



Knowledge Bomb

Interactive learning support with smart mobile technologies

Interim Group Report

Group 5

Henry Riches, Matthew Price, Luke Bandy, Ethan Preece, Alex Stewart, Kenneth Cynric
Medina Dasalla, Stanislav Borissov, Ahmad Syazani Mohammad, Nearchos Nearchou

Client: Professor David Marshall

Contents

Introduction.....	2
Software Development	3
User Interface Design	3
Connecting to the university LDAP server.....	4
Forum.....	6
Functional Dependencies.....	12
Test Cases	15
Use Cases.....	23
Justification for design and implementation	29
Strengths and Limitations.....	31
Design Strengths.....	31
Design Weaknesses	32
Implementation Strengths.....	33
Implementation Weaknesses	34
Legal, Social, Ethical and Professional Issues	35
Work Plan.....	38
Interim Deliverables.....	38
Interim Tasks	38
Future Deliverables	39
Future Tasks	39
Risks and Mitigations.....	40
Conclusion	41
References.....	42
Appendix A - Balsamiq Screenshots.....	43
Appendix B - Creating New Tables	48
Appendix C - Gantt Chart.....	50

Introduction

Since the time of writing the Initial Report, the team has made many steps towards completing the final product; a web application allowing students to ask and answer questions along with giving feedback to lecturers during lectures. For the last report we covered system requirements, scope and boundaries, as well as risks that could occur in terms of software development and social and legal issues. This time, we firstly designed roughly how the user interface should look and how we can test the website (using test cases), then we designed how the back end of the website should work using programming languages HTML, PHP, SQL*PLUS. With these in place we then began to create a prototype, demonstrating some basic but vital functionality. Along the way we have thought further about social and legal issues that can be implemented into our system, and decided what we need to deliver for the final report. This report will be a concise testimony of the work that we have done and the goals we have achieved during this part of the project along with further planning towards the final stage of the project.

We remained as a team of 9 members (names available on the front page of the report) where everyone contributed to the report and the project's development. Working as a group has enabled us to achieve a wide range of goals regarding our project in a short space of time along with building everyone's group skills to aid professional development as a student.

The report shows major aspects of software design along the frame work for our software implementation. At this stage we wanted to get the back bone/backend side of the program to working before we implemented any more difficult/additional features.

Software Development

User Interface Design

To help further visualize the representation of our ideas we decided on using a wireframe methodology to create a simple, yet fully functional version of our design for our program/applet. This gave us a better understanding of how the program might work in different circumstances along with allowing us to look further into positioning and size of buttons, text and input/output on screen. 'Wire-frames form the basis of creating websites and final software. They give you an idea about the space of the web page, positioning of elements and the navigation of the website' (Nishadha, 2012).

The wireframing tool we decided to use was 'Balsamiq', this is a rapid wireframing tool that emulates the creativeness and ease of using a whiteboard/sketching on paper but using a computer. We found Balsamiq to be easy to learn and use whilst still being an effective way of creating a black and white, minimalistic version of our system. Using wireframing allowed us to visually show the system's architecture, help improve ease of use and usability whilst making the design progress iterative. Using Balsamiq allowed us to collaborate easily and effectively as a group through their easy to use sharing system. We were able to share the project with everyone on the team so they could then reference this design in other areas of the project e.g. coding, further design etc.

The link to our Balsamiq project is:

<https://cardiffuniversity.mybalsamiq.com/projects/groupwork/grid>

Screenshots of this design work are also displayed in appendix A

Connecting to the university LDAP server

After talking with our lecturer he mentioned that we may be able to use the university LDAP server as a means of authenticating our users. Doing this provides us with a lot of benefits but there are a few drawbacks from using this approach as well.

The main benefit is that users no longer need to register to use this app, this is a great time saver as well as a great way to provide added security to the system. What we plan to do is the first time a student logs in with their university details we can encrypt their student ID as a MD5 hash and store that in the database. Every time they log in we can re-encrypt their student ID and use that to match them up to the existing MD5 value in the database. By doing this we can provide complete anonymity for our students, even if there was a database breach. The only way someone could figure out who someone else was would be by manually repeating this process and then comparing the values. We could even salt the hash for extra security.

By allowing the students to bypass registering it means that time in the lectures will be saved as students can get straight to using the app. Another benefit of this is that it is one less password for students to change, if they change their university password their app password will automatically change as well.

The biggest drawback for this comes with the white labelling aspect and allowing people to use this from places outside of Cardiff University. To circumvent this issue we would need to add a standard registration process. However for the purpose of testing this project internally we feel that this is the best approach and a standard registration could be added with ease at a later date.

Figure 1 below is a screenshot of the code that enables us to authenticate the users with the university LDAP server.

```
<?php
    if(isset($_POST['submit'])){
        $username = strtoupper($_POST['username']);
        $password = $_POST['password'];
        $ldap_server = "ldap.cs.cf.ac.uk";
        $ldap_binddn = "dc=cs,dc=cardiff.ac.uk";
        $connect = ldap_connect("$ldap_server");
        ldap_set_option($connect, LDAP_OPT_PROTOCOL_VERSION, 3);
        $r=ldap_bind($connect);

        $sr=ldap_search($connect, $ldap_binddn, "uid=$username", array("dn"));
        $info = ldap_first_entry($connect, $sr);
        $trydn = ldap_get_dn($connect, $info);

        $connect = ldap_connect("ldaps://" . $ldap_server);
        ldap_set_option($connect, LDAP_OPT_PROTOCOL_VERSION, 3);

        if( $bind = @ldap_bind($connect, $trydn, $password) ) {
            echo 'woo woo';
        }else{
            echo 'Noppppppe!';
        }
    }
?>

<html>
    <body>
        <form action="login.php" method="POST">
            <input type="text" name="username" id="username" />
            <input type="password" name="password" id="password" />
            <input type="submit" name="submit" value="submit" id="submit" />
        </form>
    </body>
</html>
```

Figure 1 - LDAP Server Connection

Forum

The Forum is a page in our website where the students can ask different questions about the lecture. This page also has a “like” button where the students can vote for their favourite question. The questions with the most votes go to the top of the list. One benefit of the Forum page is that the questions will be anonymously posted.

For this Forum page I’ve used the school’s server which can be found on “websites.cs.cf.ac.uk”, and the database to create the table and store the data is on

<https://www.cs.cf.ac.uk/phpMyAdmin/index.php>.

The Forum page can be found on the following link:

<https://project.cs.cf.ac.uk/NearchouN/Forum1/Forum.php>.

Figure 2 below shows how the front forum page will look:



Figure 2 - Forum Front Page

Figure 3 below is an example of what the forum questions will look like:

Question	Votes	Like
Can you show us the answers for the test?	9	Like
What is your teaching philosophy?	6	Like
How could we develop the Masters programme?	4	Like
Do your research on the institution and department specifically - how does the department fit into the whole? How popular are the courses? Think about how the take up could be improved.	3	Like

Figure 3 - Forum Questions

The code for the Forum page is split into 4 different files:

Forum.php

```
<!DOCTYPE html>
<html lang="en">
<link rel="stylesheet" type="text/css" href="Forum1.css" />
<head>
  <title>Forum</title>
</head>
<body>
<header>
<h1>Forum</h1>
<div class="boxed">
  Welcome to the Forum. Here you can ask your lecturer questions as the lecture is taken.
  Questions you would like to be answered can be voted using the like button
</div>
<script>
function validateForm() {
  var x = document.forms["myForm"]["fquestion"].value;
  if (x == null || x == "") {
    alert("Question must be filled out");
    return false;
  }
}
</script>

<div id="relative">
<form name="myForm" action="Forum1.php"
onsubmit="return validateForm()" method="post">
Question: <input type="text" name="fquestion">
<input type="submit" value="Submit">
</form>
</div>
<?php
require 'connect.php';
if (!$con){
  die("Connection failed: " . mysqli_connect_error());}
if (!$con){
  die("Connection failed: " . mysqli_connect_error());}
$query= "SELECT * FROM Forum1 ORDER BY Votes DESC";
$result=mysqli_query($con,$query);

echo "<table style='width:100%'>";

echo "<table>";
echo "<tr>";
echo "<th>". "Question". "</th>";
echo "<th>". "Votes". "</th>";
echo "<th>". "Like". "</th>";

echo "</tr>";

while($row = mysqli_fetch_assoc($result)){
  echo "<tr>";
  echo "<br/>";
  echo "<td>". $row["Question"]. "</td>";
  echo "<td>". $row["Votes"]. "</td>";
  echo "<td><a href='\"Forum2.php?id=' . $row[\"ID\"] . '\">Like</a></td>";
  echo "</tr>";}
echo "</table>";
mysqli_close($con);
?>

</body>
</html>
```

Figure 4 - Forum Main Page Code

Figure 4 above is the main file for the Forum page.

-Submit Button: When the user presses the submit button, code reference in figure 4, there is validation that checks if the textbox is empty. If the textbox is empty, then a message will be shown saying that the question must be filled out. If is not empty then it will go to the Forum1.php file where the question will be added to the database.

-Data Retrieving: This is the code to retrieve the data from the database and show them to the page and create the table, referenced in figure 5.

Forum1.php

```
</body>
</html>
<?php
$ID=rand(1,1000000);
require 'connect.php';
    if (!$con){
        die("Connection failed: " . mysqli_connect_error());}
$query= "INSERT INTO Forum1 (Question,ID) VALUES ('$_POST[fquestion]', $ID)";
    mysqli_query($con, $query) or die ("Could not insert rows because " .mysqli_error($con));
    mysqli_close($con);
    header('Location: ' . $_SERVER['HTTP_REFERER']);

?>
```

Figure 5 - Database Storage Code

Figure 5 above is the file where the question is stored in the database after the user press the "submit" button. It creates a random ID for each question that is not shown anywhere in the page but it will be later used for the Forum2.php file when the users will vote for a question.

Finally the array "header("Location: ' . \$_SERVER['HTTP_REFERER'];")"

will redirect the page to the initial page.

Forum2.php

```
</body>
</html>
<?php
$id=$_GET['id'];
require 'connect.php';
    if (!$con) {
        die("Failed to connect: " . mysqli_connect_error());
    }
    $query = "SELECT ID, Votes FROM Forum1 WHERE ID IN ($id,2,3)";
    $result = mysqli_query($con,$query);
    if(mysqli_num_rows($result) > 0) {
        while($row = mysqli_fetch_assoc($result)) {
            echo "<p>id: " . $row['ID'] . ", votes: " . $row['Votes'] . "</p>";
        }
    }
    else {
        echo "<p> Nothing here. </p>";
    }

    $update = "UPDATE Forum1 SET Votes = Votes + 1 WHERE ID=$id";
    if(mysqli_query($con, $update)) {
        echo "<p>Updated votes.</p>";
    }
    else {
        echo "<p>Something went wrong.</p>";
        die("Failed to connect: " . mysqli_connect_error());
    }
    mysqli_close($con);
    header('Location: ' . $_SERVER['HTTP_REFERER']);
?>
</body>
</html>
```

Figure 6 - Code for adding votes

Figure 6 above is the file that is been used for the votes. When the user press the “like” button it comes to this page and it adds 1 to the votes of that question.

The array “header('Location: ' . \$_SERVER['HTTP_REFERER']);” will redirect the page to the initial page.

connect.php

```
<?php
$con = mysqli_connect("csmysql.cs.cf.ac.uk", "group5_2015", "flicVol6","group5.2015");
?>
```

Figure 7 - Code for database connection

Figure 7 above is the file where the connection takes place

Functional Dependencies

The functional dependency diagram displayed in figure 8 presented below gives us a relational structure. This shows all attributes that we shall be using in our web application. For each attribute, we can consider which other attributes are dependent on it making it a key. With the help of the diagram we are able to use different normal forms. To reduce duplicates of attributes, we have decided to normalise the diagram below into its 3rd Normal Form (3NF). When it is in 3rd normal form, we are able to re-structure the diagram into different relations (or tables). When creating the functional diagrams, there is no definitive answer as different users may interpret data in different ways, however, we have listed below the assumptions that we have made to reach the solution we have.

The relations or tables implemented from the diagram, shown in figure 8, have given us a better look at what the database is like. We are able to create tables such as Student, Lecturer and Group containing relative attributes. Also the attributes that are underlined (bold letters for easier viewing) will be used as a key in the database structure. When the diagram is normalised, we are able to use the keys to as a relation between each table.

Using this structure of the database gives us a cleaner perspective of saving data. Third Normal Form (3NF) is most preferable normal form in RDBMS (relational database management systems). Normalization is the process of designing a data model to efficiently store data in a database' (Pinal, 2007). There is no mix up when querying and finding the correct attribute as each relation gives us a set of self-explanatory attributes. For example, when searching for a particular student's detail it is obvious to look at the table called Student and not the Lecturer table. This setup of the database is useful to us as if further modification of the database is needed then it only requires editing a table or even another table for the entire database. This means that we are able to edit the database while a lecture is in session without affecting the lecture itself.

Creating a simple table from the relations can be easily accomplished using SQL as a query language as seen in appendix B. Each individual tables are related to one another in some form.

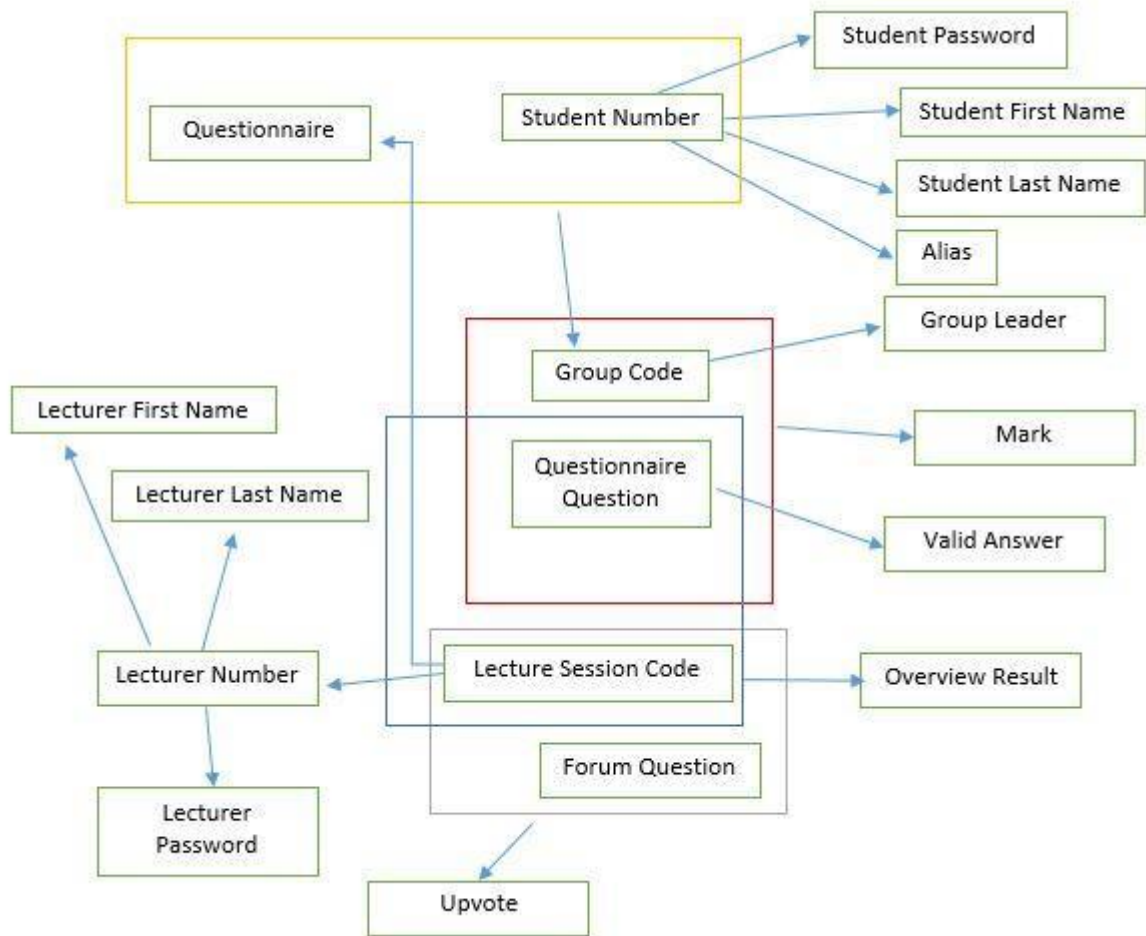


Figure 8 - Functional Dependency Diagram

Below are the SQL tables that we will be using for our system in 3NF form as displayed in figure 8 above which is our 3NF diagram:

Tables (or Relations):

Student(Student Number, Student First Name, Student Last Name, Student Password, Alias)

Group(Student Number, Questionnaire, Group Code)

Group Leader(Group Code, Group Leader)

QuestionnaireQA(Questionnaire Question, Valid Answer)

Group Mark(Group Code, Questionnaire Question, Mark)

Questionnaire Result(Questionnaire Question, Lecture Session Code, Overview Result)

Lecturer(Lecturer Number, Lecturer First Name, Lecturer Last Name, Lecturer Password)

Lecture Session(Lecture Session Code, Lecturer Number, Forum)

Forum(Lecture Session Code, Forum Question, Upvote)

We have also made assumptions regarding our system to help explain why we have made our tables this way:

Assumptions:

Each Lecture session only has one lecturer

Each Lecture session only has one set of questionnaires

Test Cases

The following is a comprehensive set of software test cases. The purpose of these test cases is to describe the appropriate system reactions to user actions. Each test case has:

- an ID associated with the functionality being tested
- a set of conditions that need to be met for the specific testing procedure to take place
- a step counter which identifies a given procedure within a test case
- procedure description - the way the user interacts with interface
- result - the system reaction to the user action
- pass/fail indicating whether or not the result in the test case reflects the actual implementation

At this point in the project we have completed part of our implementation and have conducted user testing on it, the rest of the test cases will be used when we have future development of the system. The following are the test cases for the features we have finished so far:

TS01: Login (student)			
Preconditions: None			
Step	Procedure	Result	Pass/Fail
1	Student inputs valid student number and password		
2	Login button is pressed	User is logged into the system, redirected to the user control panel and recognised as a student	Pass
Comments: Login verification successfully goes through the university LDAP server			

TS03: Login (negative test)			
Preconditions: None			
Step	Procedure	Result	Pass/Fail
1	Invalid credentials are submitted		
2	Login button is pressed	Access is denied and a message is shown indicating this	Pass
Comments: Login verification successfully goes through the university LDAP server			

TS11: Student adds question from the forum page			
Preconditions: Student is on the forum page			
Step	Procedure	Result	Pass/Fail
1	Student inputs questions in a text box		
2	"Submit question" button is pressed	Question is added to the forum board	Pass
Comments: The list of questions on the forum page is successfully updated			

TS12: Student up votes question on the forum			
Preconditions: Student is on the forum page			
Step	Procedure	Result	Pass/Fail
1	Student up votes a question by pressing a button next to it	Updated question score is calculated and kept, questions are displayed in descending order of score	Pass
Comments: The new score of the up voted question is calculated and the order of questions is changed if necessary			

Below are the rest of the cases, whose functionality has not yet been implemented:

TS02: Login (lecturer)			
Preconditions: None			
Step	Procedure	Result	Pass/Fail
1	Lecturer inputs valid id and password		
2	Login button is pressed	User is logged into the system, redirected to the user control panel and recognised as a lecturer	
Comments:			

TS04: Registration			
Preconditions: None			
Step	Procedure	Result	Pass/Fail
1	Student presses "Sign up" or "Forgotten password" button	Student is redirected to registration/password recovery page	
2	Student inputs student number		
3	"Send email" button is pressed	New temporary password is sent to student's university email	
Comments:			

TS05: Enter lecture room			
Preconditions: Student has logged in and is on the user control panel			
Step	Procedure	Result	Pass/Fail
1	Student enters valid lecture code in the "Join lecture" text field		
2	Student presses "Join lecture" button	Student joins a lecture room and is redirected to a page with the following options: "Submit question", "Create group", "Join group", "Solo join"	
Comments:			

TS06: Change personal information			
Preconditions: Student has logged in and is on the user control panel			
Step	Procedure	Result	Pass/Fail
1	Student presses "Change personal details" button	Student is redirected to details change page	
2	Student inputs new password in the appropriate text field and/or adds a nickname		
3	"Submit" button is pressed	If new password is set, an email is sent to the student's university email containing it. If a new nickname is selected, it is set to the user.	
Comments:			

TS07: Student submits question			
Preconditions: Student has logged in and joined a lecture			
Step	Procedure	Result	Pass/Fail
1	Student presses "Submit question" button	Student is redirected to question submission page	
2	Student inputs questions in a text box		
3	"Submit question" button is pressed	Question is added to the forum board and the student is redirected to it.	
Comments:			

TS08: Student creates group			
Preconditions: Student has joined a lecture and a questionnaire has been created			
Step	Procedure	Result	Pass/Fail
1	Student presses "Create group" button	A new group is generated and the student is redirected to the group room	
Comments:			

TS09: Student joins group			
Preconditions: Student has joined a lecture and a questionnaire has been created			
Step	Procedure	Result	Pass/Fail
1	Student presses "Join group" button	Student is shown a list of groups to choose from	
2	One of the groups is selected	Student is redirected to the group room	
Comments:			

TS10: Student solo join			
Preconditions: Student has joined a lecture and a questionnaire has been created			
Step	Procedure	Result	Pass/Fail
1	Student presses "Solo join" button	Student is redirected to a solo room and is automatically marked as ready	
Comments:			

TS13: Student removes own question			
Preconditions: Student is on the forum page and has posted at least one question			
Step	Procedure	Result	Pass/Fail
1	Student presses "Remove question" button next to one of the question they have posted	Question is removed from the forum board	
Comments:			

TS14: Lecturer views forum board			
Preconditions: Lecturer is on the user control panel			
Step	Procedure	Result	Pass/Fail
1	Lecturer presses "View forum" button	Lecturer is redirected to forum screen	
2	New question is submitted	The forum board is appropriately dynamically updated	
3	Question is up voted and order is change	The forum board is appropriately dynamically updated	
Comments:			

TS15: Lecturer prepares questionnaire			
Preconditions: Lecturer is on the user control panel			
Step	Procedure	Result	Pass/Fail
1	Lecturer presses "Prepare questionnaire button"	Lecturer is redirected to a question creation screen	
2	Input is given for question type, question title, question text and all possible answers		
3	"Add question button" is pressed	Question is successfully added to the questionnaire	
4	Steps 2 and 3 are repeated as many times as the lecturer wishes		
5	"Create questionnaire" button is pressed	"Questionnaire created!" message is given and group/solo join buttons become available for students.	
Comments:			

TS16: Questionnaire completion			
Preconditions: Lecturer has created a questionnaire and all groups are ready			
Step	Procedure	Result	Pass/Fail
1	Lecturer presses "Begin questionnaire button"	Lecturer is redirected to page which shows student answers and how many groups are left to answer the first question. Students are redirected to page showing the first question and possible answer.	
2	Student selects answer to question and selects "Ready"	Answer is saved and shown on the lecturer's device	
3	Lecturer presses "Next question" button when he feels that enough groups have submitted answers	Same as result of step 1 for the next question in the questionnaire. This step is repeated until the last question.	
4	End of questionnaire is reached and lecturer presses "Results" button	Lecturer is redirected to a page with the summary of the results, students are redirected to a page with their mark and their results on the questionnaire. Students are given the option to go to the forum, control panel or logout	
5	Students presses "Logout" button	Student is successfully logged out of the system	
Comments:			

Use Cases

Use cases are a great way to document how your system is going to work and the overall flow of the system from a user end view. They help reveal sometimes un-thought of process alternatives, process exceptions, undefined terms and outstanding issues. Using use cases as part of a design stage can help recognize patterns and contexts in functional requirements along with ensuring the delivered software works as it should. Figure 9 below we have created a use case diagram to visually display the use case requirements and gone through some key aspects of our program complete with basic flows, as well as alternative flows that could occur.

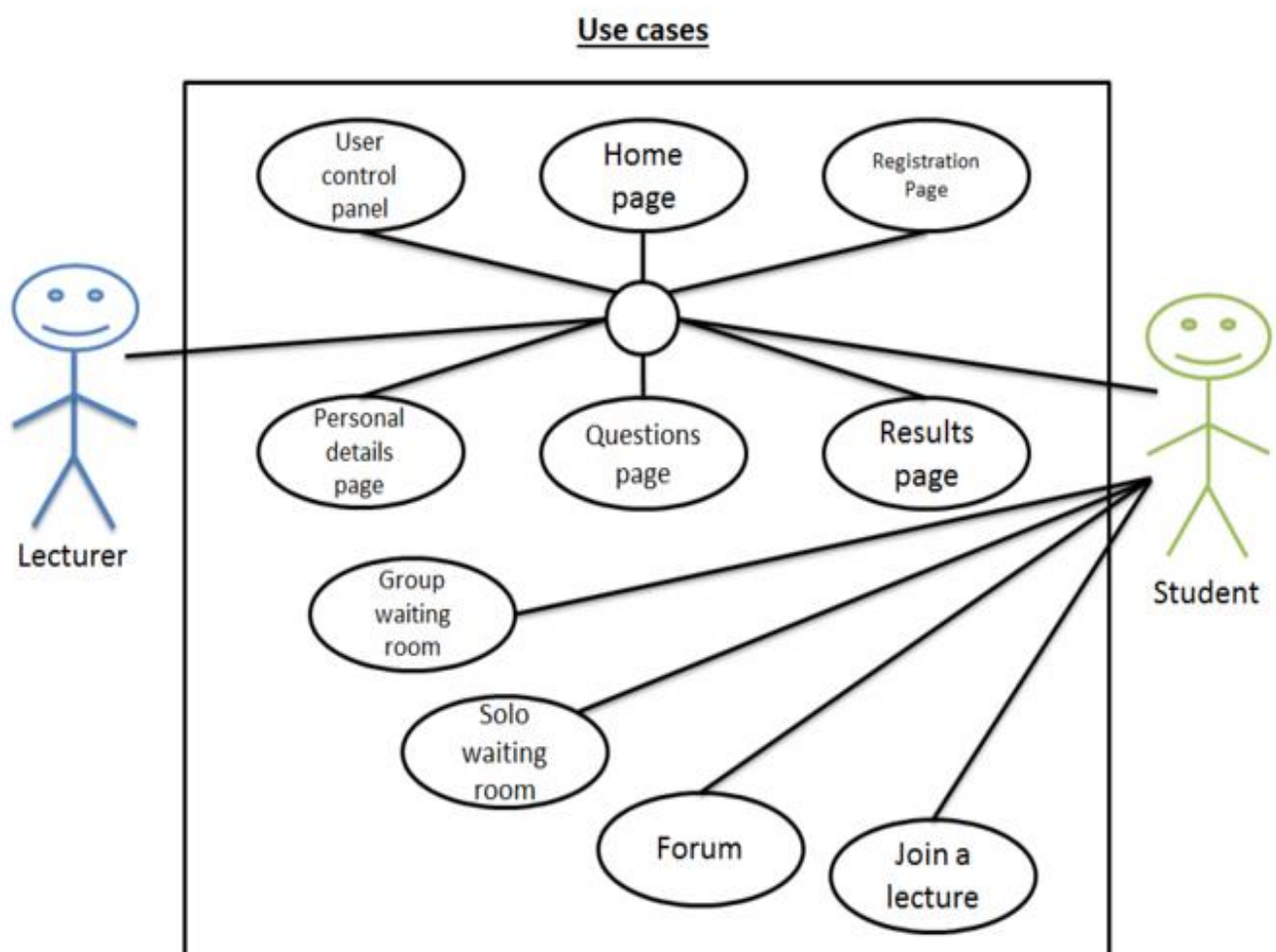


Figure 9 - Use Case Diagram

Use Case ID: 01

Use Case Name: Login Page

Pre-condition: The user has a university student login.

Post-condition: The user's account will work on the system correctly

Brief Description: The user will input their university student number and password into the username/password input boxes then press the 'login' button and be logged into the system, redirecting them to the student/lecturer control panel

Basic Flow:

- 1) The user inputs their correct student number into the username box
- 2) The user inputs their correct password into the password box.
- 3) The user then clicks the 'login' button
- 4) The user is then redirected to their corresponding control panel

Alternative Flows:

3A) The username given in the username input box is incorrect/not a valid university login number

3A1) The error message 'Please input valid university login details' is displayed

3B) The user's password that they have entered into the password input box is incorrect

3B1) The error message 'Please input valid university login details' is displayed

3C) At least one form field is left empty.

3C1) The error message 'Please input valid university login details' is displayed

4A) The user inputs the correct login details but the LDAP server is unresponsive

4A1) An error message is displayed saying 'Something went wrong, try again'

Use Case ID: 02

Use Case Name: Join a lecture

Pre-condition: The user is logged into the system as a student user

Post-condition: The user will now be in their specified lecture room where they can submit questions, create groups, join groups and answer a quiz solo.

Brief Description: The lecturer will have a randomly generated code from his version of the software that students will use as a passkey to join a lecture when prompted.

Basic Flow:

- 1) The user clicks the 'join lecture' button
- 2) The user enters the code given to them by their lecturer into the prompt box labelled 'Enter your unique lecture code'
- 3) The user inputs the correct code and clicks the 'join lecture' button
- 4) The user is redirected to the lecture waiting room.

Alternative Flows:

3A) The user inputs an incorrect code

3A1) The error message 'Please input the code given to you by your lecturer' is displayed

Use Case ID: 03

Use Case Name: Join a Group

Pre-condition: The user is logged in and is in a lecture waiting room

Post-condition: The user will be able to view who is in their group and the leader can answer questions on behalf of the group

Brief Description: The user will join a group through the join group button by inputting a randomly generated code that is given to the group leader when the group is created

Basic Flow:

- 1) The user clicks the 'join group' button
- 2) The user inputs the code given to them by their group leader into the prompt box
- 3) The user then clicks the 'join group' button
- 4) The user is then redirected to their group waiting room where they can see current members in their group

Alternative Flows:

3A) The user inputs an incorrect code and presses the 'join group' button

3A1) The error message 'Please enter the code given to you by your group leader' is displayed

3B) The user is already in a group and attempts to change to another

3B1) The error message 'You can only be in one group at a time, leave the other group before joining another' is displayed

3C) The 'input group code' field is empty and the 'join lecture' button is pressed.

3C1) The error message 'Please enter the code given to you by your group leader' is displayed.

Use Case ID: 04

Use Case Name: Quiz as a group

Pre-condition: The user is part of a group and is the group leader, user is in the group waiting room and a quiz has been set-up by the lecturer

Post-condition: The user will be in a current quiz and be able to answer questions on behalf of his/her group

Brief Description: The user will join the quiz set up by their lecturer and answer a question by selecting one of the answer boxes and go onto the next question until there are no more questions

Basic Flow:

- 1) The user clicks the join quiz button
- 2) The user is redirected into the quiz room where the first question along with answers is displayed
- 3) The user selects an answer using the answer boxes and clicks next question
- 4) Repeat step 3 till there are no more questions
- 5) The user clicks end quiz

Alternative Flows:

3A) The user clicks the 'join quiz' button but there is no quiz set up

3A1) The error message 'There are no current quizzes available' is displayed

3B) The user clicks the 'next question' button but has not chosen an answer

3B1) The error message 'Please choose an answer before going onto the next question' is displayed

Use Case ID: 05

Use Case Name: Change alias

Pre-condition: The user is logged into the system and is on their corresponding control panel

Post-condition: The user's alias/nickname will be displayed to what they changed it to

Brief Description: The user will access their personal details, change their alias/nickname to whatever they want and then click the 'submit' button to save changes made

Basic Flow:

- 1) The user clicks the 'Personal Details' button
- 2) The user is redirected to the personal Details page
- 3) The user inputs their desired alias name into the 'Change Alias' box
- 4) The user clicks the 'submit' button to save their alias

Alternative Flows:

3A) The user clicks submit without entering a valid alias

3A1) The page is redirected back to the control panel and the alias remains what it was before

3B) The user enters a forbidden word(s) as their alias

3B1) The error message 'Please input a valid alias name' is displayed

Use Case ID: 06

Use Case Name: Submit a question to the Lecturer

Pre-condition: The user is logged into the system and is on the forum page

Post-condition: The question submitted by the user will now be shown on the forum page

Brief Description: The user will submit a question to the lecturer from the forum page and it will be displayed under the 'active questions' feed on the forum page

Basic Flow:

- 1) The user writes a question into the 'question' input field.
- 2) The user clicks the submit button.
- 3) The user's question is then displayed in the 'active question' feed on the forum page

Alternative Flows:

3A) The user clicks submit with the 'question' field empty

3A1) The error message 'Please enter a valid question' is displayed

Use Case ID: 07

Use Case Name: Upvote a question

Pre-condition: The user is logged into the system and is on the forum page

Post-condition: The question the user has upvoted will have 1 more vote

Brief Description: The user will click the arrow next to the question to upvote it

Basic Flow:

- 1) The user clicks the arrow next to the question
- 2) The question gains one more vote and moves up the list corresponding to the amount of votes it has

Alternative Flows:

3A) The user clicks the upvote arrow for a question they have already upvoted

3A1) nothing happens as a question can only be up votes once per user

Justification for design and implementation

The application is coded in HTML and PHP which has a number of benefits. HTML is highly flexible and can be supported on almost every browser on all platforms which allows for the application to work on computers as well as portable devices such as mobile phones and tablets. HTML and PHP code is also easy to change and maintain. PHP can be used to connect, update and retrieve a database. We shall be using MySQL to store results of our forms as well as students and lecturer information.

We have included a colour-blind assist option so that the application can be used by people with inability to see colour or perceive colour differences. As part of the first design, we have implemented it as a button which allows the user to manually control the assist option by toggling on/off. The colour-blind assist is positioned at the top-right corner of every page so that it can be accessed wherever the user is on the application. We can see examples of the web application on our balsamiq designs. We have decided to keep the design in the report in order to show of our previous designs as well. To further improve the usability of the application, we have decided to make the assist option as a feature to be used in the personal details of the user. In preference of the assist, the student is able to choose if it is on or off at default. The user is then able to make changes by changing it on his personal details page.

In the forum, the questions are positioned at the centre of the page with the number of votes and the vote button beside it. The questions are also spaced from each other. This is so that each question can be seen clearly along with the number of votes it has. Help and clear instructions are provided at the top of each page that is necessary which tells users what each page is for and what the user can do on them. Students are able to use this forum as a chance to be able to ask questions for the lecturer. The most up voted question is then sent to the lecturer to which the lecturer can answer.

The answers to the questions asked by the lecturer are in multiple choice so it can give a rough idea of what the answer is. It can also test the students' understanding of the lecture as the answers to the questions can confuse the students. Each page only displays one question and there are buttons which can be used to see the next and previous questions. This is so that the page does not seem cramped and that it is organised.

We have carried forward with the ad-hoc grouping feature as it is one of the desired features wanted in our previous research. It shall be carried forward as a group task if desired by each user when dealing with the questionnaires set by the lecturers. The students (users) are able to choose to answer solo or within a group. For the design aspects of ad-hoc grouping, there will be a group leader who can choose when the group is ready. After the questionnaire is completed, the group is then shown where they are correct, the correct answer as well as their mark per question.

We have implemented our web application with a voting system that allow us to up vote the most 'worthy' question to be given to the lecturer. This feature allows not only for students to share their own thoughts with one another but also to be in agreement to a question to which they deem is relevant. The voting system will opportunities for students to up vote or down vote questions to be asked to the lecturer. This design aspect allows all the students to have their own thoughts on all questions sent in the forum. They are able to up vote or down vote all questions if they do desire.

The use of the university's LDAP server saves us time for the students to create an account and a new password. When implementing the LDAP login, we are ultimately within the university network. This saves time for both the development team and the lecture session. We are able to save database space as we do not need to create space for the new passwords of the students. It is relevant for the students (users) as they do not require to remember a new password. Even if it is the same password, if the student decides to change his/her university login detail, then the updated information can is applicable with our application.

We have considered white labelling as one of our added features in the application so that our application will be of commercial use. This means that it can then be used by other external users that require a similar use of the application. It can then be used by other universities or even entertainment system that requires multiple users and one main user (e.g. lecturer). White labelling also allows external users to change the theme and layout of the application to their own desires.

We have integrated a code to be used to access the lecture session given at the start of the lecture of the lecturer. This allows the target audience to be for students who are only in the current lecture. The lecture session within the application cannot be accessed without the code for lecture entry. This allows restrictions so that no other students are able to access the current lecture.

For a bigger range of target audience in terms of device use, we are making sure that our application is cross-platform. This means that students with different OS are able to handle and use the application. Our focus shall be on mobile OS such as iOS, Android and windows. As most students have a mobile device, then they are able to enter the lecture session. Cross-platform allows to have better compatibility and more participation as the students have a wider choice that allows them to use in terms of using the application.

Strengths and Limitations

Design Strengths

Using a Wireframe tool to outline our interface design allows us to easily create and present the general idea behind our website. Our Balsamiq project clearly displays and explains the functionality of the site and how the user will interact with it, and will serve as a guide and template when the group is implementing the majority of the functionality in the next stage of the project. The interface design itself covers all of the acceptance criteria set in the initial group report, with any changes we've made to this point being reflected appropriately. All of our website's functionality, as described in the 'User Interface Design' section is visualised. In making the interface we took into account ease of usability (e.g. avoiding having too much clutter on different pages, descriptive buttons and easy navigation) and consistency (site layout is simple, UI elements are where a typical user would expect them to be).

Test cases allow us to define what we're expecting out of our finished implementation in terms of system reactions to user actions. They give us a clear idea of what each user interaction should result in and the ability to document and analyse these results. Through a series of pass/fail tests in each test case we will be able to adequately judge how successful our implementation has been once it is finished.

A use case diagram provides an easy to understand way to describe what our system will do. It clearly displays who the users of the system will be and what actions they'll be able to perform. The use case diagram helped with laying the groundwork for more complex elements of design, by giving a less complicated way of describing our system. Our use cases provide us with the specific descriptions of system behaviours that occur when the user interacts with the system. These will serve to further clarify what a proper implementation would be like. Our design also includes a database design, which explains the relationships between various entities within our application, giving us a better idea of what the overall structure of our implementation should be.

Design Weaknesses

Whilst using wireframes as a design pattern is a good process in helping to visualize the system as a whole and the overall layout and flow of the system, it lacks any thought into the system aesthetics, for example the colour scheme we will implement, the pictures we will use and the overall look and feel of the app. Using wireframes also does not take into consideration technical implementations of the software and whether what is being displayed by the wireframe is actually achievable. As a whole wireframing should be done at the early stages of the design process to get user and client approval of the layout of key pages and navigation, but as we have done this quite late in our project development it has proved less useful as it was done simultaneously with the actual implementation of the system.

Use case diagrams are an effective way to show user requirements from the user's point of view however there are some risks involved in using them. Use cases do not address the usefulness and usability of the program, they should be coupled with the wireframes and other design work to fully showcase the motivations of the use case. There is also no systematic way to handle non-functional requirements with use cases. Although our systems use case diagram is relatively small as our system is planned to be quite compact and simple, use case diagrams for larger, more complex systems can become messy and cluttered, even if professional UML software is used.

Using an agile design model is great for the ad-hoc group that we have but there are some risks involved with using it. One of the risks/weaknesses of using agile is that in the case of the software deliverables, especially the larger ones, it is quite difficult to assess the effort required for these tasks at the beginning of the software development life cycle. Another issue that we encountered is there is a lack of necessary design and documentation for the project.

Implementation Strengths

LDAP

By authenticating users via the University LDAP server we are able to reduce the time that it takes for people to use the software by bypassing the registration process. We have also been able to increase the security of the software by not needing to store any of the user's passwords in our database. As they are all externally managed and since we are connecting via HTTPS any security breaches on our side would not result in compromising anyone's university account. Unless of course the file server got compromised and they changed the code of the underlying system.

Forum

The Forum Page can be a very useful tool for the students and the lecturers as well. It gives the ability to the students and lecturers to have high-quality discussions to express and see differing views and opinions for a specific topic. Also the use of the "like" button will help to up vote the best questions and show them on the top of the list. Additionally, the use of a Forum can help the lecturers and students to save a lot of time. If a student posts a question especially with a lot of detail, this information can be referenced in the future, so students will not have to ask the same questions. In general the Forum will increase the interest of the students for the lectures. Having a Forum can be an effective way of making students want to come back and use the app within lectures, especially if they need help or advice on a certain topic.

Implementation Weaknesses

LDAP

Since we have now switched to LDAP authentication we currently don't have any way for users from outside of the university to register or access the system. This is not a problem at this moment in time, however when we plan to white label the system we will need to come up with a universal registration process that can allow anyone to use the software. This can easily be achieved at a later date if it is required. another weakness of using the LDAP system is that if the university LDAP system were to fail/go down then we would not be able to use our application as we depend on this system to be operational for students to login to our application, however this is highly unlikely as the LDAP system is a university wide system with high levels of security.

Forum

The lack of posts onto the forum can have a negative impact on the app as if students find that the forum is not very useful and they find asking questions verbally is easier, then the forum will have been a waste of time. If the students see that there are very few posts on the Forum page then they will probably not want to post themselves and lose their interest for the app and unless told to do so by their lecture, not use the app regularly.. Another problem that can be caused from the use of a Forum is the inappropriate posts. Another issue from letting the students post themselves is that some rude students might use swear words/forbidden words in their posts, this can cause upheaval between the students making the forum less attractive and unprofessional, thus making it less likely for a lecturer/student to use our system.

Legal, Social, Ethical and Professional Issues

Legal issues

Once we started creating the application there were many Legal issues which we had to make sure we adhered to.

One of the main legal issues which we had to make sure we had followed was to ensure the protection and security of user's personal information. This also means that any kind of personal data should be processed fairly and lawfully.

To do this we have decided to run the application of the University's server. This was very effective because it enabled us to use each user's existing university login (This is the only kind of data which could be classed as sensitive data within our application at this time). This was very effective because it meant we did not have to get each user to create a new login, as they were able to use their own university login details. This worked out very well as it would save each user a lot of time once they start using the application, as well as because it is being run of the university sever there is no chance or risk of any personal information been accessed in any way. This was a very useful technique to use because it enabled each of the users to have trust in the application knowing that there is no risk of unauthorised access to their personal information.

As we decided to use the university's server to access the application, this already makes our application as secure as it can be. All of the information within the university's server is protected. This also means any information that each of the user's input using the application will be secure as there is no way any user outside the server will be able to access this information. Another very effective point about this is that we are using the university LDAP server. This means that we do not need to store any of the user's passwords at all as they are already been stored within the LDAP server. This is a very effective mechanism because it means that there is no way of any unauthorised access of accounts because the passwords are kept securely within the server, and not actually within the application. This will make all the information and data as secure as possible (Brown, 2008).

Social issues

There are many social issues which must be adhered to when we started implementing and creating the application. One of the main social issues which we had to make sure we addressed properly was the security issues with the information which is going to be kept within the application. This is to make sure no information ends up been in the wrong hands.

As we decided to use the university's LDAP server this meant that there were no passwords been stored within the application whatsoever. All of the user's passwords are already been stored on the server, which can help prevent any kind of un-authorised access to accounts. This is very effective because it will enable each user to have a lot more trust within the application.

This is essential to have because otherwise if the user does not trust the application with their personal data they won't want to use it as it could cause many problems.

Another social issue which we have taken into consideration when creating our application is to make sure there is appropriate options for disability access. For example, we are aiming to have a personal details section within our application. Each user will be able to access this section and they will be able to select or change options depending on what disabilities they may have. We believe this is an essential set of options to have to make sure all users who are going to use this application have no problems when using it what's so ever.

For example, we are trying to implement a colour-blind assist option. This will be very effective for users who may be colour blind as it will enable them to turn this on and it will then make the application a lot more suitable considering their disability. Another option which we aim to try and put into the personal details section would be a screen reader. This is an essential option which we believe we must include as it will enable users who are visually impaired to read the text within the application. We believe all of these options are essential to include within our application so it is easy to use for all kinds of users. Ease of use of the application is a huge aspect which we have taken into consideration when designing our application. Without this users may not wish to use the application as it may not suit everyone's needs.

Ethical issues

Also when creating our application there were many ethical issues which we had to make sure we were adhering to. If we did not adhere to these issues it could cause a lot of time to be wasted as problems may arise. Also fines could occur if certain things were not been adhered to properly. Many companies have recently been trying to get away with using illegal software. To prevent this from occurring BSA (Business Software Alliance) have been taking aggressive and severe action against the companies who not adhering to software privacy. Fines up to £110,000 have been issued to companies which their employees have been using unauthorised copies of software (Garrett & Lewis, 2009).

The first ethical issue which we had to make sure we addressed while creating our application was to make sure all of the software we are using has the correct licensing. If there was any licensing problems or disagreements with using a certain software this could cost the production team a lot of unnecessary money, as well as a lot of time been wasted.

To prevent anything like this from occurring when creating our application we have used a text editor called Sublime Text. This is a very effective and useful piece of software which would enable us to code the application completely from scratch. Sublime Text is a free piece of software which anyone is allowed to download and use. However, if there is continued use of the software and it is intended for a company to continue to create software, a licence must be purchased. However for the purpose of our application the free download of the software was most efficient and effective.

Another ethical issue we had to make sure we were addressing while creating our application would be to address all known bugs. This is essential to do consistently throughout the production of the application to make sure there are no major faults within the application at the end of the process. To make sure we have addressed this issue, we are testing the application thoroughly throughout the application. If we notice any kind of bugs or unexpected behaviour from the application. We will then be able to identify where the fault has come from and then address it appropriately so this does not occur in the future. This is essential to do throughout the production of the application because it will hopefully enable the final application to work smoothly without any unnecessary faults. This will save a lot of time as well as it will ensure the application is working appropriately.

Professional issues

There are many professional issues that we must follow when creating our application. One of the professional issues which we pointed out before we even started the development was the market. The market entails four main points. These are the Product, price, place and promotion (The 4 P's).

First of all we needed to make sure we understood the product which we are creating thoroughly. To do this we have followed the client's brief very closely to make sure we are addressing all of the essential needs. Also we have met up with the client on a regular basis

to make sure we are meeting the needs of the client and everything's been done properly. This is a very effective and useful technique to use because it allowed us as a team to stay on track with what we are doing, and meet the client's requirements without any changes needing to be done. By sticking to this method it will save us a lot of time and allow us to meet the client's requirements efficiently.

Another factor in which we will have to take into consideration is the price of the application. As we are planning to white label our application, price is an essential professional issue which we must take into consideration. For example depending on how we decide to distribute our application will determine how much the price should be.

For example, we are planning to give each different company (University/school) a choice to adjust or change the logo's/colour scheme of the application, depending what company is buying/using the application. This will be a very effective option because it will enable the application to feel more suited for each of the companies. This will enable us to increase the price of the application because the application will then be more personal and suited for each company.

This is where place comes into it. Originally the application has been created for Cardiff University use only. But depending on how the application does and if it is to be successful when used on a regular basis, the application could possibly be used in other universities. We have to take many points into consideration such as franchising and how we will make the application personal for each company. This is why we have decided to white label our product/application. If the application is to be successful this means that we will then be able to change and modify the application to suit a particular user's needs (Colour schemes etc.).

I believe this will be a very useful and effective decision because it will enable the application to be widely used once completed as it won't be designed for one particular company.

Finally the last point of professional issues would be the promotion. As this is only a university project the promotion of the product is most likely going to be done within Cardiff University itself and at a very low cost. If the application is successful on completion there are many options which we will be able to look at for promoting our product. One option that we have already taken into consideration would be student enterprise schemes. These kind of schemes would provide us with useful information for promoting as well as advertising for the application itself. This is all depending on how well the application does and if it is successful in what it is needed for. We are even hoping that word of mouth could help us promote our product/application through the different departments within the university if our product is successful for the client's original needs (Boundless Business, 2015).

Work Plan

Interim Deliverables

In our initial report we outlined most of the features of our application that we would deliver by the interim deadline. These features were: Overall framework, ad-hoc grouping and a base for our custom CMS (content management system). These features are those which are most important to us and our client. The idea of ad hoc grouping has always been our USP (unique selling point) and the custom CMS will allow the application to be tailored very easily to individual users and organisations, making it much easier to market and sell our product.

When we started building our application and after conducting further research we discovered that it was possible to link our application to the Cardiff LDAP system. This means that students are able to login to the application using their existing student numbers and passwords without having to go through a registration process. This also then takes some of the pressure off us when it comes to data protection as the storage of data is external to our application and is already part of a secure system. This task took precedence over what we had originally planned for at this stage.

In addition to the application itself we are required to produce this report, the size of our group has allowed us to split the tasks, some group members will work on the application itself while the rest will perform the additional research required for the report.

Interim Tasks

The application has been split into various parts for different group members to work on simultaneously. These parts consist of design, framework, database storage, use case diagrams and creation of a forum. These were identified as the main features of our project that could be created independently and later brought together easily. Our group consists of nine members so this leaves four to work on other tasks. These other tasks consist of work plan creation, design justification, strengths and limitations of our design and legal, social, ethical and professional issues.

Due to the size of our group and the working methodology we have employed it allows us to have multiple milestones at once. The first set of milestones for the interim stage are: LDAP Login, Wireframe designs and a start to the forum.

Not all of our tasks will be required to be complete at this stage of the project, but some of these will take a long time to complete and our group has decided it is best to start work on these now so they are completed well before the final deadline to allow for adequate testing and the creation of the final report.

Future Deliverables

For our final deadline we hope to have successfully implemented the following features:

- Framework
- CMS
- Ad-hoc grouping
- Multiple question style templates
- Students to ask questions via forum
- Voting on questions, most popular appear at top of list
- Cross platform support (prioritise smartphones and tablets)
- Students join lecture room via code
- White labelling
- LDAP Login

Alongside the application we must also prepare a presentation which will include a demonstration of the finished product.

Future Tasks

The features as listed above will form the basis of our task list for the final stages of our project. Once we reach this stage these tasks will be better understood and can either be split further for individuals to cover or approached in pairs or larger groups.

Our first set of milestones for the final project phase are the framework, finalised design and the CMS. The design can then be implemented and the CMS in place allowing for easy user content editing. These will form the backbone of the application and allow for other features to be implemented and tested. We aim to create a fully responsive design allowing for a wide range of mobile and tablet devices to be used. Our next major milestones will be ad-hoc grouping, forum and question templates. As stated previously the ad-hoc grouping is our most important feature and the main stipulation from our client. Once the forum has been completed we will implement the voting system to re-order the questions accordingly. When these features are complete we can as our final milestone link the lecturers quiz to the students app. Once all this is completed we can perform any further tweaks and begin our final testing.

Risks and Mitigations

Where these tasks have been split up they can easily be assigned to an individual group member. Once a task has been completed that group member will become free to perform other tasks or to assist other members with larger tasks, these needs will become more apparent as development continues. Working in this modular manner we hope to alleviate the bottleneck that occurs with final implementation. Each feature can be implemented and tested separately upon its completion. This will mean that when the final task is complete everything will already be ready and working as opposed to having to put together everything at the same time at the end of the project. We have also kept tasks that are dependent on a previous tasks completion to a minimum but this is unavoidable in some areas.

This plan comes with inherent risks. We are relying on each member contributing to their own section in good time, this can become very problematic when one task relies on the completion of a previous task. Having the tasks divided among all members of our group also leaves us with the risk of non-completion if a member falls ill or is unable to continue their work for any reason.

To try and mitigate these risks we are relying on a solid communication network between our group members and imposing strict deadlines on tasks, these deadline are documented in our Gantt chart (appendix C) for our project timeline. If a task is not reported as complete by this deadline then we can address this as quickly as possible. We also have a social networking group set up between our members as well as sharing other work via google docs and other online tools. This all serves to allow us to monitor what work has been completed and what else is required to do. If we can see that one section has not been completed we can contact that group member to make sure there are no issues or provide assistance if necessary. Minimising these risks still depends on group members reporting when they come up against issues, as a group we have endeavoured to create a friendly social atmosphere between us to encourage members to talk to us if they have issues and reallocate tasks where necessary.

Conclusion

With all of the design documentation in place, a basic working prototype and a good plan on how to test our final product we are certainly on the way to making a well-made application to be used in lectures by students and lecturers. Users can already login using their existing university accounts using the LDAP system, and a page for students to pose questions to their lecturers just needs to be implemented when the rest of the system is live.

In theory, after the work from this stage of the project it will be relatively easy to create the remaining features of the website using the designs for the user interface, database backend and use cases. We have identified some positive and negative aspects of our design and implementation, which we will hopefully be able to work around/avoid during the third and final stage of development. With an updated Gantt chart and our future deliverables clearly stated, we will confidently be able to complete this task on schedule.

In reflection we all worked well as a group and attendance to weekly meetings remained high from most group members throughout the entire stage of our development. We delegated tasks evenly amongst the group and stayed in contact at all times in between group meetings through Facebook messenger which we found effective in the initial stage. During this stage we tried to help each other with tasks as much as possible so that everyone had a clear understanding of the task they had to complete. We did this during our weekly group meetings that proved to be invaluable to the group's progression. We also gained important and valuable information regarding system requirements and report writing from our monthly meetings with our module leader Frank Langbein along with team member Harry being in contact with our client Dave Marshall throughout the development process.

To improve our group's skills as a whole for the next stage of the development process we make sure that everyone attends every group meetings where possible, this would make sure that everyone has a full understanding of the task at hand and that everyone is fully involved in the development of our project.

References

Boundless Business. (2015, July 21). *The Marketing Mix*. Retrieved February 11, 2016, from Boundless: <https://www.boundless.com/business/textbooks/boundless-business-textbook/marketing-and-the-customer-relationship-14/marketing-strategies-92/the-marketing-mix-433-3332/>

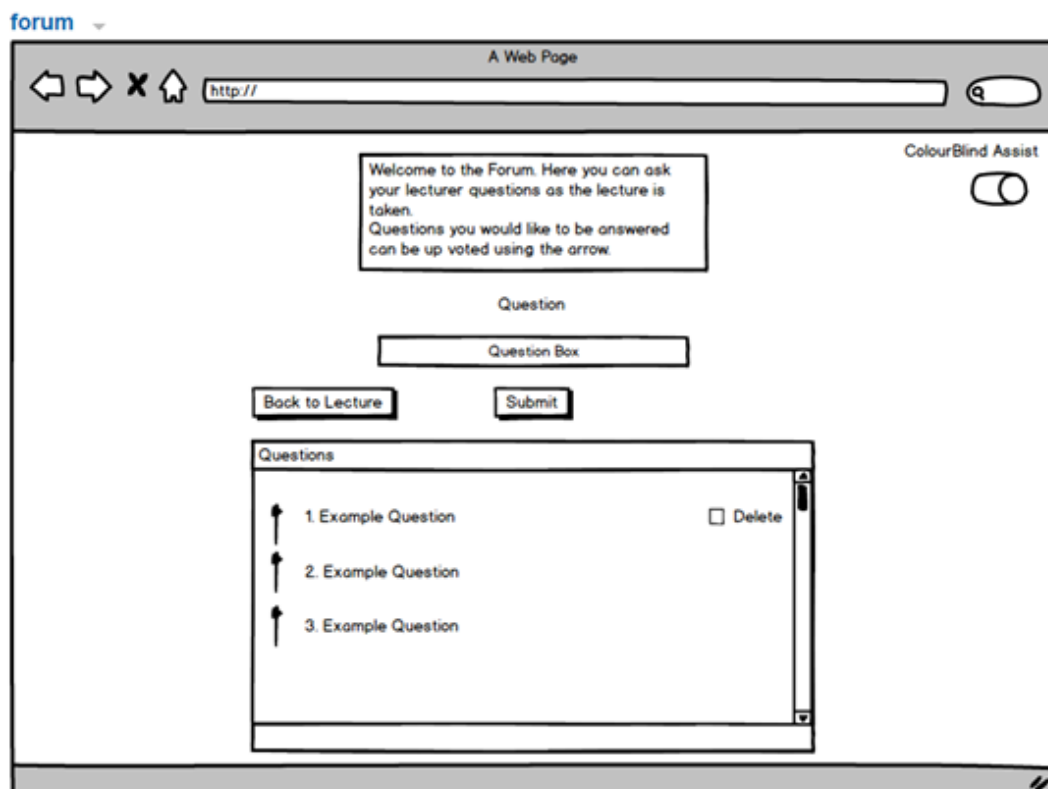
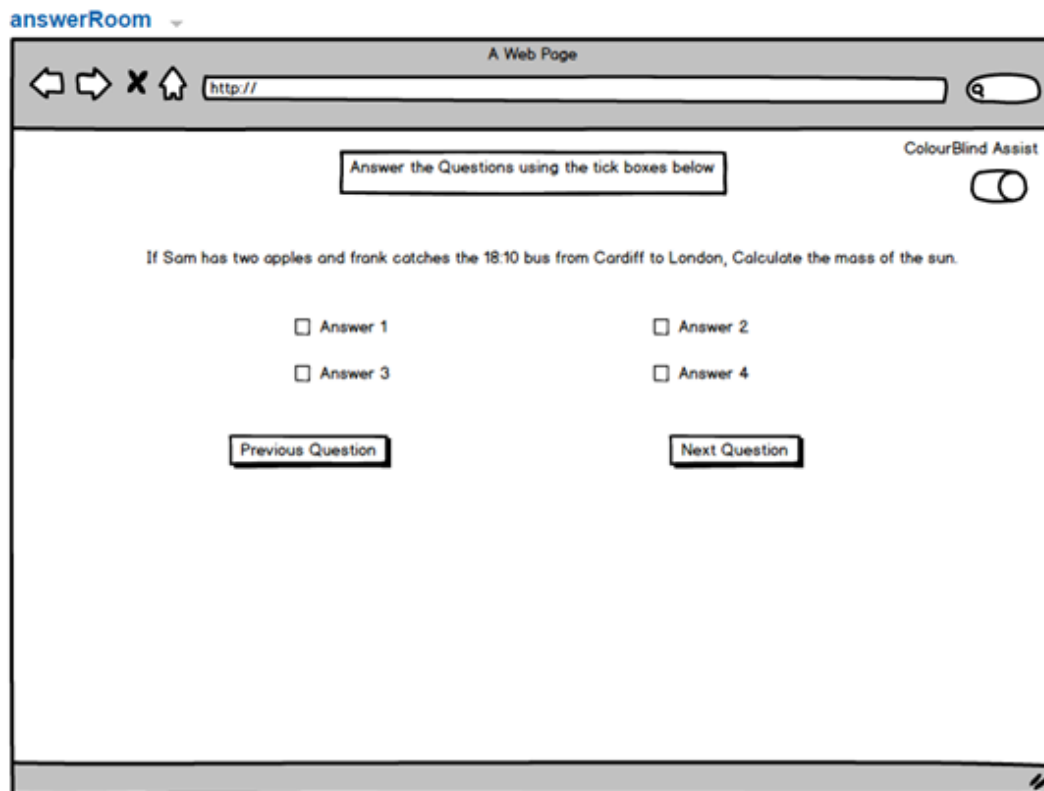
Brown, I. (2008, September 7). *Software Legal Issues*. Retrieved February 11, 2016, from Slideshare: <http://www.slideshare.net/blogzilla/software-legal-issues-presentation>

Garrett, R., & Lewis, J. (2009, January 19). *Ethical Issues in Software Development*. Retrieved February 11, 2016, from Scribd: <http://www.scribd.com/doc/10880744/Ethical-Issues-in-Software-Development#scribd>

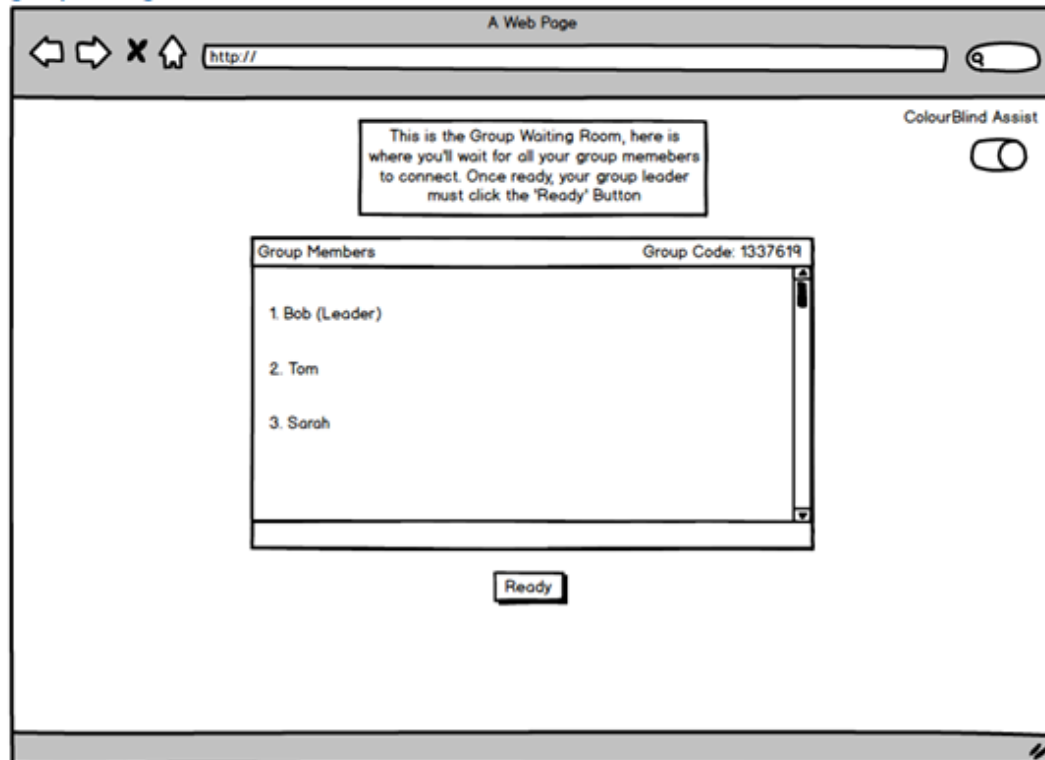
Nishadha. (2012, March 16). *Benefits of Using Wire-Frames*. Retrieved February 11, 2016, from Creately: <http://creately.com/blog/diagrams/wire-frame-benefits/>

Pinal, D. (2007, November 26). *SQL SERVER – Rules of Third Normal Form and Normalization Advantage – 3NF*. Retrieved February 11, 2016, from SQLAuthority: <http://blog.sqlauthority.com/2007/11/26/sql-server-rules-of-third-normal-form-and-normalization-advantage-3nf/>

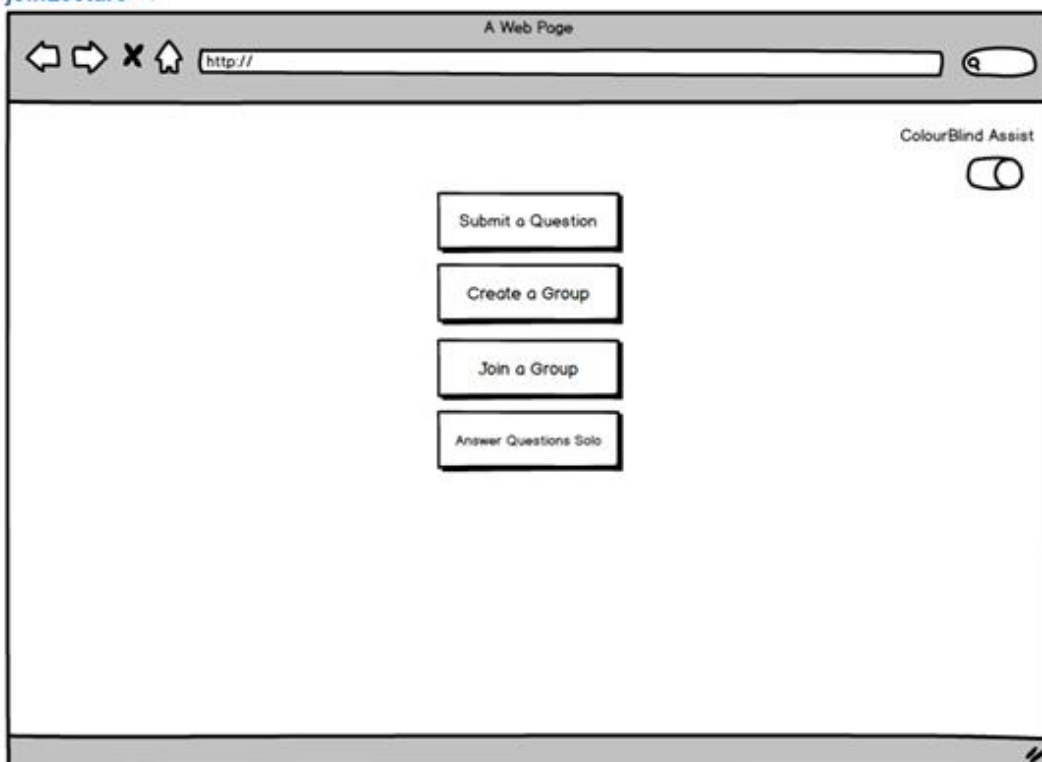
Appendix A - Balsamiq Screenshots



groupWaitingRoom



joinLecture



A Web Page

http://

ColourBlind Assist

Start Session

View Forum Board(s)

Prepare Questionnaire

Personal Details

not design
discussed

A Web Page

http://

ColourBlind Assist

Please Enter your Cardiff Student Number.

Student Number

Upon submitting this form an E-mail will be sent to your Cardiff Student e-mail address with your new password

Send E-mail

personalDetails ▾

A Web Page

http://

ColourBlind Assist

Would you like to set a nickname?

Nickname

Would you like to change your password?

Old Password

New Password

Type New Password Again

Submit

results ▾

A Web Page

http://

ColourBlind Assist

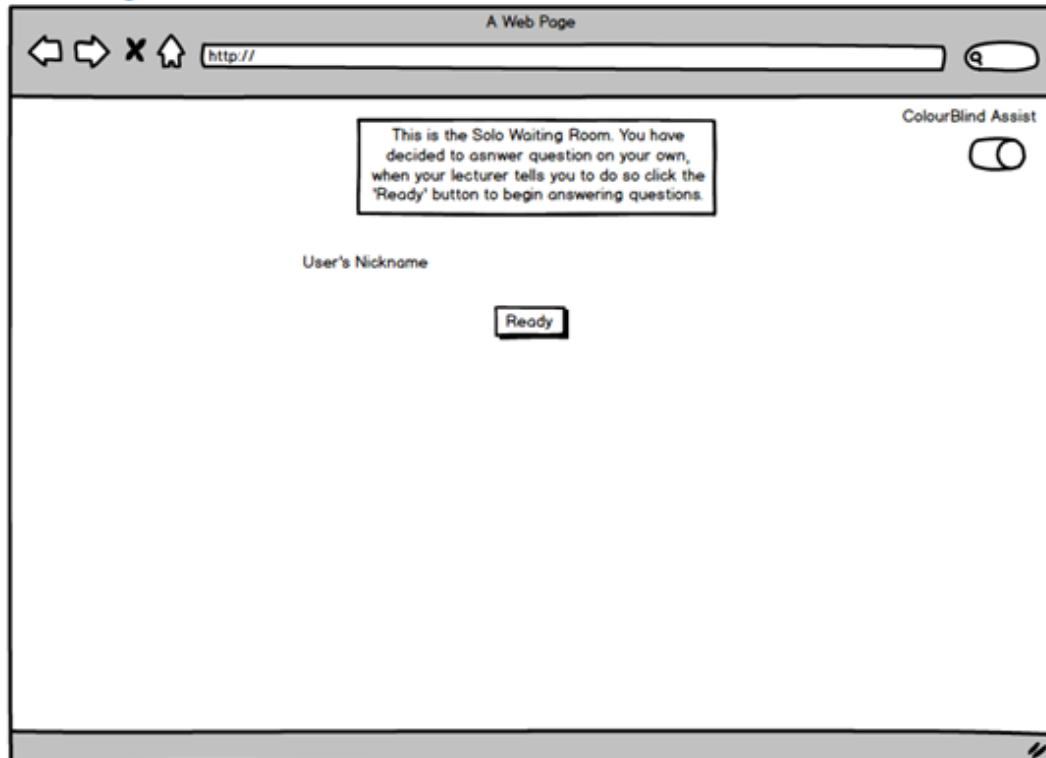
Congratulations, Your answers from the quiz are displayed below

Question	Answer	Your Answer	Average
1. Some Question	A	A	A
2. Some other question	A	B	C

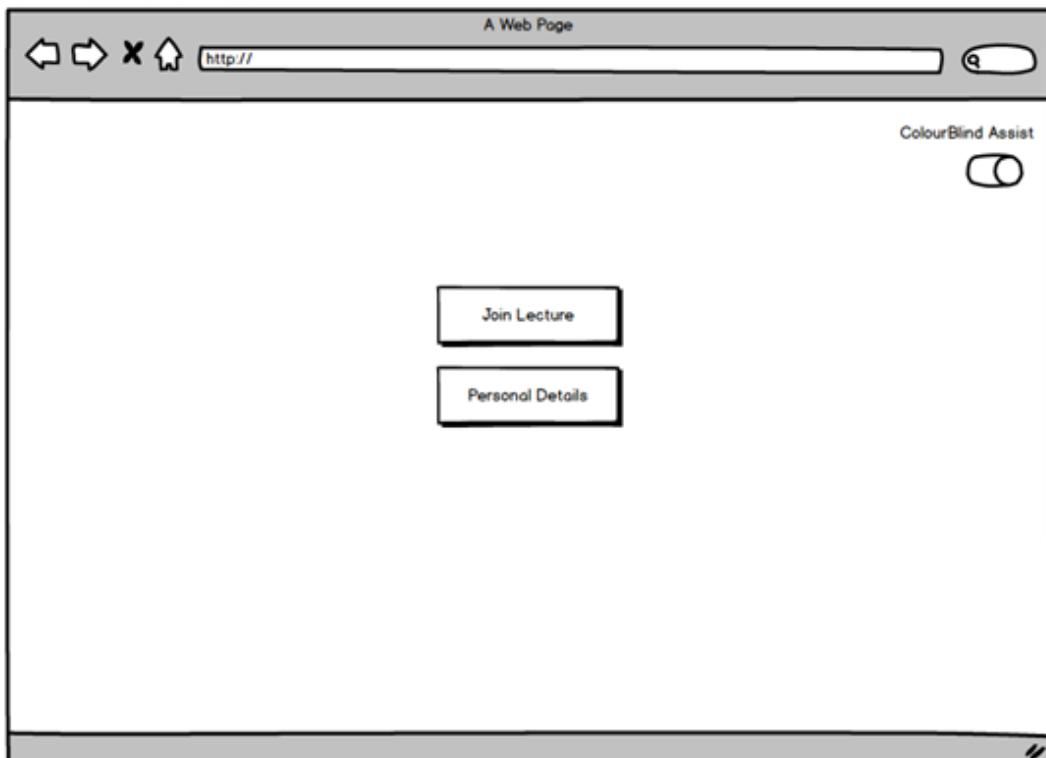
You can now use the 'Forum' to ask questions about the quiz to your lecturer

Forum

soloWaitingRoom ▾



studentControlPanel ▾



Appendix B - Creating New Tables

Using the functional dependencies diagrams we were able to create relations. The relations (or tables) that we have used can be easily implemented into code for creating database tables. We have used MySQL as our query language as shown below. The tables created below are only made from its simplest form without adding any additional constraints. However, if need be, we can easily alter a table by creating an alter query on the desired table.

```
CREATE TABLE Student
(
    SNumber varchar(255) NOT NULL,
    SFirstName varchar(30),
    SLastName varchar(30),
    SPass varchar(255),
    Alias varchar(255),
    PRIMARY KEY (SNumber)
);
CREATE TABLE GROUP
(
    SNumber varchar(255) NOT NULL,
    Questionnaire varchar(500) NOT NULL,
    GroupCode int,
    PRIMARY KEY (Questionnaire),
    FOREIGN KEY (SNumber) REFERENCES Student(SNumber)
);
CREATE TABLE GroupLeader
(
    GroupCode int NOT NULL,
    GroupLeader varchar(255),
    PRIMARY KEY (GroupCode)
);
CREATE TABLE QuestionnaireQA
(
    QQuestion varchar(500) NOT NULL,
    ValidAnswer varchar(500),
    PRIMARY KEY (QQuestion)
);
CREATE TABLE GroupMark
(
    GroupCode int NOT NULL,
    QQuestion varchar(500) NOT NULL,
    GroupMark int,
    FOREIGN KEY (GroupCode) REFERENCES GroupLeader(GroupCode),
    FOREIGN KEY (QQuestion) REFERENCES QuestionnaireQA(QQuestion)
);
CREATE TABLE QuestionnaireResult
(
    QQuestion varchar(500) NOT NULL,
    LSessionCode varchar(50) NOT NULL,
    OverviewResult int,
    FOREIGN KEY (LSessionCode) REFERENCES Lecture(LSessionCode),
    FOREIGN KEY (QQuestion) REFERENCES QuestionnaireQA(QQuestion)
);

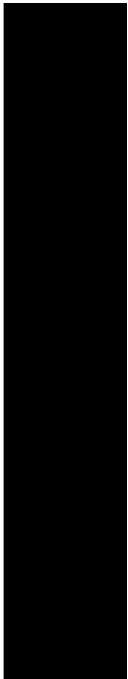
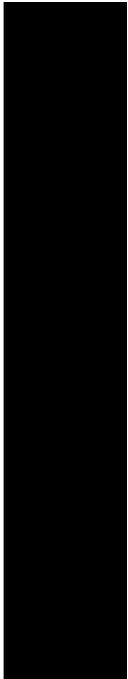
CREATE TABLE Lecturer
(
    LNumber int NOT NULL,
```

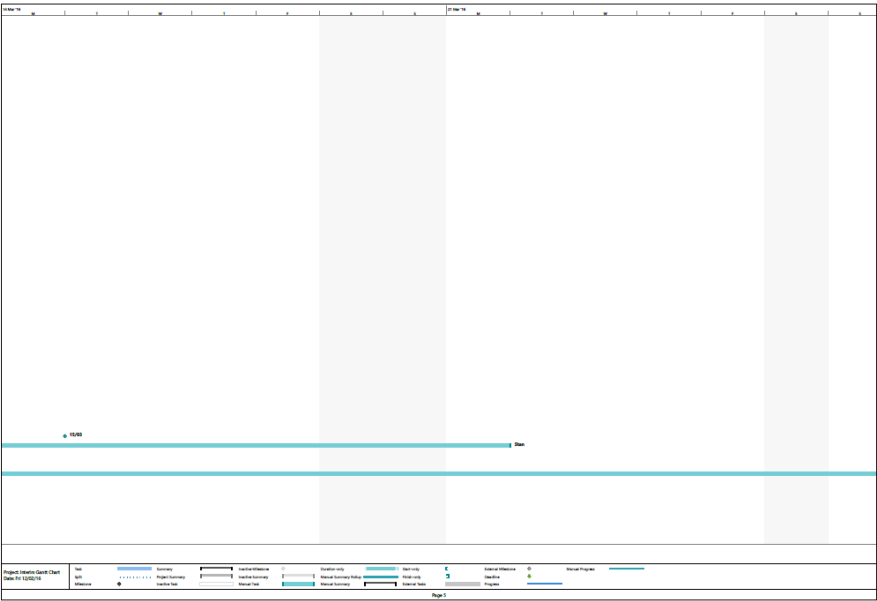
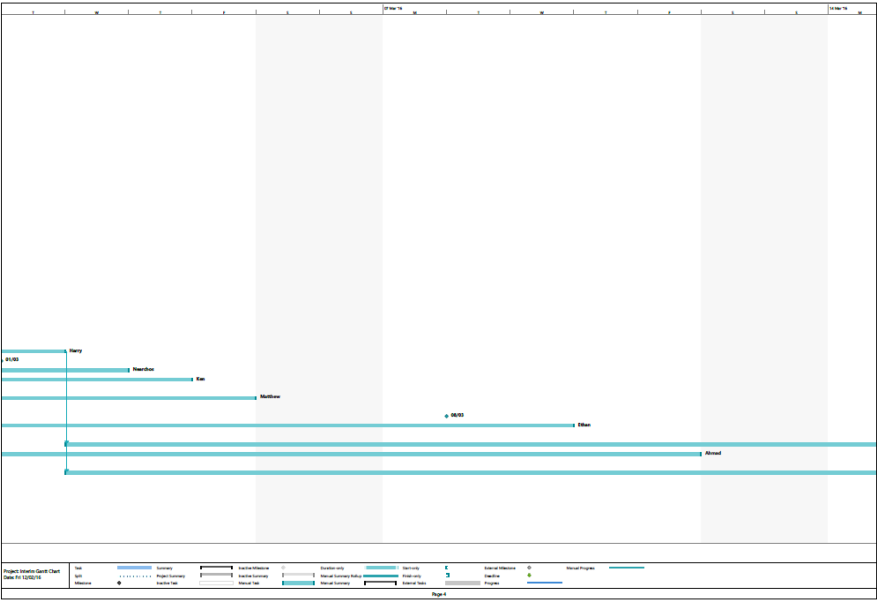
```

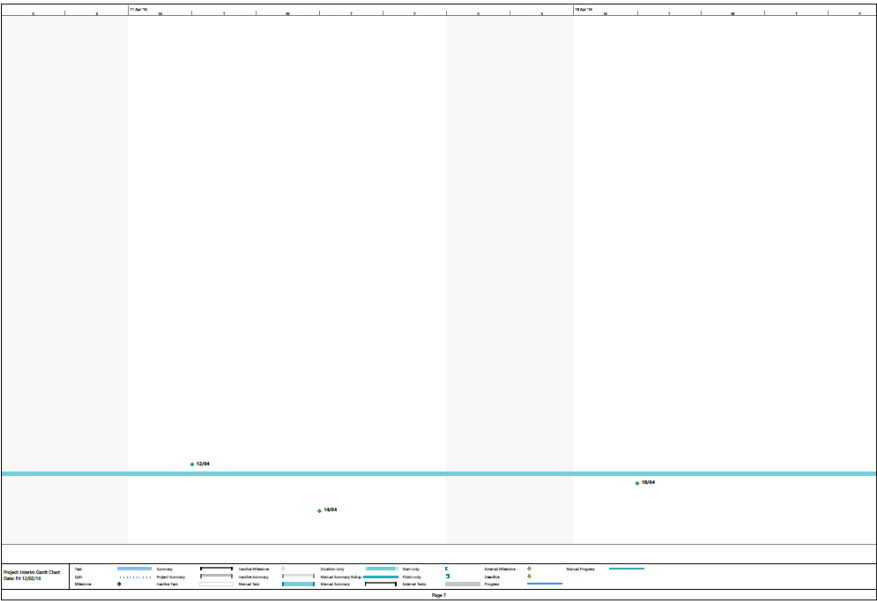
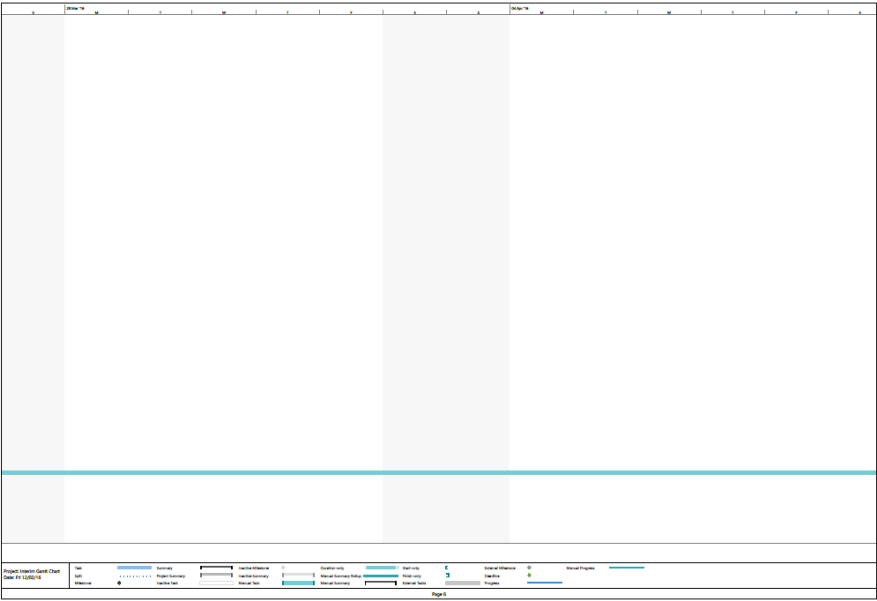
        LFirstName varchar(30),
        LLastName varchar(30),
        PRIMARY KEY (LNumber)
);
CREATE TABLE LectureSession
(
    LSessionCode varchar(50) NOT NULL,
    LNumber int,
    PRIMARY KEY (LSessionCode)
);
CREATE TABLE Forum
(
    LSessionCode varchar(50) NOT NULL,
    FQuestion varchar(500) NOT NULL,
    Upvote int,
    FOREIGN KEY (LSessionCode) REFERENCES LectureSession(LSessionCode),
    PRIMARY KEY (FQuestion)
);

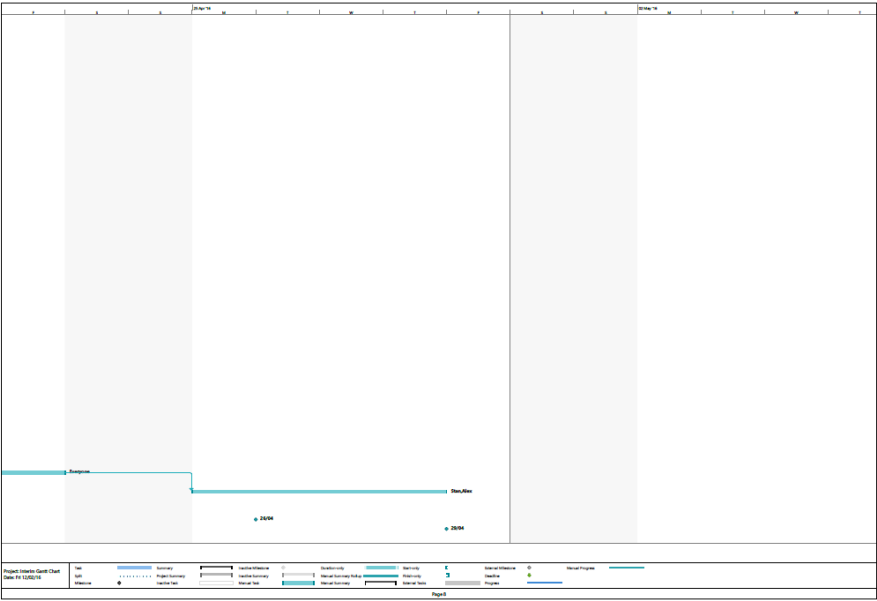
```

1. *Journal of the American Medical Association*, 2000; 283: 2689-2693.









(Full Gantt chart available in separate file)