

Student ID: 1030889

Module: CM0343 – Individual Project

Credits: 40

Project ID: 125

Project Name: Online booking system for
After-School Childcare

Supervisor: David Walker

Moderator: Steven Shockaert

CM0343 – Individual Project

Final Report

David Manwaring

Abstract

This project could possibly change the way in which parents can monitor where their children go when school time comes to a close at the end of every day. My project started out with a design, where I looked into how I would solve this unsafe and unreliable method of children signing up to activities. Also there were potential problems with children not signing up for these activities and doing different things to what their parent/carer was thinking.

The problems that my project aims to resolve are removing hard copies of after school sign-up sheets in schools all around. The aim for the project was to have information regarding children and the activities was to all be stored in one secure place, the importance of changing the original method of children signing up was very important given the age of the child and possible consequences such as children going somewhere the parent/carer were unaware of.

The approach that I undertook was an online booking system, I would go on to create a website where parents/carers would login and view different activities that a school were offering and then sign their child up for the activity. My system would aim to deliver this protection over children where a parents and carers could register children for activities and parents having the satisfaction that that is where their child was. For the majority of the project I implemented in PHP and using a MySQL database to store all the information. To create all this I created prototypes of what each screen would look like on the system, this provided me with a general idea when coming to implementation of what the screen would look like and what functionality it would contain. I also spent time researching what similar systems were currently in existence and from what information I found in the field it gave me ideas of what I could use and improve upon when coming to creating the after school child care system.

The answer to carrying out this project is simple, it provides protection and convenience. Firstly, all data would be stored in a secure location which is the database, and the original convention of having hard copy signup sheets would be there for extinct. Secondly, it provides convenience, the original method would require finding a version of the hardcopy and then filling it in, whereas this method would require doing it from just the users own comfort and nothing less.

When creating the system there are of course going to be implications, firstly the problem of having signup sheets that can be lost and children not signing up for after school activities will be completely eliminated. The results will be shown and discussed into further detail in the report of why this is the case and potential drawbacks that this new convention may have.

Acknowledgements

I would like to first and foremost thank my supervisor David Walker for what a great job he has done this year in looking over me and my project. He has given me some great ideas and overall has been constructive on feedback given to me about the work I have produced.

Secondly, I would like to thank Steven Shockaert for the constructive criticism he has given on my initial plan and interim report. With the feedback he gave me it aided me to produce higher quality work and achieve heights in these reports.

Contents

Abstract.....	1
Acknowledgements.....	2
1.0 Introduction	6
1.1 Aims.....	6
1.2 Target Audience	6
1.3 Scope.....	6
1.4 Approach Taken	6
1.5 Important Outcomes.....	7
2.0 Background	8
2.1 Likely Stakeholders	8
2.2 Waterfall Model Constraints.....	8
2.3 Outstanding Systems	9
2.3.1 Contrasting Systems.....	10
2.3.1.2 Online School Booking	11
2.3.1.3 Activity Booking	12
2.4 Methods and tools	13
2.5 School and Parent Requirements	14
2.6 PHP.....	14
2.7 Short Round Up.....	15
3.0 Specification and Design	16
3.1 Specification.....	16
3.1.1 Core Functionality	16
3.1.2 Site Map	17
3.1.3 Web pages inside the system	17
3.1.5 Code Modules	25
3.2 Design.....	28
3.2.1 Flow charts.....	28
3.2.3 Use Case Diagrams.....	37
3.2.4 Activity Diagrams	40
3.2.2 Designs	45
3.3.3 CSS Framework	50

3.3.4 Database Design.....	51
4.0 Implementation	52
4.1 My method of implementation	52
4.1.1 Manual	52
4.1.2 Software	52
4.1.2 Method chosen	52
4.2 Page break down.....	53
4.2.1 Database script	53
4.2.2 User level check	54
4.2.2.4 Unforeseen problems	54
4.2.3 Header.....	54
4.2.4 Footer.....	56
4.2.5 About Us.....	58
4.2.6 User home.....	59
4.2.7 Activities page	62
4.2.8 Child sign up.....	64
4.2.9 Writing feedback.....	65
4.2.9.1 Description	65
4.2.9.2 External pages, functions and constants used.....	65
4.2.9.3 Code	65
4.2.10 Directions	66
4.2.11 Transactions	69
4.2.11.1 Description	69
4.2.11.2 External pages, functions and constants used.....	69
4.2.11.3 Code	69
4.2.12 Instant Messenger	69
5.0 Results and Evaluation	73
5.1 Results	73
5.1.2 Tests Conducted.....	74
5.2 Evaluation	75
6.0 Future Work	76
6.1 End of the project	76
6.2 Future work.....	76
6.2.1 Timetable to a PDF file	76

6.2.2 News Feed.....	77
6.2.3 Shopping Cart.....	78
6.2.4 Social Media.....	78
6.2.5 Mobile Application.....	79
7.0 Conclusions	81
8.0 Reflection	83
Bibliography	84

1.0 Introduction

1.1 Aims

The aim of the project is to design and implement an online system that parents can use to book their children in for after school child care. Along the way to achieving this aim, there are also several goals that I wish the project to reach. Firstly there should be a secure login system for users to login to access the system content. Secondly, there will be three different user categories which are parents, instructors and administrators each with their own privileges, some of which will be higher than others. Also, two of the user categories will be able to setup new activities where the other will be able to sign up their child for the activities. Finally, the Administrator, who is one of the user groups will have access to parent/carer contact information such as telephone number and email address should they need to contact them and will be able to create new instructors.

1.2 Target Audience

When designing and building something like this, there not just has to be a reason why but there needs to be an intended audience. The intended audience for this project is a small one, the people that will be using it the most are the parents/carers who will be registering their children for after school childcare and also the instructors who will be setting up the activity online. Also, since this new method will eliminate the use of generating hard copies, it will also save both time and paper, because it takes time for someone to create a hard copy saying what the activity is where and what time this will all be online and of course requires no paper. In time, it means that no one has to hunt down the whereabouts of where the hardcopy is and then record into a register who is attending the given activity.

1.3 Scope

When beginning this project, I already had a rough idea of what ground I will be covering and boundaries that I would have before starting. I knew I would be using two technologies that were the strongest of my skill set which are PHP and MySQL as well as HTML and CSS to complete all the mark-up and design for the project. Things such as client side scripting and libraries under client side scripting were outside of my comfort zone and were something I researched into.

1.4 Approach Taken

The approach that I chose for my project was the waterfall model, this wouldn't have been the case if it this project isn't controlled the way it is. For instance in the autumn I produced an initial plan that said a brief amount of what was to come followed by the interim report in the winter that unveiled a little more about the project. If the module hadn't laid out reports and plans like this I would have personally opted to start implementing straight away. This way it requires no build up to the implementation and you can get on straight away. However I have learnt that by carrying out the project in this way it has aided me to plan each part of the project. Areas of the project haven't been rushed and therefore the outcome is going to be a lot better. I am speaking strictly for the design phase of the project, this is where I was able to critically analyse what was within the scope of the project and decide:

- A. Is the feature going to be worth it?

- B. Is the feature relevant to the project? (Is it something that would be currently on the market, yet still able to make enough of an impact)
- C. Is the feature going to take so long that I overlook something more important.

1.5 Important Outcomes

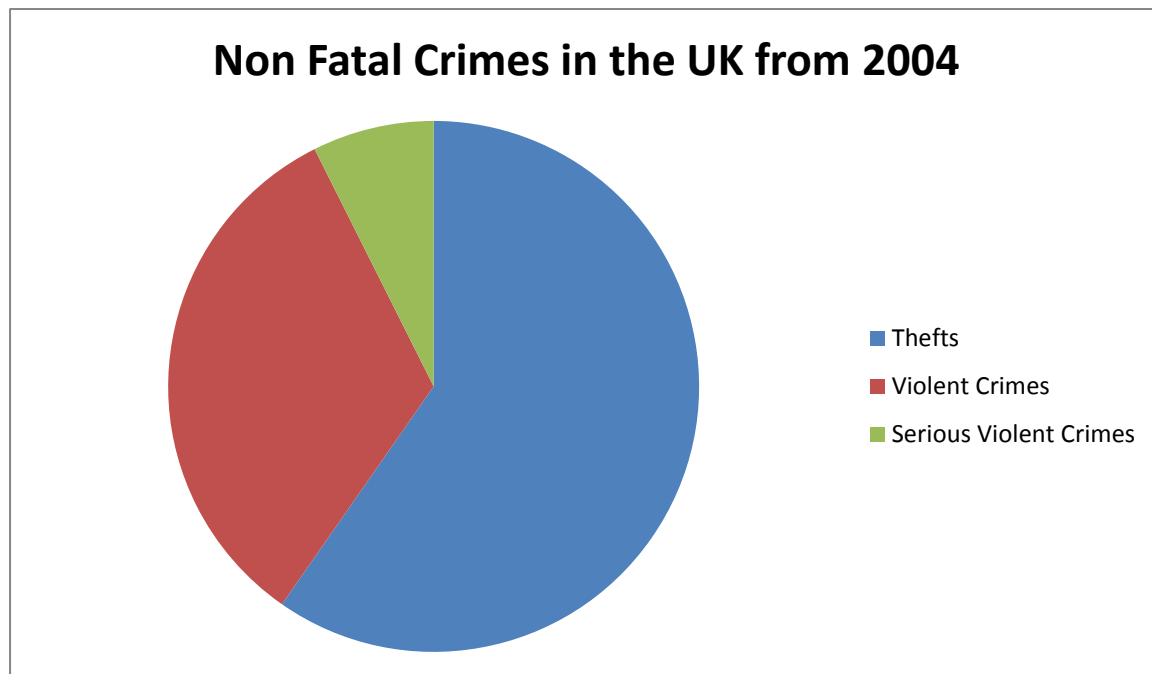
At the very beginning of the project I thought hard about what this project is going to give me personally. One thing was for sure and that was that I would further my knowledge with PHP and MySQL. I have my own personal experience with these two languages but I never really studied them in any real depth like I would be about to. Another outcome I wanted to achieve was a working system that had all features specified in the project briefing, I wasn't going to be satisfied with all features minus a few exceptions. Also, I wanted to learn more about different technologies that weren't currently in my skill set such as JavaScript, jQuery and AJAX. These were only technologies that I had heard about but nothing that I had any experience using. These were the outcomes I wanted to achieve for myself, for the project I wanted it to be something that was on the same quality scale as something currently on the market for this area. It would have been brilliant if I could use other features that other developers had and improved them to make my system better than what was available.

2.0 Background

This project is designed to aid children's safety, it's something that will help change after school activities everywhere and it's a system that is needed in today's schools. Every year children go missing, when home time comes around every day it's important that contact between child and parent is sharp and to have such a system in place can help this.

2.1 Likely Stakeholders

This system isn't something that could be used in a lot of different environments and work places; it is built for its purpose and therefore only has one kind of stakeholder. The only stakeholder that can be identified for this particular systems are schools, however it isn't built for a particular school. What I mean by that is that primary schools can use this system just as well as secondary schools. However, having conducted research into the safety of schools, by which I mean the surrounding area that schools are in there are some alarming statistics.



This research is official from 2004 based on 12-18 year olds; the figures for the graph are that in 2004 there were 863,000 thefts, 476,000 violent crimes and 107,000 serious violent crimes which are aggravated assault and robbery. These statistics take affect around end of school hours or before and during which can't be helped. But however with this system it helps parents monitor the safety of their children which is the point I am trying to get across. This system shouldn't be confused as being similar to "Big Brother" but it is there for protection. There are always risks of children going elsewhere without parents knowing, but with the system parents will know the whereabouts of their children. Of course instructors will provide feedback, if a child fails to attend, this is something that would be noted and reported to the parent.

2.2 Waterfall Model Constraints

The waterfall model has some drawbacks in regards to the way the project worked. In the requirements stage you can gather all the requirements from the client (supervisor) and of course any input of yours that you think will make a good addition too. However, as the

project went on I found myself having more and more ideas and a little too much time was spent in phases further down the line looking at these ideas into further detail and having to analyse:

1. What impact is this to have on the system?
2. Is this feature a valid feature?
3. Do I have enough time to implement the feature?
4. Is it too far outside the scope of the project?

So overall this part did add extra time onto the project. Also looking at these additional features and functions it meant that other features that were perhaps a little more important weren't receiving enough time spent.

Another drawback of this model is that "working software" isn't produced until the end of a single cycle. This means that when a first release was available, there were a number of improvements to be made which of course are errors that aren't accounted for in the implementation stage of the model. However the last stage of the model is maintenance which is where a lot of these errors and bugs are fixed.

2.3 Outstanding Systems

Back in late September and October I conducted further research into other systems that had been developed and are currently in place in schools, I was able to see what features they had, what kinds of information were recorded, this was done by looking at screen shots of their system and their registration form. The problem that I am addressing in this report has already been solved, there are outstanding systems. However, there would be a distinct feature that I would have in mine and that is a feature that produces directions from their current location to where the activity is if it were external to the school, such as an away fixture of a football match. I thought this would have been great to have as directions to places can be a challenge. By looking at other systems they do have the same core functionality to mine but I believe that if my system were to be put against these other systems mine would have a marginal head start in being sold. My system offers an instant messenger, all the user simply does is login and ask a question and this is a screen that is automatically updated inside the conversation window and this keeps the parents and the school in close contact

2.3.1 Contrasting Systems

Now I will talk and contrast between the system that I have developed and the outstanding systems on the market that I discovered during my research. I will be selecting one feature from my system and one from the opposing system.

I will begin like I did in the Interim report with:

2.3.1.1 Parents Booking

School Contact Details: Netmedia Lower Middle Grammar Upper School.
Phone: 01234 567890 Email: info@netmedia.co.uk

Parents' title* Parents' surname* *Required

E-Mail address Your email address is used to receive booking confirmations.

The system can only take bookings for one child at a time.
Please enter the details of the child you want to make bookings for below.

Child's DOB (dd/mm/yyyy)* Child's First Name*

Child's Surname* Child's Register Class*

Login

This is the parent's login page. As you can see there are a lot of fields that need to be correct in order to be authenticated.

The design of the page is very nice, and it's modern. The text fields are big and clear and it's also obvious for the user that the Login button will submit this form and process the login action.

MyECA Home About Contact Username Password Sign in

Please sign in

Username

Password

☐ Remember me

Sign in

MyECA © 2013 UK

This is my login page, it takes two values which are the username and password and nothing more.

Again, the design is very clear and it's clear where to click to process the login function.

To self-evaluate what I have done, it could be misleading on my design having two login forms on one page.

2.3.1.2 Online School Booking

Larry

Add list

Add list form

List name

Add list

Your lists

List name	Created	Privacy	Actions
After school club	30/10/2011 04:50:08	Private	Edit Delete Archive
Class trip returns	30/10/2011 04:49:35	Public (Can edit)	Edit Delete Archive
Homework 11/10	30/10/2011 04:44:59	Private	Edit Delete Archive
Late students 11/10	30/10/2011 04:44:16	Public (read only)	Edit Delete Archive
...

After school club

Student search

Surname

Click to add

Bloggers, Joe (Year 10)

Bloggs, Joe (Year 10)

Added so far - Click to remove

Student name	Ad No	Form	Date added
Bloggerson, Joe (Year 10)			30/10/2011 05:14:58

Larry

Class trip returns

Change list form

List name

Public list?

Private Public (can add & delete) Public (can view)

Change list

These three screen shots show attendees for a club, and a table that displays information on lists, which in my case for MyECA would be the activities page.

The colour coordination is a nice touché and it's similar to mine for admin to colour coordinate what activities have expired. Also, instead of having ID's for everything like mine they have three different options which are edit, delete and archive which is something I wasn't able to get working with my system.

Feedback

Admin

Upcoming Activities

Please note, once you have paid for an activity it can not be refunded due to our school policy.

Today's Date is: 2013-05-02

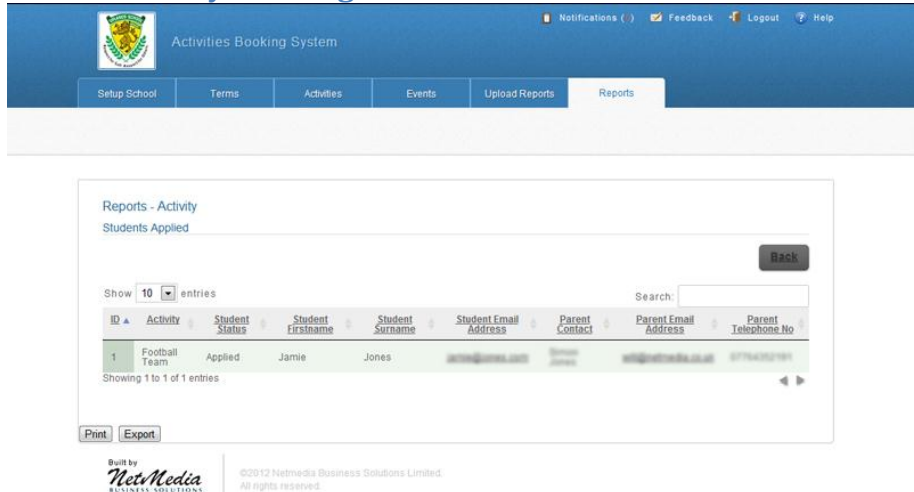
The time is now: 03:05

ID	Name	Description	Leader	Created	Student Capacity	Date (YYYY-MM-DD)	Posted By	Payment Required
18	Football Training	EXCLUSIVE! For one time only, come and train on the Millennium stadium pitch.	Sheldon Cooper	2013-04-09 15:57:40	14	2013-05-30	Admin	Buy Now
19	Swimming	Swimming lessons	Sheldon Cooper	2013-04-10 12:52:42	10	2013-04-30	Admin	Buy Now

As you can see, the options I have are to click on the ID for directions instead of perhaps having a link called "directions" and a remove button that is fully functioning.

Like the other design colour coordinating makes the user experience that little bit more enjoyable.

2.3.1.3 Activity Booking



The screenshot shows the 'Reports - Activity' page of the 'Activities Booking System'. The page has a blue header with navigation links: Setup School, Terms, Activities, Events, Upload Reports, and Reports. The 'Reports' link is active. Below the header, the page title is 'Reports - Activity' and the subtitle is 'Students Applied'. There is a 'Back' button in the top right corner. A search bar is located on the right side of the table. The table has columns: ID, Activity, Student Status, Student Firstname, Student Surname, Student Email Address, Parent Contact, Parent Email Address, and Parent Telephone No. The table shows one entry: ID 1, Activity Football Team, Student Status Applied, Student Firstname Jamie, Student Surname Jones, Student Email Address jamie@jones.com, Parent Contact Mr Jones, Parent Email Address m.jones@jones.com, and Parent Telephone No 01753 632181. Below the table, there are 'Print' and 'Export' buttons. The footer includes the NetMedia logo and copyright information: ©2012 Netmedia Business Solutions Limited. All rights reserved.

ID	Activity	Student Status	Student Firstname	Student Surname	Student Email Address	Parent Contact	Parent Email Address	Parent Telephone No
1	Football Team	Applied	Jamie	Jones	jamie@jones.com	Mr Jones	m.jones@jones.com	01753 632181

This is showing a report which is something I failed to get working due to time constraints. What this feature is on the opposing website is displaying information about an activity with an option to print it or to export the results to a PDF (this is an assumption).

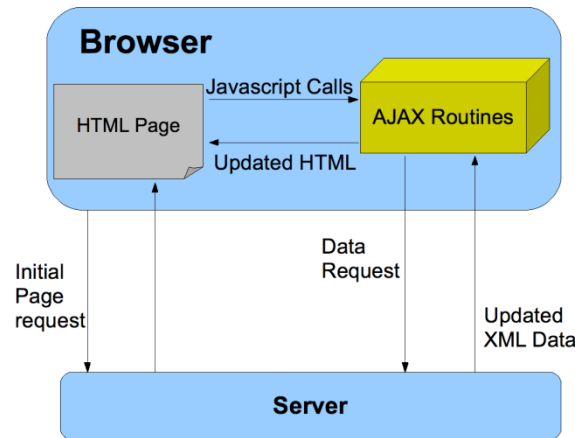
The functionality looks to be very good and is certainly something I wish I could have adopted in mine.

The design of this particular system is very nice; I like the tabular style because it's clear and it's something a bit different to what you would normally see.

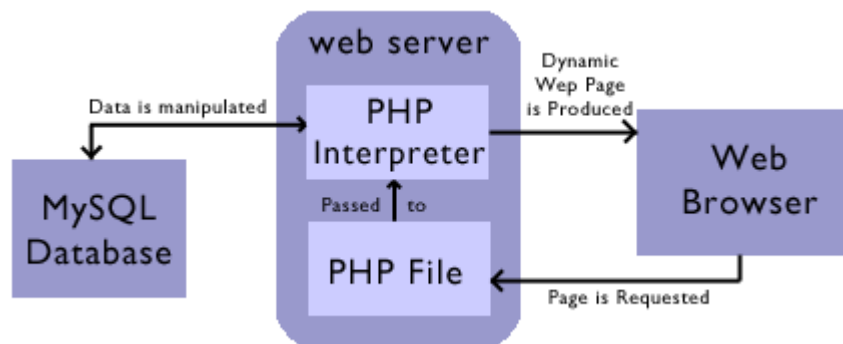
Unlike what I did with the other two, I can't provide a screenshot from the system I have developed because it does not exist but this could be used as a bar to reach for future developments if a developer was to come back to my code.

2.4 Methods and tools

The project consists of a lot of different technologies that glue together to build the system that is. The main technology used for page content is HTML and for design I have used CSS. One technology in particular is called AJAX. AJAX has been used to process and send requests without having to refresh the page the user is already on. Here is an illustration to aid what I am trying to explain.



For example, all the forms contain some sort of validation that checks user input. So in the browser JavaScript fetches the form data and posts it to a PHP script that then does the validation with the database and then returns a value back to AJAX and produces the feedback in the browser. This is also used for the one of my advanced features. Other than AJAX there is of course PHP which is something I will go on to talk about later in this report. PHP was used to request and process requests on the server side.



As you can see in the diagram above MySQL is a completely separate technology and entity. The PHP works to store data about parents, activities, payments, location, feedback and more. So having this technology allowed me to dynamically store content for the website and generate it too.

By using a combination of all these technologies I was able to design and build a good solution.

2.5 School and Parent Requirements

With a project like this when having to design, implement and then analyse there are requirements that need to be met. When designing the system it was important to consider requirements for parents, instructors/high level users such as Admin, basically the people who are at the school.

A key requirement is communication, if there were limited forms of communication inside the system it would be just as hard as the problem that the project is addressing. One of the requirements the feedback system that is in place, what this allows is for the instructor to leave comments about each individual student based on how they performed in a particular activity. What this does is help the school keep in regular close contact with parents that belong to the school and the parents are receiving regular updates (if the child is attending activities on a regular basis). Without this the only update the parent could receive would be from a parent evening, letter, phone call etc. but with this useful and productive feature this hindrance is therefore eliminated. Another feature is setting up the new activity, of course schools will want their parents to know when a new activity is being listed as they may find that their child will benefit and therefore try to encourage them to go, whereas before without the system the parents would not have known what extracurricular activities were available unless informed otherwise. Also with the payment system that is in place using PayPal Sandbox, it is now extremely beneficial to both schools and parents that transactions now take place online in a secure place. When looking at this part without the system you can clearly see that the most common method for parents to make a payment to the school was the child acting as a third person and transporting the money to school when leaving their house. However with the payment system in place this task now no longer needs to be accounted for. This means parents know where money is going and it means schools will know exactly when to receive payment and for it to be a little more reliable, by which I mean parents wouldn't know if they're using the money and spending it elsewhere.

On completion of this projected I had hoped that this would be a system that would set a benchmark that other systems should meet. My system has key characteristics like the instant messenger; directions and payment that completely extend what other systems are currently available on the market.

2.6 PHP

The main technology in this system is PHP, in particular I am using PHP5, it's the latest version of the language that has been released, there are some new features and functions in the language, for example a new supported function is "mysqli_connect()" that the creators of PHP are aiming to replace the original "mysql_connect()" function. There are other functions like this which are newer and more efficient than the older ones.

(PHP, 2013)

The last thing to mention too is that on the Object Oriented Programming side of things which is only something I have grazed the top of is that for properties and functions you can set the visibility of them using public, private and protected which are all new to PHP5. For additional information on this version of the language please see my references section of the report. A specialised technology that is incorporated to the project is MySQL which is an open source database management system that I am using. This database management

system is one of the most widely used database system in the world. Inside of which I am storing all the records about the systems users and activities etc.

2.7 Short Round Up

To finish the background information of my project, the problem that I am trying to solve and also eliminate is to remove hard copy signup sheets that children have to sign up for by themselves. This is a problem that exists in schools all over. What I aim for this project to achieve is:

- For all information to be stored in one secure place,
- Payments are also secure and are guaranteed to go directly to the school
- An increase in the protection of children due to parents signing up children for them.

To be able to fully clarify that I have achieved the project aim, a sufficient amount of testing will need to be conducted and also I will have to compare my system against current systems that are available on the market and analyse those once again to ensure that I have included the same features and also to make sure I haven't underachieved on any aspects. Finally, it I will need to look at how long this sort of system would last in a school, for example schools may not want to have this sort of system in place permanently and only have it on a trial basis, other than that is there enough beneficial functionality to the school for them to want it?

3.0 Specification and Design

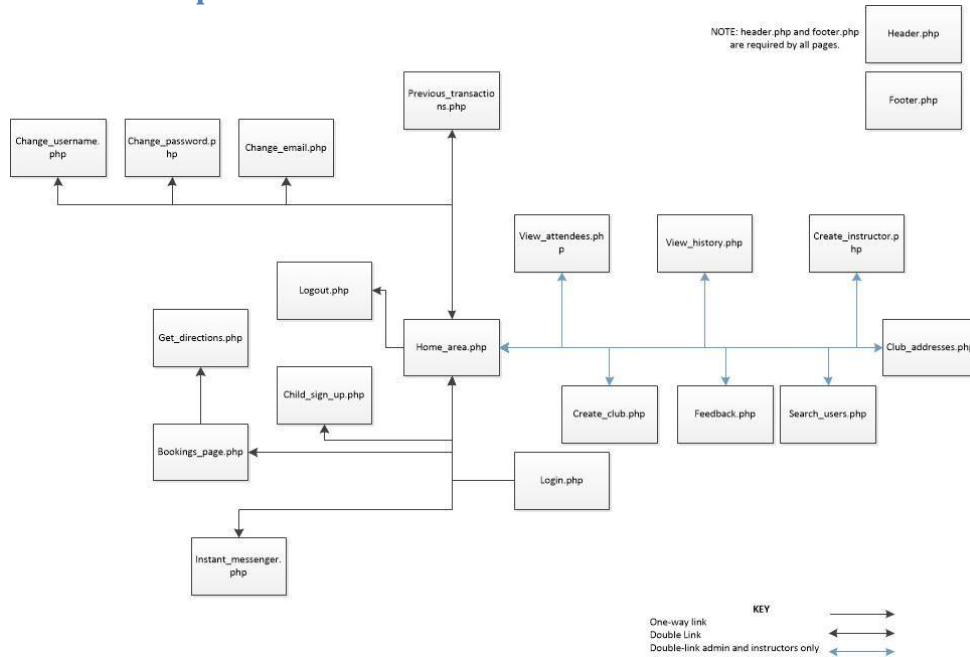
3.1 Specification

3.1.1 Core Functionality

Like with any system there are always a set of core functionalities of what it is detrimental to make it what it needs to be. For example to have a conversation with someone, eyes and ears are absolutely necessary, and legs would be an added extra. This concept is similar for software engineering; you need to break down what is absolutely essential to solve the problem you want to solve. The system that I have developed is a booking system and these are its core functionalities and why:

- Login system: the login system plays as a key part, what it allows is for users to login with their own login credentials to access the exclusive content inside the booking system. If this feature were to be removed then it wouldn't affect it "not being an online booking system" but it would mean anyone could come on to the domain and access the content and see the data about students and activities which is something that quite clearly cannot be aloud. By having this login system in place though it keeps the data and its users safe.
- Signing up a child: This is core functionality for the system, in order for the system to work there needs to be a method in which a parent (user) can login and then sign up their child to an extracurricular activity. Of course without this feature it wouldn't be a "booking system" and is a functionality that has to be in the system for it to work.
- Student feedback: The final core functionality that I have selected is for the instructor (coach) to write feedback about the student and how they performed, this can include information such as progress made, things that need to be improved, or any problems that the child may have faced during the activity. This is a core functionality because as one of the requirements is for parents to receive feedback about the student, it needs to be in place. Otherwise without the feature students will be attending activities but the parents not receiving any feedback and not knowing things like was it worth the student going and if it's an activity that requires a fee is this something that is going to be of a regular occurrence in the future.

3.1.2 Site Map

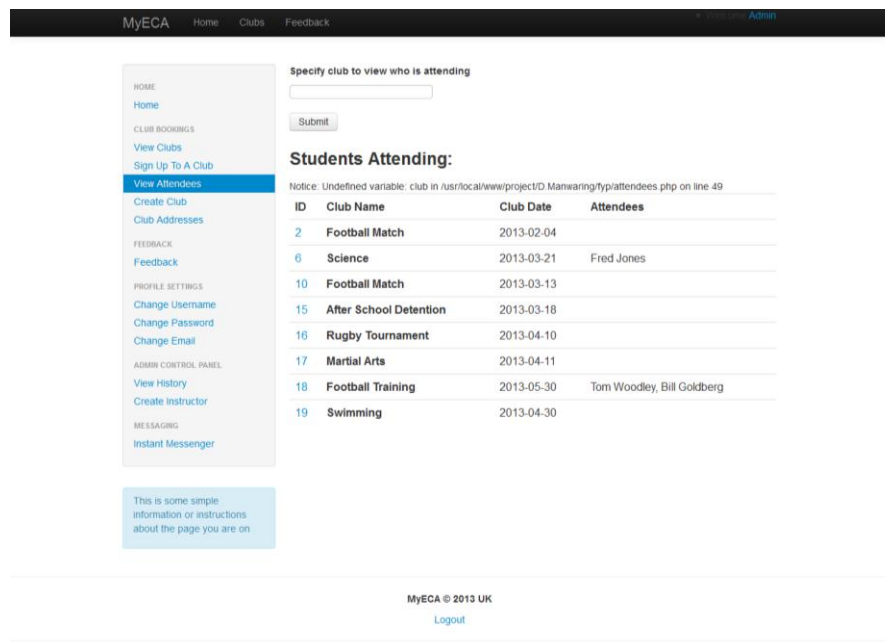


This is a map of the system; I created this to help show how everything was going to piece together and connect. As you can see by looking towards the centre of the diagram the page “home_area.php” is the main page where all of the content can be accessed from. I have also represented what users like Admin and Instructors can access by changing the colours of the connectors between these represented pages. Also, some of the links are one way in which I am trying to get across is that once the user traverses to a page they can’t navigate back like you can with the other pages. For example with the “Login.php” once you’re logged in the user just go back to that page without having to logout.

3.1.3 Web pages inside the system

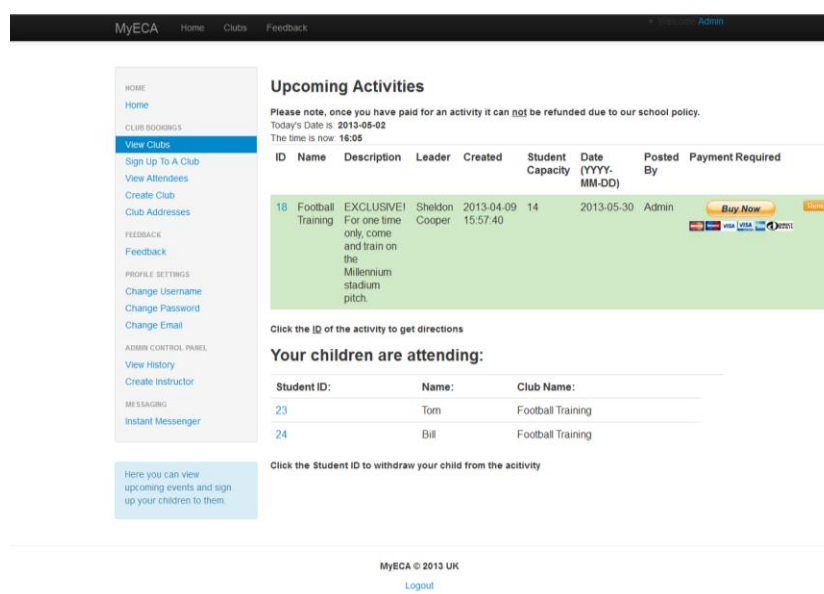
In this section of the report I will write about the different web pages that are currently in existence inside the system and what their functionality is. NOTE: all these pages have got a header and footer that contain navigation for the users to work their way around the system.

1. Attendees - on this page Instructors and Admin users can view which students have been signed up for what clubs. It shows the ID of the activity, the name, and which students. When the user clicks on the ID of the activity it takes them to a feedback page where it shows a data for what you just selected along with the student ID's. From here the user enters the student ID and what feedback they would like to write regarding that student.



This is what the page attendees looks like. Where there are blank values under the column attendees it means that no one has yet registered for that activity.

2. Bookings page - this is the page that shows all current activities that are available for parents to sign their children up to. It shows the ID of the activity, the club name, the time and date and shows if it's a activity that requires payment also. When the user clicks the ID of an activity at this stage they are directed to another page that shows them directions to the activity based on where they are logged in. Further down "bookings_page.php" there is some more information that is provided to the parent, they can see what activities that they're child is signed up for already and they also have the option to remove them. If the user is logged in as an admin they can see activities that are passed the current date and can remove them, these are highlighted in red.



3. Child Sign Up - On this page parents can register their child for an activity, it's a very simple form that has a drop down of the different clubs that are currently available and then they enter the name of their child this is then recorded into the database.

MyECA Home Clubs Feedback Admin

HOME
Home

CLUB BOOKINGS
View Clubs
Sign Up To A Club
View Attendees
Create Club
Club Addresses

FEEDBACK
Feedback

PROFILE SETTINGS
Change Username
Change Password
Change Email

ADMIN CONTROL PANEL
View History
Create Instructor

MESSAGING
Instant Messenger

This is some simple information or instructions about the page you are on

Sign up your child here

Step 1. Select the child you want to sign up.

Bill Goldberg

Step 2. Choose the club you want to sign your child up for.

Football Training on 2013-05-30

Football Training on 2013-05-30

Sign Up

MyECA © 2013 UK
Logout

4. Feedback - this is the screen that allows instructors to leave feedback to students based on how they thought they performed in a particular activity.

MyECA Home Clubs Feedback Admin

HOME
Home

CLUB BOOKINGS
View Clubs
Sign Up To A Club
View Attendees
Create Club
Club Addresses

FEEDBACK
Feedback

PROFILE SETTINGS
Change Username
Change Password
Change Email

ADMIN CONTROL PANEL
View History
Create Instructor

MESSAGING
Instant Messenger

This is some simple information or instructions about the page you are on

Feedback Form

Student ID:

Activity

feedback here ...

Save

Club Name	Student ID	First Name	Second Name
Football Training	23	Tom	Woodley
Football Training	24	Bill	Goldberg

MyECA © 2013 UK
Logout

5. Create Instructor – this page allows Admin users only to create new instructors (coaches) to the system. Typically in the real life situation these would be teaches, the system just recognises them as “Instructors”. Once they are created they then have their own set of privileges on the system such as creating a club and leaving feedback to a student.

MyECA Home Clubs Feedback Welcome Admin

HOME
Home
CLUB BOOKINGS
View Clubs
Sign Up To A Club
View Attendees
Create Club
Club Addresses
FEEDBACK
Feedback
PROFILE SETTINGS
Change Username
Change Password
Change Email
ADMIN CONTROL PANEL
View History
Create Instructor
MESSAGING
Instant Messenger

Title:
 First Name:
 Last Name:
 Subject:
 Username:
 Password:
 Birthday:
 Gender: Male ☐ Female ☐
 Contact:
 Email:
 Register Clear

Add a new instructor to the system with the Create An Instructor feature

MyECA © 2013 UK
Logout

6. Change username, password and email – these are all profile options for all the users on the system, it allows them to update the information that I hold on the system.

MyECA Home Clubs Feedback Welcome Admin

HOME
Home
CLUB BOOKINGS
View Clubs
Sign Up To A Club

Change of username
 New Username:
 Change Clear

MyECA Home Clubs Feedback Welcome Admin

HOME
Home
CLUB BOOKINGS
View Clubs
Sign Up To A Club
View Attendees
Create Club

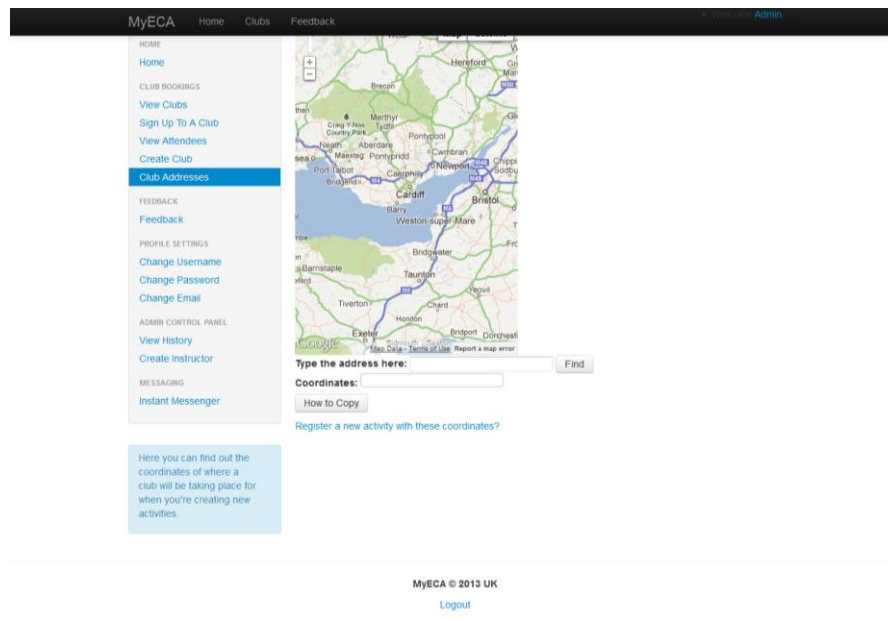
Change of password
 Old Password:
 New Password:
 Confirm Password:
 Change Clear

The screenshot shows the 'Change of email address' page. At the top is a dark navigation bar with 'MyECA' and links for 'Home', 'Clubs', and 'Feedback'. On the right of the bar, it says 'Welcome Admin'. On the left is a sidebar menu with 'HOME' (containing 'Home' and 'View Clubs') and 'CLUB BOOKINGS'. The main content area has the title 'Change of email address' and a form with a label 'New Email Address:' followed by a text input field. Below the input field are two buttons: a blue 'Change' button and a red 'Clear' button.

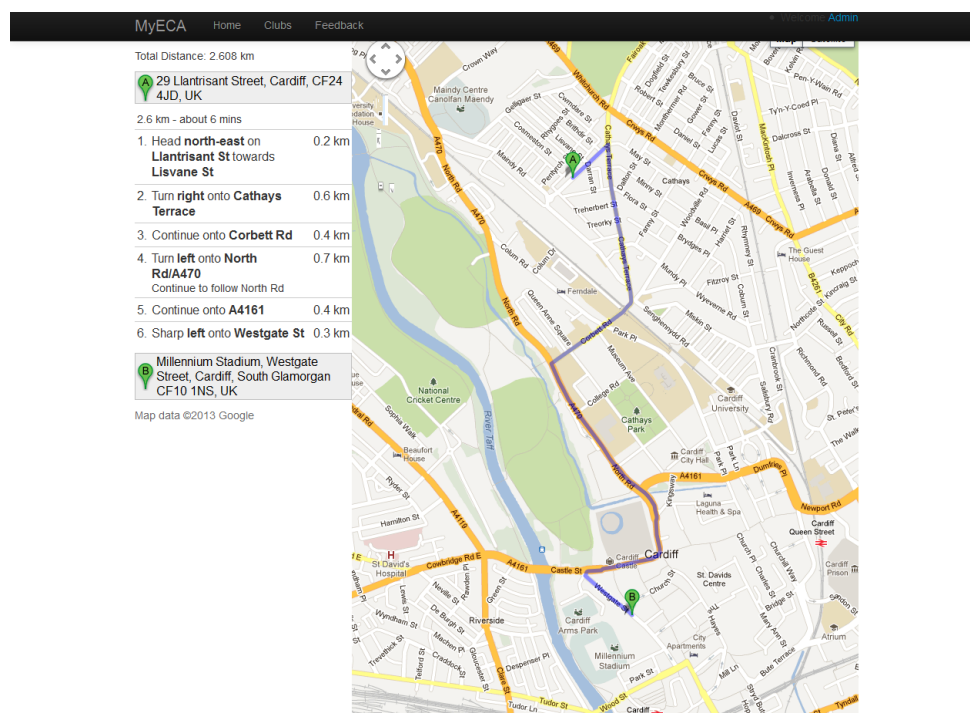
7. Enter payments – this is an administrator only feature; it requires the administrator to have access to the PayPal account which is not something that should be handed out to all the instructors. On this screen they can enter who has paid for activities, the transaction ID and it stores a timestamp of when they carried out the transaction.

The screenshot shows the 'Enter Customer Payments' page. The top navigation bar is identical to the previous page. The sidebar menu on the left is more extensive, including 'CLUB BOOKINGS' (with 'View Clubs', 'Sign Up To A Club', 'View Attendees', 'Create Club', and 'Club Addresses'), 'FEEDBACK' (with 'Feedback'), 'PROFILE SETTINGS' (with 'Change Username', 'Change Password', and 'Change Email'), 'ADMIN CONTROL PANEL' (with 'View History', 'Create Instructor', and 'Enter Payments' which is highlighted), and 'MESSAGING' (with 'Instant Messenger'). The main content area has the title 'Enter Customer Payments' and a note: 'Please note, for this you will need access to the MyECA PayPal account'. Below the note are four input fields: 'User ID:', 'Transaction ID:', 'Amount:', and 'Date of transaction:' (with a placeholder 'YYYY-MM-DD'). At the bottom of the form are two buttons: a blue 'Process' button and a red 'Clear' button. Below the form is a light blue box containing the text: 'This is some simple information or instructions about the page you are on'. At the bottom of the page, there is a footer with 'MyECA © 2013 UK' and a 'Logout' link.

8. Geocode Service – this is the page that instructors or admin access to get the addresses of activities that are external to the school. What they do here is type in the address of the club and a set of coordinates are returned to the user. These coordinates are then entered into the screen that adds a new activity.



9. Get Directions – when a parent is on “bookings_page.php” they can see the ID of an activity, when they click this ID they are then redirected to this page, on here it provides directions to the activity based on their location. It shows using Google Maps with two pointers that are start and finish and it’s connected with the route that Google has provided. Down the left side of this made are the directions from start to destination.



10. Home Area – this is the main page users see when they login to the system. On this page contains a form where parents can register their child to the system and view what children are already registered on the system.

The screenshot shows the MyECA Home Area. At the top is a navigation bar with 'MyECA', 'Home', 'Clubs', 'Feedback', and a user profile 'Admin'. On the left is a sidebar menu with categories: HOME (Home, My Previous Transactions), CLUB BOOKINGS (View Clubs, Sign Up To A Club, View Attendees, Create Club, Club Addresses), FEEDBACK (Feedback, Admin Feedback), PROFILE SETTINGS (Change Username, Change Password, Change Email), ADMIN CONTROL PANEL (View History, Create Instructor, Search Users), and MESSAGING (Instant Messenger). The main content area is titled 'Welcome to the MyECA Home Area' and contains a 'Register Child' form with fields for First Name, Last Name, and School Year, and buttons for 'Register' and 'Reset'. Below this is a 'Registered Children' table:

First Name	Last Name	School Year	Date/Time
Tom	Woodley	11	2013-02-22 19:58:20
Bill	Goldberg	11	2013-02-22 19:58:36
Jamie	Facey	11	2013-03-06 14:05:55

Below the table is a 'Register your location' section with a text input, a 'Find' button, and a 'Save' button. At the bottom of the sidebar, a message says 'Welcome to the home area of MyECA!'. The footer of the page reads 'MyECA © 2013 UK'.

11. Instant Messenger – this is where all questions can be asked Live to the people at the school, the aim of this page is for people that don't want to wait for a response to get a quick response about queries they may have about activities.

The screenshot shows the MyECA Instant Messenger interface. The top navigation bar is the same as the previous page. The sidebar menu is identical. The main content area is titled 'Ask a question Live to our MyECA Team' and features a chat window. Above the chat window is a link '52Enter chat?'. The chat window shows a list of messages: 'bloggssays: Hello yourself' and 'Adminsays: Hello!!'. Below the chat window is a text input field with the placeholder 'Type and hit return' and a 'Send' button.

12. My Transactions – this is the page where parents can see what activities have been purchased, this is not automatically updated when the user has completed a transaction as this isn't a supported feature. An admin would update the users information and there is a small notice at the top the page to pre-empt the user that it can take up to one working day for this information to be updated.

My Previous Transactions

Note, it can take up to 1 working day for your up to date transactions to appear.

ID:	Transaction ID:	Amount (£):	Author:	Paid:
4	ID6TD47729X1856374X	5.00	Admin	2013-04-12

This is some simple information or instructions about the page you are on

13. Search – this is a feature that allows instructors and admin to search for information about user's on the system.

Search MyECA

Search Database

Results: 2 result(s) found for David

ID: 52

First Name: David
Last Name: Manwaring
Username: Admin
Birthday: 2013-01-26
Gender: Male
Contact: 0
Email: ManwaringD@cardiff.ac.uk
Join Date: 2013-02-22 00:00:00
Successful Logins: 73
Login Attempts: 76
Online: yes

ID: 53

First Name: David
Last Name: Walker
Username: dwalker1
Birthday: 2013-02-22
Gender: Male
Contact: 234832423
Email: David.WWalker@cs.cardiff.ac.uk
Join Date: 2013-02-22 00:00:00
Successful Logins: 1
Login Attempts: 2
Online: yes

Search the users inside the MyECA database

MyECA © 2013 UK
Logout

14. View History – this is another exclusive admin only page, this page shows all the records of user activity on the system, it shows information such as the username, what page they were on, what IP address and what time.

Notice: A session had already been started - ignoring session_start() in C:\xampp\htdocs\Project\config.php on line 2

MyECA Home Clubs Feedback Welcome Admin

HOME
Home

CLUB BOOKINGS
View Clubs
Sign Up To A Club
View Attendees
Create Club
Club Addresses

FEEDBACK
Feedback

PROFILE SETTINGS
Change Username
Change Password
Change Email

ADMIN CONTROL PANEL
View History
Create Instructor

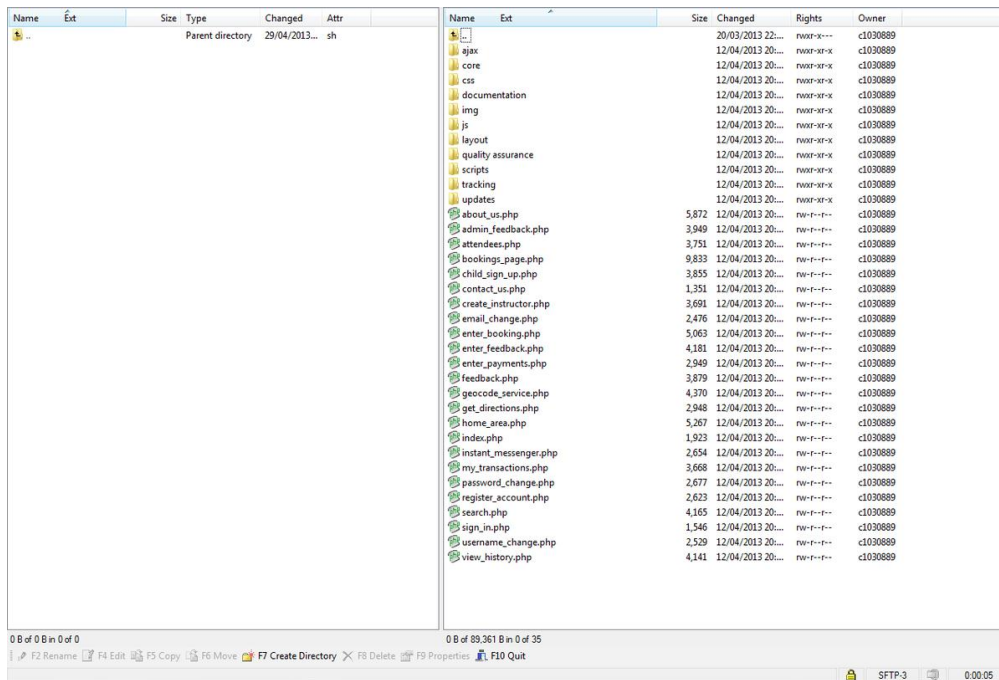
MESSAGING

User History of MyECA

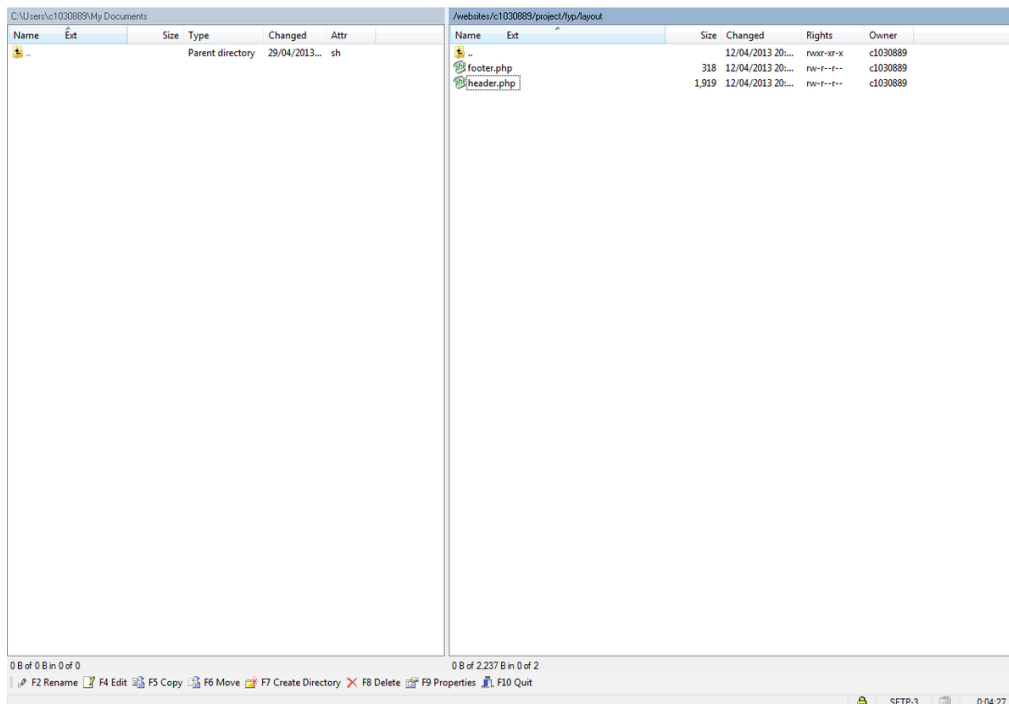
ID #	Username	IP Address	Date/Time	Referrer	Internet Browser
1	Admin	10.72.2.27	2013-04-09 12:03:13	http://project.cs.cf.ac.uk/D.Manwaring/fyp/bookings_page.php	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:13.0) Gecko/20100101 Firefox/13.0.1
2	Admin	10.72.2.27	2013-04-09 12:03:23	http://project.cs.cf.ac.uk/D.Manwaring/fyp/bookings_page.php	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:13.0) Gecko/20100101 Firefox/13.0.1
3	Admin	10.72.2.27	2013-04-09 12:04:39	http://project.cs.cf.ac.uk/D.Manwaring/fyp/bookings_page.php	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:13.0) Gecko/20100101 Firefox/13.0.1

3.1.5 Code Modules

In my system I have reused code to improve how efficient the system runs. It will run more efficiently because instead of having the same code repeated in all the files, instead just have it one file and use one line of code to call it is a lot more efficient. Also, in future developments there could be a change in what content is displayed or even in design. If this code was not dynamically used and was completely static etc. it would mean having to go into each file that has the code in and manually update it. Whereas where I have included it all that would be required is to change it once and in general it's a better programming practice.



This is a print screen of the root folder inside WinSCP for the system. As you can see everything is well divided into sections to help organise code and keep things separate, also helps me not overwrite files if I want to call it something the same as something else or very similar to.



Each page that contains user page content includes two files which are “header.php” and “footer.php”. The header file contains the navigation bar that runs along the top of the page. The footer contains the link to logout of the system and terminates the user’s session also along with the MyECA copyright.

Other than reusing code to display page content, I have code that is reused to perform functions and do things in the background without the user knowing that they're happening. All of this is inside my “scripts” folder if you see the root directory at the beginning of the code modules section of the report. This is what the contents of the scripts folder looks like:

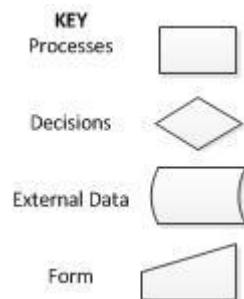
Name	Ext	Size	Changed	Rights	Owner
db		12/04/2013 20:...	12/04/2013 20:...	rw-r--r--	c1030889
config.php	.php	471	12/04/2013 20:...	rw-r--r--	c1030889
functions.php	.php	1,012	12/04/2013 20:...	rw-r--r--	c1030889
scr_add_club_type.php	.php	248	12/04/2013 20:...	rw-r--r--	c1030889
scr_create_instructor.php	.php	1,049	12/04/2013 20:...	rw-r--r--	c1030889
scr_email_change.php	.php	669	12/04/2013 20:...	rw-r--r--	c1030889
scr_fetch_bookings.php	.php	2,885	12/04/2013 20:...	rw-r--r--	c1030889
scr_fetch_user_level.php	.php	579	12/04/2013 20:...	rw-r--r--	c1030889
scr_get_location.php	.php	447	12/04/2013 20:...	rw-r--r--	c1030889
scr_log_activity.php	.php	593	12/04/2013 20:...	rw-r--r--	c1030889
scr_login_script.php	.php	1,745	12/04/2013 20:...	rw-r--r--	c1030889
scr_logout.php	.php	519	12/04/2013 20:...	rw-r--r--	c1030889
scr_password_change.php	.php	555	12/04/2013 20:...	rw-r--r--	c1030889
scr_payments.php	.php	744	12/04/2013 20:...	rw-r--r--	c1030889
scr_process_activity.php	.php	713	12/04/2013 20:...	rw-r--r--	c1030889
scr_process_booking.php	.php	1,049	12/04/2013 20:...	rw-r--r--	c1030889
scr_register.php	.php	993	12/04/2013 20:...	rw-r--r--	c1030889
scr_register_child.php	.php	551	12/04/2013 20:...	rw-r--r--	c1030889
scr_submit_feedback.php	.php	471	12/04/2013 20:...	rw-r--r--	c1030889
scr_username_change.php	.php	591	12/04/2013 20:...	rw-r--r--	c1030889

Inside of the “db” folder are my database backups for previous versions of the system and in case of any data loss that the system could potentially have. For an example I will use “scr_check_user_level.php”. This file is included in all of the pages, what this allows is on each page it checks what user level the user is, based on that result it will output information (privileges) based on what that user is allowed to see. So if the user level is set to 3, it identifies the user as an admin and shows admin only options on top of everything else. Again, if updates are needed to be made to the code it is only done once instead of several as the code only exists in one place and also reduces the amount of redundant work that could exist if the code existed in one big file all together.

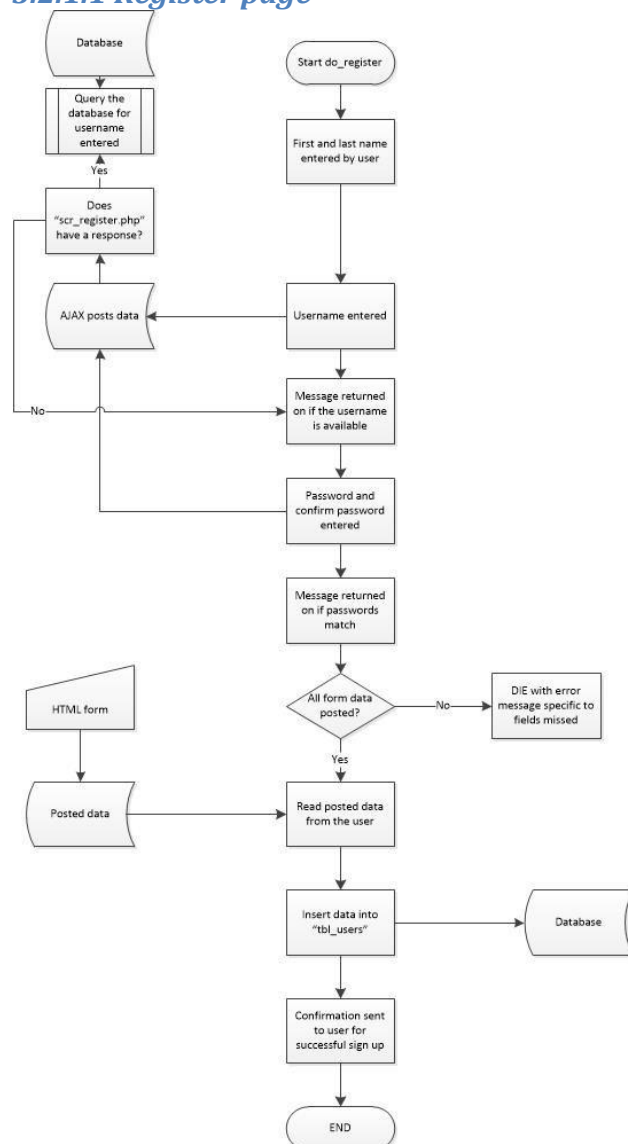
3.2 Design

3.2.1 Flow charts

In this part of the report I am going to represent how data travels through the system. I have done this for all characteristics of the project. Here is a key to be able to differentiate between the different boxes in my diagrams:



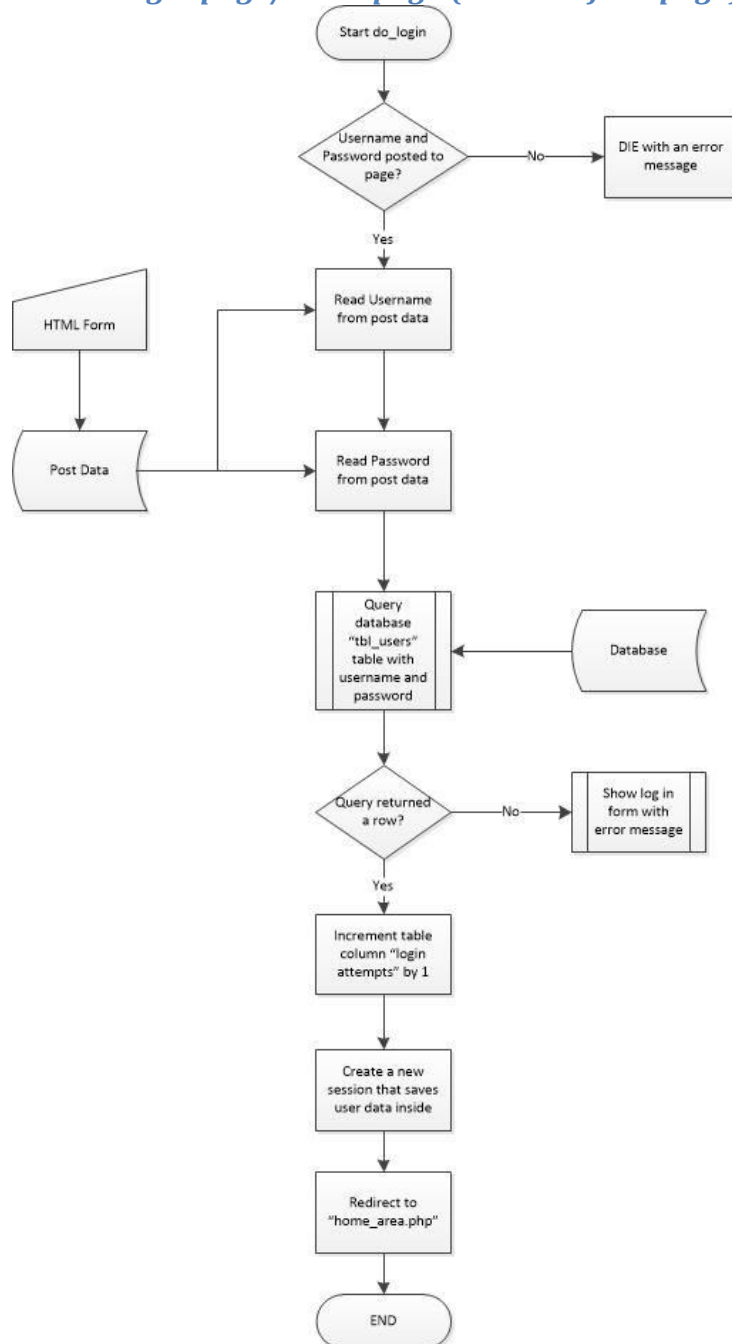
3.2.1.1 Register page



These are the stages that are carried out when a user registers for an account to the system. The only part to mention here is that where the username is entered, AJAX automatically

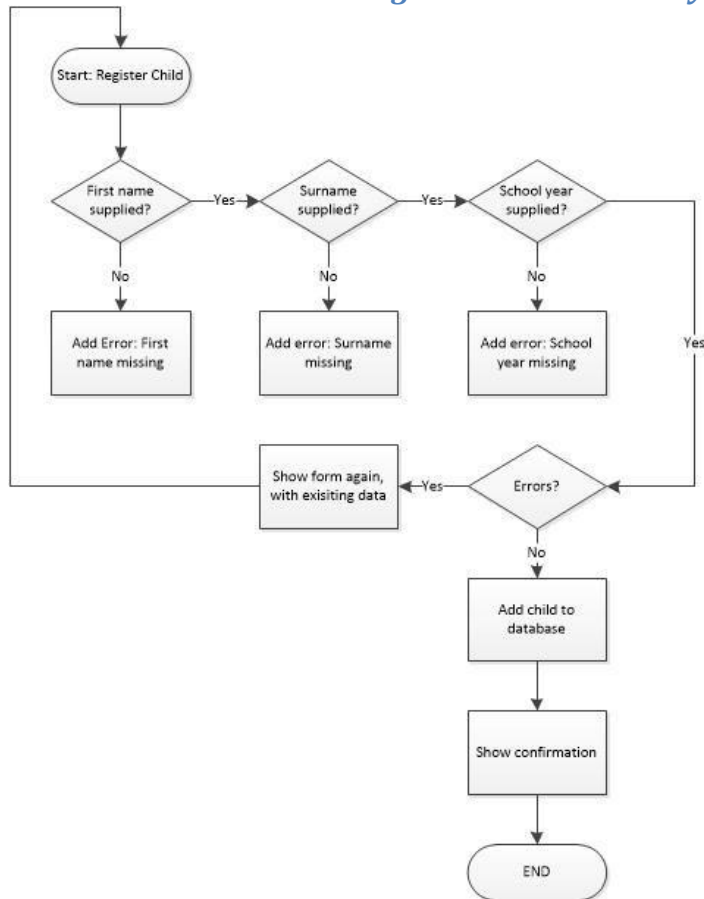
gets this data and sends a request to a .PHP file to check if the username is available and passes a message back to say whether it is or not.

3.2.1.2 Login page/Index page (header of the page)



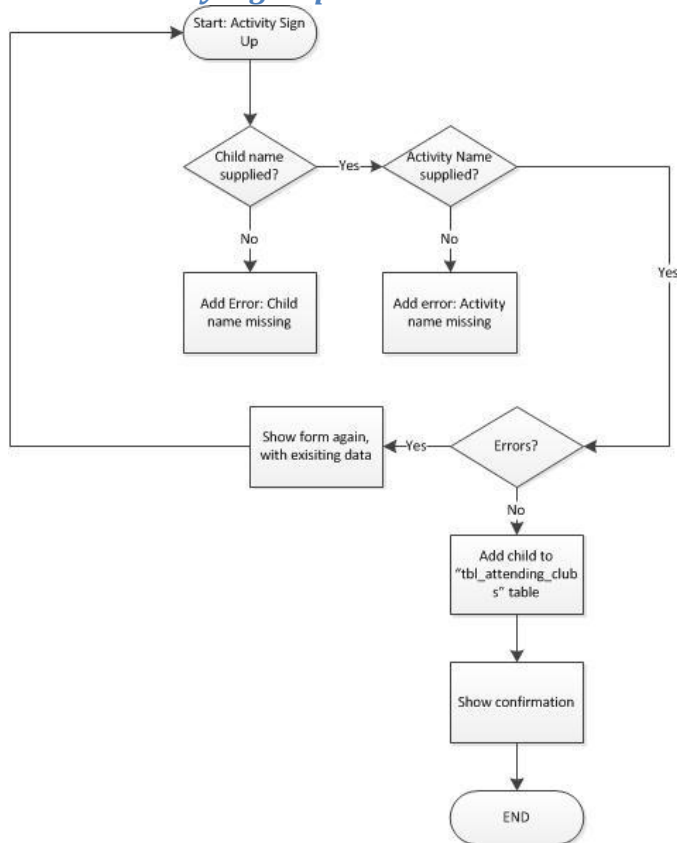
This how data flows through the system for a user to login to access the main content of the system. As you can see the data has to go through a number of checks and a couple of decisions are made before the user can be logged in successfully.

3.2.1.3 User Home Area: Register a child to the system

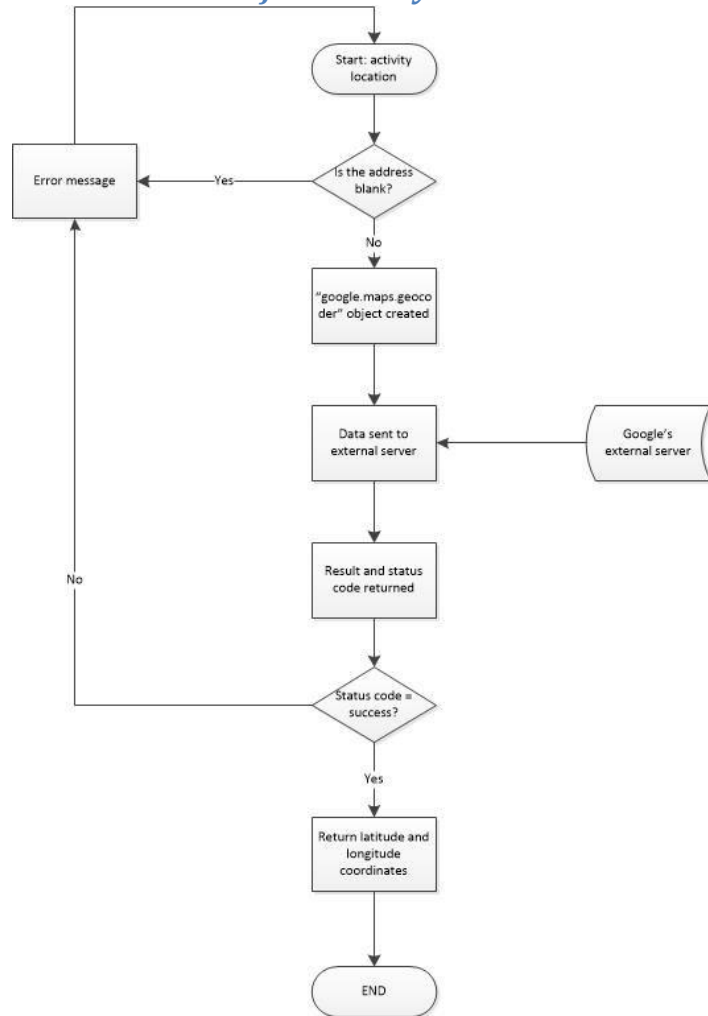


This part of the page is concerned with registering a child to the system. By looking at this diagram you can see that when a user is entering the data, it is checking to see if a first name, second name and school year have been supplied, or it will add an error. When doing the final check it's asking if there were any errors with what has happened, it will show the form again with errors specific to the fields they have incorrectly filled out. Otherwise the child is successfully registered.

3.2.1.4 Activity Sign Up

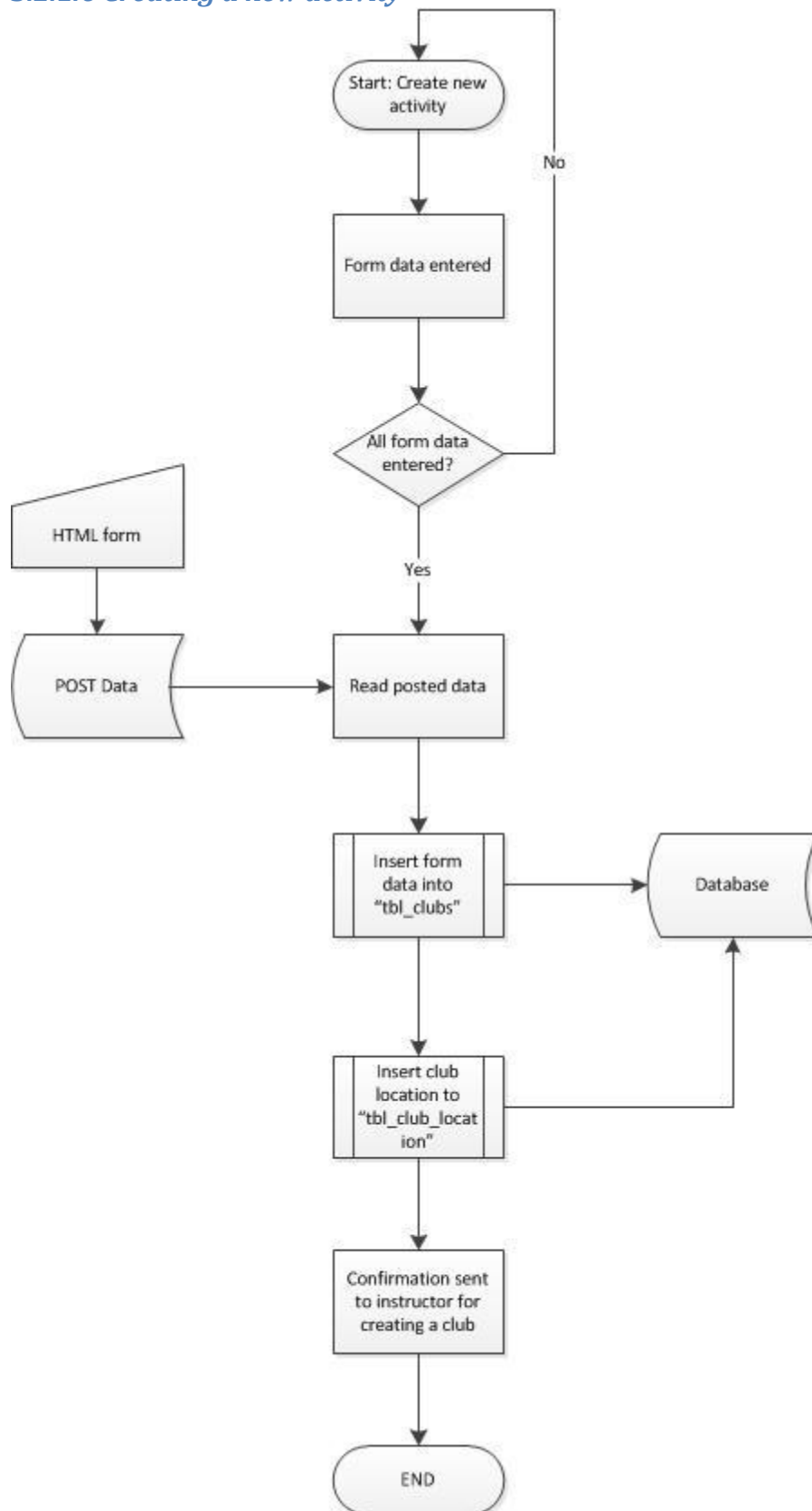


This is how data is processed once it is submitted at the front end of the system. For this diagram there are two decisions made at the very beginning of the diagram. I have put these here because it's important to check that values are going to be passed through to sign up for an activity, otherwise a booking would exist that contained no or incorrect information.

3.2.1.5 Location of an activity

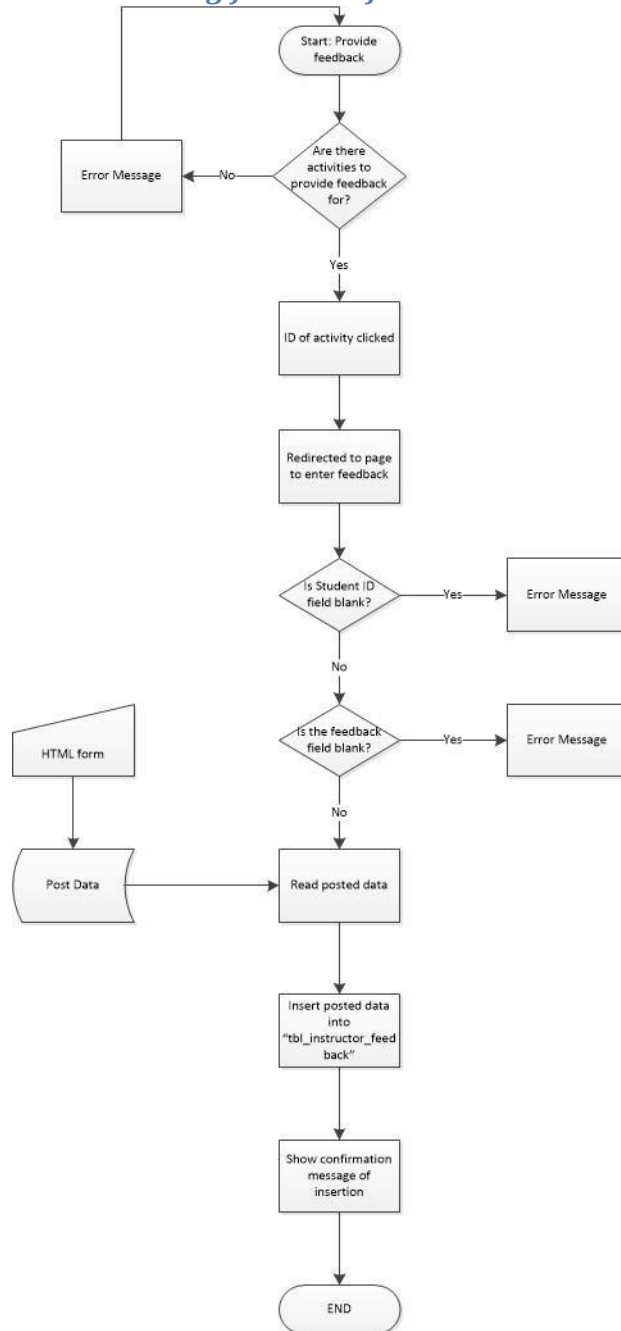
This is a flow chart to illustrate what happens when a request and a call back is sent through Google's Map API. With this I am trying to illustrate that the data is passed to Google, processed and returned. This is different from other diagrams because the majority of the information that is passed through the system doesn't get sent off to external data sources.

3.2.1.6 Creating a new activity



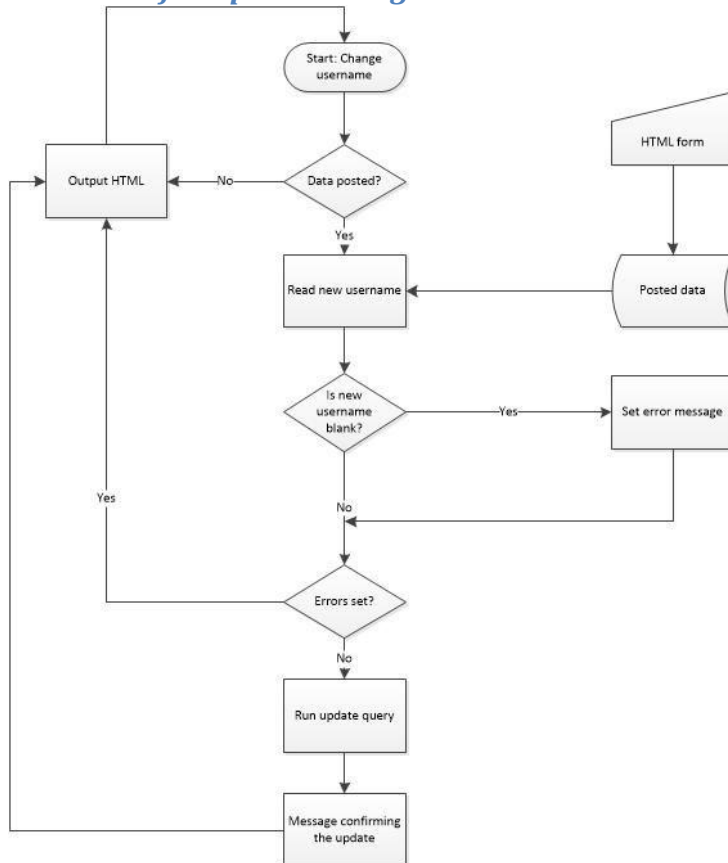
This is a series of events of what happens to data that is entered in the front end of the system and then stored into the database on the back end.

3.2.1.7 Writing feedback for a student



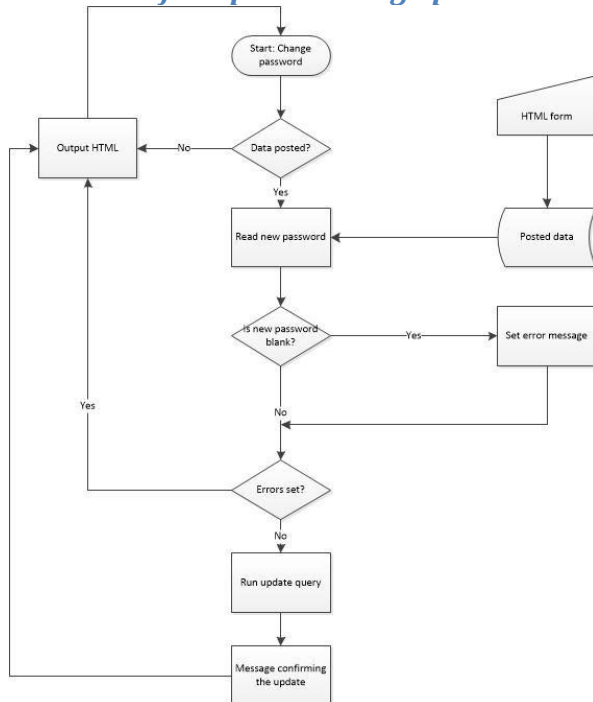
This is a flowchart to show the series of events that data goes through before being stored into the database.

3.2.1.8 Profile option: change username



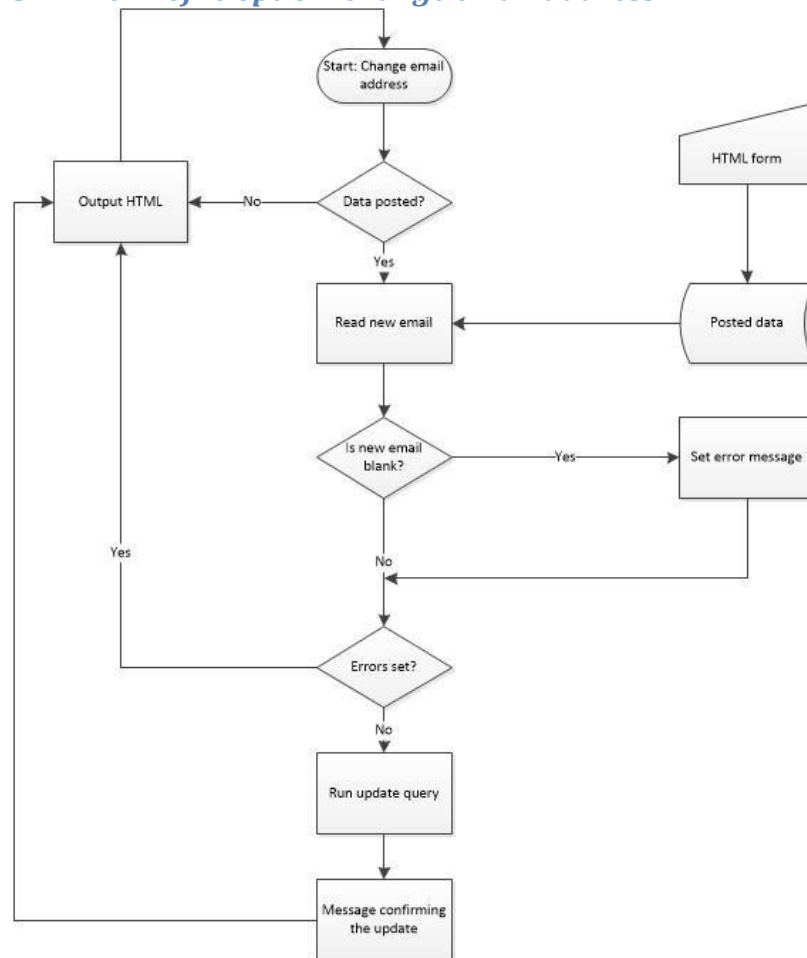
This is another flow chart of how data is processed and then updated when the user changes their username.

3.2.1.9 Profile option: change password



This is another flow chart of how data is processed and then updated when the user changes their password.

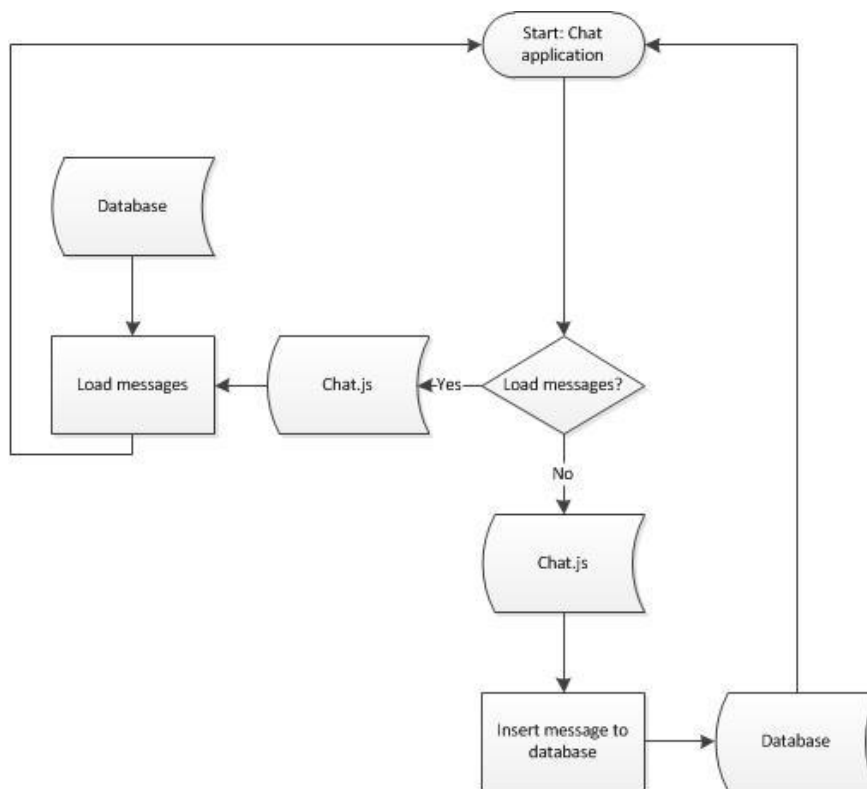
3.2.1.10 Profile option: change email address



This is another flow chart of how data is processed and then updated when the user changes their email address

3.2.1.11 Chat Application

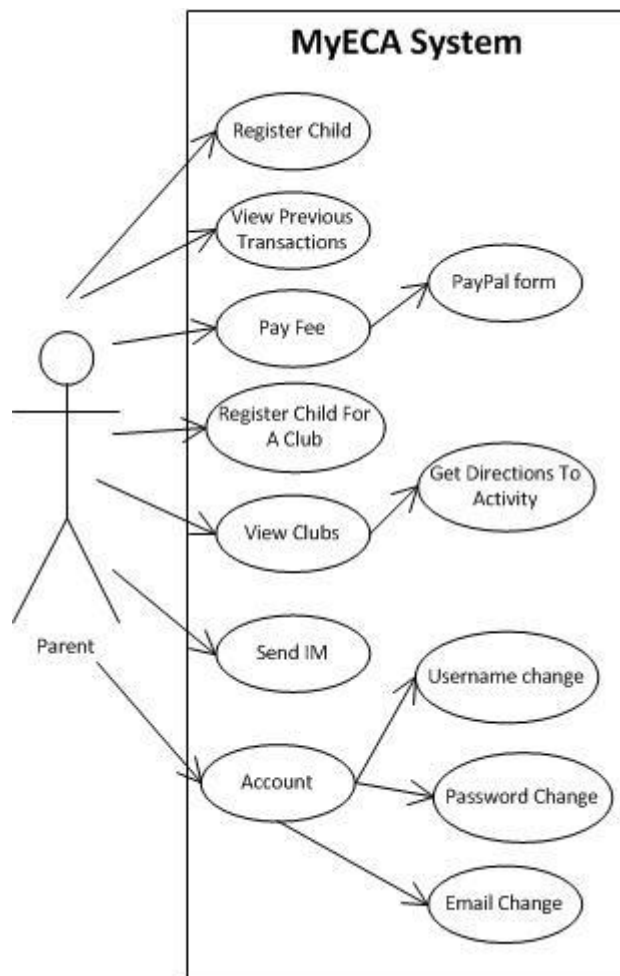
This is a series of steps to illustrate how data is sent via an instant messenger on the system.



3.2.3 Use Case Diagrams

For the next part of the design, I have created three use case diagrams which is one for every user category that currently exists inside the system. I have designed these diagrams so I can put myself into each persona and be able to work out what functional requirements are needed for the individual user category. (For all use cases, it is assumed that the user has already logged in)

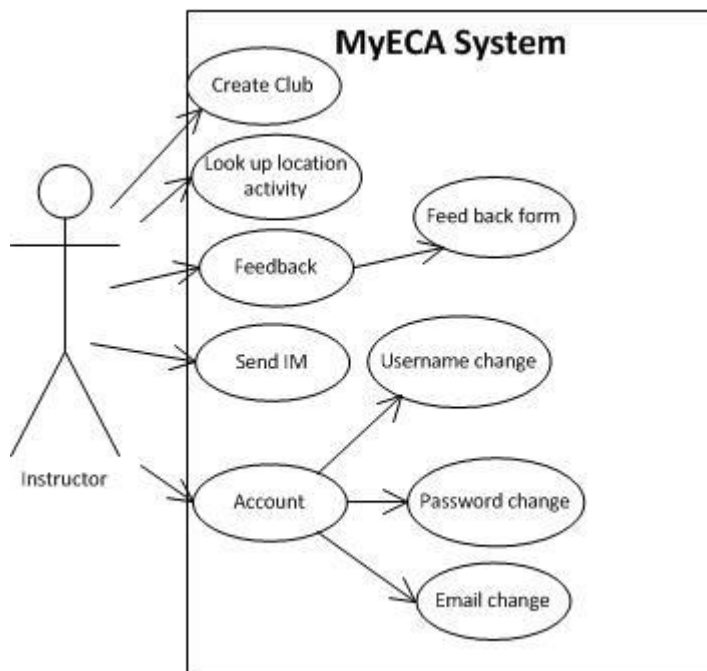
3.2.3.1 Parent Use Case



This is the use case diagram I have created for a parent user. To create this I have tried to rein act what a parent would need to do on the system by navigating round and performing different actions. The left most use cases are actions they are performing unless it is linked to another use case, this means that the second use case it's linked with inherits that case. So what I mean is that to get directions for an activity, the parent must go to view clubs and then select the club to get the directions.

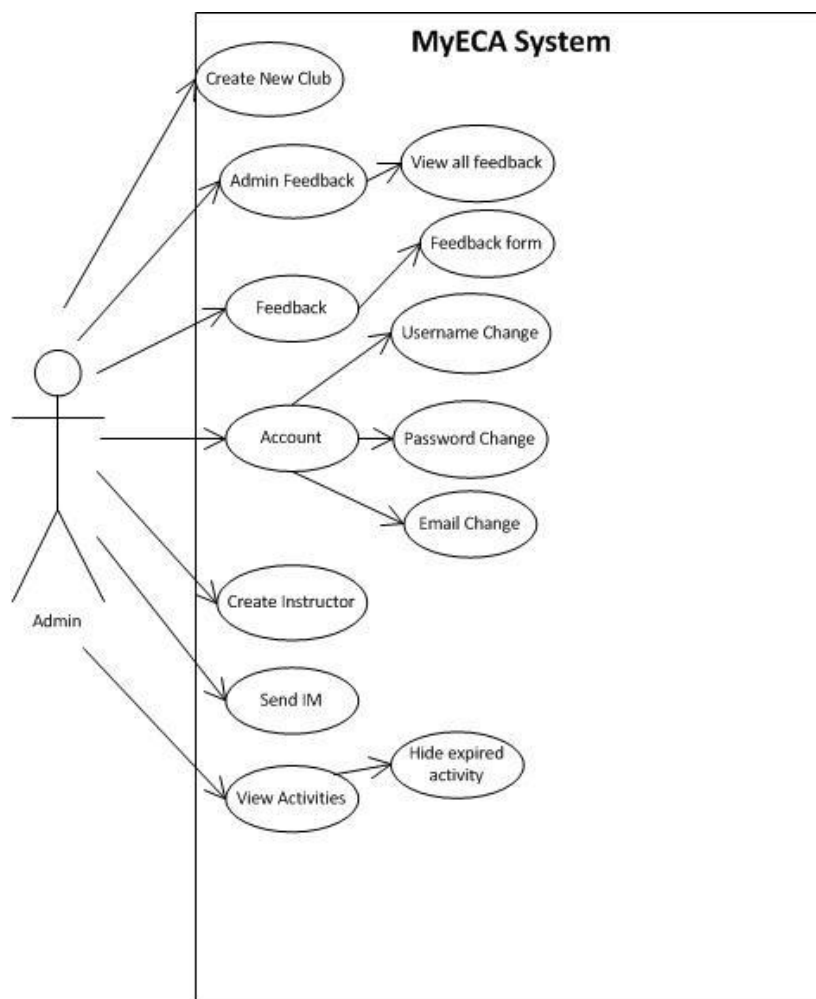
As you can see the amount that a parent can do is quite limited because I don't want them to have a lot of control other than the core functionality in the system.

3.2.3.2 Instructor Use Case



This is the use case I have created for an instructor user type. Instructors can create clubs; leave feedback to students based on their progress in an activity. They can send Instant Messages and like all the other users they have account settings and also they can look up an activity location with the geocode service.

3.2.3.3 Admin Use Case



The Admin has all options and privileges available to them.

Admin feedback allows for viewing all feedback that has been submitted as well as being able to write their own feedback if they are responsible for an activity.

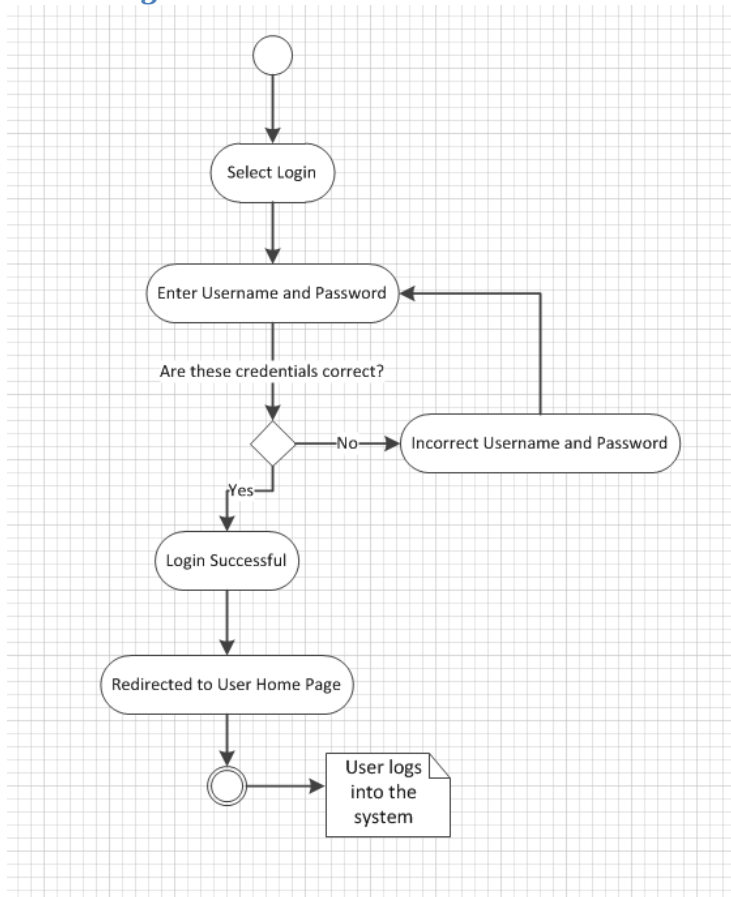
Also, in the activities page they can “hide” activities that have expired, they are not deleted so that I can maintain a record in the database. Later on if this was setup on a server, I could have a cron job that could run monthly to clean out activities that had expired over a month ago to try and clear the database.

Finally, they are the ones responsible for adding instructors to the system.

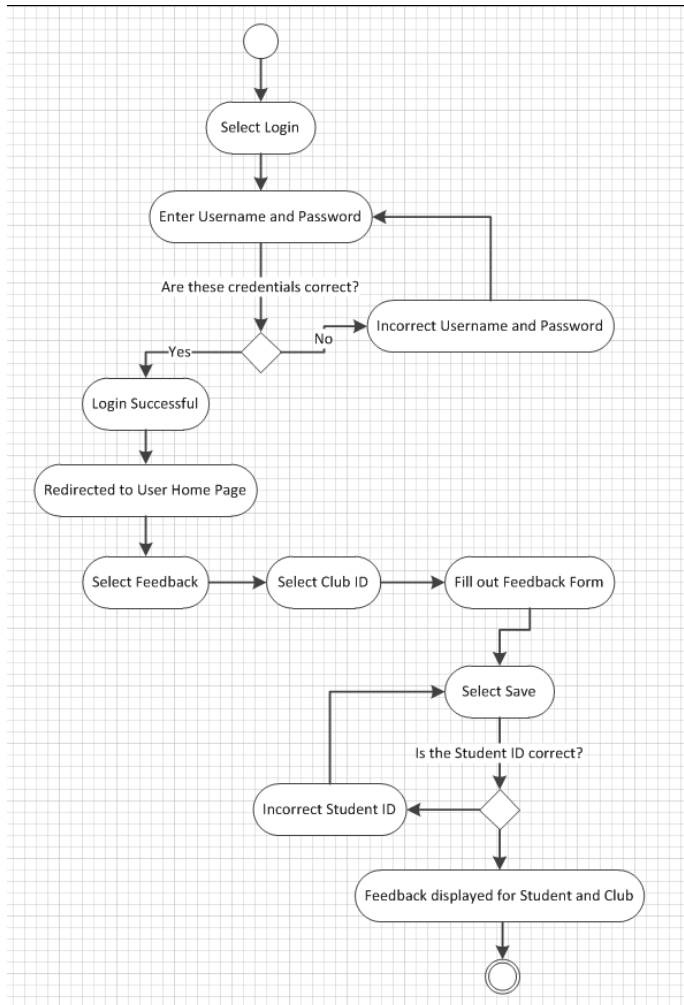
3.2.4 Activity Diagrams

I have created these diagrams to represent different workflows inside the system in a step by step action guide to completing the action. I have provided activity diagrams for the core functionalities in the system instead of doing it for them all. I decided that it was better to show the core functionalities to put emphasis on the fact that without them there would be no system.

3.2.4.1 Login

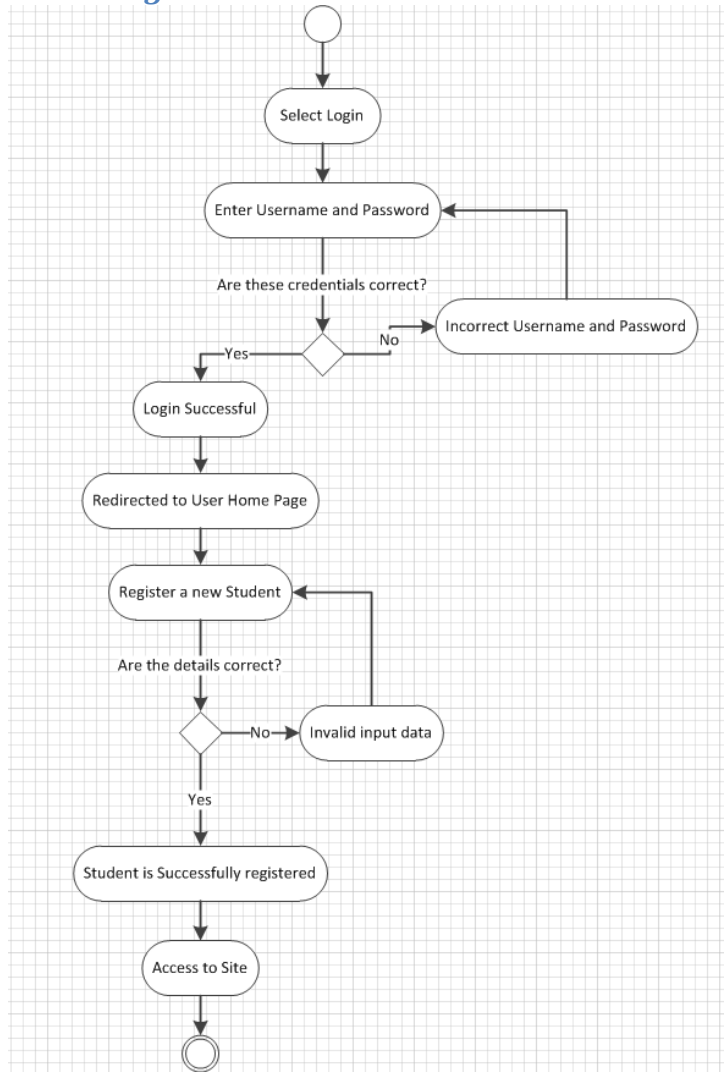


The idea of the login activity diagram is to show the different actions that take place when going through the process. The first part is where the user enters a username and password, then there is a decision element that decides where the credentials are correct, if they aren't then the user has to re-enter their username and password otherwise the login successful message is shown and the user is redirect to the user home page. The circle then represents that the action is complete and then I have written a note to say the current state of which the user is in.

3.2.4.2 Feedback

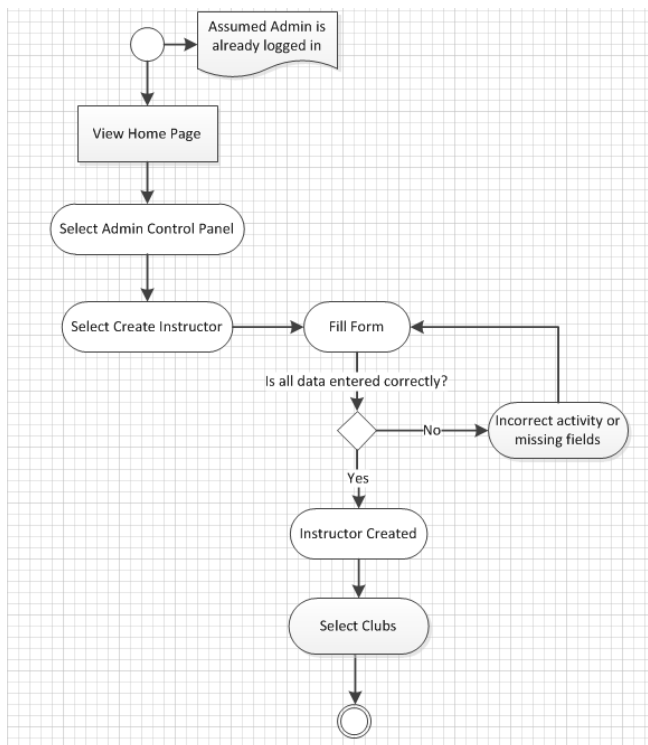
This next diagram is to show the series of events that occur for an instructor to leave feedback about a student who has finished an activity. It starts off with the circle that denotes the start of the activity and then it's the login section which I have described in the previous section 3.2.4.1. After this the instructors selects "select feedback", then they select the ID of the club that has been completed. They must then fill out a feedback form and this is then submitted, then a decision element is used to check if a valid student ID has been entered. If it is the correct ID the feedback is successfully submitted for the parents viewing.

3.2.4.3 Register new child



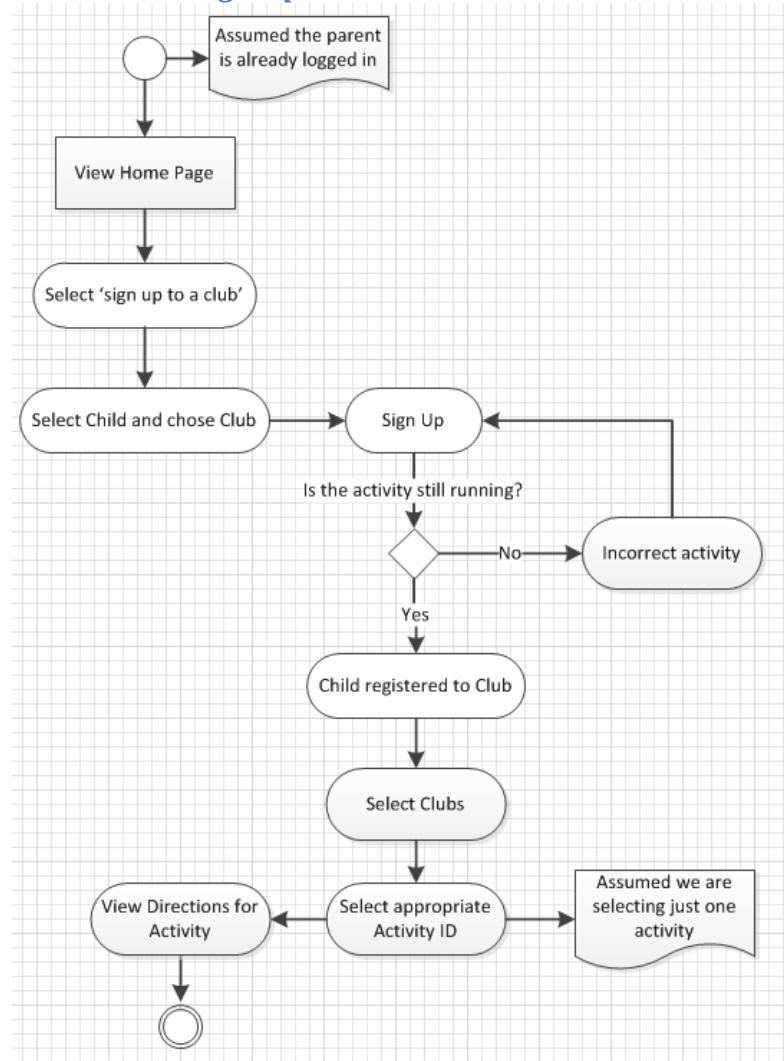
Again, the circle is to denote the start of the diagram and the login section is then carried out. Once the user has logged in, they must then locate the register child form on the user home page. They must then enter valid data, by that I mean no numbers for first names or symbols for a year number. If the details are correct the student is successfully signed up otherwise they must re-enter the data for their child.

3.2.4.4 Create Instructor



This activity diagram is for creating instructor. To save space inside of Visio, I have added a note to the top of the screen to assume that the admin of the system is logged in. The first thing that they must do is access the admin control panel and navigate to the option for creating a new instructor. Once they have done this they must then add the correct form data, again by this I mean no numerical values for names or symbols etc. If the data that the admin has supplied is correct the data is then submitted to the database and the instructor is then created on the system.

3.2.4.5 Child Sign Up & Directions



This is the final activity diagram that I created and inside contains how to sign a child up for an activity and how to get directions to and from the activity based on the user's location. The first thing that the user must do is navigate to the option of signing up to a club and then once the page has loaded select the child name and the activity. NOTE: this doesn't require a decision because the inputs are from dropdowns and doesn't require the user to manually enter any data. The next process is inserting to the correct database tables to sign up the child for the activity. After this, the parent would then navigate to the clubs page and then select the ID of the activity and then this would redirect to the directions page where a map is presented along with directions.

3.2.2 Designs

3.2.2.1 What does the system look like?

The system that I have developed is completely web based it of means that it will be made up of different links, buttons, text boxes, drop downs and text areas which is typical to any web site. With all of these collaborated website characteristics it would then go on to produce a user interface that will allow its users to book activities online from the comfort of their own home and allow school staff in charge of the running of the system to do what they need to on their end, e.g. setting up a new activity etc.

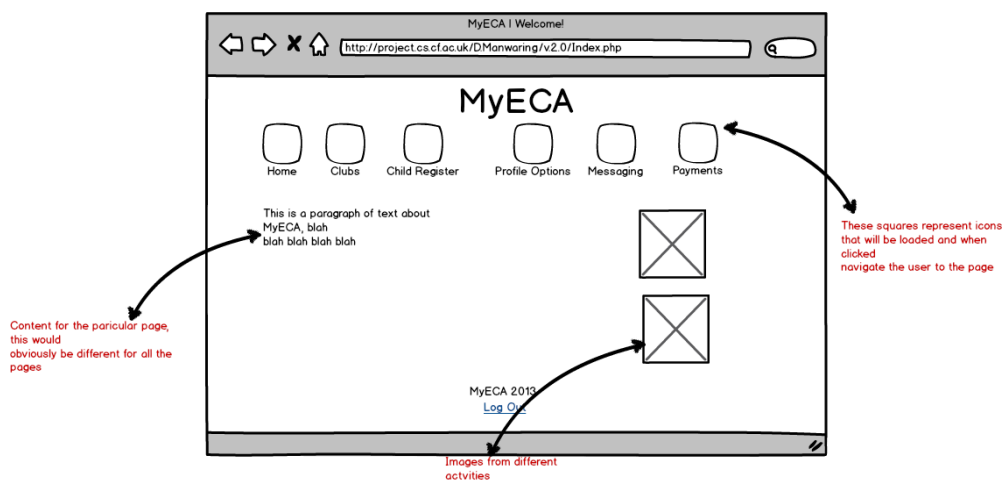
The user interface is clean cut, by which I mean each page does not contain an overwhelming amount of content. On each page there is the navigation menu trailing down the side of the page, along the top contains the bar with the system name, more navigation options and it will state your user name with a small greeting and finally the content for the particular page that a user is on. If you refer to the Approach in the Interim report that I produced earlier this year, you will find digital prototypes of what I wanted the envisaged system to look like at the beginning of the implementation. Of course from what I have described it doesn't look like that anymore and I will explain why later on in this report. In terms of constraints to the system “feel” there aren't any that are humanly possible. At this stage the only real thing to discuss is efficiency and speed of which the content for each page is rendered by the browser.

3.2.2.2 Design choices

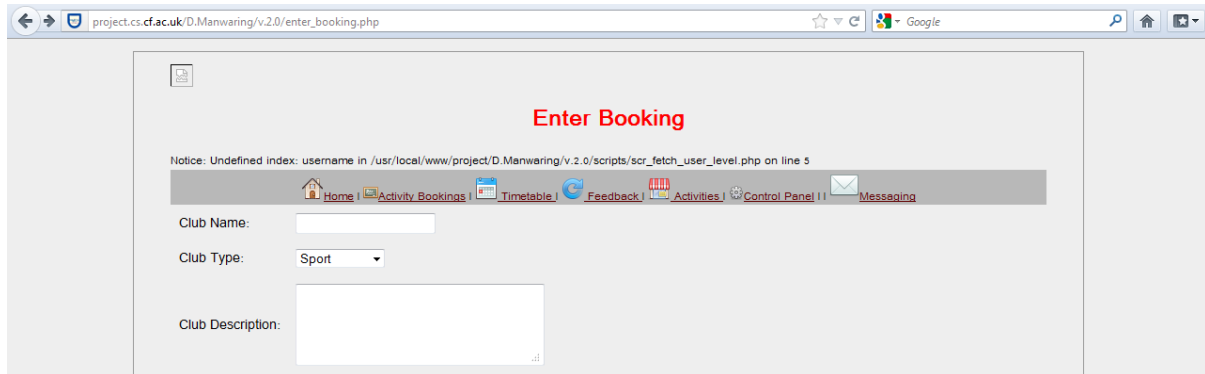
In this subsection of the report, I will talk about the different design choices that I have faced with a full justification and review of each and why I have chosen the final design that is currently in place. At the design phase of the project I created several mocks up using a Google Chrome plugin called Balsamiq, it allowed me to create what I envisaged the system to look like, also it allowed me to trial a few designs to see things such as how user friendly are they, is there too much page content on them etc.

3.2.2.2.1 Original Design

When the implementation phase first began I created my own CSS for the system, it was very basic and I also found online a pack for free to use icons that matched the system. Here is the design of the original system:



I have briefly annotated the diagram inside of Balsamiq but now I will give a full review over the design. The idea at the time was to have a links menu with an icon that corresponds to the page going across the top. I decided it would look a lot nicer to have the icons alongside next to each other. This is just the design, I then went on to implement this in CSS and this is what the outcome of it was:

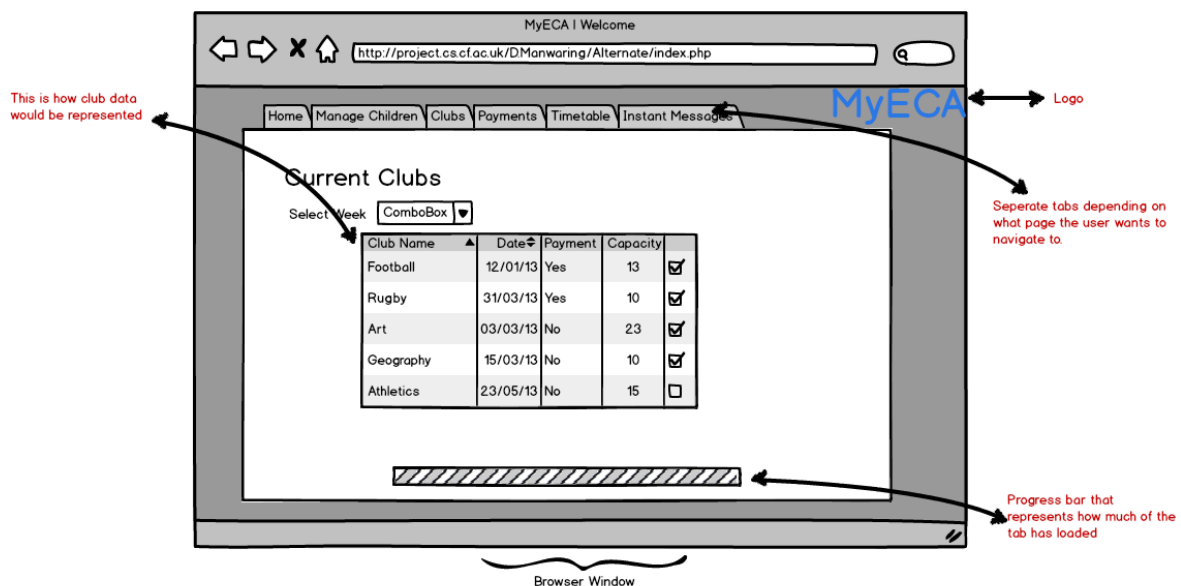


As you can see this is extremely basic, the use of colours is not very nice and if I was to put myself into the user's shoes my first thoughts would be unattractive and not very inviting to use the system. (Nielsen, 1995)

When looking at Jakob Nielsen's 10 usability heuristics, the heuristics that are violated are "Aesthetic and minimalist design" and "Flexibility and efficiency of use", this is because there are a lot of icons and links that could give the user the impression that the browser window is overcrowded. Finally the last heuristic that is violated because the design is quite overwhelming with content on the page, the user will have to spend time having to look where to navigate.

3.2.2.2 Alternate Design

Other than the original design I had designed an alternate in case the design had flaws or I felt the usability wasn't as great as it could be.

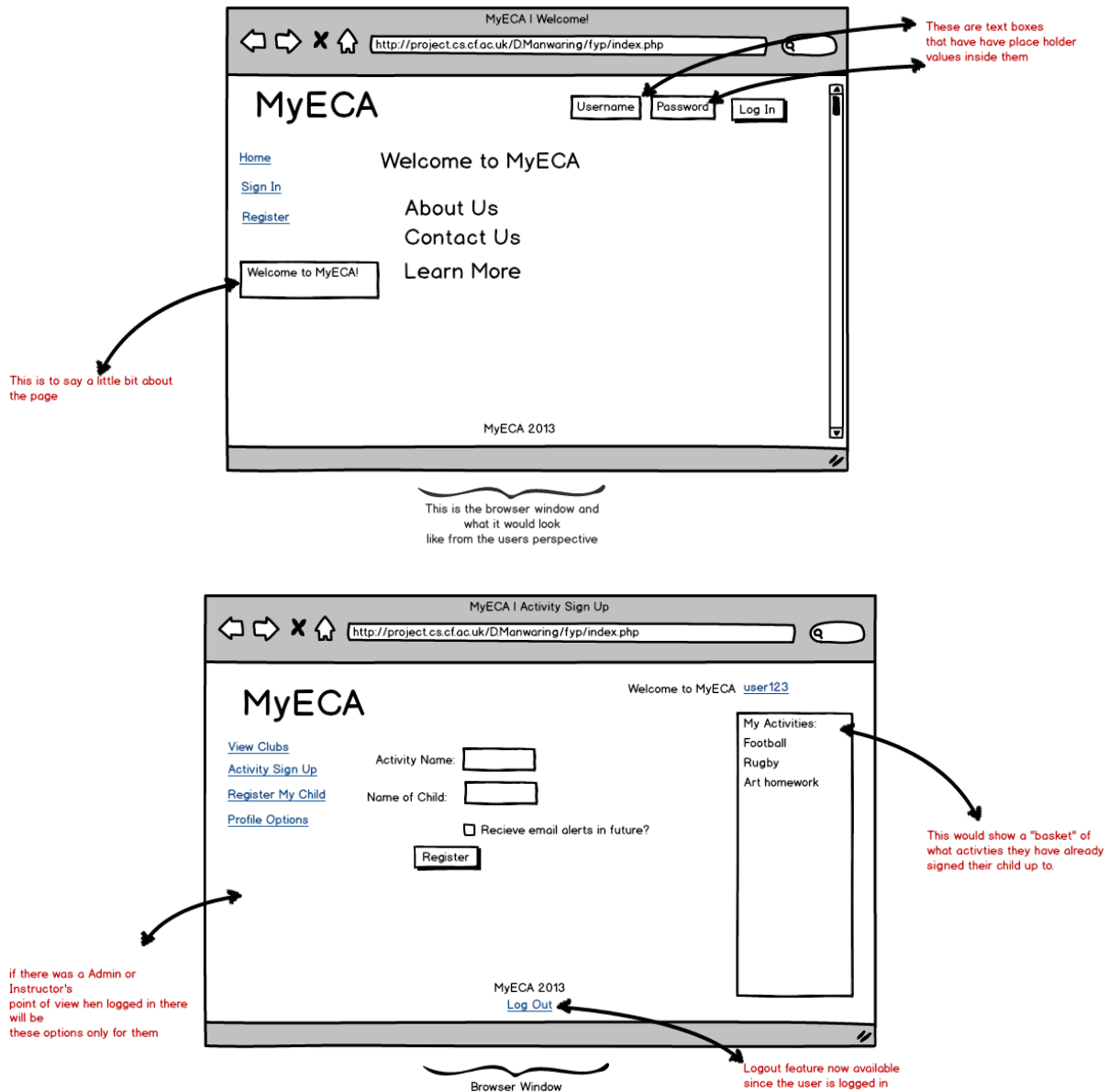


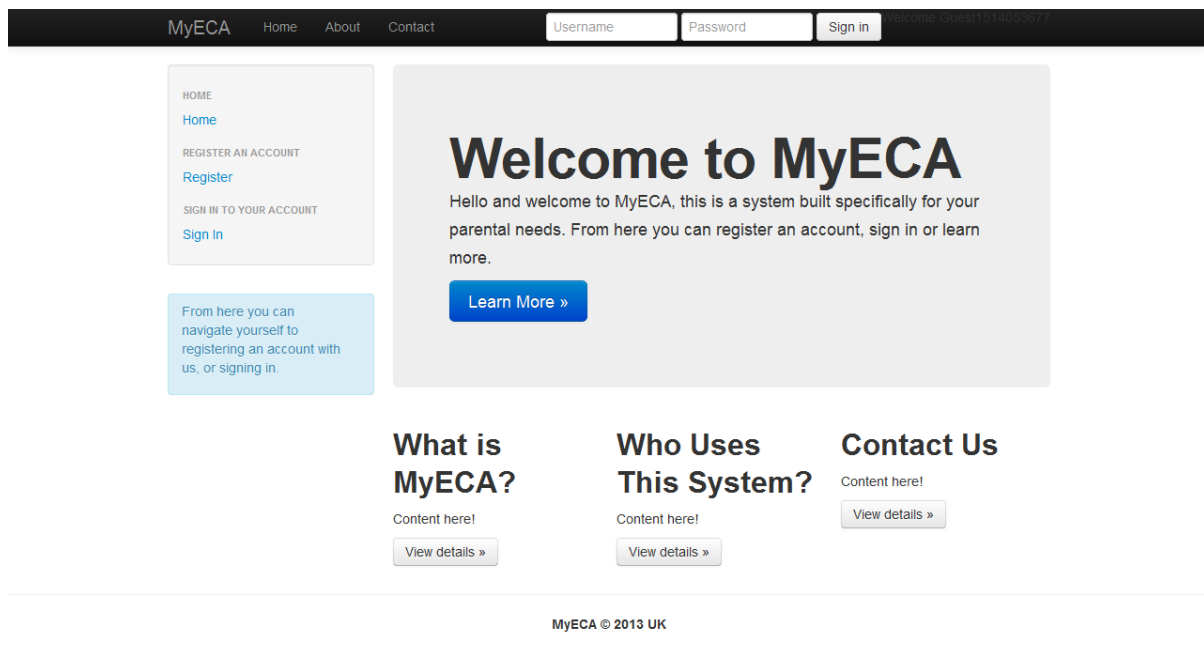
In this approach for the design I entertained the idea of having separate "tabs" that when clicked it will display with a loading bar and present what the user wanted. This would have

been very good to implement as it's something quite different to any other webpage, also the navigation for this sort of thing is straight forward.

3.2.2.3 Final Design

Whilst implementing the core functionality for the original design I felt like the features that were being implemented were too good for the design. I started to look at what other people had done for their designs; I then came across the idea of using a CSS framework. I looked at Twitter Bootstrap, it had a number of different designs that could be implemented and I found one that was very similar to a different design that I had, it wasn't the one with tabs it was one that had a navigation bar going across the top of the page.





As you can see the final outcome is similar to the design, except it is clearly marked that the content is separate from each other. Also the navigation bar is black and contains navigation to home page, about page and contact page. Here is another page while logged into the system:

MyECA Home Clubs Feedback Welcome Admin

HOME
Home
CLUB BOOKINGS
View Clubs
Sign Up To A Club
View Attendees
Create Club
Club Addresses
FEEDBACK
Feedback
PROFILE SETTINGS
Change Username
Change Password
Change Email
ADMIN CONTROL PANEL
View History
Create Instructor
MESSAGING
Instant Messenger

Upcoming Activities

Please note, once you have paid for an activity it can not be refunded due to our school policy.
Today's Date is: 2013-04-30
The time is now: 16:04

ID	Name	Description	Leader	Created	Student Capacity	Date (YYYY-MM-DD)	Posted By	Payment Required
18	Football Training	EXCLUSIVE! For one time only, come and train on the Millennium stadium pitch.	Sheldon Cooper	2013-04-09 15:57:40	14	2013-05-30	Admin	Buy Now
19	Swimming	Swimming lessons	Sheldon Cooper	2013-04-10 12:52:42	10	2013-04-30	Admin	Buy Now

Activity successfully removed
Click the ID of the activity to get directions

Your children are attending:

Club Name:	Name:	
Football Match	TomWoodley	Withdraw
Football Match	BillGoldberg	Withdraw
Science	JamieFacey	Withdraw

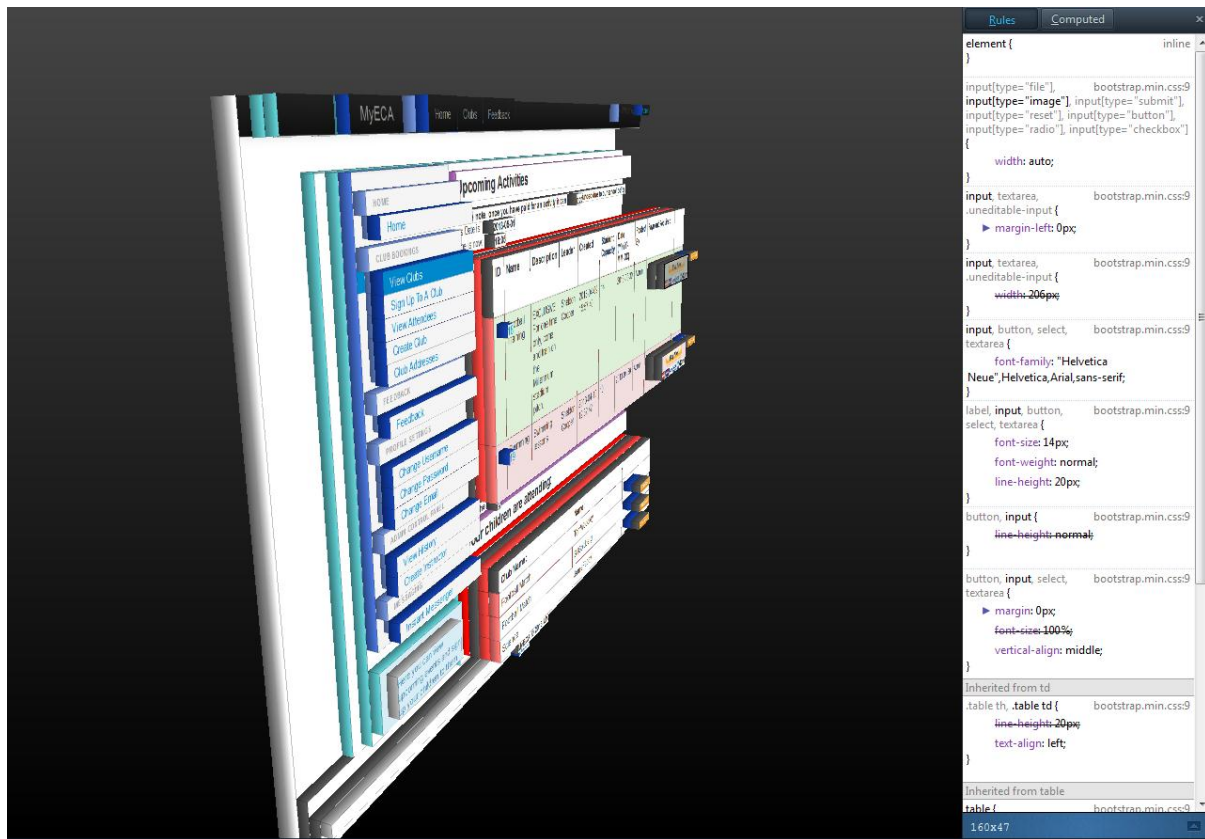
Here you can view upcoming events and sign up your children to them.

MyECA © 2013 UK

This is the bookings page, when the user is logged in there are a lot more navigation options as this is where all the exclusive content is held. If I was to critically evaluate this page in particular which does have a lot of content on, I could say it violates the same heuristic from 3.2.2.2 because when the user has entered this page for the first time and not knowing what to expect, the amount of information is quite a powerful amount. I could change this by moving the bottom part of the page where it says “Your children are attending” to another part of the system to make it tidier and give it that clean cut finish.

3.3.3 CSS Framework

To get an idea of how big the CSS framework this is a 3D style view of what one of my pages looks like when broken down into divisions:

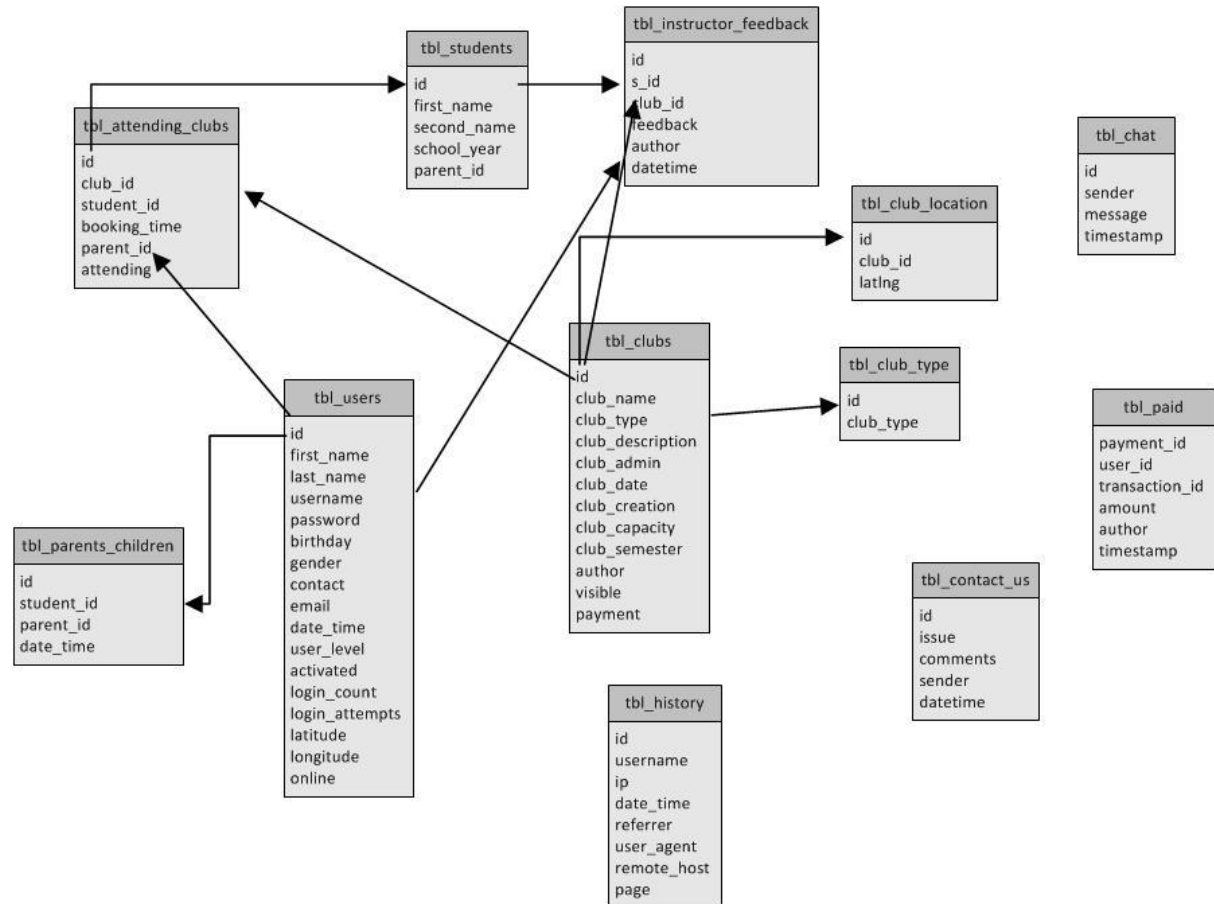


Each part that is split up is inside of a division, when looking at how many there are there are a lot. For me to have hand coded this it would have taken me an extremely long time which is why the framework is used. The whole point is reusing what someone else has instead of recreating the wheel.

Although that using this frame work saved me a lot of extra time having to redesign the entire thing, I have also encountered problems with it. When I first installed it to the system all the page content was queued up behind each other instead of different elements being positioned next to each other. This meant having to load different modules of the frame work and altering the style sheet where certain division classes were called.

(Twitter, 2012)

3.3.4 Database Design



This is the database design for my system. It consists of a series of table that have columns dependent on others. For example, for the **tbl_attending_clubs** table to function correctly it needs values from **tbl_users** such as user ID and it needs club id from **tbl_clubs**.

4.0 Implementation

4.1 My method of implementation

4.1.1 Manual

My chosen method of implementation was to type the code line by line. I didn't like the thought of using a web editor and having code generated. I felt if I were to do it in that way there could be clashes between code that is generated and what I wanted to sue to adapt the system to how I wanted it. Furthermore I have a strong skill set when it comes to web design and knew the vast majority of the code except for parts I had to research so it seemed better and more personal to create the system with my own knowledge and skills. In reflection, it would have been quicker to use a web editor as it would have required not writing much code as it takes care of that side.



I created the majority of the system in Sublime Text Free and Notepad++. I used these text editors as it supported what is called “syntax highlight” which based on type of file you're working in can detect what language you're currently writing and highlight certain language keywords that are reserved.

4.1.2 Software

The other alternative of carrying out the project would have been to use a library of web tools that are currently available. As I mentioned earlier in the report it would have cut the amount of time to do the project but it does offer some drawbacks that I will now discuss.

1. You don't have such good control – by this if you are using a web editor it is generating the code behind the scenes and if you for example want to position a div in an absolute position that you want you would need to go into the dialog box and change it yourself by which point you had might as well have hand written it yourself.
2. Client side scripting – in regards to using JavaScript, people all the time who are involved with implementing different systems are constantly building better scripts, ones that wouldn't be in a web editor, these scripts are often reused rather than starting from scratch
3. Improving/expanding the system – when returning to do maintenance, its always best to know what your code does rather than have no clue what's happening, this is always good when it comes to debugging errors.

4.1.2 Method chosen

At the beginning of the project I did have knowledge and personal experience of object oriented programming. When looking at the scope of the project I didn't feel I had enough time to brush up my skill set in PHP for object oriented programming. I thought that taking a risk like that would put the project in a bad position before barely starting. It would have meant having to research and practice object oriented approaches in PHP before carrying going ahead and coding the system. For a while I weighed up a couple possible paths that I

could go down. With the knowledge I had I could use a procedural approach which I was a lot more comfortable with but meant making a system that wasn't as efficient as it could be or go so far out of my comfort zone, risk not being able to implement all core functionality and advanced features but having some sort of efficient code. When looking at this from a different perspective, if the project went on for longer then maybe an object oriented approach would be more suitable as I would have longer to learn different skills and execute them.

In the end I decided to use the procedural approach and use object oriented programming where I felt it would be necessary and wouldn't require over whelming amounts of object oriented solutions to solve different problems in the system.

4.2 Page break down

NOTE: to help see what code I am referring to where print screening, I have drawn arrows and colour coordinated the code I am talking about in this report to what is inside the code editor. Also I am assuming that the reader has a standard understanding of PHP and MySQL. Also, parts that don't include unforeseen problems mean that there are no problems to be reported.

4.2.1 Database script

4.2.1.1 Description

The purpose of this script is for the system to connect to and from the Host. It allows the system to access records that are stored in tables in the database and manipulate data inside. The script is purely for connection purposes, it sets up properties that assign values for the host address, the database username and password along with the database you want the system to be working off. This script took very little time to build due to previous personal developments I had done before the project.

4.2.1.2 External pages, functions and constants used

There are no external pages or functions for the script. There are only the built in PHP MySQL connection methods.

4.2.1.3 Code

```
1 <?php
2 session_start();
3
4 $db = array(
5     'host' => 'ephehus.cs.cf.ac.uk',
6     'user' => 'c1030889p',
7     'pass' => 'CitNam',
8     'name' => 'dbp1030889'
9 );
10
11 mysql_connect($db['host'], $db['user'], $db['pass']);
12 mysql_select_db($db['name']) or die('Unable to connect to database');
13 ?>
```

The first thing to mention here is **session_start()**. This is used for creating a new session variable when a user logs in. Also you can only use this variable when you have this line of code in a file.

Another reason it's needed is for when logging out the system, when you log out you have to start the session in the log out script before you destroy it.

The next part of the file is the **array** that is assigned to the variable **\$db**. Inside it creates a host value that is assigned the value of the host address of where the database is. Then inside user is the value of the database username, and the same for password. Name is assigned the database name. After these values have been assigned, there is a built in MySQL function in PHP called **mysql_connect()** that connects to the database. Finally there

is also the other built in functional called `mysql_select_db()`. Or die is used in case the script cannot connect to the database so it will provide an error message.

4.2.1.4 Unforeseen problems

With this sort of script there were no unforeseen problems to report. This is due to the simplicity of the purpose of this particular script. There are also no

4.2.2 User level check

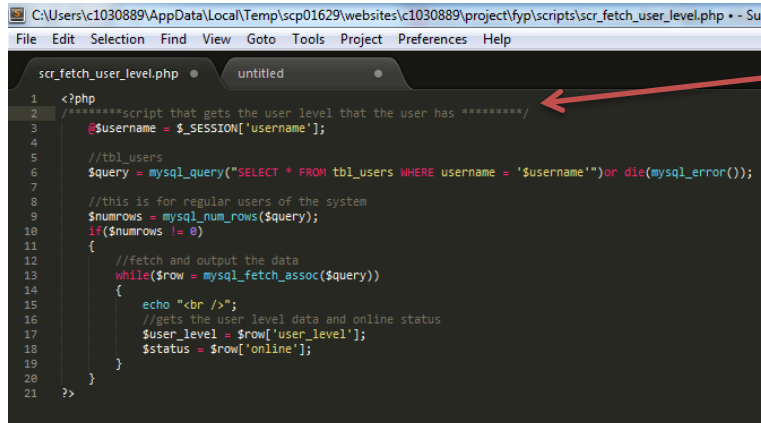
4.2.2.1 Description

This is a script stored inside the scripts directory of the root folder. This script is included on all pages to check the user level of a logged in user.

4.2.2.2 External pages, functions and constants used

There are no external pages or functions in this script.

4.2.2.3 Code



```

1 <?php
2 /*****script that gets the user level that the user has *****/
3 $username = $_SESSION['username'];
4
5 //tbl_users
6 $query = mysql_query("SELECT * FROM tbl_users WHERE username = '$username'") or die(mysql_error());
7
8 //this is for regular users of the system
9 $numrows = mysql_num_rows($query);
10 if($numrows != 0)
11 {
12     //fetch and output the data
13     while($row = mysql_fetch_assoc($query))
14     {
15         echo "<br />";
16         //gets the user level data and online status
17         $user_level = $row['user_level'];
18         $status = $row['online'];
19     }
20 }
21 ?>

```

This is a comment for if future developers who had little knowledge of the system would know what the script does.

The next part stores the username session username inside the variable **\$username**.

The next part runs a query on the database.

It uses another MySQL built in function to run a query. This then selects the columns from the database based where the username column value is equal to the users username. Next it counts the rows using another MySQL function, and the next part fetches the data from the database. This is done by looping through the results from the SQL query and assigning the column “user_level” to **\$user_level**. This is the variable that is then used everywhere else in the system to perform the check.

4.2.2.4 Unforeseen problems

This script took very little time to build, also no problems were encountered when developing except for the **\$username** variable, I suppressed the error with “@” and having looked back I could have put an if statement around it and changed it to `isset()`.

4.2.3 Header

4.2.3.1 Description

Every page of the system contains a header file, inside of it contains all code that is at the top of the page for all the files. It contains PHP logic to check if there a user session set so it can display options for a user that is logged in or logged out. At the top of the file it includes the database connection script which allows for the external functions to work. By calling this function in header.php it means the database connection script only needs to be called once.

4.2.3.2 External pages, functions and constants used

External pages are:

1. Config.php – this is the database connection script
2. Sct_fetch_user_level.php – checks the user level of a logged in user

These two pages are detrimental to the system (these are explained later this section into further detail) but they are required for a few reasons.

The file Config.php contains the database connection script to the server and selects the database that the project will be working off on the backend. Instead of having this database connection all pages that require it, it would be easier to just include it one file so that when the database is needed to be accessed it can be without having to add additional liens of code. Plus it's more secure to have this sensitive code separate from the other files.

Scr_fetch_user_level.php is required on all pages; the script in total is 23 lines. It is required on all pages because in the system there are different options for the users that need to be printed based on their user level. For example the Admin has its high level options whereas the parent doesn't. On each of the pages inside the system there is a navigation menu that contains all the links to other pages.

4.2.3.3 Code

```

1  <?php require_once('scripts/config.php'); ?>
2  <?php require_once('scripts/scr_fetch_user_level.php'); ?>
3  <!DOCTYPE html>
4  <html>
5
6      <head>
7          <title></title>
8
9          <link rel="stylesheet" type="text/css" href="css/layout.css" />
10     </head>
11     <body>
12
13         <header>
14             <div class="nav navbar-inverse navbar-fixed-top">
15                 <div class="nav navbar-inner">
16                     <div class="container">
17                         <?php if(isset($_SESSION['username'])): ?>
18                         <a class="brand" href="home_area.php">MyECA</a>
19                         <?php else: ?>
20                         <a class="brand" href="index.php">MyECA</a>
21                         <?php endif; ?>
22                         <ul class="nav">
23                             <?php if(isset($_SESSION['username'])): ?>
24                             <li><a href="home_area.php">Home</a></li>
25                             <?php else: ?>
26                             <li><a href="index.php">Home</a></li>
27                             <?php endif; ?>
28                             <?php if(isset($_SESSION['username'])): ?>
29                             <li><a href="bookings_page.php">Clubs</a></li>
30                             <li><a href="feedback.php">Feedback</a></li>
31                             <?php else: ?>
32                             <li><a href="about_us.php">About</a></li>
33                             <li><a href="">Contact</a></li>
34                             <?php endif; ?>
35                         </ul>
36                         <?php if(isset($_SESSION['username'])): ?>
37                         <p class="pull-right">Welcome <a href="#"><?php echo $_SESSION['username']; ?></a>
38                         </p>
39                         <?php else: ?>
40                         <li class="pull-right">Welcome Guest<?php echo rand(); ?></li>
41                         <form class="navbar-form pull-right" method="post" action="scripts/scr_login_script.php">
42                         <input class="span2" type="text" name="username" placeholder="Username">
43                         <input class="span2" type="password" name="password" placeholder="Password">
44                         <button type="submit" class="btn">Sign in</button>
45                         </form>
46                         <?php endif; ?>
47                     </div>
48                 </div>
49             </div>
50         </header>
51
52         <div id="content-container" class="container">

```

Database connection script and user level check inclusion.

The next thing is the style sheet for the system, this is what produces the design for the system.

Next inside the **<header>** tags is the navigation bar. This has a series of **<div>** tags that have classes that link to the style sheet to style the particular division.

Inside **<div class="container">** is where the content goes for the navigation bar, it consists of logic that checks if the user is logged in. This section of code is for the diagram below.

MyECA Home About Contact

Welcome Guest1233306800

What's happening is that PHP is performing a check to see if a username session is currently set (user is logged in) if it is set then it will provide options for a logged in user, otherwise it

will provide options for a user who isn't logged in. The built in function `isset()` in PHP takes care of this. Also, conventional PHP syntax is usually as follows:

```
<?php
If(some condition)
{
//Content here
}
?>
```

For what I'm trying to do though is I have done an inline IF statement, this prevents having to use the echo function in PHP to output the HTML and instead to put the closing PHP tag and then have the raw HTML content and then end the if statement. All I'm trying to say is that it's tidier and easier for me the developer to go back to later if I need to debug or update code. I have done this approach quite a lot in the system but this is merely to justify why I have taken this approach of doing conditional logic the way I have.

The final thing to mention is that there is a login form in the navigation bar, this is also based on the same logic which is if there is a username session variable set in PHP.

4.2.3.4 Unforeseen problems

The only problem I encountered when developing the header was deciding what content needed to be in the top of the page. For a while I trialled having the second navigation bar but it created too many bugs and errors and for the project it was easier for development to have it included in all of the files. When it was in the same file I faced the problem of trying to get the menu option to highlight based on what page it was on etc. This took up a few days of the project and didn't cause for any serious concern.

4.2.4 Footer

4.2.4.1 Description

This page is included in all of the web pages for the system, it is a very small file but in terms to make this system as efficient as possible, looking at what is going to appear in the bottom of all pages is a must. At the bottom of mine I have a `<footer>` tag that has content enclosed, and it contains the name of the system and the option to logout.

4.2.4.2 External pages, functions and constants used

For this file there are no external pages or functions used, it just PHP logic that checks something based on a certain global variable being set. Also there are no constants set as the code for it doesn't have any object oriented programming.

4.2.4.3 Code

```

1      </div>
2      <footer>
3      <hr>
4      <center>
5      <p><strong>MyECA &copy; 2013 UK</strong></p>
6      <?php if(isset($_SESSION['username'])): ?>
7      <a href='scripts/scr_logout.php'>Logout</a>
8      <?php endif; ?>
9      </center>
10     <hr>
11     </footer>
12     <script type="text/javascript" src="js/bootstrap.min.js"></script>
13 </body>
14 </html>

```

The first thing is that there are **<footer>** tags. Inside the footer tags enclosed are **** tags that make the text bold. The next thing is the name of the system and the copy right. The copyright is made with **©**. Then there is some PHP logic, it says if there is a user name session set then show the logout option. The other tags close elements that have been opened in the main body of the content.

NOTE: as I mentioned above in section 4.2.1.3 I have used the same style of conditional logic.

4.2.4.4 Unforeseen problems

Unlike the header file, this is very small in size and consists of fourteen lines of code. For this I was able to clearly identify what needed to be contained in the footer of each page which meant it took barely any time to complete. Also there have been no problems testing this file when included or by it elf.

4.2.5 About Us

4.2.5.1 Description

This page contains dynamic content that gives the user information about the system and what its main features are. Like all of the pages there is a navigation menu at the top of the screen and down the side there are further navigation options. On this webpage is a jQuery feature that is a photo carousel, it allows the user to scroll through screenshots of the system that are slightly blurred for data and user protection. To create this affect a series of `<div>` tags were created to hold the content of each “slide” of the photo carousel and also there is a link to jQuery’s uncompressed version of the JavaScript library.

4.2.5.2 External pages, functions and constants used

These are the external files that this web page inherits:

- Header.php
 - Config.php
 - Scr_check_user_level.php
- Scr_register_child.php
- jQuery.js
- Footer.php

4.2.5.3 Code

```

29      <!-- Carousel -->
30      <div id="myCarousel" class="carousel slide">
31          <div class="carousel-inner">
32              <div class="item active">
33                  
34                  <div class="container">
35                      <div class="carousel-caption">
36                          <h1>Register today with MyEcal</h1>
37                          <p class="lead">When you join our system we promise to deliver a simple, fast and effective solution to enter curricular activities. With the use of our c
38                          <a class="btn btn-large btn-primary" href="register_account.php">Sign up today</a>
39                      </div>
40                  </div>
41              </div>

```

This is a code snippet for one of the “slides” inside the photo carousel. As you can see it is a series of embedded `<div>` tags where there is a link to an image along with a concise description about why the user should join.

4.2.5.4 Unforeseen problems

For the smart feature that lives inside this file, it took me a couple days to get the photo carousel fully functioning. I experienced problems with where I was placing the link to the external jQuery file and it was a little difficult trying to figure out how this feature worked before implementing it. In also faced problems with the design, the text I was trying to have placed on it was white and you could barely see it so time was spent searching through the CSS framework to alter the code specifically for this. Testing this in different environments has also been fine too. Even though these problems were experienced it didn’t put any strain on the project and getting it completed in time.

4.2.6 User home

4.2.6.1 Description

This is the first page that any user will see once that they have successfully logged into the system. On the page contains a large choice of different options (navigations) that the user can traverse down. Also, on this page contains a few functions. The first one which is very important is the sign up form for a parent to register a child. It takes three inputs, one is their child's first name, then second name and finally what year that they are in at school.

4.2.6.2 External pages, functions and constants used

These are the external pages and functions they have:

- Header.php
 - Config.php
 - Scr_checker_user_level.php
- Footer.php

4.2.6.3 Code

```

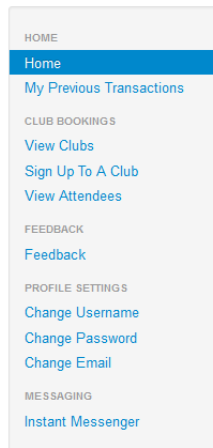
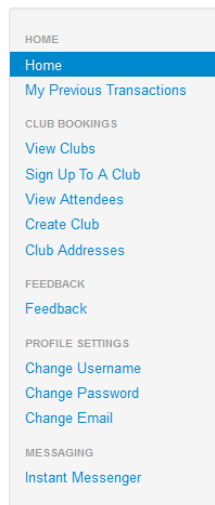
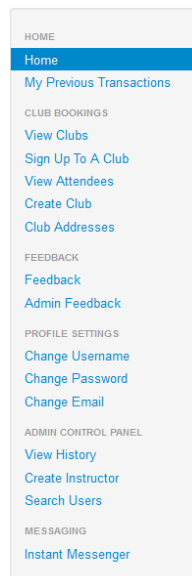
1 <?php require_once('layout/header.php'); ?>
2 <div class="row">
3   <div id="sidebar" class="span3">
4     <ul class="nav nav-list well">
5       <li class="nav-header">Home</li>
6       <?php if(isset($_SESSION['username'])): ?>
7       <li class="active"><a href="home_area.php">Home</a></li>
8       <li><a href="my_transactions.php">My Previous Transactions</a></li>
9       <?php else: ?>
10      <li class="active"><a href="index.php">Home</a></li>
11      <?php endif; ?>
12      <li class="nav-header">Club Bookings</li>
13      <li><a href="bookings_page.php">View Clubs</a></li>
14      <li><a href="child_sign_up.php">Sign Up To A Club</a></li>
15      <li><a href="attendees.php">View Attendees</a></li>
16      <?php if($_SESSION['level'] == 1): ?>
17      <li><a href="enter_booking.php">Create Clubs</a></li>
18      <li><a href="geocode_service.php">Club Addresses</a></li>
19      <?php endif; ?>
20      <li class="nav-header">Feedback</li>
21      <li><a href="feedback.php">Feedback</a>
22      <?php if($_SESSION['level'] == 3): ?>
23      <li><a href="admin_feedback.php">Admin Feedback</a></li>
24      <?php endif; ?>
25      <li class="nav-header">Profile Settings</li>
26      <li><a href="username_change.php">Change Username</a></li>
27      <li><a href="password_change.php">Change Password</a></li>
28      <li><a href="email_change.php">Change Email</a></li>
29      <?php if($_SESSION['level'] == 3): ?>
30      <li class="nav-header">Admin Control Panel</li>
31      <li><a href="view_history.php">View History</a></li>
32      <li><a href="create_instructor.php">Create Instructor</a></li>
33      <li><a href="search.php">Search Users</a></li>
34      <?php endif; ?>
35      <li class="nav-header">Messaging</li>
36      <li><a href="instant_messenger.php">Instant Messenger</a></li>
37    </ul>
38  </div>

```

Just to reiterate what I mentioned in section 4.2.1.3 I have used a different style of conditional logic, here is the code for the navigational bar that runs down the side of the page.

In the code there is conditional logic based on the user level, for the instructors options it is "`!= 1`" which is 2 levels of 2 and 3 and for the admin I have "`= 3`". Where there isn't logic, it means there are options that are applicable for everyone.

On the next page you can see the three different menus that are available when logged in as each of the three user categories

Parent**Instructor****Admin**

By looking at these three different menus, you can see a contrast between the amount of options (privileges) available to each user.

Final notes on this module, these screenshots were taken inside of “home_area.php” and as you can see the “Home” link is highlighted, this is done with:

```

6      <?php if(isset($_SESSION['username'])): ?>
7      <li class="active"><a href="home_area.php">Home</a></li>
8      <li><a href="my_transactions.php">My Previous Transactions</a></li>
9      <?php else: ?>
10     <li class="active"><a href="index.php">Home</a></li>
11     <?php endif;?>
12     <li class="nav-header">Club Bookings</li>

```

What’s happening is that the list element class is calling “active” from the CSS sheet and is styling it to be highlighted in the way that it is.

Looking further at this page, the next part of the interface is where the user can register their child to the system. For it I created an HTML form that contained three fields which were first name, second name and school year, these values are then passed to “scr_register_child.php” which does the following:

```

1  <?php
2  include "config.php";
3  $firstname = $_POST['fname'];
4  $lastname = $_POST['lname'];
5  $schoolyear = $_POST['year'];
6  $parent = $_SESSION['username'];
7  $level = 4;
8  $activated = 0;
9  $login_count = 0;
10 $login_attempts = 0;
11
12 if($firstname && $lastname && $schoolyear && $parent)
13 {
14     $query = mysql_query("INSERT INTO tbl_students VALUES(NULL, '$firstname', '$lastname', '$schoolyear', ' ', '$parent', NOW(), '$level', '$activated', '$login_count', '$login_a
15     header("Location: ../home_area.php");
16 }
17 else
18 {
19     echo "empty";
20 }
21 ?>

```

This script gets form data and stores them into variables, it then performs a check to make sure the values contain values. Then a query is run, this query inserts the values into tbl_students. One thing to note here is “NOW()” inside the query, this is MySQL syntax that inserts the date and time to the table.

The next part of this page is the table that displays what children the parent has registered on the system.

```

69      <h3>Registered Children</h3>
70      <?php
71          $name = $_SESSION['username'];
72          $query = mysql_query("SELECT
73              `tbl_students`.*, `tbl_users`.`first_name` AS pFirstname,
74              `tbl_users`.`last_name` AS pLastname FROM `tbl_students`,`tbl_users`
75              WHERE `tbl_students`.`parents` = `tbl_users`.`id` AND
76              `tbl_users`.`username` = '$username'" ) or die(mysql_error());
77          $numrows = mysql_num_rows($query);
78          if($numrows != 0)
79          {
80              echo "<table class='table table-hover'>";
81              echo "<th>First Name</th>";
82              echo "<th>Last Name</th>";
83              echo "<th>School Year</th>";
84              echo "<th>Date/Time</th>";
85              while($row = mysql_fetch_assoc($query))
86              {
87                  echo "<tr>";
88                  echo "<td>" . $row['first_name'] . "</td>";
89                  echo "<td>" . $row['second_name'] . "</td>";
90                  echo "<td>" . $row['school_year'] . "</td>";
91                  echo "<td>" . $row['timestamp'] . "</td>";
92                  echo "</tr>";
93              }
94              echo "</table>";
95          }
96          mysql_close();
97      ?>
98  </div>

```

The first part of this script is a query; this is one of the more complex SQL queries in the system. In the database there is “tbl_users” where the parents are and “tbl_students” where their children are. In the “tbl_students” table recorded with them is the ID of their parent, so what the query does is perform a join where the ID from the parent table matches their child’s table. That way other people’s children don’t appear on the screen.

After this the table is then displayed and then the values of the table are looped using the **while loop**.

The final part of this page gives the parent an option to register their location to the system (this is used later for getting directions to and from activities and is also discussed later on).

4.2.6.4 Unforeseen problems

This page took about a week and a half to be constructed the way it is now. I had different features on this page that were then removed and I wasn’t sure about the way things were worded and position. Again I encountered additional problems with the CSS frame work, what was happening was that the second navigation bar was at the top of the page and everything was appearing below it rather than next to it. I overcame this technical problem by changing some of the code in the framework.

4.2.7 Activities page

4.2.7.1 Description

This is the page where all activity information is shown. On the page it displays the ID of an activity which in actual fact is an anchor because when clicked it redirects, the name of the club, description, instructor name, when created (which is an admin only view), the capacity, when the club is, who created it (admin only view) and if there is a small fee required for the activity.

Further down the page is a table that displays what children have been currently signed up to what activity, and also the feature to withdraw them from an activity.

4.2.7.2 External pages, functions and constants used

These are what are included:

- Header.php
 - Config.php
 - Scr_check_user_level.php
- Footer.php

4.2.7.3 Code

```

69     <?php while($row = mysql_fetch_array($query)): ?>
70     <?php
71     $today = date("y-m-d");
72     $date = $row['club_date'];
73     $visible = $row['visible'];
74     $payment = $row['payment'];
75
76     if($user_level == 3) //for admin users
77     {
78         if(strtotime($today) > strtotime($date))
79         {
80             echo "<tr class='error'>";
81             $enc = base64_encode($row['id']);
82             echo "<td><a href='get_directions.php?id=' . $enc . '>' . $row['id'] . "</a></td>";
83             echo "<td>" . $row['club_name'] . "</td>";
84             echo "<td>" . $row['club_description'] . "</td>";
85             echo "<td>" . $row['club_admin'] . "</td>";
86             if($user_level == 3)
87             {
88                 echo "<td>" . $row['club_creation'] . "</td>";
89             }
90
91             if($user_level == 3)
92             {
93                 echo "<td>" . $row['club_capacity'] . "</td>";
94             }
95
96             echo "<td>" . $row['club_date'] . "</td>";
97             echo "<td>" . $row['author'] . "</td>";
98
99             if($payment == 1)
100             {
101                 echo "<td>" . "
102                 <form action='https://www.sandbox.paypal.com/cgi-bin/webscr' method='post' target='_top'>
103                 <input type='hidden' name='cmd' value='_xclick'>
104                 <input type='hidden' name='business' value='FAGW748CETQXQ'>
105                 <input type='hidden' name='lc' value='GB'>
106                 <input type='hidden' name='item_name' value='" . $row['club_name'] . "'>
107                 <input type='hidden' name='amount' value='5.00'>
108                 <input type='hidden' name='currency_code' value='GBP'>
109                 <input type='hidden' name='button_subtype' value='services'>
110                 <input type='hidden' name='no_note' value='0'>
111                 <input type='hidden' name='cn' value='Add special instructions to the seller:'>
112                 <input type='hidden' name='no_shipping' value='2'>
113                 <input type='hidden' name='bn' value='PP-BuyNowBF:btn_buynowCC_LG.gif:NonHosted'>
114                 <input type='image' src='https://www.sandbox.paypal.com/en_US/GB/i/btn/btn_buynowCC_LG.gif' border='0' name='submit' alt='PayPal - The safer, eas
115                 <img alt=' border='0' src='https://www.sandbox.paypal.com/en_GB/i/scr/pixel.gif' width='1' height='1'>
116                 </form>
117                 " . "</td>";
118             }
119             echo "<td>" . "<a href='bookings_page.php?term=' . $row['id'] . '>'><input type='submit' class='btn btn-mini btn btn-warning' name='submit' value='Remove'
120             echo "</tr>";

```

The first thing that is happening is that we are fetching the date value from “tbl_clubs” and storing it in the **date** function along with the date of the club, visibility and payment.

The next part is for if the user is an admin, the whole point of this block of code is for the admin to see what activities have expired and which aren’t. Ones that have expired can be set to invisible and can’t be shown then. Inside the if statement there is a function called **strtotime()** that converts a string value into date and time format. It checks if the date received from the date base is greater than today’s date (date has passed). If it has then this particular row is highlighted red. There is also an else statement that prints the row in green (date hasn’t passed).

The if statement that checks if equal to one (true) then it displays the PayPal “buy now” button which then connects to Sandbox to simulate the transaction.

4.2.7.4 Unforeseen problems

On this page was a feature that I overlooked far too much. That was the PayPal “buy now” button. It took me a long time to figure out how this button worked and what additional files were required for it to work. On the website you can custom create the button but because this is in my system I needed a way of dynamically creating a different price for activities along with the name etc., this was all done through the use of variables. This problem did set me back a little which meant extra work had to be done in other areas on completion of this functionality.

4.2.8 Child sign up

4.2.8.1 Description

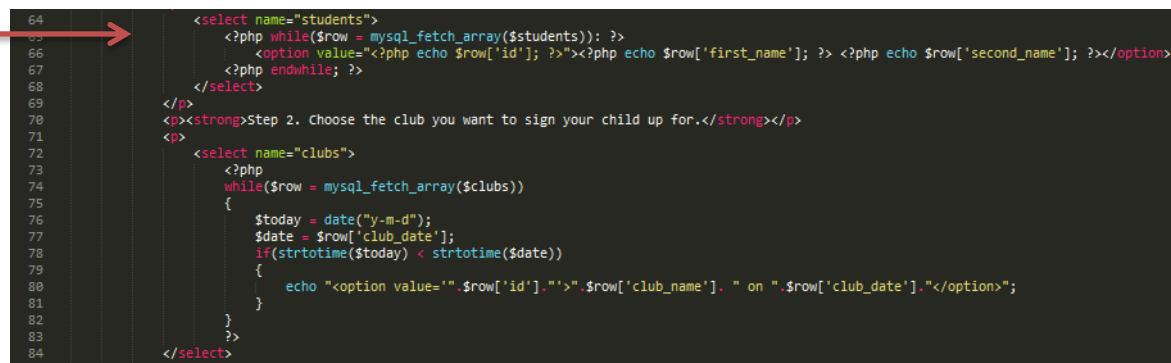
This page is the page where parents can sign their children up for an activity. There is two drop down boxes one contains a list of registered children belonging to the parent. The other drop down produces a list of available activities and what week number that the activity is happening in. this page is very basic because the idea is that it's meant to be easy for the user to complete the action.

4.2.8.2 External pages, functions and constants used

- Header.php
 - Config.php
 - Scr_check_user_level.php
- Footer.php

4.2.8.3 Code

NOTE: this is not all the code for the page; to save being repetitive I have included a snippet of code that does something a little different to what has already been shown.



```

64 <select name="students">
65 <?php while($row = mysql_fetch_array($students)): ?>
66 <option value="<?php echo $row['id']; ?>"><?php echo $row['first_name']; ?> <?php echo $row['second_name']; ?></option>
67 <?php endwhile; ?>
68 </select>
69 </p>
70 <p><strong>Step 2. Choose the club you want to sign your child up for.</strong></p>
71 <p>
72 <select name="clubs">
73 <?php
74 while($row = mysql_fetch_array($clubs))
75 {
76 $today = date("y-m-d");
77 $date = $row['club_date'];
78 if(strtotime($today) < strtotime($date))
79 {
80 echo "<option value='".$row['id']."'>".$row['club_name']. " on ".$row['club_date']."'</option>";
81 }
82 }
83 </select>
84 </p>

```

What I am doing here is creating a drop down box with the **<select>** tags, inside I'm doing something similar to what I did with the IF logic I am writing the code for the **while** statement and then ending it with a PHP tag. Then I have indented a line and I am echoing the ID, first name and second name of the students which would then appear as a single option in the drop down. I am then doing the same thing for the clubs, except I only want to show the ones that have dates that haven't expired.

4.2.9.4 Unforeseen problems

This was my most over looked problem in the project and in effect it set me back an entire calendar month. For a long time I had problems deciding how I was going to retrieve data regarding the student and the activities and then store them together. I tried storing multiple students in a single row for an activity which meant the database at the time was very poor. I had to do extra reading about the MySQL database management and learnt how it was a non-relational database. I then looked at work people had done in books which was storing multiple data about something in different tables. Having learnt this I finally had my solution for solving this particular problem. There have been no hardware or software problems with this development now it has been fixed properly.

4.2.9 Writing feedback

4.2.9.1 Description

On this page an instructor can see a list of activities that have already happened (expired) along with a list of attendees for the activity. When they click the ID of a particular ID they are redirected a page that has a form that takes the student ID and the feedback they want to give. (On this page the student ID is visible).

4.2.9.2 External pages, functions and constants used

Here are the external pages and functions for this page:

- Header.php
 - Config.php
 - Scr_check_user_level.php
- Enter_feedback.php
- Footer.php

4.2.9.3 Code

```
<?php $first = base64_encode($row['id']); ?>
```

This is the only line for the first part, what this does is it encodes the ID, this is only for security reasons. Because the way this works is one of PHP's global variables `$_GET` displays the variables in the URL which means the information is completely exposed to the user and they can alter this as they please. However, when encoding it it encodes it to random characters and in the file "enter_feedback.php" this line of code is used:

```
$club_id = base64_decode($_REQUEST['action']);
```

This then decodes the character string back to the original ID and this is then processed to retrieve the activity that the instructor wants to provide feedback for. After this the instructor enters the data and the `$_POST` data is inserted into the "tbl_instructor_feedback" table. This code is similar to what has explained before so it won't be repeated.

4.2.9.4 Unforeseen problems

Again, there are no unforeseen problems for this functionality, it is similar to what other features do. What I mean is that it's a form that takes post data and inserts it into different tables of the database. As well as development there have been no problems anywhere else or with testing.

4.2.10 Directions

4.2.10.1 Description

This is one of the systems advanced features. This is the page that shows directions to the activity based on where the user currently is. It works by taking the ID from the page “bookings_page.php” and passes it through to identify which activity it is. Down the left of the page is a panel that shows the directions next to the map.

4.2.10.2 External pages, functions and constants used

This is what is inside the directions:

- Header.php
 - Config.php
 - Scr_check_user_level.php
- Bookings_page.php
- Google Maps API
 - google.maps.DirectionsRenderer(rendererOptions)
 - google.maps.DirectionsService()
- Footer.php

4.2.10.3 Code :

```

5 <script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>
6 <style>
7 html, body {
8   height: 100%;
9   margin: 0;
10  padding: 0;
11 }
12
13 #map-canvas, #map_canvas {
14   height: 100%;
15 }
16
17 @media print {
18   html, body {
19     height: auto;
20   }
21
22   #map_canvas {
23     height: 650px;
24   }
25 }
26 </style>

```

The first line imports the Google Maps API.

The next part provides the design for the canvas. This has to be in the file otherwise the map won't know how to position and to what scale it should produce itself.

```

28 <body onload="initialize()">
29 <?php
30   require_once('scripts/config.php');
31   $id = base64_decode($_GET['id']);
32   $quer_home = mysql_query("SELECT latitude, longitude FROM tbl_users WHERE username = '".$_SESSION['username']."' " or die(mysql_error());
33   $query = mysql_query("SELECT 'tbl_club_location'.club_id, 'tbl_club_location'.latlng FROM tbl_club_location WHERE 'tbl_club_location'.club_id = '".$id."' " or die(mysql_error()));
34
35   while($row = mysql_fetch_array($query))
36   {
37     $latlng = $row['latlng'];
38   }
39
40   while($row = mysql_fetch_array($quer_home))
41   {
42     $lathome = $row['latitude'];
43     $longhome = $row['longitude'];
44   }
45
46   <input type="hidden" value="php echo $lathome.",".$longhome; ?" name="homelatlong" id="homelatlong" />
47   <input type="text" value="php echo $latlng; ?" id="latlong" name="latlong" />
48   <div id="map-canvas" style="float:right;width:50%; height:100%;></div>
49   <div id="directionsPanel" style="float:left;width:30%;height: 100%; font-family: Arial;">
50     <p>Total Distance: <span id="total"></span></p>
51   </div>
52 </script>

```

The body has an **onload()** attribute that runs a function call initialize.

After this the user's coordinates are called which are stored in the database. It runs this and uses the ID that was passed from "bookings_page.php" and these values are then stored into **input** fields of type **hidden**.

```

52 <script>
53 var rendererOptions = {
54   draggable: true
55 };
56 var directionsDisplay = new google.maps.DirectionsRenderer(rendererOptions);
57 var directionsService = new google.maps.DirectionsService();
58 var map;
59
60 var cardiff = new google.maps.LatLng(51.483521,-3.17997);
61
62 function initialize() {
63
64   var mapOptions = {
65     zoom: 7,
66     mapTypeId: google.maps.MapTypeId.ROADMAP,
67     center: cardiff
68   };
69   map = new google.maps.Map(document.getElementById("map-canvas"), mapOptions);
70   directionsDisplay.setMap(map);
71   directionsDisplay.setPanel(document.getElementById("directionsPanel"));
72
73   google.maps.event.addListener(directionsDisplay, 'directions_changed', function() {
74     computeTotalDistance(directionsDisplay.directions);
75   });
76   calcRoute();
77 }
78
79
80 function calcRoute() {
81   var home = document.getElementById('homeLatLng').value;
82   var latlong = document.getElementById('latlong').value;
83   var request = {
84     origin:home,
85     destination:latlong,
86     travelMode: google.maps.TravelMode.DRIVING
87   };
88   directionsService.route(request, function(response, status) {
89     if (status == google.maps.DirectionsStatus.OK) {
90       directionsDisplay.setDirections(response);
91     }
92   });
93 }
94
95 function computeTotalDistance(result) {
96   var total = 0;
97   var myroute = result.routes[0];
98   for (i = 0; i < myroute.legs.length; i++) {
99     total += myroute.legs[i].distance.value;
100   }
101   total = total / 1000;
102   document.getElementById("total").innerHTML = total + " km";
103 }
104 </script>
105 </body>
106 </html>

```

Firstly there are a few objects being created related to Google maps, the first one is for rendering the map and the second is for providing the directions which then makes the properties of each of these classes available to me the developer. Also it sets a variable called map with its latitude and longitude position as a default in case the map isn't provided latitude and longitude values by `calcRoute()`.

As I mentioned just previously, when the body element has been rendered by the browser it calls the initialise function. What this does is sets some properties for the map. Firstly how zoomed in the map will be, the type of map is by default a road map, then it gets the pre-sets for the map canvas and directions panel. Afterwards it calls the next function which is `calcRoute()`.

This function gets the values passed to the hidden field previously that were fetched from the database and stores the user's position in a variable called **home** and the destination in **latlong**. Then the travel mode is set which is driving in this case so road routes will be generated to the destination.

This function then computes the distance of the journey and the length and produces the result of this in the directions panel.

4.2.11 Transactions

4.2.11.1 Description

This page is for admin only, it allows them to login to the PayPal account and look at their transaction history, and they can then take these records and manually enter it so that the user has a record associated with their account.

4.2.11.2 External pages, functions and constants used

- Header.php
 - Config.php
 - Scr_check_user_level.php
- Footer.php

4.2.11.3 Code

No code is provided for this as it performs functionality that is the same as other pages that have already been talked through. This page performs a simple `mysql_query()` insert to the transactions table inside the database. It saves who the person was and their transaction ID.

4.2.12 Instant Messenger

4.2.12.1 Description

The idea of this page is for users to ask live questions to the admin of the system. It uses AJAX, PHP and CSS to create a conversation window like no other.

4.2.12.2 External pages, functions and constants used

For this there are a number of external functions and pages:

- Header.php
 - Config.php
 - Scr_check_user_level.php
- Chat.php
 - fetchMessages()
 - throwMessage()
- Init.php
- Chat.php – this file also exists in an AJAX folder
- Footer.php

4.2.13.3 Code

```

1 <?php
2 require("../core/init.php");
3 if(isset($_POST['method']) == true && empty($_POST['method']) == false)
4 {
5     $chat = new Chat();
6     $method = trim($_POST['method']);
7
8     if($method == "fetch")
9     {
10         //fetch message and output
11         $messages = $chat->fetchMessages();
12
13         if(empty($messages) == true)
14         {
15             echo "No messages in the chat";
16         }
17         else
18         {
19             foreach($messages as $message){
20                 >>
21                 <div class="message">
22                     <a href="#"><?php echo $message['username']; ?></a> says:
23                     <p><?php echo $message['message']; ?></p>
24                 </div>
25
26             <?php
27             }
28         }
29     }
30     else if($method == "throw" && isset($_POST['message']) == true)
31     {
32         //throw message into database
33         $message = trim($_POST['message']);
34         if(empty($message) == false)
35         {
36             //throw it
37             $chat->throwMessage($_SESSION['user'], $message);
38         }
39     }
40 }
41 ?>

```

This is Chat.php, this firstly checks to see if method is set, if it is it performs another check to see if method is set to "fetch" or "throw" fetch will called the fetchMessages() method which contains the query for selecting messages.

If it's set to throw then it means data is going to be inserted to the database.

```

1 <?php
2 class chat extends core
3 {
4     public function fetchMessages()
5     {
6         //query db and return rows
7         $this->query("
8             SELECT
9                 `tbl_chat`.`message`,
10                `tbl_users`.`username`,
11                `tbl_users`.`id`
12            FROM
13                `tbl_chat`
14            JOIN
15                `tbl_users`
16            ON
17                `tbl_chat`.`sender` = `tbl_users`.`id`
18            ORDER BY
19                `tbl_chat`.`timestamp`
20            DESC
21        ");
22        return $this->rows();
23    }
24
25     public function throwMessage($user_id, $message)
26     {
27         //insert into db
28         $this->query("
29             INSERT INTO `tbl_chat` (`sender`, `message`, `timestamp`)
30             VALUES ('.(int)$user_id.', '.$this->db->real_escape_string(htmlentities($message)).', NOW())
31         ");
32     }
33 }
34 ?>

```

This is what the class chat looks like. It has a property called **fetchMessages()** and another for throwing messages into the database.

```

1  var chat = {}
2
3  chat.fetchMessages = function()
4  {
5      $.ajax({
6          url: 'ajax/chat.php',
7          type: 'post',
8          data: {method: 'fetch'},
9          success: function(data){
10             $('#chat .messages').html(data);
11         }
12     });
13 }
14
15 chat.throwMessage = function(message){
16     if($.trim(message).length != 0)
17     {
18         $.ajax({
19             url: 'ajax/chat.php',
20             type: 'post',
21             data: {method: 'throw', message: message},
22             success: function(data){
23                 chat.fetchMessages();
24                 chat.entry.val('');
25             }
26         });
27     }
28 }
29
30 chat.entry = $('#chat .entry');
31 chat.entry.bind('keydown', function(e){
32     if(e.keyCode == 13 && e.shiftKey == false)
33     {
34         chat.throwMessage($(this).val());
35         e.preventDefault();
36     }
37 });
38
39 chat.interval = setInterval(chat.fetchMessages, 3000);
40 chat.fetchMessages();

```

This is the JavaScript that gets the functionality working for the feature.

Firstly, there is some AJAX that sets up three variables, one is the URL, the next is for the type of data which is post and then the method which is fetch. On success a function is then run chat and messages are executed.

The next part is for saving a message to the database. This is done similarly to above, except it throws the message and then fetches the messages so its up to date. After that it checks that the input field is set to nothing (no message).

The final part is for allowing the user to press the enter key to submit a message instead of going to a new line and at the very bottom is how often the script makes a request for to fetch messages which is 3000 milli seconds

4.2.13.3 Network activity

The screenshot shows a web browser displaying the MyECA website. The website has a sidebar with links: HOME, CLUB BOOKINGS, View Clubs, Sign Up To A Club, View Attendees, Create Club, Club Addresses, and FEEDBACK. The main content area has a chat interface with a text input and a '52 Enter chat?' button. Below the chat input, there's a section titled 'Ask a question Live to our MyECA Team' with a chat log showing messages from 'bloggsays' and 'Adminsays'.

The Firefox Network Monitor is open, showing a list of requests. The first request is 'GET instant_messenger.php' with a status of '200 OK' and a size of '3.1 KB'. The subsequent requests are 'GET layout.css', 'GET styles.css', 'GET jquery.js', 'GET chat.js', 'GET bootstrap.min.js', and 'GET bootstrap.min.css', all with a status of '304 Not Modified'. The final request is 'POST chat.php' with a status of '200 OK' and a size of '202 B'. The Network Monitor also shows a detailed view of the 'POST chat.php' request, displaying the request phases and their durations.

URL	Status	Domain	Size	Remote IP	Timeline
GET instant_messenger.php	200 OK	project.cs.cf.ac.uk	3.1 KB	131.251.168.115:443	96ms
GET layout.css	304 Not Modified	project.cs.cf.ac.uk	102 B	131.251.168.115:443	102ms
GET styles.css	304 Not Modified	project.cs.cf.ac.uk	308 B	131.251.168.115:443	100ms
GET jquery.js	304 Not Modified	project.cs.cf.ac.uk	259 KB	131.251.168.115:443	258ms
GET chat.js	304 Not Modified	project.cs.cf.ac.uk	731 B	131.251.168.115:443	267ms
GET bootstrap.min.js	304 Not Modified	project.cs.cf.ac.uk	27.9 KB	131.251.168.115:443	263ms
GET bootstrap.min.css	304 Not Modified	project.cs.cf.ac.uk	103.5 KB	131.251.168.115:443	130ms
POST chat.php	200 OK	project.cs.cf.ac.uk	202 B	131.251.168.115:443	10.2s
POST chat.php	200 OK	project.cs.cf.ac.uk	202 B	131.251.168.115:443	5.06s
POST chat.php	200 OK	project.cs.cf.ac.uk	202 B	131.251.168.115:443	112ms
POST chat.php	200 OK	project.cs.cf.ac.uk	202 B	131.251.168.115:443	105ms
POST chat.php	200 OK	project.cs.cf.ac.uk	202 B	131.251.168.115:443	85ms

This particular feature checks for new posts in intervals of five seconds, with Firebug installed as a plugin for Firefox, I can monitor the network activity of this particular feature. The first part with the line down the centre records the time taken to load each file external to the page such as the layout and the CSS. After this it makes its first call to the file “chat.php”.

By setting the JavaScript interval method to a low value like 2000 which is 2 seconds, this would mean every two seconds it performs a check to see if new messages are available. This wouldn't be practical if this product was used in a big place with more users and more activity as it would clog up the bandwidth. However by setting it as something higher it's not making requests as regularly.

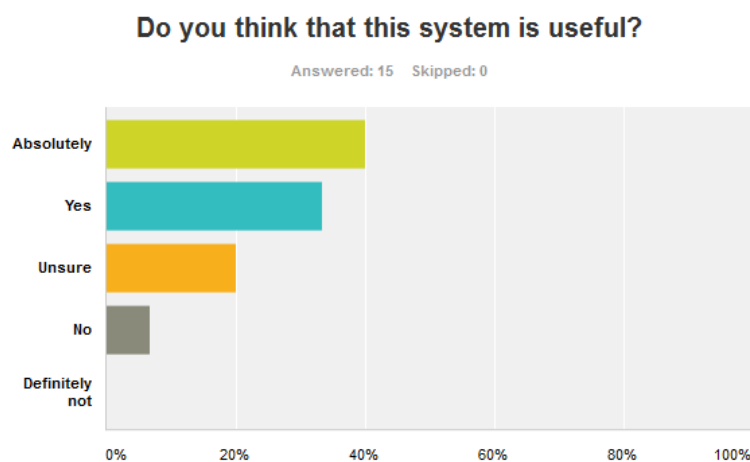
5.0 Results and Evaluation

In this part of the report I will document the results that I have found from conducting this project and also talk about the extent of the goals that I have reached and what I didn't reach. As well as goals I will be talking about the testing of the system that has been developed and what I did to test it and why I thought it was a suitable testing solution. After that I will go on to discuss if I think I've designed an adequate and reasonable solution to the initial project problem, I will bring into account the different tests that were carried out to support my justification.

5.1 Results

In the testing section of the project, I conducted a usability questionnaire to fifteen people based on how they found navigating and using the different functionalities on the system. I got a lot of mixed feedback which was very useful, it's always good to have good and bad feedback because it means you can reflect on different reactions and try to respond accordingly with it. In the testing stage all test cases that were created all passed, there were no failures for anything tested. A lot of things were for providing feedback to the user based on their input. I believe I have managed to design an exceptional solution to the initial project problem and I have created a system that performs all the core functionality that is a part of the requirements.

By looking at feedback and the results I have analysed in the testing section it leads me to think the system is good and is of great use to people.



The majority of the population agreed that the system is useful and other charts that I have analysed have all had positive feedback with little negative feedback which is mainly small constructive criticisms about the system. In the future work section of this report I talk about a feature that was suggested in the feedback to show that I am taking all views and opinions into account from the users.

In my testing report I mentioned that I did not have enough time to test everything about every piece of functionality, this is because some of the features and functionalities are very large and take a wide range of inputs. To be able to test everything I would have needed to start the phase a long time ago. Also I believe that testing the core functionalities of the system is perfectly adequate as they are the features that make the system what it is and without them fully functioning it would not be a working system. Overall I am very confident

I have built a good project system, I am filled with this confidence from a number of tests that I have done and adding features that make it exceed all expectations. Also being able to add a core functionality like the directions makes it very good because that feature in itself is an advanced feature.

5.1.2 Tests Conducted

5.1.2.1 Test Cases

For this project I have looked into four different types of testing which test cases to test data that is entered and how it is manipulated, it also allows me to isolate each feature from the system and enter data to see how it copes and what it does to the data. It tells me is the data stored? Or is the data sent back to the user and tells them to do something different with it.

5.1.2.2 Usability Testing

Usability testing tests how different users can use the developed system and help identify defects and potential problems that I the developer can't see. Instead of creating different personas for potential users and trying to fit myself into their mind set, it would be a lot easier and less time consuming to get test users to test the system. This is good because they have no idea what I have developed or what is going to be on the screen it also allows them to see things that I wouldn't have, because in my viewpoint I have developed it and think it can't be improved.

5.1.2.3 Performance Testing

Thirdly, performance testing to see how well the system can deal with different volumes of traffic and data usage and looking at how quick it can render pages and external scripts. In the testing report I was unfortunately able to justify why this type of testing wasn't necessary for this project, but at the same time I attempted to outline the importance of why it would be needed if the system was a lot bigger and would have to deal with larger amounts of traffic. The results that you could get from this would allow me try and develop more efficient scripts, for example Object Oriented Programming may improve speed up of some scripts.

5.1.2.4 Browser Testing

Finally, browser testing which is one of the more important tests that I conducted; this allowed me to see how the system looks and performs in different browser environments. This type of testing in particular is important because people everywhere user different browsers for their own reasons to it's important to check that the only way of accessing this system which is a web browser, works for all kinds. The results I would get would then allow me to go back and go through CSS to try and adapt it to work in browsers if the design wasn't quite working the way it should.

5.2 Evaluation

Having conducted different types of tests on my system I am now able to see just how good it is. Before testing every developer thinks they have done everything they could have and it can't be improved, however for me it has allowed me to see different avenues that I haven't considered before. For example a user suggested a mobile application which isn't something I would have considered as a future development for the project. Also, I have noticed trends in feedback that people had given me; I have combined all feedback and created another tag cloud using TagCrowd.

(Crowd, 2013)



As you can see from the TagCloud, the word change was used in a lot of people's feedback. Also looking to the left of the image you can see the word "Google" from where people had talked about the Google Maps feature as being one of their favourite features in the system. Also, uses of the word smart, clever and stylish are all compliments towards the systems design. Also if you refer to Appendix A and you see the charts that have been generated from the questionnaire, the majority of the population have left high positive feedback.

However, the word change is the most heavily used word in the feedback, a lot of people had criticisms on things they would like to see done differently which is obviously a weakness to the system. Also, the negative feedback I have received has been good; I haven't had feedback that has come with no justification of why it's negative.

Overall, the methodology that was used which is Waterfall Model is one that worked very well for the project and also for me personally. This is because that I was able to carry out each stage of the project and complete it before carrying on to the next stage. I think that carrying out the project this way worked in my advantage because it gave me time on each part to plan in advanced accordingly before doing anything.

6.0 Future Work

6.1 End of the project

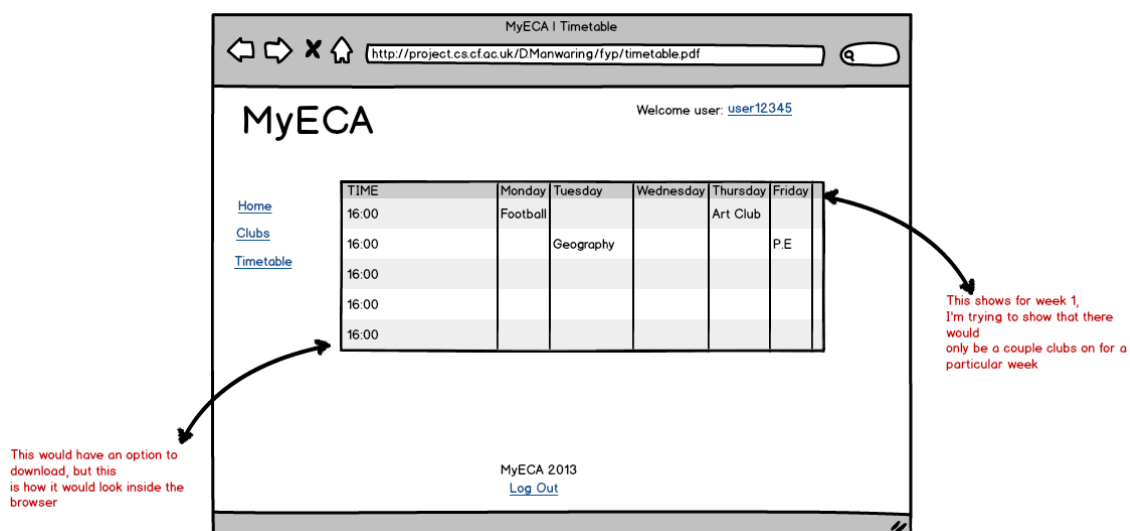
Upon reaching the end of the project it was natural for me to automatically reflect on what I had just completed and what I hadn't finished which is natural for any developer in a project. At the end of the project I had all core functionalities implemented except for one which was generating a timetable of the activities for a week into a PDF document. All other features and advanced features that were designed and carefully analysed were implemented. Also, in regards to the design there is nothing that needed to be styled in any particular way to match the design pattern of the system; this was all done at the time when the system went under a redesign.

6.2 Future work

Now I will begin talking about the future work that can be done to the system, and for this I don't just mean about talking about implementation, but as a business perspective look at the means for marketing and trying to sell this as a product in its respected market.

6.2.1 Timetable to a PDF file

As I mentioned in 6.1, I was unable to get one of the features working which was the PDF timetable. This is something that definitely needed more time, in reflection it's something I should have looked at more carefully and aloud more time for and perhaps dropped a feature that wasn't as necessary or not detrimental to the system. This would certainly be a useful feature because this system is all completely internet based, there may be times where the systems users aren't able to connect to the internet so it would be good to have an offline hardcopy of a timetable of activities for the present week so that they can view it and make future plans based on the document. Also, this feature would be able to display nicely all the activities and have them split into Monday-Friday and their times etc. Here is what I would envisage the feature to look like in a future development using Balsamiq mark-up as a design:



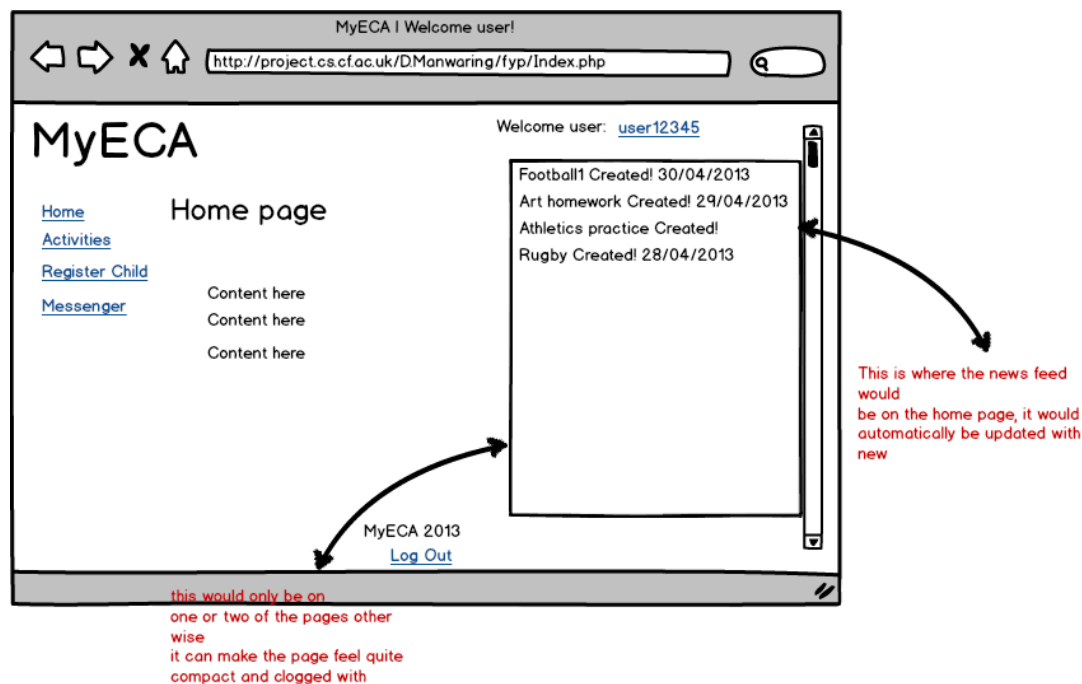
In the above is how I would like the particular feature to look once it had got working. Also this is how I would like to think future developers would pick up after where I finished. It's a very straight forward and easy to follow time table structure, I would not want this to over

complicate things for the systems users as there may be some users that are computer illiterate.

An example scenario of this would be for a user to lose their internet connection for a prolonged period of time. Luckily though the user has saved a PDF copy of the time table for the current week(s) and is able to view activities for their child and see when and where they are.

6.2.2 News Feed

This would be very useful for people that don't access the system as often as other users do. What this would allow is for the users to be notified of when a new activity has been added and for there to be a timeline style section of their interface of when the activity was added and what it was. Here is an illustration using Balsamiq to further the point I'm making:



As you can see on the right of the prototype screen is where I envisage the feature would appear, as I have annotated I would not want this feature to appear on all the screens because from the beginning I have wanted something clean cut and very straight forward. The idea of this feature would be to notify its users of newly created activities, this stops users constantly having to access the "bookings_page.php" to see if new activities are accessed. The good thing about this feature is that it would be created using AJAX which means it wouldn't require the page to be refreshed constantly.

So a scenario would be for a user to leave their workstation for an extended period of time (without their session being timed out) and be able to return to the workstation without touching it and be able to see there are new activities.

6.2.3 Shopping Cart

In the system I have a payment system that uses PayPal sandbox which basically simulates metaphorical payments which means that no money is actually sent anywhere. In future developments I would like for there to be a feature where parents would browse the current activities and if they see multiple activities that require a fee that they would like to sign their child up for, they can click “add to cart” and it would save inside their session activities that they have added to this cart. This would allow for multiple activities to be paid for inside of a single transaction which would be more efficient than singular transactions for multiple activities. For this I have also created another prototype of what the feature would look like running in the system:

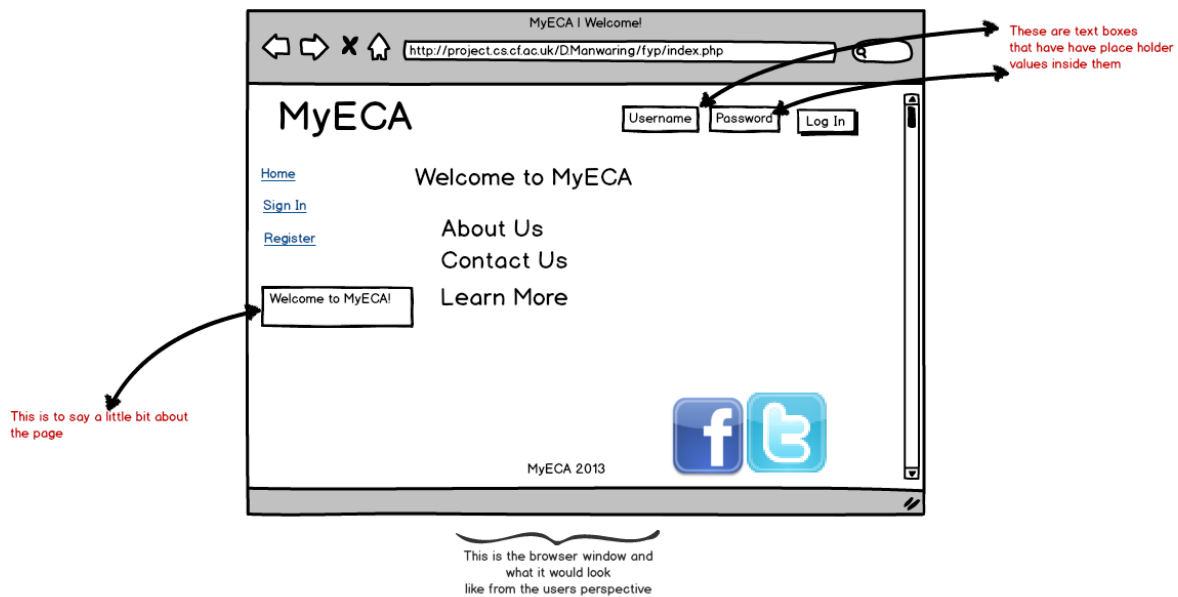


As you can see multiple activities have been temporarily saved until the user “checks out”.

A scenario of this would a user seeing three different activities that they would like to sign up a child for, they don’t have much time to do it and it needs to be processed quickly. The user would “add to cart” the activities that they want to pay for, and then when checking out they would pay for all activities at once.

6.2.4 Social Media

For this part I am not looking so much as to what I can implement, I am more concerned at looking at web marketing which is extremely important for getting the system on the market and getting used by different customers and clients. Since I.T is progressing at a very fast pace it’s important that marketing strategies are kept up to date, since search engine Giants such as Google and social networks like Facebook and Twitter these are then possible means of pushing the system and promoting it. For this next future work idea, I propose that having Facebook and Twitter links on the page, and having this link for potential customers (schools) to be able to reach the developers personally and learn a little more about the system. By having this as a feature means that we are able to keep up with our customers and potential clients. This type of advertising is cheap and fast and is an effective business strategy for promoting this product. Here is what it would look like on one of the pages of the system:



It is very common in the present day to see websites with these logos on them because they are so widely recognised people know what they are and are enticed to clicking on them to see what more they can find out.

These features are all things that I have considered having over features that I have opted to have in the current version of the system. The reason I chose to use them is that I felt their presence was far more needed than these. My only mistake is that I didn't allow enough time for the time table PDF file, if I had implemented that in the time I had for the project then I would have perhaps had another feature to consider and then progress to discuss inside this report.

6.2.5 Mobile Application

Make a version that can be supported for mobile as I don't own a PC
5/3/2013 1:09 AM [View respondent's answers](#)

Having looked at the results of the MyECA usability feedback questionnaire there was a very good suggestion from a user to create a mobile application. Looking at things in the long run if this product was to go Live, having a mobile version of the system would expand the intended audience because some people work off their phones more than computers. I have created another mock-up in Balsamiq of what this application may look like.



As you can see the content is close up to the screen and not far away as it would be on the desktop site. By having it like this it would hugely improve usability for mobile users.

7.0 Conclusions

The aims of this project were to produce an online booking system that supported functionality for activity registration, activity feedback and a payment system as its core functionalities. On top of this it was down to me to add miscellaneous functionality which was the advanced features to the system. En-completion of this project I have produced a system with all core functionality except for a problem that was addressed which was producing a PDF document, other than this I have managed to successfully singlehandedly product a system worthy to be amongst competitors in its respective market.

In my opinion I did overlook parts of the project that I thought would be easier than others, this was a mistake because in a project things should be treated with the same severity and that isn't something that I did and it did prove costly when I fell behind because I didn't allow enough time for the development.

Having developed this system I have furthered my web technology skill set. I went out of my comfort zone into AJAX to explore what functionalities I could use using this technology and how I can incorporate it to the current features. Also with MySQL I had never had much personal experience with it, but with this I looked at this like Aliasing, performing different joins and other built in functions to the syntax.

Looking back to my initial plan, this is a table that shows my project aims and objectives, if I have met it and how.

Aim	Achieved?	How?
I aim to carefully plan every approach that I take with this project, this includes mostly when it comes to the implementation.	Yes	Every feature was researched and planned individually before implementing.
All ideas on my behalf are fully implemented	No	I was unable to implement all functionalities due to being over ambitious and not allowing enough time
To create a fully functioning system worthy of any school	Yes	Feedback that I have received from usability testing indicates people are happy with the system and find it useful for its purpose
To implement a login system for the users	Yes	There is a login system in place.
A messaging service for the different user groups to keep in contact	Yes	There is an instant messenger in place.
Administrators will add/edit/remove activities	Yes	This is currently in place on the system
Instructors will view attendance and add additional notes.	Yes	They can do this and there is a feedback feature in place.

Parents will register their children to the system and sign them up	Yes	This feature exists on the user home page.
There will be payment system	Yes	The PayPal sandbox simulates the payments for activities in my system.
To extend my knowledge in web based languages such as HTML, CSS, XML, PHP, MySQL.	Yes	I have extended my skill set for each of the technologies and even learnt about others such as AJAX.
To stick to the realistic deadlines that I set and meet each of them without fail.	No	I didn't set realistic deadlines and meant I fell an entire month behind for not allowing enough time.

8.0 Reflection

Now that I am at the end of the project I have had time to look back on the last nine months at what I did. Firstly, my time management skills have vastly improved, when dealing with a project to this sort of scale you have to manage time on the project and work separate to this, because this project is worth so much you always have to be that little more lenient to the project. Also, arranging and scheduling meetings around other commitments falls under this category.

Communication skills are another area that has been furthered and improved. When in a project like this you need to be in contact with whoever is supervising you, they need to know what stage the project is at, what you're doing and when is it going to be completed by. This means communication methods such as email and face to face meetings are of great importance. I found out for myself that the most effective tool I had were the meetings I had, I was able to pick the brains of my supervisor and retrieve a lot of information and then take that away and work with what I had.

The project that I opted to do was something of great interest to me. The fact that I enjoyed the project as it progressed was certainly a project strength, if it was something that I didn't enjoy there are aspects that would have been challenging and would have been less inclined to do it.

In my time of undertaking this project, not only have I been able to further my skillset but I have been able to educate myself about new technologies that are available. For instance I was able to learn about AJAX and the different methods that it has as a part of its library. Other than the practical I was unable to understand the theory and be able to apply it to new solutions that were adapted to how I wanted the system to function.

Also, other than all the positive parts about the project, I have encountered problems along the way and had to use skills such as communication and time management between myself and my supervisor to try and work around problems I encountered. Apart from working with the problems, conquering them is another part of my project experience, there were areas in the project where I struggled but I proved that by showing hard work and perseverance I can conquer any problem just by tackling it with the skills I've already been taught and shown.

Before starting, I already had a number of technologies under my belt as a part of my skill set. I was comfortable and familiar with PHP, MySQL, HTML and CSS etc. through my own personal experience. By tackling problems in the project it meant I had to research new avenues and functionalities that I did not know which meant my current knowledge was being furthered and strengthened to quite some extent.

From this project I am now looking at developing my own Object Oriented applications, which is something I wasn't familiar or comfortable at the beginning of the project. Also, having completed this project I am now looking to take my skills into the work place and become a web developer.

Bibliography

Crowd, T., 2013. *TagCrowd*. [Online]

Available at: <http://tagcrowd.com/>

[Accessed 03 05 2013].

Nielsen, J., 1995. *nngroup*. [Online]

Available at: <http://www.nngroup.com/articles/ten-usability-heuristics/>

[Accessed 03 05 2013].

PHP, 2013. *PHP change log*. [Online]

Available at: <http://php.net/ChangeLog-5.php>

[Accessed 02 05 3013].

Twitter, 2012. *Twitter Bootstrap*. [Online]

Available at: <http://twitter.github.io/bootstrap/>

[Accessed 03 05 2013].