Registration number 100024721

Module Support Website Project Report

Supervised by Dr Pierre Chardaire

University of East Anglia
Faculty of Science
School of Computing Sciences

Abstract

This document identifies what work needs to be done, why it needs to be done and how it will be done with regard to the Module Support Website. It brings forward some of the strategies that have been developed to undertake the task as well as the risks/shortcomings associated with them and how they may be dealt with. It also includes a work plan of the project showing the various tasks involved in the project and their timelines.

Contents

1 Introduction

Sometimes, there is not enough material to aid students with their revision. Students cannot (in most cases) set themselves questions and also mark themselves after answering those questions. They can however, see the lecturers to help them do this but it is a time consuming and sometimes convoluted process. Keep in mind that the lecturer might have to repeat the process for several students.

While this may not seem like a big deal, great reward could be yielded if this issue is fixed. Firstly, the increased level of organization and structure would save the lecturers and students some time allowing more productive use of their time. Secondly, with the wide variety of revision aid that will be provided, students can learn their subject material better and score better marks on exams. This would also increase the Student Satisfaction at the University.

At the moment, the University of East Anglia provides revision aid through the use of *Module Seminars*. Lecturers prepare questions and upload these to blackboard for the Students. The students then attend these weekly seminars, where a lecturer or Teaching Assistant goes through each question with the students. This is good but does not aid personal revision a whole lot.

The Module Support Website project aims to create a strong platform for revision aid for the students at the University of East Anglia. This report details the ideas behind the project and the activities undergone within it from conception through implementation. The aims and objectives of the project are discussed in *Section 2 (Project Outline)*. Some research was carried out in the field of Computer Assisted Learning to better understand the problem domain. This is documented in *Section 3 (Background Research)*. Following this, are the processes involved in the design and implementation of the project. This involves things like the database design, page design, programming languages and software architectures used, and more and can be found in *Section 4 (Project Design and Implementation)*. The resulting product was put under some tests to ensure that a quality product was delivered. *Section 5 (Testing and Usability Assess-*

ment) expands on the processes, methods and results of the tests carried out on the system. Finally, *Section 6 (Evaluations)* covers an evaluation of the outcome of the project and takes a reflective look on the project, suggesting ideas for possible improvements.

2 Project Outline

The project involves the design and implementation of a Website to support a Module (as a case study the Database Module). The Website will allow students to enroll and once their enrollment has been approved by their administrator, will allow log in facilities. Students enrolled on the module will have a bank of questions that they can access for each lecture or topic. The lecturer will be able to add or delete multiple choice questions, with their answers. Students will then be able to answer the questions and be given feedback on their performance. Their progress will be kept and in certain occasions be displayed in a league table. The website will act as a revision aid. Further functionality may be developed if time allows. The website will have to have good interface design should enable easy interaction and result in an attractive environment for the students.

Essentially, the aim of the project is to create a Computer Assisted Learning system to aid revision in universities. There are a number of interesting issues on Computer Assisted Learning and they will be covered in this report.

2.1 Aims

- Create a website which can serve as a useful revision aid for university students
- The website should allow lecturers to track students' progress
- The website should provide opportunities for students to work together while revising

2.2 Goals

- Students should be able to sign up and log in to the website
- Lecturers should be able to log in, add questions and view student progress
- Students should be able to select and drop modules of their choice
- Students should be able to answer questions and get feedback on their answers
- Students' progress should tracked and available for display on league tables
- Students should be able to discuss amongst themselves and help each other via a forum/discussion board medium

3 Background Research

For the project to create a very good product that not only fulfills but exceeds its requirements, several areas of knowledge need to be investigated and analyzed. The main areas that will be discussed are:

- An Introduction to Computer Assisted Learning (CAL),
- Rationale,
- Feasibility and Architecture,
- Comparison with traditional approaches,
- Concerns.

3.1 What is Computer Assisted Learning

Computer Assisted Learning, often abbreviated CAL, refers to the act of using Computers to offer new learning methods and augment existing ones. The crash in price of the microprocessor over the last 10 years has lead to the increased availability of microcomputers at home and at work. Unlike in the past, where mainframe computers were

very expensive to work on and had to be used as productively as possible, the cheapness of today's microcomputers allows for some machines to be used for a single purpose only. This means that there is plenty of time for a machine to be used to prepare CAL programs. Also, today's machines are much faster than they were in past times, allowing for virtually instant interactions with the CAL programs enabling them to provide immediate feedback for students.

In the early days of using computing in teaching, a large number of acronyms were used to describe different styles of CAL. Some examples are CAI (Computer-Aided Instruction), CBL (Computer-Based Learning)and CML (Computer-Managed Learning). In addition to these examples, there were many others and they all described different ways of incorporating computers in the learning process.

However, in 1977, Kemmis, Atkin & Wright proposed a framework for evaluating CAL which has stood the test of time. Kemmis *et al.* suggested that all forms of CAL could be categorised under one or more of four paradigms, and that these paradigms could be directly related to general educational trends and theories (Kemmis et al. (1977)). The paradigms were:

(a) The Instructional (drill and practice)

(b) The Revelatory (discovery learning)

(c) The Conjectural (what if?)

(d) The Emancipatory (electronic servant)

3.1.1 The Instructional Paradigm

This concept is based on the behaviourist psychology of B. F. Skinner. The idea is that learning takes place by the reinforcement of success. A correct response is rewarded by an appropriate reinforcer, an incorrect response is ignored. Since learning is organised to take place only when a correct response is reinforced, learning material must be prepared in such a way that correct responses are most likely to follow. This means that,

in practice, material must be split into small sections, each of which represents such a small advancement in understanding that success is guaranteed and rewarded before moving on to the next section.

This theory was tested with animals and yielded very positive results. This led to the development of teaching machines and the introduction of programmed learning, first in the US and then in the USSR and Europe. However, many teachers, particularly in the UK, were sceptical about the value of programmed learning, and their doubts were not confined to the technological aspects of the concept. They were aware that students soon became weary with the repetitive nature of the learn-test-reward cycle. Also, a number of them questioned the whole basis of behaviourist psychology, and felt from their own experiences that human beings learn as much from mistakes as from success.

Thus, the Instructional Paradigm is a style of teaching that is not in much favour in this country at present. Instructional programs are likely to take the form of branching programs, testing the student's knowledge, and allowing remedial loops to be entered if a certain pre-determined score has not been achieved. However, such programs still represent the bottom line of CAL, since they do not provide any freedom of choice to the learners, but herd them through the pre-prepared dialogue of the instructor. Because of this though, the Instructional Paradigm is well suited for Multiple Choice Questions (MCQs).

In summary, the Instructional Paradigm has little to offer beyond a convenient way of revision and for familiarity and practice with MCQs (Clayden and Wilson (2009)).

3.1.2 The Revelatory Paradigm

The Revelatory Paradigm is derived from cognitive psychology. The concept is that learning takes place when one encounters and overcomes problems and difficulties. This technique is favoured in the English educational system, and exploited in the familiar "project" method of learning.

In terms of Computer Assisted Learning (CAL), this means that more emphasis is placed on the students and their needs, and less on the instructional material (that is, the program). Instead of the program being in total control, with the students as passive responders (as is the case in the Instructional Paradigm), the program allows and encourages the students to choose paths through the knowledge base, to explore what is hidden and what is gradually revealed. Thus the program acts as a middle-man between the subject material and the student. Usually, the program sets up a model of reality which the learners can explore at their own speed and in their own way (Clayden and Wilson (2009)).

This model is usually in the form of a simulation. A simulation is appropriate when reality itself is too dangerous, too remote, too complex, too costly or otherwise inappropriate for the student. Hence, a 10-year-old child might explore the simulation of a Saxon settlement or an older pupil might simulate the industrial manufacture of sulphuric acid. One good example of a simulation commercially available is CRISIS, a major accident simulation, available from Open Learning Software ([1]) , which is relevant to medical students, nurses and administrators.

However, no simulation can hope to model every aspect of the real-life situation, and it is all too easy to draw a simplistic representation of reality. It is obviously important that the program makes this perfectly clear and refers users to other sources of information and data.

3.1.3 The Conjectural Paradigm

The Conjectural Paradigm is simply an extension of the Revelatory Paradigm. Like the Revelatory Paradigm, the Conjectural Paradigm may take the form of a simulation. The difference is that it allows the students to not only explore the model, but to actually alter the nature of the model, or at least the parameters underlying it (Clayden and Wilson (2009)). It is sometimes described as the 'What if…?' approach to learning.

[1] https://www.openlearning.com

In the Conjectural Paradigm, there may not be any 'correct' answer. The simulation/model exists for the students to manipulate as they desire, and to draw conclusions from it as they seem to account for the model's behaviour. It can be seen as a sort of simulation of scientific research, but without the cost, dangers and other drawbacks of the real thing.

Programs of this paradigm tend to mainly be physiological representations of reality simply because if reality can be expressed in mathematical terms, then any adjustment to the simulation/model's parameters can be accommodated and explored. Some popular examples are the learning program series for the Apple Mac computer (McPuff, McPhee, etc.).

In summary, the ultimate use of this genre in CAL is the knowledge base or expert system. This system consists of logical rules expressing all the known wisdom on a particular topic. This type of application is currently beyond the power of the microcomputer except in the most trivial of examples. However, with the ever-decreasing cost of memory, simple but worthwhile knowledge base systems are very close to being implemented on microcomputers.

3.1.4 The Emancipatory Paradigm

It is important to note that not all aspects of CAL are to do with teaching. It is significant that when thinking of CAL, emphasis is placed on the "learning" aspect rather than the "teaching" aspect of education. Learning may involve the writing of essays, gathering of notes and the analysis of results. Today, our technological progress has revolutionised students' ability to process large amounts of information. Tasks which were previously time consuming and error prone can now be handled quickly and with confidence. Much of the graft of essay-writing is mitigated when a word-processor makes the revision of previous drafts so easy. Ideas can be expressed quickly as they occur, and then tidied up into a more literary form at leisure.

Here, the computer becomes a tool for the students, to be used in whatever role they

require. It is the students who are in charge rather than the program (in contrast to the other paradigms of CAL (Clayden and Wilson (2009)). The students choose the program, whether it be a word-processor, a spreadsheet or a database. In the Emancipatory Paradigm, the computer sets the students free to manipulate information in a way which would not have been possible otherwise.

This system of CAL is extremely different from programmed learning. Students learn what they want, when they want, at their own pace. However, these advantages come with a cost. The software is very difficult to write and an effective system requires the dedicated collaboration of program designers, educationalists and subject specialists over a considerable period of time.

3.2 Rationale Behind CAL and Theoretical Underpinnings

Constructivism has become quite popular in the fields of teacher preparation, education, scholarship and policy formation (Richardson (1997); Teets and Starnes (1996)). Its approaches maintain that individuals create knowledge through active involvement with content rather than through imitation or repetition. In the traditional approach, the teacher fills the students with blocks of information deemed by the teacher to be knowledge, and the students store these pieces of information intact, until needed (Oldfather (1993)). Constructivists argue that when information is obtained purely by this sort of transmission model, it is not always integrated with prior knowledge and is often only accessed and used for academic purposes such as formal tests and exams (Richardson (1997)). Teaching and learning via constructivist models have been argued to produce in-depth understanding and application of knowledge (Gaile and Reiff (1994); Richardson (1997)).

Nevertheless, such arguments do not mean that traditional approaches to teaching and learning are wrong or useless. In most subjects taught at secondary or primary levels, students typically require both domain knowledge and the ability to apply such knowledge to solve problems that may be ill or well structured.

One of the goals of Computer Assisted Learning is to shift students' attitudes towards education from one that is mainly exam oriented to one that involves active learning and knowledge construction (Goh et al. (2005)). For this reason, students perusing questions and solutions should also be able to access and interact with related resources. That is, in addition to a bank of questions, the system should also contain a pool of relevant information and references on the subject material. The two systems should work in concert, providing focused examination preparation while simultaneously facilitating the exploration of relevant concepts in general.

3.3 Advantages of Computer Assisted Learning

Computer Assisted Learning is often wrongly overlooked by educational institutions as not very useful or simply not worth the cost. The rationale behind CAL and thus, why it is useful has been explained in the previous section. In this section, we look at the reasons why educational institutions should invest in CAL.

(a) **Computer assisted learning is inevitable -** Individual lecturers and departments are already beginning to introduce a wide range of computer based applications, sometimes in a haphazard way. Planned and coordinated development is better than indiscriminate expansion (Greenhalgh (2001)).

(b) **Unique presentational benefits -** Computer presentation is particularly useful for subjects that are visually intensive, detail oriented, and difficult to conceptualise, such as complex biochemical processes or microscopic images (Phillips (1996)). Furthermore, "virtual" cases may reduce the need to use actual animal or human tissue in learning.

(c) **Personalised learning -** Each student can progress at his or her preferred pace. They can repeat, interrupt, and resume at will. This may have particular advantages for weaker students.

(d) **Economies of scale -** Once the software has been set up, the incremental cost of offering it to additional students is relatively small.

(e) **Competitive advantage -** Potential applicants to an educational institution may use the quality of information technology to discriminate between schools. A modern, technological campus is likely to attract good students.

(f) **Achieves the ultimate goal of higher education -** The goal is to link people into learning communities. Computer applications, especially the internet and world wide web, are an extremely efficient way of doing this (Haag et al. (1999)).

(g) **Expands pedagogical horizons -** The most controversial argument for using CAL in higher education is the alleged ability to fundamentally change the relation between people and knowledge (Daniel (1998)).

3.4 Disadvantages and Concerns with Computer Assisted Learning

Computers, software and internet-based services are not seen solely as positive agents of change in the classroom; they also face criticism. One of such criticisms of the use of computers in learning is the possibility of the complete omission of learning traditional information systems. An example of this is how a reliance on electronic calculators has discouraged some students' basic numeracy skills.

As mentioned above, traditional approaches to learning tend to follow a set scope and sequence. A strength (and possibly a weakness) of some CAL systems is the lack of a clearly determined scope and sequence of learning. Instead, they allow individuals to pursue links which they perceive as being useful or interesting. This means that the students are not all constrained to learn the material in the same way but may instead find new and different solutions to the same problem. Also, students are liable to become overwhelmed with all the information immediately available to them and become distracted or lost in the sea of information available to them. For such resources to work, students may need more guidance in the form of on-screen help dialogs that appear not

just when requested but also when the student seems to be stalled or going off-track.

It is necessary to assess the role of computers and computer software in the classroom. Various CAL applications have been promoted as virtual teachers, however, software programs still have a long way to go before they can begin to offer comprehensive and effective teaching without human intervention. Although it is easy to see how a computer can assume some of a teacher's roles, there are clear limits to the ways in which a computer program is able to take the place of a teacher (Beatty (2003)).

Also, there is the issue of cognitive styles and Computer Assisted Learning. Cognitive style has been defined in different ways by different people (e.g. Kogan (1971)) but essentially refers to the unique and preferred way in which individuals process information. It is important that the concept of cognitive style as a significant trait of the student is catered for in the development of CAL programs. There are a vast number of cognitive styles and it is very difficult (if not impossible) for a CAL program to cater to each and every one of these styles. (Clarke (1993))

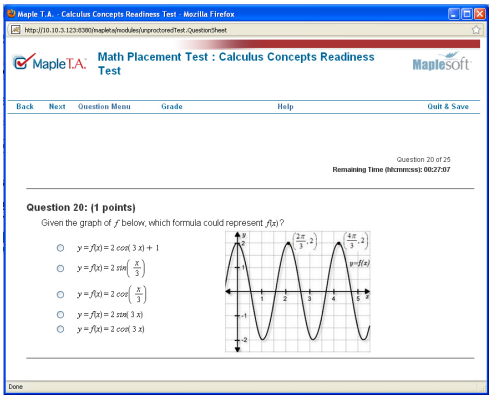
3.5 Module Support Website Similar Systems Analysis

Before starting on my project, I carried out some investigation to find other systems or products with features similar to those of my project. While I did not find a system that was exactly the same as mine, I did find a number of systems with which my project shared common elements. MapleTA and Blackboard both offer question creation and answer facilities. Another similar system I found was an iOS app called "myCQs".

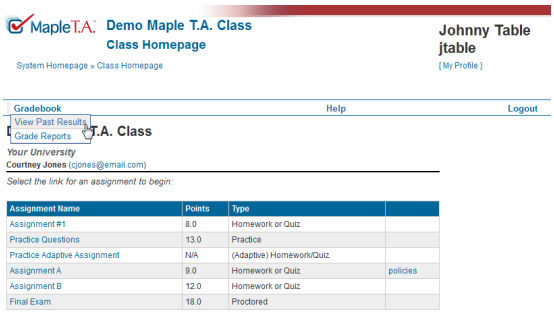
3.5.1 MapleTA

MapleTA is a web-based testing and assessment website for mathematics. It has 15 question types, control over numerous aspects of the assignments, and a gradebook that students can use to keep track of their progress. Some of these 15 question types include mathematical free response, multiple choice, fill-in-the-blank, matching, graph sketching, essay and free body diagrams. MapleTA also allows for different assign-

ment types (ie practice, homework, high-stake-testing etc). MapleTA also provides an optional proctored browser as a testing environment which requires students to stay inside Maple T.A. until the test or assignment is completed, so they cannot access other web sites or programs. It also provides add-ons and can be integrated with large course management systems such as Blackboard.



(a) MapleTA Question



(b) MapleTA Gradebook

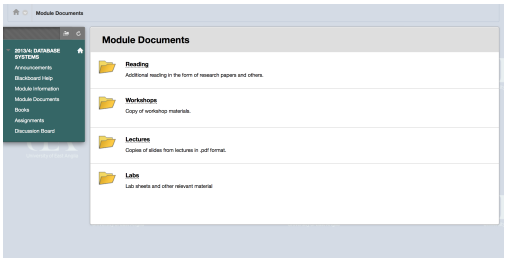
Figure 1: MapleTA Screens

3.5.2 Blackboard

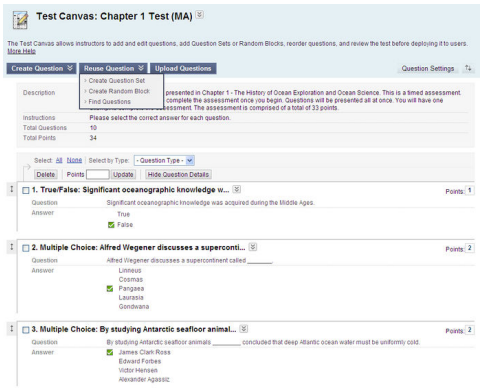
The Blackboard Learning System is a virtual learning environment and course management system developed by Blackboard Inc. It is a Web-based server software which features course management, customizable open architecture, and scalable design that allows integration with student information systems and authentication protocols (Wikipedia (2000)). It can be installed on local servers or hosted by Blackboard ASP Solutions. Its main purposes are to add online elements to courses that are typically delivered face-to-face and to develop new completely online courses with few or no face-to-face meetings.

Blackboard allows lecturers to post course content such as lecture notes, articles, assignments and videos. Students can also submit their solutions to posted assignments via Blackboard. It also allows teachers to post quizzes which students can access via the internet. It has different question types such as multiple choice, true-or-false, fill-

in-the-blank etc. It also has a “Discussions” feature which allows students and lecturers to discuss course content by means of a monitored discussion board. Blackboard can also be used to create surveys and course evaluations for students to fill which serves as feedback for the lecturers.



(a) Blackboard layout

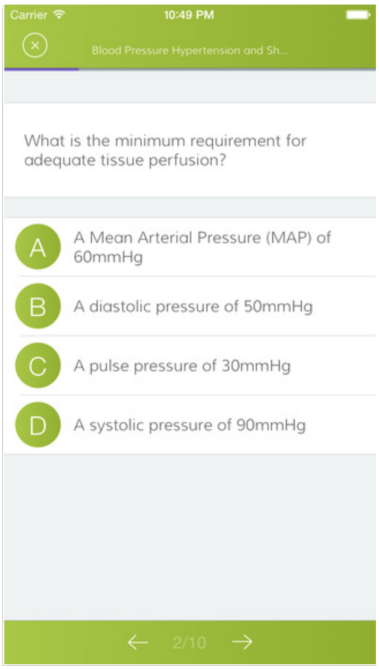


(b) Blackboard test maker

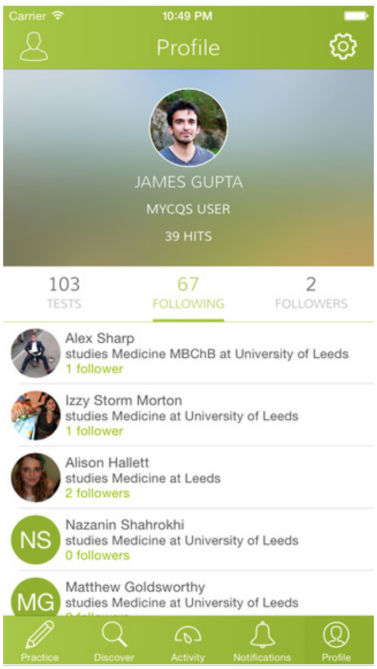
Figure 2: Blackboard Screens

3.5.3 MyCQs

MyCQs is a learning platform that lets you create, practice and share Multiple Choice Question (MCQ) tests with others (Ltd (2013a)). It is in the form of a mobile app and accompanying website. MyCQs is largely reliant on its community in that it encourages its users to not only be test-takers but test creators. Its idea behind this is that creating tests forces the user to read through the course material in an active and critical manner, bringing any underlying gaps in their understanding to the surface, and leaving them with a challenging set of questions that they can use to revise any time. Being a mobile app, the user can practice tests anytime. MyCQs also encourages its users to share their created practice tests for others to attempt. This adds a social as well as a competitive element to the system. It currently has over 200,000 user created/shared questions (Ltd (2013b)). The system organises tests and questions into different categories (such as Engineering, Computer Science, Dentistry, Law etc) and also allows users to create and join groups. Users in groups can share knowledge and discuss tests.



(a) MyCQs test



(b) MyCQs user

Figure 3: MyCQs App Screens

4 Project Design and Implementation

4.1 Planning

4.1.1 Architecture and Environment

The project is going to be a dynamic and interactive website application. It will be specific to and maintained by the University of East Anglia. The application will be written using an AMP architecture. An AMP architecture is a web development stack consisting of the the **Apache HTTP Server**, the **MySQL relational database management system**, and the **PHP programming language**. It is a cross platform stack as it encompasses LAMP (Linux), WAMP (Windows) and MAMP (Macintosh). As a solution stack, AMP is suitable for building dynamic web sites and web applications. Furthermore, the website could possibly be hosted using Heroku[2]).

[2]https://www.heroku.com

The application will be developed without any particular IDE (Integrated Development Environment) but rather with powerful text editor programs (such as *SublimeText*[3] and *TextMate*[4]). For development and testing, the *XAMPP*[5] software will be used to provide a local Apache server and a mySQL database. Also the PaaS (Platform-As-A-Service) *Parse*[6] will be used to store some data and provide some extra features such as Analytics and Cloud Code.

4.1.2 Implementation Issues, Challenges and Risks

The most difficult issues of the project will be those of scale and thread-safety/atomicity. This is because as the website gains more and more users, resources and features will be accessed at the same time by a lot of people. This could create some unforseen situations in the program.

To combat this, the code written is meticulously thread-safe and the database follows the ACID properties to the latter. As for the issue of Scalability, the finished application will be hosted using the PaaS (Platform-As-A-Service) called *Heroku*. Unlike the average web hosting service where resources and performance become constrained as the number of users rise, Heroku is specifically tailored to address scalability issues. The Heroku service is free until a certain load of users is reached. This will direct development focus to creating the best product possible without having to worry much about scale until it is not only complete but has gained dedicated use. At that point, the university can take over hosting and scale and is very well equipped to do so.

4.1.3 Deliverables

This project is going to have two main deliverables:

[3] http://www.sublimetext.com
[4] https://github.com/textmate/textmate
[5] http://en.wikipedia.org/wiki/XAMPP
[6] https://www.parse.com

- The project report
- The Module Support Website
- The Module Support Mobile App

The project report will contain a detailed record of all the activities that have gone into the project. This will include all the preliminary research in the form of a Literature Review, an analysis of the problem, methods of solving it and the issues associated with these methods, implementation activities, testing methods and results and a summary and discussion.

The actual website will be a finished version of the proposed solution, thoroughly tested and ready for use. It will be a working module support website for the Databases module. The website will greatly push student personal revision and improve performance in the university.

In addition to the website, an accompanying mobile app will also be created to complement the website. This is not listed in the requirements but would be a very useful addition nonetheless.

4.2 Timeline

The project will be carried out over the space of 39 weeks. Below is a Gantt Chart showing the timeline of activities that will be carried out in the project.

4.3 Requirements Specification

Before beginning work on the project, its requirements need to be identified and prioritized. The project requirements have already been identified in the previous section. In this section, the prioritization of the requirements is documented. The requirements were prioritized using the MoSCoW system. The MoSCoW system splits the project into four categories with varying levels of priority - Musts, Shoulds, Coulds and Won'ts.

Musts are requirements vital to the project's success and without which, the project would not be viable. They are also known as the Minimum Usable Subset (MUS). Shoulds are requirements that are important and have a significant impact on the final outcome of the project, but can be left out without making the project a failure. Similarly, Coulds are requirements that can be left out if necessary, but with less impact on the project outcome than Shoulds. They involve desirable requirements that are not vital to the project. Finally, Won'ts are requirements that would be nice to have but will not be delivered due to limitations of the project. They ultimately do not affect the viability of the project but would improve the user experience or efficiency of the system. Table 1 shows the MoSCoW requirements breakdown for this project.

Table 1: MoSCoW Requirements

Category	Requirement
Musts	Login and Registration facilities.
	Students able to enroll on modules.
	Student access to a bank of questions for enrolled modules.
	Lecturers able to add multiple-choice-questions.
Shoulds	Students able to answer questions.
	Feedback on answered questions.
	Student progress persistence.
	League tables/rankings.
Coulds	More question types in addition to multiple-choice.
	Forum/Message board.
	User activity log.
	Accompanying mobile app.
Wonts	User avatars.
	Student created questions.

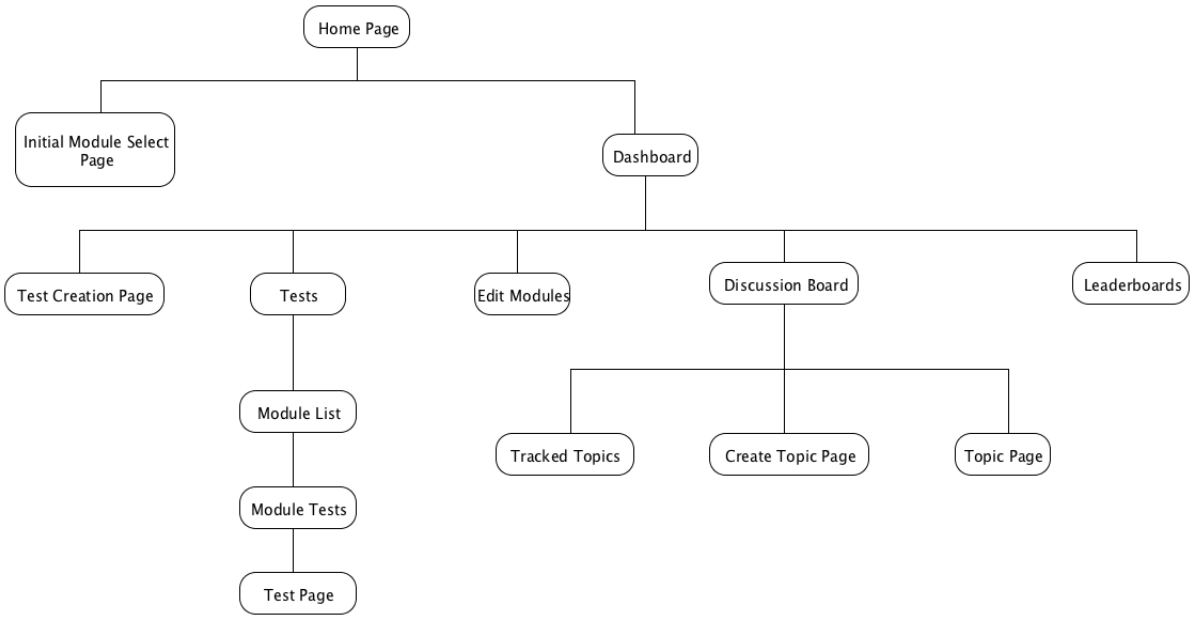


Figure 4: Module Support Website Sitemap

4.4 Navigation and Sitemap

An overview of the layout of the website is shown in the sitemap in figure 4. The website starts on a home/landing page. On this page the user can either log in or sign up to use the service. If the user logs in successfully, they are taken to the **Dashboard** page (if the login fails, an informative error message is shown). On the other hand, if the user chooses to sign up, after the sign up process is completed successfully, the user is taken to the **Initial Module Select** page. On this page, the user is given the option to select modules in which they are interested and want to engage with on the website. From the dashboard, every user has access to the 4 main parts of the website - **Tests**, **Edit Modules**, **Discussion Board** and **Leaderboards**. If the user possesses admin privileges, an additional page will be available - the **Test Creation** page. The Test Creation page contains an interface for lecturers to add and remove tests from the website.

The tests page displays a list of the user's modules for which tests are available. When a module is selected, the user is navigated to the **Module Tests** page. This page displays

a list of tests for the selected module. When a test is clicked, the user is taken to the **Test** page, where they can then attempt the test.

The Edit Modules page provides an interface for the user to add and remove modules to their module list. It is the module management interface. When a module is removed from the user's list, their test progress is deleted.

The Discussion Board page resembles a traditional forum or message board. It displays all of the topics pertaining to the user's modules. Additionally, there is be a "filter" option, to only display topics for a single one of the user's modules as well as a search function. From here, the user may navigate to the **Tracked Topics**, **Create Topic** and **Topic** pages. The Tracked Topics page shows a list of topics that the user has "favourited". This is to give the user quick access to the topics that they are really interested in. The Create Topic page provides an interface for the user to add a new topic to the discussion board. Finally, the Topic page is navigated to when the user clicks on a topic. It shows the posts within the topic.

The Leaderboards page shows the rankings of users in different modules.

4.5 Database Design

Persistence is achieved using a relational database. MySQL is used as the Database Management System. This decision was made based on the fact that the website was written in PHP and PHP has numerous APIs for database connectivity with MySQL built into it.

The database consists of 16 tables. 9 of these tables are main entities being modelled and the remaining 7 are associative entities to decompose complex entity relationships and make the database more normalised. The tables are listed and explained as follows:

- User - Stores data relevant to the user. Its attributes are *email*, *username* and

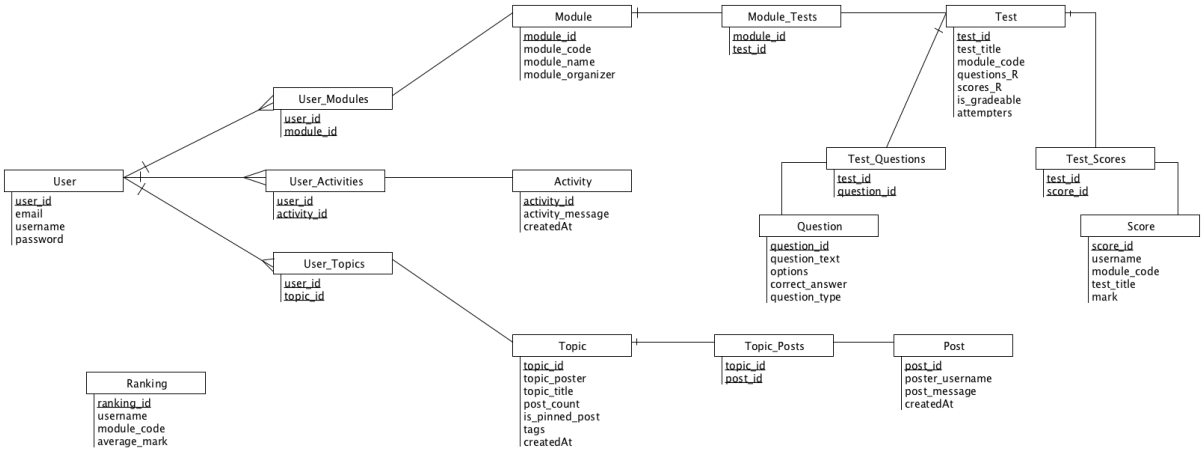


Figure 5: Database Entity Relationship Diagram

password. All attributes are strings.

- Module - Stores data relevant to the module. Its attributes are *module_name*, *module_code* and *module_organizer*. All attributes are strings.

- Test - Stores data relevant to a revision test on the website. Its attributes are *test_title*, *module_code*, *is_gradeable* and *attempters*. Test_title is the name of the test. Module_code is the code for the module to which the test belongs. Is_gradeable is a boolean value representing whether the test is for practice or not. Attempters is a string containing the usernames of users who have taken the test separated by some delimiter to allow for parsing.

- Question - Models a single question that is part of a test. Its attributes are *question_text*, *options*, *correct_answer* and *question_type*. Question_text is a string containing the question being asked. Options may contain different kinds of values depending on the type of question. For multiple choice questions, it is a string containing the answer options for the question separated by some delimiter. If the question is a single-answer question, options is an empty string. Correct_answer is a string containing the correct answer (not just an option letter). Finally, question_type is a string that reflects the type of question (ie multiple-choice, true-or-

false etc).

- Score - Stores a user's score for a test. Its attributes are *username*, *module_code*, *test_title* and *mark*. Username is the username of the user whose score it is. Module_code is the code for the module in which the test this score was obtained in. Test_title is the name of the test this score was obtained in. Mark is numeric value representing the mark obtained by the user in a test.

- Topic - Stores data for a topic on the discussion board. Its attributes are *topic_poster*, *topic_title*, *post_count*, *is_pinned_post*, *tags* and *createdAt*. Topic_poster is the username of the user who created the topic. Topic_title is the title of the topic. Post_count is a numeric value denoting the number of posts within a topic's thread. Is_pinned_post is a boolean value that represent whether a post is pinned or not. What is a pinned post? A useful or important post may be pinned by an admin so that it is always shown at the top of every page on the discussion board. Tags is a string containing a list of tags that may be relevant to the topic (to aid discovery and search) separated by some delimiter to allow for parsing. CreatedAt is the timestamp for the moment the topic was created.

- Post - Stores data for a post under a topic. Its attributes are *poster_username*, *post_message* and *createdAt*. Poster_username is the username of the user who made the post. Post_mesage is a string containing the post. CreatedAt is the timestamp for the moment the post was made.

- Activity - Stores a log of user activity. Examples of activities that are logged are taking a test, posting in the discussion board or taking on a new module. Its attributes are *activity_message* and *createdAt*. Activity_message is a string containing a message conveying the activity that a user has performed. CreatedAt is the timestamp for the moment the activity was performed.

- Ranking - Stores users' rankings within modules. Its attributes are *username*, *module_code* and *average_mark*. Username is the username of a user whose ranking is being stored. Module_code is the code for the module with which

a ranking is concerned. Average_mark is a numeric value containing the mean score achieved by a user in all tests for a given module.

- User_Activities - An associative entity used to model the list of activities pertaining to each user. Its only attributes are the *user_id* and *activity_id* which together form the primary key. The activity_id is used to fetch the remaining activity data for the user from the Activity table.

- User_Modules - An associative entity used to model the list of modules taken by each user. Its only attributes are the *user_id* and *module_id* which together form the primary key. The module_id is used to fetch the remaining module data for the user from the Module table.

- User_Topics - An associative entity used to model the list of topics created by each user. Its only attributes are the *user_id* and *topic_id* which together form the primary key. The topic_id is used to fetch the remaining topic data for the user from the Topic table.

- Module_Tests - An associative entity used to model the list of tests available for each module. Its only attributes are the *module_id* and *test_id* which together form the primary key. The test_id is used to fetch the remaining test data for the module from the Test table.

- Test_Questions - An associative entity used to model the list of questions contained within each test. Its only attributes are the *test_id* and *question_id* which together form the primary key. The question_id is used to fetch the remaining question data for the test from the Question table.

- Test_Scores - An associative entity used to model the list of scores for each test. Its only attributes are the *test_id* and *score_id* which together form the primary key. The score_id is used to fetch the remaining score data for the test from the Score table.

- Topic_Posts - An associative entity used to model the list of posts contained within each topic. Its only attributes are the *topic_id* and *post_id* which together form

the primary key. The post_id is used to fetch the remaining post data for the topic from the Post table.

4.6 Website Features and Implementation

4.6.1 Website Overview

The Module Support website contains 4 main sections - Tests, Edit Modules, Discussion Board and Leaderboards. Students can select a number of modules and take tests on these modules. There are two kinds of tests - ***Practice Tests*** and ***Gradeable Tests***. While both tests can be completed multiple times, the score attained in gradeable tests is saved the first time and will not change upon completing the test subsequently whereas with Practice tests, the user can improve their score with multiple attempts. Users can also interact with each other on the discussion board. Each module has its own board. Topics on a board can also be tagged with relevant words. For example, a topic in the Graphics module can have a "collision detection" tag applied to it to make it easy for users to find relevant content. The website also has a leaderboards feature where the top scorers in tests for each module are displayed. The aim of this is to foster healthy competition.

4.6.2 Module Management

The module management page can be accessed from the dashboard by clicking on the edit icon in the modules sidebar. The module management page shows a list of the modules a user has added. Next to each of these modules is a delete button. When the delete button is pressed, that module is removed from the user's list. This also clears this user's scores in tests for that module, hence removing them from the leaderboard for that module.

At the bottom of the module management page is an add module button. When this button is pressed, the user is shown a list of all the modules on the site with checkboxes next to them. The user can then check the boxes for modules that they wish to add to their module list. Once satisfied with their selection, they can click on the "Confirm Selections" button to save their changes.

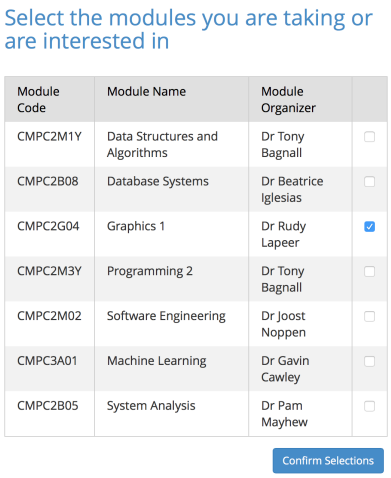


Figure 6: Add module interface

4.6.3 Test Taking

A user can take tests by clicking on the “Tests” link on the navigation bar or every page. This function has been placed here to make it quickly and easily available to the user as it is one of the major functions of the system. When the user clicks this link, they are taken to a a page showing a list of their modules. This list also shows the number of tests available for each module. A module is clicked to show tests for that module. As explained above in the overview, a test can either be a practice test or a gradeable test. Each test shown on this page has a marker on it denoting whether it is a practice or gradeable test. A test is clicked to attempt it. If the test clicked is gradeable, a warning is displayed, telling the user that while the test may be attempted multiple times, only their first score is saved. This is to enable them to go away to prepare and then return to take on the test when they feel more ready. Once a test is completed and submitted, it is graded and the user is given feedback illustrating what answers they entered and what the correct answers are.

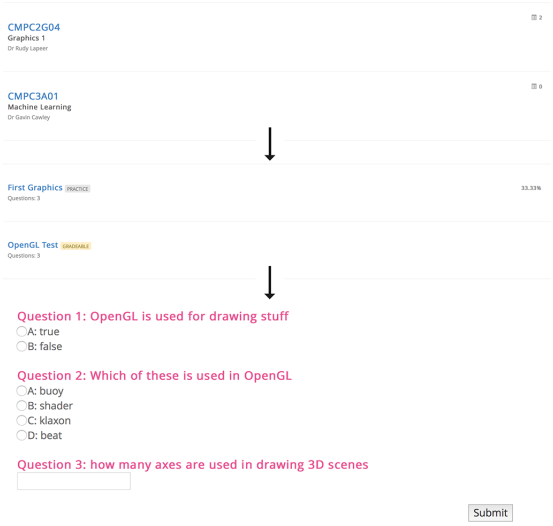


Figure 7: Test Taking Process Illustration

4.6.4 Test Creation

Users with admin privileges have access to the "Create Test" link on their dashboard. Lecturers will be given admin privileges so that they will have access to this link. Lecturers can then click on this link to take them to the test creation page. The test creation page offers an interface with which to create new tests. At the moment, all lecturers have the ability to create tests for all modules, even modules with which they are not concerned. While aware that this could be a security risk, this issue was not addressed because this project is a prototype. If the project is subsequently carried forward, this issue will then be addressed before launch.

From the similar system analysis, it was found that most of the other vendors offering similar services to those of this project (such as Blackboard and MapleTA) offered multiple question types. This system was hence also designed to do the same. At the moment, test creators have the option of creating multiple-choice questions, true-or-false questions and single answer questions. More question types may be added in the future.

4.6.5 Discussion Board

The discussion board can be accessed by clicking the "board" link on the navigation bar. The discussion board shows a list of topics. Within each topic is a number of posts (or messages) by different users about that topic. Each topic belongs under a module. In addition to this, each topic can have tags. A tag is a label attached to a topic to aid discovery and give more information about the topic. For example, a topic created under the Graphics module may have the tag "Collision Detection". This gives more information on the nature of the topic and allows users to find relevant content quickly. Also, some topics may be "pinned" by users with admin privileges. Pinned topics are displayed at the top of every board page. This ensures that important information will be seen by everyone.

By default, the board page displays topics for all modules that the viewing user has added to their module list. The user can filter the board by any of their modules so that

it only shows topics for that module. There is also a search bar with a dropdown that allows the user to search for topics either by checking the search text against topic titles or topic tags. A user can also "track" topics by clicking on the "track topic" button in a topic's page. This adds that topic to the user's tracked topics. This gives the user a quick point of access to those topics in which they are particularly interested in. There is a button for viewing tracked topics on the board page. The board page also houses a "new topic" button which displays an interface for creating a new topic.

4.6.6 Leaderboards

The Leaderboards page can be reached by clicking on the graph icon on the navigation bar. The leaderboards page simply shows the rankings of the users for different modules. It shows a list of all the users who have added a particular module to their module list. There is a dropdown which is used to select what module to view rankings for. The list is sorted, showing the users with the highest average mark for that module on the top. It displays the user's position, username and mark percentage. The mark is derived by calculating the mean value of the user's scores in the tests under that module.

4.7 Mobile App Features and Implementation

4.7.1 Mobile App Overview

A native iOS app was written to complement the Module Support Website. This app will run on both iPhones and iPads (that is, Apple phones and tablets). The app provides users with the opportunity to carry the main features and strengths of the module support website along with them at all times. The app can be used to manage one's modules and take tests with feedback. However, the mobile app is not aimed at lecturers and so does not offer admin services. Users of the app can also choose to receive push notifications whenever a new test is made available. The app is not intended to be used solely by itself without the website. It is to serve as a refined point of access to the system's services while on the go.

While developing the app, care was taken to use the same colours and fonts as the

website where possible. This was to ensure that users feel at home no matter which platform they are using to access the system. This allows the user to use both platforms interchangeably. When the app is opened, it presents the user with the option to either login or signup. Once signed up, the user's session persists between opening and closing the app, until he/she logs out. Once a user logs in or signs up, they are taken to the main area of the app, which consists of three tabs, for "Activitiy", "Modules" and "Settings" respesctively. The user can select these tabs to navigate to those areas of the app.

4.7.2 Module Management (Mobile)

The module management feature of the app is accessed from the "Modules" tab. On the navigation bar of this tab, there is an edit button. Pressing this button initiates the module management process.

Each row of modules displays a red button that can be pressed to remove that module from the user's module list. At the bottom of the list is a blank row with a green button containing a plus symbol (+). When this button is pressed, a modal list containing all the modules available is displayed. The user can then select a number of modules from this list. Once satisfied with their selections, the "Done" button on the navigation bar is pressed to save the changes and dismiss the selection list, thus navigating the user back to their newly updated module list. A shot of the module management screen, illustrating the process, is shown in Figure 8.

4.7.3 Test Taking (Mobile)

Tests are also accessed from the "Modules" tab. Selecting a module on this list navigates to a new page showing a list of the tests available for that module. Like the website, the type of the test (ie practice test or gradeable test) is also displayed beneath each test. In addition to this, the number of questions within each test and the user's score for that test are displayed for each test. If a test has not yet been attempted by a user, instead of displaying the user's score, a hyphen is displayed. To attempt one of these tests, the user may tap the test. This will navigate the user to the questions page. This page provides left and right arrows that allow the user to move between questions in the test by pressing

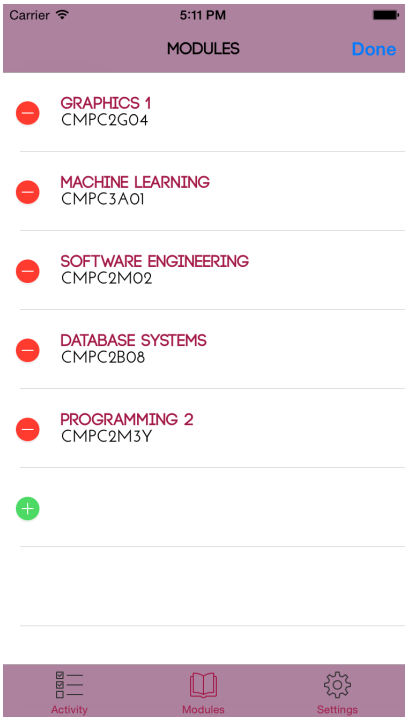


Figure 8: Module Management on iOS app

them. Once done answering all the questions, the user may press the "Submit" button to have their answers marked and receive feedback on their performance. It is possible to quit a test at any time by pressing the "Quit" button, however quitting gradeable tests will leave the user with zero as their score. Also, similar to the website, before beginning a gradeable test, a warning alert is issued, informing the user of the nature of a gradeable test. It also tells them what happens if they quit. Figure 9 an illustration of the test taking process.

5 Testing and Usability Assessment

The project was implemented using a *Test-Driven Development* approach. Throughout the implementation phase, the website and app were regularly tested at the end of each step. This allowed for bugs to be identified and fixed early on, preventing the errors from getting more complicated as the codebase grew. Often, progress could not be made until

Figure 9: Mobile Test Taking Process Illustration

the stage in development was working as expected, free of errors.

However, testing the final product as a whole is also very important as some aspects of the application cannot be tested separately. The final products were tested in two parts: ***Black Box Testing*** and ***End User Assessment***. Black box testing involves inputting data to the system and observing the output without much focus on its internal workings. The end user assessment involved a survey to assess the usability of the system and gauge customer reactions to it. These processes are covered in more depth in the following subsections. In addition to all this, all the webpages of the website were run through a HTML validator to ensure that they were up to the W3C standard.

5.1 Black Box Testing

Black box testing of the website was carried out in Google Chrome, Safari, Mozilla Firefox and Internet Explorer to ensure browser compatibility. This was done to make

sure the user experience was good for all users, despite whatever browser they were using. Tests were carried out for registered users, unregistered users and admin users to ensure that all possible interactions between the user and the site worked as intended. If the outcome of the tested interaction did not match the expected outcome, the logs and stack traces were examined to identify the problem and then solve it.

As well as correct functionality, the load speeds of the system were tested to ensure that each page ran as quickly and efficiently as possible. This was done using a combination of Google Chrome's Network Developer tool and Google's PageSpeed service.

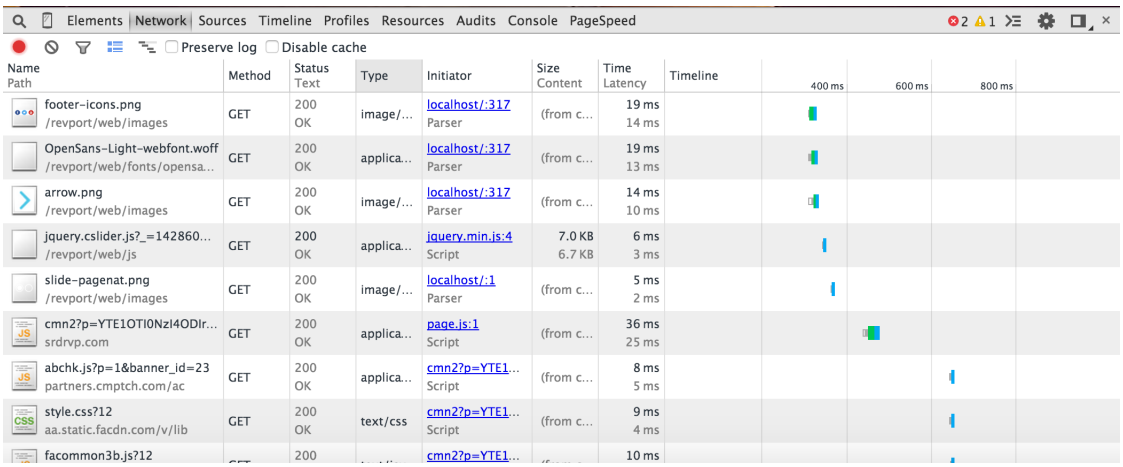


Figure 10: Page load analysis with Google Chrome network tool

Google Chrome's Network tool enabled the checking of page sizes and their load times. A screenshot of this tool in action is shown in figure 10. The network tool displays a Gantt chart showing the activities involved in the page load against the page load timeline. In the Gantt chart, it shows how much time each element takes to load. Using this, elements slowing down the page can be identified. PageSpeed is a service offered by google which is used to complement this. PageSpeed analyses a webpage and scores it, while suggesting areas that can be improved and ways these improvements might be achieved. This was used on all the pages in the website to make optimizations and fixes. A screenshot of a PageSpeed assessment of the module support website is shown in figure 11.

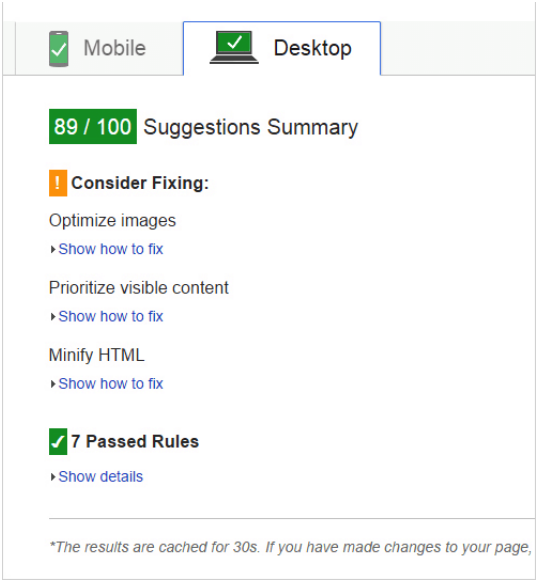


Figure 11: Web page analysis with Google PageSpeed service

5.2 User Assessment

5.2.1 Rationale for Conducting Survey

The main aim of the survey was to promote active discussions on the nature of the user experience and quality of the Module Support website. The mobile app is not involved in the survey as it is a merely an add-on and the survey is intended to test the quality of the core system and its ideas. Also, admin (lecturer) use cases were left out of the survey as it was difficult finding lecturers to participate due to both of our busy schedules.

A questionnaire approach is utilized in the investigation rather than an interview method. The rationale behind this is that, the survey will have a relatively sizeable number of subjects and interviews become more cumbersome as the number of subjects increases. While a web-based questionnaire would effectively and efficiently reach widely distributed respondents, because the website is not live yet, this would involve sending the source code to respondents and having them build and run it themselves. This is

unreasonable as it would be a massive inconvenience and thus, discourage people from participating in the survey.

For this reason, a "mixed approach" is adopted, where each subject is approached directly (almost in an interview-like manner), and presented with a laptop running the website with which they can try out the website, and then handed a questionnaire afterwards. This is good because the subjects can ask questions and can also be observed to witness their facial reactions and body language which is information that cannot be gleaned from a questionnaire alone.

5.2.2 Organization of Survey

The questionnaire used in the survey consisted of a short introduction explaining its goals followed by two sections with a set of questions each: *Website Design* and *Website Functionality*.

In the *Website Design* section, respondents were asked to provide answers for a number of questions using a discrete 4-point-scale (Very good, Good, Bad and Very bad) regarding matters of the quality of the website's design. In other words, this section assesses the quality of the system as a website without taking any of its features into account.

In the *Website Functionality* section, respondents were asked to provide answers for a number of yes-no questions as well as some discrete answer questions similar to those in the *Website Design* section. A number of the yes-no questions also included space for an open-answer response to allow respondents to expatiate on their answer choice. The questions in this section are concerned with matters regarding the usefulness of the website's features and the quality of their implementation.

5.2.3 Survey Administration and Response

The survey was carried out in the lounge area near the reception in the University of East Anglia library. This location was selected not only because it was easily accessible for the surveyors, but because a wide variety of students in different schools with dif-

ferent technological backgrounds can be found in the library. This allowed for a diverse sample of respondents for the survey.

22 respondents were found to participate in the survey. The respondents were all from different schools, doing different courses and at different stages of their degree . Of the 22, 11 were male and 11 were female. The mean age was 20.5. The respondents were from 9 schools - Law (2), International Development (5), Computing (2), Psychology (4), Economics (1), History (1), Literature and Creative Writing (3), Pharmacy (2) and Biological Sciences (2). Also, of the 22 respondents, 2 of them were postgraduate students, 8 were third-year students, 5 were first-year students and the remaining 7 were second-year students.

Table 2 shows the response breakdown of the survey. The table shows a question, its possible responses and the percentage of the respondents who selected that response.

5.2.4 Results

Overall, the feedback from respondents was positive. Some of the respondents responded negatively to the user interface design but it was a very small number (9%). This was probably due to subjective and personal preferences rather than problems with the interface.

Also, some of the respondents provided responses for the open-answer parts of the yes-no questions. One respondent said that they found the activity feed useful because it allowed them to keep track of their progress while another one said that it would refresh their memory on their previous activity after they had not used the website in a while. Another open-answer response received was for the “Are you satisfied with the module management system” question. The user answered no and expanded on their choice thusly. The respondent said that while the list of modules shown when adding modules to your list implements pagination, it would be nice to be able to search for a module to add, instead of scrolling or going from page to page when more modules are added to the website.

Question	**“Yes”**	**“Very Good”**	**“Good”**	**“Bad”**	**“Very Bad”**
How attractive and engaging was the User Interface	-	22.72%	68.18%	9.09%	0.0%
How responsive was the website	-	86.36%	13.63%	0.0%	0.0%
How quick was the speed of the load times	-	9.09%	90.9%	0.0%	0.0%
How good was the navigation across the website	-	68.18%	22.72%	0.0%	0.0%
Do you think the “Activity Feed” is useful	90.9%	-	-	-	-
Are you satisfied with the module management system	72.72%	-	-	-	-
Was the difference between Practice and Gradeable tests clear to you	95.4%	-	-	-	-
How did you find the Discussion Board experience	-	40.9%	59.1%	0.0%	0.0%
Do you think the “Tracked Topics” feature of the Discussion Board is useful	100.0%	-	-	-	-

Table 2: Questionnaire Response Statistics

In addition to this, user reception for the discussion board feature was wildly positive, split almost in half with both sides saying it was “very good” and “good”. 100% of respondents also said they liked the “tracked topics” feature of the discussion. This is interesting because it seems to hint at the fact that the discussion board feature is a very popular feature among the users. I imagine the respondents were excited by its social nature and its encouragement of collaboration.

One important thing that can be learned from the survey is that the student users seem to respond very positively to features with social and collaborative aspects. In the next iteration, it might be worth making an effort to incorporate some of these aspects into the test taking process.

6 Evaluations

6.1 Outcome

One way of evaluating a project is by checking the met requirements against the specified requirements list (or MoSCoW). For the project to be a success, all the "Musts" requirements must have been fulfilled. To achieve a good level of usability, majority of the "Shoulds" should also have been fulfilled. Table 3 shows the outcome of the project. From the table, it can be seen that all of the "Musts", "Shoulds" and "Coulds" were satisfied, making the project a success based upon the requirements laid out at the start. The website and app were tested and the web pages were validated without any major flaws. Any errors discovered were fixed so that the system ran smoothly.

The project was finished and delivered on time, making it a success in that sense as well. However, after the user testing and usability survey, light was shed on areas that could be improved. These will be highlighted in the following *Improvements* section.

6.2 Improvements

The first thing that can be improved is the module management system. In response to an open-answer question in the survey, a respondent suggested that a search feature be added to the module management feature. When the website displays a list of the modules available for them to enroll on, instead of scrolling through pages to find a module, a search feature could be implemented, allowing the user to quickly find the module on which they wish to enroll. This search feature could also be added to the user dashboard, so that the user does not have to first click on the "Manage Modules" link before adding

Table 3: Project Outcome

Requirement	Met or Not Met
Login and Registration facilities	Met
Students able to enroll on modules	Met
Students access to a bank of questions for enrolled modules	Met
Lecturers able to add multiple-choice-questions	Met
Students able to answer questions	Met
Feedback on answered questions	Met
Student progress persistence	Met
League tables/rankings	Met
More question types in addition to multiple-choice	Met
Forum/Message board	Met
User activity log	Met
Accompanying mobile app	Met
User avatars	Not Met
Student created questions	Not Met

a module. That way, they could just search for it from their dashboard and add it to their module list.

Another improvement that could be made is to make the test-taking process more social and collaborative. The discussion board feature was received very well by the users in our usability survey. From the survey, it can be deduced that the student users respond positively to social and collaborative features. One way to take advantage of this is by taking a page out of *MyCQs* book. From the similar system analysis, it was found that MyCQs allows users to create their own tests and share them with other users. This feature could be incorporated into the Module Support website to make it more social and collaborative.

Finally, more types of questions could be added. At the moment, the module support

website (and mobile app) only offer three kinds of questions - multiple-choice, true-or-false and open-answer questions. More kinds (such as multiple-choice-multiple-answer and fill-in-the-blank) could be added as well for a richer experience.

7 Conclusion

The aim of the project was to create a website in support of a module and act as a revision aid for students. Overall, the project was a success. An AMP development stack and test-driven development was used to achieve this aim. In addition to this, an accompanying mobile app for the iOS platform was also created to complement the website.

The delivered website allows users to join and log in, enroll on modules and take tests for modules on which they are enrolled. This provides an opportunity for students to practice and test their knowledge in revision. The delivered website also allows users to create topics and post messages on a discussion board. This allows students to share and gain information as well as help each other to revise. The delivered website also displays performance in tests in a leaderboard ranking, thus allowing students and lecturers alike to track students' progress. The delivered mobile app complements the website by providing the core features of the website on a portable device, allowing users to have access to the core features at all times.

In conclusion, this project proves that Computer Aided Learning (CAL) is a viable and useful option in higher education by delivering software (both the website and mobile app) which successfully aids student revision.

References

Beatty, K. (2003). *Teaching & Researching: Computer-Assisted Language Learning*. Pearson Education.

Clarke, J. (1993). Cognitive style and computer-assisted learning: problems and a possible solution. *http://researchinlearningtechnology.net/index.php/rlt/article/view/9472*.

Clayden, G. and Wilson, B. (2009). Computer-assisted learning in medical education. Medical Education, Issue 22.

Daniel, J. (1998). *Mega-universities and Knowledge Media*. Psychology Press.

Gaile, C. and Reiff, J. (1994). Individual constructivist teacher education: Teachers as empowered learners. Teacher Education Quarterly, v21 n3 p27-38 Sum 1994.

Goh, D., Ang, R., Theng, Y.-L., and Lim, E.-P. (2005). Geogdl: a web-based approach to geography examination revision. Computers & Education 45 (2005) 57âĂŞ73.

Greenhalgh, T. (2001). Computer assisted learning in undergraduate medical education. *http://dx.doi.org/10.1136/bmj.322.7277.40*.

Haag, M., Maylein, L., Leven, F., Tonshoff, B., and Haux, R. (1999). Web-based training: a new paradigm in computer-assisted instruction in medicine. Int J Med Inform.

Kemmis, S., Atkin, R., and Wright, E. (1977). *How Do Students Learn?* Centre for Applied Research in Education, University of East Anglia.

Kogan, N. (1971). Educational implications of cognitive styles. Psychology and Educational Practice, Illinois.

Ltd, M. (2013a). Mycq. *https://itunes.apple.com/gb/app/mycqs-multiple-choice-questions/id921380709?mt=8*.

Ltd, M. (2013b). Mycq. *https://mycqs.com/about*.

Oldfather, P. (1993). Drawing the circle: Collaborative mind mapping as a process for developing a constructivist teacher preparation program. *http://www.teqjournal.org/backvols/1994/21_3/S94_oldfather_mann.pdf*.

Phillips, R. (1996). *Developers guide to Interactive Multimedia*. Computing Centre, Curtin University of Technology.

Richardson, V. (1997). *Constructivist Teacher Education*. Psychology Press.

Teets, S. and Starnes, B. (1996). Foxfire: Constructivism for teachers and learners. *http://www.tandfonline.com/doi/abs/10.1080/01626620.1996.10462831#.VEZ8LsFho20*.

Wikipedia (2000). Blackboard learnng system. *http://en.wikipedia.org/wiki/Blackboard_Learning_System*.

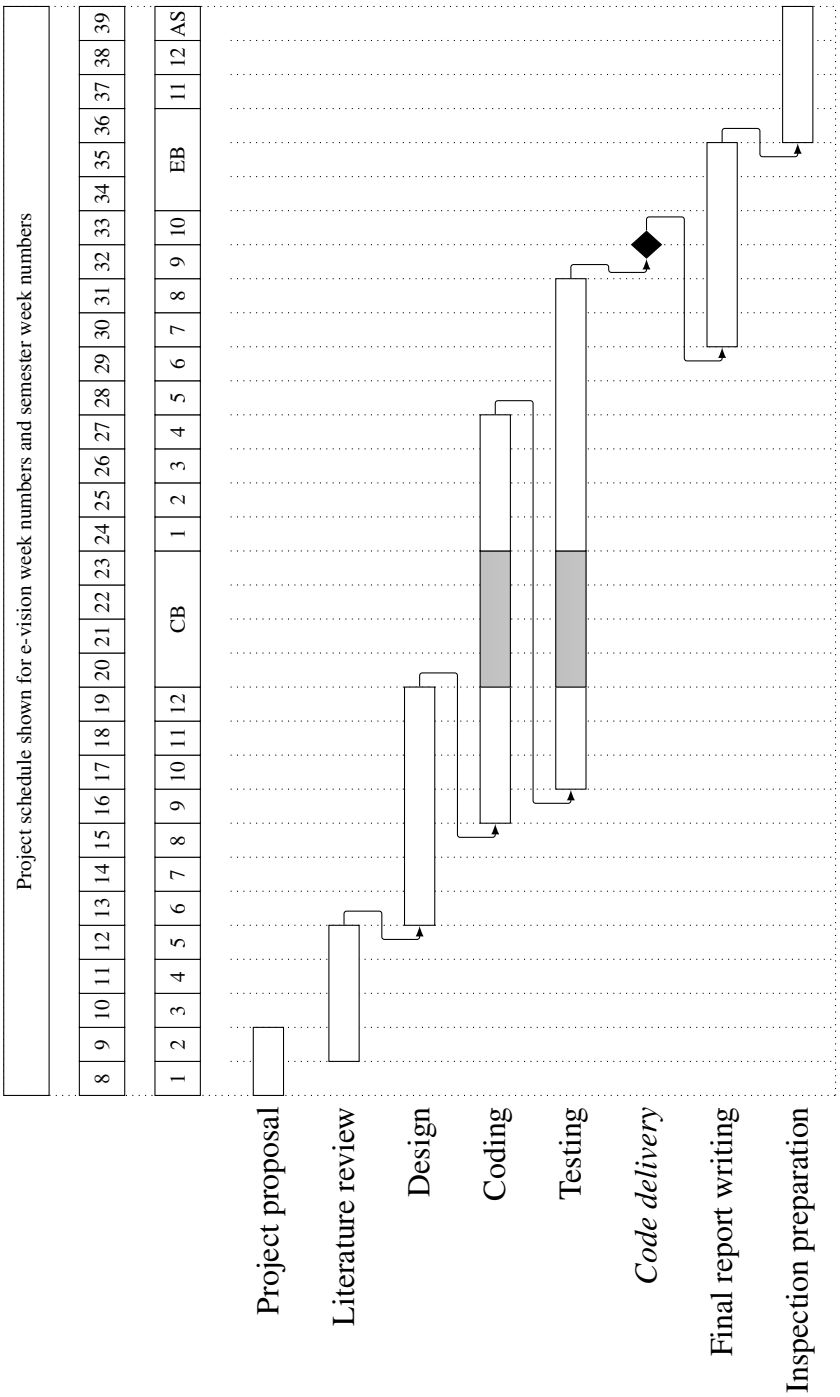


Figure 12: Project Gantt chart