****

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 3](#_Toc442800968)

[User Interface Design 4](#_Toc442800969)

[Connecting to the university LDAP server 5](#_Toc442800970)

[Forum 7](#_Toc442800971)

[Functional Dependencies 13](#_Toc442800972)

[Test Cases 16](#_Toc442800973)

[Use Cases (Not Finished) 24](#_Toc442800974)

[Justification for design and implementation (Not Finished) 26](#_Toc442800975)

[Strengths and Limitations (Not Finished) 27](#_Toc442800976)

[Legal, Social, Ethical and Professional Issues 31](#_Toc442800977)

[Work Plan 35](#_Toc442800978)

[Conclusion (Not Finished) 38](#_Toc442800979)

[Appendix A 39](#_Toc442800980)

[Appendix B 49](#_Toc442800981)

# Introduction

Since the time of writing the Initial Report, the team has made many steps towards completing the final product. Firstly we designed roughly how the user interface should look and how we can test the website, then we designed how the back end of the website should work. With these in place we then began to create a prototype, demonstrating some basic functionality. Along the way we have thought further about social and legal issues.

# 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.

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 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 automagically 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.

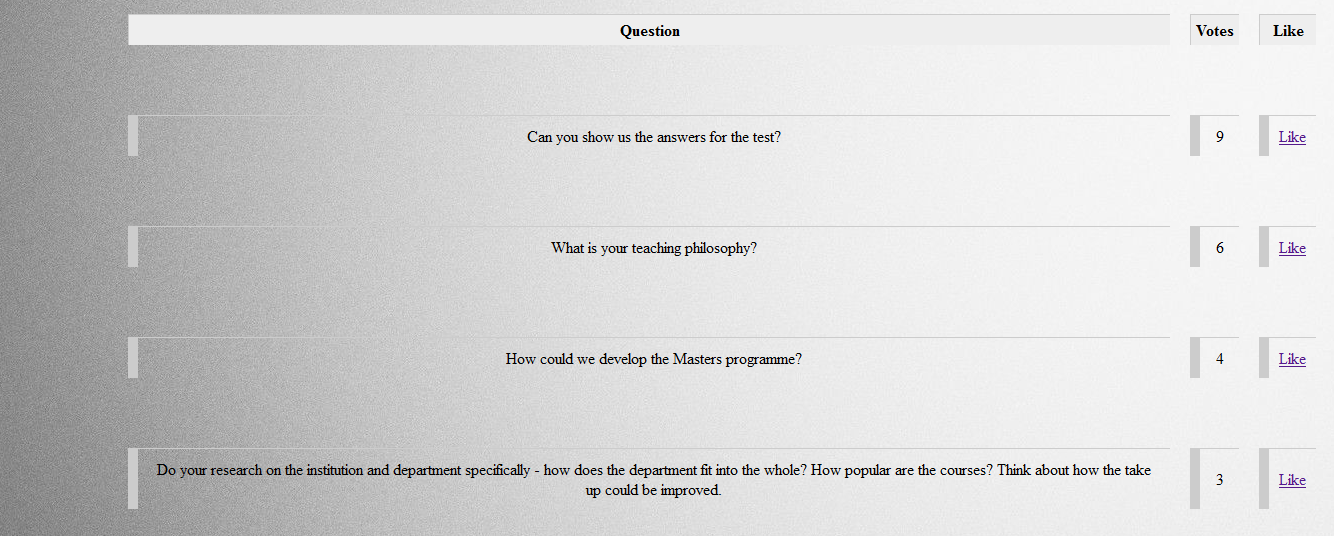
Below is a screenshot of the code that enables us to authenticate the users with the university LDAP server.



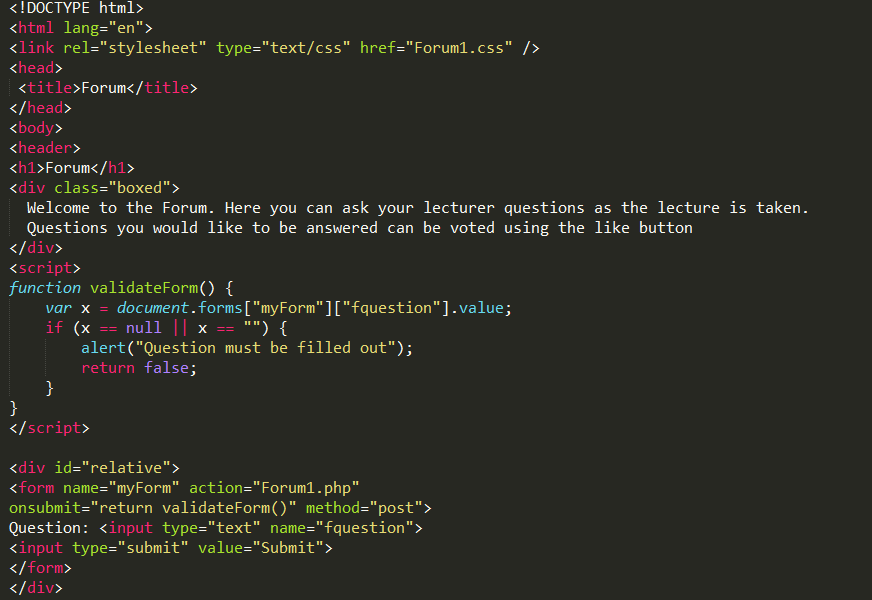
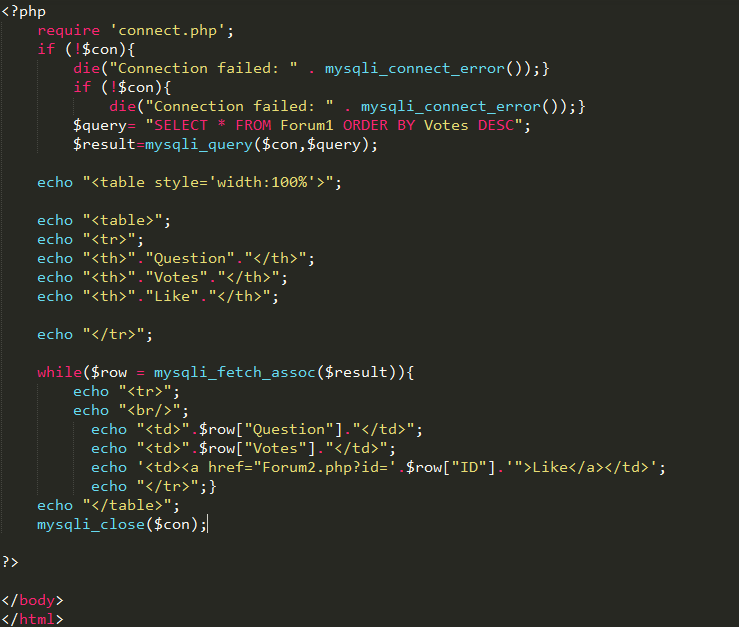
# Forum

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***](https://www.cs.cf.ac.uk/phpMyAdmin/index.php)***.***

Here is a sample of how the Forum page will look like:

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

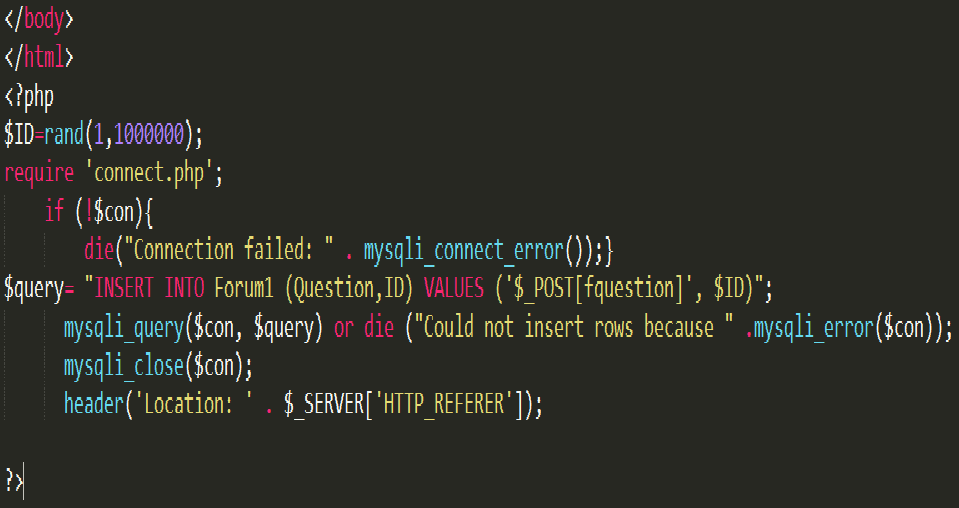
Forum.php

This is the main file for the Forum page.

-Submit Button: When the user presses the submit button there is validation that checks if the text box 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.

Forum1.php

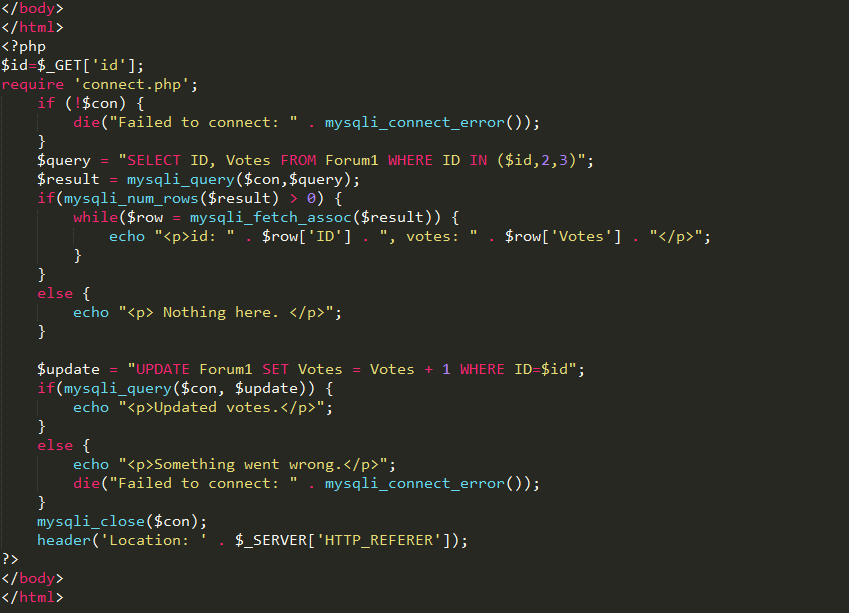


This is the file where the question is stored in the database after the user press the “submit” button. It creates a random ID for each question that is not shown anywhere in the page but it will be later used for the Forum2.php file when the users will vote for a question.

Finally the array “header('Location: ' . $\_SERVER['HTTP\_REFERER']);”

will redirect the page to the initial page.

Forum2.php



This is the file that is been used for the votes. Basically 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



This is the file where the connection takes place

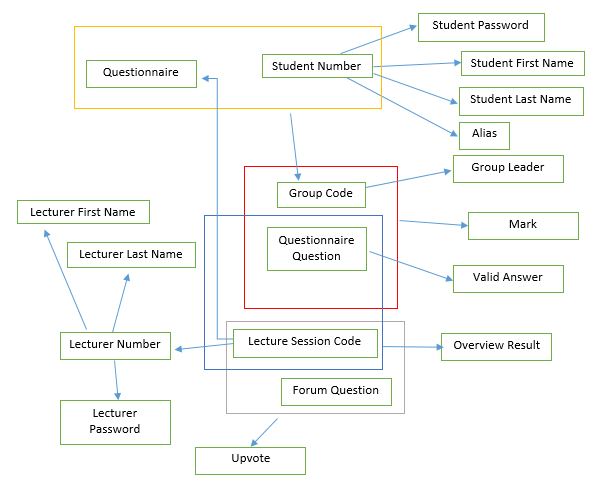
# Functional Dependencies

The functional dependency diagram presented above 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 above 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 above the assumptions that we have made to reach the solution we have.

The relations or tables implemented from the diagram 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 will be used as a key in the database structure. When the diagram is normalised, we are able to use the keys to relate the tables together.

Using this structure of the database gives us a cleaner perspective of saving data. 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 as seen in appendix B.

****

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)

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 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 (Not Finished)

Use case Diagram- Alex

Use cases – Ethan

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 fills in all the form fields asking for personal details (username, password, e-mail address, matching email-address, name, surname, DOB – any more or less?). Additional information about the input required is displayed where necessary.

4)The user ticks the box to accept terms and conditions of data privacy. (?)

5)The user presses “Register” button to finish registration.

6)The system displays the message “Registration Successful” and new user’s details are saved to the database.

(email sent to the user to confirm registration?)

Alternative Flows:

3A) The username chosen by the user is already in use by someone else.

    3A1) The error message “Username not available” is displayed. The user is required to choose other username.

3B) The e-mail address is already present in the system.

    3B1) The error message “This e-mail address already exists in our system” is displayed. The user is required to choose another e-mail address. (Password recovery?)

3C) The username is not valid.

    3C1) The error message “The username is not valid. Please enter a valid username” is displayed. The user is required to choose a valid username.

3D) The password is not valid.

    3D1) The error message “The password is not valid. Please enter a valid password” is displayed. The user is required to choose a valid password.

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

    3E1) The error message “Please fill in all the fields” is displayed. The user is required to fill in all the form fields.

4A) The user doesn’t accept terms and conditions and leaves the tick box empty.

    4A1) The error message “Please read and accept Terms and Conditions of Data Privacy before proceeding with your registration” is displayed. The user is required to tick the box to accept the Terms and Conditions.

# Justification for design and implementation (Not Finished)

# Strengths and Limitations (Not Finished)

Design

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. Our 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 design also includes a database design, which explains the relationships between various entities within our application, giving us a better idea of the overall structure of our implementation should be.

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.

Implementation - Strengths Ethan

Weaknesses - Stan

# 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.

(Legal issues:<http://www.slideshare.net/blogzilla/software-legal-issues-presentation>)

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 what so ever. 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 was 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.

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.

(Ethical issues:<http://www.scribd.com/doc/10880744/Ethical-Issues-in-Software-Development#scribd>)

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 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 is 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 will be 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 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 applications is successful on completion there are many option 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 scheme 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.

(Professional issues: https://www.boundless.com/business/textbooks/boundless-business-textbook/marketing-and-the-customer-relationship-14/marketing-strategies-92/the-marketing-mix-433-3332/)

# 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 milestone 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.

Final 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.

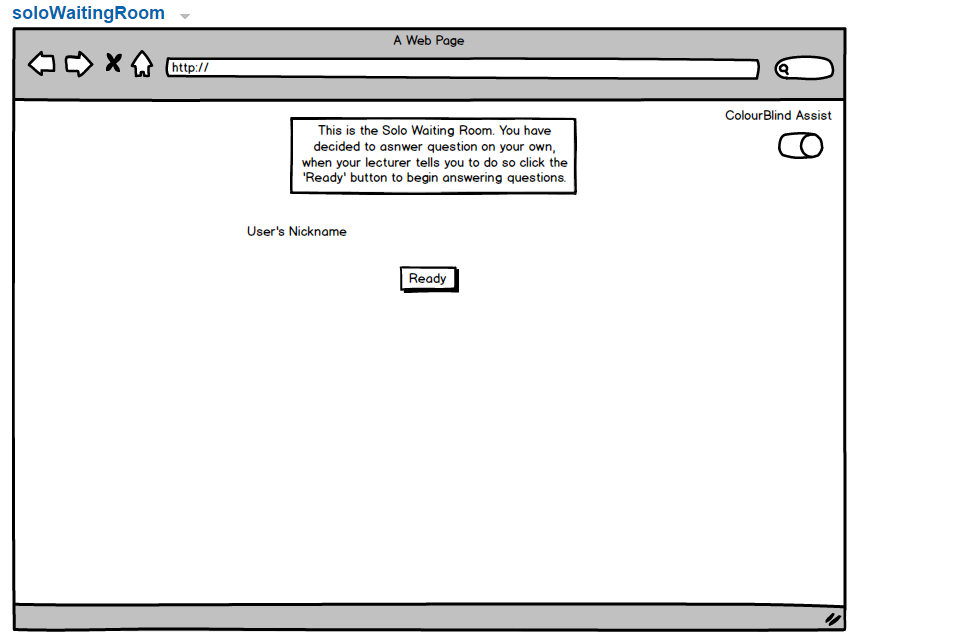
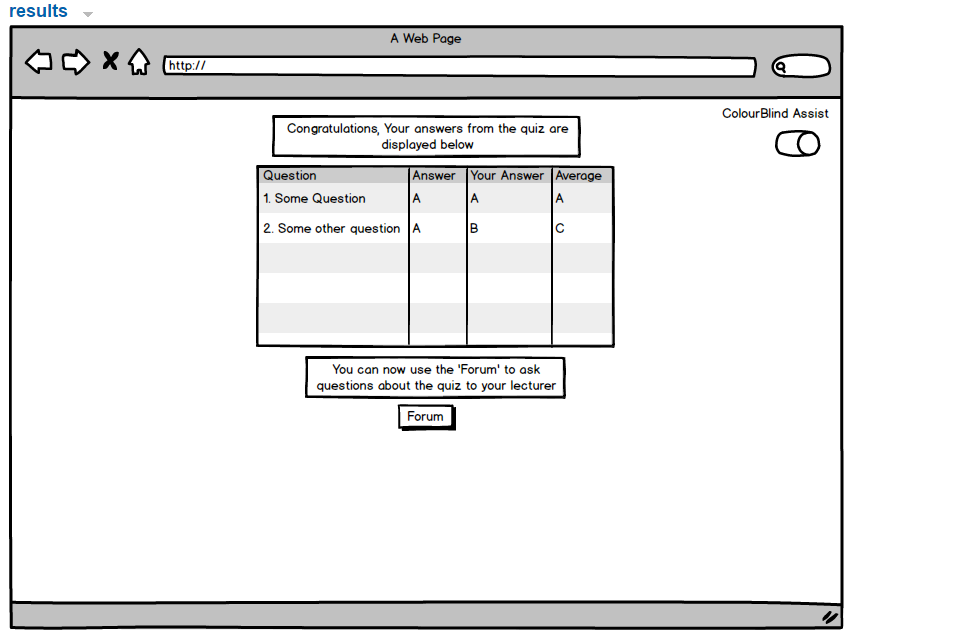
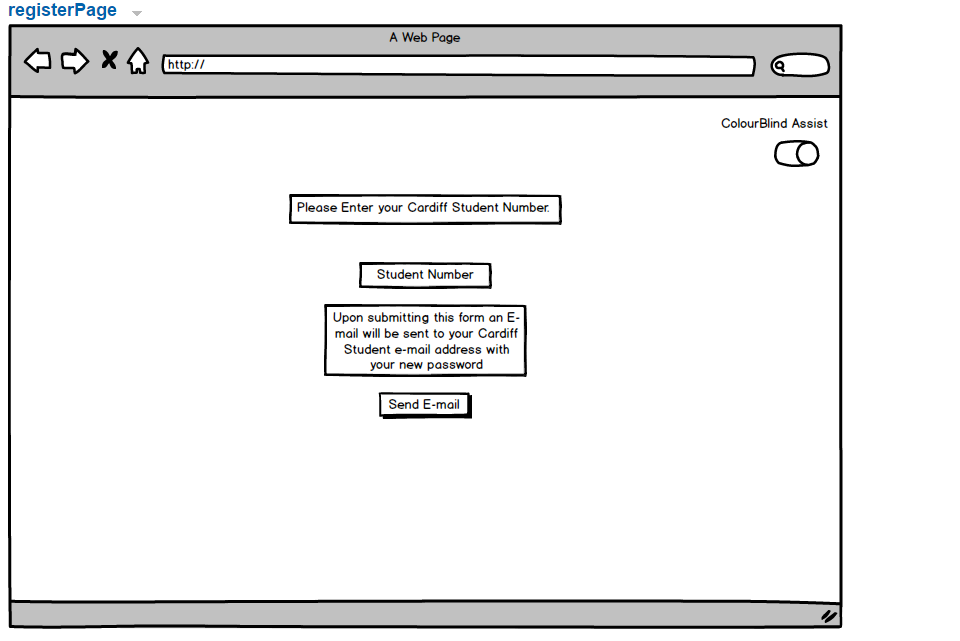
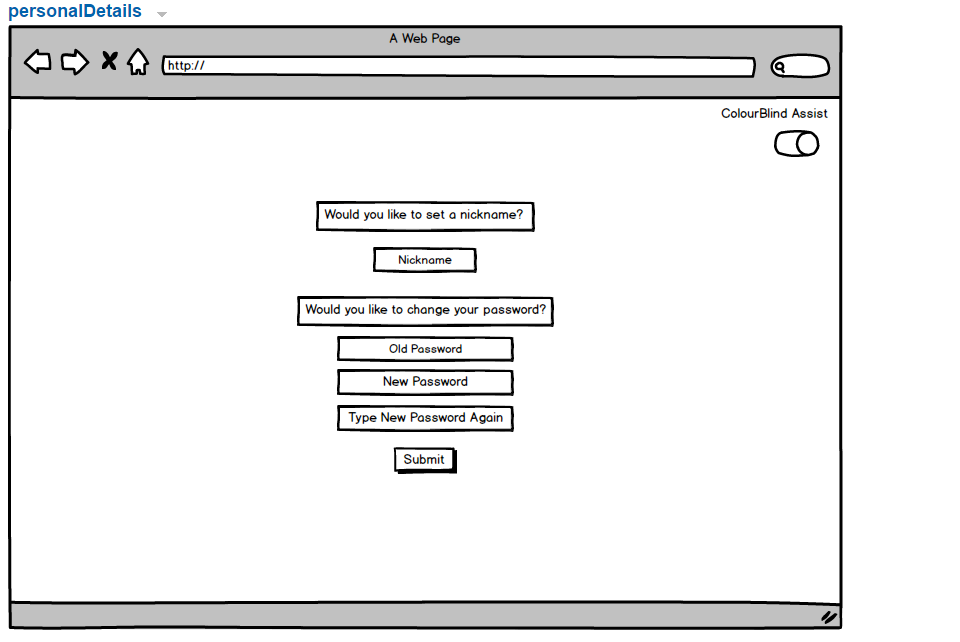
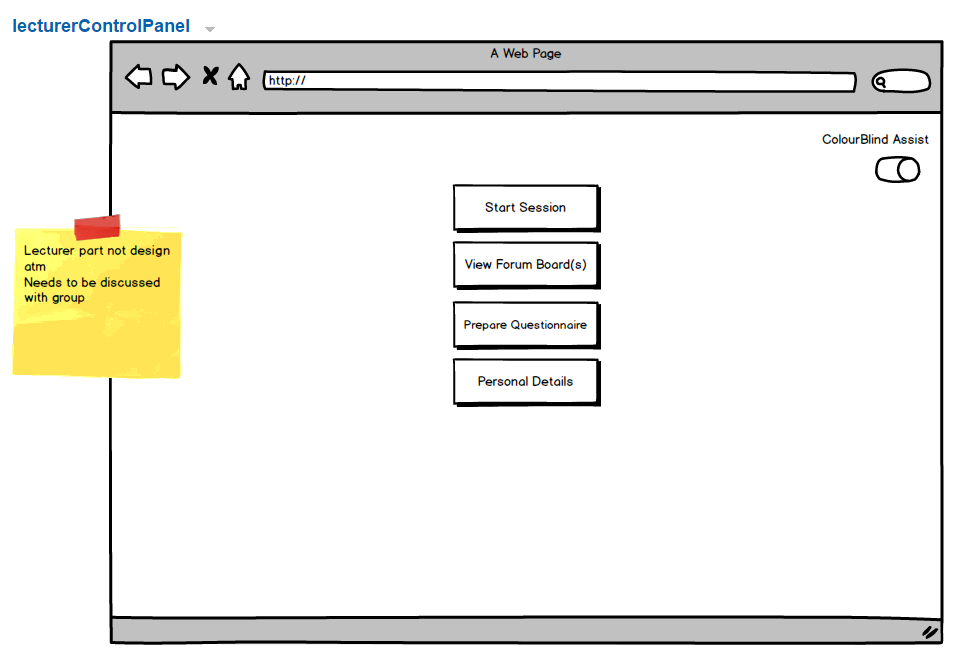
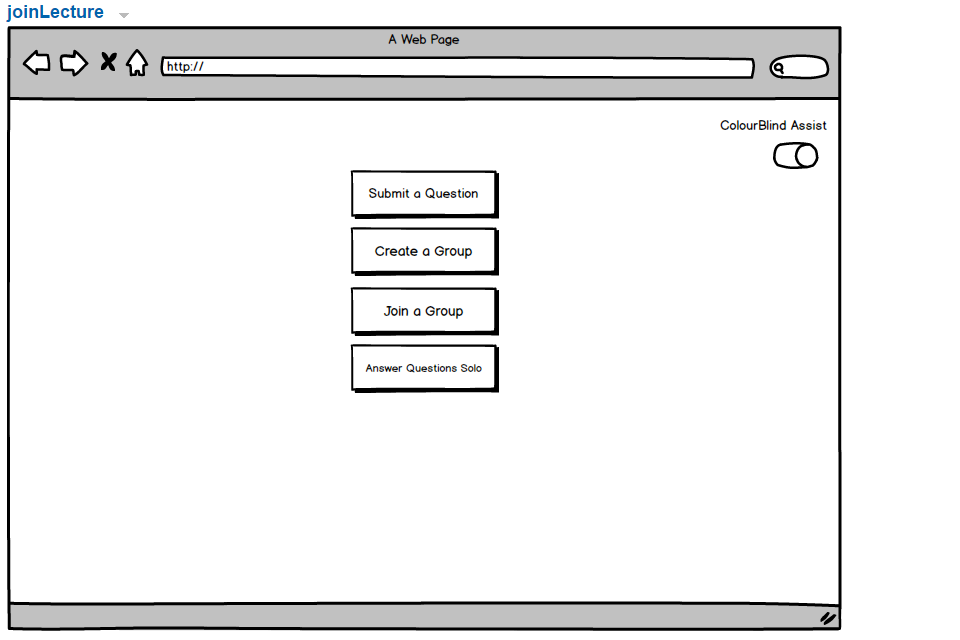
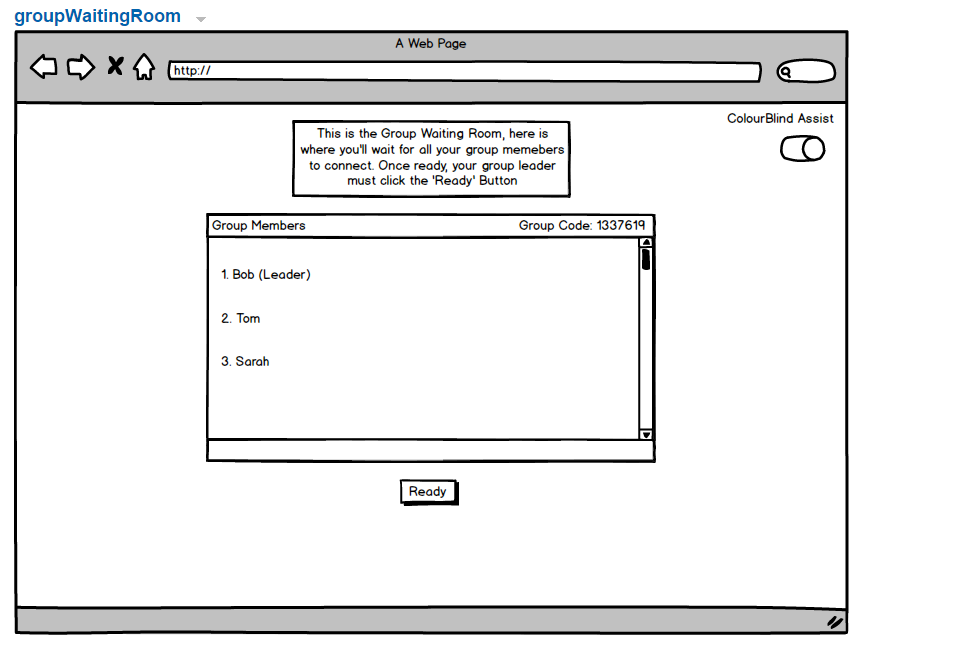
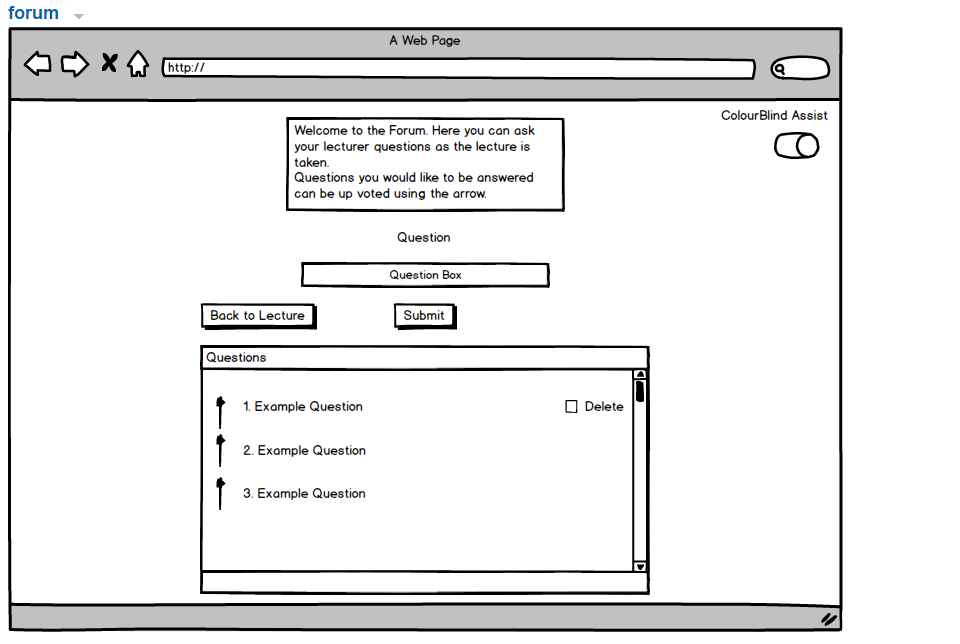
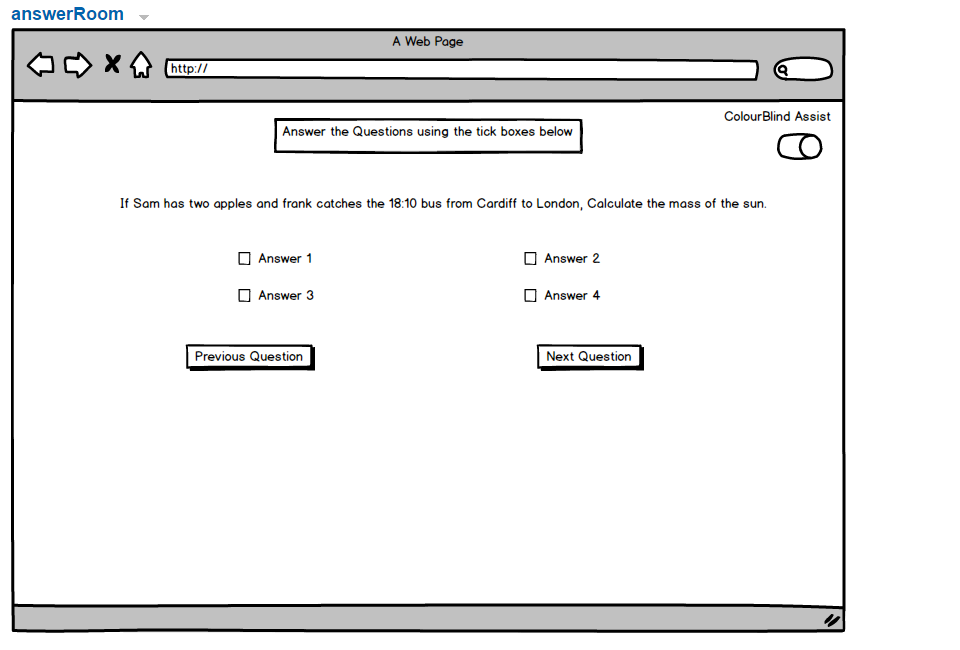
Risks and Mitigations

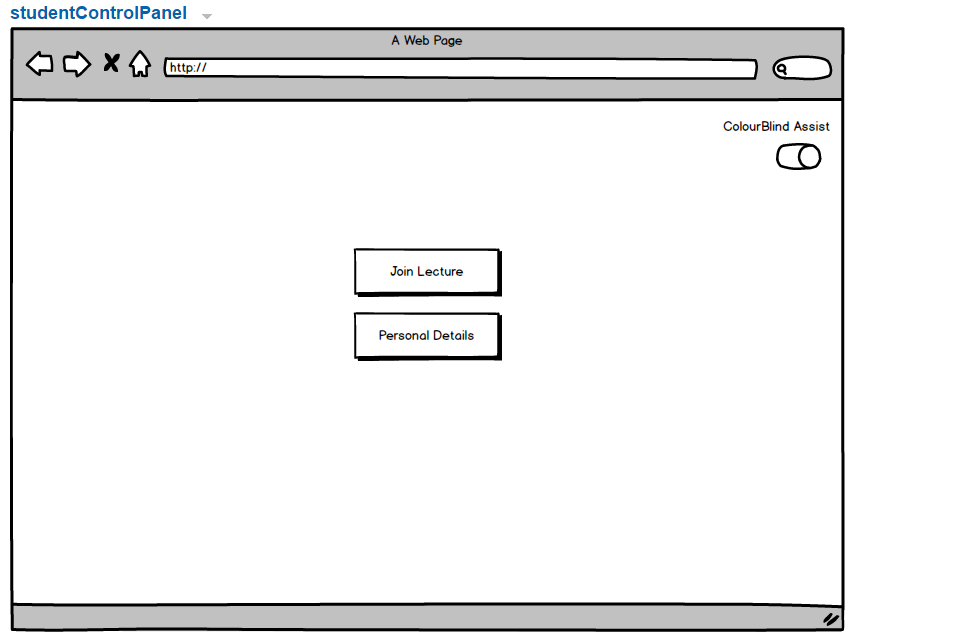
Where these tasks have been split up they can easily be assigned to an individual group member. Once a task has been completed that group member will become free to perform other tasks or to assist other members with larger tasks, these needs will become more apparent as development continues. Working 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 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 (Not Finished)

Appendix A****

****

## Appendix B

*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)*

*);*