G52GRP
Interim Group Report

WebPAS

Ryan Chapman – rxc01u
Thomas Dunford – txd01u
Daniel Morris – dxm01u
Raffi Tamizian – rht01u
Benjamin Yeomans – bmy01u
Xiaotian Zhou – xxz12u

GP12-axc

Supervisor:
Andy Crabtree

Date: 07/12/2012

Problem Description

The aim of this project is to develop a web-based peer assessment system for students to assess the group work of other members in a piece of group work.

Peer assessment is a way students can mark the members of their group based on their contribution and work ethic towards the group project. The marks can then be distributed accordingly. Currently peer assessment is conducted via a paper-based system, the main issue being the time it takes to process them all after they are completed. After collecting the sheets, currently the module convenor must physically go through each sheet of paper and physically input the results of the peer assessments into an algorithm. This is likely to take a long time, since each student completes a peer assessment for each team member, all of which needs to be input.

Our proposed solution to this problem will be an online website designed to make peer assessments easy, fast and informative. It will also provide a visualized interface for the supervisor and module convener to manage the students-database and marking results. This system will implement one specific marking algorithm and offer the ability to change the mark for later stages. The project will be built on the implementation of a MySQL database and the PHP website. This system will be accessible with given IDs and passwords, using any browser from a remote client.

Background and Research

Research into existing web peer assessments showed that there are currently a few open-source systems that currently exist. Looking further into them showed they're of a different variation of peer assessment in comparison to the requirement specification we have been given.

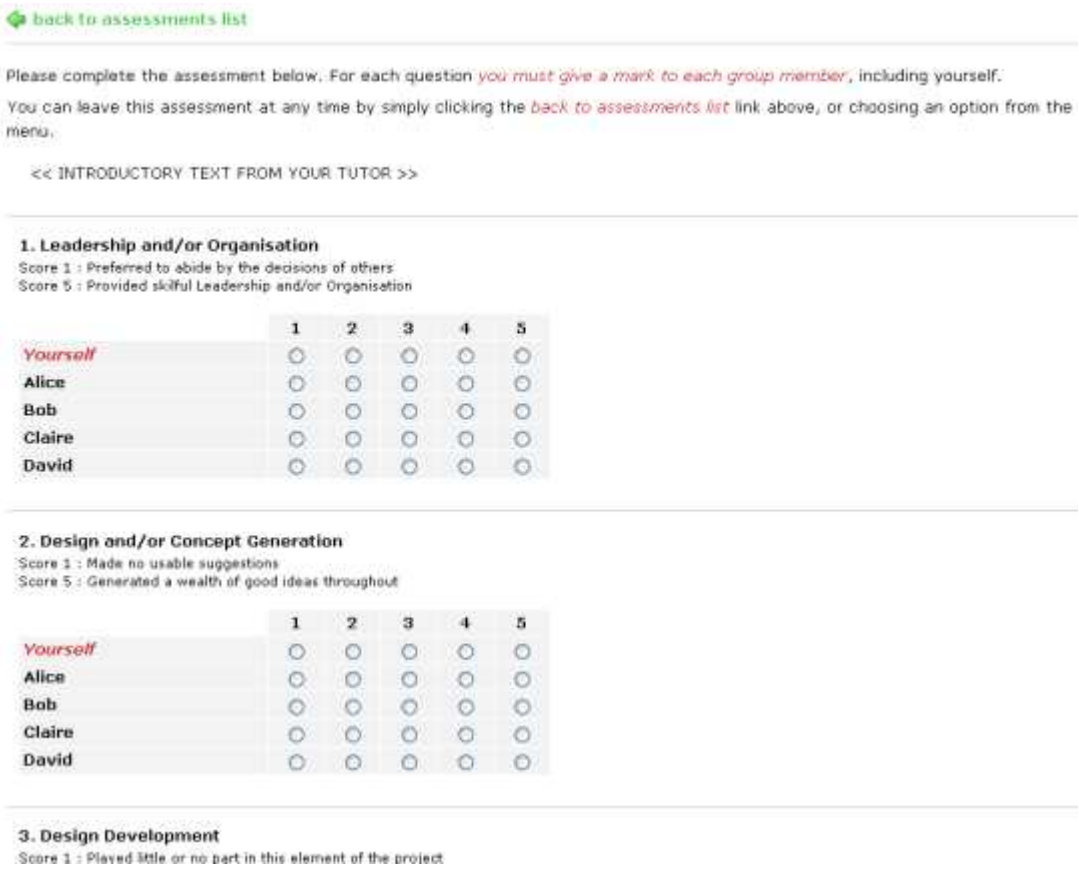
Existing Systems

Once we had decided on the project, we decided to investigate whether the idea of online/computer based peer assessment systems had been implemented elsewhere, and to what extent these looked appealing and effective.

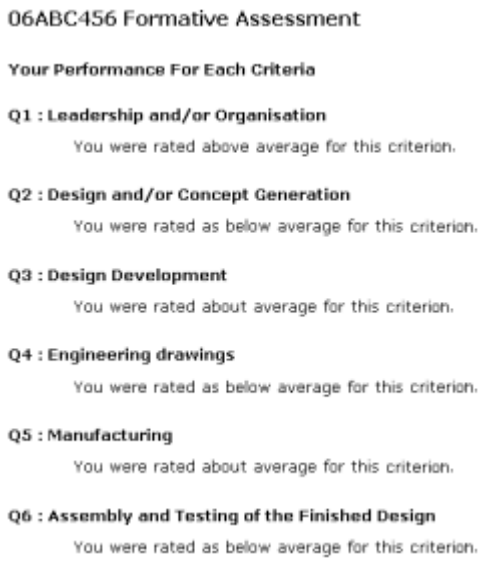
Loughborough University WebPA

http://webpaproject.lboro.ac.uk/

On quick inspection it can be seen that this is the case, with the notion of generic peer assessment systems existing on the web to download and use, the most advanced version found was Loughborough University's. There are many successes within this, but they tend to cater towards the graph presentation end of the process and upon closer inspection the system has some key design flaws which lead to the system seeming less polished. One of the key flaws which make the assessment process seem more of a chore, and reducing the likelihood of gradual completion, is the lack of split between member assessments. All group member assessments are completed upon one page, in a radio button style entry system, which in our opinion makes the process more daunting and less of a logical iterative decision making process, below shows this screen:



The above decision by Loughborough meant that as a convenor it's more difficult to monitor the progress of assessments, as due to the above screen being leave-able at any time, as opposed to an individual set of assessments, it is nigh on impossible to, at a glance, find which individual assessments have been completed by a student.. We felt another trick was missed in the presentation/review of results for the student, either at the end of a completed assessment, or when your feedback is to be viewed at the end:



06ABC456 Formative Assessment

Your Performance For Each Criteria

Q1 : Leadership and/or Organisation

You were rated above average for this criterion.

Q2 : Design and/or Concept Generation

You were rated as below average for this criterion.

Q3 : Design Development

You were rated about average for this criterion.

Q4 : Engineering drawings

You were rated as below average for this criterion.

Q5 : Manufacturing

You were rated about average for this criterion.

Q6 : Assembly and Testing of the Finished Design

You were rated as below average for this criterion.

The above screen which is received as feedback, makes no use of colour, so again at a glance nothing can be gauged from it, this would be the same when assessments are submitted, and we felt a colour based system would allow for a better, more logical, judgement on either how harsh or generous you have been or how well you have done yourself.

Dundee University Self and Peer Assessment Online

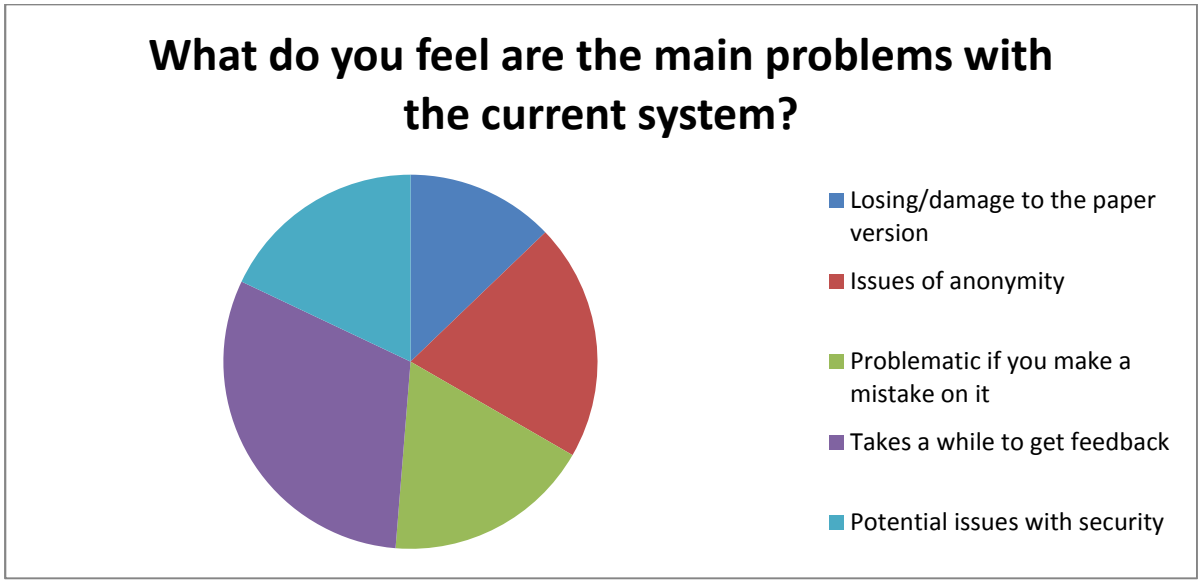
Dundee University also have a system available to be viewed online, but in comparison to the above it is far less polished in terms of design:
http://www.personal.dundee.ac.uk/cgi-bin/cgiwrap/rparsons/feedback/cc01_zz_choose.pl
There is no styling used upon the above page, and as simple as this is, it makes the system seem far less professional than it could, and again the issue of one entry page being used for the whole creation of the assessment doesn't echo the iterative process of creation or completion.

Conclusion

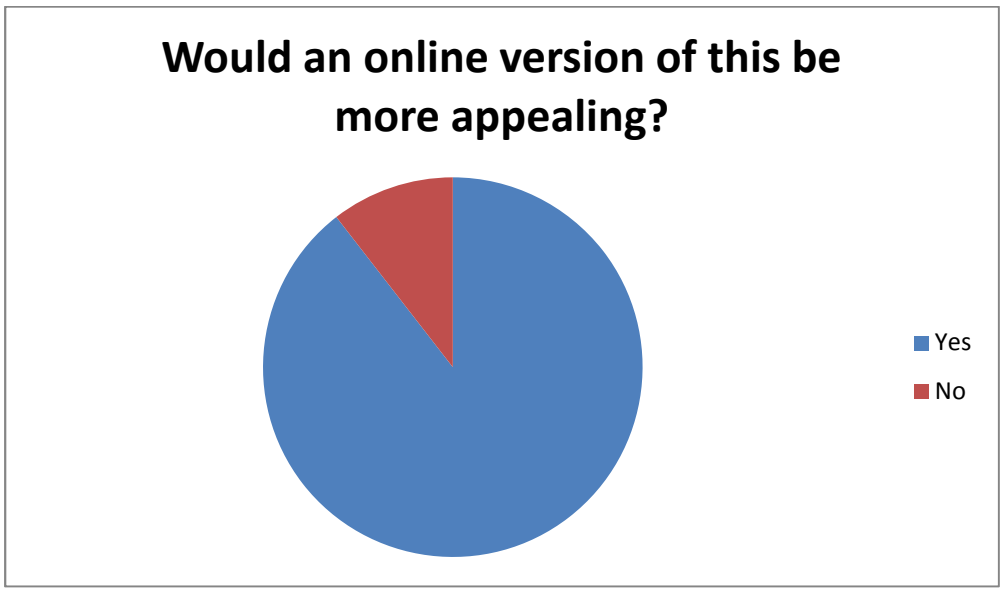
The key theme to be drawn from the research we made into existing systems is the need for iteration throughout and the need for atomicity as opposed to collation of entries upon one page or a huge creation section. This is not user friendly and these processes are not a one-step decision so it shouldn't be modelled this way.

User Research

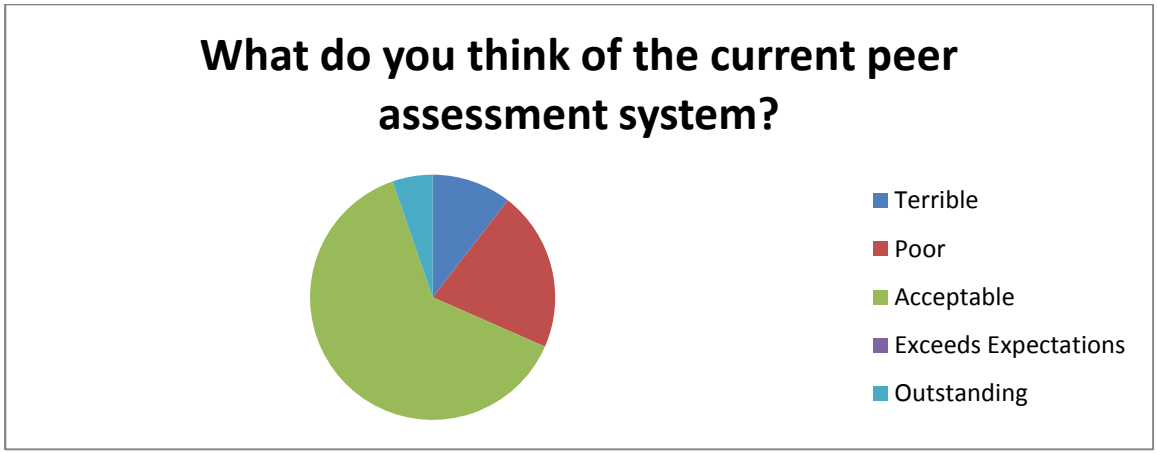
Before starting the development of any of the software, we conducted a short survey to gauge the response to the current paper-based Peer Assessment system. We used a website that allows you to host your own surveys on to create and publicise our survey. The following graphs are representations of the data we received. The survey was taken by 19 Computer Science students.



For this question in the first survey you had the capability of choosing 'other' and specifiying what that problem would have been, but it was never chosen. This graph is quite fairly split in the given sections, with the lack of quick feedback proving to be the largest issue the students had – this reflects our initial impression that the turnaround time on peer assessment was slow. With a web-based system, this would be largely negated.



The results of the second graph are fairly self-explanatory; it shows that an online version of a peer assessment would be more appealing for a large majority.



The third graph shows that there is room for improvement with the current system. Although most consider it 'acceptable', they wouldn't have had anything to compare it to.

Overall the results of the survey show an online peer assessment could make for a better system than the one currently in place.

Requirements specification for the system to be built

A set of basic requirements was provided by our supervisor, on which we expanded and embellished them to fully suit the needs of the project.

- Must be accessible for users through a website:
  - Users must have a unique login for the website involving their student ID and password
  - Website must be able to differentiate between Module Convenors and Students
  - Website must be secure and only accessible for people with valid login
  - Website should have different controls and visuals for Module Convenors and Students

- Module Convenors must have access to controls students do not:
  - Module convenors must be able to able to add modules, specifying the name and the weighting of the new module
  - Module convenors must be able to import a list of students taking that module through a .csv file
  - Module convenors must be able to add students to the module manually, specifying name and student ID
  - Module convenors must be able to view a list of all students on that module
  - Module convenors must be able to create groups in that module, specifying the name of that group
  - Module convenors must be able to add students to an existing group
  - Module convenors must be able to view a list of all groups in that module
  - Module convenors must be able to view a list of all students in a group in a module
  - Module convenors must be able to delete students from an existing group

  - Module convenors must be able to see who has not completed their peer assessments in the list of groups
  - Module convenors must be able to manually enter a group mark for a group
  - Module convenors must be able to manually adjust the weightings for the module
  - Module convenors must be able to view and edit individual peer assessments completed by students
  - Website should automatically recalculate individual marks across a group based on a new scaling of weighted assessment factors
  - Module convenors must be able to export the finalised list of students to a .csv file, specifying student name, ID, group mark and peer assessed mark
  - Module convenors must be able to view graphs of the module, including graphs of individual marks and group marks, displaying an average bar across each
  - Module convenors must be able to delete all peer assessments within a module
  - Module convenors must be able to allow students to re-do a peer assessment if the convenor believes they have not completed it honestly

- Students must have less access than a module convenor:
  - Students must be able to view a list of all current modules that have open peer assessments that have not been completed
  - Students must not be able to see what other students have said about them individually
  - Students must be able to have access to their peer assessed mark when everyone in their group has completed the peer assessments and the module convenor has reviewed it so it is acceptable
  - Students must be able to access an open module and complete peer assessments for each of their group members:
    - Students must be able to complete some peer assessments in their group and finish the rest later
    - Students must not be able to partially complete a peer assessment - they must complete it or they must lose all data in that peer assessment
    - Students must be able to add a comment at the end of a peer assessment limited to 1000 characters, which must be completed else the results will not be saved.
    - Students must be able to confirm the assessment is complete before submitting it to the system.
    - Students must be able to add a score in the range of:
      - None
      - Lacking
      - Adequate
      - Good
      - Excellent

    Students must be able to add a score for each factor:
      - Research and Information Gathering
      - Creative Input
      - Cooperation within group
      - Communication within group
      - Quality of individual contributions
      - Attendance at meetings

Initial design of the proposed system and its user interface

User-Centred Design, also known as human-centred design process, was an approach we used as the model for how to do our project. The decision upon an iterative process within the design stages meant the process of translating the first set of requirements into reality was one that drew out which were really needed and which were superficial, whilst understanding further any other requirements needed. Since the software is largely to do with users, and would involve being accessed by a number of people, it had to be simple and easy to use. For this, UCD was suitable and we focussed our goals around the users and the tasks completed during the process of completion. Using Iterative design as a template, we allowed the designs to be modified after initial testing, to reflect the usability changes, with an aim of eradicating the weaknesses from each stage. We also agreed with UCD's principles in that we needed to focus on ensuring users could very easily see the product and understand how to use it intuitively without having anyone explain it to them. To ensure that we have achieved this, once a working prototype is developed then it will be tested with actual users.

The initial design was produced and perfected using wireframe software which allowed for both the look and functionality of the proposed product to be modelled. Software was used as opposed rather than pen and paper designs to allow for continually improving lo-fi prototyping and ease of execution, essentially meaning new ideas and changes could be implemented and modelled almost instantly. A crisp white and blue colour scheme is used throughout, disrupted by only the traffic light style marking system; this is in keep with the university logo that its PA forms are generated from. This wireframe software allowed the file to also be generated within a web browser and worked through.

The main changes throughout the iterative process were made not due to the design itself, but through the actual understanding of the steps a convenor would make, and what screens should be viewable from the beginning. However design decisions were made throughout, with group opinions dictating the final version. Below is a step by step set of screens of what the desired software will resemble.

Student Side

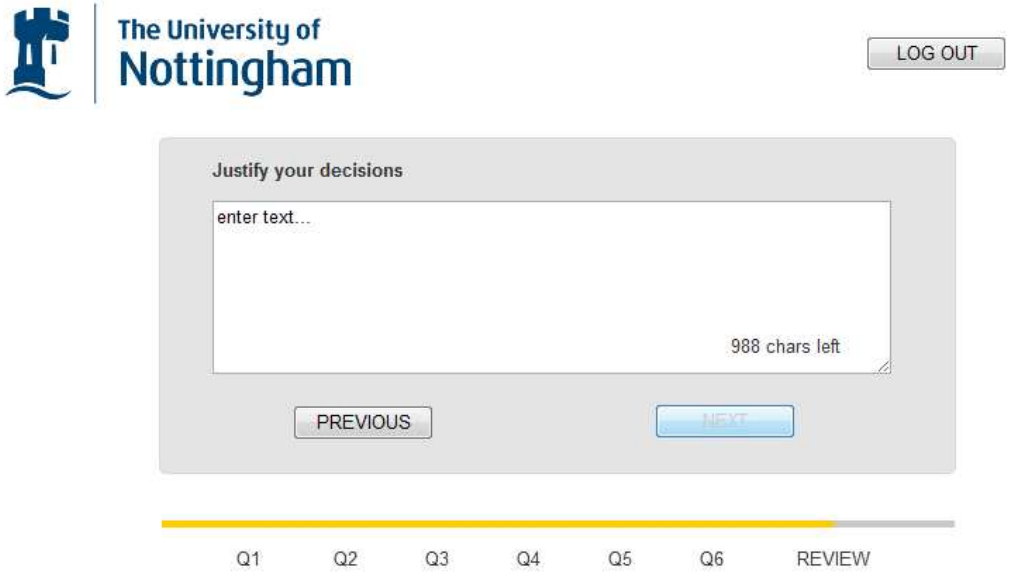
Group member select screen

Above is the group member select screen, which follows the user sign in screen (which consists of the usual username and password entries) and module select screen which dictates which set of group members a student will be completing forms for. A completed tickbox is included here to clearly be able to tell which PA's are left to be done.

First version of the entry screen

Second version of the entry screen

The above images show two versions of the typical entry form screen, with a 0-4 scaling system being used, for each question to be asked. The first image shows what was proposed initially, with it being decided that a radio button entry would be more logical and familiar. Below that is a process bar which shows the user how long of the assessment is left to complete, the bar colour depends upon what marks have been given so far, reds indicating a low set of marks and vice versa for green.



Justification /comment screen

Above is the required justification screen which prompts the user to justify their ratings given, if this isn't completed then no progress can be made, and also if the justifications aren't adequate or believable enough then a mark may be changed at a later date in a scheduled review section with the student's supervisor.

Review entries screen

Logout dialog box

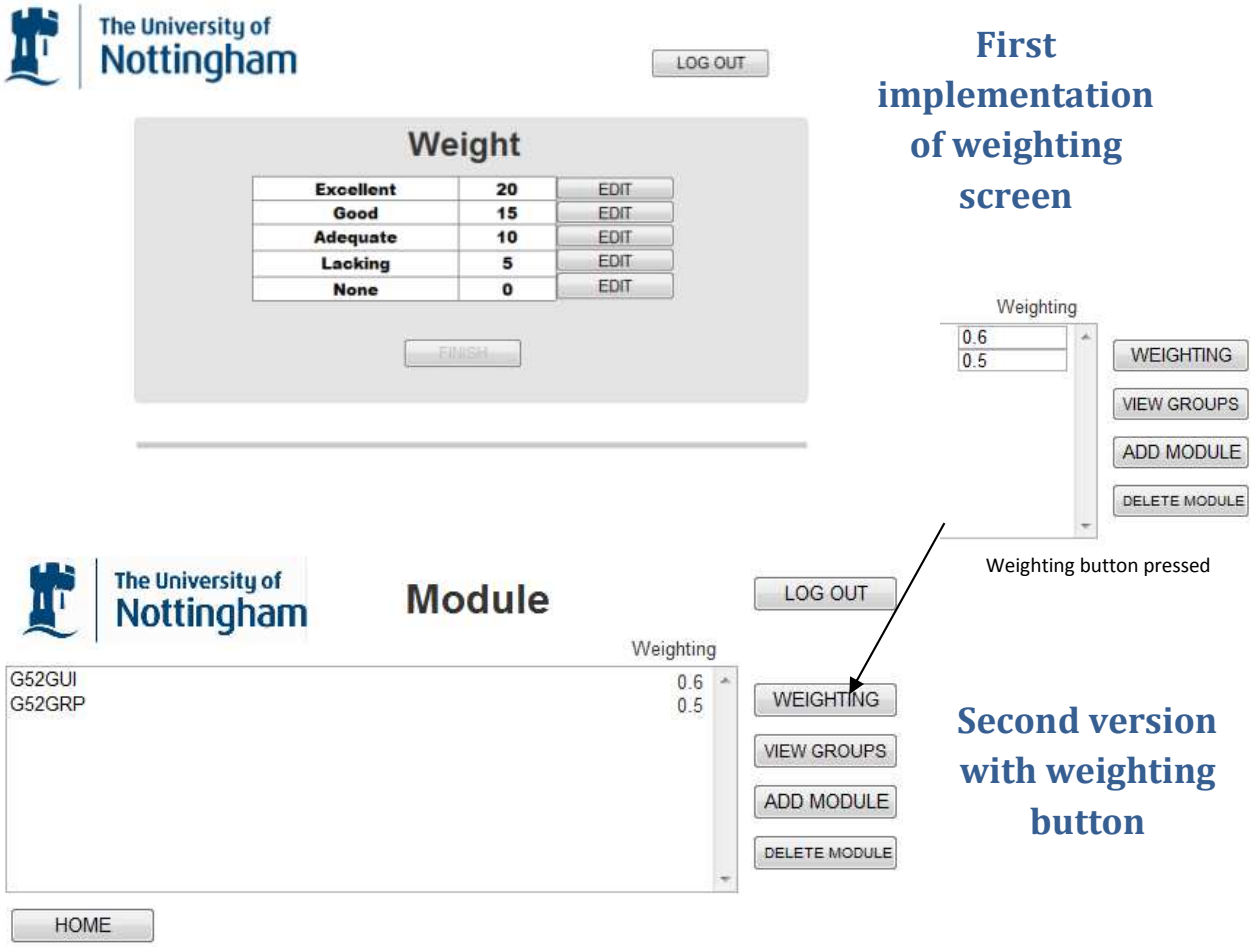
Above is the final review screen which allows for marks to be changed before they are finally submitted, this is done by simply pressing the re-enter button for the desired question. A traffic light style system is used to show what marks were given, this adds to the clarity of the review page, as you can see clearly whether you have been particularly harsh or generous is one section. Also the logout button is used throughout which when pressed would bring up the above dialogue box, because if a form isn't fully completed marks are discarded.

Convener Side

First version of convenor home screen

Second version of convenor home screen

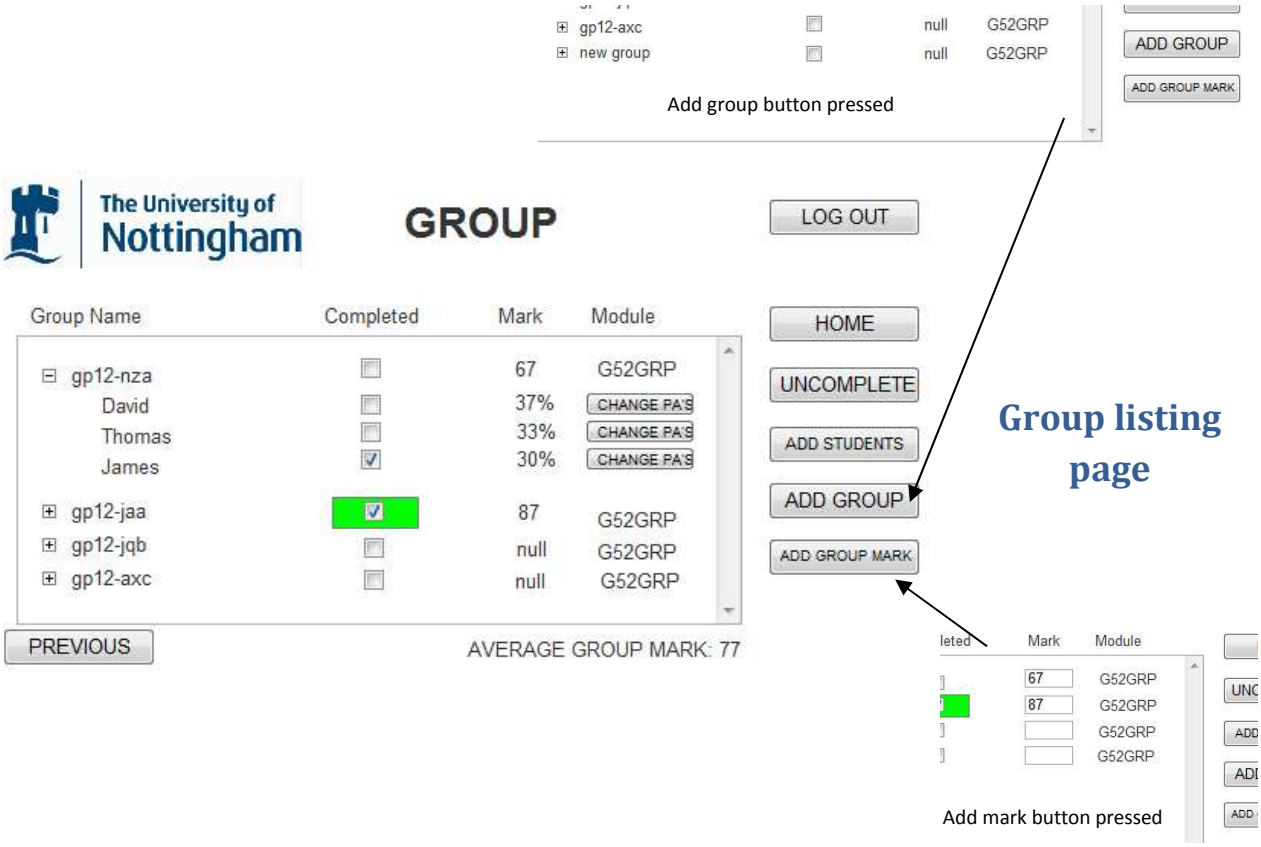
Above shows one of the key iterations made from prototyping the wireframes, understanding that only modules should be accessible initially for the convenor for a peer assessment system to be a logical process, this then makes the addition of groups to modules and students to groups much easier to implement. Above shows the page the convenor will see upon entering their sign in details. The difference between the first version and the second version echoes this.



First implementation of weighting screen

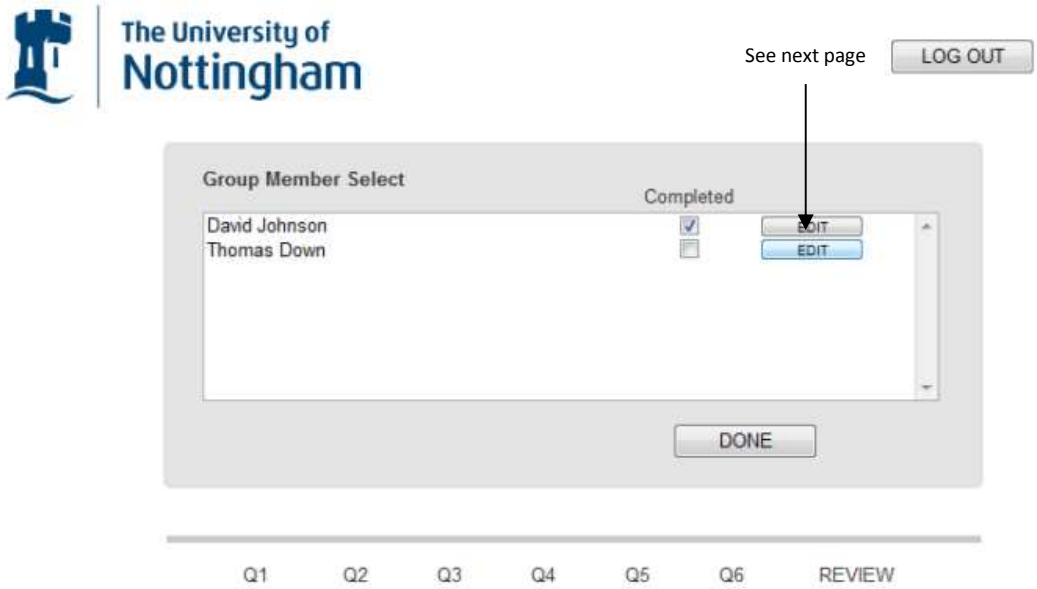
Second version with weighting button

At the convenor greet screen, after sign-in, if the modules path is chosen the above screen will be what's shown, if the weighting button is pressed (as shown above) the weightings of the current modules will become editable, when pressed again it will revert. The other buttons either add a new module to the list as would be expected, and vice versa for the delete module button. When a particular module has been selected it will be highlighted, if then view groups is pressed, the groups for that module are shown. Above that shows the initial weighting screen design, which was decided against due to it adding more pages, and the decision being made that a weighting button is more user friendly and logical.



Above shows the group listing page, which is appropriate to the particular module that has been chosen for viewing. It makes use of a drop down style table, which essentially is when the group name is pressed, the members with in are shown, with the group mark split accordingly as well as the tickbox theme continued to show who has completed their PA's. If a whole group has completed theirs then the completed field is coloured green. Some of the functionality of this page is also listed, with the add group button adding a new empty group, and the group mark button making the mark field editable. Each student within that group may have completed a set of Pa's and as required these need to be changeable by the convener at a later date if needed. So each student's assessments can be accessed by pressing the change Pa's button which shows the user the below screen, with edit buttons which allows for individual editing of answers.

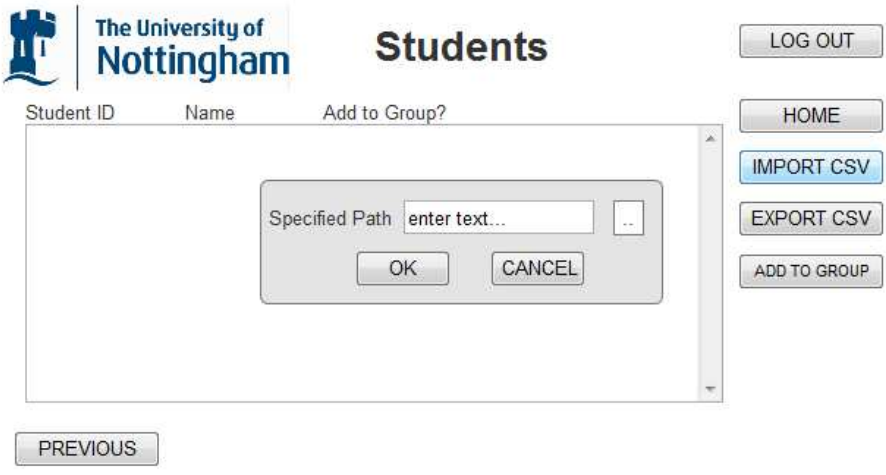
Below shows the screens if the change pa's button is pressed:



Change PA Group member select page

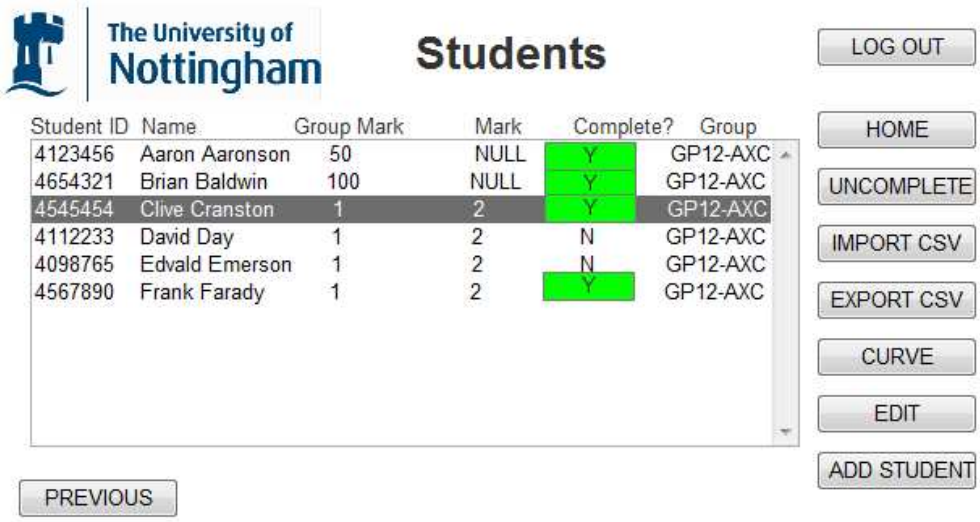
Edit individual assessment page

The above images show the process of a convenor editing a set of assessments, the topmost image showing the list of group members, and the above showing the edit page which will be shown for the user that that edit button press coincides with.

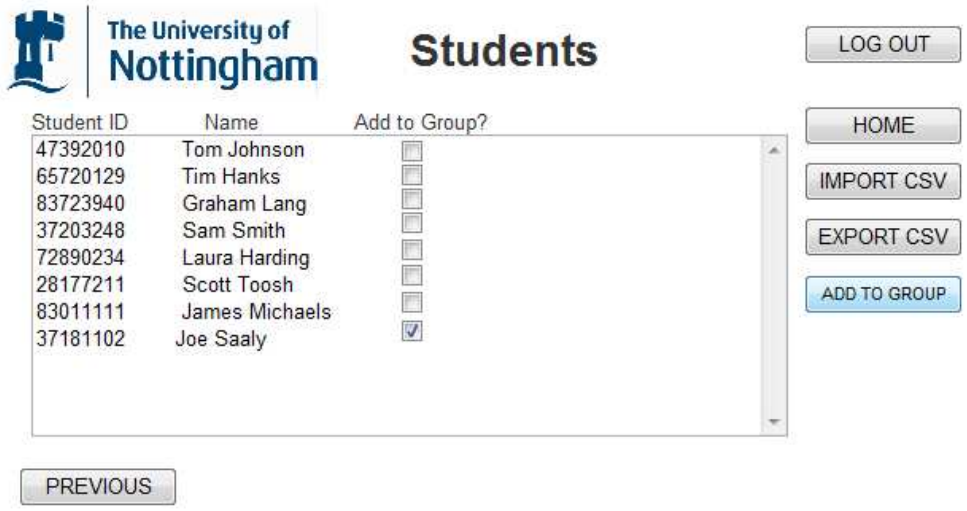


Add students page

The above image shows what would display if on the previous page the add students button was pressed. Initially an empty list is shown to which can be filled by importing a csv file, when “IMPORT CSV” is pressed the included dialog box above would be shown, which asks for the csv’s specified path, and the usual confirm or cancel buttons being displayed afterwards. A csv file would then have to be navigated to as expected and if successful the list of students contained within it would be displayed, as shown on the next page



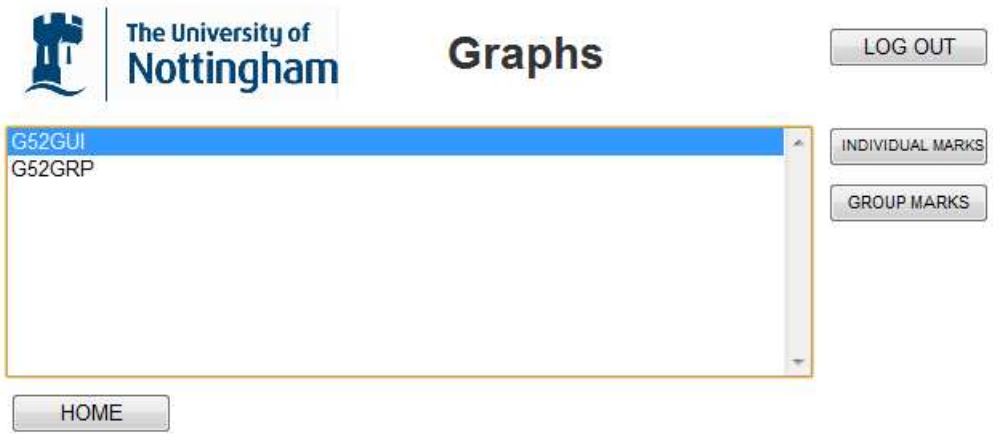
First version - Add students page



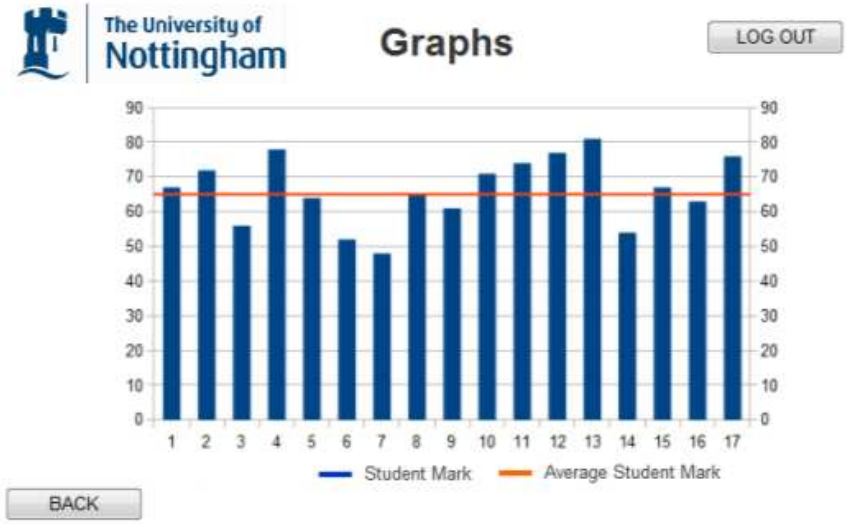
Second version - Add students page

Above shows the many changes made in result of the gradual understanding of the most logical set of steps for the user to take, i.e students only being a page for adding students to a group within a module. As a result far fewer fields are required to be shown, making the screen much simpler and easy to comprehend. In the final frame to the right, the imported list of students, and their ID’s are shown, this should be read in automatically, with a name showing only if it isn’t in a group already/or this will be prompted afterwards when adding. If a checkbox is pressed and the “add to group” button is then pressed, then that particular student/set of students will be added to the group which has been navigated through previously and the groups page will be redirected to.

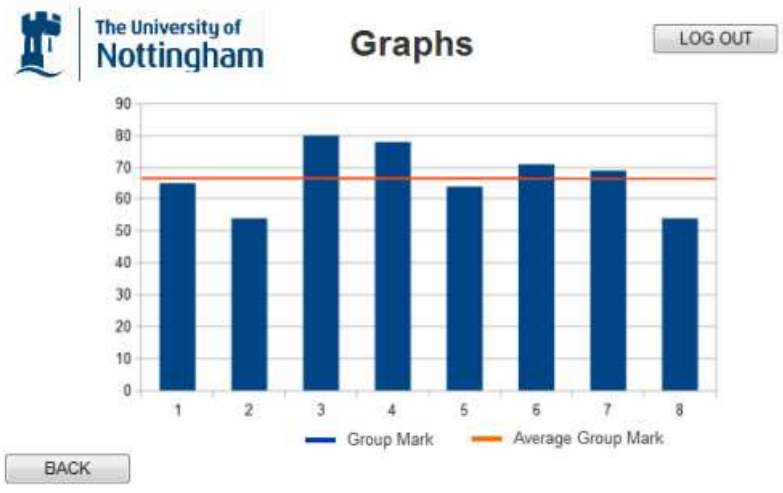
Graph Side



Graphs home screen



Individual marks graph screen



Module marks graph screen

Another of the features available to a convenor upon sign-in is to view two maps for each module chosen, these will consist of either individual marks, and an average of these plotted upon a simple bar graph, or a group marks graph which again consists of an average but instead averaged out across the group marks allocated for that module. The colour scheme is continued throughout as expected. The final graphs used may vary to the above in type and content, but this is the initial design.

Key Implementation Decisions

Operating Systems

For development, we will be making the project in Windows 7 and Mac OS, with Linux Shell being used on the servers for the Repository and for the MySQL. We decided on these since they are commonly used and what the group is comfortable with working with.

Programming Languages

- **Java**: for the weighting algorithm. We chose Java since it was a procedural language we were comfortable with using, and provided the tools necessary to create the algorithm.
- **MySQL**: for the database. We decided on using MySQL since we had prior experience in using it, and since it provided the ability to create relational databases rather than flat files, meaning that the database would be much more ordered and efficient. MySQL is a popular choice since it is flexible and can be run on many platforms.
- **JavaScript, HTML, PHP, CSS**: for the website. The general visual page of the WebPAS site is programed with HTML language. It is embedded with PHP code to make a connection to the database. For database type choices, we select MySQL because it is the worlds' most popular database system and can be run in many platforms, which provide lots of flexibility to program. Due to the requirement to enables programmatic access to computational objects within a host environment and dynamic content than static HTML (e.g. expanding a tree with a given ID, adding an event listener), JavaScript language is used to realize these functions. Cascading Style Sheets (CSS) is also applied to further define the appearance and layout of text and other material of the website. CSS enable the separation of document content from document presentation, which improve content accessibility and provide more flexibility and convenient in the specification of presentation characteristics.

Software

- **Axure RP Pro:** for creating the wireframes. After viewing different wireframe creation tools, we decided on Axure since it allowed us to create professional looking wireframes. The benefits of using this included allowing comments on the wireframes and allowing the website to be simulated using basic links to other pages. It was ideal for creating high fidelity prototypes to demonstrate different states of complex interactions to developers and business owners.
- **Eclipse, Notepad++, MySQL Workbench:** for using Java, HTML/PHP and MySQL. These IDE's allowed us to write in the languages we chose while providing valuable assistance with refactoring, formatting and testing.
- **MAMP (MAC, Apache, MySQL, PHP):** for team members using Macs. This was a simple piece of software that bundled the tools for creating websites so that the team members using Macs could work on the project easier.
- **TortoiseSVN:** for the repository. Since we were provided a repository on the School's CS servers, we decided on using TortoiseSVN as a visual tool for applying source control to our project. It was chosen for its simplicity and its ease of use. In comparison to typing in the subversion command lines, it is much more efficient.

Implementation Steps/Prototyping

Group Website

gp12-axc

Development Team
Admins
Henrik Nilsson
Benjamin Michael Yeomans
Happy Crew
Andy Crabtree
Ryan Chapman
Raffi Haig Tamizian
Daniel Morris
Thomas Dunford
Xiaotian Zhou

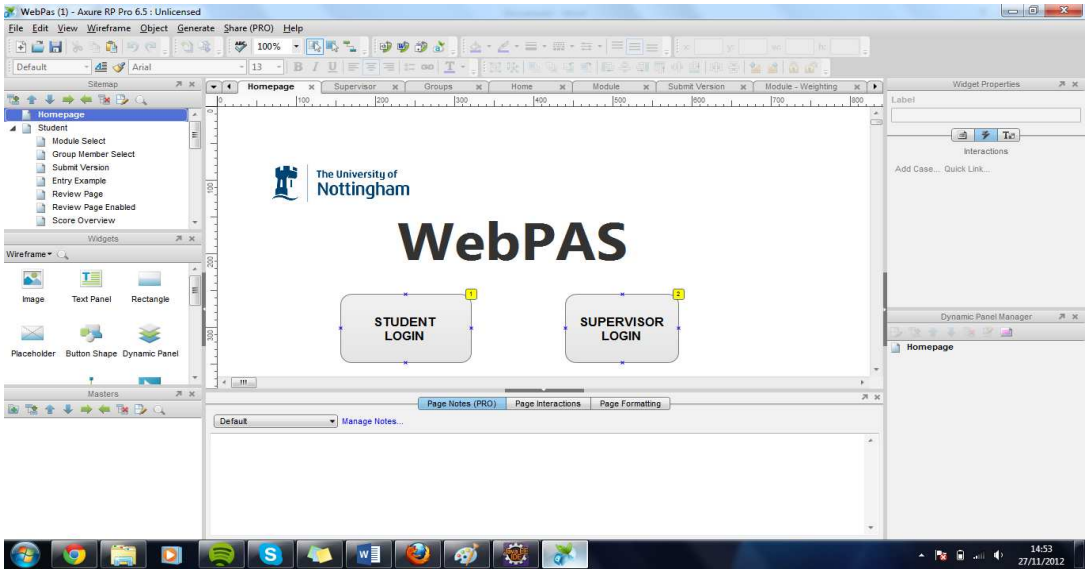
WebPas.

Web-Based Peer Assessment System- allows a separate log in for module conveners and students. Module conveners allowed to create groups and allocate group marks which the individual marks are then automatically calculated from depending on the peer assessments. Students can log in and fill out peer assessments for several modules and for several students in their group.

Meetings: Tuesday at **10AM**, Andy Crabtree's office. (3rd Floor CS) **Group: Tuesday 2PM**, Labs. (Pokémon Lab)

When we started the project, we were provided a project site where we could centralise our work – it was tied to our repository and we started utilising it immediately by recording the minutes and agendas of each group meeting. We set up a description of the project on the home page to give it a more professional look and inform visitors of our aims.

Wireframe of peer assessment website

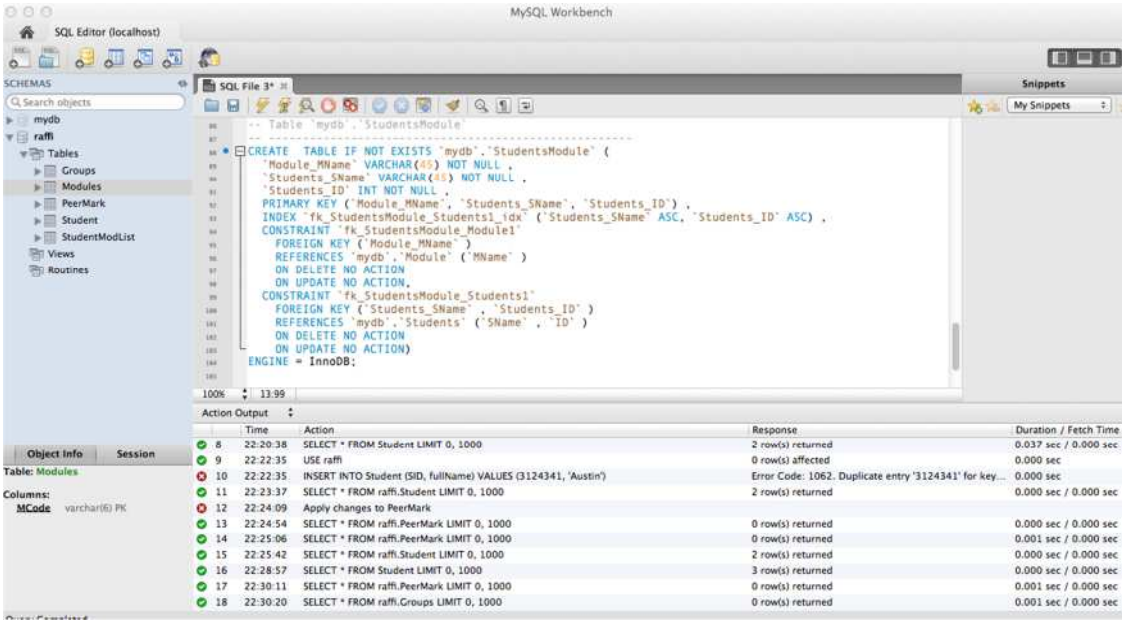


We started off by designing how the project would look. Axure RP Pro 6.5, a wireframe design software, was used to implement this. We designed each page of the website as a wireframe using Axure -these were displayed as a click-through prototype, focussed on the types of information displayed, the range of functions available and the relative priorities of the information and functions. We completed these first as part of the design of the project, since we could work from them when creating the pages for the website. A more thorough explanation of the wireframes can be found in "Initial design of the proposed system and its user interface"

Algorithm implemented in Java

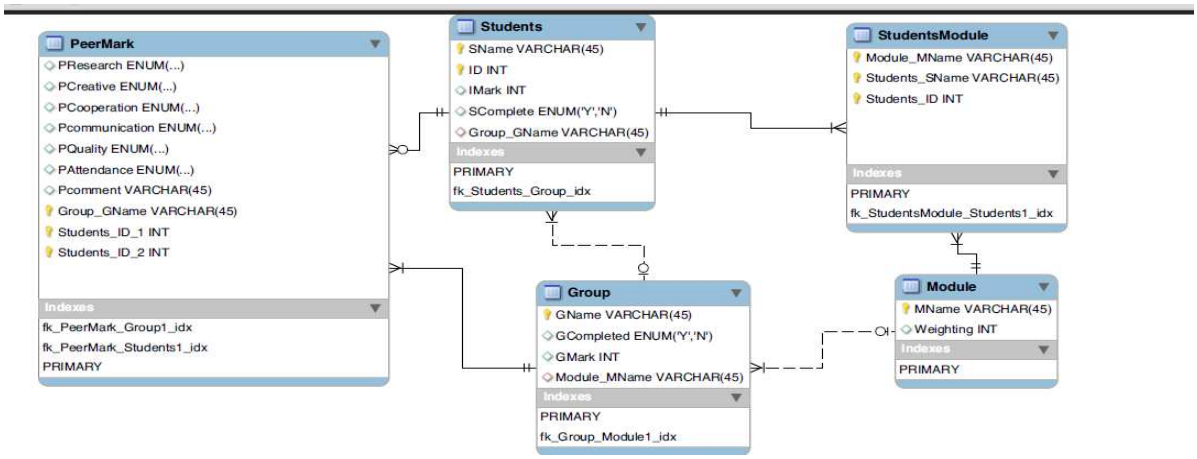
The marking program will be written in JAVA under the Eclipse integrated development environment. It will calculate the students' final mark using a specific algorithm, based on the group mark given by the supervisor and personal marks from team members. Each student needs to be assessed for six criteria with the mark from 0 to 5, so a two-dimensional array will be chosen to store the data. The supervisor will just need to input the weight and group mark and then this algorithm will automatically calculate each student's individual mark based on the peer assessments provided. If the weight increases, the gap between different students will also similarly increase. However, the average mark will not change, which is still the group mark that is given by supervisor. In some special cases, a student's mark may be larger than 100. At that time, this algorithm will automatically set that mark to 100.

Database

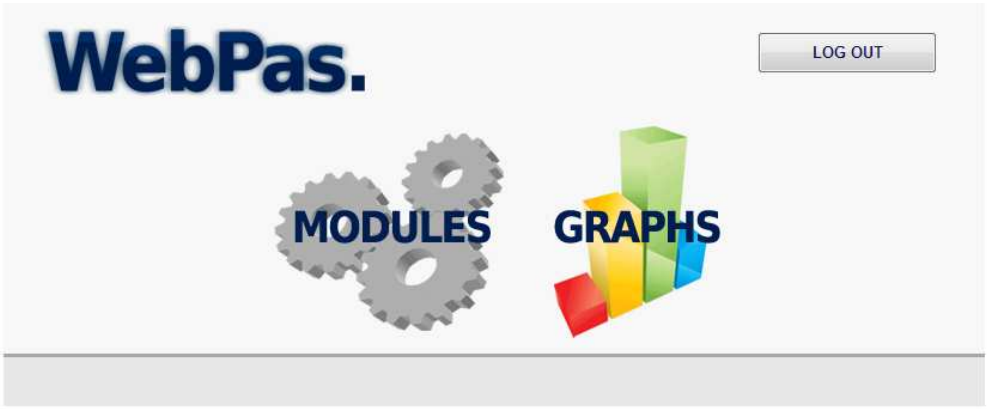


To create the database, we used MySQLWorkbench to first sketch the ER diagram to describe the abstract relations, attributes and keys among different tables. Then, the ER diagram is converted into the SQL code to create these tables in real database system. In total, there are five tables: PeerMark, Students, Groups, Modules and StudentsModule. One module can have lots of groups and one group can have lots of students. The relationship between Students and Modules is many-to-many (one student may register in lots of modules and one module may be chosen by lots of students). Hence, the junction table StudentsModule is used to connect them in order to improve the efficiency of SQL searching and avoid mess of data.

ER Diagram

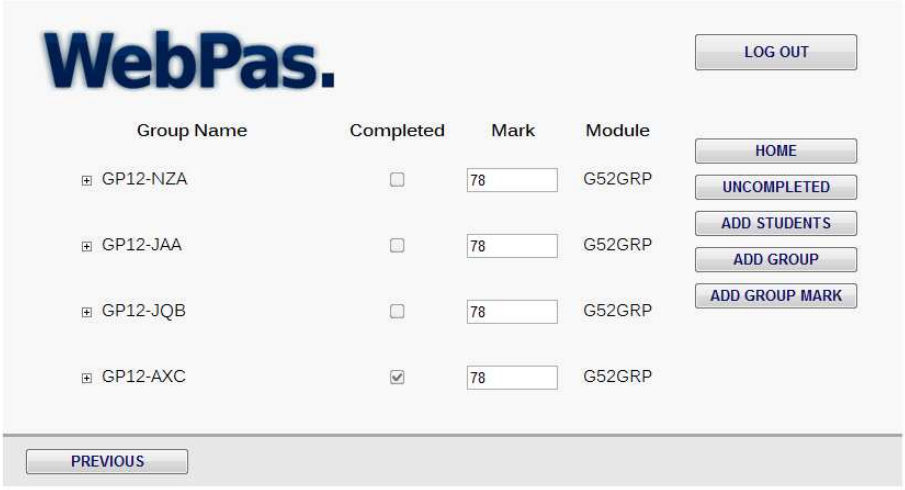


Website



Above is the module convenor page for the website if the user selects the modules option the below page is displayed.

This screen where the modules are displayed in a webpage – the module convenor can choose from the options on the right to change the weighting, view groups, add module or delete a module.



This last screen is the current groups page, which shows when group mark is editable, and the completed tickbox being implemented

Problems encountered so far

Technical Issues

- **Lack of MySQL Support**: Since our project was to incorporate a database of some sort, we had envisioned that we would use MySQL, since we had used it on a course last year and knew it was installed on the server already. However, when we first tried to login to the accounts we had set up last year we found that we could not access those accounts – this was resolved by e-mailing the module convenor and getting IS to allow us back into the MySQL accounts.
- **MySQL Accounts being tied to individual users**: We each had a MySQL account linked to our Computer Science usernames – this meant that each person's MySQL was different. This meant that to resolve this issue, the team member that was responsible for creating the database had to export it from his MySQL account and then upload it to the repository – this meant that each of the other members had to import that file into their MySQL so they could work with the current database. This could have been resolved better if there was a shared MySQL account for the group that the team could have logged into.

Management Issues

- **Task Scheduling:** Since we could not feasibly all work on all sections at the same time, it became necessary for us to divide out the workload. This was resolved by having each member of the group state what their strengths were (coding in specific languages, writing skills etc). We then all sat down and discussed how the workload should be divided by splitting it into certain skill areas, and then attributing those skills to team members who felt strong in that area. Another minor problem was that whilst there was an algorithm necessary to be built, it was only needed near the end. This was resolved by having the team member set to work on that temporarily re-assigned to help out in the database.
- **Language Barriers:** Within the group we had a team member whose native language was not English, so this led to a few mix-ups early on as we tried to understand each other. This was resolved as we learnt how to make each other understand better.

Time Plan for the project