

Interim Report
University of Leicester
Computing Bsc

Jack Charlesworth
Multiple Choice Question and Answer
Web Application

Table of Contents

Aims and Objectives	3
Planning and Timescales	3
Description of Prototype.....	3
Use Case Diagram	3
Use Case Tables.....	4
Aspects of the prototype to be extended in the final system.....	5
Parts to be added to those in the prototype.....	5
Play Quiz Functionality	5
Create Quiz functionality	7
Class Diagram	8
Software Architecture, Algorithms and DataStructures	9
Tier Architecture Diagram.....	10
Database Design	10
Storing users marks during the quiz	11
Traversing through questions of a quiz	11
Bibliography.....	12
C03015/C03016/C03120 Career Plan.....	13
1. Where do I want to go after graduation?	13
2. What will I do this academic year to get there?	13
3. How does my project contribute to my career?.....	14

DECLARATION

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s).

Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s).

I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Jack Charlesworth

Signed:

Date: 11/12/2014

Aims and Objectives

The aims and objectives outlined in the original plan are still very much relevant, in that the system will allow teachers to be able to create customisable quizzes for their students to sit and receive immediate feedback from.

Therefore there no changes from the original, many of the objectives are currently being worked upon for example, the research of existing web applications was done early on in order for me to gather ideas on how my application can stand out in the marketplace.

There are however a few additional objectives:

- Research the use of JavaScript and AJAX, how it can be utilized for a multiple choice web app
- Research use of data sanitization and validation for user login and register

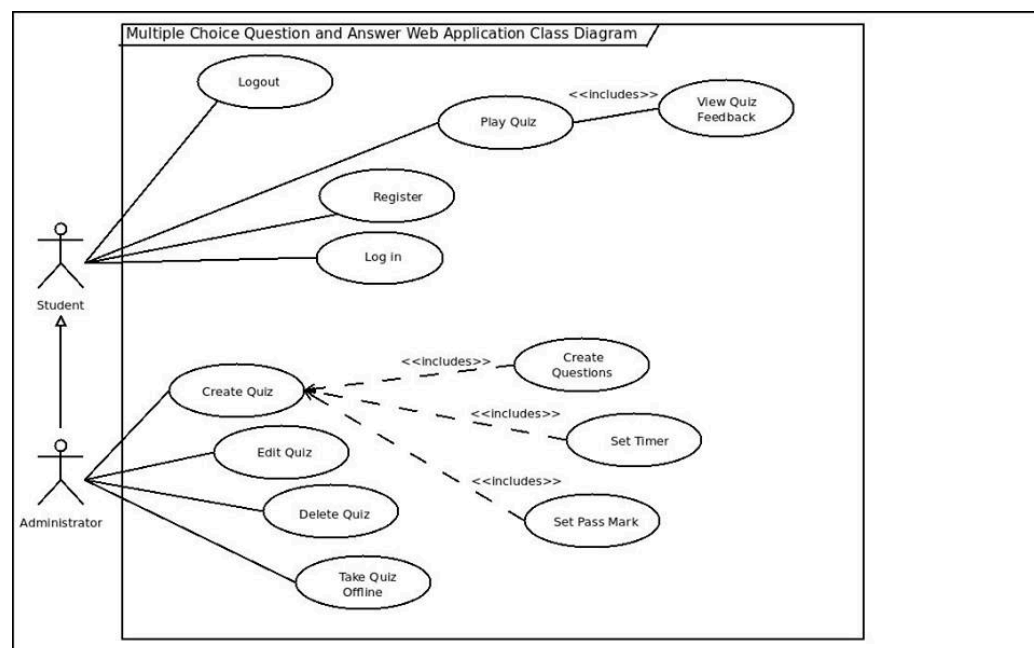
Planning and Timescales

According to the semester one Gantt chart, I had planned to test the web application across multiple web browser platforms however this has not been done and will be pushed back however this was due to higher priority tasks being undertaken. Having said this, progress has been ahead of what I expected at this point, core functionalities being on the way to completion.

Description of Prototype

The software system is written in PHP, although also making use of HTML, JavaScript and MySQL, built upon a three-tier software architecture of presentation HTML, application logic PHP/JavaScript and a database.

Use Case Diagram



The use case diagram above shows the user interaction with the system depending on their user type group of student or administrator. A student inherits some of the functionality from an administrator whereas an administrator can perform all actions a student can, plus more. Before a user can perform any of the actions, they must log in and they can only log in once they have successfully registered.

The nature of this project requires a lot of data entry and retrieval; therefore nearly every use case requires database interaction.

Use Case Tables

Use case name	Create Quiz
Participating Actor	Administrator
Entry Conditions	The user must be logged in as an administrator
Exit Conditions	The quiz is saved in the database and the account of the administrator
Flow of Events	<ul style="list-style-type: none"> - The user clicks on 'create quiz' to be found on the home page after successfully logging in - The user fills in following mandatory fields: <ul style="list-style-type: none"> - Question description - Four possible answers - Selects the radio button corresponding to the appropriate answer that is correct - Once completed the user clicks 'add to quiz' - The question is entered and stored in the database - The user then logs off

Use case name	Play Quiz
Participating Actor	Student, Administrator
Entry Conditions	The user must be logged in
Exit Conditions	The user completes the quiz, gaining a score which is saved into the database under the users account
Flow of Events	<ul style="list-style-type: none"> - The user clicks on 'play quiz' to be found on the homepage once successfully logged in - The system retrieves from the database the corresponding questions and answers for the first question, displaying them on screen to the user - The time starts counting down - The user answers a question by selecting one of the four radio buttons, they then click 'submit' - The system stores whether they got the answer correct or not in the answer array] - Making use of AJAX the next question loads onto the page along with the answers and the timer restarts - If a user fails to answer a question within the time limit they will received 0 for that question and be moved onto the next question - Once at the end of the quiz, the user is shown their scorecard containing their score and pass mark - The users score is recorded and stored in the database, associated with their user account

A GUI with large buttons and text has been used, throughout the website every day English has been used in order for students and administrators to be able to use the system without assistance. Currently the interfaces are built using Bootstrap [1] as I did not give GUIs a high priority in early implementation as I focused on functionality. Although these interfaces work well and create a professional looking system I hope to design from scratch the interfaces during the next phase of development.

Aspects of the prototype to be extended in the final system

- Currently questions are created but do not belong to a uniquely identifiable quiz, therefore an extension in the final system will be for administrators to create a number of quizzes containing their specified questions, each quiz will be identifiable by quizID and quizName.
- This in turn allows students to select a quiz to play from a list of available quizzes
- Currently all questions are worth 1 mark, in the final system this feature will be extended to allow admins to set a point weighting for each question in their quiz.

Parts to be added to those in the prototype

- When playing a quiz, users will be able to go back to questions they are unsure on, saving their answer to each question as they go but then at the end of the quiz they are able to do a 'final submission' overwriting past saves.
- Administrators able to set a pass mark for their quiz
- Users able to click on their profile to see their past quiz scores and attempts
- The ability for administrators to share their quiz on social networks by use of a unique URL
- Administrators having the ability to share their quiz on social networks by use of a unique URL
- Currently questions are played in order of question_id, administrators should be able to specify the order in which questions are played

Play Quiz Functionality

One of the core functions is the ability to play through a quiz, starting from the first question in the quiz, questions are obtained from the database through SQL queries and displayed question by question by making use of JavaScript, once a question is submitted by the user the questions reload by AJAX code as opposed to reloading the whole page.

When playing through the quiz, questions are played through in order of their 'question_id' from the question table, the four corresponding answers are collected from the answers table and displayed in a random order next to their appropriate radio button. A timer function has been included, implemented in JavaScript, timer counts down in seconds starting from the time that question was loaded.

The timer displays to the user in real time how many seconds left they have for that question, once the timer reaches 0 the submit button for that question is hidden and a message appears notifying the user they must continue onto the next question and they will not receive any marks for the previous question.

Once there are no more questions left in the quiz, the user is notified they have successfully completed the quiz and are directed to view their scorecard. The scorecard displays to the user their percentage mark for the quiz as well as how many they got right. This function will be extended in the final delivery of the system, as users will be able to see which questions they went wrong in, the correct answer will be shown allowing them to learn from their mistakes.

The screenshot shows a web interface titled "Quiz Master". At the top, a dark header contains the title. Below the header, the main content area has a light gray background. It displays a countdown timer: "You have 17 seconds remaining." Below this is a question: "what colour is grass". Underneath the question are four radio button options: "red", "green", "blue", and "yellow". At the bottom of the options is a green "Submit" button. Blue arrows point from the text annotations on the right to the timer, the question, and the list of options.

Counter, starting from 20 counting down in seconds

Question, retrieved from the questions database

The four corresponding answers to the question displayed in a random order

The screenshot shows the same "Quiz Master" interface but in a completion state. The header remains. The main content area displays the message "You have completed the quiz!" followed by "Click below to see your scorecard." Below this text is a button labeled "Scorecard".

Once there are no more questions left in the quiz ($\$arrCount \geq \$numQuestions$). Where $arrCount$ is the number of questions answered and $numQuestions$ is the total number of questions in the quiz, this screen appears.

Once the 'scorecard' button is clicked, the users scorecard is displayed, currently showing only their percentage score. This feature will be extended in

the final application to show the right answers for the questions taken

Create Quiz functionality

The create a quiz functionality is currently not complete however a question can be typed out by the user, four answer text fields filled out and then a radio button selected to indicate which answer is correct, the question and answers are then submitted and stored in the database.

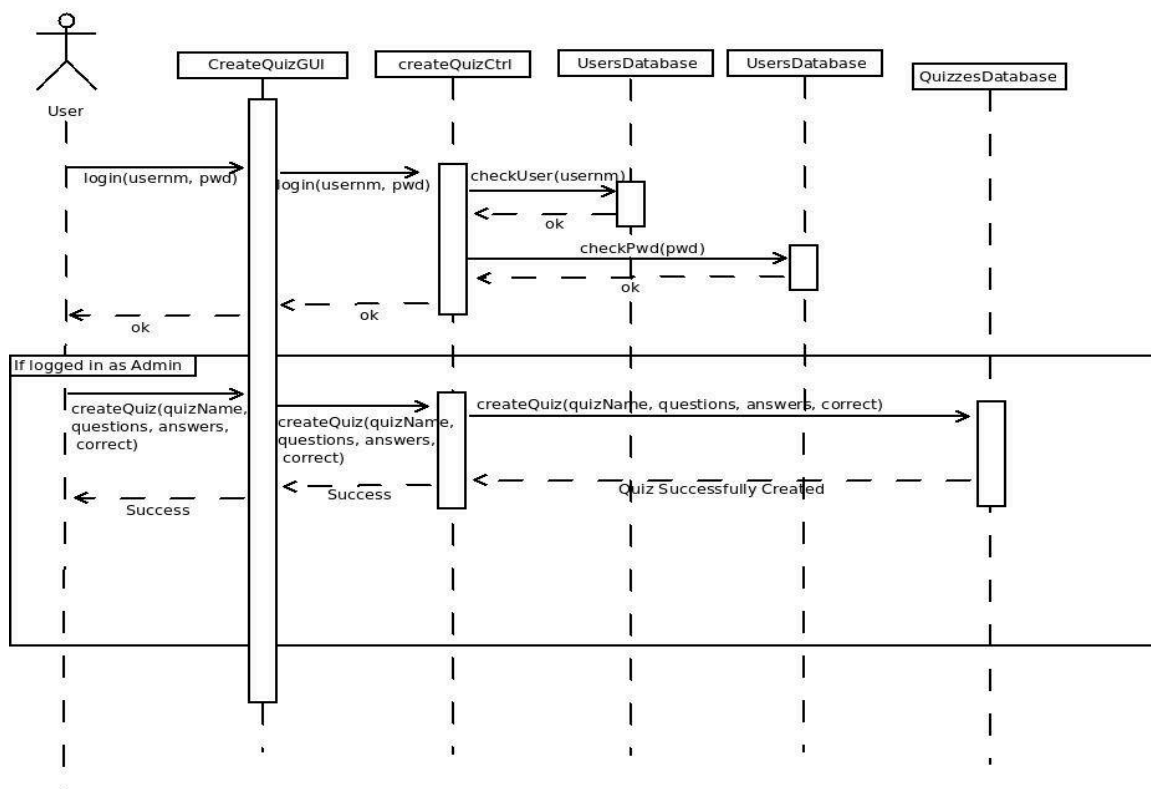
However, currently as user profiles and log in has not been set up questions are just stored in the database and not assigned to created quizzes, this is a feature that will be extended in the final system allowing an admin to create many quizzes as well as log in to edit and update their collection of quizzes, a quiz can only be edited by the writer.

The screenshot shows a web form titled "Create a Quiz" under the "Quiz Master" header. It includes a text area for a question, four text input fields for answers, and four radio buttons to select the correct answer. A green "Add To Quiz" button is at the bottom.

Text area for question

Text boxes for each of the four possible answers, one of which must be correct

Radio buttons, one radio button is selected corresponding to the correct answer



Above is a sequence diagram showing how processes within the system interact with each other during the execution of a task depicting objects and classes involved in the scenario and the sequence of messages exchanged.

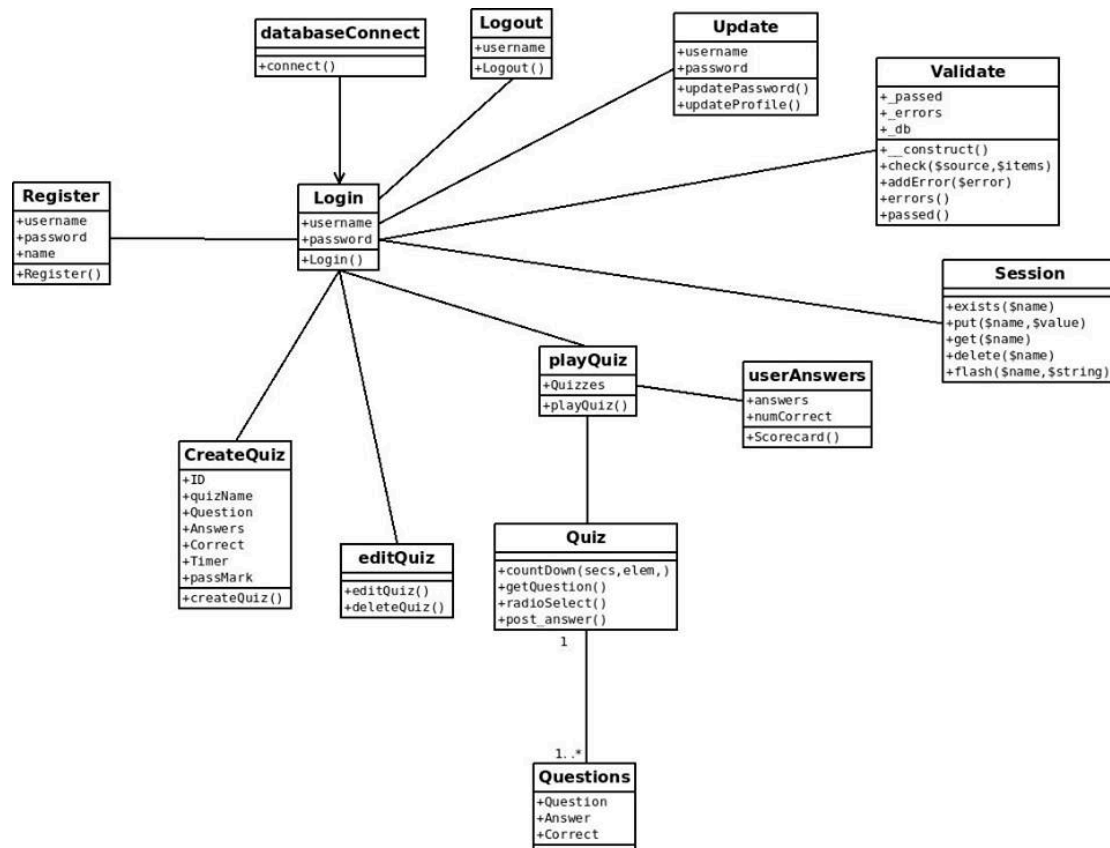
As shown above, before a user can create a quiz they must be successfully logged in as an administrator in order to perform this function. When creating a quiz, data is input by the user in the createQuizGUI and passed through the application layer createQuizControl to the persistence layer and stored in the appropriate database table. Once the quiz has been successfully created a message is sent through the application to the GUI notifying the user that the quiz has been successfully created.

Additionally in the final system, quizzes will be more customisable e.g. the quiz creator will be able to set a pass mark, indicating to students whether they have passed or failed as well as giving each question a weighting in terms of the points available they carry. Another feature that may be added on future releases is the ability for creators to set multiple correct answers for a question through use of tick boxes, meaning students may receive minus points for choosing a wrong answer.

Connection to the database is kept in one file “Databaseconnect.php”, this is then included onto every PHP script which requires database interaction. This makes database connectivity cleaner throughout the code as a full connection is not needed to be wrote in every file.

Class Diagram

Below is the class diagram depicting the relationship between classes in the multiple choice question and answer web app system.



Software Architecture, Algorithms and DataStructures

The multiple choice question and answer web app has been implemented making use of the three-tier software architecture style, splitting the application into presentation, application and persistence layer. This style was chosen for the web application due to the advantage of potential scalability should the web application need to deal with a larger number of users in future.

The presentation layer being the web browser the user views the web application on, rendering the HTML embedded into my PHP pages. The application layer is where my server side PHP scripts reside, interacting with the database currently running through local host and XAMPP.

The application layer, the play quiz script requests the appropriate questions and answers from the persistence layer databases and displays the appropriate quiz to the user through the presentation layer.

The persistence layer is the interaction with the database, inserting and updating tables for example when creating quizzes the PHP logic runs SQL queries to insert data into the questions and answer table.

When a quiz is played, a user interacts with the presentation layer through their browser displaying the HTML GUI, which runs through the PHP scripts to retrieve the appropriate questions and answers from the database by way of SQL statements. This resulting data is presented back to the user on screen.

One reason why I chose a three-tier architecture was due to security; if one tier in the system were to fail there would be no data because you are always secure by accessing the

other tier. Also presentation can easily be modified without affected the other two tiers, this is important for my web application as interfaces may be added or modified as times goes on. Separating the application into multiple layers makes it easier to implement re-usable components, in terms of the multiple choice web application a component in the application layer may be accessed by multiple components in the presentation layer. Performance is another factor I considered when choosing a suitable software architecture, with multiple users accessing the web application three-tier should be capable of handling multiple user requests at once.

Tier Architecture Diagram

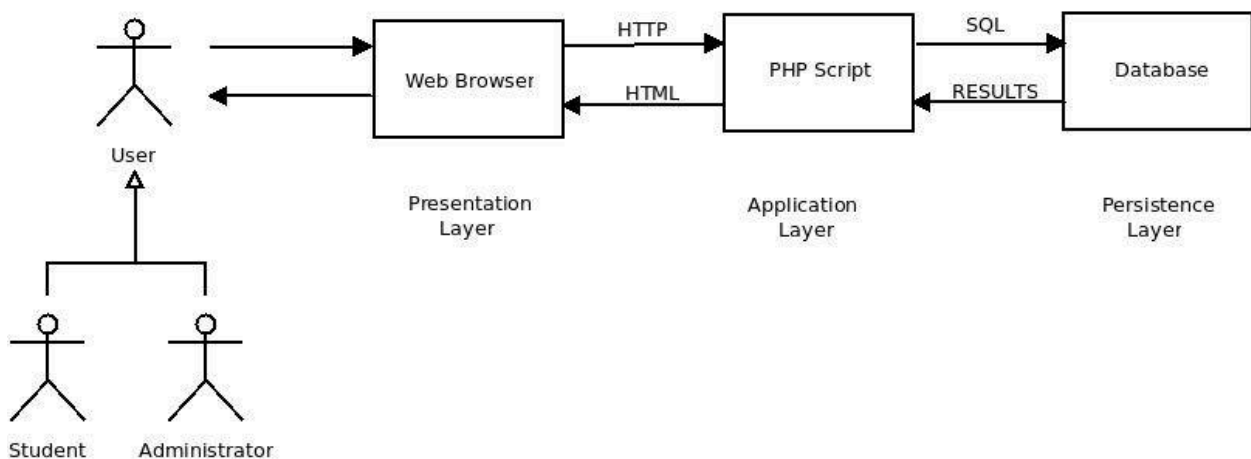


Figure [2]

Database Design

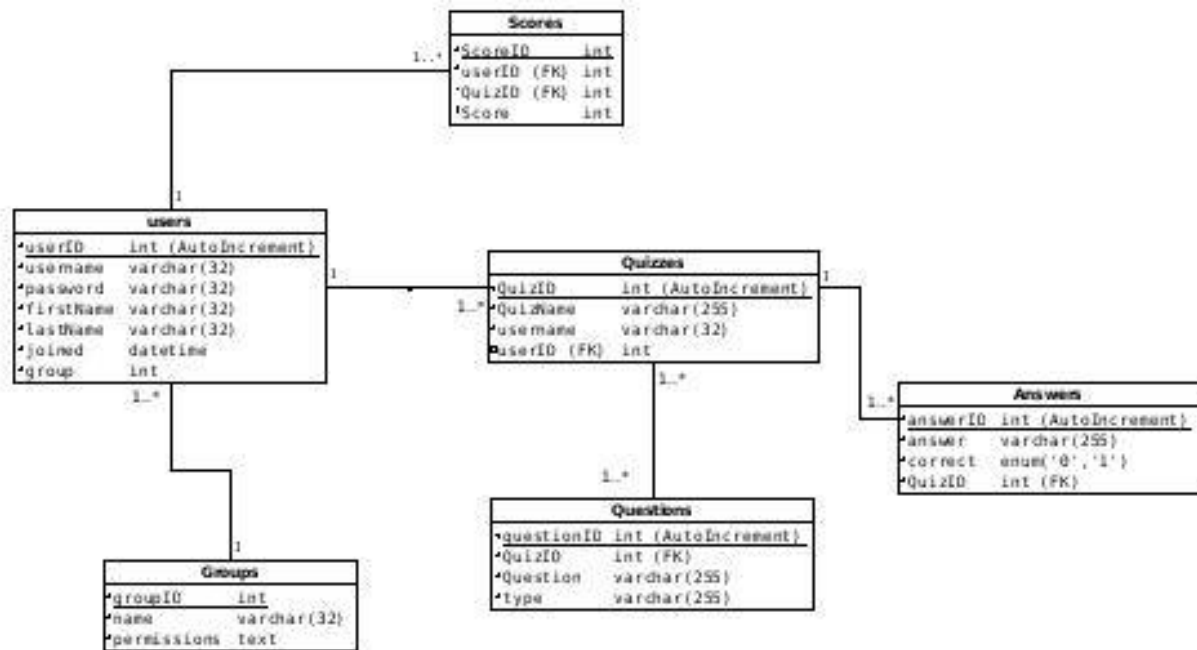
Below, is a database design diagram depicting the relationship between tables in the database that the PHP scripts interact with, inserting and reading data.

As shown, a user can log onto the web application and create a number of unique quizzes; this quiz can contain many questions. A user can be part of only one group signifying their privilege level of either student or administrator, whereby a group can contain a number of users.

Normalization has been adopted, the process by which data is organised efficiently to achieve the following goals:

- Elimination of redundancy
- Ensuring data is stored in the correct table and data dependencies make sense
- Eliminating the need for restructuring the database when data is added [3]

The database design follows 3rd normal form, in that functional dependency on non-key fields has been eliminated and all non-key fields are dependent only on the key.



The system makes use of a number of important data structures to organise and store data throughout the system.

Storing users marks during the quiz

Questions.php makes use of arrays, e.g. keeping track of the number of questions the user taking the quiz has got correct, the line “\$arrCount = count (\$_SESSION ['answer_array']);” holds an array of session variables, containing 0 or 1 whether the answer was correct or not, \$arrCount then stores however many elements are in that array and thus how many answers the user has got correct. This means the users quiz progress is updated question-by-question, gaining points as they go.

Also in Questions.php is where an array of questions is built from the questions table, by use of an SQL selecting questions where id is equal to that of the question number, “\$thisQuestion” takes the value of each question from the, “\$thisQuestion = \$row['question'];”. Similarly the same is done for the answers and the accompanying radio buttons “\$answers .= '<input type="radio" name="rads" value="'. \$correct. '">'. \$answer.'". [4]

Traversing through questions of a quiz

When playing through a quiz, Questions.php deals with the retrieving and storing of questions and answers in arrays, this data is then passed to Quiz.php that extracts the questions, answers and radio buttons one at a time displaying them in the quiz format.

The function “post_answer()” in Quiz.php makes use of JavaScript to post the answer submitted by the user for a particular question, feeding it through to the answers array. The script then brings up the next question in order of question_id on screen through the line “var url = 'quiz.php?question=<?php echo \$next; ?>';” whereby “\$next = \$question + 1;” and so on until the quiz has come to the end of its questions and “\$arrCount >= \$numQuestions”.

When a user logs into the system, whether they be a student or administrator a session will be created meaning their respective created quizzes can be retrieved from the database and

displayed on screen allowing the user to edit their quizzes. If they have completed any quizzes previously then their scores will be displayed on a dashboard interface.

Security is a factor, users will not be entering highly sensitive information into the system however they will be entering name, email and password. MD5 encryption will be used on the password making use of the md5() function in PHP. MD5 is an algorithm that is used to verify data integrity through the creation of a 128-bit message digest from data input that is claimed to be as unique to that specific data as a fingerprint is to the specific individual. [5]

Bibliography

- [1] <http://getbootstrap.com/> 2014
- [2] <https://alitarhini.wordpress.com/2011/01/22/concepts-of-three-tier-architecture/>, Ali Tarhini, 2014
- [3] http://www.databasedev.co.uk/database_normalization_process.html 2013
- [4] Isaac Price, <https://www.youtube.com/watch?v=u8mvNRBZGwg> 2014
- [5] Margaret Rouse, <http://searchsecurity.techtarget.com/definition/MD5> 2014

CO3015/CO3016/CO3120 Career Plan

1. Where do I want to go after graduation?

After graduation I hope to learn more about and go into web development, particularly front end focusing on user experience and design working as part of an effective team as the group project in the second year was a module I enjoyed. With regards to what type of company, I am still unsure, the opportunity to be part of a large corporate organization has its benefits in that a lot can be learnt during the initial time there particularly joining a graduate scheme.

In the past few weeks I have been applying to the technology graduate schemes offered by large companies based in and around London.

However joining a startup has always been something that I had liked to do, being part of a small close-knit team although many of the positions out there are understandably asking for 3+ years industry experience.

2. What will I do this academic year to get there?

Throughout the academic year, particularly the first semester I will prepare my CV and book an appointment with the careers service to refine my CV. Once prepared, I will begin applying for the graduate schemes on offer from the major companies in the technology industry, primarily those based in London and the south. The career plan presentation I attended in November gave me a number of websites I could use to start, GradDiary.com has so far been valuable in that I can apply to vacancies in the order of the soonest closing date. At the start of this academic year I attended the on campus careers festival, this gave me an insight into the possible areas I could venture into, many of which not being directly computer science related.

So far I have applied to a number of companies, I applied to the American Express technology graduate scheme whereby I just recently took their mathematics and English tests and had a phone interview where I am now waiting on their decision.

By the end of the Christmas break I hope to have applied to over 40 graduate schemes, preferably all technology related graduate schemes however if I feel the scheme is right for me I will not restrict myself and show employer I have learnt transferable skills that makes me a good fit.

3. How does my project contribute to my career?

I chose this particular project as I had an interest in web application development; this project has allowed me to learn more about aspects of development for the web, particularly languages that I had not studied in my time at university so far such as JavaScript.

When applying for jobs directly via email many ask for examples of my work, I feel once completed this project will be a piece of work that will reflect my ability and experience in the design and implementation of fairly complicated web applications.

While undertaking the project I hope to build upon my organizational and time management skills as well as that of being able to communicate effectively to someone what my project is about in simple terms.