**NATIONAL COLLEGE OF IRELAND**

**Technical Report**



|  |  |
| --- | --- |
| **Name** | Andrew Monaghan |
| **Student No.** | 09102825 |
| **Email** | andrew.monaghan@student.ncirl.ie |
| **Degree** | BSc (Hons) in Business Information Systems |
| **Date** | 18th May 2014 |
| **Supervisor** | Paul Hayes |

**Table of Contents**

[1 Executive Summary 4](#_Toc388091946)

[2 Introduction 5](#_Toc388091947)

[2.1 Background 6](#_Toc388091948)

[2.2 Requirement Elicitation Questionnaire 6](#_Toc388091949)

[2.3 Aims 8](#_Toc388091950)

[2.4 Technologies 9](#_Toc388091951)

[2.5 Structure 10](#_Toc388091952)

[3 System 11](#_Toc388091953)

[3.1 Requirements 11](#_Toc388091954)

[3.1.1 Functional Requirements 11](#_Toc388091955)

[3.1.2 User Requirements 13](#_Toc388091956)

[3.1.3 Usability Requirements 14](#_Toc388091957)

[3.1.4 Performance/Response Time Requirements 14](#_Toc388091958)

[3.1.5 Availability Requirements 14](#_Toc388091959)

[3.1.6 Usability Requirements 14](#_Toc388091960)

[3.1.7 Reliability Requirements 14](#_Toc388091961)

[3.2 Design and Architecture 15](#_Toc388091962)

[3.2.1 System Architecture 15](#_Toc388091963)

[3.2.2 Design Structure 16](#_Toc388091964)

[3.2.3 Development Folder Structure 17](#_Toc388091965)

[3.2.4 Database Design 18](#_Toc388091966)

[3.2.5 Database Relationships 19](#_Toc388091967)

[3.2.6 Database Constraints 20](#_Toc388091968)

[3.3 Implementation 21](#_Toc388091969)

[3.3.1 Open Source Code 21](#_Toc388091970)

[3.3.2 User Login 22](#_Toc388091971)

[3.3.3 User Registration 23](#_Toc388091972)

[3.3.4 File Sharing 24](#_Toc388091973)

[3.3.5 File Searching and Download 29](#_Toc388091974)

[3.3.6 Integration of Open Source Study Forum 33](#_Toc388091975)

[3.3.7 Integration of Open Source Chat Room 34](#_Toc388091976)

[3.3.8 Administrator Access Only – Manage Classes and Manage Users 36](#_Toc388091977)

[3.3.9 Common Functions 38](#_Toc388091978)

[3.3.10 Security Features 39](#_Toc388091979)

[3.4 Testing 43](#_Toc388091980)

[3.4.1 Unit Testing 43](#_Toc388091981)

[3.4.2 Integration Testing 44](#_Toc388091982)

[3.4.3 System Testing 45](#_Toc388091983)

[3.4.4 Client Testing 46](#_Toc388091984)

[3.4.5 End-User Testing 49](#_Toc388091985)

[3.5 Graphical User Interface (GUI) Layout 50](#_Toc388091986)

[4 Conclusions 52](#_Toc388091987)

[5 Further Development 53](#_Toc388091988)

[5.1 Proposed Business Model 53](#_Toc388091989)

[6 References 54](#_Toc388091990)

[7 Appendix 55](#_Toc388091991)

[7.1 Test Cases 55](#_Toc388091992)

[7.2 UAT Questionnaire Results 58](#_Toc388091993)

[7.3 Project Proposal 67](#_Toc388091994)

[7.4 Requirement Specification 75](#_Toc388091995)

[7.5 Monthly Journals 124](#_Toc388091996)

[7.6 User Manual 135](#_Toc388091997)

# Executive Summary

The objective of this project is to create a fully functioning and operational web application that can be used by a college to promote collaborative learning. It will provide the students with an online platform in which they can upload and share study resources, ask questions and discuss college topics in both a forum and a chat room and also access study resources.

The web application will aim to provide the following functionality:

* advanced file sharing (private, public or specific user sharing)
* advanced file searching and download (search by multiple criteria)
* study forum to discuss college related topics or ask for advice on projects
* chat room to allow students discuss college topics in real-time
* access to study tips and advice
* administration section to provide the administrator (e.g. college lecturer) with various options for monitoring and controlling the web application

The target audience for this web application are small colleges from a non-technical background who do not have the resources or technical capacity to operate a mainstream Learning Management System (e.g. Moodle) but who still wish to provide their students with a platform for collaborative learning.

The impetus for this project stems from a personal interest in the e-learning arena and from personal experience being a class representative, becoming aware of the desire for students to have a more collaborative learning experience.

Having successfully achieved these objectives and completed end-user testing with students from a variety of third level colleges, the website is now hosted live on the internet (<http://studybettertogether.com>).

# Introduction

Collaborative learning can be broadly defined as a “situation in which two or more people learn or attempt to learn something together.” From a practical perspective it enables the students study together, learning from one another’s strengths and skills through asking questions, discussing projects and sharing study notes or other study resources.

This project aims to create a website called studybettertogether.com which can facilitate this type of practical collaborative learning – file sharing and online communication. This website aims to promote collaborative learning among students by providing students with an online platform in which they can upload and share study resources, ask questions and discuss college topics in both a forum and a chat room and also access study resources.

The target audience of this web application are small colleges from a non-technical background who do not have the resources or technical capacity to operate a mainstream Learning Management System (e.g. Moodle) but who still wish to provide their students with a platform for collaborative learning. As such, the website has been developed with a simple and user friendly approach designed to be used by students with no previous technical knowledge.

The client for this project is Dr. Mimi Kelly who delivers both undergraduate and postgraduate modules in History, Sociology and Study Support for a third level College in Dublin named the Irish Bible Institute. Mimi was instrumental in the requirements gathering, planning, design and testing phases of this project and though several meetings with the developer has helped tailor the website to be used by non-technical users for the practical purposes of collaborative learning.

This scope of this project has changed slightly since the original project proposal in that the original plan was to create both a collaborative learning website and also a mobile web application. A decision was made between the developer and the client during the requirement elicitation phase to drop the mobile web application from this project and focus solely on the website. While the mobile web application was originally identified as a key component, a decision was made not to implement it as to do so would only be duplicating the project functionality albeit in another technology. Narrowing the scope of the project has also allowed the client to add further functionality and ensured the developer could implement this new functionality to a high standard.

## Background

The impetus of the project stems from a personal interest in the e-learning arena and as a result of feedback from students while being a class representative in a college setting

As a class representative the developer received significant feedback from other students who expressed a need for an online platform where students could discuss upcoming exams or projects and also share files with each other. Interestingly these were students who had access to Moodle and a college with excellent IT facilities but their need for a student oriented collaborative learning platform was still not being met.

One situation that the developer noticed where this website would be beneficial is providing a study forum where students can discuss revision topics for an upcoming exam. Currently what is happening is students are emailing each other questions and tips for an upcoming exam. Soon however these email threads become very large (70+ emails) which makes searching through the threads and replying to specific questions a tedious task. Further problems arise when there are multiple email threads, each with different recipients and discussing different aspects of the upcoming exam. The developer believes it would prove beneficial to move this exam discussion away from email threads into a study forum where topics could be assigned, sorted and searched in a user-friendly and efficient manner.

In addition to the developer’s interest in this project, the client also was extremely keen on the idea stating that she believed the project could be of great benefit in helping students study together. The client was a strong advocate that the website be designed for a non-technical audience as many of her own students had little or no IT experience. She also identified a gap that exists in the e-learning market which is small colleges (less than 100 students) who do not have the technical expertise to utilise mainstream Learning Management Systems.

## Requirement Elicitation Questionnaire

To further investigate if there was a need for this project, the developer conducted a requirement elicitation questionnaire among a sample of fifteen third level college students during the requirements gathering phase of this project. Below is an overview of the results, the full questionnaire results including graphs can be found in the Requirement Specification documents **appendix 7.4.**

**Collaborative Learning**

* 87% of respondents thought that the proposed StudyBetterTogether website would be beneficial in helping them study better.
* Previous experiences of collaborative learning were mixed.
  + 20% - low benefit
  + 33% - medium benefit
  + 27% - high benefit
  + 20% - very high benefit

**Sharing Files**

* 93% of respondents had previously shared learning resources (study notes, essays, projects) with other students and had found it helpful to some degree.
  + 33.3% - somewhat helpful
  + 60% - very helpful
* The primary means of file transfer used to share these learning resources was:
  + Email – 50% | Dropbox – 28.5% | Google Drive – 21.5% | Moodle – 0%

**Study Forum**

* 87% of respondents had previous difficulties in not receiving answers to their college related questions from either lecturers or other students.
* 100% of respondents stated that a study forum to allow students discuss college related topics would be beneficial to some degree.
  + 33.3% - medium benefit
  + 13.3% - high benefit
  + 53.3% - very high benefit

**Priority of Functionality**

The students ranked the proposed functionality based on what they would find most useful. The results of this, from most to least useful were:

1. Upload and share files with only specific users that you can choose
2. Upload and share files with other users
3. Upload files to a personal folder so that only you have access
4. Access to detailed study tips, hints and advice
5. Access to a study forum where you could post questions, search questions and reply to questions
6. Access to advice for handling exam pressure
7. Create a quiz that requires typing in the answer.
8. Create a MCQ quiz and share this with other users

This questionnaire proved extremely useful in the requirement elicitation phase by first demonstrating the usefulness of this project idea (87% of respondents stating that it would be beneficial) and also providing some very interesting insights which were used to help form the functional requirements. Having each respondent rank the proposed functionality according to its usefulness was very helpful in prioritizing the functional requirements.

It is also interesting to note that while 93% of respondents found sharing files for learning purposes helpful, none of these respondents had used Moodle to do so, despite using Moodle as their learning management system on a daily basis. This supports the claim that Moodle and other similar learning management systems, while very useful for teacher to student interaction, are not used for student to student interaction.

## Aims

This project aims to create an operational and secure website which will provide the following key features:

1. **User registration and login**

Students will be able to create their own account based on a registration code provided by the client. Once registered, students can log into the website using their username and password.

1. **Advanced file sharing**

Students will be able to upload and share files in three different ways:

* Public – all users on the site will have access to the files
* Private – only the student who uploaded the file can access it
* Specific User – the student can chose from other students in his class who he / she wants to share the files with

1. **Advanced file searching and download**

Students will be able to search for files by multiple search criteria including the sharing status, file type, file name, file description or file subject.

1. **Study forum**

An open source forum will be successfully integrated into the website which will allow students post questions, search through previous posts and reply to posts.

1. **Chat room**

An open source chat room will be successfully integrated into the website which will allow students who are logged into the website, chat in real time.

1. **Access to study tips and advice**

The website will contain study tips and links to study resources

1. **Manage users (administrator access only)**

The client will be able to view and update user details, change user passwords and assign users to a different class if necessary.

1. **Manage classes (administrator access only)**

The client will be able to create new classes and set their registration code, see how many students are assigned to each class, update class details and also delete classes.

1. **Website hosted online and secure**

The website will be hosted online with a secure and functioning user account system in place so only registered users can access the above features.

## Technologies

This project was developed using the following technologies:

* **PHP** is the server side language responsible for most of the websites functionality including the registration, login, file sharing and searching, forum and chat room features. PHP is also used throughout the website for server side security, session control and interacting with the MySQL database.
* **MySQL** is the relational database system used by the website to store data. This includes information of registered users, uploaded files, study forum posts and chat room conversations.
* **PhpMyAdmin** is an open source tool writing in PHP which provides the user with a GUI for interacting with MySQL. This was extensively in his project to create the MySQL tables and relationships.
* **HTML** is a client side web structuring language and was used to create the front end structure of the website. All HTML code (except the forum and chat room) was written by the developer.
* **CSS** is a client side style sheet language and was used to create the design and layout of the website. 90% of the CSS used in this website was written by the developer.
* **JavaScript** is a client side scripting language used in the website for client side validation and in the study forum and chat room.
* **jQuery** is a popular JavaScript library mainly used in this website for the index page scrolling banner, the specific user sharing dropdown feature and also in the study forum and chat room.
* **Eclipse IDE** is an open source development environment used to create the website.
* **Apache Hosting Server (provided by HostingIreland.ie)** is the server hosting the website. It provides the functionality to incorporate PHP and MySQL into the website.
* **My Little Forum** is the open source forum that was integrated into the website
* **AJAX Chat** is the open source chat room that was integrated into the website
* **GitHub** is an online repository service and was used throughout this project by the developer for the purpose of version control. This provided the security of being able to back up and restore code in the event of a hardware failure. It also provided the developer with the ability to work on the development of the website from multiple locations.
* **Survey Monkey** is an online survey tool which was used two create and distribute two questionnaires throughout this project, a requirements elicitation questionnaire at the start and an end-user acceptance testing questionnaire at the end.

## Structure

This document is structured in the following order:

* **Section one** contains an executive summary of the project
* **Section two** introduces the project and contains an overview of the objectives the developer set out to achieve along with the technologies he used to achieve these.
* **Section three** contains five sub sections which describe the system development lifecycle.
  + **Section 2.1** describes the functional and non-functional requirements of the project along with detailing how these requirements have evolved since the original requirement specification document.
  + **Section 2.2** describes how the project was designed, including the system architecture, the development folder structure and the database design.
  + **Section 2.3** outlines how the project was developed. This includes samples of the code used in creating key features of the website and an explanation on what the code does.
  + **Section 2.4** describes how the websites was tested through unit, integration, system, client and end-user testing.
  + **Section 2.5** provides screenshots of the main website features. Further screenshots can be found in the user manual contained in **appendix 7.6.**
* **Section** **four** discusses the main conclusions that the developer arrived at upon completion of this project. These include the advantages/disadvantages, opportunities and limits of the project.
* **Section five** outlines what further development could be completed on this project with further time and resources
* **Section six** contains the references used in creating this technical report along with all links to all resources used in the development of this project.
* **Section seven** contains the appendices which include any additional information relating to this project but not included in this technical report.

# System

## Requirements

This section describes both the functional and non-functional requirements of my project.

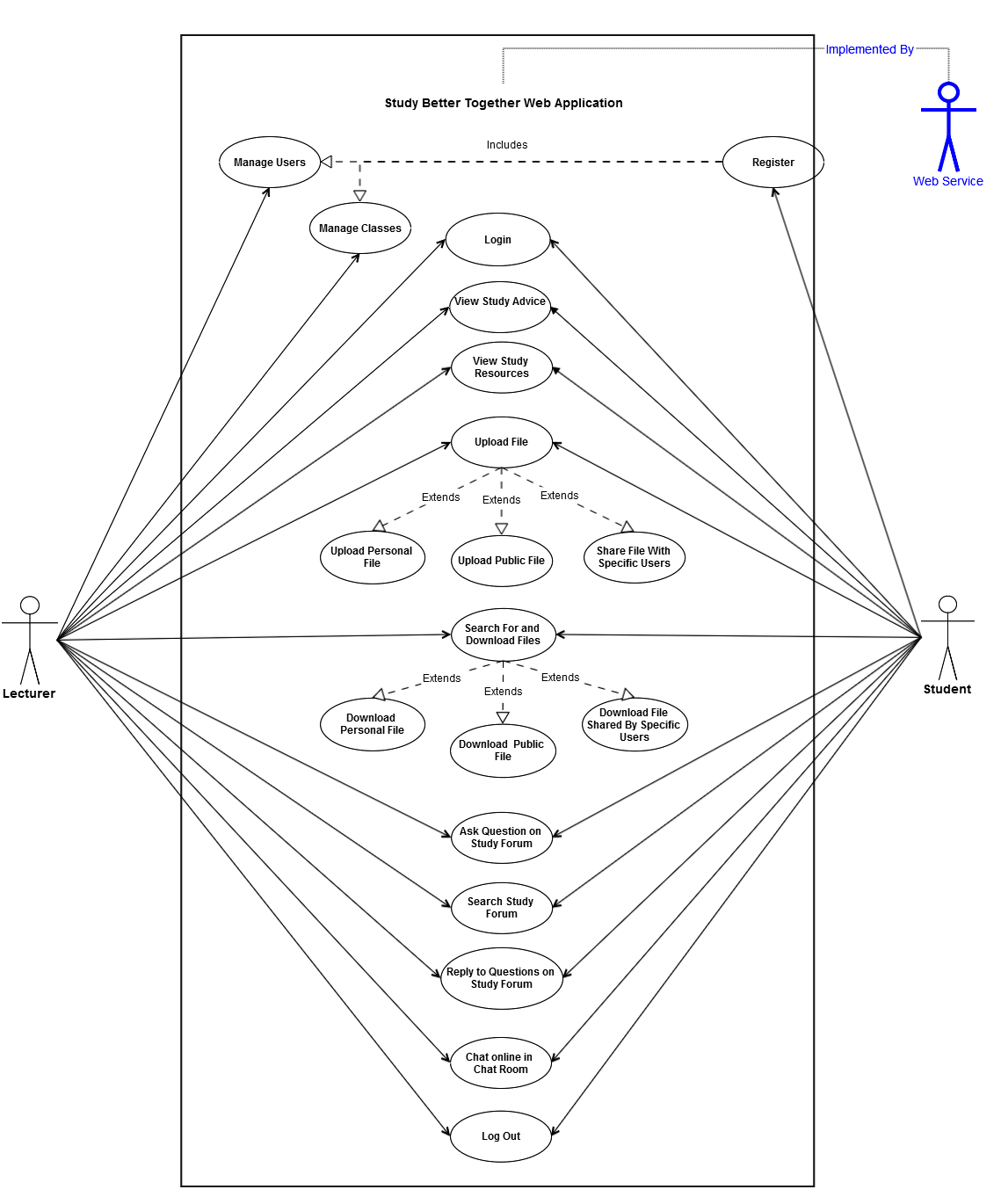
### Functional Requirements

The original requirement specification document (**appendix 7.4**) identified 15 functional requirements, all of which have been successfully implemented by the developer. Two optional requirements relating to a quiz maker were not implemented however. The reason for this is due to the low priority rating they received in the requirement elicitation questionnaire and also to allow development of two new requirements the client added after the requirement specification document was submitted. These new requirements are the inclusion of an online chat room and the ability for the client to create classes and have students register according to their class.

**Full List of Functional Requirements**

|  |  |
| --- | --- |
| Requirement 1 | <Manage Users> |
| Requirement 2 | <Student Registration> |
| Requirement 3 | <Logging into the Application> |
| Requirement 4 | <Upload a Personal File> |
| Requirement 5 | <Upload a Public File> |
| Requirement 6 | <Upload a File and Share it with Specific Users> |
| Requirement 7 | <Search For and Download Personal Files> |
| Requirement 8 | < Search For and Download Public Files > |
| Requirement 9 | < Search For and Download Files Shared by Specific Users > |
| Requirement 10 | <Ask a Question on the Study Forum> |
| Requirement 11 | < Search for Questions and Answers on the Study Forum> |
| Requirement 12 | < Reply to a Question on the Study Forum> |
| Requirement 13 | <View Study Tips and Advice> |
| Requirement 14 | <View Study Resources> - *Previously “View Exam Preparation Tips & Advice”* |
| Requirement 15 | <Log out of Application> |
| Requirement 16 | <Create a New Quiz> *- Optional Requirement Not Implemented* |
| Requirement 17 | <Take a Previously Created Quiz> *- Optional Requirement Not Implemented* |
| Requirement 18 | <Chat online in the Chat Room> *- Additional Requirement added by Client* |
| Requirement 19 | <Manage Classes> *- Additional Requirement added by Client* |

**Note:** Detailed use case descriptions for each requirement can be found in the requirement specification document (**appendix 7.4**)

**Overall Use Case Diagram **

### User Requirements

1. The overall aim of this project should be to provide a user friendly and accessible platform that will assist students in studying collaboratively. The website should be built in such a way so that it can be used by non-technical users. The style and layout of the website should be clean, minimal and clutter free.
2. The website should allow the client to manage the users of the website without having to create new accounts for each user. This means the client wants to be able to see who has created an account on the system and be able to edit and delete their accounts if needed, but does not want to be responsible for creating a new account for each student. It should be the student’s responsibility to register and create their own accounts.
3. The website should allow the client to create new classes and also create a registration code for each class. Students will need this registration code in order to register for the website. The client should be able to see the number of students assigned to each class, edit class details and change the registration code if desired.
4. The website should contain detailed study hints, tips and advice along with similar advice for helping students cope with exam pressure. This should be displayed in a manner that is keeping with the websites clean and minimal style.
5. The website should allow students upload and share study resources with each other. These resources could be documents, essays, projects, revision notes or summarized chapters of books that the students have written. The students should be able to choose if they want to share files with everyone or with only specific people. Also there should be the functionality to allow a student upload a file but choose not to share it with anyone (i.e. keep it private).
6. The website should facilitate conversation on any college / study related topics by providing an online study forum for the users to ask questions. The forum should be built in a simple and basic fashion in keeping with the websites clean and minimal style.
7. The website should facilitate instant communication for students currently logged into the website through the use of a chat room feature. The chat room should be built in a simple and basic fashion in keeping with the websites clean and minimal style.
8. The lecturer should have the same access to the file sharing, study forum and chat room features as the students. This will allow for times when the lecturer wants to share files with her students or else she wants to either start a conversation or answer a student’s question on the study forum.

### Usability Requirements

The system should be user friendly and require no previous technical knowledge to use all provided functionality. Each web page should ideally focus on one main function. If a single web page contains two or more primary functions and these can be easily separated into two separate web pages this should be done.

The design, colour scheme and layout of the web application should be simple, minimal and clean in keeping with the user requirement for a ‘clutter free’ website.

### Performance/Response Time Requirements

All web pages contained in this application should have a fast load time within the client’s browser in keeping with end user expectations. A maximum load time of 2.5 seconds 90% of the time is the end goal.

The system should provide fast data transfer speed as data is sent from the client browser to the MYSQL database via the web server and then back to the client browser. This data transfer should appear seamless and instant to the end user.

### Availability Requirements

The system should be available to users 24/7 with little or no downtime. If the system is unavailable to the users, a detailed error message should be displayed along with accurate information on when the system will be next available.

### Usability Requirements

The system should implement best practice in securing all user data stored in the database. User passwords should be encrypted and salted and should not be stored in clear text format in the database.

Application level security should be implemented to protect the system against SQL injection, cross-site scripting, cross-site request forgery and other web application attacks. This added security should not hinder the systems performance in any way.

### Reliability Requirements

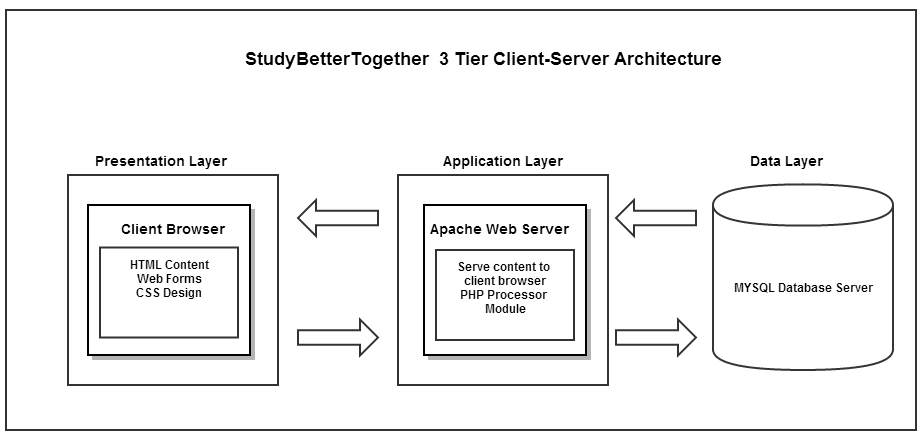
The system should be able to handle concurrent access from multiple users without any significant decrease in performance, security or availability.

## Design and Architecture

### System Architecture

This system will use the Apache Web Server to serve up the HTML content to the clients browsers and handle security and memory issues. The Apache Web Server will be configured to use a PHP processor module that will interpret the PHP code used for all server side programming of this application.

The following diagram shows the 3-Tier client-server architecture that this web application will utilize.

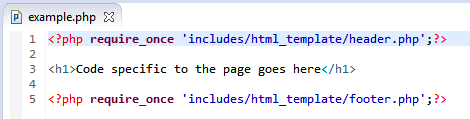


### Design Structure

This website was a new development from the beginning, and a decision was made to write all HTML code instead of using a front-end template. In order to avoid rewriting the same code in every front-end page, the developer created a HTML template and divided this between two PHP files which are included in every front-end page (except for index.php and register.php) through the use of two PHP require\_once statements.

* **Header.php** – contains all header information, style sheets, opening HTML tags and the navigation menu.
* **Footer.php** – contains the closing HTML tags and footer information.

Using this design structure greatly reduces the amount of duplicated code resulting in efficient development and the rapid creation of new pages as demonstrated in the below example.

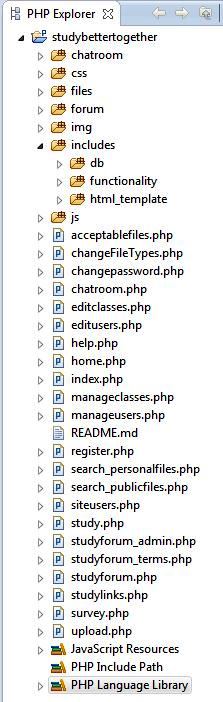
****

These three lines of code result in the below page when opened in a browser.

****

### Development Folder Structure

To provide efficient separation and organisation for development content, a folder structure was created that separated client-side content from server-side content and also allocated separate files (CSS, JavaScript, images, uploaded files. the open source forum and the open source chat room) to their own corresponding folder. This folder structure is shown below.



Multiple **administration options** for configuring the Study Forum

Front end design template

JavaScript files

Open source chat room files

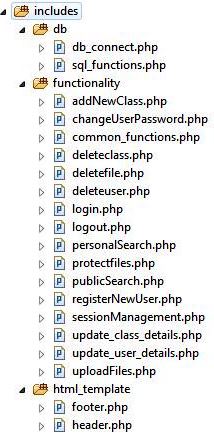
CSS style files

Database connection files

Image files

Open source study forum files

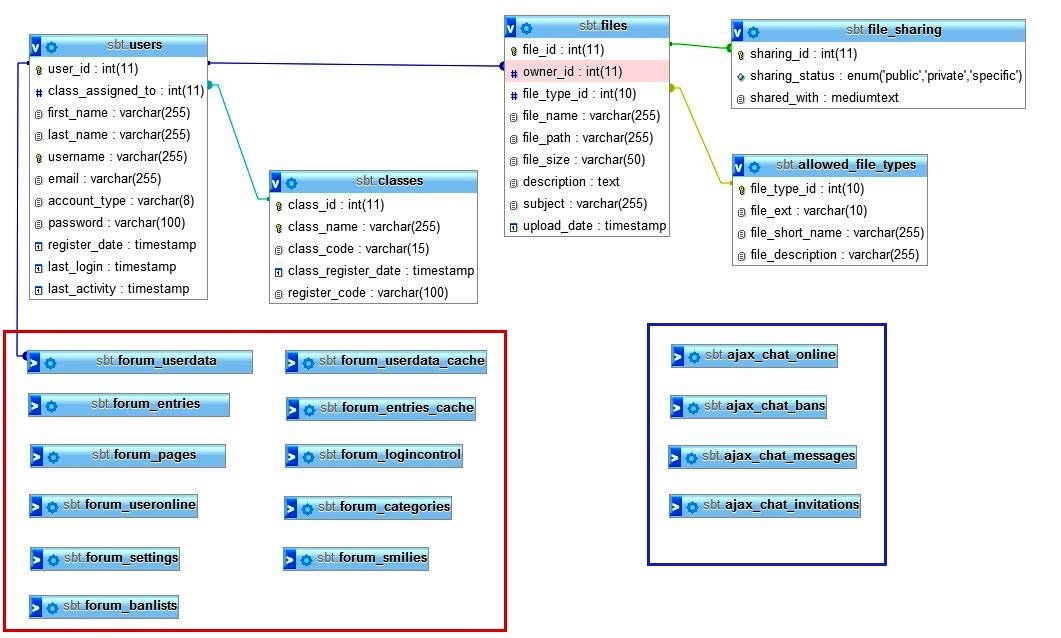
Files uploaded by users



Client-side files. These are the files that the users see and interact with

### Database Design

The below picture displays the database structure of the website, including five tables created by the developer, eleven tables relating to the study forum (contained in red box) and four tables relating to the chat room (contained in blue box).



***Tables pertaining to the chat room***

***Tables pertaining to the study forum***

**Description of Tables created by the Developer**

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| **users** | Contains all information relating to the website users including their name, email, password, register date and date of last activity. |
| **classes** | Contains information relating to the classes (degree programmes) which students are registered to. |
| **files** | Contains information on the files uploaded to the website including the file name, size, path, description and date of upload. |
| **file\_sharing** | Contains information on the sharing status of each file uploaded to the website including the sharing status and who the file is shared with. |
| **allowed\_file\_types** | Contains information on the file types allowed to be uploaded to the website including the file extension, name and description. |

### Database Relationships

There exist various relations between the tables which are outlined below.

1. **Users.user\_id to files.owner\_id**

A one-to-many relationship exists through the primary key users.user\_id and the foreign key files.owner\_id. One user can be the owner of many files.

1. **Users.user\_id to forum\_userdata.user\_id**

A one-to-one relationship exists through the primary key users.user\_id and the primary key forum\_userdata.user\_id*.*

1. **file\_sharing.sharing\_id to files.file\_id to**

A one-to-one relationship exists through the primary key files.file\_id and the foreign key file\_sharing.sharing\_id. One file can have one and only one sharing status.

1. **allowed\_file\_types. file\_type\_id to files.file\_type\_id to**

A one-to-many relationship exists between the primary key allowed\_file\_types.file\_type\_id and the foreign key files.file\_type\_id in that one file type can belong to multiple files. For example the one file type (.doc) can belong to many different files, or another way of putting this is there can be many .doc files.

1. **Classes.class\_id to users.class\_assigned\_to**

A one-to-many relationship exists between the primary key classes.class\_idand the foreign key users.class\_assigned\_to in that one class can belong to multiple users or put another way, many students can belong to the one class.

### Database Constraints

In order to maintain referential integrity and avoid using unnecessary application code, the following database constraints have been created between the database tables.

1. **users.user\_id to forum\_userdata.user\_id**

SQL Statement

ALTER TABLE `forum\_userdata` ADD CONSTRAINT `forum\_userdata\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

Description

If the primary key users.user\_id is deleted than the record containing the foreign key forum\_userdata.user\_id field will also be deleted. This means if a user is deleted from the main users table than the corresponding record for that user will automatically be deleted from the study forum table.

1. **users.user\_id to files.owner\_id**

SQL Statement

ALTER TABLE `files` ADD CONSTRAINT `files\_ibfk\_1` FOREIGN KEY (`owner\_id`) REFERENCES `users` (`user\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

Description

If the primary key users.user\_id is deleted than the records containing the foreign key files.owner\_id field will also be deleted. This means if a user is deleted from the website, than all records for the files that the user has uploaded will also be deleted.

1. **file\_sharing.sharing\_id to files.file\_id to**

SQL Statement

ALTER TABLE `file\_sharing` ADD CONSTRAINT `file\_sharing\_ibfk\_1` FOREIGN KEY (`sharing\_id`) REFERENCES `files` (`file\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

Description

If the primary key files.file\_id is deleted than the record containing the foreign key file\_sharing.sharing\_id will also be deleted. As every file must have a file sharing status contained in the file\_sharing table, this constraint ensures that when a file is deleted, its file sharing status in the file\_sharing table will also be deleted.

These three constraints have a cascading effect so that if a user is deleted from the main users table, they are also automatically deleted from study forum users table and any files they own are automatically deleted from the files table which in turn deletes the corresponding file sharing record for the deleted files.

## Implementation

To implement this project the developer created 42 PHP files, 1 CSS file, 2 JavaScript files and a large SQL database in addition to incorporating two open source plugins - the study forum and the chat room. Due to the large amount of code written by the developer it is outside the scope of this document to show and explain all code used in this project.

Instead of describing all code used to implement this project, a selection of code from each of the main website features will be shown and explained. However, all code is available for viewing on the public GitHub repository: <https://github.com/andrewmgh/studybettertogether>

The website is also hosted live on the internet and can be tested using the following login details:

* **URL**: <http://studybettertogether.com/>
* **Username**: tester
* **Password**: Password1

### Open Source Code

The following list outlines the open source code incorporated into the website but not created by the developer.

1. All code contained in the folder ‘forum’. This folder contains the study forum which was incorporated into the website.

**Source:** <http://mylittleforum.net/>

1. All code contained in the folder ‘chat room’ and the CSS file ‘chat room.css’. This folder contains the AJAX chat room which was incorporated into the website.

**Source:** <http://frug.github.io/AJAX-Chat/>

1. The JavaScript ‘ui.dropdownchecklist.js’ and CSS file ‘ui.dropdownchecklist.standalone.css’. These files were used in the ‘upload.php’ page to enable a dropdown check list for the specific-user sharing options.

**Source:** <https://code.google.com/p/dropdown-check-list/>

1. The JavaScript ‘unoSlider.js’ and CSS file ‘unoSlider.css’. These files were used on the ‘index.php’ page to create the image slider.

**Source:** <http://unoslider.decodigothemes.com/>

1. The CSS file ‘reset.css’ which was used to reset all CSS styles across the website.

**Source:** <http://html5doctor.com/html-5-reset-stylesheet/>

### User Login

The system has two public facing webpages – index.php which contains the login form and register.php which contains the user registration form. All other webpages are only accessible by registered users.

Index.php contains an HTML login form which posts the data to the file includes/functionality/login.php. This file runs the loginUser function to validate the inputted username and password and if correct, store the username in a session variable and redirect the user to the home.php page. If the login fails an error message will be displayed to the user underneath the login form on index.php.

**Login User Function – login.php**



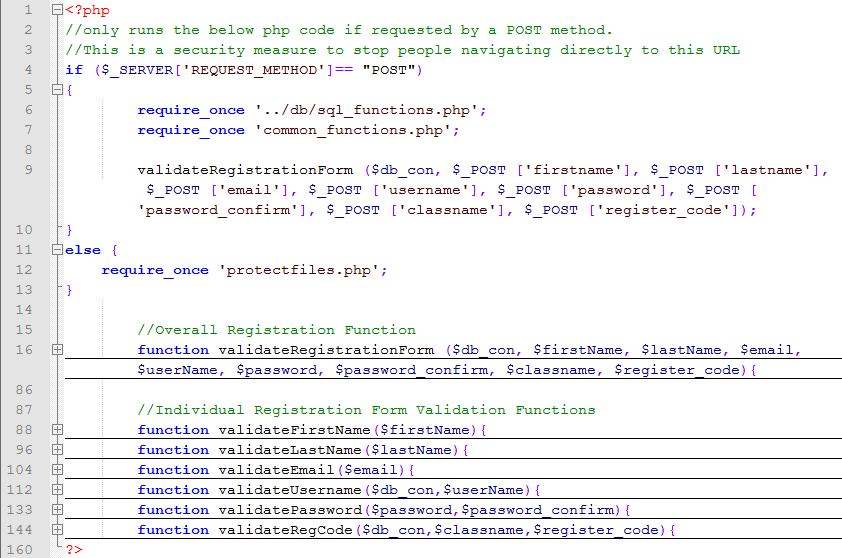
### User Registration

In order for students to register an account on the register.php page they will need to know their class name and the registration code for that class, as provided by their lecturer.

If client-side validation passes, the registration form takes in the user details including their class name and registration code and posts these to the registerNewUser.php page.

This page contains several individual functions to perform server-side validation on the registration form. These individual functions are then utilized in the overall function validateRegistrationForm which performs the following tasks:

* sanitises the input data using the custom sanitiseInput function
* validates the form and ensures the registration code entered was correct
* if form validation passes then a mysqli transaction is used to commit the new user to the users table, retrieve the user ID from this query and then commit this to the forum\_userdata table
* starts the session, stores the username in a session variable and redirects the user to the home.php page
* if form validation fails than the relevant error message is displayed to the user above the registration form

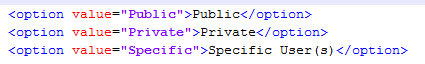
**registerNewUser.php with Functions Collapsed**

### File Sharing

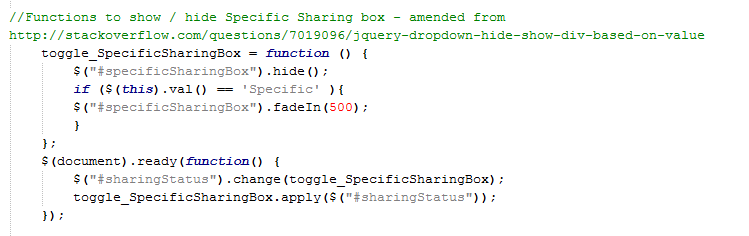
The file sharing feature consists of a front-end upload.php page which contains an HTML upload form and the uploadFiles.php page which takes the information from this upload form, validates it, stores the uploaded file on the server, commits the file information to the database and outputs a success message to the user.

**Front-End File Sharing Options**

One of the fields contained in the upload form is a dropdown box called Sharing Status when the user must choose one of three different sharing options for the uploaded file:

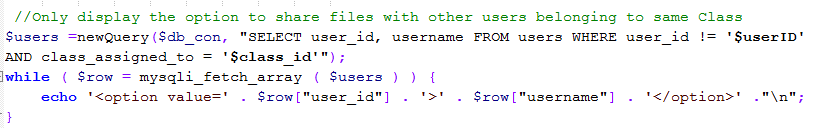


If the user selects the option “Specific User (s)” than a new dropdown box called specificSharing[] will appear containing a list of all users belonging to the same class as the logged in user from which specific people can be chosen to share a file with.

This is performed using jQuery to fade in the new dropdown box (shown below) and a JavaScript plugin to display the users in a drop down check list.

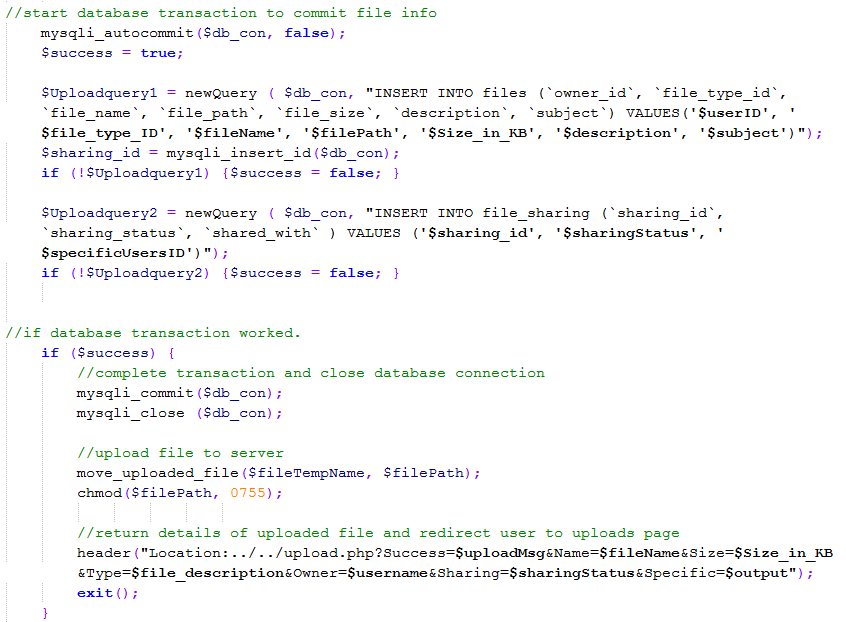
This specificSharing[] dropdown box contains PHP code to enforce the following logic:

* The administrator can specifically share files with anyone
* Other users can only specifically share files with students registered to their same class

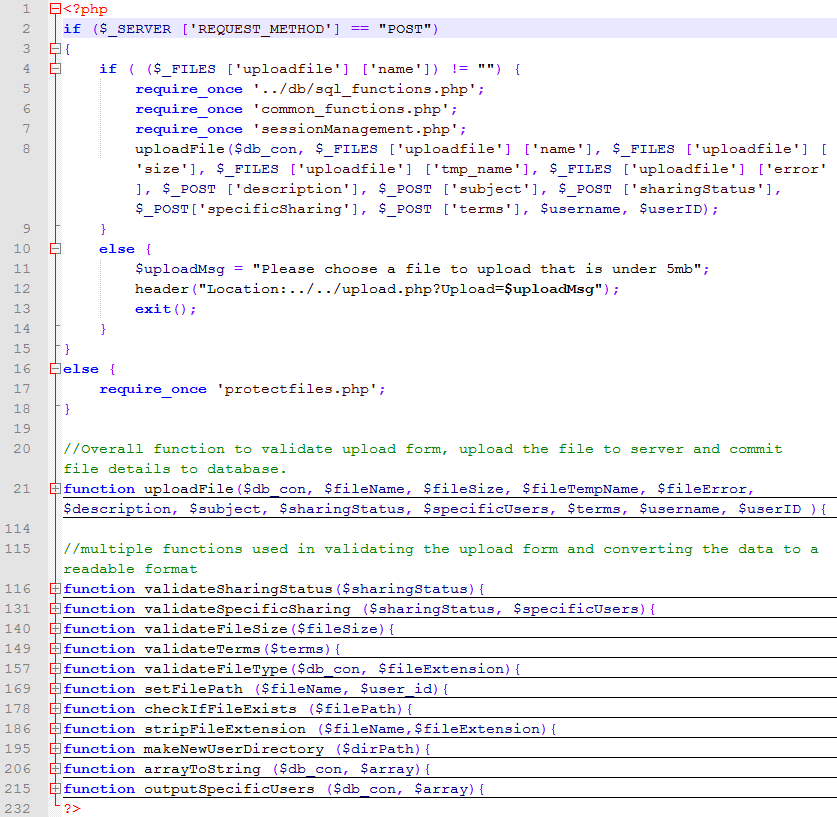
This is achieved by retrieving the logged in user’s class ID from the database and then searching the database to only display users belonging to the same class as shown below:

If client-side validation passes, the upload form posts the data to the uploadFiles.php. This page contains several individual functions to perform server-side validation on the upload form. These individual functions are then utilized in the overall function uploadFile which performs the following tasks:

* sanitises the input data using the custom sanitiseInput function
* validates the form ensuring among other things that the file type is acceptable, the file is under the allowed size and the file does not already exist
* if form validation passes then a mysqli transaction is used to commit the file details into the files table, retrieve the file ID from this query and then commit this file ID along with the sharing status specific-user sharing information to the file\_sharing table
* uploads the file to the server
* returns details of the uploaded file along with a success message back to the user
* if form validation fails than the relevant error message is displayed to the user above the upload form

**Commit File Details to Database and Upload File to Server**

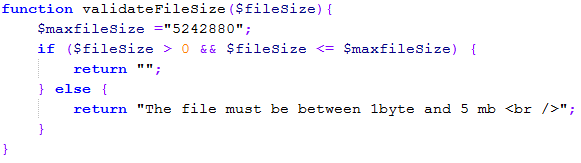
**uploadFiles.php with Functions Collapsed**

To provide an overview of how the uploadFiles.php page is structured, this screenshot displays the page with each of its functions collapsed.

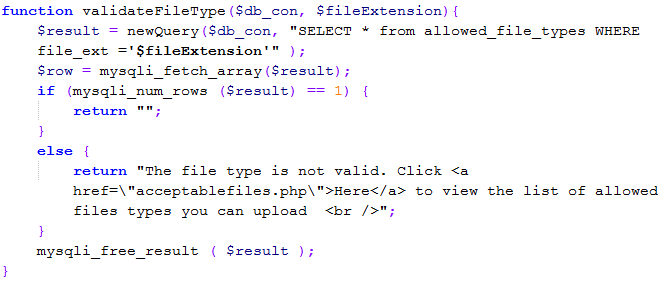
**Individual Upload Functions**

This section highlights some of the individual functions used in the uploadFiles.php page along with comments on what each function does.

This function ensures that the file size is under the set limit - 5mb.



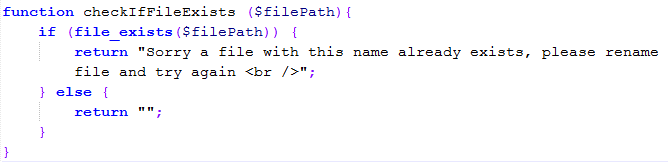
This function ensures that the uploaded file was one of the allowed file types as specified by the allowed\_file\_types table in the database.



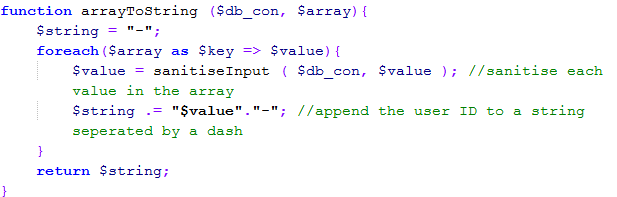
This function create a new directory for the user if not already created and also adds a blank index file to this new directory in order to stop people from viewing directory contents.



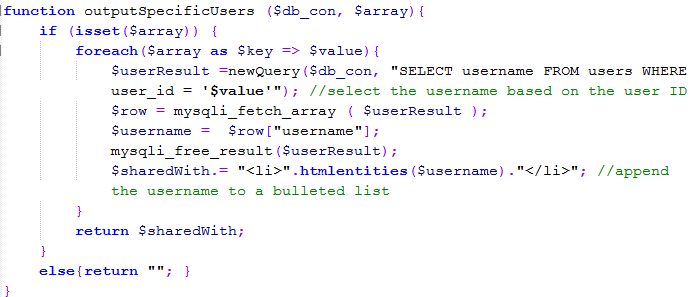
This function ensures that the uploaded file does not overwrite a previously uploaded file of the same name.



This function takes in the array of user ID's from specific sharing and converts these into a string separated by a dash



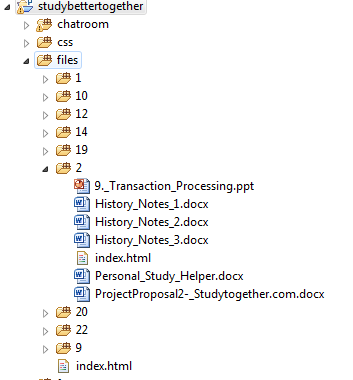
This function takes in the array of user ID's from specific sharing (if set), retrieves the relevant usernames and then converts these into a bulleted list to display to the user



### File Searching and Download

When a user uploads a file, that file is stored in a folder named after the users ID. All of these user folders are stored on the server in the folder called files. As shown in the previous function makeNewUserDirectory, each of the user’s folders also contains a blank index.html page. This is a security measure to stop other users viewing the contents of a folder by navigating to it directly in the address bar.

**Files Folder with Subfolders for Each User Who Has Uploaded Files**



All information on the uploaded files including the file name, size, path, description, date of upload and sharing status details are stored in the two database tables – files and file\_sharing.

The file searching and download feature involves querying these two database tables based on search parameters inputted by the user and then displaying these search results in an HTML table. The following query is used to search for all files that a user owns (i.e. has uploaded).

SELECT file\_id, file\_name, username, sharing\_status,DATE\_FORMAT(DATE(upload\_date),'%D %b %Y') AS new\_date,file\_size, shared\_with, owner\_id, file\_short\_name, description, subject, file\_path FROM files INNER JOIN users ON files.owner\_id = users.user\_id

INNER JOIN file\_sharing ON files.file\_id = file\_sharing.sharing\_id INNER JOIN allowed\_file\_types ON files.file\_type\_id = allowed\_file\_types.file\_type\_id WHERE

owner\_id = '$userID'AND file\_name LIKE '%$fileName%' AND username LIKE '%$fileOwner%'

AND sharing\_status LIKE '%$sharingStatus%' AND file\_short\_name LIKE '%$fileType%' AND description LIKE '%$description%'

AND subject LIKE '%$subject%'ORDER BY `sharing\_status` ASC

The results of this query are displayed in a HTML results table which is stored in the php variable $searchResults. On the front-end php page a ternary operator is used to display the HTML results table if this variable has been set as shown below:



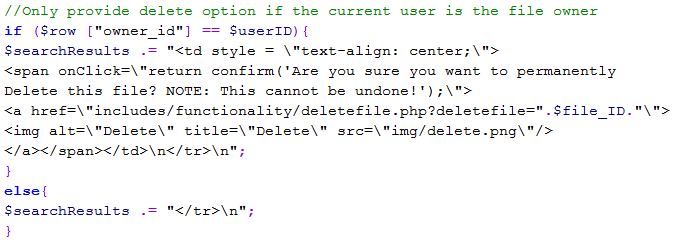
Two image links are displayed beside each file in the HTML results table used to either download or delete the file are implemented as follows:

**Download file**



This line of code uses the protectURL function (shown in section 3.3.10 Common Functions) to display a download button while also hiding the download URL from the users.

**Delete file**



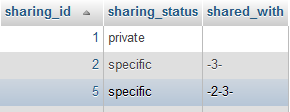
This code checks to see if the displayed file belongs to the logged in user and if so it displays a delete buttons beside the file. The delete button contains a link to the deletefile.php page and along with the file ID.

To prevent XSS attacks, the deletefile.php page contains security checks to see if the file ID supplied through the URL actually belongs to the logged in user and only deletes the file if this is true. The file is deleted from both the server using the PHP unlink() function and from the files table in the database. A success message is then displayed to the user.

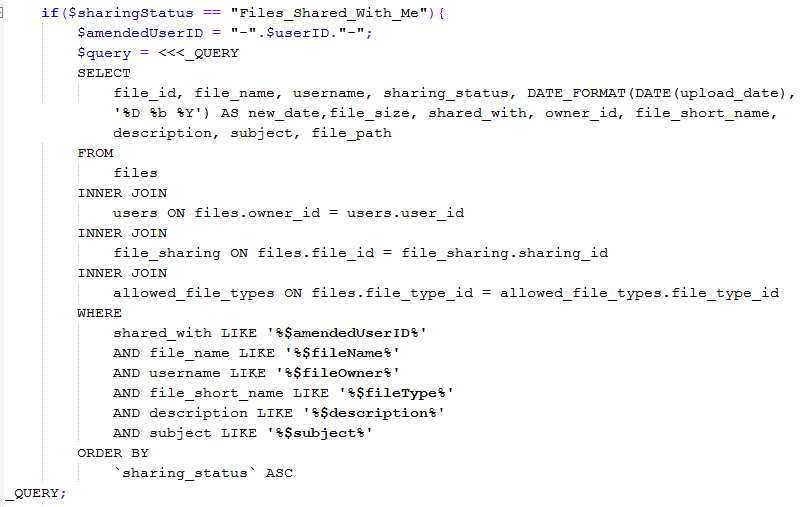
**Searching for Specifically Shared Files**

When a file is shared with specific users, the user ID’s of those users is stored in the shared\_with field of the file\_sharing table as shown below. Each user ID also has a dash appended onto either side of it. Converting this array of user ID's a string and appending the dash to either side is implemented in the arrayToString() function shown earlier.

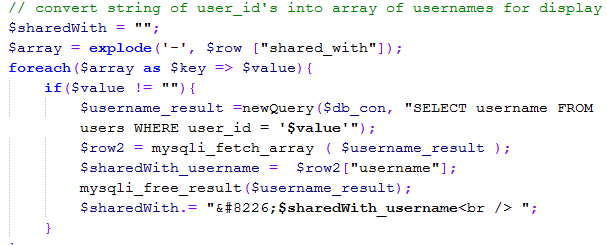
**The file\_sharing table**



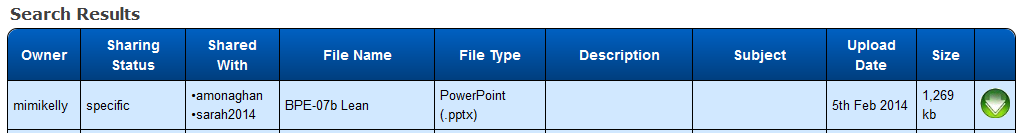
When a user searches for the sharing status “Files shared with me” a new variable is created called $amendedUserID which contains their user ID with a dash appended to either side. For example the user ID ‘2’ would become ‘-2-‘.

As part of the search query for that sharing status, the shared\_with field of the file\_sharing table is searched through to see if it contains the variable $amendedUserID as shown below:

In order to display the username’s of the users that a particular file is shared with, the following code uses the PHP explode() function to break up the shared\_with string into an array of User Id’s. It then loops through this array, searching the users table for the corresponding username and appending this to the $sharedWith variable which is later added to the results HTML table.



Using this method, the string **-2-3-** is converted into the corresponding usernames – **amonaghan** and **sarah2013** and shown in the **Shared With** column of the results HTML table below:



### Integration of Open Source Study Forum

All code contained in the folder ‘forum’. This folder contains the study forum which was incorporated into the website.

**Source:** <http://mylittleforum.net/>

### Integration of Open Source Chat Room

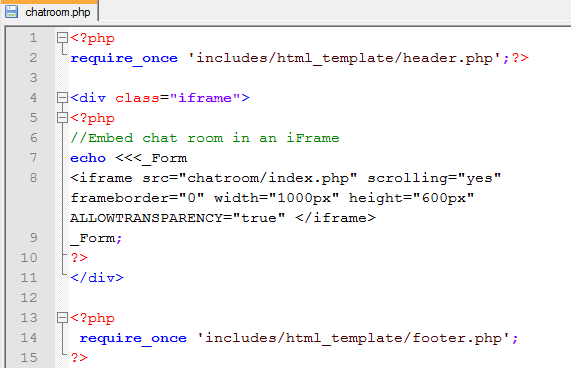
From analyzing the user requirements for an open source chat room the developer needed to choose one that was user-friendly, coded in PHP, utilized a MySql database and also provided detailed installation documentation and support. **(Custom integration howto · Frug/AJAX-Chat Wiki · GitHub. 2014.)**

After researching a number of possible options, the developer chose the chat room “AJAX Chat” as it met each of these requirements as the below tagline states:

*“AJAX Chat is a free and fully customizable open source web chat implemented in JavaScript, PHP and MySQL which integrates nicely with common forum systems like phpBB, MyBB, PunBB, SMF and vBulletin.”* **(Blueimp's AJAX Chat. 2014.)**

**Implementation**

All files pertaining to the chat room are contained in the folder ‘chatroom’. The root page from this folder (chatroom/index.php) is embedded through an iframe into the website’s front-end page chatroom.php as shown below:

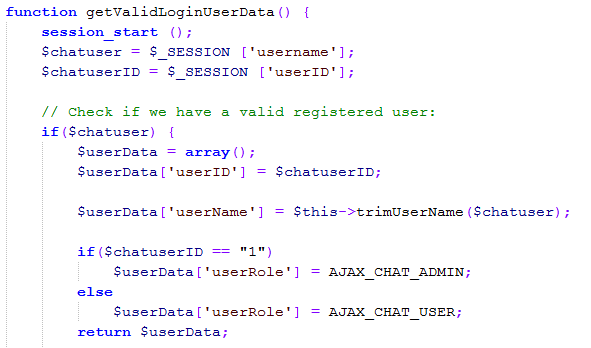


The main difficulties that arose when integrating the chat room into the website were:

* Configuring the chat room login to integrate with the website login.
* Defining the chat room’s user roles so that the administrator (the client) of the website had administrator privilages and the users (the students) had user privilages.
* Preventing direct navigation to the chat room’s root index.php page.

**Integrating the Login and Designating User Roles**

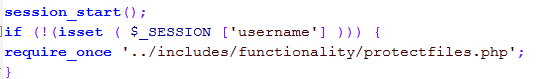
The following code was added to the CustomAJAXChat.php page in order to use the website’s session variables to log users into the chat room. This code takes the session variables ‘username’ and ‘userID’ and stores them in the chat room variables ‘$chatuser’ and ‘$chatuserID’ which are then used to perform the chat room login.



A check is made to see if the ‘$chatuserID’ variables is equal to ‘1’ and if so assigns this user the administration role. As the website administrator (the client) has the userID of 1, this code ensures that only she will have administration access to the chat room.

**Preventing Direct Navigation**

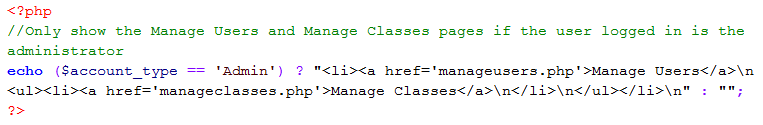
Direct navigation to the chat room’s root page ([studybettertogether.com/chatroom/index.php](file:///C:\Users\Andrew\Dropbox\4th%20Yr%20Project\Tech%20Report\studybettertogether.com\chatroom\index.php)) by any user not already logged into the website was prevented by adding the following code to the top of the chatroom/index.php page.



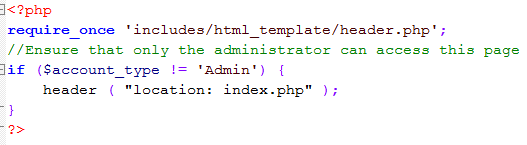
This code prevents direct navigation to the chat room unless the session variable ‘username’ is set (as would be the case if a user has sucessfully logged into the website).

### Administrator Access Only – Manage Classes and Manage Users

There are a number of webpages which are only accessible when the administrator (client) is logged in. These administrator pages are protected in two ways:

1. The menu links are only visible displayed on the menu bar when the administrator is logged in. This is implemented in the header.php template file through a ternary operator that checks the $account\_type session variable to see if the administrator is logged in as shown below:

1. Each of these administrator pages contains the following code segment at the top of the page to ensure that only the administrator can access the page.



**Manage Users - Updating User Details**

The manageusers.php page contains an HTML table which displays the details of all registered users on the website. There is an option beside each user to either update the user details or delete the user and their files.

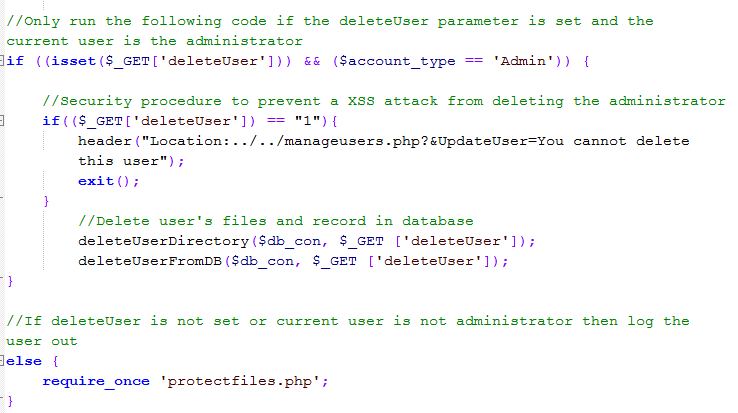
Updating the user details is implemented by sending the userID as a URL parameter from the manageusers.php page to the editusers.php page. This page uses the GET method to retrieve this userID and then searches the database for current information on this user. This information is displayed as the values in the Edit Details HTML form which the administrator can then amend and submit.

Upon passing client-side validation, this form uses the POST method to send these updated user details to the update\_user\_details.php page which performs server-side validation and then commits the changes in user details to the database.

**Manage Users – Deleting a User**

Deleting a user is implemented by sending the userID as a URL parameter from the manageusers.php page to the deleteuser.php page. This page runs two different security checks to ensure a user is not maliciously deleted.

1. The first security check is to ensure that the deleteUser URL parameter is set and that the logged in user is the administrator.
2. The second security check ensures that the administrator cannot be deleted by checking to see if the deleteUser parameter is 1 as 1 is the userID of the administrator.

If both of these security checks pass, that the deleteUserDirectory() and deleteUserFromDB() functions are run as shown in the below code:

**Manage Classes**

The manageclases.php page contains two HTML forms.

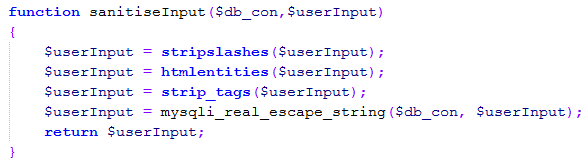
1. The first form, which allows the administrator create a new class, uses the POST method to send this new class information to the addNewClass.php page which then validates the information and then commits it to the database.
2. The second form displays the details of the current classes including how many students are registered for each class. There is an option beside each class to either update the class details or delete the class.

This manage classes functionality is implemented in a similar way to the above manage users functionality.

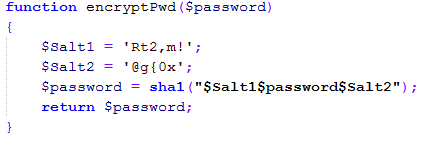
### Common Functions

The following three functions are common to several pages and so are contained in the file common\_functions.php which is included in the required php pages.

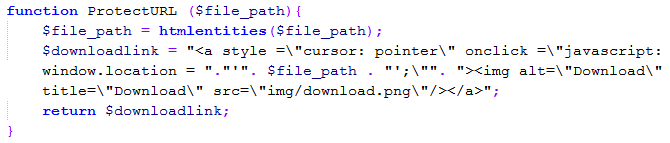
This function is used to sanitise all user input.



This function is used to encrypt and salt all passwords.

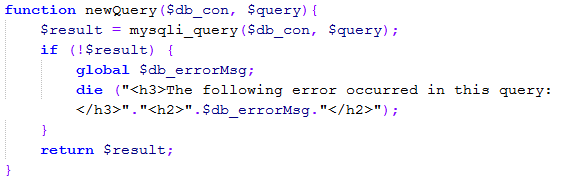


This function displays a download image and also uses JavaScript to hide the file download URL from the user’s view. The purpose for this is to stop users from navigating directly to the files directory.

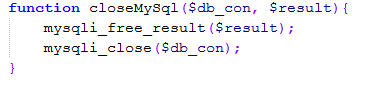


In addition to the above three functions, there are also two common functions used to interact with the MySQL database which are contained in the sql\_functions.php page and explained below.

This function is used as a quicker way to generate a MySQLi query and also display an error if the query was not successful. Due to the large amount of database queries throughout the website, this function has been extremely useful.



This function is used as a quicker way to free a MySQLi query results and also close the database connection.



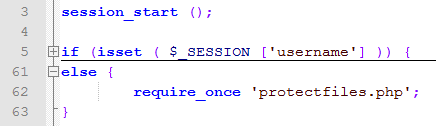
### Security Features

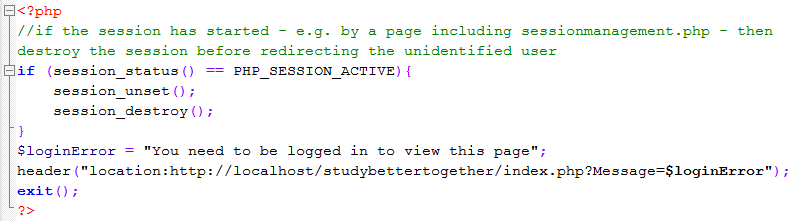
As this website is hosted live on the internet and handles confidential user information, implementing efficient web application security was of paramount importance. This section outlines some examples of how security was implemented in this website.

**Session Management**

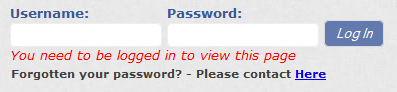
A file called sessionManagment.php is included in each of the protected pages (every page except the index.php and register.php pages) through the use of a PHP require\_once statement.

This page checks to see is the username session variable set and if so retrieves other commonly used information from the database and stores them in session variables. If the username session variable is not set a file called protectfiles.php is included as shown in the below outline:



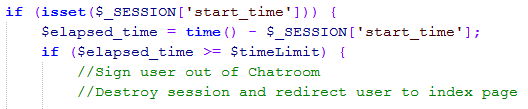
**protectfiles.php**

If someone who was not logged into the website tried to access any of the protected pages they would be redirected to the index.php page with a suitable error message as shown below:

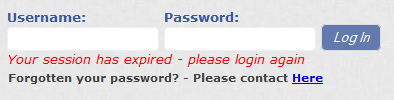


The file sessionManagment.php also contains the code to log a user out of the website after 30 minutes of inactivity. This is achieved by storing the time of every page refresh in a variable called $\_SESSION['start\_time'] and a time limit of 1800 seconds (30 minutes) in a variable called $timelimit.

The following code is then run to see if it has been more than 1800 seconds since a user last used the website:

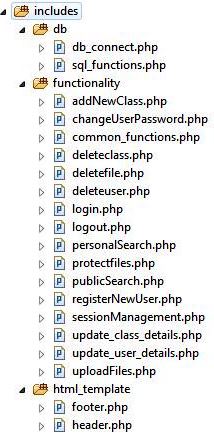


If there has been more than 30 minutes of inactivity the user would be redirected to the index.php page with a suitable error message as shown below:



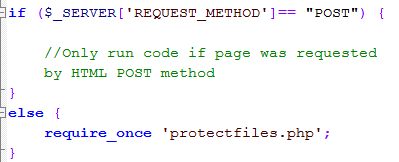
**Protecting Direct Navigation to Functionality Pages**

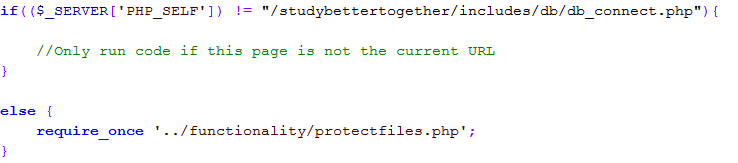
All PHP pages that are not seen or directly interacted with by the user are contained in the ‘includes’ folder as shown in the diagram below.



A security feature is implemented on each of these pages to ensure they cannot be accessed by navigating directly to their URL.

For pages that receive input from the HTML POST, this is achieved through a PHP if function that only runs the code contained in the page if that page was requested by the POST method as shown below:



For pages that are not requested by the POST method, such as the database connection file – db\_connect.php – this security check is implemented by only running the code in the page if that page is not the current URL as shown below:

**Security Functions**

As shown in the previous section (3.3.10 Common Functions) two custom functions have been created to implement the following security on the website:

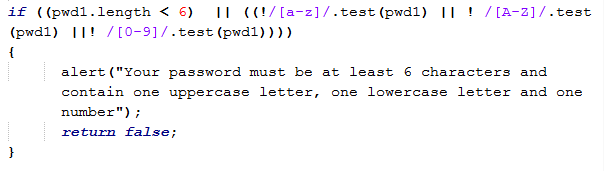
* **sanitiseInput()** – all user input is sanitised by stripping the slashes and tags from it, converting any applicable characters to HTML entities and also escaping any special characters in a string for use in an SQL statement. This is a key function as it protects the website against SQL injection, JavaScript injection and other user input attacks.
* **encryptPwd()** – all passwords and registration codes stored in the database are salted twice and encrypted using the SHA1 hashing algorithm. This returns a 40 digit hexadecimal number which can be safely stored in the database.

**Form Validation**

Validation is implemented on all website forms on both the client-side through the use of JavaScript and HTML5 and on the server-side through the use of PHP.

* Client-side validation is extremely useful as it outputs the error message to the user before submitting the form thus providing a smoother end user experience.
* However client-side validation can easily be manipulated by the end-user so is not secure. To ensure the website is secure, server-side validation must also be in place.

**Example of JavaScript Validation being used on the Registration Form**



## Testing

Testing of this project was carried out at the unit, integration and system levels in the following manner.

### Unit Testing

Each individual webpage was tested by the developer to ensure all components on that webpage were fully functioning as per the functional requirements. This was carried out by opening the webpage in multiple browsers and meticulously testing each element of its functionality. When an error was identified, further investigation was carried out on the client-side by using Google Chrome’s developer tools and on the server-side by using the PHP echo function to highlight a particular variables value when the error was occurring.

**Example of One Problem Identified during Unit Testing**

An example of how one problem was identified and fixed during unit testing using this approach was in the development of the file upload page. During unit testing the developer found that files were not uploading to the user’s directory on the server despite the file details displaying on the website and also successfully committing to the database. This particular problem proved hard to solve as all inputs were correct and the correct data was committing to the database.

However, through using the PHP echo function to show the values of the uploaded files variables, it was discovered that the value stored in the temporary file name variable was incorrect and as such the file was not uploading.

* **Actual Value*:***$\_FILES['uploadfile']['tmp\_name'] = C:xampptmpphpBA4D.tmp
* **Expected Value*:***$\_FILES['uploadfile']['tmp\_name'] = C:\xampp\tmp\phpA078.tmp

Having identified the cause of the problem the developer found the reason for the difference in actual and expected value names was due to input security being applied to the temporary file name variable. It was being put through the same sanitiseInput function as the other input variables as shown below:

* $fileTempName= sanitiseInput($db\_con, $\_FILES['uploadfile'][tmp\_name]);

This is the standard security procedure for all input variables on this website; however in this instance the PHP function stripslashescontained in thesanitiseInputfunctionwas removing the required slashes from the temporary file name thus causing it to be an invalid directory path.

This problem was fixed by simply not sanitising the tmp\_namevariable. This variable is just a placeholder for files on the server and does not provide an input security risk. Using this approach, the developer was able to ensure each component was fully functional before starting to integrate these components into a fully functioning website.

### Integration Testing

Integration testing was conducted throughout this project by the developer and also through continuous client testing. After completing unit testing to ensure each individual component (web page) was fully functioning, it was vital to ensure these individual components worked well together.

**Example of One Problem Identified during Integration Testing**

An example of how one problem was identified and fixed during integration testing can be found in how the login page and session variables were affecting the file searching feature.

A problem was identified during client testing that in some instances a user could not see the delete button in their personal file search results and as such could not delete their files. This was occurring when a user logged in to the website with their username in a different case than the username they had registered. For example if the registered username was “mimikelly” but the user logged in as “MimiKelly”.

The reason for the delete button not showing was that when a user logs into the website, their username is stored in a session variable called $username. When a user searches for their files a check is made to see if that logged in user is the file owner and if so the option to delete their files is displayed. If they are not the file owner, then they cannot delete their files. This check was being made by comparing the username of the file owner to the username of the logged in user as shown below:

//Only provide delete option if the current user is the file owner

**if** ($row ["username"] == $username){

//Code for displaying delete option

}

The problem with this is if the username stored in the database is “mimikelly” but the user logs in as “MimiKelly”. These two stings are not equal to each other and the delete option will not be displayed.

To solve this problem the developer changed the above code to compare the file owner id to the user id of the logged in user as shown below:

//Only provide delete option if the current user is the file owner

**if** ($row ["owner\_id"] == $userID){

//Code for displaying delete option

}

This problem would not have been identified had integration testing not been conducted as both individual components (login and file searching features) were working fine on their own. Using this approach the developer ensured all components integrated successfully with each other.

### System Testing

The fully integrated website was tested using by the developer, the client and also by potential end-users. The following test cases were used to assist in whole system testing.

**Test Case 1 –** Register an account using a valid registration code

**Test Case 2 –** Login to website using valid login credentials

**Test Case 3 –** Upload a file and share it publically with all users

**Test Case 4 –** Upload a file and choose to make it private

**Test Case 5 –** Upload a file and share with specific users

**Test Case 6 –** Search for and download public files

**Test Case 7 –** Search for and download private files

**Test Case 8 –** Search for and download files shared with specific users

**Test Case 9 –** Search for and download files specifically shared with me

**Test Case 10 –** Delete a file previously uploaded

**Test Case 11 –** Upload, Search and Download files of varying file types

**Test Case 12 –** Post a new topic on the study forum

**Test Case 13 –** Reply to a previously created topic on the study forum

**Test Case 14 –** Search through topics on the study forum

**Test Case 15 –** Delete a post on the study forum

**Test Case 16 –** Login to the chat room

**Test Case 17 –** Chat with another online user in the chat room

**Test Case 18 –** Change your password

**Test Case 19 –** Edit student’s details

**Test Case 20 –** Delete a student’s account

**Test Case 21 –** Add a new class

**Test Case 22 –** Test new class details are integrating into study forum

**Test Case 23 –** Edit class details

**Test Case 24 –** Delete a class

**Test Case 25 –** Test all menu links

**Test Case 26 –** Test security of all input forms

**Test Case 27 –** Log out of website

A selection of these test cases is contained in **Appendix 7.1.**

### Client Testing

The client for this project is Dr. Mimi Kelly who delivers both undergraduate and postgraduate modules in History, Sociology and Study Support for a third level College in Dublin.

Mimi was instrumental in the requirements gathering, planning, design and testing phases of this project and though several meetings with the developer has helped tailor the website to be used by non-technical users for the practical purposes of collaborative learning. A brief description of these meetings is outlined below.

|  |  |
| --- | --- |
| **Date of Meeting** | **Brief Review of Meeting** |
| 01/10/2013 | I had an initial meeting with Mimi to discuss the project idea and to ask her to consider being my customer for the project. Mimi liked the project idea and thought it would be extremely useful in helping students study together.  Mimi wanted me to design the website for a non-technical audience, keeping it simple and user-friendly while still providing the required functionality.  We agreed the target audience for this project would be smaller colleges of less than 100 students who don’t have the resources to maintain a LMS such as Moodle but who still want to facilitate online collaborative learning among their students. |
| 19/10/2013 | Mimi and I spent over an hour discussing the requirements of the website in great detail and discussing the results of the student requirement elicitation questionnaire I had conducted earlier that week.  It was decided to change the quiz maker from a required to an optional requirement as it was listed as the least desired feature in the requirement elicitation questionnaire.  In preparation for this meeting, I had created sample use case descriptions and wireframe diagrams. These proved very useful explaining my ideas for the website to Mimi and gathering her exact user requirements.  Mimi liked the proposed look of the website and reiterated her request that the design of the website be minimal and clutter-free so students without any technical experience can use it.  Mimi was impressed with the planning, design and requirement elicitation work completed to date. |
| 28/11/2013 | Mimi and I met up to review the work I had completed so far, namely the front end design, student registration and login, the database structure and a draft version of the file upload feature.  Mimi was impressed with the design of the website and the functionality completed so far. She liked the file upload feature but requested that a confirmation message containing information on the uploaded file be shown to the user when they upload a file. |
| 30/12/2013 | Mimi and I met up to review the website and discuss plans for implementing the remaining requirements. She wanted two updates to be made to the file upload feature - the ability for students to search for files by file type and the ability for students to delete files they have uploaded.  The feedback I had received from my mid-point presentation was not to code my own study forum but to use an open source forum instead. With this feedback in mind I created a short list of several open source forums and showed them to Mimi during our meeting. We reviewed these forums together and decided to go with “My Little Forum” due to its simple and user-friendly design.    Mimi requested that further functionality be added to the website. She wants the students to be able to register an account according to what class they are in. In the administration section of the website, she wants a new page called “manage classes” where she will be able to add new classes and see how many students are registered for each class. |
| 19/02/2014 | Mimi was delighted with the progress made on the website stating that she loved the design and how easy it was to use. She was impressed with the integration the open source forum and also how I had already completed her previous request for a “manage classes” section.  To further enhance the functionality of the website, Mimi asked that I add an instant Chat room feature. She stated this would complement the study forum and provide students with greater options in communicating online. |
| 26/03/2014 | Mimi was very happy with the open source chat room feature I had implemented. She feels the website is almost ready for end-user testing and is planning on getting some of her students to test the website.  We discussed the UAT feedback questionnaire I had created and agreed it would be beneficial to keep the questionnaires anonymous in order to elicit more accurate feedback.  While testing the website in this meeting we identified one error with the searching feature where the user was being given the option to delete their files. Upon further analysis, I discovered this due to a problem in case-sensitivity and implemented the required fix. |
| 06/04/2014 | Mimi conducted a final in-depth review of the website, looking at the functionality and design of every web page and how each web page integrates together to create a smooth end-user experience.  Mimi stated that she is delighted with the final product and that it exceeds meets her initial user requirements. |

### End-User Testing

The final website was tested through extensive end-user testing by both the client and her students along with other students a variety of third level colleges. To facilitate this end-user testing the website was hosted live at the following URL <http://studybettertogether.com/> and a questionnaire was created to capture the results of this user acceptance testing (UAT).

Eighteen completed questionnaires were returned and the results were largely positive. While the full results are contained in **appendix 7.2** some of the key figures are:

* 94% of respondents had a positive overall experience of using the website with 72% rating their overall experience as excellent and 22% rating their experience as above average.
* 78% of respondents thought continuous use of the website would help them in studying collaboratively.
* All respondents were either very satisfied or satisfied with the navigation and content layout.
* The design, graphics and layout and the uploading files proved to be the most popular features of the website.
* 83% of respondents stated they would use the website again.

The last question on the survey provided respondents with a chance to provide any further feedback if they wished. These comments were particularly positive as shown by the below examples:

* *“I found that this website is particularly user-friendly which would be of great use at the Open University due the wide variety of students of varying ages and technical abilities.”*
* *“Nice website, will be handy for a lot of students”*
* *“Well done Andrew. Brilliant concept and very well put forward, easy to use site.”*
* *“Excellent site. Functional and professional in appearance. Well done.”*
* *“This is a great and simple resource that is easy to use and it is very student friendly.”*
* *“An excellent job of work. I wish our extensive, multinational I.T. department could make our systems as easy to use. They don't seem to be able to put themselves in the user’s position and viewpoint.”*

Once respondent had the made the following suggestion regarding the file upload feature: *“When you forgot to click "accept terms of upload" the form resets. It would be nice if it kept the session values in the form so you can just click ’yes’ in t&c”*

**Update to Website based on User Feedback**

Since receiving this feedback, the developer has addressed this issue by incorporated front-end JavaScript validation into the file upload form that prevents the form from submitting unless the T&C are accepted.

## Graphical User Interface (GUI) Layout

**Login Page** - <http://studybettertogether.com/>

This is the first page visitors to the website will see. On this page they have two options, either login to the website or register for an account. This page is made up of static HTML and a JQuery sliding banner.

**Home Page –** <http://studybettertogether.com/home.php>

This is the first page registered users will see when they log in to the website. This page contains graphical links and descriptions for each of the websites main features.

Further screenshots and descriptions for each of the websites main features can be found in the **User Manua**l contained in **appendix 7.6**.

# Conclusions

Completing this project has been both a challenging and rewarding experience. When starting this project I wanted to create something that could have a real life application and on a subject matter which interested me. By creating this collaborative learning website, I believe have achieved both of these objectives and I am very pleased with the final result.

Working on this project has not been without its difficulties, particularly in the early stages of development. When starting this project I had no prior experience is using PHP or designed my own website so it took a lot of research, practice and long hours of coding in order to create the website as it is now. Due to this lack of experience, I encountered numerous problems in the artefact development, integration of open source code and online hosting of the database and website. Overcoming these problems however is all part of the learning experience and as a result my knowledge of web development has grown substantially.

Having a real life client to provide input into the requirement’s, design and testing phases of the project was key to the final product. It has been an excellent experience to take a project from concept through to design, development and finally deployment while also having to regularly check in with a client and amend the functionality based on their feedback.

I believe this project has many advantages and real life applications for students who want to study together. Even larger colleges that already use a LMS such as Moodle could benefit from my project as proven by the requirement elicitation questionnaire which was completed largely by students from such a college. 93% of respondents found sharing study resources with each other helpful although none of these respondents had ever used Moodle to do so.

During end-user testing, one fourth year student who uses Moodle frequently commented that he wished he had access to this website since first year. This type of feedback supports the claim that Moodle and other similar learning management systems, while very useful for teacher to student interaction, are not used for student to student interaction.

The one disadvantage of this project is that the CSS is not responsive and thus has limited usability from smaller screens. However this is due to the fact that I chose not to use a front-end template but to design the website myself. Adapting the website for mobile use is a feature I have planned as part of the future development of this project.

# Further Development

As evidenced by the positive results of both the requirements elicitation questionnaire and the user acceptance testing questionnaire, there is still a strong need for collaborative learning applications that facilitate student to student interaction as opposed to the classical teacher to student interaction offered by most mainstream learning management systems.

Further additions to the application requested by end-users during UAT testing include:

* **Email functionality** - users would be notified by email when files are specifically shared with them or when their forum posts are replied to. This would be implemented using the phpmailer code library.
* **Link to social media** - users would be able to “like” certain files or forum posts and share these on their social media pages.
* **Sharing of video content** – users would be able to upload and share video content with each other.

In addition to implementing the above suggestions, further development of this project would involve developing a mobile application version of the website. This would enable collaborative learning among students from any device or location, putting collaborative learning directly into the hands of the students.

## Proposed Business Model

In completing this project the developer has identified a potential gap in the e-learning market for collaborative learning applications that target small colleges of a non-technical audience. A proposed business model for this project is to sell the web application as packaged software where the college purchasing it has full ownership of the code and uploaded files.

The application would be hosted by the college and become an extension of their own website domain. For example the college [www.dublincollege.com](http://www.dublincollege.com) would purchase this packaged software and add it to their domain name to become [www.dublincollege.com/sbt](http://www.dublincollege.com/sbt) or another extension name of their choosing.

As part of this business model the developer would charge a flat rate fee for the application and database code. For an additional fee, the developer would set up the application and configure the database for the college in addition to providing staff training and three months free technical support. It is believed this approach would best suit smaller colleges and the added benefits of training and technical support would be particularly appealing to colleges of a non-technical background.

# References

* Robin Nixon, 2012. Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites. Second Edition Edition. O'Reilly Media.
* Julie Meloni, 2012. Sams Teach Yourself PHP, MySQL and Apache All in One (5th Edition). 5 Edition. Sams Publishing.
* Julie Meloni, 2011. Sams Teach Yourself HTML, CSS, and JavaScript All in One. 1 Edition. Sams Publishing.
* PHP Video Courses and Tutorials from lynda.com. 2013. PHP Video Courses and Tutorials from lynda.com. [ONLINE] Available at: <http://www.lynda.com/PHP-training-tutorials/282-0.html>
* PHP and MySQL Video Tutorials. 2013. PHP and MySQL Video Tutorials. [ONLINE] Available at: <http://www.developphp.com/list_php_video.php>.
* Dillenbourg P. (1999) What do you mean by collaborative learning?. In P. Dillenbourg (Ed)
* Collaborative-learning: Cognitive and Computational Approaches. (pp.1-19). Oxford: Elsevier [ONLINE] Available at: <http://halshs.archives-ouvertes.fr/docs/00/19/02/40/PDF/Dillenbourg-Pierre-1999.pdf>
* My little forum - a simple PHP/MySQL based threaded internet forum. [ONLINE] Available at: <http://mylittleforum.net/>.
* Blueimp's AJAX Chat. 2014. [ONLINE] Available at: <http://frug.github.io/AJAX-Chat/>.
* Custom integration howto · Frug/AJAX-Chat Wiki · GitHub. 2014. [ONLINE] Available at: <https://github.com/Frug/AJAX-Chat/wiki/Custom-integration-howto#wiki-1-readmetxt>.
* Dropdown-check-list - Transforms a html multiple select into a dropdown checkbox list - Google Project Hosting. 2014. [ONLINE] Available at: <https://code.google.com/p/dropdown-check-list/>.
* unoSlider | Lightweight jQuery Content Slider. 2014. [ONLINE] Available at: <http://unoslider.decodigothemes.com/>.
* HTML5 Reset Stylesheet | HTML5 Doctor. 2013. [ONLINE] Available at: <http://html5doctor.com/html-5-reset-stylesheet/>.
* Create a Responsive Website Using HTML5 and CSS3 - Video Tutorial. 2013. [ONLINE] Available at: <http://www.1stwebdesigner.com/css/create-a-responsive-website-video-tutorial/>.
* How to Create a Pure CSS Dropdown Menu. 2013. [ONLINE] Available at: <http://line25.com/tutorials/how-to-create-a-pure-css-dropdown-menu>.

# Appendix

## Test Cases

These three test cases are a sample of the test cases used during system level testing.

**Test Case 1 – Register an account using a valid registration code**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | 1 | | |
| **Title** | | Register an account using a valid registration code | | |
| **Priority** | | High | | |
| **Pre-conditions** | | User has received a valid registration code from their lecturer | | |
|  | | | | |
| **Step** | **Description** | | **Expected Result** | **Pass/Fail** |
| 1 | In a browser, navigate to <http://studybettertogether.com/> | | The web page <http://studybettertogether.com/> opens | Pass |
| 2 | Click on the register button | | The web page opens -<http://studybettertogether.com/register.php> | Pass |
| 3 | Enter First Name and Last Name | |  | Pass |
|  | Check for error message on leaving any field blank and clicking the register button | | Warning message appears beside the blank field – “*Please fill in this field*”. | Pass |
| 4 | Check for error message on entering an invalid email address. | | Warning message appears beside the email field stating what characters are missing from the invalid email entry | Pass |
| 5 | Enter a valid email address | |  | Pass |
| 6 | Check for error message on entering the usernames – “*te st*”, “*test\_test*” and “*t*”. | | Warning message appears –“*The username must be at least 6 characters and can only contain letters and numbers. Please try again*” | Pass |
| 7 | Check for error message on entering a username that is already taken – “*amonaghan*” | | Warning message appears –“*That username is already taken, please choose another username*.” | Pass |
| 8 | Enter a valid username | |  | Pass |
| 9 | Check for error message on entering the password – “*test*” | | Warning message appears –“*Your password must be at least 6 characters and contain one uppercase letter, one lowercase letter and one number*” | Pass |
| 10 | Enter a valid password | |  | Pass |
| 11 | Check for error message on not entering the same password in the Confirm Password field | | Warning message appears –“*The passwords do not match. Please try again*” | Pass |
| 12 | Enter the same password in the Confirm Password field | |  | Pass |
| 13 | Select your class from the dropdown list. | |  | Pass |
| 14 | Check for error message on entering the wrong registration code for your class | | Warning message appears – “*You have not entered a valid registration code for this class*” | Pass |
| 15 | Enter the correct registration code for your class | |  | Pass |
| 16 | Click the register button | | The system registers your details in the users table in the database | Pass |
|  | | | | |
| **Post-conditions** | | Users details are registered in the database and the user is redirected to <http://studybettertogether.com/home.php> | | |

**Test Case 2 – Login to website using valid login credentials**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | 2 | | |
| **Title** | | Login to website using valid login credentials | | |
| **Priority** | | High | | |
| **Pre-conditions** | | User has successfully registered an account. | | |
|  | | | | |
| **Step** | **Description** | | **Expected Result** | **Pass/Fail** |
| 1 | In a browser, navigate to <http://studybettertogether.com/> | | The web page <http://studybettertogether.com/> opens | Pass |
| 2 | Check for error message on not entering any username or password and clicking the login button. | | Warning message appears beside the blank field – “*Please fill in this field*”. | Pass |
| 3 | Check for error message on entering the wrong username or password | | Warning message appears - “*Your login details are incorrect*” | Pass |
| 4 | Enter the correct username and password. | |  | Pass |
| 5 | Click the login button | | The system logs you in | Pass |
|  | | | | |
| **Post-conditions** | | User is logged into the website and redirected to <http://studybettertogether.com/home.php> | | |

**Test Case 10 – Delete a file previously uploaded**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | 10 | | |
| **Title** | | Delete a file | | |
| **Priority** | | High | | |
| **Pre-conditions** | | User is logged into the website and has successfully uploaded a file | | |
|  | | | | |
| **Step** | **Description** | | **Expected Result** | **Pass/Fail** |
| 1 | Click on the menu option “user’s Files” | | The web page [http://studybettertogether.com/search\_ personalfiles.php](http://studybettertogether.com/search_personalfiles.php) opens | Pass |
| 2 | Search for files using the Sharing Status “*All My Files*” | | A list of all your files appears below the search box with a download and a delete button beside each file | Pass |
| 3 | Click the delete button beside your file | | A warning message appears – “*Are you sure you want to permanently Delete this file? Note: This cannot be undone!*” | Pass |
| 4 | Check the Cancel button is working by clicking “Cancel” | | The warning message disappears and the same output is displayed as in step 2 | Pass |
| 5 | Check for security measures by trying to delete another users file by editing the file ID in the URL parameter (e.g. deletefile=37) | | A warning message appears – “*You cannot delete a file that you do not own!*” | Pass |
| 10 | Redo step 3 and click the “Ok” button | | The file is deleted and a message appears above the search form “*The requested file "file\_name" has successfully been deleted*” | Pass |
|  | | | | |
| **Post-conditions** | | The deleted file is removed from the user’s directory on the server and its corresponding record is removed from the database. | | |

## UAT Questionnaire Results



**1. What college do you currently attend?**

**2. How would you rate your overall experience of using this website?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifHow would you rate your overall experience of using this website?  Excellent  https://www.surveymonkey.com/i/t.gifAbove average  https://www.surveymonkey.com/i/t.gifAverage  https://www.surveymonkey.com/i/t.gifBelow average  https://www.surveymonkey.com/i/t.gifPoor |

**3. Do you think continuous use of this website would help you in studying collaboratively?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifDo you think continuous use of this website would help you in studying collaboratively?  Very helpful  https://www.surveymonkey.com/i/t.gifSomewhat helpful  https://www.surveymonkey.com/i/t.gifNeither  https://www.surveymonkey.com/i/t.gifNot so helpful  https://www.surveymonkey.com/i/t.gifNot at all helpful |

**4. How satisfied were you with the navigation and content layout?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifHow satisfied were you with the navigation and content layout?  Very satisfied  https://www.surveymonkey.com/i/t.gifSatisfied  https://www.surveymonkey.com/i/t.gifNeutral  https://www.surveymonkey.com/i/t.gifDissatisfied  https://www.surveymonkey.com/i/t.gifVery dissatisfied |

**5. How would you rate each of these features?**

|  | Excellent | Above average | Average | Below average | Poor | Feature not used |
| --- | --- | --- | --- | --- | --- | --- |
| Design, graphics and layout | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| Upload personal files | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| Upload public files | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| Upload and share files with specific users | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| Search for public files | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| Search for your own files | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| The study forum | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| The study Chatroom | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |
| The study advice and resources pages | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif | https://www.surveymonkey.com/i/t.gif |

**6. Would you use this website in the future?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifYes  https://www.surveymonkey.com/i/t.gifNo |
| If No (please specify why) |

**7. Would you like to see any new features (functionality) on this website?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifYes  https://www.surveymonkey.com/i/t.gifNo |
| If Yes (please outline below) |

**8. Did you experience any technical difficulties when using this website?**

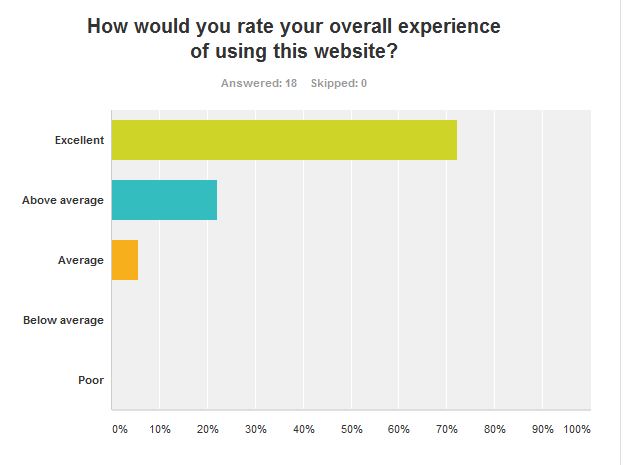
|  |
| --- |
| https://www.surveymonkey.com/i/t.gif Yes  https://www.surveymonkey.com/i/t.gifNo |
| If Yes (please outline below |

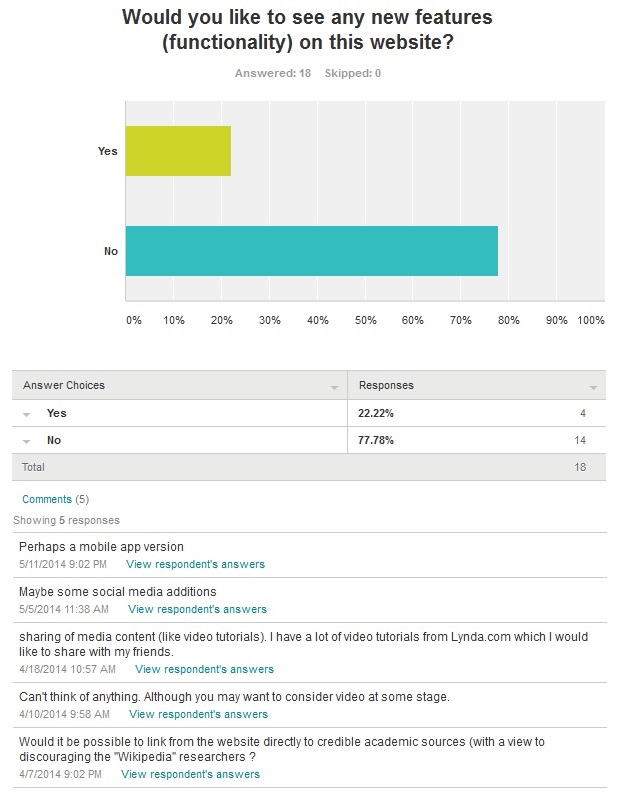
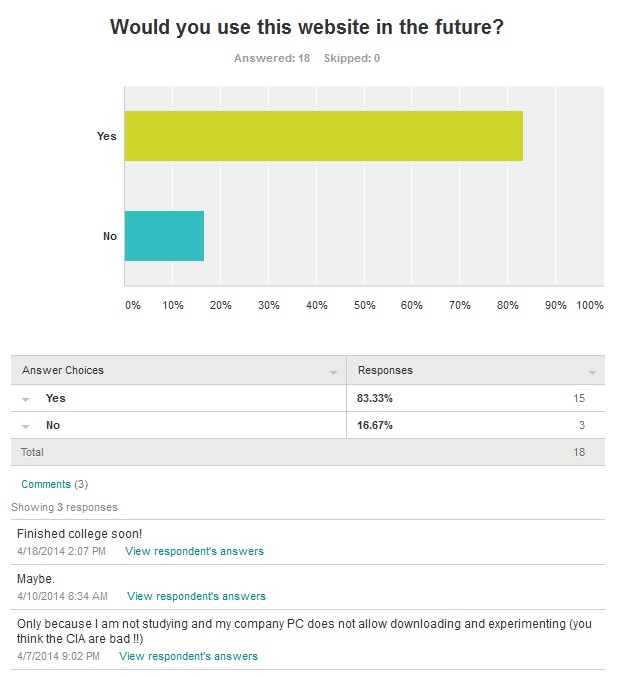
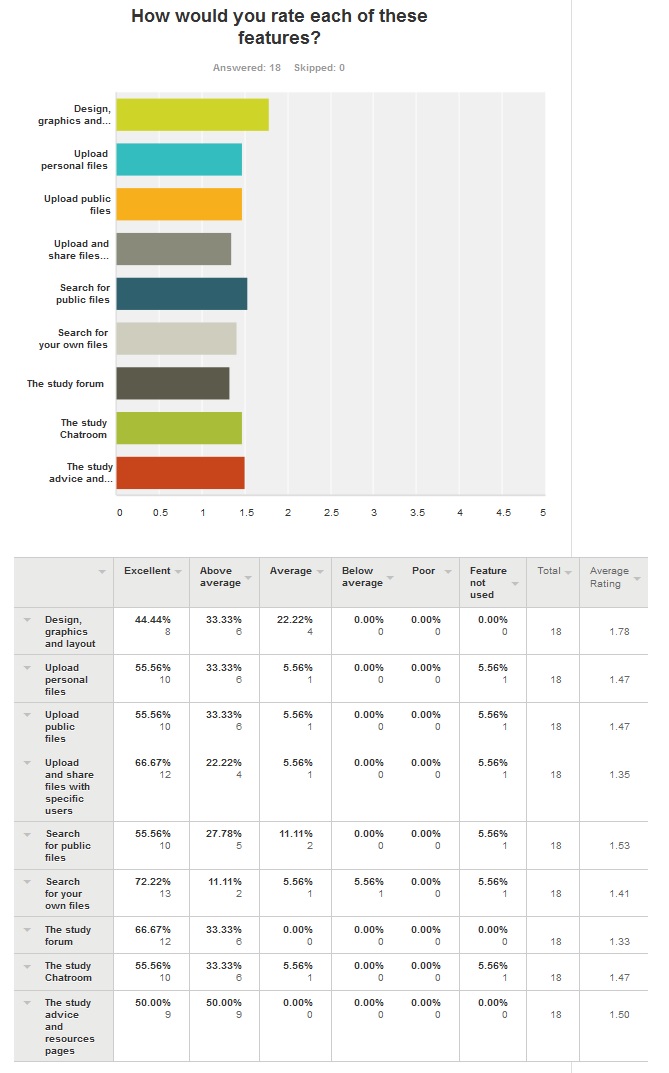
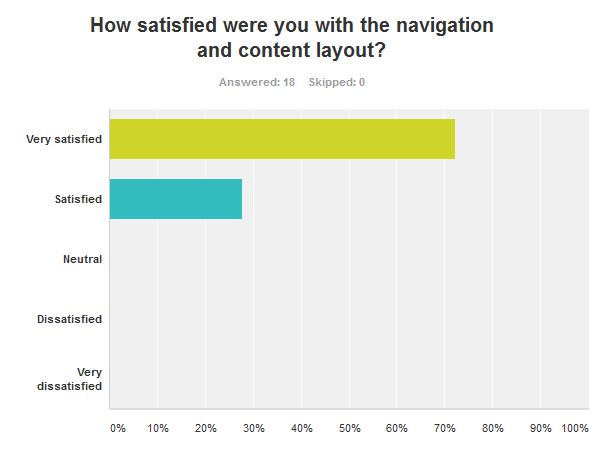
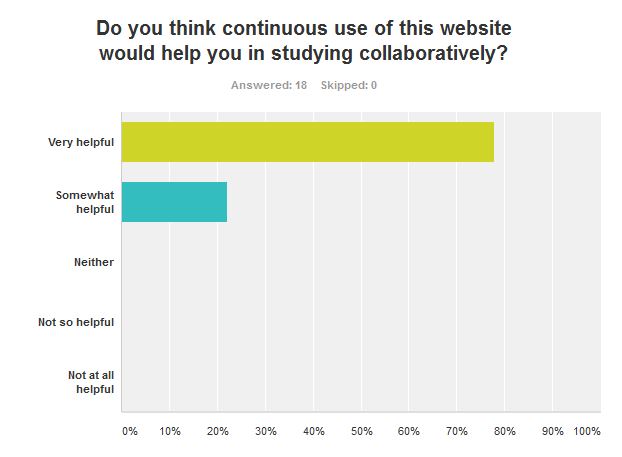
**9. Do you have any other feedback you would like to give?**

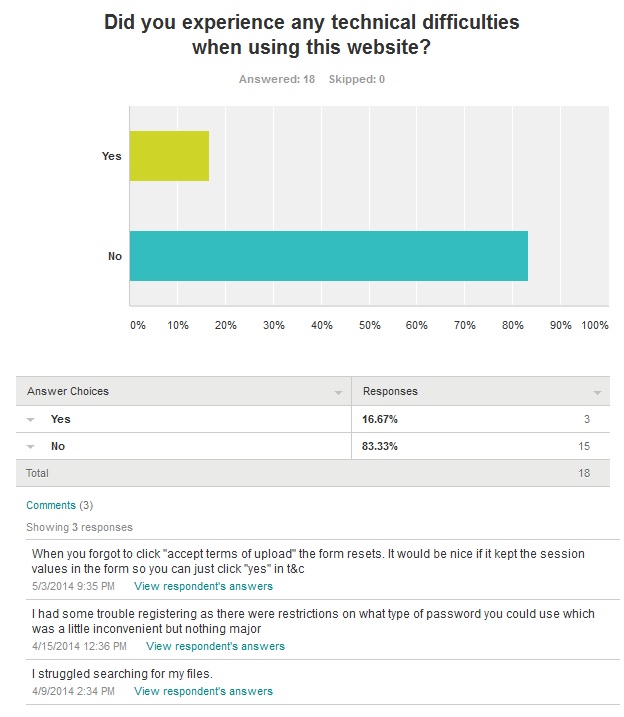
**Study Better Together – UAT Feedback Survey – Results**

<https://www.surveymonkey.com/s/LFMYJWJ>

**Note**: Due to the restrictions placed on exporting results from a free Survey Monkey account, I have had to take screenshots of each result.







**Do you have any other feedback you would like to give?**

* I found that this website is particularly user-friendly which would be of great use at the Open University due the wide variety of students of varying ages and technical abilities.
* Nice website, will be handy for a lot of students
* Well done Andrew. Brilliant concept and very well put forward, easy to use site.
* Excellent site. Functional and professional in appearance. Well done.
* Great work.
* Well done, Andrew.
* It is a really easy and helpful website! Good job :)
* This is a great and simple resource that is easy to use and it is very student friendly.
* An excellent job of work. I wish our extensive, multinational I.T. department could make our systems as easy to use. They don't seem to be able to put themselves in the user’s position and viewpoint.

## Project Proposal

**Project Proposal**

**Study Better Together**

***‘A collaborative file-sharing, quiz-making mobile study platform’***

Andrew Monaghan

09102825

andrew.monaghan@student.ncirl.ie

BSc (Hons) in Business Information Systems

06/10/2013

**Objectives**

The scope of this project is to create a website and mobile web app that will facilitate collaborative learning among students by allowing them share notes, ask questions and create and share MCQ quizzes which can be easily accessed on mobile devices.

My client is Mimi - a lecturer in a third level Theology College in Dublin who delivers both undergraduate and postgraduate modules in History, Sociology and Study Support. She will use this website to communicate with her students through their mobile phones and also facilitate collaborative learning among her students.

One of the key features of this project is that the web application can be both accessible and user friendly to all students regardless of the device they are using. To achieve this, two versions of this web application will be created, a standard website and a mobile web app.

I envision *(see Note below)* this project providing the following functionality:

* The lecturer can post study related content to the site which can be accessed by the students
* The lecturer can create MCQ quizzes for her students and likewise the students can create MCQ quizzes for themselves. These quizzes can be easily accessible on any device.
* Students can post questions on the study forum. Both the lecturer and other students can reply to these questions meaning that when the lecturer is busy a student can still receive replies to their queries.
* Students can upload and share study notes, projects, essays and any other information with each other. These files can be easily accessible on any device.

**Note**: As this project is still in the initial planning phase, client requirements are still being elicited and analysed, thus the project scope and the project functionality is subject to change.

**Background**

The idea for this project came about from my experiences as a part-time student and also my strong interest in e-learning and mobile learning. It is my belief that students study best when working as a team, sharing their notes and studying together. The functionality to enable students create their own quizzes to test their knowledge and then share these quizzes with each other would provide a fun, interactive method of collaborative learning. The ability to access these features easily on mobile devices helps take learning out of the classroom and literally puts it in the hand of the student.

When I had the initial idea for my project, I approached Mimi to see if she was also interested in the idea and if it might be something she might want to use. Mimi thought the idea was very interesting and that a user friendly system that enabled her to communicate with her students through their mobile phones and also facilitated collaborative learning among her students would be a very useful application.

One of the questions I have been asked when discussing my idea with colleagues is what is the difference between my project and other Learning Management Systems such as Moodle? While my project does share some similar features with Moodle, I see the main difference being that Moodle is primarily used for colleges to interact with their students and has limitations in its abilities for students to interact with each other. For example Moodle does not easily allow a student upload files and share them with other students or create a quiz and share this quiz with other students. Another difference between my project and Moodle is accessibility. Moodle is predominantly used on laptops and PC’s. While reaching the mobile environment is something that Moodle are currently looking into, there still remain limitations in terms of usability and functionality when accessing Moodle on mobile devices. This project does not aim to replace Moodle but to be used by a college alongside Moodle to provide a mobile platform for students to study together through sharing notes, asking questions and quiz based revision.

**Technical Approach**

**Implementation**

This project will consist of two main parts, a website which will provide the main elements of functionality and a mobile web app that will contain most, if not all the functionality provided in the main website.

**Software Development Model**

The development approach will largely follow a sequential development model (aka Waterfall Model) which will consist of five main phases - requirement elicitation, system design, development, testing and finally maintenance. Each phase will follow the previous phase in a sequential manner. That said I will also be adapting elements of an iterative and incremental model to allow for changes in client’s requirements throughout the course of the project lifecycle.

I believe a combination of both these models will enable this project to be both structured and well documented while also being client driven and flexible to any changes in the client’s needs.

**Requirements Elicitation**

The primary form of requirement elicitation will be through several meetings with the client. This process has already begun with a first meeting to discuss the proposed features and functionality. The secondary form of requirement elicitation will be through the use of a questionnaire which will be distributed to a random sample of students. All requirements and their technical implementation will be recorded in the requirement specification document. Project functionality will be subject to change in order to adapt to any changes in the clients requirements.

**Special resources required**

To implement this project I will first need to learn a number of different technologies. As I have never used PHP or JQuery before it will be a real challenge to learn these technologies to implement the required functionality within the allotted time. To do this, I will need access to a number of books, video tutorials and technical support platforms.

The following list is a collection of the resources that I am planning to use:

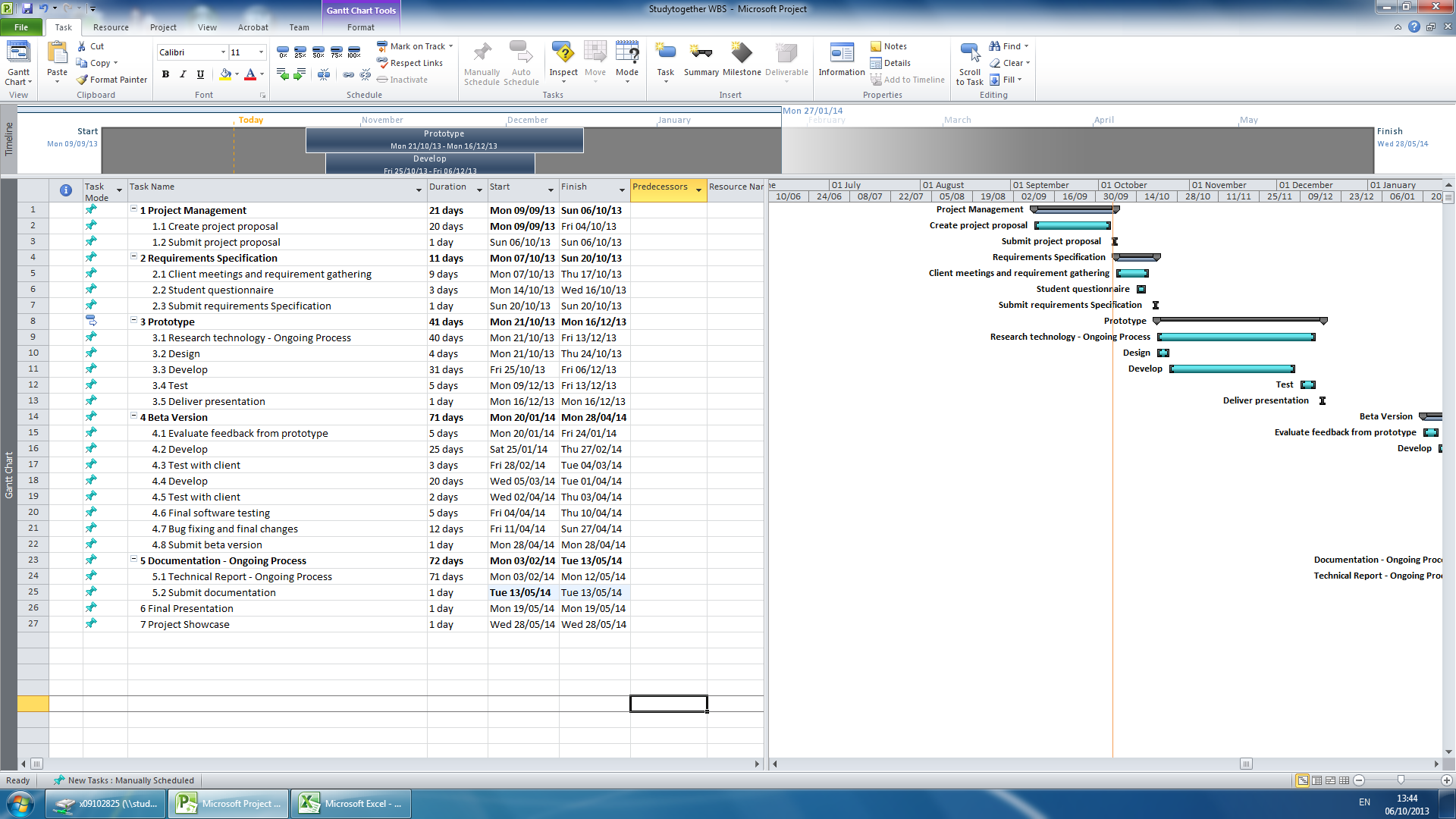
**Books**:

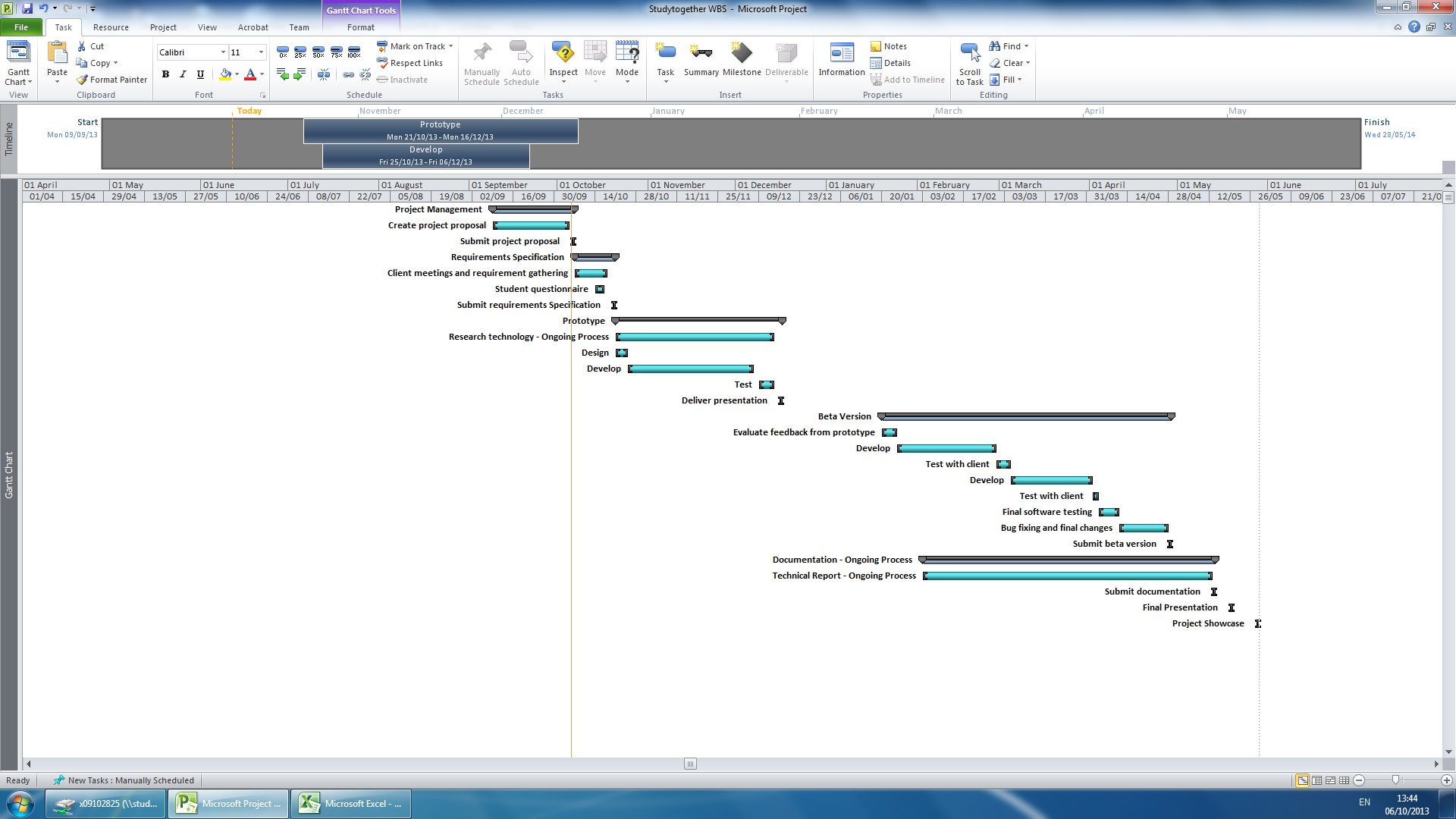
* Learning PHP, MySQL, JavaScript, and CSS, 2nd Edition - Robin Nixon 2012
* PHP Application Development with NetBeans - M A Hossain Tonu - 2012
* Sams Teach Yourself PHP, MySQL and Apache All in One - Julie C. Melon- - 2012
* jQuery Mobile: Up and Running - Maximiliano Firtman – 2012

**Videos – Lynda.com**:

* Installing Apache, MySQL, and PHP with David Gassner, 2013
* PHP with MySQL Essential Training with Kevin Skoglund, 2013
* MVC Frameworks for Building PHP Web Applications with Drew Falkman, 2012
* jQuery Essential Training with Joe Marini, 2009
* jQuery Mobile Web Applications with Ray Villalobos, 2012
* jQuery Mobile Essential Training with Joe Marini, 2011

**Project Plan**





**Technical Details**

**Website Application**

The website frontend will be built using HTML5 and CSS3 to create the site structure and design. JavaScript will be used for client side validation and client side interactive features such as the quiz builder.

The website backend will be built using PHP as the server side language and MYSQL as the relational database. This combination of PHP and MYSQL will be used to create the user account management system, all server side validation, the quiz builder (along with JavaScript), the study forum and the ability for the lecturer to post news items and study tips.

**Mobile Website Application**

The mobile version of the website will be created using a combination of the above technologies alongside the JQuery Mobile Framework.

**Evaluation**

**Software Testing**

This website and web app will be evaluated using the following testing methods:

* **Unit Testing**: This will test the various individual components of the project.
* **Integration Testing**: This will test how the individual components integrate with each other.
* **System Testing**: This will test the completely integrated system and verify that it meets the requirements as defined in the requirement specification document.
* **Compatibility Testing**: This will test that the software is compatible with other application software or if accessed through other software. For example the website will be tested to ensure it is still functional when accessed through different browsers.
* **Performance Testing**: This will test how the software handles under stress. For example will the database still function if accessed my multiple users.
* **Usability Testing**: This will test the user interface of the software and how easy it is to use and understand.

I will also be evaluating this project through continuous end user testing. This will be undertaken through regular communication (both electronic and personal meetings) with the client to ensure they are happy with the progress and that any changes or suggestions they have can be implemented in good time. Upon completion of a first prototype, a test group will be formed from a panel of the clients choosing who can use the prototype in a test environment. This will provide further end user testing to evaluate the project.

**Consultation with Project Specialisation Coordinator**

**Paul Hayes**

Paul liked my idea and thought I should focus more on the quiz making functionality. He stated that it could be very useful for a lecturer to create quizzes that can be easily accessed on a student’s mobile phone.

**Consultation with Academic Staff (1)**

**Jonathan McCarthy**

Jonathan thought the idea of a quiz maker website sounded really interesting and could prove a really useful idea. From my initial conversation with Jonathan, he encouraged me to find a real life client for my project as it was vital to receive real life input in the design stage and also having someone who can evaluate the project would be important. Based on this advice I found a client who was interested in my idea and wanted me to build it for her.

**Consultation with Academic Staff (2)**

**Frances Sheridan**

I talked to Frances about my initial idea for this project which was a mobile Study Smart app containing mainly static content such as study tips and quizzes. Frances said she liked the idea and thought it could be quite useful for students. However she encouraged me to expand on the idea as static content and MCQ in an app is fairly straightforward and would not meet the required functionality for this project. On reflection of this feedback I changed my project slightly to include both a website and mobile website that would offer more interactive functionality such as a web forum, file sharing and quiz builder.

Proposed Supervisor

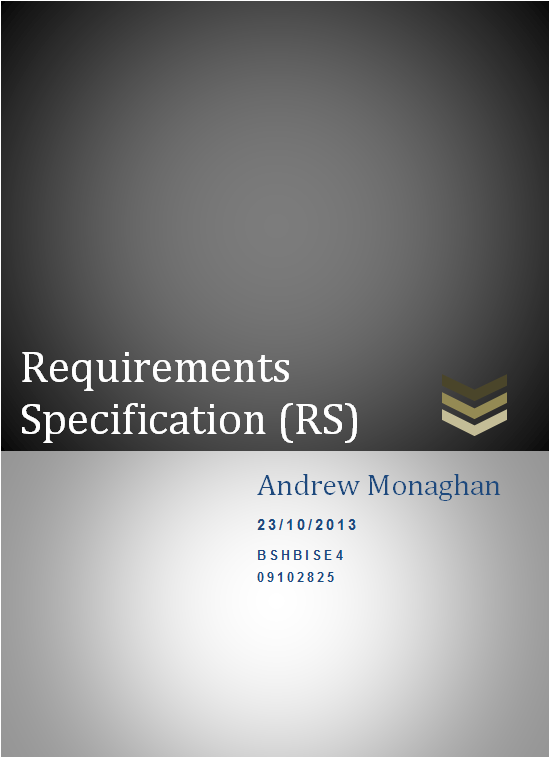
I would like the college to choose a supervisor for me who can work with me and provide technical advice for my project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Andrew Monaghan

6th October 2013

## Requirement Specification



**Requirements Specification (****RS)**

**Document Control**

**Revision History**

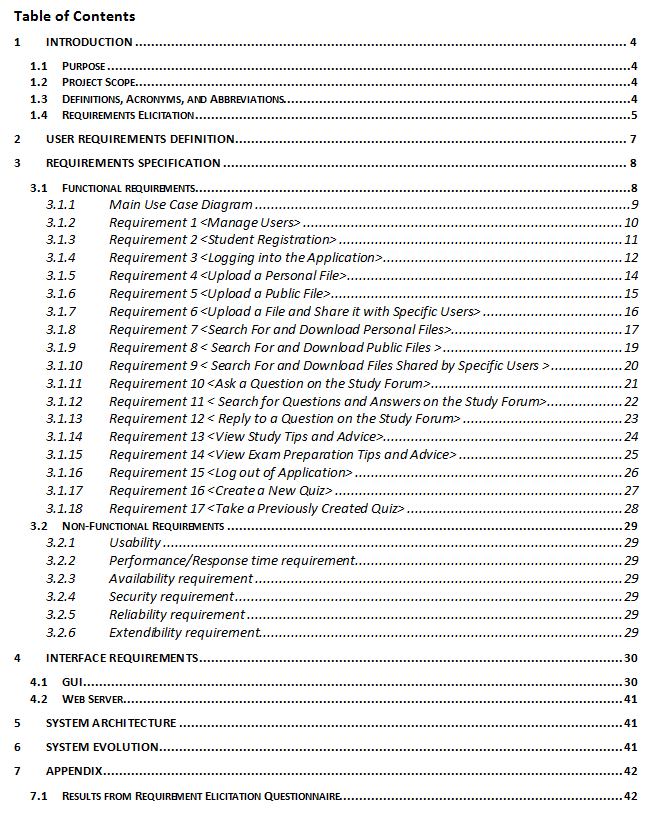
|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared by** |
| 16/10/2013 | 1 | Create | Andrew Monaghan |
| 17/10/2013 | 2 | Update | Andrew Monaghan |
| 23/10/2013 | 3 | Finalise | Andrew Monaghan |

**Distribution List**

|  |  |
| --- | --- |
| **Name** | **Title** |
| Andrew Monaghan | Student / Developer |
| Jonathan McCarthy | Project Coordinator |
| Paul Hayes | Project Supervisor |

**Related Documents**

|  |  |
| --- | --- |
| **Title** | **Comments** |
| ProjectProposal-StudyBetterTogether.docx | Project Proposal Document |



1. **Introduction**
   1. **Purpose**

The purpose of this document is to set out the requirements for the development of a collaborative learning website called studybettertogether.com.

The intended customer of this application is Mimi, a lecturer in a small Theology college in Dublin who among other subjects lectures students in study skills.

* 1. **Project Scope**

The initial scope of this project as outlined in the project proposal has been slightly modified during the requirement elicitation phase. The primary change made was the decision to remove the mobile version of the website from this project. While this was originally identified as a key component, a decision was made not to implement it as to do so would only be duplicating the project functionality albeit in another technology.

It is still the intended long term goal to include a mobile version but for the purpose of this project it will be the desktop web application that will be created. Narrowing the scope of the project has allowed the client add more functionality and will also ensure the developer can implement this new functionality to a high standard.

* 1. **Definitions, Acronyms, and Abbreviations**

**PHP** The server side development language that this project will be using.

**MYSQL** The relational database management system that this project will be using.

**HTML** A mark-up language used to create websites.

**CSS** A styling language used in conjunction with HTML to style websites.

**System**, **Web Application**, **Website**, **Project**

* These four terms all refer to the StudyBetterTogether project and any of these words may be used throughout this document depending on the context.

Throughout this document the terms lecturer and client are used interchangeably. They both refer to the same person – Mimi.

* 1. **Requirements Elicitation**

To ensure a strong foundation for the upcoming development stage of this project I have implemented a thorough requirements elicitation process utilizing several different elicitation techniques. These included client meetings, brainstorming, mind-mapping, front-end prototyping using wireframe diagrams and a requirement elicitation questionnaire.

**Client Meetings**

I have had two meetings with the client in which I gathered user requirements through the use of interviewing, brainstorming, wireframe diagrams, and analyzing the results of the student questionnaire. The wireframe diagrams were particularly beneficial to the client in helping her visualize how the final application might look. These wireframes are displayed in the interface requirements section - 4.1 -GUI.

After our first meeting I documented what I understood to be the client’s user requirements and from these I created initial use cases. In our second meeting we discussed my interpretation of the user requirements and the initial use cases I had created. After making some small changes, the client was happy that I had successfully interpreted and documented her requirements for this project. These client requirements are specified in section 2 - User Requirements Definition.

**Student Questionnaire**

A requirement elicitation questionnaire was distributed to a sample of third level students. Fifteen completed questionnaires were returned. Below is an overview of the results, the full questionnaire results including graphs can be found in appendix 7.1.

**Collaborative Learning**

* 87% of respondents thought that the proposed StudyBetterTogether website would be beneficial in helping them study better.
* Previous experiences of collaborative learning were mixed.
  + 20% - low benefit
  + 33% - medium benefit
  + 27% - high benefit
  + 20% - very high benefit

**Sharing Files**

* 93% of respondents had previously shared learning resources (study notes, essays, projects) with other students and had found it helpful to some degree.
  + 33.3% - somewhat helpful
  + 60% - very helpful
* The primary means of file transfer used to share these learning resources was:
  + Email – 50% | Dropbox – 28.5% | Google Drive – 21.5% | Moodle – 0%

**Study Forum**

* 87% of respondents had previous difficulties in not receiving answers to their college related questions from either lecturers or other students.
* 100% of respondents stated that a study forum to allow students discuss college related topics would be beneficial to some degree.
  + 33.3% - medium benefit
  + 13.3% - high benefit
  + 53.3% - very high benefit

**Quiz Based Learning**

* 100% of respondents stated that the ability to create their own quizzes for revision purposes would be beneficial to some degree.
  + 53.3% - medium benefit
  + 26.7% - high benefit
  + 20% - very high benefit
* 60% of respondents had previously used quizzes to help them learn and had found them helpful.

**Priority of Functionality**

The students ranked the proposed functionality based on what they would find most useful. The results of this, from most to least useful were:

1. Upload and share files with only specific users that you can choose
2. Upload and share files with other users
3. Upload files to a personal folder so that only you have access
4. Access to detailed study tips, hints and advice
5. Access to a study forum where you could post questions, search questions and reply to questions
6. Access to advice for handling exam pressure
7. Create a quiz that requires typing in the answer. (An option would be available to provide a hint if you are stuck.)
8. Create a MCQ quiz and share this with other users

This questionnaire proved extremely useful in the requirement elicitation phase by first demonstrating the usefulness of this project idea (87% of respondents stating that it would be beneficial) and also providing some very interesting insights which were used to help form the functional requirements. Having each respondent rank the proposed functionality according to its usefulness was very helpful in prioritizing the functional requirements.

It is also interesting to note that while 93% of respondents found sharing files for learning purposes helpful, none of these respondents had used Moodle to do so, despite using Moodle as their learning management system on a daily basis. This supports the claim that Moodle and other similar learning management systems, while very useful for teacher to student interaction, are not used for student to student interaction.

1. **User Requirements Definition**

As a result of the requirement elicitation phase, the following user requirements for this application have been identified.

* The overall aim of this project should be to provide a user friendly and accessible platform that will assist students in studying collaboratively. The website should be built in such a way so that it can be used by non-technical users. The style and layout of the website should be clean, minimal and clutter free.
* The website should allow the lecturer to manage the users of the website but not have to create new accounts for each user. This means the lecturer wants to be able to see who has created an account on the system and be able to edit and delete their accounts if needed, but the lecturer does not want to be responsible for creating a new account for each student. It should be the student’s responsibility to register and create their own accounts.
* The website should contain detailed study hints, tips and advice along with similar advice for helping students cope with exam pressure. This should be displayed in a manner that is keeping with the websites clean and minimal style.
* The website should allow students upload and share study resources with each other. These resources could be documents, essays, projects, revision notes or summarized chapters of books that the students have written. The students should be able to choose if they want to share files with everyone or with only specific people. Also there should be the functionality to allow a student upload a file but choose not to share it with anyone (ie keep it private).
* The website should facilitate conversation on any college / study related topics by providing an online study forum for the users to ask questions. The forum should be built in a simple and basic fashion in keeping with the websites clean and minimal style.
* The lecturer should have the same access to the file sharing, study forum and quiz functionality as the students. This will allow for times when the lecturer wants to share files with her students or else she wants to either start a conversation or answer a student’s question on the study forum.
* An optional feature would be to include the functionality for students to create their own revision quizzes to test themselves. This would enable the use of quiz-based learning to help students learn and revise for exams.

1. **Requirements Specification**
   1. **Functional requirements**

From analyzing the above user requirements, 17 functional requirements were identified for this project. Two of these functional requirements relate to the quiz maker functionality will be treated as low priority, optional requirements only to be attempted if all other functionality has been successfully implemented.

These requirements were prioritized based on discussions with the client and from the results of the requirement elicitation questionnaire.

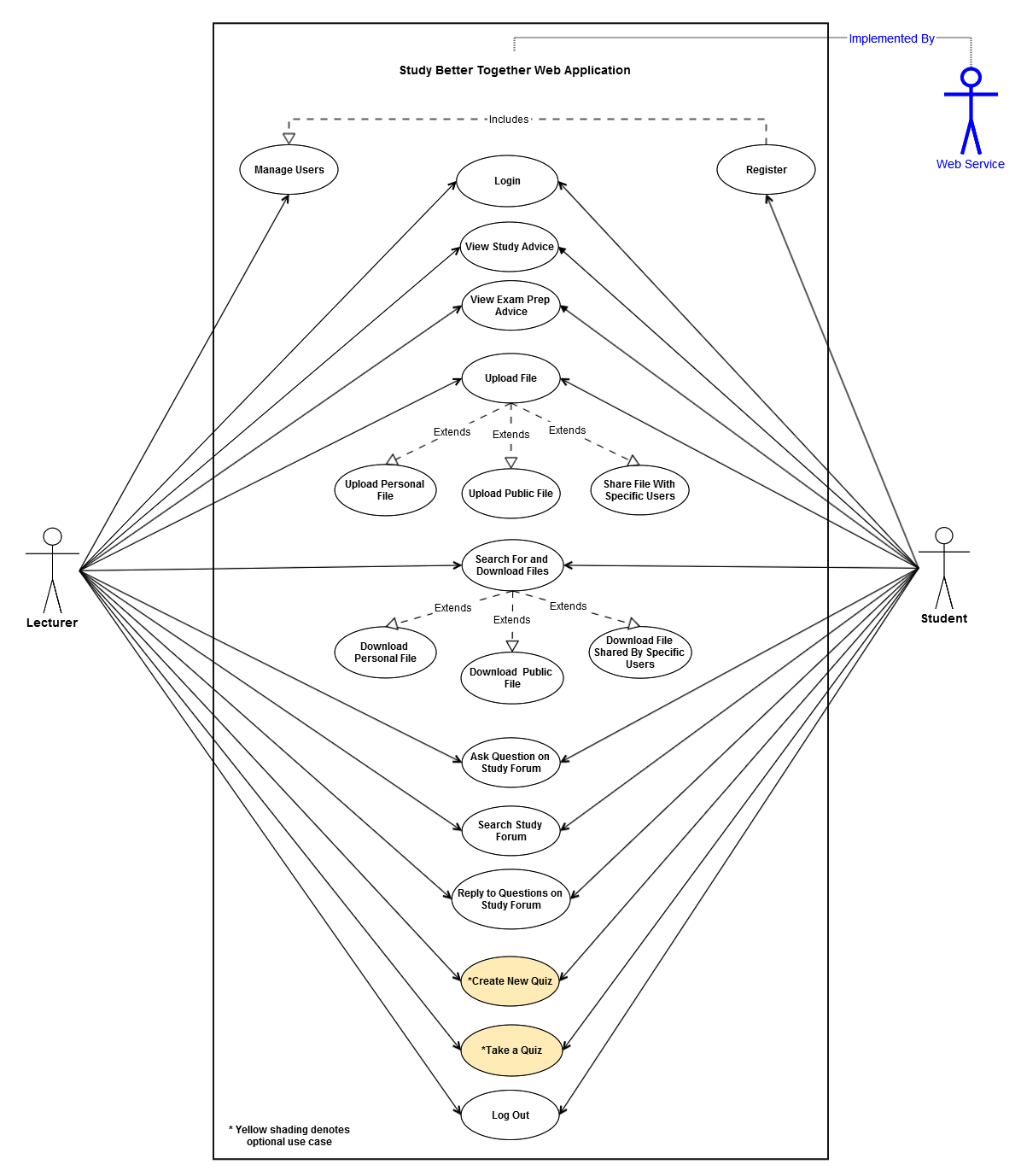
**Full List of Functional Requirements**

|  |  |
| --- | --- |
| 3.1.2 | Requirement 1 <Manage Users> |
| 3.1.3 | Requirement 2 <Student Registration> |
| 3.1.4 | Requirement 3 <Logging into the Application> |
| 3.1.5 | Requirement 4 <Upload a Personal File> |
| 3.1.6 | Requirement 5 <Upload a Public File> |
| 3.1.7 | Requirement 6 <Upload a File and Share it with Specific Users> |
| 3.1.8 | Requirement 7 <Search For and Download Personal Files> |
| 3.1.9 | Requirement 8 < Search For and Download Public Files > |
| 3.1.10 | Requirement 9 < Search For and Download Files Shared by Specific Users > |
| 3.1.11 | Requirement 10 <Ask a Question on the Study Forum> |
| 3.1.12 | Requirement 11 < Search for Questions and Answers on the Study Forum> |
| 3.1.13 | Requirement 12 < Reply to a Question on the Study Forum> |
| 3.1.14 | Requirement 13 <View Study Tips and Advice> |
| 3.1.15 | Requirement 14 <View Exam Preparation Tips and Advice> |
| 3.1.16 | Requirement 15 <Log out of Application> |
| 3.1.17 | Requirement 16 <Create a New Quiz> *- Optional Requirement* |
| 3.1.18 | Requirement 17 <Take a Previously Created Quiz> *- Optional Requirement* |

**Note:**

In the following use case descriptions, when a use case contains the same process flow steps for both the lecturer and the students, both actors will simply be referred to as ‘the user’. This is to improve the readability of the use case descriptions.

* + 1. **Main Use Case Diagram**

****

* + 1. **Requirement 1 <Manage Users>**
       1. **Use Case Scope & Priority**

This use case concerns the lecturer managing the users of this application. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of the lecturer creating a registration code for the users, changing user passwords deleting users from the system. This fulfils the user requirement for the lecturer to control user accounts but not to have to register the users herself. |
| Actors | Lecturer |
| Precondition | The lecturer has successfully logged onto the application. |
| Activation | This use case starts when the lecturer clicks the ‘admin.php’ page. |
| Main flow | 1. The system provides a text box for the lecturer to update the registration code. 2. The lecturer types in a new registration code and clicks the ‘update’ button. |
| Alternate flow | A1 : <Change User Passwords>   1. The system displays a list of all registered users. 2. The lecturer chooses a user and clicks the ‘edit user’ button. 3. The lecturer changes the user’s password and clicks an ‘update’ button. 4. The user’s password has now been updated.   A2 : <Delete User from System>   1. The system displays a list of all registered users. 2. The lecturer chooses a user and clicks the ‘delete user’ button. 3. The user has now been deleted from the system. |
| Exceptional flow | E1 : <No Users are Registered on the System>   1. The system will not display any registered users 2. Alternate flows A1 and A2 will not be available 3. The use case continues at position 1 of the main flow |
| Termination | The lecturer presses the update button. |
| Post condition | The system goes into a wait state. |

* + 1. **Requirement 2 <Student Registration>**
       1. **Use Case Scope & Priority**

This use case concerns the student registration process. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the registration process that a student of the lecturer would follow to register an account in order to use this application. Note that the lecturer does not need to register as she already has an admin account. |
| Actors | Student |
| Precondition | The lecturer has created a registration code and shared this with the student |
| Activation | This use case starts when the student opens the web application and clicks the ‘register’ button. |
| Main flow | 1. The system provides a registration form for the student to input their details and create a password. 2. The student types in their first name, last name and email address. 3. The student creates a new username. 4. The student creates a password 5. The student enters in the registration code as provided by the lecturer. 6. The student presses the ‘register’ button. |
| Alternate flow | There is no alternative to the main flow in registering a user |
| Exceptional flow | E1 : <The chosen username is already taken>   1. The system will display an error message 2. The student will choose another username 3. The use case continues at position 4 of the main flow   E2 : <The student does not fill in all required fields in the form>   1. The system will display an error message 2. The student will need to complete all required fields 3. The use case continues at position 6 of the main flow |
| Termination | The student presses the ‘register button’ after completing the registration form. |
| Post condition | The student is successfully registered and is given an option to log into the application. |

* + 1. **Requirement 3 <Logging into the Application>**
       1. **Use Case Scope & Priority**

This use case concerns the login process to the application. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process involved in a user logging into the application. |
| Actors | Lecturer, Student |
| Precondition | The student has successfully registered an account. |
| Activation | This use case starts when the user opens up the ‘index.php’ page of the application in their browser. |
| Main flow | 1. The system provides a simple login form. 2. The user types in their username and password. 3. The user clicks the ‘login’ button. |
| Alternate flow | There is no alternative to the main flow in logging into the application. |
| Exceptional flow | E1 : < The users username or password is not recognised by the system >   1. The system displays an error message. 2. The user types in the correct username and password 3. This use case continues at position 3 of the main flow   E2 : < The Student has forgotten their password >   1. The student contacts the lecturer. 2. The lecturer logs into the application and changes the student’s password – see the ‘Manage Users’ use case. 3. This use case continues at position 1 of the main flow |
| Termination | The user clicks the login button. |
| Post condition | The user is logged into the web application |

* + 1. **Requirement 4 <Upload a Personal File>**
       1. **Use Case Scope & Priority**

This use case concerns the user uploading a personal file to the web application. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process involved in a user uploading a file to the web application that only he/she can access. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application.  The user has a file on their computer ready to upload to the application |
| Activation | This use case starts when the user clicks the ‘uploadpersonalfile.php’ page. |
| Main flow | 1. The system displays a form for the user to upload a file. 2. The user enters in the file name, description and type 3. The user clicks a ‘browse’ button which allows the user to select a file from their computer. 4. The user selects the radio button to signify that the file is a personal file and should not be shared with other users 5. The user clicks the ‘upload file’ button |
| Alternate flow | A1 : <Upload Files by Dragging and Dropping>   1. The above main flow continues until position 2 2. The user selects the file to upload by dragging the file from his computer into a specified part of the upload form. 3. The use case continues at position 4 of the main flow |
| Exceptional flow | E1 : <The Required Fields in the Upload Form are not Filled in>   1. The system will display an error message to the user 2. The user will need to fill in all required fields 3. The use case continues at position 5 of the main flow   E2 : <The User uploads a file that is above the maximum allowed file size>   1. The system will display an error message to the user 2. The user will need to upload a smaller file 3. The use case continues at position 5 of the main flow |
| Termination | The user clicks the ‘upload file’ button and the file is successfully uploaded. |
| Post condition | The file is uploaded to the system so only the user can access it. |

* + 1. **Requirement 5 <Upload a Public File>**
       1. **Use Case Scope & Priority**

This use case concerns the user uploading a public file to the web application. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process involved in a user uploading a file to the web application that can be accessed by all users. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application.  The user has a file on their computer ready to upload to the application |
| Activation | This use case starts when the user clicks the ‘uploadpublicfile.php’ page. |
| Main flow | 1. The system displays a form for the user to upload a file. 2. The user enters in the file name, description and type 3. The user clicks a ‘browse’ button which allows the user to select a file from their computer. 4. The user selects the radio button to signify that the file is a public file and should be shared with all users 5. The user clicks the ‘upload file’ button |
| Alternate flow | A1 : <Upload Files by Dragging and Dropping>   1. The above main flow continues until position 2 2. The user selects the file to upload by dragging the file from his computer into a specified part of the upload form. 3. The use case continues at position 4 of the main flow |
| Exceptional flow | E1 : <The Required Fields in the Upload Form are not Filled in>   1. The system will display an error message to the user 2. The user will need to fill in all required fields 3. The use case continues at position 5 of the main flow   E2 : <The User uploads a file that is above the maximum allowed file size>   1. The system will display an error message to the user 2. The user will need to upload a smaller file 3. The use case continues at position 5 of the main flow |
| Termination | The user clicks the ‘upload file’ button and the file is successfully uploaded. |
| Post condition | The file is uploaded to the system so all users can access it. |

* + 1. **Requirement 6 <Upload a File and Share it with Specific Users>**
       1. **Use Case Scope & Priority**

This use case concerns the user uploading a file to the web application that will be shared with specific users. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process involved in a user uploading a file to the web application and then specifying the users they want to share the file with. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application.  The user has a file on their computer ready to upload to the application |
| Activation | This use case starts when the user clicks the ‘uploadfile.php’ page. |
| Main flow | 1. The system displays a form for the user to upload a file. 2. The user enters in the file name, description and type 3. The user clicks a ‘browse’ button which allows the user to select a file from their computer. 4. The user selects the radio button to signify that the file should be shared with specific people. 5. The system displays a list of all registered members. 6. The user chooses the specific person he/she wants to share the file with 7. The user clicks the ‘upload file’ button |
| Alternate flow | A1 : <Upload Files by Dragging and Dropping>   1. The above main flow continues until position 2 2. The user selects the file to upload by dragging the file from his computer into a specified part of the upload form. 3. The use case continues at position 4 of the main flow |
| Exceptional flow | E1 : <The Required Fields in the Upload Form are not Filled in>   1. The system will display an error message to the user 2. The user will need to fill in all required fields 3. The use case continues at position 5 of the main flow   E2 : <The User uploads a file that is above the maximum allowed file size>   1. The system will display an error message to the user 2. The user will need to upload a smaller file 3. The use case continues at position 5 of the main flow |
| Termination | The user clicks the ‘upload file’ button and the file is successfully uploaded. |
| Post condition | The file is uploaded to the system so only the chosen members can access it. |

* + 1. **Requirement 7 <Search For and Download Personal Files>**
       1. **Use Case Scope & Priority**

This use case concerns the user searching for and downloading their personal files. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user searching for and then downloading the files that they uploaded as ‘personal files’ |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application.  The user has successfully uploaded a personal file |
| Activation | This use case starts when the user clicks the ‘downloadpersonalfile’ page. |
| Main flow | 1. The system displays a form for the user to search all their files 2. The user enters in a file name, type or description and clicks the ‘search’ button. 3. The system displays a list of files that meet the search criteria in a results table below the search form. 4. The user downloads the file they want by clicking on the file name. |
| Alternate flow | There is no alternative to the main flow in searching for and downloading a personal file. |
| Exceptional flow | E1 : <The system cannot find the required files>   1. No files will be displayed in the results table. 2. The user will need to redefine their search criteria to include a file that has been previously uploaded to the system. 3. The use case continues at position 3 of the main flow |
| Termination | The user downloads the file. |
| Post condition | The system goes into a wait state. |

* + 1. **Requirement 8 < Search For and Download Public Files >**
       1. **Use Case Scope & Priority**

This use case concerns the user searching for and downloading publically shared files. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user searching for and then downloading the files that have uploaded by any user as ‘public files’ |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application.  Any user has successfully uploaded a public file |
| Activation | This use case starts when the user clicks the ‘downloadpublicfile’ page. |
| Main flow | 1. The system displays a form for the user to search all public files 2. The user enters in a file name, type or description and clicks the ‘search’ button. 3. The system displays a list of files that meet the search criteria in a results table below the search form. 4. The user downloads the file they want by clicking on the file name. |
| Alternate flow | There is no alternative to the main flow in searching for and downloading a public file. |
| Exceptional flow | E1 : <The system cannot find the required files>   1. No files will be displayed in the results table. 2. The user will need to redefine their search criteria to include a file that has been previously uploaded to the system. 3. The use case continues at position 3 of the main flow |
| Termination | The user downloads the file. |
| Post condition | The system goes into a wait state. |

* + 1. **Requirement 9 < Search For and Download Files Shared by Specific Users >**
       1. **Use Case Scope & Priority**

This use case concerns the user searching for and downloading files specifically shared with them. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user searching for and then downloading files have been specifically shared with him/her by other users. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application.  Another member has successfully shared a file with the user |
| Activation | This use case starts when the user clicks the ‘downloadspecificfile’ page. |
| Main flow | 1. The system displays a form for the user to search all shared files 2. The user enters in a file name, type or description and clicks the ‘search’ button. 3. The system displays a list of files that meet the search criteria in a results table below the search form. 4. The user downloads the file they want by clicking on the file name. |
| Alternate flow | There is no alternative to the main flow in searching for and downloading specifically shared files. |
| Exceptional flow | E1 : <The system cannot find the required files>   1. No files will be displayed in the results table. 2. The user will need to redefine their search criteria to include a file that has been previously shared with them. 3. The use case continues at position 3 of the main flow |
| Termination | The user downloads the file. |
| Post condition | The system goes into a wait state. |

* + 1. **Requirement 10 <Ask a Question on the Study Forum>**
       1. **Use Case Scope & Priority**

This use case concerns the user creating a new question on the study forum. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process that a user would follow when they want to ask a new question on the study forum. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application. |
| Activation | This use case starts when the user clicks the ‘studyforum.php’ page. |
| Main flow | 1. The system displays a form for the user to create a new question. 2. The user types in their question. 3. The user clicks the ‘Ask Question’ button |
| Alternate flow | There is no alternative to the main flow in a user asking a question on the study forum. |
| Exceptional flow | E1 : <The user does not fill in all required fields in the form>   1. The system will display an error message 2. The user will need to complete all required fields 3. The use case continues at position 3 of the main flow |
| Termination | The user clicks the ‘Ask Question’ button. |
| Post condition | The question is stored in the database and displayed on the forum for other users to answer. |

* + 1. **Requirement 11 < Search for Questions and Answers on the Study Forum>**
       1. **Use Case Scope & Priority**

This use case concerns the user searching for questions and answers on the study forum. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process that a user would follow when they want to search through previously asked questions on the study forum. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application. |
| Activation | This use case starts when the user clicks the ‘studyforum.php’ page. |
| Main flow | 1. The system displays the study forum and a search box 2. The user enters in their search criteria 3. The user clicks the ‘Search Form’ button 4. The system will display the forum questions that relate to the users search criteria. |
| Alternate flow | There is no alternative to the main flow in a user searching through the study forum. |
| Exceptional flow | E1 : <The users search criteria is not met>   1. The system will not display any forum questions 2. The user will need to amend their search criteria 3. The use case continues at position 3 of the main flow |
| Termination | The user clicks the ‘Ask Question’ button. |
| Post condition | The system displays the forum questions that relate to the users search criteria. |

* + 1. **Requirement 12 < Reply to a Question on the Study Forum>**
       1. **Use Case Scope & Priority**

This use case concerns the user replying to a question on the study forum. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process that a user would follow when they want to reply to a question on the study forum. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application. |
| Activation | This use case starts when the user clicks the ‘studyforum.php’ page. |
| Main flow | 1. The system displays the study form to the user. 2. The user selects the question they want to reply to. 3. The system displays that question and all its previous answers. 4. The system displays a text box for the user to enter in their answer. 5. The user types in their reply. 6. The user clicks the ‘Submit Reply’ button. |
| Alternate flow | There is no alternative to the main flow in a user replying to a question on the study forum. |
| Exceptional flow | E1 : <The uses does not fill in all required fields in the reply text box>   1. The system will display an error message 2. The user will need to complete all required fields 3. The use case continues at position 6 of the main flow |
| Termination | The user clicks the ‘Submit Reply’ button. |
| Post condition | The reply is stored in the database and displayed on the forum for other users to read. |

* + 1. **Requirement 13 <View Study Tips and Advice>**
       1. **Use Case Scope & Priority**

This use case concerns the user viewing study tips and advice. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user accessing study tips and advice once logged into the application |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application. |
| Activation | This use case starts when the user clicks the ‘study.php’ page. |
| Main flow | 1. The system displays detailed study tips and advice to the user 2. The scrolls up and down the page to read all displayed content. |
| Alternate flow | There is no alternative to the main flow in a user viewing the study content |
| Exceptional flow | None |
| Termination | The user either logs out of the application or navigates to another page in the application |
| Post condition | The system goes into a wait state. |

* + 1. **Requirement 14 <View Exam Preparation Tips and Advice>**
       1. **Use Case Scope & Priority**

This use case concerns the user viewing exam preparation tips and advice. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user accessing exam preparation tips and advice once logged into the application |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application. |
| Activation | This use case starts when the user clicks the ‘exam.php’ page. |
| Main flow | 1. The system displays detailed exam preparation tips and advice to the user 2. The scrolls up and down the page to read all displayed content. |
| Alternate flow | There is no alternative to the main flow in a user viewing the exam preparation content |
| Exceptional flow | None |
| Termination | The user either logs out of the application or navigates to another page in the application |
| Post condition | The system goes into a wait state. |

* + 1. **Requirement 15 <Log out of Application>**
       1. **Use Case Scope & Priority**

This use case concerns the user logging out of the application. This is a high priority use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user logging out of the system. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged onto the application. |
| Activation | This use case starts and finishes when the user clicks the ‘Logout’ button. |
| Main flow | 1. The system displays a ‘logout’ button in the top right hand corner of every webpage. 2. The use clicks the ‘logout’ button. 3. The user is now logged out of the application. |
| Alternate flow | There is no alternative to the main flow in a user logging out of the application. |
| Exceptional flow | None |
| Termination | This use case starts and finishes when the user clicks the ‘logout’ button. |
| Post condition | The user’s session data is wiped and they are directed back to the ‘index.php’ home page |

* + 1. **Requirement 16 <Create a New Quiz>**
       1. **Use Case Scope & Priority**

This use case concerns a user creating a new revision quiz. This is a low priority, optional use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user creating a new revision quiz. |
| Actors | Student, Lecturer |
| Precondition | The user has successfully logged into the application. |
| Activation | This use case starts when the user clicks the ‘createaquiz.php’ page. |
| Main flow | 1. The system provides a form for the user to create their quiz. 2. The user enters in a name for the quiz. 3. The user chooses the number of questions for the quiz. 4. The user enters in all the questions and possible answers. 5. The user selects what answers for each question are the correct answer 6. The user clicks the ‘Create Quiz’ button. |
| Alternate flow | There is no alternative to the main flow in a user create a new revision quiz. |
| Exceptional flow | None |
| Termination | The user clicks the ‘Create a Quiz’ button. |
| Post condition | The quiz is created and saved in a database. |

* + 1. **Requirement 17 <Take a Previously Created Quiz>**
       1. **Use Case Scope & Priority**

This use case concerns a user taking a previously created quiz. This is a low priority, optional use case.

* + - 1. **Use Case Description**

|  |  |
| --- | --- |
| Description | This use case describes the process of a user searching for and taking a previously created revision quiz. |
| Actors | Student, Lecturer |
| Precondition | 1. The user has successfully logged into the application. 2. A previous quiz has been created and saved to the database. |
| Activation | This use case starts when the user clicks the ‘takeaquiz.php’ page. |
| Main flow | 1. The system provides a form for the user to search previously created quizzes. 2. The user searches for a quiz by entering the quiz name. 3. The starts a quiz by clicking the ‘Start Quiz’ button. 4. The system displays the quiz on the screen. 5. The attempts the quiz and then clicks the ‘Finish Quiz’ button. 6. The system displays the user’s results on the screen. |
| Alternate flow | There is no alternative to the main flow in a user taking a previously created quiz. |
| Exceptional flow | None |
| Termination | The user clicks the ‘Finish Quiz’ button and is displayed their results. |
| Post condition | The system goes into a wait state. |

* 1. **Non-Functional Requirements**
     1. **Usability**

The system should be user friendly and require no previous technical knowledge to use all provided functionality. Each web page should ideally focus on one main function. If a single web page contains two or more primary functions and these can be easily separated into two separate web pages this should be done.

The design, colour scheme and layout of the web application should be simple, minimal and clean in keeping with the user requirement for a ‘clutter free’ website.

* + 1. **Performance/Response time requirement**

All web pages contained in this application should have a fast load time within the client’s browser in keeping with end user expectations. A maximum load time of 2.5 seconds 90% of the time is the end goal.

The system should provide fast data transfer speed as data is sent from the client browser to the MYSQL database via the webserver and then back to the client browser. This data transfer should appear seamless and instant to the end user.

* + 1. **Availability requirement**

The system should be available to users 24/7 with little or no downtime. If the system is unavailable to the users, a detailed error message should be displayed along with accurate information on when the system will be next available.

* + 1. **Security requirement**

The system should implement best practice in securing all user data stored in the database. User passwords should be encrypted using a secure encryption function such as MD5 or SHA1. Passwords should not be stored in clear text format in the database.

The system should be securely protected against both HTML injection attacks and SQL injection attacks. This added security should not hinder the systems performance in any way.

* + 1. **Reliability requirement**

The system should be able to handle concurrent access from multiple users without any significant decrease in performance, security or availability.

* + 1. **Extendibility requirement**

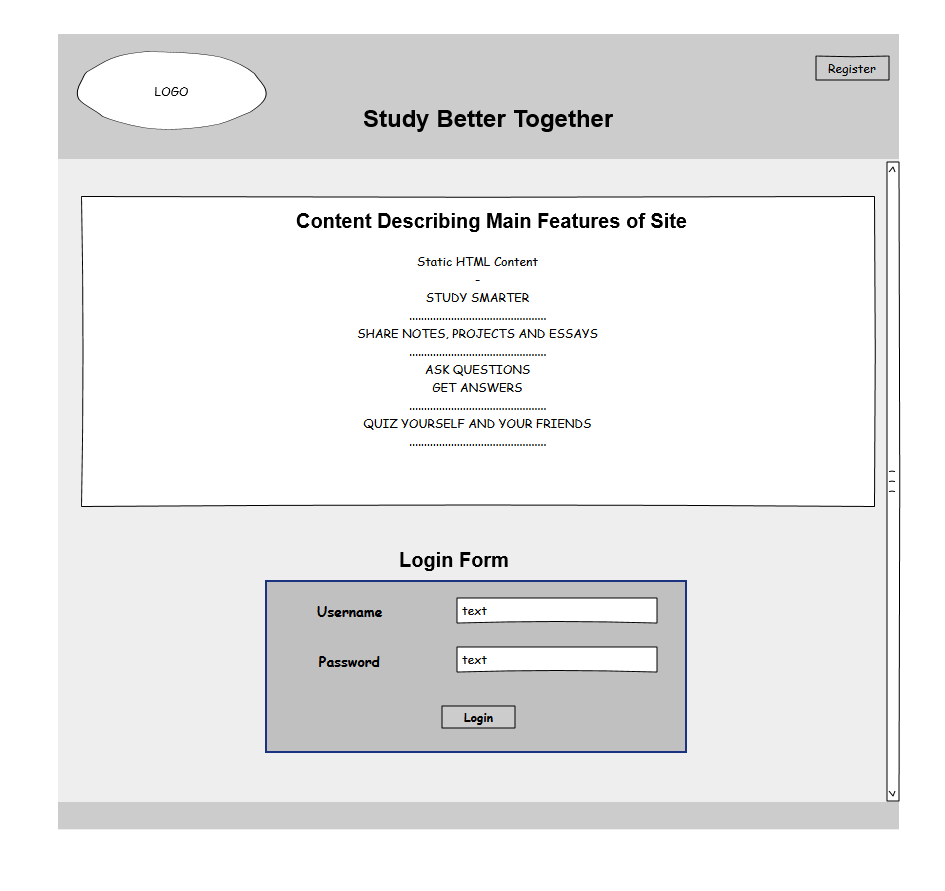
The system should be built in such a way as to allow extra features be added with relative ease.

1. **Interface requirements**
   1. **GUI**

Provided below are wireframe diagrams that were created in the conceptual planning phase and outline the proposed design for the front-end of the web application. The layout of these pages is subject to change as UI development begins. These wireframes proved an extremely useful tool in the requirement elicitation phase as they provided the client a visual representation of the proposed application.

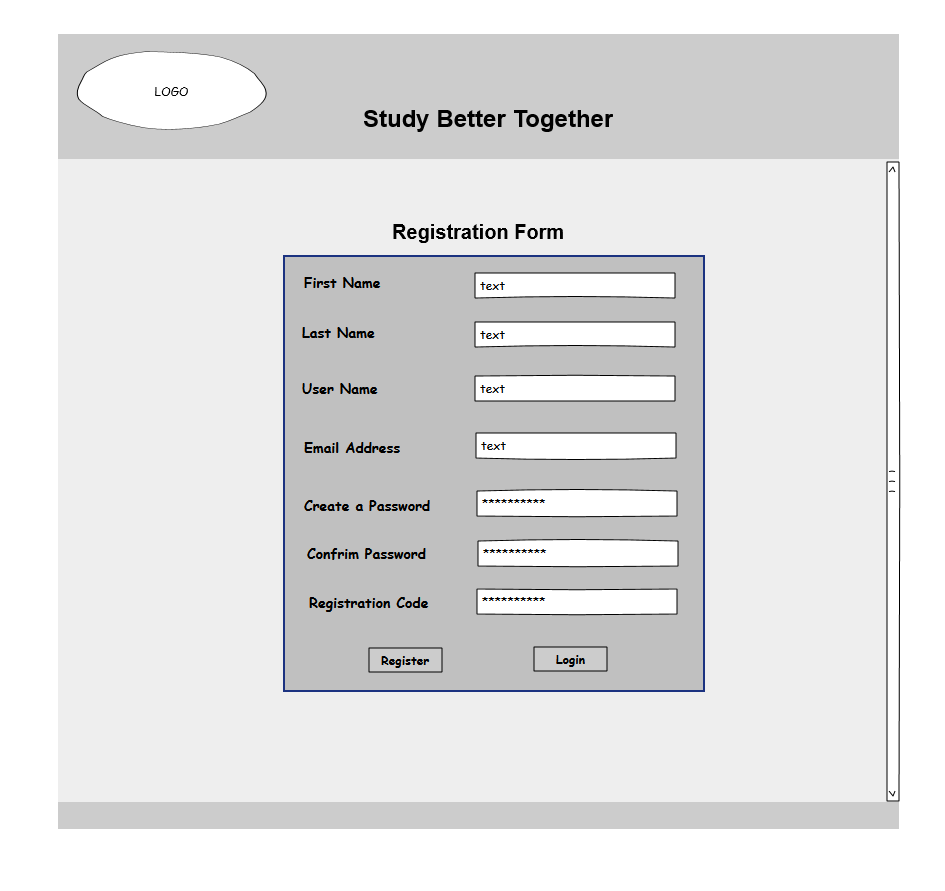
**Index Page**

This diagram represents the index/home page for the website. This is the page that people will see when they load the root URL - <http://studybettertogether.com/> into their client browser.



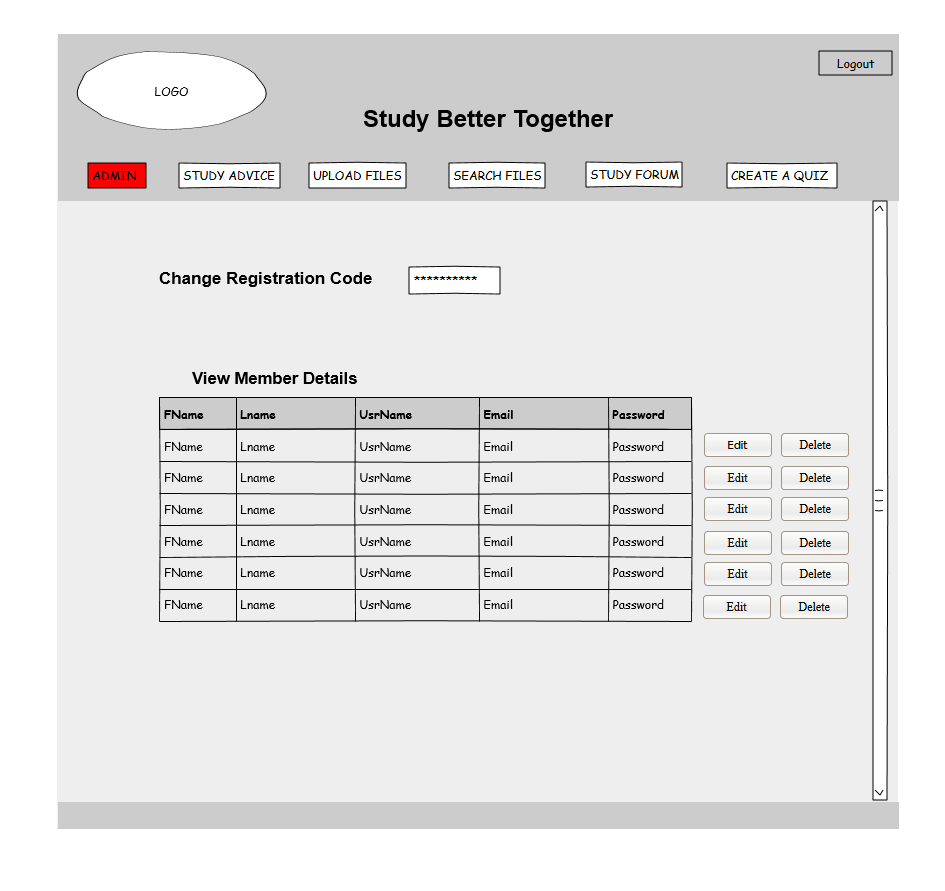
**Registration Section**

This diagram represents the student registration page which students will use to register for the web application. The students will require a registration code from the lecturer in order to create an account. This is a security measure to ensure that only the lecturer’s students can access the application. The lecturer herself will not need to register as she has already been allocated an account.



**Admin Panel Section**

This diagram represents the admin page which is the first page the lecturer will see when she logs in. The system will identify that she is the lecturer and thus redirect her to this admin page. Here she can change the registration code and update / delete user accounts.



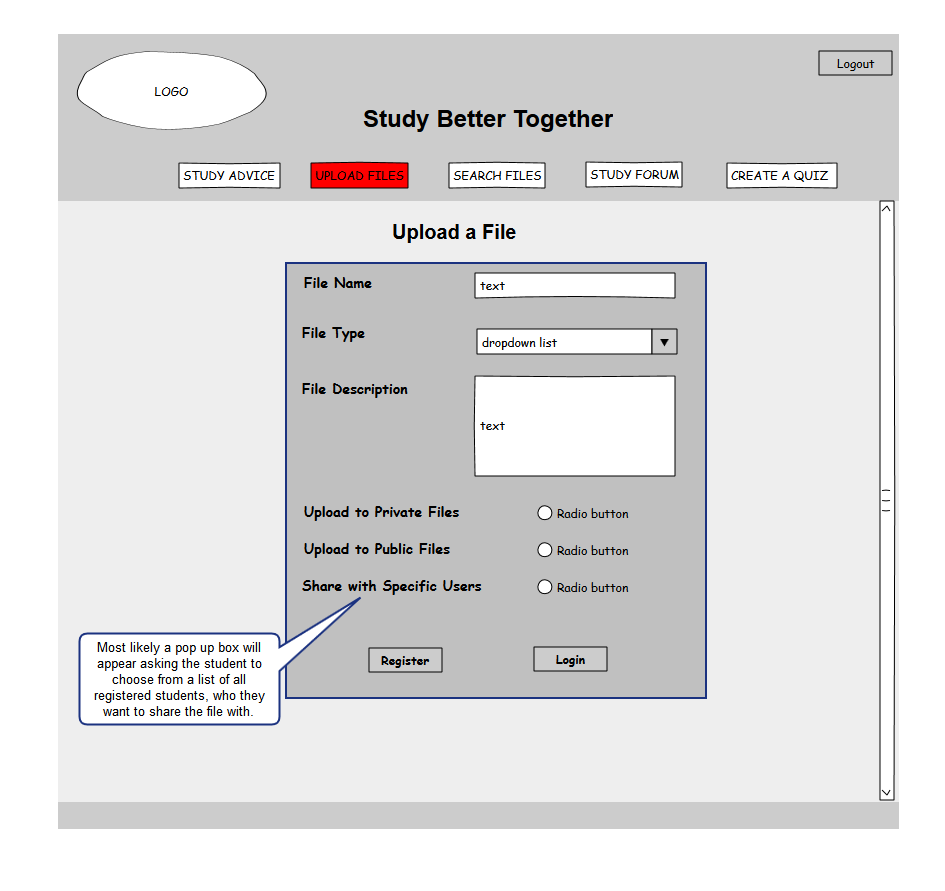
**Study Advice Section**

This diagram represents the study advice sections that will display detailed study hints, tips and advice to the students.



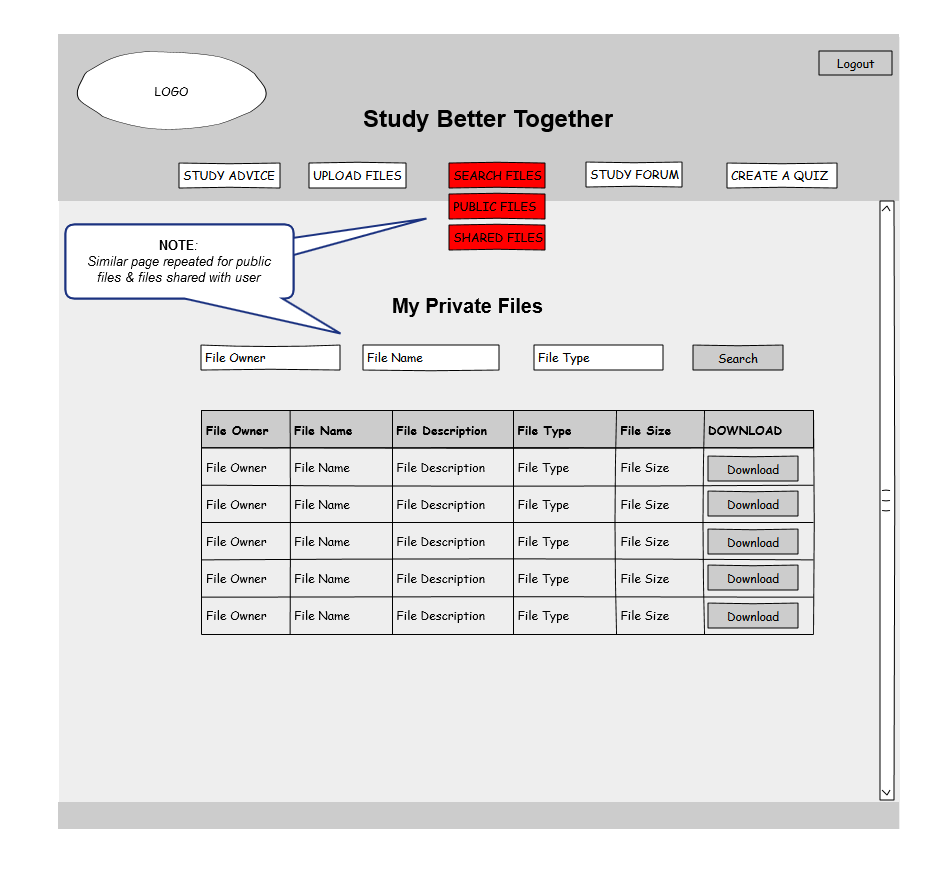
**File Upload Section**

This diagram represents the upload files page where users of the application will upload a file. Files can either be uploaded as personal files (not shared with anyone), public files (shared with everyone) or else shared with specific users.



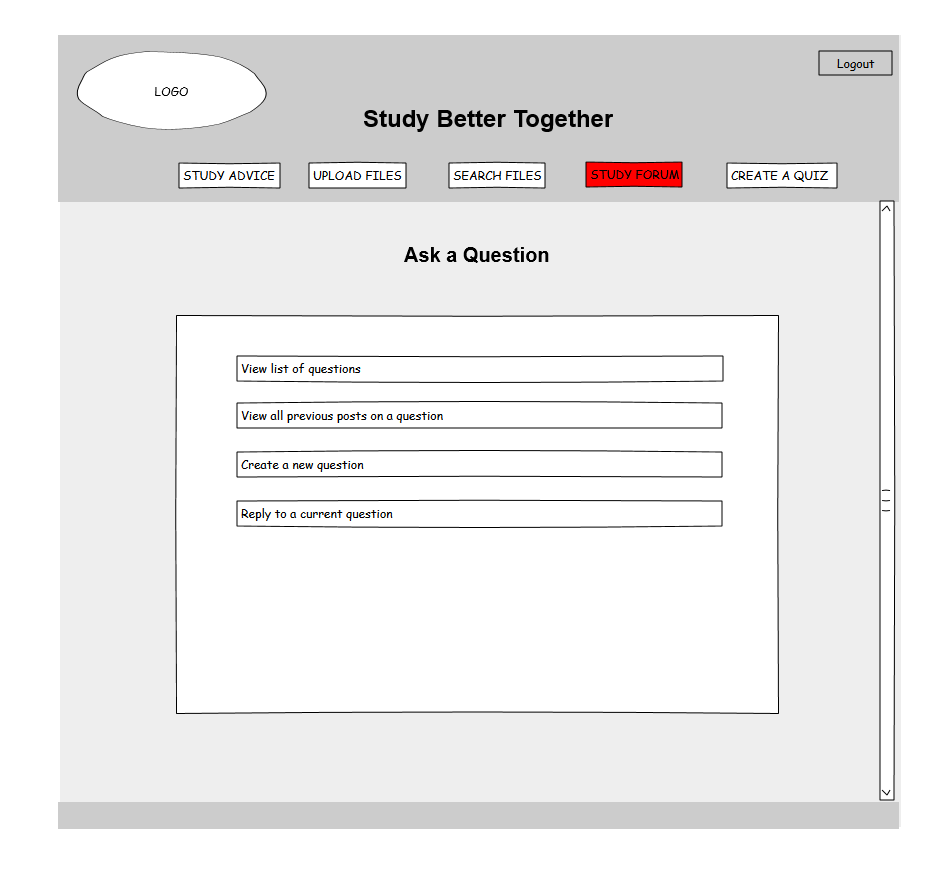
**File Search / Download Section**

This diagram represents the file search and download pages. Here the user can search for files based on several criteria and then choose the files he/she wants to download. Similar to the file upload options, files can be searched under three main headings – private, public and specifically shared.



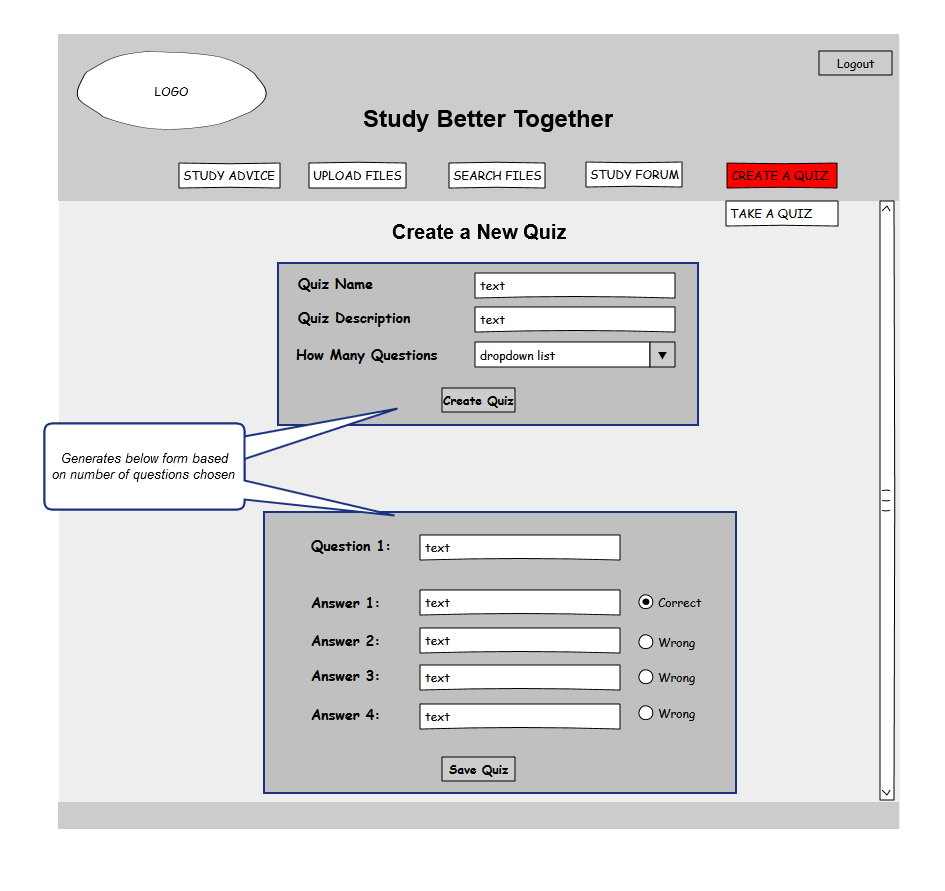
**Study Forum Section**

This diagram represents the study forum page which alongside the file-sharing functionality and the study advice pages will make up the required use cases. The study forum will be used to discuss all college related topics and enable both the lecturer and her students ask and reply to questions.



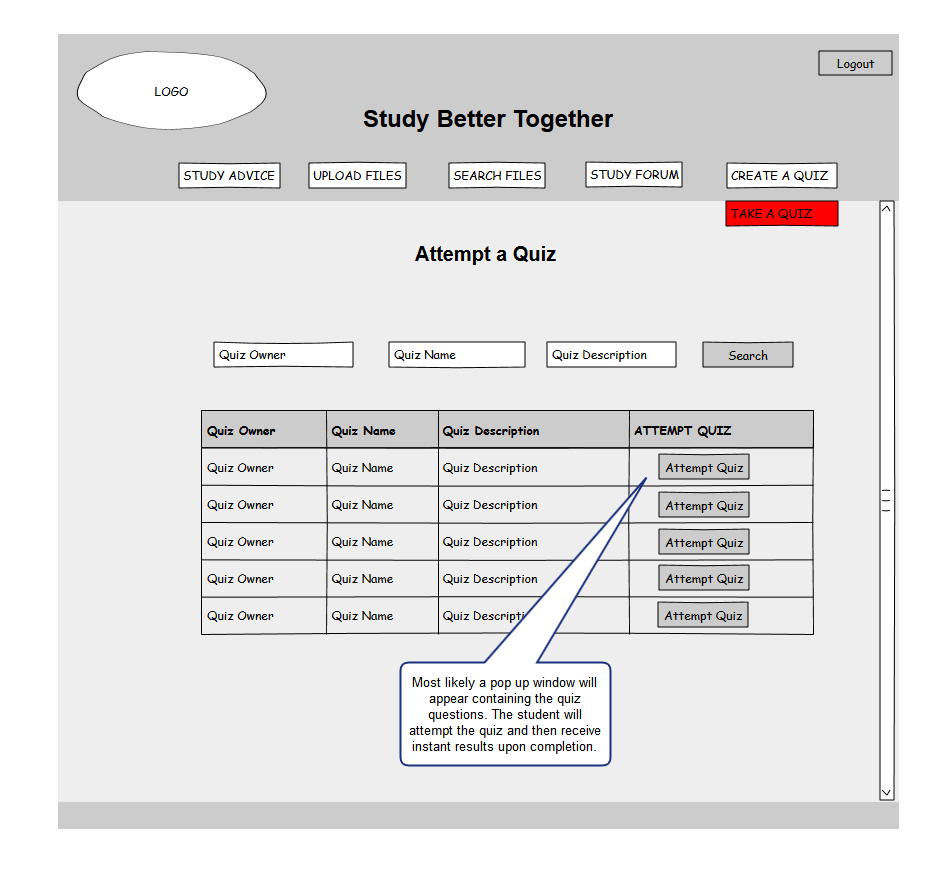
**Quiz Creator Section**

This diagram represents the quiz maker page which will enable users to create their own quiz for revision purposes. This is an optional use case and may not be implemented within the timeframe of this project.



**Quiz Taker Section**

This diagram represents the quiz taker page which will enable users to attempt previously created quizzes for revision purposes. As the purpose of these quizzes is quiz-based learning and not formative assessment, results will be displayed to the user upon completion but will not be permanently stored in a database. This is an optional use case and may not be implemented within the timeframe of this project.

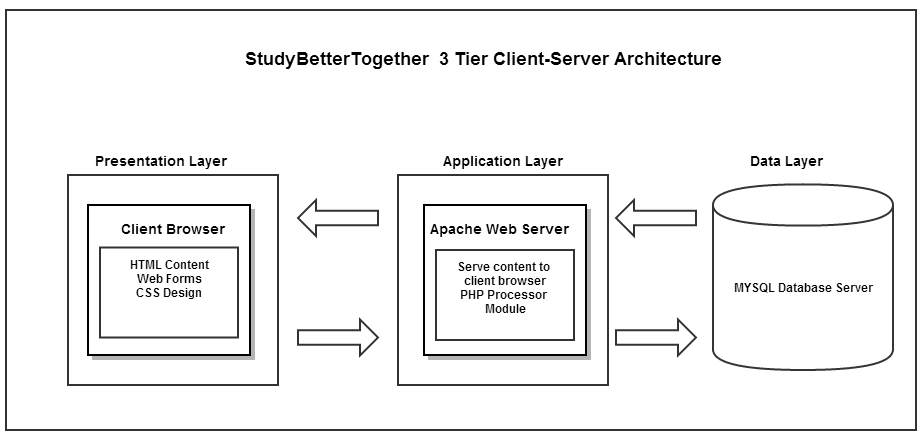
****

* 1. **Web Server**

This system will use the Apache Web Server to serve up the HTML content to the clients browsers and handle security and memory issues. The Apache Web Server will be configured to use a PHP processor module that will interpret the PHP code used for all server side programming of this application.

1. **System Architecture**

The following diagram shows the 3-Tier client-server architecture that this web application will utilize.



1. **System Evolution**

As evidenced by the results of the requirements elicitation questionnaire, there is still a strong need for collaborative learning applications that facilitate student to student interaction as oppose to the classical teacher to student interaction offered by most mainstream learning management systems. I believe this application will provide a valuable collaborative learning tool to students and has tremendous scope for growth after the allocated development period running up to May 2013.

The next major step after May 2013 would be to implement the mobile version of this project. This would provide the accessibility benefits as specified in the project proposal.

1. **Appendix**
   1. **Results from Requirement Elicitation Questionnaire**

**Studybettertogether.com - Requirement Elicitation Questionnaire – October 2013**

<http://www.surveymonkey.com/s/5VMRBWL>

**Q1. My project is a collaborative study website that allows students upload and share files with each other online, ask questions and discuss topics in a study forum and create and share quizzes with each other for revision purposes.  
Do you think such a website would be beneficial to you in helping you study better?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifYes  https://www.surveymonkey.com/i/t.gifNo |

**Q2. Do you prefer learning on your own or with other people?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifOn my own  https://www.surveymonkey.com/i/t.gifWith other people  https://www.surveymonkey.com/i/t.gifA mixture of both |

**Q3. How beneficial have you found studying collaboratively with other students?  
(1 being not beneficial at all / 5 being extremely beneficial)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://www.surveymonkey.com/i/t.gif1 | https://www.surveymonkey.com/i/t.gif2 | https://www.surveymonkey.com/i/t.gif3 | https://www.surveymonkey.com/i/t.gif4 | https://www.surveymonkey.com/i/t.gif5 |

**Q4. Have you ever shared any learning resources (eg. study notes, essays, projects ) with other students for the purpose of helping each other learn or study better? Did this help you learn / study better?**

|  |  |
| --- | --- |
| https://www.surveymonkey.com/i/t.gifYes - it was very helpful  https://www.surveymonkey.com/i/t.gifYes - it was somewhat helpful | https://www.surveymonkey.com/i/t.gifYes - It was not helpful  https://www.surveymonkey.com/i/t.gifNo |

**Q5. If yes - What is the primary means of file transfer that you most frequently used?**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifEmail  https://www.surveymonkey.com/i/t.gifDropbox  https://www.surveymonkey.com/i/t.gifGoogle Drive  https://www.surveymonkey.com/i/t.gifSky Drive  https://www.surveymonkey.com/i/t.gifMoodle |
| Other (please specify) |

**Q6. Have you ever used quizzes (ie. MCQ quiz) to help yourself learn or study for an exam and did you think they were a helpful tool for revision?**

|  |  |
| --- | --- |
| https://www.surveymonkey.com/i/t.gifYes - I found them a helpful tool for revision  https://www.surveymonkey.com/i/t.gifNo - but I would like to try it | https://www.surveymonkey.com/i/t.gifYes - but I did not find them a helpful tool for revision  https://www.surveymonkey.com/i/t.gifNo - and I wound not be interested in trying it |

**Q7. Would the ability to easily create your own revision quiz and share it with other students (and have access to previously shared quizzes) be something that you would find useful in helping you study?  
(1 being not useful at all / 5 being extremely useful)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://www.surveymonkey.com/i/t.gif1 | https://www.surveymonkey.com/i/t.gif2 | https://www.surveymonkey.com/i/t.gif3 | https://www.surveymonkey.com/i/t.gif4 | https://www.surveymonkey.com/i/t.gif5 |

**Q8. Have you ever had questions regarding your studies and have not received a satisfactory answer for any of the following reasons:  (tick all that apply)**

|  |
| --- |
| https://www.surveymonkey.com/i/t.gifNot known who to ask  https://www.surveymonkey.com/i/t.gifEmailed a lecturer and had to wait a long time for a response  https://www.surveymonkey.com/i/t.gifEmailed a lecturer and never received a response  https://www.surveymonkey.com/i/t.gifEmailed another student but they were too busy to reply or did not know the answer  https://www.surveymonkey.com/i/t.gifWanted to ask other students but were too shy |
| Other (please specify) |

**Q9. Would a study forum where you could ask questions, search for previous questions and discuss any college related topics be something that you would find useful?  
(1 being not beneficial at all / 5 being extremely beneficial)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https://www.surveymonkey.com/i/t.gif1 | https://www.surveymonkey.com/i/t.gif2 | https://www.surveymonkey.com/i/t.gif3 | https://www.surveymonkey.com/i/t.gif4 | https://www.surveymonkey.com/i/t.gif5 |

**Q10. Please rank in order the features of this website that you would find most useful?   
(1 being the most useful)**

Upload and share files with other users

Upload files to a personal folder so only you have access

Upload and share files with only specific users that you choose

Access to detailed study tips, hints and advice

Access to advice for handling exam pressure

Create a MCQ quiz and share this with other users

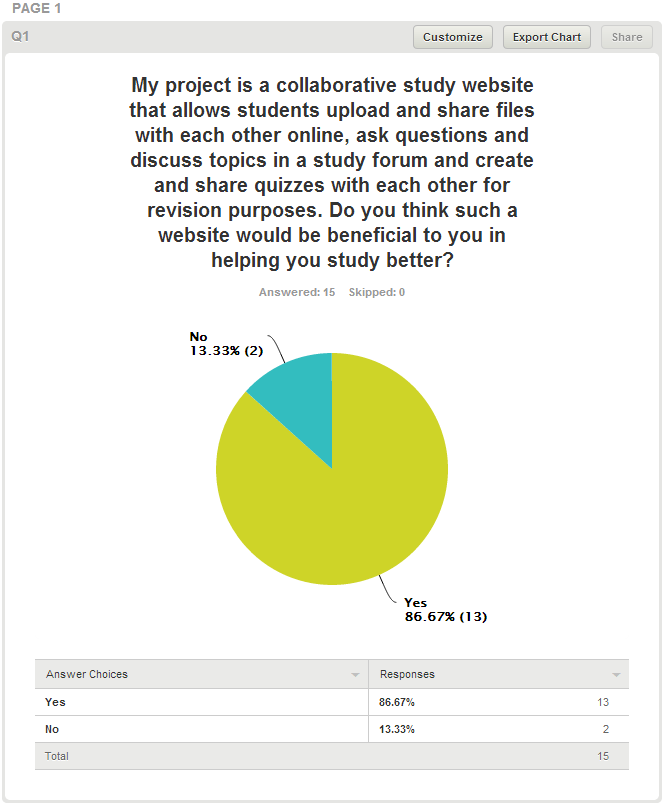
Create a quiz that requires typing in the answer. (An option would be available to provide a hint if you are stuck.)

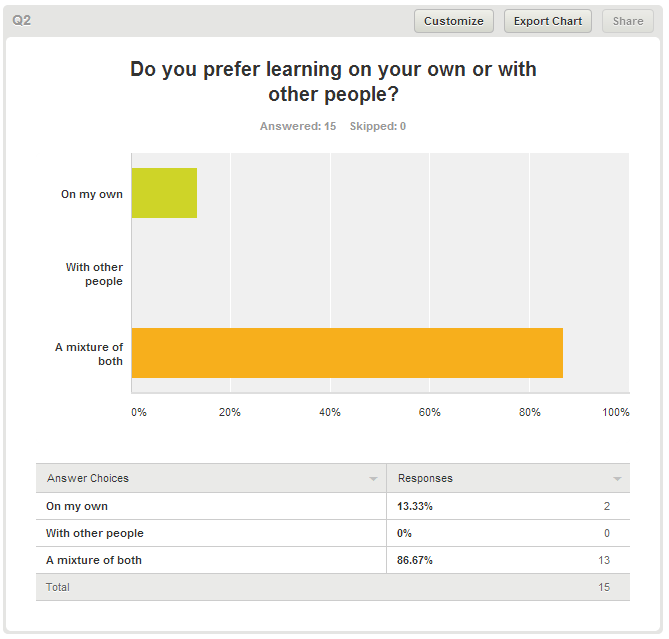
Access to a study forum where you could post questions, search questions and reply to questions

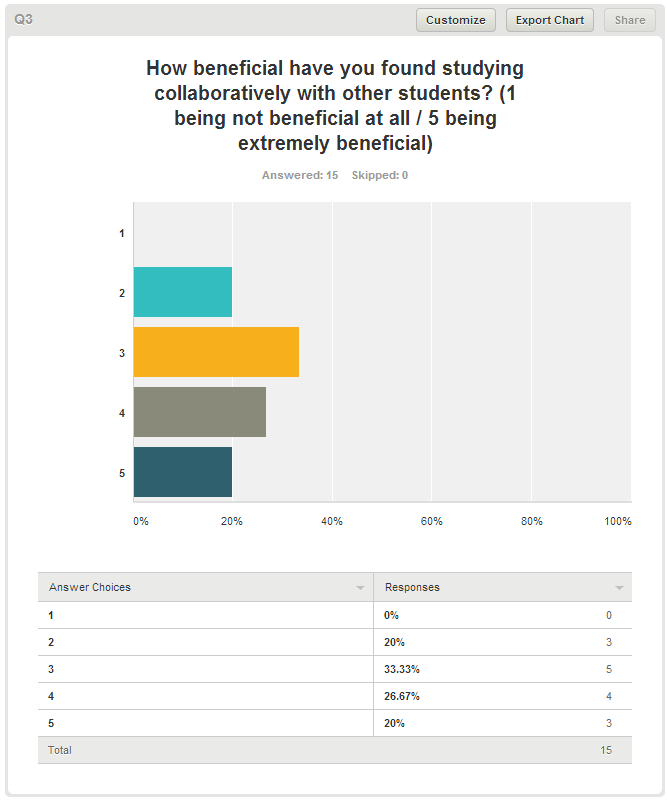
**Study Better Together – Requirement Elicitation Questionnaire – Results**

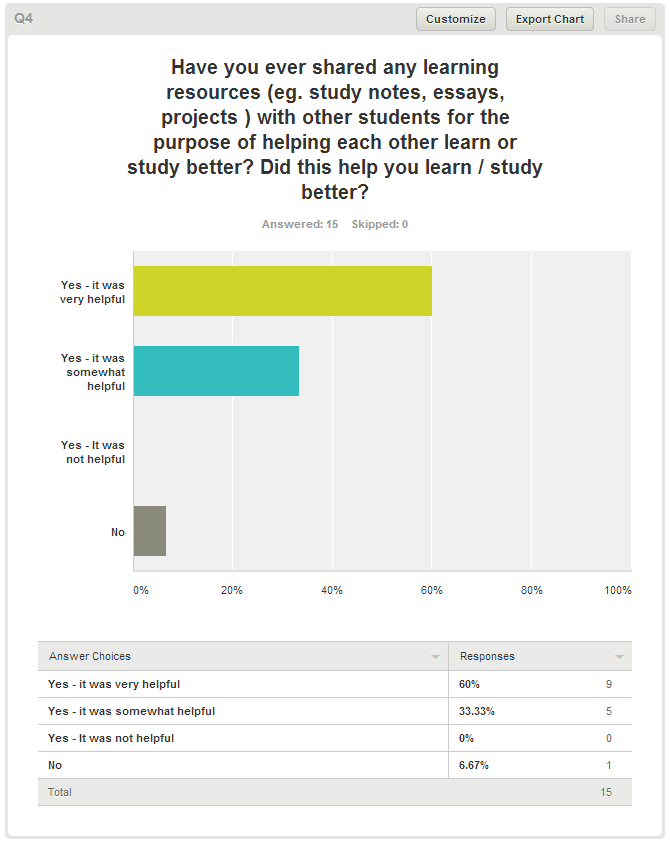
<http://www.surveymonkey.com/s/5VMRBWL>

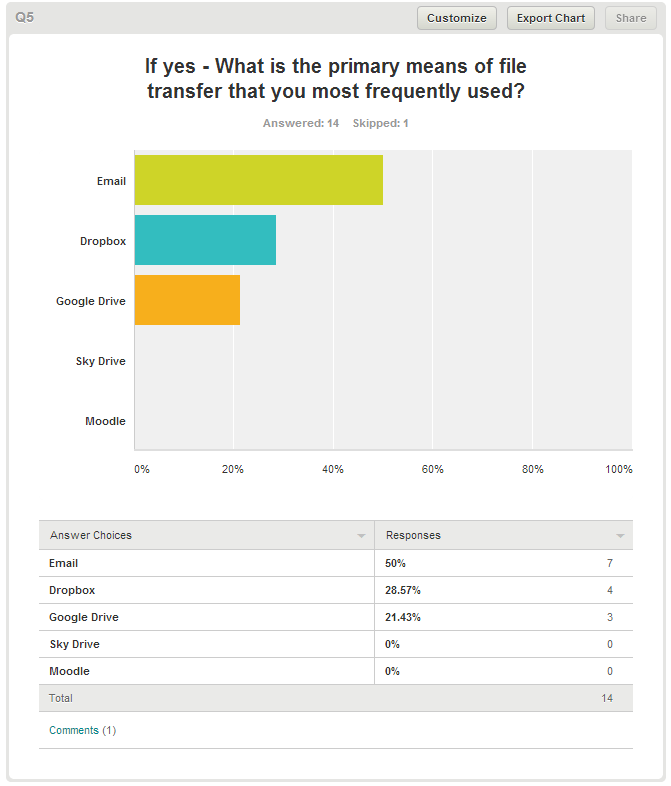
**Note**: Due to the restrictions placed on exporting results from a free Survey Monkey account, I have had to take screenshots of each result.

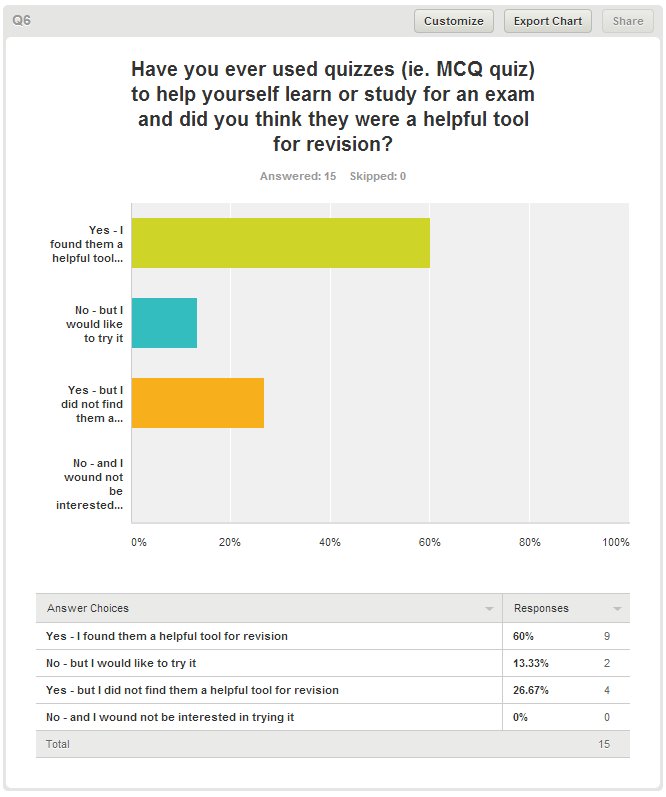


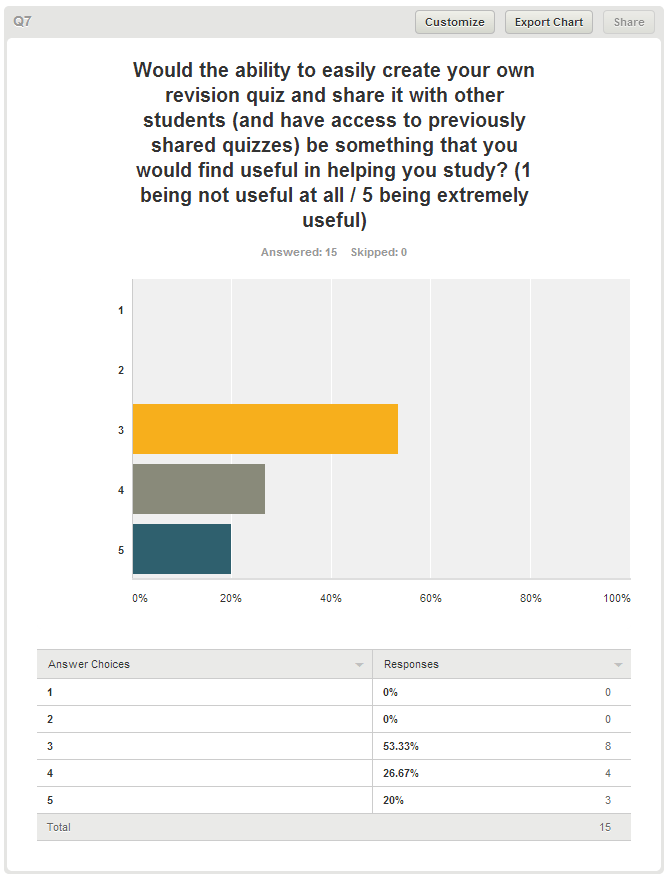




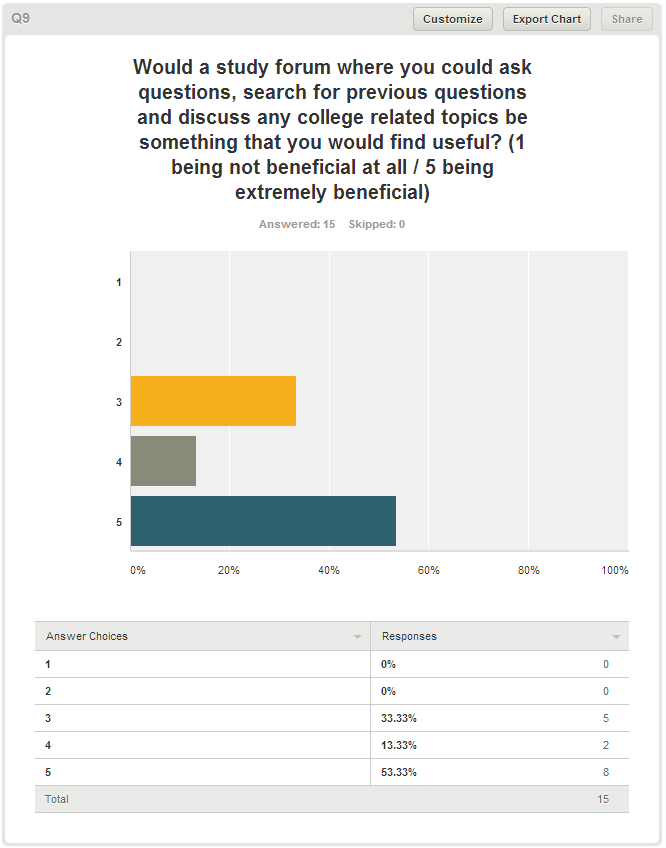


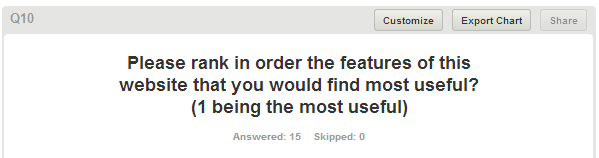
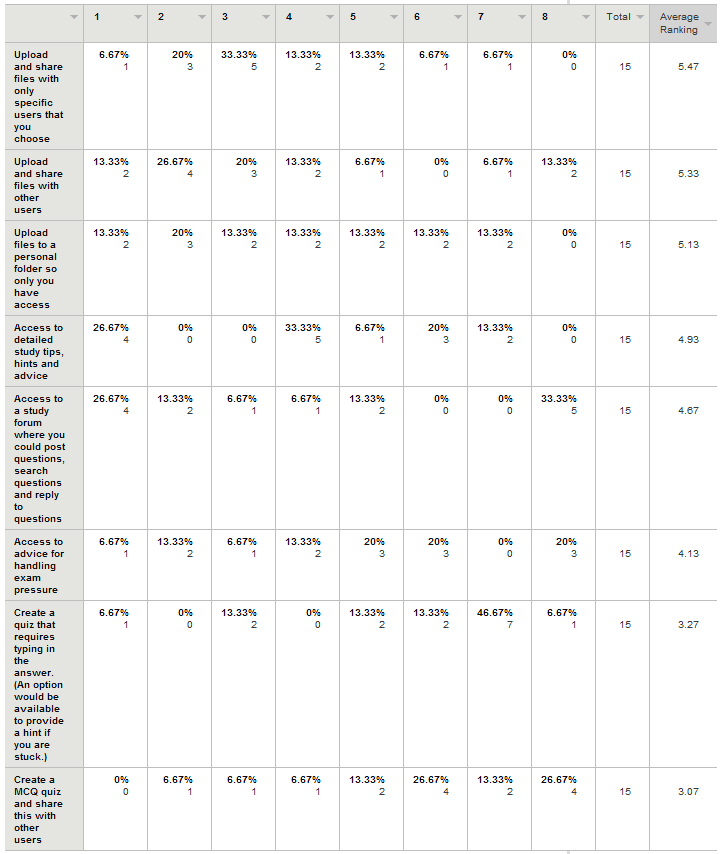












## Monthly Journals

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: September 2013

**My Achievements**

The main achievement for this month was to outline and decide a project idea to use. This process involved creating a shortlist of seven different project ideas that I was interested in. I then conducted a brief feasibility study into each idea assessing if it was possible to create and if I could find a real life client for it. This feasibility study narrowed my project short list down to three ideas. I sent these ideas to my course coordinator who gave his feedback on my ideas. Based on this feedback I then choose my idea to implement. After choosing my idea, I documented it in a project proposal.

**My Reflection**

Deciding on a project idea was a lot tougher than I expected. As this was my first time working on a medium to large scale project myself and a large percentage of our degree marks are allocated towards this project I found it quite pressurised to choose a project. I was torn between choosing an idea that I thought I would be able to implement within a year and an idea that would serve a useful and functional purpose. I am very happy however that I have now reached the stage where I have my project idea and can progress with the next stage in my project – the requirement specification. I think I am now in a great position to start the requirement analysis stage and then start creating the design documentation for my project.

**Intended Changes**

Next month, I aim to complete my requirement specification and start coding my project prototype.

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: October 2013

**My Achievements**

This month a number of achievements have been made.

1. I have been appointed my Project Supervisor – Paul Hayes and have discussed my project idea with him in further detail. I talked to Paul about my progress to date and my future plans for building the website. He is of the opinion that I am making good progress.
2. I undertook a detailed requirements elicitation process which consisted of client meetings, front-end prototyping using wireframe diagrams and a sending out a requirement elicitation questionnaire. I received 15 responses from the questionnaire and have used these to help shape the requirements for my project.
3. I completed my requirements specification document ensuring that it was as detailed and comprehensive as I could make it (49 pages including appendixes).

**My Reflection**

I found this month to be more productive than last month mainly because I had my project idea chosen and could start getting to work on it. I put a lot of work into the requirements specification document this month and I was happy with the final document. I met up with my client who was impressed with the amount of requirements gathering and prototyping (wireframe diagrams) I had completed.

Despite the progress made this month on the RS, I am struggling with starting to create the prototype. I have purchased two development books book on PHP and MYSQL and have been reading through them but I am finding it to be a slow learning curve. I have also started brushing up on HTML and CSS which I will use for the front end of my website but again am finding it a lot to take in.

**Future Plans**

My plan for next month is to continue trying to learn PHP, MYSQL, HTML and CSS and then start building the initial prototype version of my website. By the end of next month I aim to have the structure of my website built using HTML /CSS and then also create the login, registration and manage user’s functionality.

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: November 2013

**My Achievements**

This month a number of achievements have been made.

1. In terms of front end design I have created the structure of my website using HTML and CSS. I have also added text and images to the users profile page.
2. In terms of back end functionality, I have created a prototype version of the following features:

* User login
* Student registration
* The ability for the administrator to change the registration code
* The ability for the user to change their password
* File Upload (ongoing)
* Database structure (ongoing)

1. Other achievements this month included a one hour meeting with my client to review the website and discuss any possible changes in user requirements. The client – Mimi, is delighted with the work so far.
2. I also had a productive meeting with my supervisor in which we reviewed the website and discussed my future plans. Paul was very impressed with my progress to date.
3. I created a GitHub repository for my project and am using this for version control. Link: <https://github.com/andrewmgh/studybettertogether/tree/master> -

**My Reflection**

This month, I put an enormous amount of work into the development of my website. At the start of the month I made a decision not to let my lack of experience in PHP stop me from developing an excellent project. I literally worked day and night trying to learn PHP and slowly start building my website. Looking back at my progress this month, I am delighted with myself and feel the hard work has paid off.

One concern I had was that my style of coding may not be appropriate for a fourth year project. As I am learning PHP for the first time, I am using a procedural / functional style of coding rather than a fully object orientated approach. However I sent samples of my code to both my supervisor and project coordinator for review and they both confirmed that my style of coding was ok. This was excellent feedback to receive.

In my client meeting this month, Mimi has requested some additional features be added in place of the quiz maker functionality. We are in the process of ironing out these additional features and they will be documented in next month’s journal.

**Future Plans**

My plan for next month is to continue working on the file upload and file search functionality. I have my prototype presentation on the 16th December so I’m aiming to have both of these features working (at least to an initial prototype level) by then.

Next month I also aim to implement the following security features:

* Salt and encrypt passwords in database
* Implement session control to stop unregistered users from accessing the website
* Remove the spaces from the uploaded file names to avoid errors on Linux systems
* Sanitise all user input to avoid SQL injection and Cross Site Scripting attacks

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: December 2013

**My Achievements**

This month has seen number of developments to the website along with a successful prototype presentation. I also met up with Mimi to discuss and review the website. She is very happy with the progress to date and highlighted that the simple and user-friendly design of the website is excellent.

In terms of artefact development this month I have:

1. Completed the file upload, search and download functionality for both public and personal files.
2. Completed the front-end design to a prototype standard.
3. Implemented session control to stop unregistered users from accessing the website.
4. Successfully implemented SHA1 encryption (and salting) of user passwords along with sanitizing all user input to protect against possible injection attacks.

On the 16th December I had a successful prototype presentation with Vikas Sahni in which I presented my project idea and demonstrated the website in its current form.

**My Reflection**

I am very happy with the progress made this month. In particular I am delighted with how my prototype presentation went and the feedback received from Vikas.

Vikas was impressed with my project so far and the work I have completed to date. He only had two changes for me to implement for next May.

1. To make my search form sticky so the search criteria won’t disappear when the results are displayed.
2. Not to code my own study forum but to use an open source forum (choose from the many open source PHP forums available).

For my prototype he gave me a result of a 2.1. He stated that the only reason he is not giving me a 1st is that some of my advanced features (specific-user file sharing) are not yet implemented.

He also stated that as a BIS student I am not expected to have as complicated code written as the computing students, but I need to ensure my documentation and end user testing is excellent. This was great feedback to receive as I have already completed a thorough RS document (50 + pages) and plan to complete end-user testing with the client along with a detailed technical report by next May.

**Future Plans**

My plans for next month are to:

* Work on creating a draft version of the specific-user file sharing functionality.
* Research the available options for implementing an open source forum.

I aim to involve the client in the process of deciding what forum to use by researching and creating a short list of suitable forums, showing them to the client and then together choosing which forum I will implement.

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: January 2014

**My Achievements**

The main achievement this month was the successful implementation of the specific-user sharing functionality – at least to a first draft level.

I was able to incorporate a JQuery show/hide effect based on advice found on stack overflow - <http://stackoverflow.com/questions/7019096/jquery-dropdown-hide-show-div-based-on-value> along with a dropdown check-list plugin I found at - <https://code.google.com/p/dropdown-check-list/> to create the front end for this specific-user sharing. When a user selects Specific Sharing as a sharing option a new dropdown box appears on the upload page that contains a list of other users on the website which can then be selected to share files with. I also created the backend functionality for this specific-user sharing using PHP and MYSQL.

This month I also decided what open source forum to use ([http://mylittleforum.net](http://mylittleforum.net/)) and have been working on trying to integrate this into my website.

**My Reflection**

I am very happy that I managed to implement the specific-user sharing functionality. I really struggled with the database design aspect of this however and eventually found the most efficient way to implement this functionality was to go against the rules of database normalisation. In my database I now have a table called *file\_sharing* where each row represents a particular file. In this table I have a column called [*shared\_with*](https://localhost/phpmyadmin/sql.php?db=sbt&table=file_sharing&sql_query=SELECT+%2A+FROM+%60file_sharing%60%0AORDER+BY+%60file_sharing%60.%60shared_with%60+ASC&session_max_rows=30&token=ec0ed4b4bcabe12b622139b54ded3fff) that holds the usernames of all users that the particular file is shared with. I know this method breaks the rules of database normalisation but I have also read that sometimes in software design it is faster and more efficient for your software to use a non-normalised database, particularly if a lot of searching will be conducted on your database – as is the case in this project. To further aid the speed of my application I will be applying a full index to tables that will be searched through regularly.

This month I was happy to have decided on an open source forum to use. I researched several PHP forums and discussed the possibilities with the client. Based on the client’s feedback, the requirements of this project and my ability to integrate the forum I have chosen to use a forum called - [http://mylittleforum.net](http://mylittleforum.net/).

**Future Plans**

The client has requested that I include some additional functionality to my project - she want the students to be able to choose their degree programme or class when they register. She also wants the ability to add new classes and see how many students are registered for each class. I will work on this functionality next month.

My plans for next month are to:

* Work on the admin sections of the website, namely manage users and the new manage classes section
* Continue integrating the study forum into the website
* Add the functionality to allow students delete their files
* Investigate the possibility of using JQuery to show file information in a pop-up box instead of putting all the information in one HTML table

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: February 2014

**My Achievements**

This month has seen a number of significant developments in my project such as:

* Successful integration of the study forum into my website. The forum tables in the database are now linked to my tables so when a user is added, updated or deleted from my users table this is automatically reflected in the forum tables.
* The new manage users and manage classes sections requested by the client last month have been successfully implemented. The client can now create, update and delete degree classes which the students can then choose from upon registering for the website. There is also a feature to show how many students are registered to each class. In addition to this, the client can now edit student details and reset their passwords.
* I changed the design of my *file\_sharing* table to store the user\_id instead of the username in the [*shared\_with*](https://localhost/phpmyadmin/sql.php?db=sbt&table=file_sharing&sql_query=SELECT+%2A+FROM+%60file_sharing%60%0AORDER+BY+%60file_sharing%60.%60shared_with%60+ASC&session_max_rows=30&token=ec0ed4b4bcabe12b622139b54ded3fff) column. This enables the client to change (if desired) a student’s username without affecting any files previously shared with that student.
* The users of my website can now delete any files they have uploaded. For security reasons, this feature is only available when the currently logged in user is also the file owner.
* I implemented a different JQuery slider on the index.php page as the previous slider – UnSlider – was causing incredibly frustrating CSS design errors when viewed using the Google Chrome browser.
* I have started trying to upload the work I have completed so far to the online location <http://studybettertogether.com>. This is proving difficult and I hope to make more progress on this next month.

This month I also had another successful meeting with the client – Mimi Kelly – where we spent over an hour reviewing the website. Mimi is delighted with the progress to date and has requested another feature be added to the website – an instant chat room feature to allow all logged in users instantly chat together. I believe this new feature will complement the existing study forum in providing excellent collaboration tools for the users of this website.

**My Reflection**

I am delighted with the progress of my website so far. It was encouraging to hear positive feedback from Mimi after we reviewed the website together. I have already started looking into open source chat rooms that I will try integrate into my website in the same manner as I did the study forum.

One issue I am facing is the allocation of my time. While I have made huge progress this month, I ended up spending a large majority of my time working on this project and although this had proven beneficial I am conscious that I cannot ignore the other subjects I have in my degree. With this in mind, I intend to reduce my time allocated to this project next month and spend more time working on my other subjects.

**Future Plans**

I am having a lot of difficulty uploading online the work I have completed so far. The main problem I am facing is that errors are appearing in the online version of my site that are not appearing when the site is run on Localhost. I have been searching for answers on <http://stackoverflow.com/> and hope to have these issues resolved by the end of next month.

My plans for next month are to:

* Integrate an open source chat room as per the client’s request
* Work on resolving the errors that are occurring on the online version of the website
* Add content to the help page and study advice pages
* Work on the CSS and overall design of the website

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: March 2014

**My Achievements**

Project developments for this month included:

* Successful integration of an open source chat room (<http://frug.github.io/AJAX-Chat/>) as per the clients request last month.
* Most of the errors occurring on the live site have been resolved. These mainly involved session control errors and the following heading error: “*Warning: Cannot modify header information - headers already sent by (output started at…)* “.
* Content was added to the study advice and study resources pages
* Client side validation was added to the all the user entry forms (login, registration, file upload etc) through the use of JavaScript and HTML 5. Using client side validation aids in the usability of the website as it provides a seamless validation process. However I am also using server side validation for increased security.

I had another meeting with the client this month. Mimi is very happy with the website and in particular the new chat room feature. She feels the website is almost ready for end-user testing.

**My Reflection**

After another productive month I am very happy with the progress I have made on my project.

This month did however involve a lot of difficult and frustrating work. In particular I found integrating the chat room feature to be a very difficult and time consuming job. The hardest part was trying to read through and understand the code used in the chat room files. Embedding the chat room into my website and combining the chat room tables with my own database tables also proved to be quite difficult.

From reviewing the website with my client, I estimate that the website is now about 95% complete with only CSS changes and the “Help” page left to be completed.

**Future Plans**

Next month will see the completion of the website and the commencement of end-user testing.

Specifically my plans are to:

1. Finish the CSS and design of the website (the chat room login page still requires a lot of work).
2. Create a user manual and embed this manual into the “Help” page.
3. Upload these changes to the live site.
4. Complete integration testing to ensure all features are working together.
5. Create a user acceptance testing feedback questionnaire.
6. Commence end-user testing.

Reflective Journal

**Student name:** Andrew Monaghan

**Programme:** Business Information Systems

**Month**: April 2014

**My Achievements**

In the last month I have completed the development of my website, integrated all changes into the live site (<http://studybettertogether.com/> ) and also conducted end-user testing.

Some of the specific objectives completed this month have been:

* Finalised all CSS and design elements of the website
* Conducted a complete review of the website with the client and made several changes to the text content
* Integrated all of these changes into the live site
* Created a user manual which is embedded into the *help.php* page via an *iframe*
* Created an end-user acceptance testing questionnaire
* Commenced end-user testing with students from both the clients college (IBI) and from NCI

**My Reflection**

Despite having my semester two exams this month, I am delighted that I was able to finish my project. Already I have received positive feedback from my supervisor and from students conducting end-user testing which is encouraging.

A number of small issues have been identified during end-user testing which I have so far been able to fix. One such issue was that when class details were being updated, these changes weren’t reflected in the study forum categories section. To fix this I added a column to the forum categories table called “*class\_id*” which linked to the “*class\_id*” field in the classes table. I then changed the PHP code in the “*editclasses.php*” page to update both of these tables.

I did likewise to the “*deleteclass.php*” file so that now when the client deletes a class, this also reflects in the categories section of the study forum.

I showed the final version of my website to my supervisor Paul who stated it is excellent work. He has recommended my project to the project coordinator Jonathan as an outstanding project, which I am delighted about.

**Future Plans**

As the project is due up in three weeks’ time my remaining plans are to complete end-user testing, create the technical report document and present my project to NCI faculty. I foresee no problems in completing these three tasks on time.

## User Manual

**User Manual**

**Study Better Together**

**A Collaborative Learning Website**

[**http://studybettertogether.com**](http://studybettertogether.com)

**Contents**

ACCESSING THE WEBSITE AND LOGGING IN 2

CREATING A NEW ACCOUNT 3

THE HOME PAGE 4

SITE NAVIGATION 5

UPLOADING FILES 6

SEARCHING AND DOWNLOADING PUBLIC FILES 8

SEARCHING AND DOWNLOADING YOUR FILES 9

USING THE STUDY FORUM 11

USING THE CHAT ROOM 13

STUDY PAGES 14

HELP PAGES 14

ADMIN USE ONLY – MANAGE USERS 15

ADMIN USE ONLY – MANAGE CLASSES 16

ADMIN USE ONLY – FORUM SETTINGS 17

**Accessing the Website and Logging In**

To access the website, open your browser and navigate to <http://studybettertogether.com>. You will then see the following page where you have the option to either log in with an existing username and password or else register for a new account. Once logged in you will be redirected to the Home Page.

****

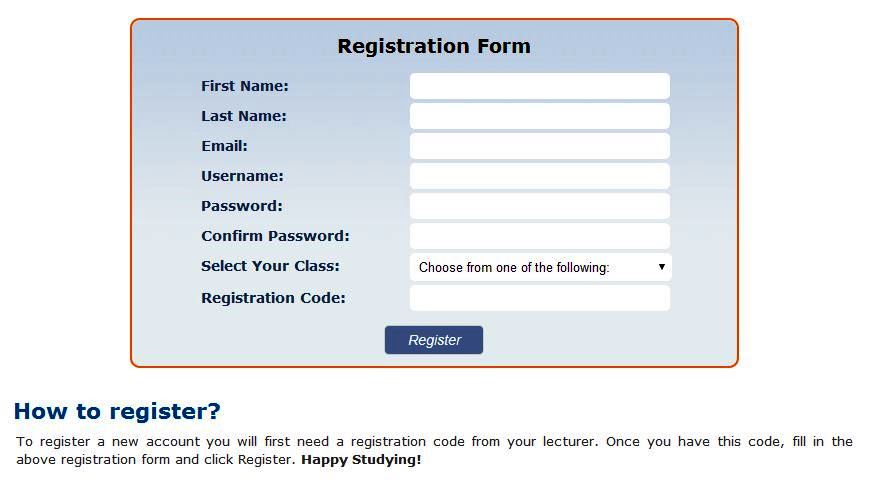
Click here to create a **new account**

Enter your username and password here and click **Log in**

**Creating a New Account**

In order to register for this website you will first need a class name and registration code which will be provided to you by your lecturer. Once you have these details, follow the below steps to create an account.

1. Fill in your details in the text boxes provided
2. Usernames must be at least 6 characters and can only contain letters and numbers
3. Passwords must be at least 6 characters and also must contain one uppercase letter, one lowercase letter and one number, e.g. Password1
4. Click the Register button to complete your registration; if successful you will be redirected to the Home Page
5. If your registration is not successful and error message will be displayed with further instructions



Select your **Class** from this dropdown list

Enter the **Registration Code** provided by your lecturer

Click the **Register button** to complete your registration; if successful you will be redirected to the **Home Page**

**The Home Page**

This is the page you will see when you first log into the website andcontains information and picture links to each of the main website features.

Logout Button

Username of student currently logged in

**Site Logo** –click here from any page to return to the home page



Picture Links to the main website features

Menu Bar

**Site Navigation**

There are two ways to navigate around the website. You can click the image links contained on the **Home** **page** or you can use the **menu bar**. Clicking the *StudyBetterTogether* logo (top left of website) from any page will take you back to the Home page.

***Menu Bar***



The menu bar contains seven links (Level 1 links) which can be accessed by all users. When you bring the mouse over some of these links a sub menu may appear underneath (Level 2 links).

Table 1.1 briefly outlines the functionality provided on each of the web pages linked to from the menu bar. A more detailed description including screenshots is contained throughout the remainder of this user manual.

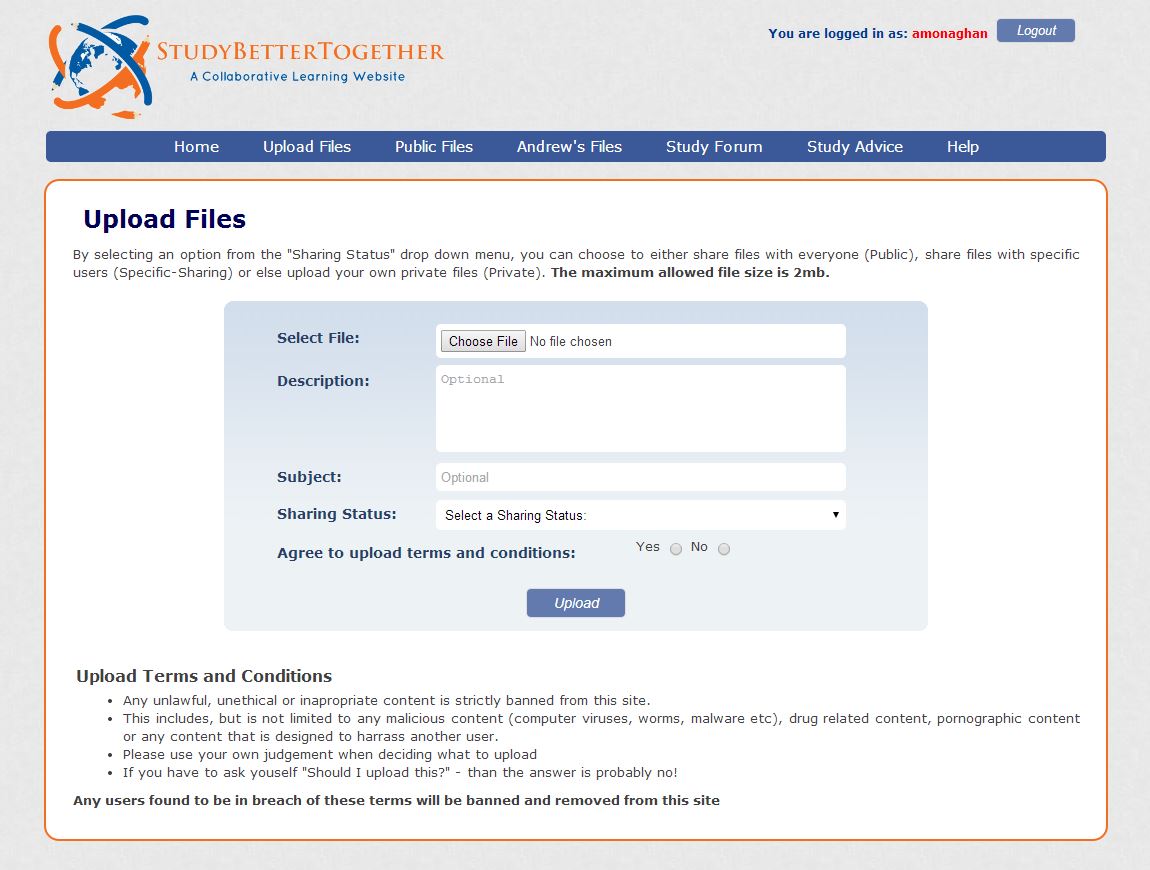
***Table 1.1 – Brief description of website pages***

|  |  |  |
| --- | --- | --- |
| Level 1 Links | Level 2 Links | Description |
| Home |  | Contains a description and picture link to each of the main website features. |
| Upload Files |  | Upload files and choose to either make them private, public or else share files with specific students. |
|  | **Acceptable Files** | Contains a list of all the file types that are allowed to be uploaded. |
| Public Files |  | Search for (and download) all public files on the website. |
| Andrew’s Files |  | Search for (and download) all of your own files on the website. These are files that either you have uploaded yourself or else files that have been shared with you by another student. |
| Study Forum |  | Create new conversation topics or search through and reply to previously created topics. |
|  | **Chat Room** | Log in and chat with any students who are also logged into the chat room |
| Study Advice |  | Read through an excellent study document containing advice and tips on a variety of study related topics |
|  | **Useful Resources** | Contains a list of external links to recommended study resources found online |
| Help |  | Contains this user manual |
|  | **List of users** | View the list of all students registered for this website. This can be used for matching usernames to students for the purpose of specifically sharing files. |
|  | **Change Password** | Change your login password |

**Uploading Files**

To upload files simply follow these 6 steps.

1. Click the **Choose File** button and select a file on your computer you want to upload.
2. Enter a **Description** of the file if you wish. This step is optional.
3. Enter a **Subject** for the file if you wish. This step is optional.
4. Choose the desired **Sharing Status** for your file.
   * **Public** – File can be searched for and downloaded by all users
   * **Private** – File can only be searched for and downloaded by yourself
   * **Specific User(s)** – You can select the specific users (students) that you want to be able to search for and download your file
5. Click **Yes** to confirm you agree to the upload terms and conditions which are listed underneath the upload form.
6. Click the **Upload** button



**1.**

**2.**

**3.**

**4.**

**5.**

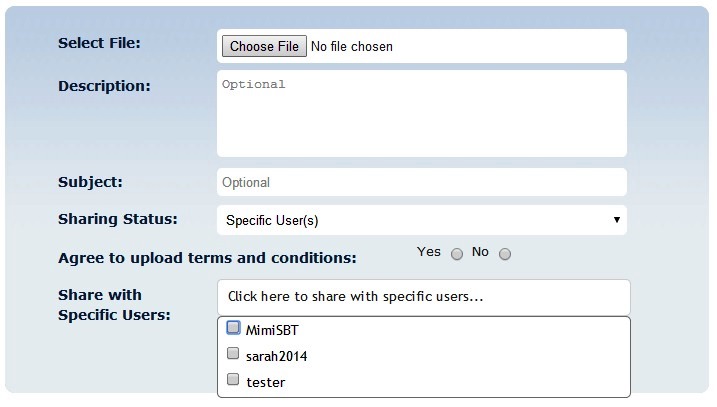
**6.**

***Choosing a Sharing Status***



You will need to select a **Sharing Status** for your file.

***Specific User Sharing***



If you choose **Specific User(s)** as your Sharing Status – a new dropdown menu will appear from which you can select the users you want to share your file with.

**Note:** When you move the mouse over the *Upload Files* link in the menu bar, a sub menu link will appear called *Acceptable Files.* This page contains a list of all the file types that are allowed to be uploaded to the website.

***Sub Menu – Acceptable Files***



**Searching and Downloading Public Files**

To search for and download public files simply enter your search criteria in the provided boxes and click search. The results will be displayed below the search form. To display a list of all the public files available, click search without entering any search criteria.

1. Enter in any **search criteria** you wish to search for.



**2)** Click **Search**



The **results** of your search will appear below the search form

**Search criteria** **entered**

Click this button to **Download** the file

**Searching and Downloading Your Files**

Here you can search for and download all of your own files on the website. These are files that either you have uploaded yourself or else files that have been shared with you by another student.

1. Enter in any **search criteria** you wish to search for



**2)** Click **Search**

***Sharing Status - Search Options***

You can choose from one of the following five options under the dropdown menu – Sharing Status

1. **All My Files** – This will return all files where you are the file owner (i.e. files you have uploaded).
2. **My Public Files** – This will return all public files you have uploaded.
3. **My Private Files** – This will return all private files you have uploaded.
4. **My Specifically Shared Files** - This will return all specifically shared files that you have uploaded.
5. **Files Shared With Me** – This will return all files that have been specifically shared with you by another user. Since another user is the file owner you will not be given the option to delete these files.



***Search Results for “My Files”***

The **results** of your search will appear below the search form.



Click this button to **Download** the file

**Search criteria** **entered**

Click this button to **Delete** the file.

***Different Results Table for “Files Shared with Me”***

When you choose the search status “Files Shared with Me” you will be presented with a slightly different results table. The two main changes are:

1. The username of the file owner is displayed
2. The usernames of whom the file is shared with is displayed
3. There will be no delete button as you are not the owner of the file



File Owner

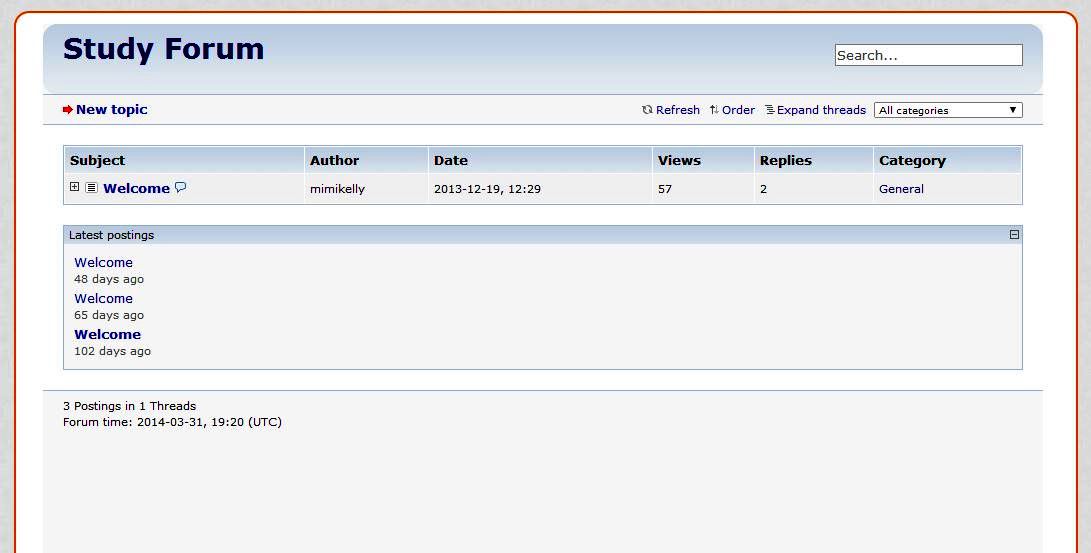
Usernames of students whom the files have been shared with

**Note:** No delete button as you are not the file owner

**Using the Study Forum**

When using the study forum you have several options depending on what you want to do.

1. **Create a new conversation topic**
2. **Search through previously created topics**
3. **Reply to a previously created topic**
4. **Filter conversation topics by category**



**4.**

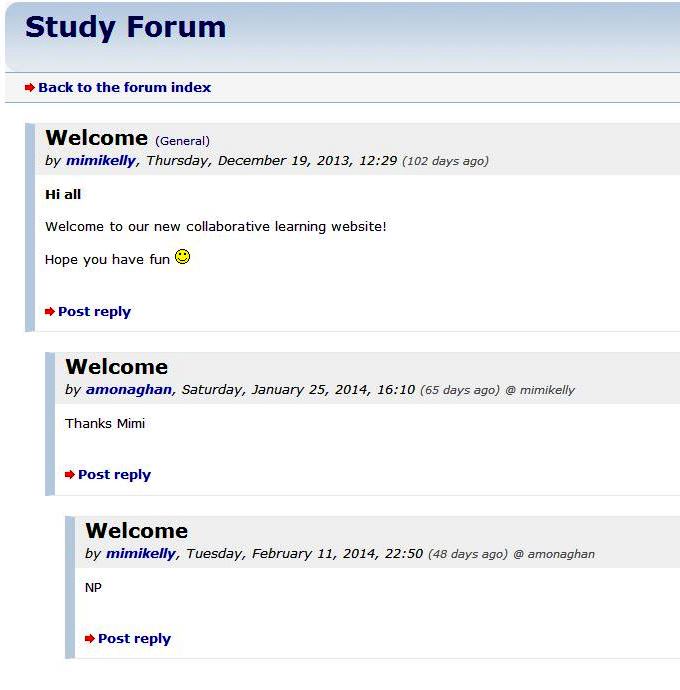
**2.**

**3.**

**1.**

Clicking on a topic heading will expand the topic allowing you to

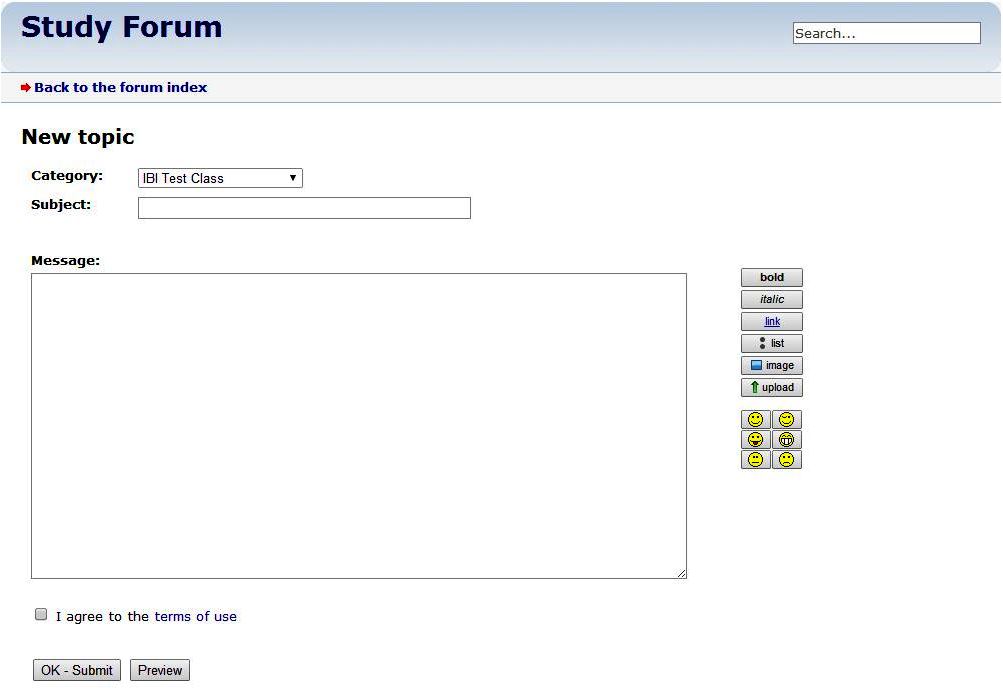
1. View the conversation history
2. Reply to specific posts



***Creating a new Conversation Topic***

Clicking the **New Topic** link on the main Study Forum screen will open a new window. To create a New Topic simply follow the below 5 steps.

1. Choose a category from the dropdown list
2. Enter a subject
3. Enter your message
4. Click the button to agree to the terms of use
5. Click the Submit button to complete your new topic post.



Use these formatting options to enhance the design of your message

**5.**

**4.**

**3.**

**1.**

**2.**

Clicking this **Preview** button will allow you to preview your new topic post before you submit it

**Using the Chat Room**

Moving the mouse over the **Study Forum** link in the menu bar will make a sub link appear called **Chat Room**.



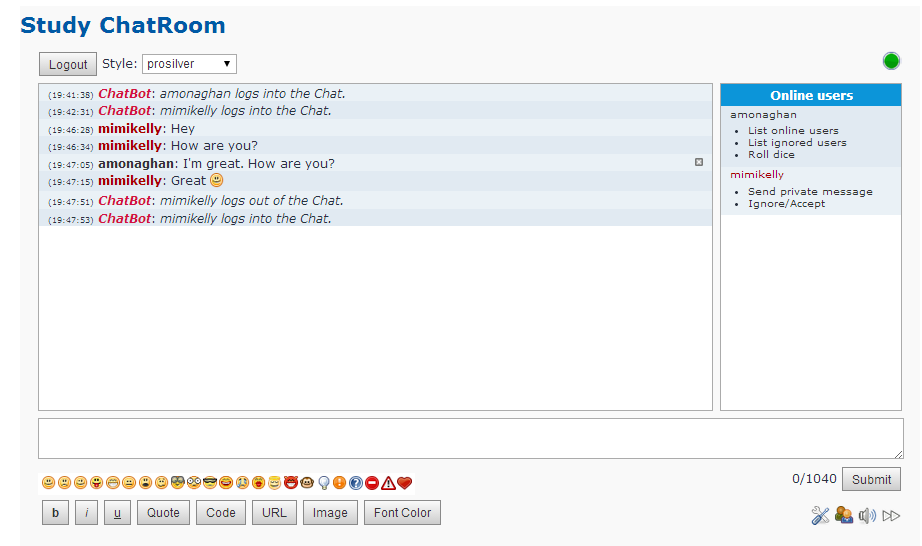
Submenu link appearing on mouse hover

1. After clicking this link, the first thing you will need to do is to log into the chat room. To do this, simply click the **Login button**.



1. Once you are logged in you will see the below screen. Here you can chat with other students who are also logged into the chat room.

List of other users who are also logged in to the chat room

1. If you wish to logout of the chat room, simply click the **Logout button**.

Type your Message here and either and click Submit

Previous messages will appear here

You can delete your own messages by clicking this button

Advanced options for sending private messages

Use these formatting options to enhance the design of your message

**Study Pages**

When you move the mouse over the **Study Advice** link in the menu bar a sub link will appear called **Useful Resources**. These pages make up the Study Advice section and contain the following information.

* **Study Advice** – On this page you can read through an excellent study document containing advice and tips on a variety of study related topics
* **Useful Resources** – This page contains a list of external links to recommended study resources found online



Submenu link appearing on mouse hover

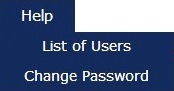


Name and link to various types of study resources

**Help Pages**

When you move the mouse over the **Help** link in the menu bar two sub links will appear called **List of Users** and **Change Password**. These pages make up the Help section and contain the following information.

* **Help** – Contains this user manual
* **List of Users** – Here you can view a list of other students registered for this website. This is useful for matching a username to the relevant students name for the purpose of specifically sharing files.
* **Change Password** – Here you can change your login password



Enter your new password twice and click **Confirm**

Submenu links appearing on mouse hover

****

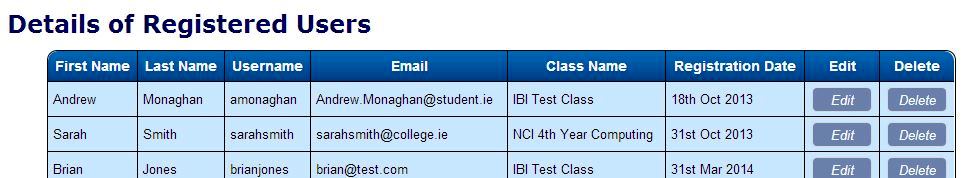
**Admin Use Only – Manage Users**

There are a number of features which are only accessible by the administrator of this website. These features include editing student details, creating new classes, creating a class registration code and configuration options for the Study Forum.

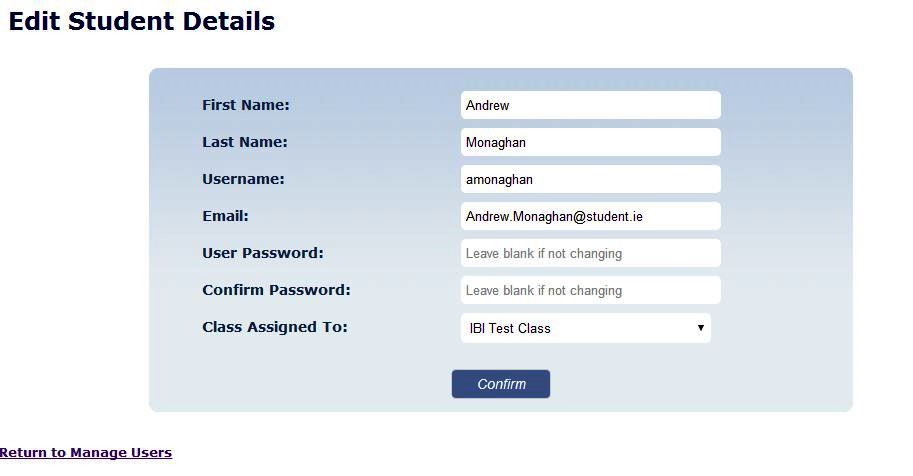
Only visible when the **Administrator** is logged in

Once the administrator logs in, a new option will appear in the Menu bar called **Manage Users**. Clicking this link will bring up a list of all users current registered on the website and provide you with two options:

* **Edit** –allows you to update student details, reset their password or assign them to a new class
* **Delete** –deletes a student and all their files from the website



Clicking the **Edit button** beside a Student will open up the **Edit Student Details** page



Make any necessary changes to the student’s details and then click **Confirm**

**Admin Use Only – Manage Classes**

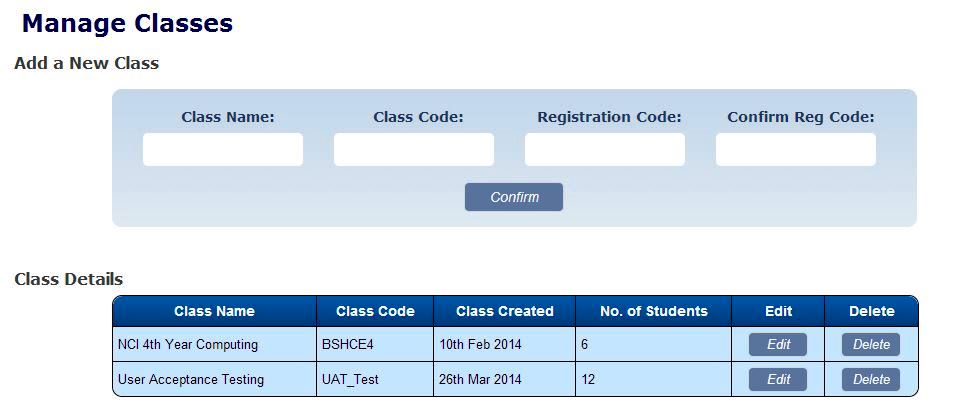
Assuming the administrator is logged in; moving the mouse over the **Manage Users** link in the menu bar will make a sub link appear called **Manage Classes**.



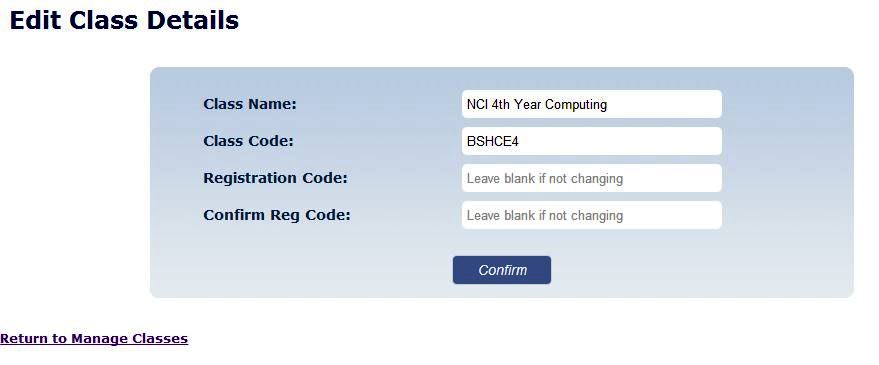
The Manage Classes page provides the following functionality:

* **Add a New Class** –Enter a class name, class code and registration code and then click confirm. This new class will be shown in the registration page and also as a category in the Study Forum.
* **Delete a Class** –Clicking the delete button beside a class will delete that class. Note: You can only delete a class if no students are currently assigned to that class.
* **Update Class Details** –Clicking the edit button beside a class will bring up the Edit Class Details page where you can change the class name, class code and registration code.

Add a **New Class** by entering in the Class details and clicking Confirm



Clicking the **Edit button** beside a Class will open up the **Edit Class Details** page



If you don’t want to change the **Registration code,** simply leave these boxes blank

Here you can change the Class Name, Class Code or create a new Registration Code

**Admin Use Only – Forum Settings**

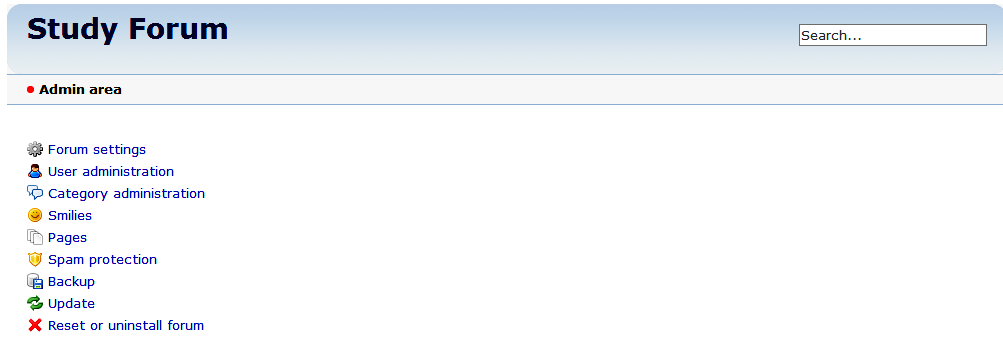
There is one other menu link that is only available if the administrator is logged in and that is the **Forum Admin** page. This is a submenu under the main Study Forum link in the menu bar as per the screenshot below.



Only visible when the **Administrator** is logged in

The Forum Admin page contains various settings for configuring the Study Form including:

* **Forum Settings** – this is the main forum configuration page and contains options for changing many aspects of the forums functionality and appearance.
* **User Administration** – this page allows you to view the forum activity of each user and also lock the account of users.
* **Category Administration** – this page allows you to view and edit all categories.
* **Smilies** – this folder allows you to edit and delete smilies used by forum users when writing posts.



Multiple **administration options** for configuring the Study Forum