in week three of the project. Any changes could also bring further inconsistencies and the process may have to be repeated.

Out of the two groups, students clearly showed that reading the text in the image added the most delay before logging in. In relation, staff were not delayed as much in the login process. From the staff responses, it was fairly equal between the time to login and the time finding the numbers in the grid.

Question 6: What would have helped you answer the CAPTCHA question quicker?

	Simpler Equations	Nothing	Larger Text	Audio
Students	14% (40%)	48% (10%)	33% (30%)	5% (20%)
Staff	5% (35%)	57% (15%)	38% (35%)	0% (15%)

Out of all the questions, some of the predictions here were close to the actual result. The pleasing result was that most of the respondents believed that changing the CAPTCHA would not have helped answer the question quicker and login faster.

The use of audio and simpler equations did not rate as high to staff as they did to the students. This relates to the answers provided in question three about the math. Based on the answer to this question, a bigger image with larger text would be implemented if this project were to continue after the ten week allocated time. The other option would be to use a three by three grid as mentioned in one of the staff comments.

Q7: As a student, which of the following would you like most?

	Using CAPTCHA	Printing	PC Facilities	Library
Students	14% (10%)	31% (25%)	48% (35%)	7% (30%)
Staff	14% (20%)	29% (20%)	38% (30%)	19% (30%)

Out of all the questions, this is one of the most interesting questions in the survey because here we find out what the students want most and what the staff think the students want the most. The highest predictions in this question that came out top in the feedback and the weightings were fairly close in most of the responses.

Students on the department want the CAPTCHA more than they want the extension of the library hours. For the designer of the CAPTCHA this was a pleasing result. The students use the computers on a daily basis and feel that the availability of the computers along with the number of free print credits is a must have.

When speaking to the technicians about these results, Phil Trew mentioned “the students just want to come in at 3am to work and take it easy during the day”. The other technician on the department, Cedric Arrow said “nobodies ever in during the day”.

To expand on the comments, the students are not using the computer rooms 24 hours a day, 7 days a week. They would like to work between 9pm and 5am to get the most amount of work done and work closer to their deadlines. In an ideal world all the students would use the rooms and equally use the bandwidth across the day.

Q8: If the CAPTCHA was implemented on the ECE Intranet, would your level of usage:

	Increase	Stay Level	Decrease
Students	8% (10%)	80% (70%)	12% (20%)
Staff	5% (10%)	76% (70%)	19% (20%)

The last two questions in the questionnaire give a feeling of whether the respondent likes the CAPTCHA or not. This question targets the respondent's feelings of the system if they were made to use it on a daily basis. The predictions in this question were the closest compared to any other question in the survey.

The students on the department gave a closer symmetrical representational view of the system than the staff. One point that came up from a fellow student was whether the respondent knew that they would only need to verify who they are outside the University (like the current system). If this project was repeated a year later or with another surveyor, clearly stating this fact, it would be interesting to find out what results here would have revealed.

13.2 Qualitative

On top of the answers from the eight questions in the feedback form, additional comments were welcome from all the respondents in the survey.

13.2.1 From the Sample

When the sample survey was carried out, the student was interviewing a member of staff and fellow students (from all year groups) on the department. There were some useful comments that may have sounded obvious but it was good that others picked up on them. The comments included: the ability to generate a new CAPTCHA image and the addition of a help section.

None of these issues were raised in the interviews as the interviewer explained everything and answered all the questions the respondents asked. It was great that these were raised as it made the developer emphasise the fact in the help section specific areas to make this even better.

On a couple of occasions the idea of highlighting the two numbers in the grid would make it easier and even quicker for the user to grab the numbers and work out the answer of the operation. This made us think about a problem that was not thought about well. Reading from grid references (or maps) used to be a common occurrence (five to ten years ago) when people go travelling. Nowadays people don't use maps as much as route planners and satellite navigation systems do the work for you.

One other issue that cropped up a number of times was the "size of the grid". When designing the system, we felt that it was too big and if it was any larger that it would take the emphasis away from the login username and password boxes and the students would be confused about what they had to do. Then the respondents would say that the CAPTCHA would then be seen right in their face.

13.2.2 From the Public Survey

When the survey went out to the public, there were some repetition points that came up. With the wider audience available there were a greater number of ideas from the group as well.

Annoyingly a small number of students did not read the help section as they have made this obvious in the comments "I did not know what to do". When the public survey was out, the help section was emphasised to the end user on the ECE forums and the website. It was a shame that some people ignored it. In any case, this would always be a problem.

On a few occasions the strength of the CAPTCHA was an issue as it was too simple and easily breakable. This is a fair comment as the respondents did not know about the aims of the project and why clearer text was used in the CAPTCHA. It's interesting that people expect new CAPTCHAs to be difficult to break and this goes against the idea of the critics (see section 16 – Literature Review).

Out of all the operations the use of multiplication raised some question. Some respondents did not find the multiplication overly difficult, but it was time consuming compared to addition and subtraction. Other respondents loved the math and others enjoyed the answering questions. Reactions varied here and we thought that this was good to hear.

One of the negative comments in the survey included: “That whole grid thing is ridiculous. It’s only an intranet where lectures help student not a bank account”. It’s great to find out all opinions and if someone did not say this, how could their views be analysed. Another student who rarely uses the Intranet said they “can manage without it”.

From the number of replies, alternative methods of different CAPTCHAs were not mentioned till a third of the way through the survey. A student picked up on the use of sound for disabled users. Another student thought that sound should be added and linked with this system. This would have taken another few weeks to implement, however sharing the information between the systems would have been a challenge in it’s self.

Along with comments from the sample survey a few more students would like a larger grid and a member of staff suggested a 3x3 grid (instead of 4x4) giving more room and easier navigation to the numbers. A student thought that with thicker lines it would make the grid easier to read. These very useful points would make us design the grid using the suggestions from these two respondents if the project were to continue after the ten week allocation.

A couple of feedback responses later, a student mentioned the idea of moving the content by a few pixels to make it harder to break. The other suggestion is to use different fonts. One thing not mentioned by this respondent is whether they prefer distortion or not. If the respondent wished to have clearer fonts, then this along with the pixel movements would be used in a 3x3 grid solution.

An idea that was similar to the sample audience, instead of highlighting the two numbers in the grid that show up, to add a line over the ones that are not selected in the operation. This sounds like a better solution than highlighting two numbers because it means that a bot would have to analyse each square in order to grab the correct numbers. Where two numbers are highlighted a certain colour, this would be easier to a bot to find as the two numbers stand out.

One of the students was kind enough to test cross site scripting (XSS). The good findings are that the website is protected and it’s always nice to know we have extra security improvements. When users missed out the CAPTCHA by default there would be a generic

image. One colleague mentioned that a separate message for the CAPTCHA would be helpful. In the design stage this was avoided as feedback from others lead the removal of individual error messages.

The staff had some interesting responses. One member of staff wished to see more maths questions and some images based answers. An idea of randomly having the CAPTCHA appear was an interesting concept from a staff member. The idea here is that every 25 login requests a user is shown the CAPTCHA or possible every 100. This would have to be monitored on the server, but it could be implemented, given more time. This method of implementation would serve a purpose without annoying every user.

When looking at some of the feedback, students are welcome to the idea of the extra security, but not so much the Intranet, maybe another resource.

The use of JavaScript confused a couple of students as the purpose was to refresh the image. One student mentioned that the CAPTCHA could be used to go to the help section and a new image should only appear on a page refresh. It's definitely a valid point and unless we spoke to a wider audience about this, we could just be personalising the website towards one or two people.

Two of the most favourable comments came from consecutive feedback. The first saying: "overall I think it's brilliant. Keep up the good work" and the second saying "how did you conceive this great idea"? Without flattering the developer, it's nice to see good comments after some negative ones.

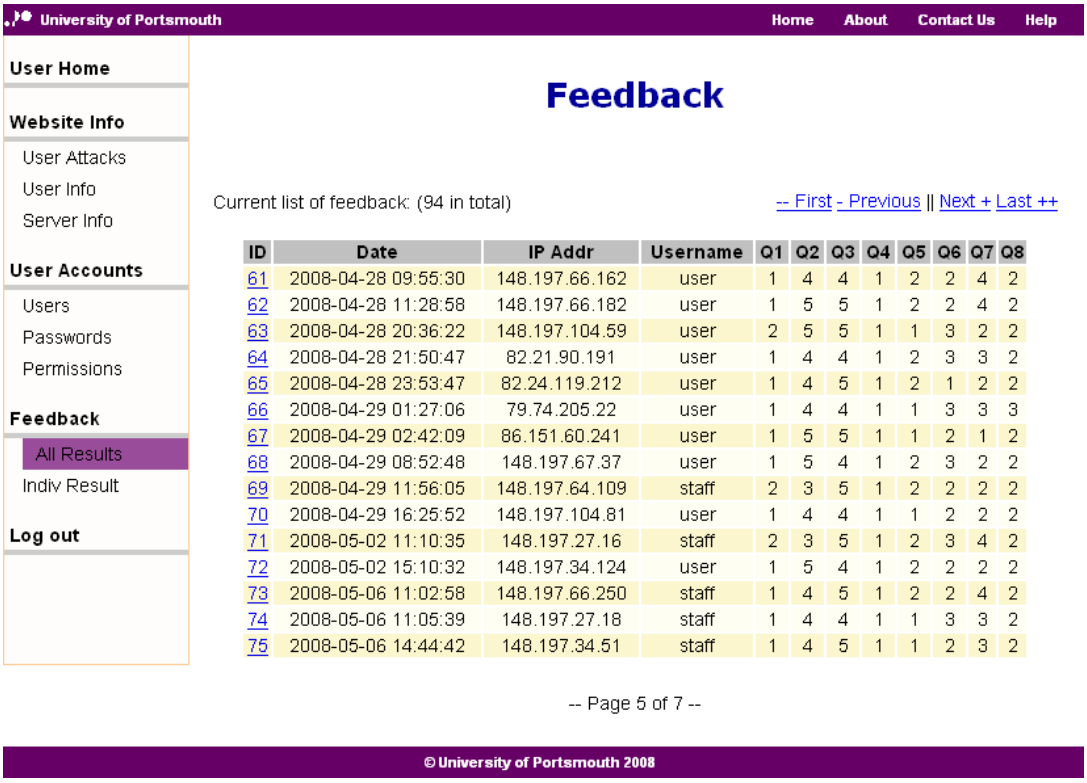
One of the most interesting comments from any respondent was the arrangements of grid references. We kept numbers on the left and letters at the top. This is because "a-Z" maps of towns and cities use the same layout. Another example is spreadsheet software like Excel. It would have been interesting to know why the respondent would find this more useful and where they would have seen another grid layout.

13.3 Cleaning

The cleaning process had to go through a few steps before this procedure could be carried out. This included: making the cleaning process easy to carry out, determining what to look out for and finally what to change if there are any inconsistencies.

13.3.1 Making Cleaning Easier

In order for the feedback to be acknowledged by a third party, a process of cleaning (Iarossi 2006) was needed to check the consistency of the respondents answers. By using MySQL of the respondent's feedback results from the database could be listed in a table with (see Figure 13-1).



ID	Date	IP Addr	Username	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
61	2008-04-28 09:55:30	148.197.66.162	user	1	4	4	1	2	2	4	2
62	2008-04-28 11:28:58	148.197.66.182	user	1	5	5	1	2	2	4	2
63	2008-04-28 20:36:22	148.197.104.59	user	2	5	5	1	1	3	2	2
64	2008-04-28 21:50:47	82.21.90.191	user	1	4	4	1	2	3	3	2
65	2008-04-28 23:53:47	82.24.119.212	user	1	4	5	1	2	1	2	2
66	2008-04-29 01:27:06	79.74.205.22	user	1	4	4	1	1	3	3	3
67	2008-04-29 02:42:09	86.151.60.241	user	1	5	5	1	1	2	1	2
68	2008-04-29 08:52:48	148.197.67.37	user	1	5	4	1	2	3	2	2
69	2008-04-29 11:56:05	148.197.64.109	staff	2	3	5	1	2	2	2	2
70	2008-04-29 16:25:52	148.197.104.81	user	1	4	4	1	1	2	2	2
71	2008-05-02 11:10:35	148.197.27.16	staff	2	3	5	1	2	3	4	2
72	2008-05-02 15:10:32	148.197.34.124	user	1	5	4	1	2	2	2	2
73	2008-05-06 11:02:58	148.197.66.250	staff	1	4	5	1	2	2	4	2
74	2008-05-06 11:05:39	148.197.27.18	staff	1	4	4	1	1	3	3	2
75	2008-05-06 14:44:42	148.197.34.51	staff	1	4	5	1	1	2	3	2

Figure 13-1 All respondent feedbacks

Although this page listed the feedback from all the respondents with the use of pagination (to show 15 results per page), it meant that interpreting the answers was a time consuming task. By just using this table there was a high risk of error by reading the incorrect row or even converting the answer from the number to the answer on the feedback form. The row colours

were alternated to help read each row easier in the process, but this was not enough to help this time consuming method of cleaning.

To help with the screening process, an extra column was added, called “id” and this would be used to reference a specific feedback response on the website. In PHP there is a method called get (or even request) to grab the information from the address bar of the website browser. When the administrator or the supervisor clicks on an id (say 71) the URL they would go to is:
http://www.baldeepbirak.com/project/admin/fbresult.php?id=71

The filename they went to is “fbresult.php”. Just going to the page would show the first result. With the addition of “?id=71”, the id of 71 would be searched from the database and then the records were printed (in PHP “echoed”) to the screen (see Figure 13-2).

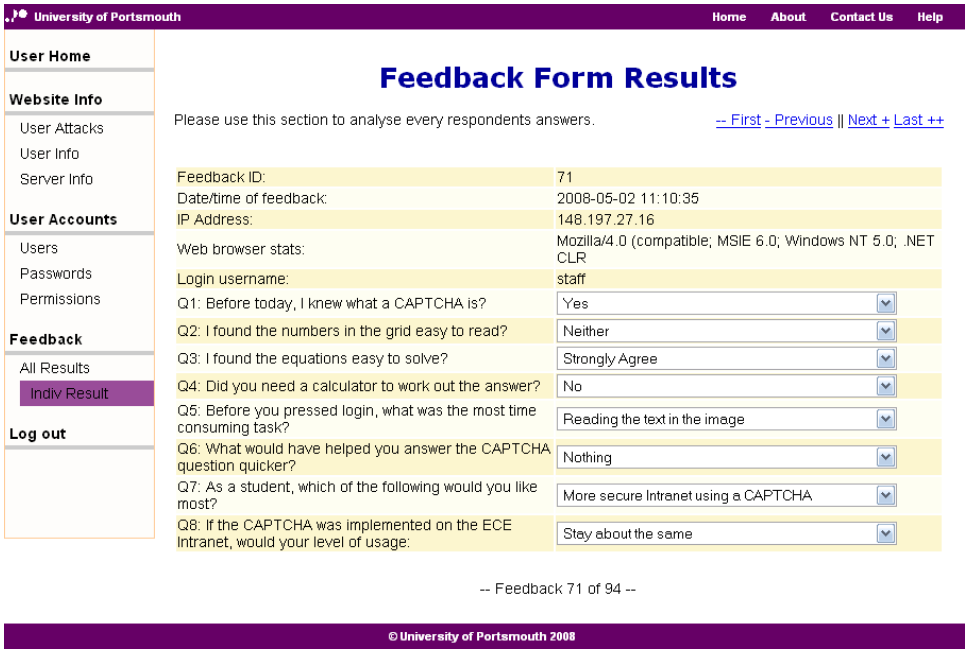


Figure 13-2 Individual feedback results

The result of each question answer was translated back to the response the respondent left. This helped read all the feedback results quickly, easily and reliably. With the addition of the id field, the navigation between the results was made easier.

There was some information not shown on here. The optional fields of name, email and comments were sent via email to keep the database small and minimise against the threat of

SQL injections. This was to make it harder for an attacker to remove the contents of the database.

The only flaw with this method is that it was vulnerable to SQL injections (with the request from the address bar), but this was resolved by using the PHP "int" function to make sure that only numbers are queried with the database.

13.3.2 What to look out for?

In order to start analysing the feedback, we need to know what we want to find. It's all good saying "looking for inconsistencies", but what were the inconsistencies? After analysing the feedback form answers, there were six inconsistencies found by the survey analyst. They were:

- **Question two and five:** The respondent found the numbers in the grid easy to read, but the most time consuming task was reading the text in the image.
- **Question two and five:** The respondent found the numbers in the grid easy to read and they required the use of a magnifying tool on their computer to read the text in the image.
- **Question three and four:** The respondent found the equations easy to solve and they required the use of a calculator to work out the answer to the question.
- **Question four and five:** The respondent did not need a calculator to work out the answer of the question, but the most time consuming task was using a calculator.
- **Question three and five:** The respondent found the equations easy to solve, but the most time consuming task was using a calculator.
- **Question seven and eight:** The respondent found would like to have the new system implemented as the University login page for the Intranet, but their level of Intranet usage would decrease of the system was implemented.

The six points above were classified into three categories with the use of a traffic light system (see Table 13-1) to determine the level of the problem.

Level	Situation	Description of the issue
Green	Could be a conflict	Point one - if the image is easy to read, would read it be the most time consuming task?
Amber	Suspicious	Point three – does the use of a calculator warrant an easy answer. Point five – if the equations are easy to solve, would the use of a calculator cause the most delay.
Red	Contradictive answers	Point two – surely if the text in the image is easy to read, a magnifying tool is not necessary, is it? Point four – if the respondent did not use a calculator, then how would this be the most time consuming task? Point six – if the respondent wants the system implemented on the Intranet, why would they use is less?

Table 13-1 Issues with cleaning

13.3.2.1 Leading the respondent

Apart from these inconsistencies, another issue needed to be looked at. This involved analysing the mass feedback looking for answers where a respondent was led to the specific answer.

Looking at all the questions and answers, only one had some eyebrows raised. This was question four. The question was “did you need a calculator to work out the answer”? The options for this question were “yes” or “no”. Believe this or not, every respondent said “no”.

Part of the analysis for this was, “was this the right question to ask”? Looking at the earlier drafts of the questions, this was worded “how did you rate the calculation” and the answers ranged from “very easy” to “very difficult”. The problem here was that this was based on personal feelings and the analyst required factual answers so the answers would not correlate across the large sample. Baring this in mind, the question four that all the respondents saw was the right question to use.

13.3.3 What Inconsistencies?

After determining what problems to look for, the next problem was to look at each respondent's feedback for answers that may contradict with an answer from another question. Using the points from section 13.3.2 and the individual feedback results page (see Figure 13-2) the analyst was able to work through all the responses with ease.

There were inconsistencies from three of the six points in the section 13.3.2. They included points one, four and five. Out of the three, point one came up on nearly half the feedback responses. This questioned whether it was a good point or not. The biggest worry was if it was not a bad point, how this would have looked for the analysis. When looking back at the point at a later date, it was clear that that is was not an issue and it could be removed. This meant that there were still two points where responses were inconsistent.

Out of all the respondents, two of them were both students and from the initial test sample of fourteen people. They both said:

Point 4	Q4: I did not need a calculator	Q5: The most time consuming task was using a calculator
Point 5	Q3: I found the equations easy to solve	Q5: The most time consuming task was using a calculator

Table 13-2 Respondent inconsistencies

The only difference between to two sample testers is that one of them "strongly agreed" that the questions were easy to solve and the other said "agree" on question three. When the survey was out to the public, no inconsistencies were found. This was a perfect result for the form developer giving a 100% clean record from the public survey.

From the sample feedback, there were issues to sort out. There were multiple corrections that could be made from the two feedbacks. They included:

- removing the two feedbacks
- ignoring the answers for question five on the two occasions
- tracking the two students, asking them to re-answer the question
- leaving the results as they are

The two feedback responses had valid points for all the other questions and by removing them it felt like skewing the data to favour the analyst. This is not what the analyst wanted, even if it's only two results. By ignoring the answers of one of the questions (believing that question five was the issue) felt like the obvious choice. This however was not an option as the two respondents may have answered questions three and four incorrectly.

Looking at the feedback form, the surprise was that no respondent selected "using a magnifier" as this was next to the option "reading the text in the image". Had this of been the case, this would have made more sense for an incorrect selection. The other issue is where was the sample survey carried out? Each of respondents all used the same computer and the same web browser so this ended up out of the equation.

The next option was to track the two respondents and ask them to answer the question again, maybe even all of three questions. This brought lots of complications, such as breaking the ethos where all users should be anonymous, unless the respondent publicized this fact in the name, email or comments column of the form.

The two respondents left their opinions back in week three of the project, so after five or six weeks, their views would be totally different. Finally, this would make them repeat testers. Most, if not all the respondents only used the system once, so by repeating the test with them would not make their results fit with the rest of the respondents as this survey would only count if everyone else took part again.

The final solution ended up being the best solution. This was where the results were left untouched. By leaving the answers as they are, everyone can see that there was an error and this makes them more reliable to a third party by showing human error.

14 CAPTCHAs

Around the internet, there are uses of CAPTCHAs on many websites. This includes the use of free services such as email, forums and blogs (Atwood 2006). Some of the CAPTCHAs are good and some of them are bad. Here we discuss them in further detail with some proof using the CAPTCHA image you would see if you went to the website.

What makes a CAPTCHA good or bad is how difficult it is to the end user who uses it. The CAPTCHA is not just about the image, but the implementation of it. To determine if a CAPTCHA is difficult to answer (making it a good one or a bad one), depends on the following criteria:

- The time it takes to answer?
- The number of attempts you are given?
- What alternative methods are in place?

14.1 Bad CAPTCHAs

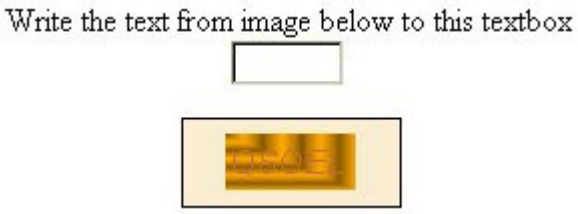
Just to answer the question of a CAPTCHA should not take more than ten to fifteen seconds when used regularly (on three or more occasions). The user should not need to struggle by squinting at the image to wonder if part of the image looks like a certain image or a character. Our definition of a bad CAPTCHA is "where a bot has a greater chance work out the content from the image than a human".

If the user were to get the answer wrong they should be given the option of a number of attempts or even an audio version. If these are not available and the next image is just as difficult as the previous one, the developer should think twice about the CAPTCHA design. Here are a few examples of some bad CAPTCHAs and what makes them bad. Some of them may not be difficult work out, but how they are implemented will be explained in that case.

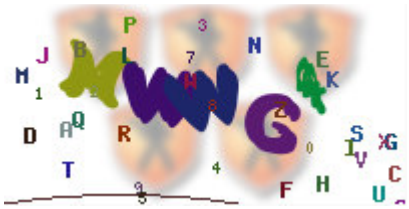
We believe that the worst CAPTCHA at this moment of time is from a website called "RapidShare" and this image was taken just before it was pasted into this report (RapidShare 2008). In this CAPTCHA you have to pick the four letters where the cat appears behind them. None of their CAPTCHA appears to be easy. The biggest problem about this CAPTCHA is that before you even see the image you have to wait over a minute as the website hosts files for the public for free.

If you were to answer this incorrectly, you will have to wait another minute to see a new image. It had taken us a second attempt to work out the answer correctly and we believe that we were lucky. Had we of failed at this attempt, we would have given up.

The reasons we say it's the worst CAPTCHA to this date are: that you have to wait at least one minute to see the image, then you only have one attempt and would be forced to wait another minute after an incorrect guess and finally there are no alternatives. You cannot click the image to see a new one and if you are blind, there is no audio option.



This next one was this small on a website (Maddaloni 2007). With some squinting we would guess the answer to be "qsoel". You would be expected to use a magnifying glass to make sure you answered this correctly. This could have been a good CAPTCHA if the following modifications were made: changing the colour of the characters and increasing the size of the image and text.



Finally this last one was taken from a student project called "incredi" (Gearing 2008). The CAPTCHA designer used this project developer's idea of using a PHP file to generate an image. When looking at the image, the first character looks like an "m" or an "n". The next pair could be "wn", "vw", "vm" or something else. The next one is a "g" and is the only clear letter from the CAPTCHA. The final letter is not clear as it could be a number or a letter.

There was no clue on how many letters were used in the image and there was not a clickable option to change the image. As this was a registration page, whoever was signing up at the time would have to enter all their details over again and this is what makes us angry with even the newest web developers implementing awful CAPTCHAs.

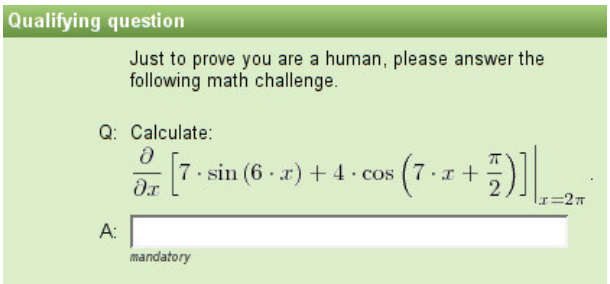
14.2 Good CAPTCHAs

A good CAPTCHA is essentially the opposite of a bad one but some come in between both these categories. After a few attempts a user should be able to look at the CAPTCHA and work out the content of the image. Where there maybe a question, this should not take longer than five to ten seconds to work out.

In this project, the sample students managed to pick the numbers in the grid between three and seven seconds. As this was a new test for them, this hopefully showed that the design of the CAPTCHA was planned well and by using the feedback left by all the other participants. The question added the most delay between five and fifteen seconds. Hopefully with practice (using the CAPTCHA in this project), the time to grab the two numbers from the grid and answer the question should take between five and ten seconds over the period of six to twelve months.

When people work out the answer incorrectly to a CAPTCHA image whether it is hard to read or by mistake, there should be an option to change the image, where the user can click the current image or a link that is nearby to the image. Some websites use two different types of CAPTCHA, possible with the image and sound (such as Hotmail and Google's Gmail registration pages). Ideally the CAPTCHA in both should be the same, but generating shared content between the two different media types adds a problem for companies that do not want their CAPTCHA to be broken with ease where one system is weaker than the other.

This brings us onto the next factor, visibility. CAPTCHA developers are distorting the content in the images to make their CAPTCHAs hard to break. A good CAPTCHA will have little to no distortion so that a human can see the content and where there is text, read it with ease. Below are some examples of this and what makes them a good CAPTCHA.



What makes this a good CAPTCHA is if it's implemented in the right place. This would be great to use for the Mathematics department but bad for the English department at Portsmouth University. The text in the image has no distortion, but

working out the answer may be time consuming for the end user.

The next CAPTCHA was made by a past student of the University and is currently used by the department forums (University of Portsmouth 2008) when people register. What makes this a great CAPTCHA is that the font is "clearly" easy to read and the developer used the same font for all the letters. The way this is implemented allows the user to change the image if they find the current image difficult to read. This is done by clicking the image for image to change and the new CAPTCHA appears (when JavaScript is enabled on the user's browser). This CAPTCHA is currently on the registration page of the forums and with this "click to change image" feature, this makes this good CAPTCHA and even more friendly for the end user.

An early day example of CAPTCHAs before people started distorting the text (Atwood 2007). What makes this good is that it is easy to read whilst serving its purpose. The unfortunate thing is that it can be broken by a programmer in a short period of time.

For the smaller websites on the internet, why not use a CAPTCHA like this as the financial value of their websites content may not be so much. If you are going to use a CAPTCHA just consider one like this. This CAPTCHA is used on the "coding horror" website for their blogs and even to this day, the CAPTCHA is still showing the same screenshot. The word is still "orange" and it has managed to beat a large percentage of the bots. The best part about this CAPTCHA is that it can be listened to and guess what the audio plays, yes it's the word "orange". This is why it gets our approval as the best CAPTCHA.

14.2.1 A New Alternative

Finally, a different sort of CAPTCHA has emerged. This CAPTCHA does not generate an image, nor does it generate sound. How could this be you may ask? This is done with text that can be typed by a keyboard. Anyone could create one as it does not require any programming, but this would be a bonus.

This CAPTCHA is called tppCAPTCHA (PHP Pro 2006) and uses ASCII text to show characters and numbers to the user. ASCII art has been around for many years, but this type of CAPTCHA is new to most and one we believe will be heard by many over the next few years. A screenshot of the CAPTCHA taken from the website can be found in Figure 14-1.

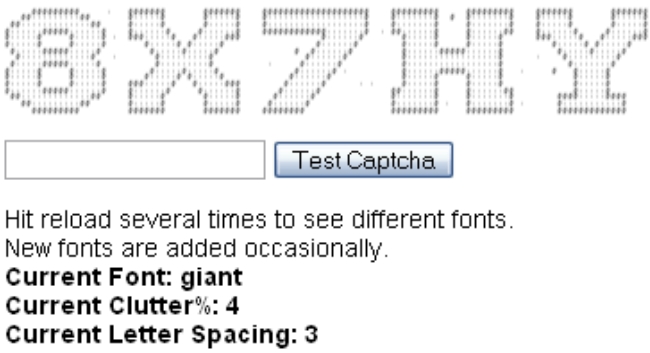


Figure 14-1 ASCII CAPTCHA

To prove that the CAPTCHA is text based, a text file of this can be found:
http://www.baldeepbirak.com/project/downloads/

With no distortion and little clutter (that can be set by the website) this type CAPTCHA gives a good name for the community and we believe that this could help improve the profile of CAPTCHAs if more people were to use this CAPTCHA on more websites. We find this CAPTCHA even better as it was created using PHP so a PHP developer should be able to create their own or develop the one used by the company even further.

14.3 Alternatives

Where registration of user accounts are needed on websites, CAPTCHAs should not be used. By implementing a registration system differently, you could bypass the need of a CAPTCHA altogether. Some workarounds to using CAPTCHAs are:

- Contacting the site administrator via a contact form or an email
- Secondary registration

A secondary registration can be implemented in two ways. The first solution is where the user enters their registration information and then the details are hashed and sent to their email address. The content in the email would include a link to finish the registration and a random

password. When the user clicks on the link they would be forced to change their password and the account is created. To prevent another user guessing the URL and abusing the system, the password in the email would need to be used as verification (under current password).

The second solution relies on two tables in a database with the same fields (or mostly the same). A user would register and their information would be stored in one table. They should then be sent an email including another URL to visit that includes a hash of the details they entered in the registration page. When they read their email and click on the link, they would have to re-enter their information and submit the details. If the details match, the details are added to a second table and the details from the first table are removed.

With these alternative methods the use of a CAPTCHA would not need to be used for a registration system. To make it easier for the user, asking for only three to five details at most would be simpler for them and the extra security features would not be an issue for them. By forcing a user to change the password and some other details, this would complete the entire registration, ensuring that it's not a computer registering. Then the user can use the system.

The major problem with these solutions is that they would require an existing email. However if a company website like Hotmail used this approach, they could allow the user to bypass the CAPTCHA if they have an existing email. If not then they could be forced to use their CAPTCHA.

Not all registration spammers create email addresses for the accounts they register with. By using the email address you are limiting the number of successful accounts created by a spammer. With the use of unique email addressed on a system, you can protect against an email address exploit even when the user tries changing the email address.

Registrations and CAPTCHAs are vulnerable to one major problem. This is not an automated attack problem, but instead a human attack. A human may be rewarded with free gifts, money or another exchange for goods in return for answering a question on a form or a manual registration on a website (Doctorow 2004). This type of attack is a major problem where hundreds and even thousands of people in poorer countries are willing to fill in forms and

answering questions everyday of the week, over many years to make a decent income (Leyden 2008).

The only benefit to the websites is that manual registrations take longer, but if the human is used to teach the computer how to use the web page ("playback bots"), this increases the problem. A method around this is to restrict the number of registration per IP address per day. This would lower the number of registration, but some spammer register multiple accounts from hundreds of computers and two to three registration per IP address can bypass these types of restrictions.

15 Limitations

Most if not all CAPTCHAs come into one of two categories that determine how to answer the CAPTCHA. They are either entering the answer in a text box or clicking on the correct image(s). The challenges you would be expected to see include:

- enter the answer of a image with text
- click on an image where it resembles something
- read a question in an image and enter the answer in the box
- or a combination of the three above.

In this project the CAPTCHA was based on a puzzle that required the respondent to perform a math operation. This CAPTCHA along with nearly all of the CAPTCHAs seen by man, have a limit for the number of answers. As this CAPTCHA was based on subtraction, addition and multiplication there was a limit to the number of answers there could be.

From the first analysis we know that the range of answers could be between -9 and 81 and many would be fooled saying that the probability of success is a 1 in 90 chance. Just basing this on answering the CAPTCHA (without the username as password) we can say that this is incorrect. This is due to the fact that two or more different mathematical operations could end with the same result. With any multiplication any prime number answers between 10 and 81 will not be reached. On top of this answers cannot be reached where a number greater than nine is needed in the multiplication (e.g. 80 require the numbers 8 and 10).

In order to show the range of values and the chances of one number appearing more often than another, three number grids are shown (see Table 15-1, Table 15-2 and Table 15-1). They include one for subtraction, addition and multiplication.

Subtraction	0	1	2	3	4	5	6	7	8	9
0	**0**	1	2	3	4	5	6	7	8	9
1	-1	**0**	1	2	3	4	5	6	7	8
2	-2	-1	**0**	1	2	3	4	5	6	7
3	-3	-2	-1	**0**	1	2	3	4	5	6
4	-4	-3	-2	-1	**0**	1	2	3	4	5
5	-5	-4	-3	-2	-1	**0**	1	2	3	4
6	-6	-5	-4	-3	-2	-1	**0**	1	2	3
7	-7	-6	-5	-4	-3	-2	-1	**0**	1	2
8	-8	-7	-6	-5	-4	-3	-2	-1	**0**	1
9	-9	-8	-7	-6	-5	-4	-3	-2	-1	**0**

Table 15-1 Subtraction grid showing all possible answers of this operation

The most common answer here is the number zero. This appears more than any other number as a result of the same two numbers used in the operation where a number minus itself results in the answer of zero. There is a symmetrical distribution of the numbers in this grid. In order to get the highest number, you would have to have the highest number and the lowest number where the highest minus the lowest keeps the highest number as the answer (e.g. 9 - 0 = 9).

The next grid is the addition table (see in Table 15-2). There is another symmetrical distribution that varies slightly with the previous one. Here the most common answer is nine as variations of the lower numbers making this total possible. The highly probable answer in this grid and the subtraction grid is the median of the range. For subtraction 0 was half way between -9 and +9 and in the addition grid, 9 was half way between 0 and 18.

Addition	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	**9**
1	1	2	3	4	5	6	7	8	**9**	10
2	2	3	4	5	6	7	8	**9**	10	11
3	3	4	5	6	7	8	**9**	10	11	12
4	4	5	6	7	8	**9**	10	11	12	13
5	5	6	7	8	**9**	10	11	12	13	14

6	6	7	8	**9**	10	11	12	13	14	15
7	7	8	**9**	10	11	12	13	14	15	16
8	8	**9**	10	11	12	13	14	15	16	17
9	**9**	10	11	12	13	14	15	16	17	18

Table 15-2 Addition grid showing all possible answers of this operation

The final grid relies on the mathematical operation of multiplication. When multiplying numbers it is not possible to have an answer where the number may be a prime number (e.g., 11, 13 and 17) but these lower numbers will appear in the addition grid (see Table 15-2) as two numbers added together can produce these results. Answers up to the number 30 make up 74 of the 100 possible answers from the multiplication grid. A list of all the totals for each number between -9 and 81 can be found after the Appendices. The most common answer in the multiplication grid was zero.

Multiplication	0	1	2	3	4	5	6	7	8	9
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	1	2	3	4	5	6	7	8	9
2	**0**	2	4	6	8	10	12	14	16	18
3	**0**	3	6	9	12	15	18	21	24	27
4	**0**	4	8	12	16	20	24	28	32	36
5	**0**	5	10	15	20	25	30	35	40	45
6	**0**	6	12	18	24	30	36	42	48	54
7	**0**	7	14	21	28	35	42	49	56	63
8	**0**	8	16	24	32	40	48	56	64	72
9	**0**	9	18	27	36	45	54	63	72	81

Table 15-3 Multiplication grid showing all possible answers of this operation

15.1 Probability

When analysing the chances of an answer that would appear with a certain operation, the results were surprising at first, but after looking at them thoroughly they were logical (meaning that they made sense). For addition and subtraction the most common answer (i.e. mode) was the median of the range. The probability of the answer in both cases was a 1 in 10 chance. For multiplication, the most likely answers were skewed towards low numbers and the mode here was zero (the lowest number result). The probability of an answer being zero was a 1 in 19 chance.

When grouping all the results together there were 42 numbers that could not appear at all (see Appendices) with 26 of them being answers that are 50 to 81. Using all this information, if you were asked the chance of a number appearing, you would have to say that it is 1 of 49 possible numbers that could appear and the greatest possibility is for the number 0 with a 1 in 10 chance (30 in 300) of the number appearing.

The next probability would be to determine the chance of logging in successfully within a number of attempts. This can be calculated with a probability tree or to make this simple using the calculation: 1 / (49^10). This is where the chance of success (which is one) is divided by the number of possible answers multiplied by itself by the number of attempts before the user is locked out (currently this is set to 10).

With the use of probability there is “in theory” a chance that every outcome has an equal chance of appearing. This does not take into account that the respondents were to enter a different answer or no answer at all in to the answer box. However for an attacker that wishes to brute force the login page and is aware of this information, then the probabilities apply to that situation.

There was one point that only one student (Samuel Groves) picked up on was the random number algorithm used for the CAPTCHA. In PHP there were two random function, “rand” and “mt rand”. They use pseudo random generation for the numbers and this is not a perfect random generator. When researching into “real random” the use of seeds were needed. The random function and Mersenne Twister’s random algorithms were used for the various random number generations for the CAPTCHA.

One type of seed that was considered was called “hotbits” as a genuine random (Walker 2006). The idea sounded like it was worth trying, but it was rejected as testing would have been limited. The website provides a seed of hexadecimal number that is unique to you. We thought that this could be less random then the current solution, so then it stuck.

16 Literature Review

Back in 1950 a man name Alan Turing invented the Turing Test (Hodges 2001) and this was used to see whether a person could distinguish between a computer and a human. Many years later, in the year 2000 three staff at Carnegie Melon University and another working for IBM (Wikipedia 2004) came up with the concept of the CAPTCHA.

This article wants to show to others that the work carried out in the project can be used by the industry. The aim is to prove that by creating simpler CAPTCHAs we can use them effectively and encourage everyone to use them. Another reason for this article is that organisations like W3C publish a document (W3C 2005) with their opinions where some points that are vague and their paper itself is out dated.

16.1 Is this Impossible?

Looking at W3C's article at the start it mentioned that CAPTCHA "systems make it impossible for users with certain disabilities to create accounts, write comments, or make purchases on these sites". With modern software blind people can use narrative software on the internet. This may add difficulty, but this is by no means impossible. CAPTCHAs tend to be on the registration pages of a website and many online shops just want your card details (such as Novatech, PC World and Argos). All of their registration systems do not require the need of a CAPTCHA, nor do they implement one. If W3C were to enlighten us on the websites they saw, this would be useful.

W3C then say the following statements: "many bloggers claim that CAPTCHA challenges are successful in eradicating comment spam" and then say "any other method of comment spam control should be reasonably successful". If members of the public agree that CAPTCHAs are helping the blogging website then this must be a good thing. Alternatives of CAPTCHAs we found include: the Wordpress "hashcash", "honeypots" and even swapping field names in forms (e.g. name and email).

16.2 The Alternatives

The system used on Wordpress called “hashcash” (Back 2007) makes use of JavaScript on the user’s browser to tell whether if the initial page load was carried out by a human or a bot. This system uses JavaScript on the client and server end to work this out. If a bot loaded the web page then this would not verify with the system. Additionally this would also be a problem in Firefox, Opera and Safari where the user can turn off JavaScript with ease.

The idea of “honeypots” (Batchelder 2008) is to attract a bot to add data into hidden fields on a form. The idea here is that a human cannot see hidden fields so only enters in information in the fields they can see. A bot would enter data into all fields and by validating the form entry on the hidden fields the administrator of the website can combat a percentage of spam on their blogs, forums or other input systems. There are multiple methods to hide text boxes on forms including: html, JavaScript and CSS.

The other option is to switch the order of field entries or even make the user enter one detail in one text box and another detail in the other. A good example of this would be where a name is entered in an email text box and an email is entered in the name textbox. What makes this a good switch is that it makes it easier to validate the data by looking for the “@” symbol.

There are some good and bad points for all three of these alternatives. The “hashcash” system relies on the user to have JavaScript enabled and more and more people are turning this off as a result of how it is used on certain websites. The idea of switching data entry on a form would be confusing to the user of the website. Unless a human read a notice somewhere on the website of what to do, they may be considered as a spammer by the system. The other option of changing two of the text box names with make the prevention only half as good as an spam attacker would just fill the information in the correct order by using the name of the text boxes.

The use of “honeypots” is a great idea to use and was used in this project. This would only work well as long as the browser supports JavaScript and / or CSS, otherwise the html option of hiding text boxes (that is easier to detect) would have to be used. This brings us to another problem of “playback bots” where a human analyses the website and creates the script that

avoids these traps. So where W3C say “techniques are as effective as CAPTCHA, without causing the human interaction step that causes usability and accessibility issues” would make this statement false. These alternatives found by us have flaws and in the article W3C do not provide such alternatives. If there are some good systems out there, please show them to us so we can use them.

In the next section of the study, W3C say “the ability to create several accounts to multiply a user's privileges is often the cause for these Turing tests to be put into place”. This shows they are aware why CAPTCHAs are used and that they serve a purpose. This is explained in the same paragraph when they say “these sites wish to provide credentials to humans while eliminating robot access to the same resources”. This shows they acknowledge the reasons for the use of CAPTCHAs and have no complains about this matter at this time in the article.

16.3 Accessibility

The next agenda is about human rights. Unfortunately there are accessibility issues for people with various disabilities. These include: content on the internet, radio and telephone communications prevent equal access to people who are unfortunately deaf. Where the internet is concerned, not all images have “alt” text for narrator software to speak out to the person. When it comes to CAPTCHAs we feel that there should be a bypass system in place or a CAPTCHA with sound as well. W3C say that the accessibility could break human rights laws.

With a lot of media on the internet, managing to make this accessible to everyone no matter what disability they may have can be a very difficult problem. Unless all website designers are forced (by the power of country wide legislation) to test all their web pages then most of the content around may not be accessible to others. Older computers, operating systems and software could prevent many websites working properly (such as Netscape 4 running on Unix at Portsmouth University) and the argument could be that website designers should make their websites backward computable and by not doing so the human rights law or some other discriminating law could be used against the website.

In the next section of the article W3C go into the use of logical puzzles. The use of “simple mathematical word puzzles, trivia, and the like may raise the bar for robots” but this is

"subject to defeat by human operators". If the websites target certain audiences then the CAPTCHA they use can be targeted by: gender, ethnicity and culture. By using puzzles there are limits to the number of possible questions, images or whatever content is shown to the user (as mentioned in section 15.1 - Probability).

For this project there was a target audience of technology students and the use of mathematical equations. Knowing the level of understanding of the audience the evidence from out survey showed that 95% of staff and 96% of students agreed or strongly agreed that the equations were easy to solve. This shows that as long as the CAPTCHA is targeted to the correct audience, it can be used in an adequate place.

16.4 Sound CAPTCHAs

As CAPTCHAs have been developed over time a means to help people with disabilities (such as the blind) has also been looked into. One thing that W3C pick up on is the fact that this type of CAPTCHA is used and websites like Hotmail use sound CAPTCHAs with distortion (background noise).

When testing the CAPTCHA systems used by three of the biggest websites on the internet including: Google's Gmail, Hotmail and Yahoo mail registrations, we found that only Hotmail and Gmail offer the sound CAPTCHA option. Just based on this does not mean they are much better. In fact, out of the three Google are the only company to not offer the ability of generating a new image if the user found the image on the screen difficult to read.

We decided to listen to the CAPTCHAs by Hotmail and Google a couple of times and what was interesting to hear was that Hotmail's CAPTCHA is spoken out once and Googles is spoken out twice every time it is played. This is very useful as the user may not have heard one or more characters with all the distortion (background noise) in the audio CAPTCHAs of these companies.

We tested the CAPTCHA on four of the most commonly used website browsers on Windows and found that Hotmail's CAPTCHA only plays in Internet Explorer (IE versions 6 and 7 were used). The same tests were carried out with Google's CAPTCHA and it worked in IE, Firefox, Opera and Safari. This did require the use of JavaScript and a sound plug-in enabled

on the users machine, but when we turned JavaScript off, a new window appear and the audio plug-in was used to play the sound file using Windows Media Player to QuickTime.

With these types of CAPTCHA it is interesting to note that the text the user sees on the screen and the sound for the audio CAPTCHA is totally different. The only reason we can see for this is that the companies do not wish to help "attackers of the system" break one system through the possible weakness of the other.

The other point noting is that these types of CAPTCHA require one or more media plug-ins to listen to the CAPTCHA. This is not helpful where you do not have administrative (or root) permissions to install a plug-in. Without a working sound card on the machine, this type of CAPTCHA is useless. W3C say that people that "work in noisy environments" are also affected, but why would people working in factories register to a website when then are supposed to be working and corporate computers are not supposed to be used in the workplace for non work related activates. Compared to the traditional CAPTCHA image, this type of CAPTCHA is even easier to break as software can filter certain frequencies of noise even better than humans (Santamarta 2008).

16.5 Restricted Accounts

W3C talk about the use of "limited user accounts" that is not linked to the content of the documentation. They mention that by restricting users to only send a certain amount of emails would help. In our opinion this type of restriction would cause even more registrations to occur that may be abused and by annoying the loyal customers, they would end up moving elsewhere.

16.5.1 Spam Filtering

A true fact from the article is that companies use methods to detect spam such as "hot words" and "Bayesian filtering". Unfortunately no system is perfect and spam will pass the filters to reach your inbox, forums posts and blogs. In some cases a system may think that a genuine message could be spam (a false negative) and you may not see the email that you expected.

On emails the use of image attachments containing text would pass the word filter whereas in a blog or forum post, pictures may not be permitted. The other spam prevention method is to block a number of messages from an IP address, however this is a useless method as spammers use multiple computers, proxies and victims of a worm or Trojan virus to spam others.

For the use of blogs and forums we find that it is useful to implement a CAPTCHA where new posts are instantly visible on the screen. Some administrators decide to implement a post system on blogs and comments of articles (such as BBC) where the comment needs approving by the administrators before it is on the screen. This is particularly useful where the website has an audience of all ages, genders and ethnicity that the content could discriminate against others. With the use of an "accept before display" option on each new post before it is shown to the public, we believe that in this circumstance the website should not use CAPTCHAs altogether.

Another method that can help with spam is the use of heuristics. This is already used with virus software to detect whether a new file or executable may be a virus whether or not it is in the list of virus definitions database on the computer or virus company website. In the W3C article an idea for heuristics is thought of to use the server load (activity level) to determine whether to use an extra authentication to validate a user. This could also be used to invalidate users as there may be a potential bot attack on the system.

These types of methods sound like a good idea but they also have a flaw. If many people are on holiday one day and go to the website, the server load will be higher and then everyone may have to use the extra authentication. This could push them away from using the website altogether. When W3C wished to promote the idea of this, shortly after that they manage to discourage the use of heuristics when using pattern matching algorithms they say that "this is not a good solution".

16.6 Secure Systems

In order to secure down a system W3C mention that "Microsoft and the Liberty Alliance are attempting to establish a federated network identity system". This is good for the large

company that have millions of customers, but is the cost benefit to the smaller companies worth the purchase of these expensive systems.

Single sign on systems (like Kerberos) are great for the end users as they may not need to use CAPTCHAs to post on the websites, but for the administrators would the costs outweigh the gain? Setting up Kerberos is no easy task and if the "ticket granting" server were to go down, a system would be needed to get this back up as quick as possible otherwise no user can authenticate with the system.

The use of biometrics sounds like the ultimate solution to any system so that we don't have to use CAPTCHAs, passwords, keys, passports or anything else to access whatever we want. If we all used biometrics to login to our email accounts, access our home or to start our car, would we be safe? All this information would have to be help somewhere and the scare factor is that our details could be mixed up or seen by people that may use them for any evil activity.

The last factor about biometrics is cost. If we all brought a finger print scanner to send our information over unsecure internet connections and be verified by a computer on the other end, how much would this cost to check that it all works. The information held about us on general websites should not be anywhere near the amount of information that is on government systems. Over the past year, millions of people's data was on two CDs that held parent's details and their financial information (such as bank account numbers) and this was lost in the post (BBC 2007).

16.7 Conclusion

Our conclusion to this paper is that what people see and hear around the internet should not be taken as the absolute truth. Large organisations that have built up a decent reputation, mislead the public and publish information based on opinion. W3C have done this on a number of occasions where there article lacks any reference of their "so called" findings.

In the paper we found a number of cases where statements were made by people working for the organisation with a lack of hard evidence to back themselves up. The article in question is nearly three years ago and systems have improved since then. People around the internet may

link others to the article and this is a bad idea where the article we find is out of date. Some of the information we gathered from our research proves that CAPTCHAs can be readable and easy to use as long as they are implemented correctly with the focus on a target audience.

17 Project Conclusion

This project has been an interesting project right from the beginning. The idea was to create some software and charm the public audience to visit the website and take part in a public survey.

This project required the use of research to find information about surveys to generate good quality questions to ask the public. With all the information from the studies, our questions were ready to go out to the public. Attracting the audience was the most difficult part. In order to make this easier, the website was accessible all day long. The questions had to be easy to understand and the feedback questionnaire was kept small to maximise the number of completed surveys.

By planning the entire process well, the number of feedback responses was just short of 100 responses and this was far higher than we expected. New terminology was learnt such as honeypots, hashcash and cleaning (in terms of feedback analysis) and this was used to a great advantage.

Our findings matched a few of our goals and this was used to write a paper that shows where our findings prove articles from other organisations when it comes to some of their opinions. Hopefully the industry will take note of this and use the information to develop lots of simpler CAPTCHAs to use in rotation that are easier to read for others and maybe even use ASCII CAPTCHAs as they are easier to read.

Where registration system are used, if companies can adopt the idea of a non CAPTCHA registration, this would be just as secure when using an email verification to prove the person really signed up or not. This way not as many spammers would access these systems as they would need to got through a two step process.

18 Self Appraisal

This project was carried out by one person and this person had to take on several roles. These roles included: project managing, report writing, interviewing the public audience, becoming a surveyor and an analyst to organise the results of a public study and use them to prove other research right or wrong.

In order to keep the project on schedule the student had to keep referring to the Gantt chart and make sure the critical path was managed well that the project was ahead of schedule to meet its deadline. One strategy the student had taken was to start the project two weeks earlier, missing out on the Easter break. This proved a wise choice as the amount of time needed to carry out a public survey was too important to do the job to its fullest.

By starting the project earlier, more time was given to come up with the questions for the feedback questionnaire. With the improved questions and being readying for a public survey even earlier the public audience would most likely be asked to participate in the survey from this project and lose their willingness to participate with other studies (in other projects) as the time went on.

This kind of thinking helped the student reach a wider audience and the number of feedback responses reached 95 responses by the last week of the project. This was far higher than the student first imagined and the expectation was at least 70. Whilst all of this was going on, one eye was kept looking at the Gantt chart at least once a week. With the extra work carried out, all these additions were added to the chart as the time went on.

The report has taken an eternity to write up and to make this worse, it was originally written first person. The student was informed to change this and this task took a fair deal of time as the student is not that confident when talking in third person.

Interviewing anyone is no easy task. For this project this was key to how the project would go. This student had to make sure that the correct audience was targeted to represent each year group in the department. Going up to lots of students at random and asking for their help with a pilot study was no easy task. Interviewing students and a member of staff meant that

the nervous student had to just get on with the task at hand. Rejection was not as easy thing and this had to make the student stronger. The information from this had helped the progression of the survey and how this would go when it was out to the public audience.

Once a student agreed, he had to keep the participant engaged and inform them of the system and how it worked. This had to be communicated well and he had to be prepared for all types of questions. It was vital that the pilot study went well as this was just as important as the public study.

The survey was split into two stages, firstly by creating the questions to use in the survey and making sure that feedback from the others did not affect the student as the questioned developed. Lots of time researching other work, surveys and similar systems meant that the student had good quality questions to use in the survey. After the pilot study was carried out, the student was carrying out the second stage of the survey. This was the public survey that many of the staff and students on the department managed to see and take part in.

The analysis of results had taken place near the end of the survey. Using the earlier work, the student's job was to make the data useful by collating the results and adding some meaning to the figures, linking up answers he found to past studies and maybe even proving other studies wrong where our answers revealed conflicting results. He thought of a great solution to "cleaning" our data and it proved fast reliable and efficient.

The development of the website had taken a long time from start to finish. The problem for the student was that he needed to make sure that it worked and displayed correctly across various OS platforms and a number of website browsers. This may all sound easy but is was one big headache. In order to make the website work across various website browsers he decided to detect a large number of website browsers with the use of a PHP function and this would be used to override the CSS of a main style sheet. This made it easier to alter styles in another browser where the fix (or the correction) would be in another style sheet to display the website correctly between the web browsers.

The student was quite confident with the scripting language because he decided to learn it a few months before the project started and it was good of him to do so. He learnt a lot during this project including:

- Pagination: to filter a number of records per page.
- Creating a PHP image that was used as the CAPTCHA.
- Using get / request to grab data from the address bar for a query to echo a limited number of results onto one page (see fbresult.php).

He then had to deal with the headache of managing the website on two hosting companies. The file structure was different at University and this meant that the "included" files needed to be configured slightly different as many files were called from different directories in the website structure.

When the website was first transferred to the University system there was another headache that was urgent. This was the problem with SendMail where no emails were sent as the mail server was bouncing all the emails. The solution ended up being the email headers. This was configured in the PHP file and emails were being sent out.

Half way through the project the backup website was down so there was only one single point of failure. After communications from the student and the hosting company, a new hosting company was used along with a new domain name. The backup website was up after a week and there was no longer a single point of failure.

The student worked wonders with the website. Nearly 100 participants saw the website and it looked like the Intranet (before someone changed the website, during the final year projects). That put aside it was functional, easy to use and the newest version of the help section was cleverly coded.

Apart from appraising the student for all his work, thanks must be given to the supervisor of the project student (without gaining any brownie points). The supervisor was never a person to be afraid of. He was always helpful throughout the project and has a brain full of knowledge that we want. Throughout the project he has communicated **brilliantly** with the project student suggesting improvements, direction and he even allowed extra meeting time for the student to answer some new questions that needed fast replies.

19 References

Ahn, L.V. Blum, M. and Langford, J. (2004). Telling Humans and Computers Apart Automatically. Communications of the ACM. 47 (2), p57-60.

Atwood, J. (2006). CAPTCHA Effectiveness. Available: http://www.codinghorror.com/blog/archives/000712.html. Last accessed 25 October 2006.

Atwood, J. (2007). Making a Good CAPTCHA. Available: http://www.buzzwordcompliant.net/index.php/2007/04/21/making-a-good-captcha/. Last accessed 21 April 2007.

Back, E. (2007). WP Hashcash Plugin for Spam. Available: http://wordpress-plugins.feifei.us/hashcash/. Last accessed 28 May 2007.

Batchelder. (2008). Honeypots: better than CAPTCHAs?. Available: http://people.w3.org/~cmsmcq/blog/?p=23. Last accessed 17 January 2008.

BBC. (2007). UK's families put on fraud alert. Available: http://news.bbc.co.uk/1/hi/uk_politics/7103566.stm. Last accessed 20 November 2007.

Bergin, J. (2008). Anonymous Feedback. Available: http://pclc.pace.edu/~bergin/patterns/FeedbackPatterns.html#anonymousfeedback. Last accessed May 2008.

Claburn, T. (2008). Yahoo's CAPTCHA Security Reportedly Broken. Available: http://www.informationweek.com/news/internet/webdev/showArticle.jhtml?articleID=205900620. Last accessed 17 January 2008.

Coates, A.L Fateman, R.J Baird H.S, (2001). Pessimal print: a reverse Turing test. Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference. 1 (1), p1154.

Connor, ML. (2007). 100 Line simple CAPTCHA JSP (Again). Available: http://www.jroller.com/mlconnor/. Last accessed 23 October 2007.

CVent. (2008). Ten Tips For Creating Effective Online Surveys. Available: http://www.cvent.com/resources/ten-tips.shtml. Last accessed 8 February 2008.

Doctorow, C. (2004). Solving and creating CAPTCHAs with free porn. Available: http://www.boingboing.net/2004/01/27/solving-and-creating.html. Last accessed 27 Jan 2004.

Edwards, J. (2008). Beyond CAPTCHA: No Bots Allowed!. Available: http://www.sitepoint.com/article/captcha-problems-alternatives. Last accessed 31 May 2008.

Gearing, A. (2008). Incredi Signup. Available: http://www.incredi.net/signup/details.php. Last accessed 31 May 2008.

Groves, R. M., R. B. Cialdini, and M. P. Couper (1992). Understanding the Decision to Participate in a Survey. Public Opinion Quarterly 56: 475-95.

Hodges, A. (2001). Alan Turing and the Turing Test. Available: http://www.turing.org.uk/publications/testbook.html. Last accessed 03 June 2008.

Iarossi , G (2006). The Power of Survey Design. Herndon, VA, USA: Washington DC. p147-158.

Iarossi , G (2006). The Power of Survey Design. Herndon, VA, USA: Washington DC. p195-218.

Maddaloni, M. (2007). Bad CAPTCHA. Available: http://www.thehotiron.com/index.php/site/bad_captcha/. Last accessed 18 April 2007.

Oleg. (2007). Is CAPTCHA Useful?. Available: http://www.freearticlesarchive.com/article/Is_CAPTCHA_Useful_/158624/0/. Last accessed 24 September 2007.

PHP Pro. (2006). tppCaptcha. Available: http://thephppro.com/products/captcha/. Last accessed 10 September 2006.

Rapidshare. (2008). Download page. Available: http://rs25.rapidshare.com/files/98942864/Legendary_The_Box_Trailer.rar. Last accessed 31 May 2008.

Rickert. (2008). Email blocking -- our methods. Available: http://www.cs.niu.edu/~rickert/net/email/rej-block.html. Last accessed May 2008.

Leyden, J. (2008). Russian serfs paid $3 a day to break CAPTCHAs. Available: http://www.theregister.co.uk/2008/03/14/captcha_serfs/. Last accessed 14 March 2008.

Santamarta, R. (2008). Breaking Gmail's Audio Captcha.. Available: http://blog.wintercore.com/?p=11. Last accessed 5 March 2008.

University, Carnegie Mellon. (2000). CAPTCHA: Telling Humans and Computers Apart Automatically. Available: http://www.captcha.net/. Last accessed 2007.

University of Portsmouth. (2008). ECE forums. Available: https://Intranet.ee.port.ac.uk/Forums/. Last accessed May 2008.

W3C. (2005). Inaccessibility of CAPTCHA. Available: http://www.w3.org/TR/turingtest/. Last accessed Nov 2005.

W3Schools. (2008). Browser Statistics. Available: http://www.w3schools.com/browsers/browsers_stats.asp. Last accessed April 2008.

Walker , J. (2006). HotBits: Genuine Random Numbers. Available: https://www.fourmilab.ch/hotbits/. Last accessed 01 September 2006.

Wikipedia. (2004). CAPTCHA. Available: http://en.wikipedia.org/wiki/Captcha. Last accessed 29 May 2008.

Zarar (aka Arsenalist). (2008). Worst CAPTCHA Ever. Available: http://depressedprogrammer.wordpress.com/2008/04/20/worst-captcha-ever/. Last accessed 20 April 2008.

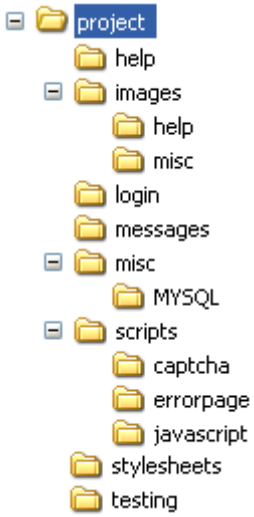
20 Appendices

The appendix contains the following:

- Feedback Results
- Probabilities for the CAPTCHA answers
- Source Code.

Findings of the research - Files	
Feedback Results	- Staff vs Student - Based on Q1 Answer
Probabilities of Answers	- Totals per Operation - Total of each outcome - Ordered total by popularity

Website File Structure



Database setup:

Place in highest dir as it needs to be called "project".

Configure the connection script:

/project/scripts/mysql_connect_db01.php

Change the password details.

Copy MSQL DB from:

/project/misc/MYSQL/

Paste entire content to create DB, tables and content.