

## Criterion A: Planning

Word count: 361 (Criteria excluded in word count)

### 1) Defining the problem

The client, Mr McAleavy, is a secondary school teacher with a busy schedule. Mr McAleavy's job, apart from regular lessons, he runs an "Electrical engineering" club for the kids from primary school. During this club Mr McAleavy teaches the students the basics of electricity and programming, with the use of Microbits and Arduinos. This however is a tedious task and paired with Mr McAleavy's busy schedule he does not always have time for the club. Which is a shame, as the children that attend are very satisfied with the club and enjoy working with little electronics projects.

Upon my inquiry, Mr McAleavy contracted me, with creating and developing a website which allows him to assign tasks to students when he is not able to attend a session. The website was requested to be made with an intuitive and easy-to-use (user friendly) user experience, so that anyone can use it. Mr McAleavy has accepted my proposal for this and agreed to be my advisor. I will be writing this site using HTML, PHP, CSS and MySQL.

### 2) Rationale for the solution

The point of the site/program is to allow for an intuitive platform that allows my client to post tasks for students. This should be easy to use both for my client as well as the students. Pages will be interactive with blog type entry. The Teacher account will be able to set tasks to specific groups that will receive email notifications with a need for submission of confirmation that the task is complete. This is a good way for my client to send tasks for anticipating students even if he is not able to conduct a lesson himself. This can be adapted to any tasks, it's an open canvas and allows for many possibilities. The database will be able to conduct complex queries that would pull data from other tables for an analysis of each students' performance, how many tasks they have completed and how many they have failed.

The reason I chose to use the aforementioned programming languages, as I feel that a combination of them allows for a very sophisticated site with many possibilities for complexities. The programmes when used together can bring a very refined result.

### 3) Criteria for success

1. Students will be able to register and login their own unique accounts. This process should be secured through means of hashing.
2. A password recovery procedure for students (Request verified using an e-mail).
3. An ability for students to change classes and upload a profile image.
4. Allow the client account to set tasks to students within certain classes.
5. Ability for students to see all the tasks assigned to them (displaying key information) and the ability to mark a task as completed.
6. A task approval page, allowing the client to approve or reject submission requests.
7. Allow the client to access records with all past notices of students.
8. Allow the client to create reports based on specific students' performances.
9. Allow the client to activate and deactivate tasks.
10. The teacher portal cannot not be accessible by a student account
11. The website design should be simple and elegant. (Intuitive)

## Criterion B: Design

Site structure:

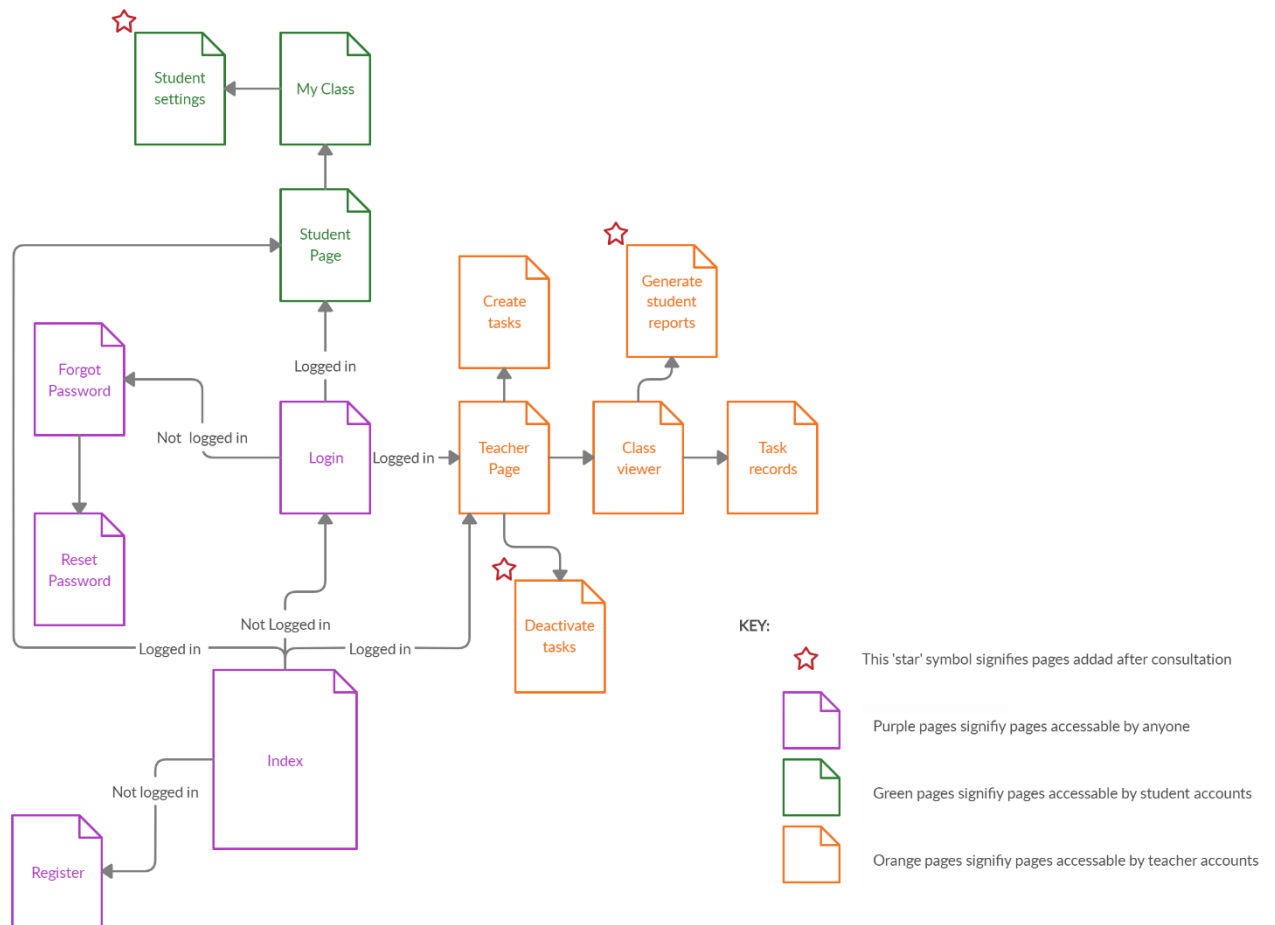


Table structure for database and data structures

***PK – Primary key***  
***FK – Foreign key***  
***CK – Composite***

**Users:**

	Name	Type	Default	Extra
<b>PK</b>	id	int(11)	None	AUTO_INCREMENT
<b>FK</b>	classid	int(11)	0	
	firstname	varchar(100)	None	
	lastname	varchar(100)	None	
	email	varchar(100)	None	
	password	varchar(50)	None	
	picture	varchar(150)	0.jpg	<i>filepath(user_uploads/)</i>

**Teacher:**

	Name	Type	Default	Extra
<b>PK</b>	teacherid	int(11)	None	AUTO_INCREMENT
	First Name	int(11)	None	
	Last Name	varchar(50)	None	
	Initials	varchar(50)	None	
	Email	varchar(50)	None	
	Password	varchar(50)	None	

**Class:**

	Name	Type	Default	Extra
<b>PK</b>	classid	int(11)	None	AUTO_INCREMENT
<b>FK</b>	teacherid	int(11)	None	
	classname	varchar(110)	None	
	classdesc	text	None	

**Student task:**

	Name	Type	Default	Extra
<b>CK, FK</b>	id	int(11)	None	
<b>CK, FK</b>	taskid	int(11)	None	
	status	smallint(1)	0	

**Task:**

	Name	Type	Default	Extra
<b>PK</b>	taskid	int(11)	None	AUTO_INCREMENT
	date_set	datetime	None	
	date_due	datetime	None	
	task_desc	longtext	None	
	task_name	longtext	None	

**Password Reset:**

	Name	Type	Default	Extra
<b>PK</b>	request_id	int(11)	None	AUTO_INCREMENT
	email	varchar	None	
	token	varchar	None	

## Database Normalisation (UNF TO 3NF)

The normalisation of the database is a rigorous process. Sample data is used to illustrate the changes that take place to the tables during the three normalisation forms.

### UNF (Unnormalised form)

teacherid	First Name	Last Name	Email	Password	Initials	id	firstname	lastname	email	status
1	Joe	Ford	j.ford@gmail.com	*****	JF	89,12	Jeff, Joe	Deer, Sponge	jd@gmail.com, joe.sp@outlook.com	0,1

extension,

password	picture	classid	classname	classdesc	taskid	date_set	date_due	task_name	task_desc
*****, *****	picture.jpg, profile.png	1,2	Microbit, Robotics	Microbit club, Robotics club	12, 22	07/08 /2020 , 06/08 /2020	09/08 /2019, 22/08 /2020	Microbit task , LEGO Mindstorms write-up	Complete Microbit task, Finish writing

### 1 N.F (First normal form)

To attain to the criteria of the first normalisation form, all cells that contain multiple results must be separated. This is shown in the tables below, when the columns that contained comma separated data has been split.

#### Teachers

teacherid	First Name	Last Name	Email	Password	Initials
1	Joe	Ford	j.ford@gmail.com	*****	JF

#### Users

id	firstname	lastname	email	password	picture	classid	classname	classdesc	taskid
89	Jeff	Deer	jd@gmail.com	*****	picture.jpg	1	Microbit	Microbit club	12

extension,

date set	date due	task name	task desc	status
07/08 /2020	09/08 /2019	Microbit task	Complete Microbit task	0

id	firstname	lastname	email	password	picture	classid	classname	classdesc	taskid
12	Joe	Sponge	joe.sp@outlook.com	*****	profile.png	2	Robotics	Robotics club	22

extension,

date set	date due	task name	task desc	Status
06/08 /2020	22/08 /2020	LEGO Mindstorms write-up	Finish writing	1

## 2 N.F (Second normal form)

To attain to the second normal form, every non key attribute is fully functionally dependent on the primary key<sup>1</sup>. As shown below: (Keys shown in bolds)

### Teacher

<b>teacherid</b>	First Name	Last Name	Email	Password	Initials
1	Joe	Ford	j.ford@gmail.com	*****	JF

### Class

<b>classid</b>	classname	classdesc
1	Microbit	Microbit club

### Task

<b>taskid</b>	date set	date due	task name	task desc	id	status
12	07/08/2020	09/08/2019	Microbit task	Complete Microbit task	12	1

### Users

<b>id</b>	firstname	lastname	email	password	picture
12	Joe	Sponge	joe.sp@outlook.com	*****	profile.png

**The other sample-data was excluded, for ease of visualisation.**

## 3 N.F (Third Normal Form)

A relation is in third normal form if it is in 2NF and no non key attribute is transitively dependent on the primary key.<sup>2</sup>

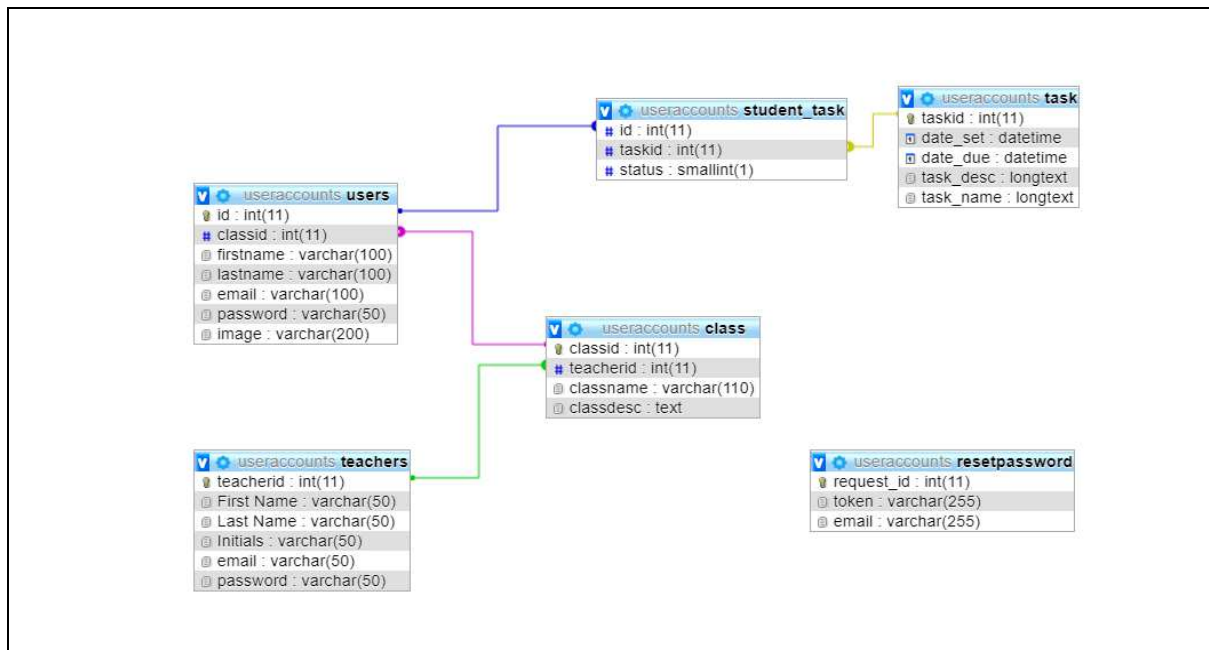
**The third normal form table is found in ‘ Table structure for database and data structures ’.**

---

<sup>1</sup> Bleisch, Susanne. “Second Normal Form (2NF).” Gitta.Info, 2013, [www.gitta.info/LogicModelin/en/html/DataConsiten\\_Norm2NF.html](http://www.gitta.info/LogicModelin/en/html/DataConsiten_Norm2NF.html). Accessed 8 Mar. 2020.

<sup>2</sup> Ibid.

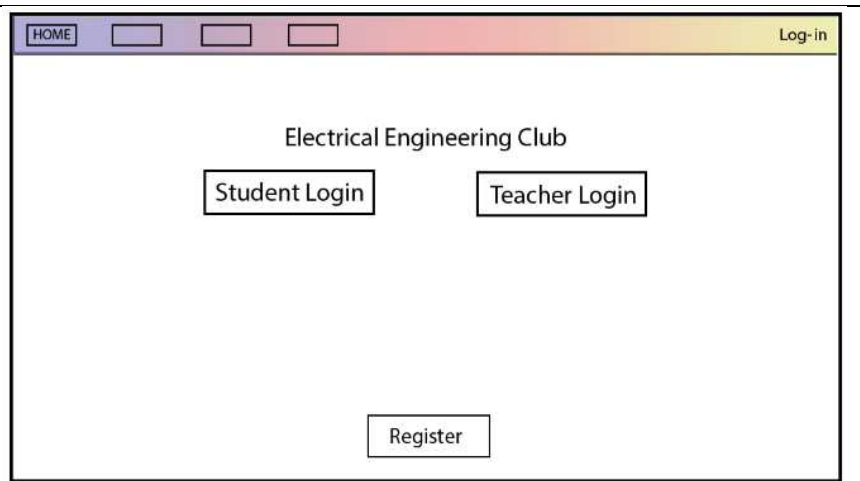
## Database Entity Relationship (3NF)



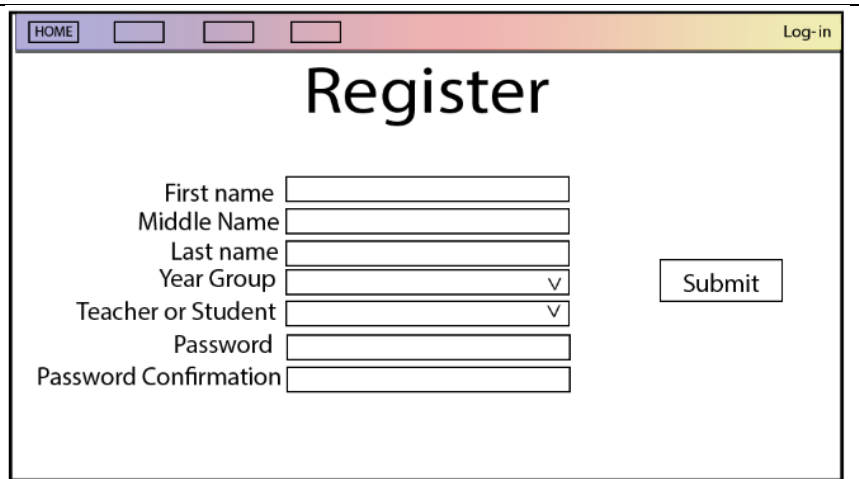
## User Interface mock

Using Adobe photoshop, I was able to create a mock site, using conceptual elements that were presented to the client, this whole process is documented in the email found at the end of this document under Figure. 1. Based on the feedback received from this mock-up clear changes are seen in the final product.

This page acts as the index page of the entire site, it has several redirects in the navigation bar to select pages.

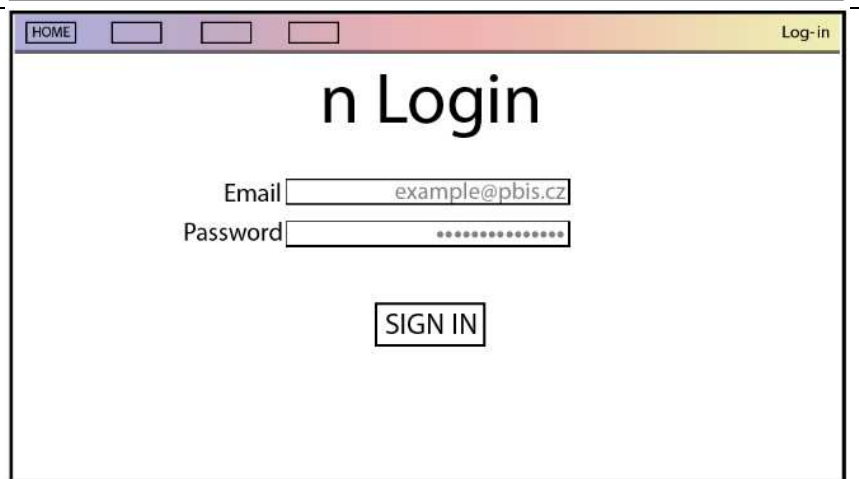


The register page is a page where the student and the teacher can make an account.




A browser window mockup for a registration page. The title bar shows 'HOME' on the left and 'Log-in' on the right. The main heading is 'Register'. Below the heading are several form fields: 'First name', 'Middle Name', 'Last name', 'Year Group' (with a dropdown arrow), 'Teacher or Student' (with a dropdown arrow), 'Password', and 'Password Confirmation'. A 'Submit' button is located to the right of the form fields.

There will be two separate login pages one for the student and one for the teacher.



A browser window mockup for a login page. The title bar shows 'HOME' on the left and 'Log-in' on the right. The main heading is 'n Login'. Below the heading are two form fields: 'Email' (with the text 'example@pbis.cz') and 'Password' (with a masked password '.....'). A 'SIGN IN' button is located below the form fields.

The main student page will have a table with all tasks with set criteria.

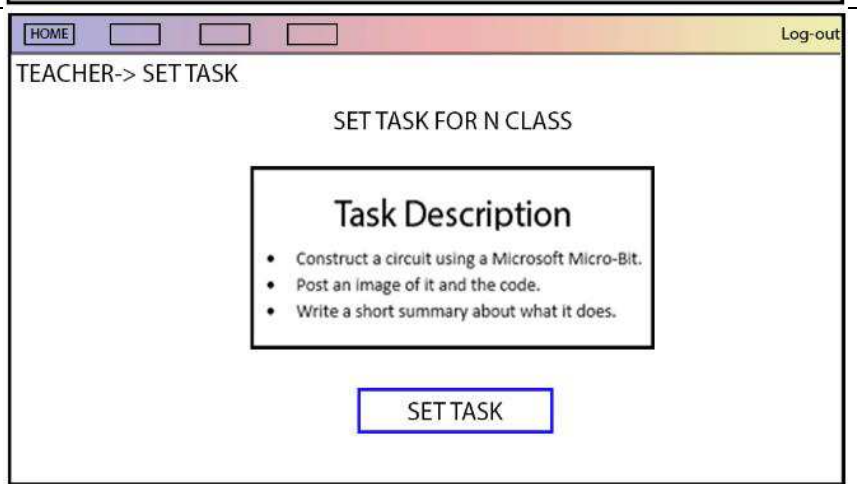


A browser window mockup for a main student page. The title bar shows 'HOME' on the left and 'Log-out' on the right. The main heading is 'TASKS'. Below the heading is a table with 6 rows and 1 column. In the bottom left corner, there is a 'MESSAGES' section with 4 rows and 1 column.

In the student page the student will be able to see the set task and when it is due.



Inside the teacher account the teacher will be able to set tasks for classes that they have selected.

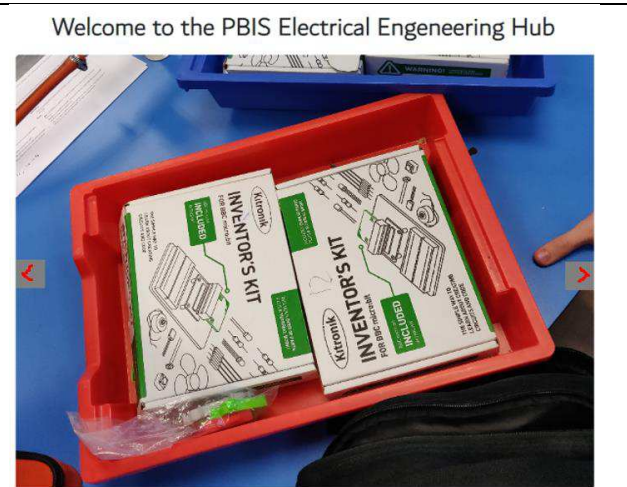


### Updated User Interface

Due to the positive reception to the navigation bar. It will always be placed at the top of the screen with buttons that when clicked will redirect to specific pages.



Circulating image reel. This creates a more appealing design and makes the website feel more personal. (Criterion 11)





As requested by the client the login page still is separated between the student and teacher logins.

LOGIN

Account Email

Password

[Forgotten password?](#)

LOGIN

As was requested by the client, the teacher account creation feature was removed from the front end of the site.

NEW ACCOUNT

First name

Last name

Email Address

Password

Confirm Password

Continue

As per the specifications of the client, the task viewer will be moved to the index of the student page, showing all the necessary details.

STUDENT PAGE

Current Tasks:

Task Name	Task Description	Date Set	Date Due	Action	Status
-----------	------------------	----------	----------	--------	--------

As per the client's request, more options were added to the task creation site more options were added.

NEW TASK

Task Name

Task Description

Date Set

Date Due

Select Class

Select your class

Submit

## Test Plan

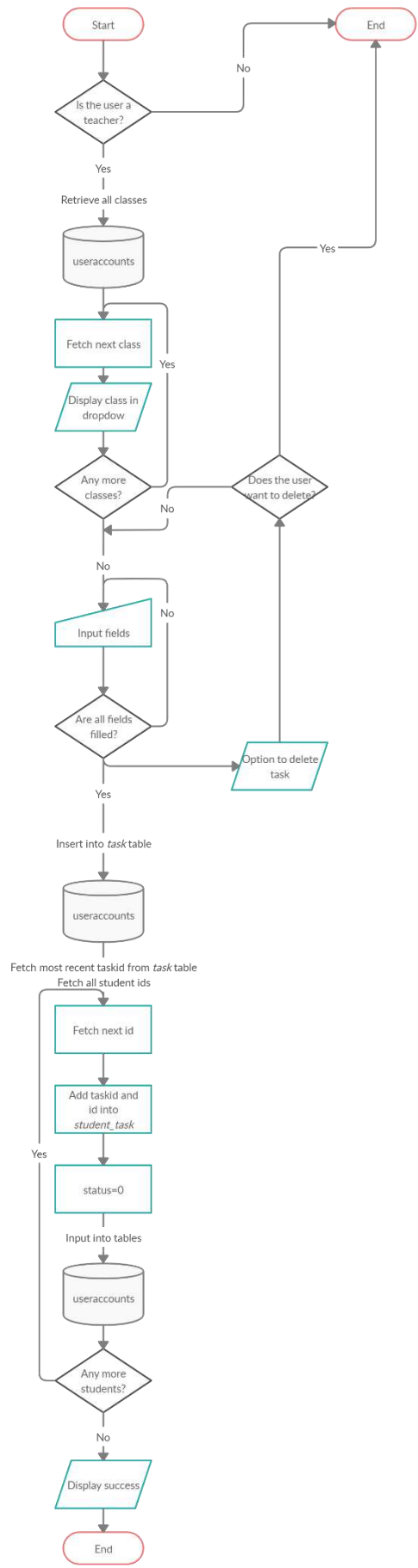
Test Number	Test description	Test data	Expected Outcome
1	Requirement 1: Email field in the registration page must contain an '@' sign.	Press submit button with an email that doesn't contain an '@' sign.	Submission failure, an error shows up displaying a need for an '@' sign in field.
2	Requirement 1: All fields within the register page must be filled.	Press submit button without all the fields being filled out.	Submission failure, an error shows up displaying a need for all fields to be filled.
3	Requirement 1: There can only be one account per email address.	Press submit button with an email that already has an account connected to it.	Submission failure, an error shows up displaying that an account already exists for that email address.
4	Requirement 1: Password must be between 6 and 15 characters long.	Type a password that is less than 6 then more than 15 characters long.	Submission failure, an error shows up displaying that the password requirements are not met.
5	Requirement 1: Password and confirm password must match.	Type a different password into the <i>password</i> and <i>confirm password</i> fields.	Submission failure, an error shows up displaying that the passwords do not match.
5	Requirement 1: All passwords are hashed.	When an account is created the password must be encrypted.	An encrypted <i>password</i> string is stored in the database
6	Requirement 2: Password recovery system will only work for accounts that exist	A forgotten password link will only be sent to emails that exist in the database.	An email will not be sent to an email that does not have a corresponding account on the site.
7	Requirement 2: Password recovery link is sent to destined email.	A password recovery link will be sent to an email that is found in the database.	A forgotten password email is sent to the email filed in a request.
8	Requirement 2: Password recovery system will contain a unique authentication token for each request.	When a password change is requested, each request will have a unique token.	No two requests will have the same authentication token.

9	Requirement 2: The password can successfully be changed.	The new password is processed and hashed.	The password will successfully change. (As shown by two differing hashes)
10	Requirement 3: The class can successfully be changed.	The student can change their class within their settings page.	The student's class will change after the designated class is selected from a dropdown menu.
11	Requirement 3: The profile picture can be changed.	The student can change their profile picture.	The default profile picture is changed to the new custom one that is uploaded by the user.
12	Requirement 4: Allow the client account to set tasks to students within certain classes.	The <i>teacher</i> account is used to create a new task.	The task created should show up in the <i>task</i> table, then all of the students that are assigned to the selected class should receive the tasks within the <i>student_task</i> table.
13	Requirement 4: Make sure that the due date cannot be earlier than the current time.	Using a drop down, an attempt to select an earlier date is attempted.	All previous times should become greyed (unclickable) out.
14	Requirement 5: Ability for students to see all the tasks assigned to them. (Option to mark a task as completed)	One task is set for the class that the test student is in.	The task should show up on the student index page. When the completed button is clicked the status of the task should change from 0 to 1.
15	Requirement 6: A task approval page, allowing the client to approve or reject submission requests.	The approval request from the student should appear on the conformation page.	The teacher should be able to either <i>accept</i> or <i>reject</i> the submission. If it is accepted the <i>status</i> should change to <b>2</b> , if it is rejected then it should change the <i>status</i> to <b>0</b> .
16	Requirement 6: The client should be able to apply a failing condition when a task is not submitted on time.	The teacher applies a failing condition to a task.	The <i>status</i> of the task should turn from <b>Pending</b> into <b>Failed condition</b> .

17	Requirement 7: Allow the client to access records with all past notices of students.	The client should see all past records when logged into the <i>teacher</i> account.	The teacher should be able to see all the past tasks with the status shown clearly.
18	Requirement 8: Allow the client to create reports based on specific students' performances.	The client generates a student report while logged into the <i>teacher</i> account.	The client should generate reports for a selected student.
19	Requirement 9: Allow the client to activate and deactivate tasks.	The <i>teacher</i> account will deactivate then reactivate a select task.	The <i>status</i> of the task when deactivated should change to <b>3</b> , then when reactivated the <i>status</i> should change to <b>0</b> .
20	Requirement 10: The teacher portal cannot not be accessible by a student account	The teacher page URL is typed into the address bar.	The student should be redirected to their profile page if such request is made using a student.

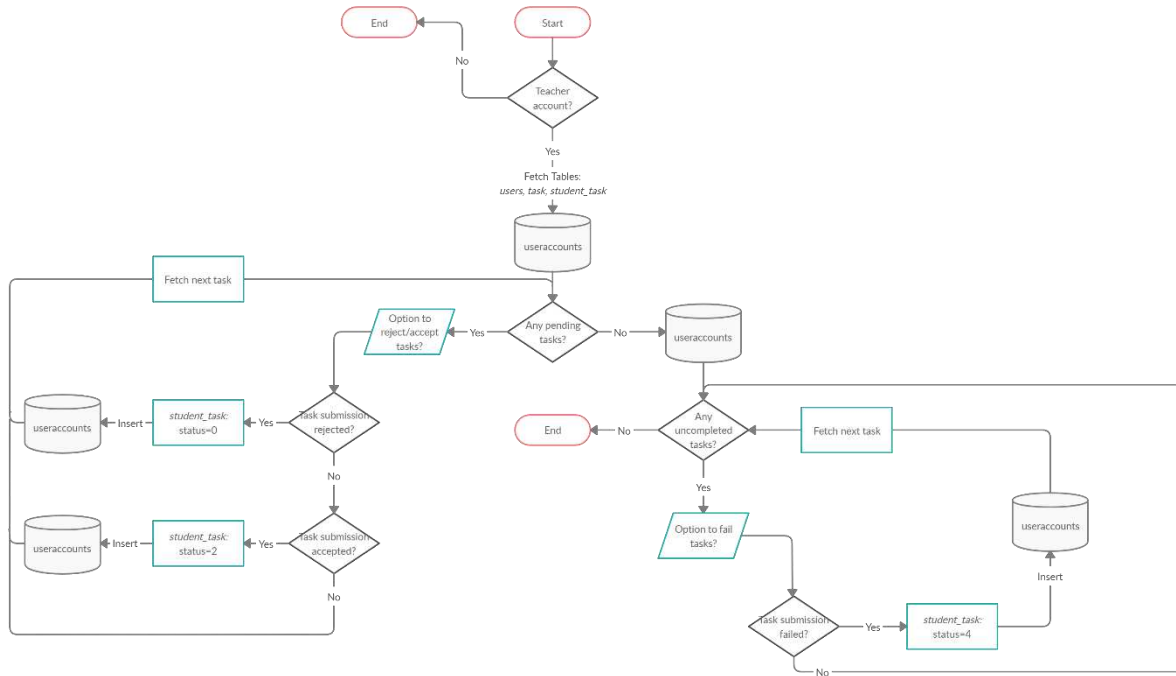
Flowchart for the task creation procedure

The flowchart below shows the process that the site and server manage in order to set a new task for a class.



## Flowchart for the task manager procedure

The flowchart below shows the process that the site and server manage showing all tasks on teacher page.



## Criterion B: Record of tasks

Date	Action	Details	Comments / Notes	Date Completed	Criterion
May 2019	Decision on project	I am part of the club and often the supervising teacher is not able to be there.	Decided to start to compose my potential proposal for my potential client. (2-3 Days)	14 <sup>th</sup> May 2019	A
May 2019	Discussion with the client about the potential project. And discussion with supervisor Mr O'niell.	I had a meeting with the Electrical Engineering club coordinator where I showed my analysis of the problem and my drafted solution.	The teacher told me that this indeed is a good idea, however he still has to ask his superiors whether such a solution can be accepted.	20 <sup>h</sup> May 2019	A
May 2019	Discussion about final ideas and criterium attained during meeting with the client.	The client has outlined all the necessary features and criteria that he believes are necessary for the product to become useful.	A follow up e-mail from the client allowed for a clear understanding of the success criteria and clarifying elements of the project.	28 <sup>th</sup> May 2019	A
May/ June 2019	Research into elements needed for the solution.	Due to the necessary complexity, for the web-based project I had to research and watch a lot of introductory videos into PHP and MySQL.	Following this, some features had to be altered with the client due to the limitations of my knowledge of all the tools that would be necessary to attain to all the projected features.	~12 <sup>th</sup> June 2019	A
June 2019	Setting schedule for the client.	Consideration of how long tasks will take based on set rubric.		28 <sup>th</sup> June 2019	A
June 2019	Final introductory meeting with the client.	The success criteria are finalised with the client's approval.	These are written down to then be converted for Criterion A.	30 <sup>th</sup> June 2019	A

<b>July 2019</b>	Designing of the first-draft site and the elements.	Starting to work on the potential design of the website and all the pages.	The client disclosed that the site should have an intuitive design. Therefore, minimalism is key.	22 <sup>nd</sup> July 2019	B
<b>July 2019</b>	Identification of data structures for the database.	I had to identify what records would be needed for the database.		25 <sup>th</sup> July 2019	B
<b>July 2019</b>	Database relationships and normalisation	Since I have never done database normalisation before I had to spend some time to learn all the forms and the steps that must be taken.		30 <sup>th</sup> July 2019	B
<b>July 2019</b>	Identification and initial creation of algorithms	I designed the logical flow of the two most complex tasks.	This was done using flowcharts.	30 <sup>th</sup> July 2019	B
<b>August 2019</b>	Completion of Criterion A Document	Everything was presented in an organised manner.		5 <sup>th</sup> August 2019	A
<b>August 2019</b>	Start of development on the UI and design of site.	The design was tweaked during this process with constant communication with the client.	Use of external libraries such as Bootstrap and Sweet Alerts for front end.	11 <sup>th</sup> August 2019	C
<b>August 2019</b>	Start of development on back-end processes of the site.	Initialisation of database within MySQL and selection of development server.		15 <sup>th</sup> August 2019	C
<b>August 2019</b>	Contact of client	Client showed a positive reception to the early product.		30 <sup>th</sup> August 2019	C



<b>September 2019</b>	Contacting the client.	Implementing new suggestions to the project.		7 <sup>th</sup> September 2019	C
<b>September 2019</b>	Software given to client for Alpha testing.	Client showed a positive reception to the early alpha of the product.		18 <sup>th</sup> September 2019	C
<b>October 2019</b>	Debugging of issues and errors found through the clients testing.	A few errors were identified by the teacher.	<p>1) The client identified that the teacher could access the student profile.</p> <p>2) The client identified that the tasks would tasks could not be acted upon correctly.</p> <p>3) Password reset did not parse a unique token.</p> <p>4) Report generation does not work at all.</p>	28 <sup>th</sup> October 2019	C
<b>December 2019</b>	Began working on Criterion B and Criterion C	Documents created.		5 <sup>th</sup> December 2019	C
<b>January 2020</b>	Optimisation of website code.	Documentation formatting.		18 <sup>th</sup> January 2020	C
<b>January 2020</b>	Export of final product.	Compiling of code into coherent product.		22 <sup>nd</sup> January 2020	C

<b>January 2020</b>	Handing over of the final product to the client and showing all its functionality.	Wait for feedback regarding any last bug fixes.		23 <sup>rd</sup> January 2020	E
<b>February 2020</b>	Final squashing of bugs before release to students.	Final optimisation of the code.		4 <sup>th</sup> February 2020	E
<b>February 2020</b>	Video of product in action	Final step.	Final product sent to client with all corresponding files.	22 <sup>nd</sup> February 2019	D

## Criterion C: Development

### Complexities:

Bootstrap: .....	2
SweetAlert2: .....	3
PHPMailer: <sup>4</sup> .....	3
Password reset: .....	5
jQuery ajax() .....	6
General security: .....	7
SQL Queries: .....	8
Image reading and writing: .....	9
Form submission (_POST): .....	10
Transmission of data across the server (_GET): .....	11
Visualisation of data using Javascript and Google Chart API : .....	12
Example of SQL logic .....	14
Bibliography: .....	15
Appendix for Criterion C: .....	16

Word count (Excluding code) : 1078

Explanations of tasks can be found **within** the code after // or <- -!> tags.

## Bootstrap:<sup>1</sup>

Bootstrap is a vast library of reusable code written in HTML, CSS, and JavaScript. It's also a front-end development framework that enables to quickly build fully responsive websites.<sup>2</sup> The use of this library attains to success criteria **11**, as bootstrap is used to create an intuitive simple and modern design of the website.

In order to use bootstrap, I had to download the library and the corresponding CSS file in order to use the library to its full potential.

```
<link href="../../../vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<script src="../../../vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
```

Bootstrap has many modules and functions, therefore as requested by the client I implemented a dynamic navigation bar that can be found on all the pages on the website.

```
<nav class="navbar navbar-expand-lg navbar-dark bg-success static-top">
  <div class="container">
    <!--Button Classes-->
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <!--Logo element-->
    <div class="mx-auto" style="width: 60px;"><a href="../../../index.php" class="pull-left"></a> </div><a class="navbar-brand" href="../../../index.php">PBIS
Electrical Engineering </a>
    <!--Here this bar can also adapt to the window-size-->
    <div class="collapse navbar-collapse" id="navbarResponsive">
      <!--Elements for navigation within the requested nav-bar-->
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="../../../index.php">Home
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="registration.php">Register</a>
        </li>
        <li class="nav-item active" >
          <a class="nav-link" href="login.php">Login</a>
          <span class="sr-only"></span>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

---

<sup>1</sup> Otto, Mark, and Jacob Thornton. "Bootstrap." · The Most Popular HTML, CSS, and JS Library in the World., getbootstrap.com/

<sup>2</sup> Ouellette, Alexandre. "What Is Bootstrap: A Beginners Guide." What Is Bootstrap? An Awesome Beginners Guide, CareerFoundry, 20 Sept. 2017, careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/.

## SweetAlert2:<sup>3</sup>

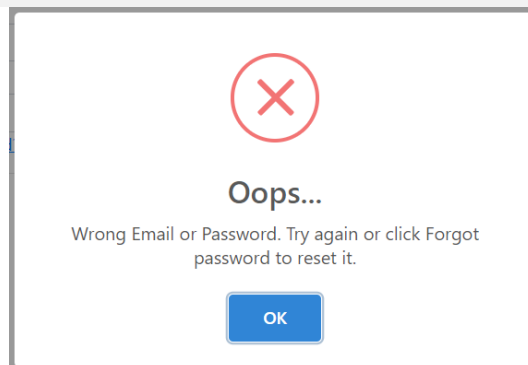
The use of this library in my project also attains to the Success Criteria **11**. Often information must be presented to the user, *SweetAlerts2* is a library that allows for elegant and clean looking popups. The library was made using CSS and JavaScript and is used by me in many areas of the website, most notably the register and login pages where errors are often exhibited<sup>4</sup>.

This time in-order to initialise the library, I opted for using the Content Delivery Network (CDN) method.

```
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@9"></script>
```

Then in order to use elements of *php* in my login page I had to convert a *php* variable into a *Javascript* one as signified by the underlined code.

```
<script type="text/javascript">
  var simple = '<?php echo $failed; ?>';
  console.log(simple);
  <!--Execution of the 'Alert'-->
  if(simple>=1){
    Swal.fire({
      icon: 'error',
      title: 'Oops...',
      text: 'Wrong Email or Password. Try again or click Forgot password to reset
it.',
    });
  }
</script>
```



## PHPMailer:<sup>4</sup>

PHPMailer was essential in the development of the Password Recovery feature, as described by criteria **2**, passwords should be recoverable using E-Mail verification. PHPMailer is most useful due to its ease of use with Gmail through a SMTP server. This recovery is based on a unique token and page being sent to the email of the account creator. The code below was adapted from the PHP Mailer GitHub<sup>5</sup> page, I changed and removed parts of the original example in order to adapt it to my needs.

---

<sup>3</sup> "SweetAlert2." SweetAlert2, [sweetalert2.github.io/](https://sweetalert2.github.io/).

<sup>4</sup> PHPMailer. "PHPMailer/PHPMailer." GitHub, 28 Dec. 2019, [github.com/PHPMailer/PHPMailer](https://github.com/PHPMailer/PHPMailer). Accessed 28 Jan. 2020.

<sup>5</sup> *ibid*

```

// (A 'Sample' Example taken from https://github.com/PHPMailer/PHPMailer)
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

require("../vendor/PHPMailer/src/Exception.php");
require("../vendor/PHPMailer/src/PHPMailer.php");
require("../vendor/PHPMailer/src/SMTP.php");
require("config.php");
//Connection to database
$conn = mysqli_connect("localhost", "root", "", "useraccounts");
$end=0;
if(isset($_POST["email"])){
//Check whether the email exists in database
$emailTo = mysqli_real_escape_string($conn,$_POST["email"]);
$QCheck= mysqli_query($conn,"select email from users where email='$emailTo'");
$existcheck= mysqli_num_rows($QCheck);
//The email will send if the email exists.
if($existcheck>0){
//Generation of random token that is used for verification.
$token = uniqid(true);
//Insertion of token and email into database.
$query = mysqli_query($conn,"INSERT INTO resetpassword(token, email)
VALUES('$token','$emailTo')");
if(!$query){
exit("Could not generate unique token.");
}

$mail = new PHPMailer(true);
try {
//Server settings
$mail->isSMTP(); Send using SMTP
$mail->Host = 'smtp.gmail.com'; Set the SMTP server to send through
$mail->SMTPAuth = true; Enable SMTP authentication
$mail->Username = 'pbis.ee.mailer@gmail.com'; SMTP username
$mail->Password = '***REDACTED***'; SMTP password
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS; // Enable TLS encryption;
$mail->Port = 587; // TCP port to connect to

//Recipients
$mail->setFrom('pbis.ee.mailer@gmail.com', 'EEHub');
$mail->addAddress($emailTo); // Add a recipient
$mail->addReplyTo('no-reply@gmail.com', 'No Reply');

// Content
// Creating password (change) site with unique token
$url = "http://" . $_SERVER["HTTP_HOST"] . dirname($_SERVER["PHP_SELF"]) .
"/resetpassword.php?token=$token";
$mail->isHTML(true); // Set email format to HTML
$mail->Subject = 'Your password reset request';
$mail->Body = "<h1>You requested a password reset</h1>
Click this <a href='$url'>link</a> to proceed.";
$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';
//Body message in Email
$mail->send();
echo 'Reset password link has been sent to your E-Mail';
$end=$end+1;

} catch (Exception $e) {
echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
}else{
exit("No account under that email exists!");
}
}
?>

```

## Password reset: <sup>6</sup>

The password reset option attains to success criteria **2**, as the client wanted students to be able to reset their password using email verification. This is a complex procedure that works alongside the token that was generated in the PHPMailer section of this document. The process is explained in the comments within the code below:

```
//If the site is not able to get the token, it will exit the procedure
if(!isset($_GET["token"])){
    exit("Cannot parse token.");
}
//If the token is attained from the site link then the program will pull data from the
databasae that match the token
$token = $_GET["token"];
$getEmailQuery = mysqli_query($conn,"SELECT email from resetpassword where token =
'$token'");
//If no rows are found then an error will show
if(mysqli_num_rows($getEmailQuery) == 0){
    exit("Can't find page");
}
//Simple post with password length check and validation.
if(isset($_POST["submit"])){
    if($_POST["password"] == $_POST["conf_password"]){
//Setting new password
        $pw = mysqli_real_escape_string($conn,$_POST["password"]);
        $pwconf = mysqli_real_escape_string($conn,$_POST["conf_password"]);
        $passwordlength=strlen($pw);
        $pw = sha1($pw);

        if($passwordlength>=15 or $passwordlength<=5){
            echo"The password must be between 6 and 15 characters.";
        }else{
            $row = mysqli_fetch_array($getEmailQuery);
            $email = $row["email"];
//Writing of new password into the database
            $query = mysqli_query($conn, "UPDATE users set password='$pw' where
email='$email'");
//After the password was successfully changed, the request record is deleted from the
'resetpassword' table.
            if($query){
                $query= mysqli_query($conn, "DELETE FROM resetpassword where
token='$token'");
                $end=$end+1;
                echo("Your password has been updated! Redirecting you to Index");
            }else{
                exit("ERROR! Something went wrong!");
            }
        }
    }else{
        echo"Passwords do not match or are the wrong length.";
    }
}
```

---

<sup>6</sup> Reece Kenney. "Sending Reset Password Emails - PHP Tutorial | Part 1 of 5." YouTube, 2020, [www.youtube.com/watch?v=BLwhJYEOVE4&list=Ply9OqUw9V30b4YiPEng75T6jd0WBA6UEv](https://www.youtube.com/watch?v=BLwhJYEOVE4&list=Ply9OqUw9V30b4YiPEng75T6jd0WBA6UEv). Accessed 12 Jan. 2020. .

## jQuery ajax()

Due to the fact that I had split my registration page into one front-end page and one back-end page, I had to convene data across the two pages, this was done using ajax which helped me send data across the server without reloading, similarly to the \$\_GET request. In this example the data is sent from *registration.php* to *process.php* then the output string is sent back under the variable 'data'.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
//Ajax functions are initialised
$(function() {
    $("#register").click(function(e) {
        var valid = this.form.checkValidity();
        if(valid){
            //PHP variables are parsed
            var firstname = $("#firstname").val();
            var lastname = $("#lastname").val();
            var email = $("#email").val();
            var password = $("#password").val();
            var permissions = $("#permissions").val();
            var passwordconf = $("#passwordconf").val();
            e.preventDefault();
            //Sending of previously parsed data across the server
            $.ajax({
                type: "POST",
                //Setting of destination
                url: "process.php",
                data: {firstname: firstname,lastname: lastname,email: email,permissions:
permissions,password: password,passwordconf: passwordconf},
                success: function(data){
                    //Sweet alert
                    Swal.fire({
                        //Sweet alert displaying data received from process.php
                        "title": data,
                        "animation":false,
                        "customClass": {
                            popup:'animated tada'
                        }
                    })
                }
            });
        }
    });
});
```



## General security:

The site will be used by many students therefore certain precautions had to be included. The first being the use of \$\_SESSION. As defined by success criteria **10**, a student account cannot be able to access the teacher's section of the website. This means that the site must be able to distinguish what type of account is logged in. For this reason, the PHP Session procedure is used, as it is a way to store information/data globally across multiple pages, and by default a session is remembered until the browser is closed <sup>7</sup>.

```
session_start();
include("teacherconfig.php");
// mysqli_connect() function opens a new connection to the MySQL server.
$conn = mysqli_connect("localhost", "root", "", "useraccounts");
// Storing Session
$user_check=null;
// Create a new session and store it for recognition
if(isset($_SESSION["t_login"])){
    $user_check = $_SESSION['t_login'];
}else{
    unset($_SESSION['t_login']);
}
// SQL Query To Fetch Complete Information Of User
$query = "SELECT email from users where email = '$user_check'";
$ses_sql = mysqli_query($conn, $query);
//SQL retrieval of email from database
$row = mysqli_fetch_assoc($ses_sql);
//Session is now recognisable by email of the user that's logged in
$login_session = $row['email'];
```

This is the initialisation of the session and happens after a user is logged in. In order to redirect users that do not match criteria or are not logged in at all to their respective home pages. This is done with the following code:

```
<?php
include("teachersession.php");
//If the session is not set and it doesn't match the t_login criteria then the user will be
redirected to index.php and if they are logged in as a student then the index page will
redirect them to their student profile.
if(!isset($_SESSION['t_login'])){
header("location: ../index.php"); // Redirecting To Home Page
}
```

---

<sup>7</sup> "PHP Sessions." W3schools.Com, 2020, [www.w3schools.com/php/php\\_sessions.asp](http://www.w3schools.com/php/php_sessions.asp). Accessed 17 Feb. 2020.

## SQL Queries:

The quintessential part of a database project for a website is the retrieval of information from the database that match specific criteria. For this project it was necessary to use many queries which often required information from different tables and thus I had to use multi-table queries and use an assortment of SQL Keywords<sup>8</sup> in order to retrieve categorical results based on selected criteria. The full code for the queries can be found in the appendix.

The query below is used on the student page in order to show tasks that the student must complete. This query incorporates data from two tables, these being *task* and *student\_task*. The binding parameter is **taskid**. And the results will be arranged in a descending order by taskid's. This is done using the '**Right Join**' parameter. [1]

```
SELECT * FROM student_task RIGHT JOIN task ON student_task.taskid = task.taskid WHERE id='$id' AND (status=(0) OR status=(1)) GROUP BY student_task.taskid DESC
```

The next query is for the teacher's task submission viewer on their home page, the query retrieves data from 3 tables using the SQL connection '**Inner Join**' (The INNER JOIN keyword selects records that have matching values in both tables.<sup>9</sup>) And the results are again, ordered by **taskid** in descending order. [2]

```
SELECT * FROM student_task INNER JOIN users ON users.id=student_task.id INNER JOIN class ON class.classid = users.classid INNER JOIN task ON task.taskid=student_task.taskid WHERE (student_task.status = 1) ORDER BY task.taskid
```

The following query is used to get values of completed tasks for the student generation protocol which attains to Criterion **8**. The **count** keyword outputs an integer which is then used to calculate the percentages seen on the report graph for the selected student. [3]

```
SELECT COUNT(task.taskid) FROM student_task INNER JOIN users ON users.id=student_task.id INNER JOIN class ON class.classid = users.classid INNER JOIN task ON task.taskid=student_task.taskid WHERE (users.id='$id') AND (student_task.status=2)
```

The following query is used in the task activation/deactivation page which attains to **criterion 9**, the distinct SQL clause is used to remove duplicates from the result set of a SELECT statement.<sup>10</sup> This is due to multiple students sharing a *taskid*, so distinct doesn't output more than 1 of the same value. [4]

```
SELECT DISTINCT taskid FROM student_task INNER JOIN users ON users.id=student_task.id INNER JOIN class ON class.classid = users.classid INNER JOIN task ON task.taskid=student_task.taskid GROUP BY student_task.taskid DESC
```

---

<sup>8</sup> "SQL Keywords Reference." W3schools.Com, 2019, [www.w3schools.com/sql/sql\\_ref\\_keywords.asp](http://www.w3schools.com/sql/sql_ref_keywords.asp). Accessed 20 May. 2019.

<sup>9</sup> "SQL INNER JOIN Keyword." W3schools.Com, 2020, [www.w3schools.com/sql/sql\\_join\\_inner.asp](http://www.w3schools.com/sql/sql_join_inner.asp). Accessed 7 Feb. 2020.

<sup>10</sup> [www.techonthenet.com](http://www.techonthenet.com). "SQL: DISTINCT Clause." Techonthenet.Com, 2020, [www.techonthenet.com/sql/distinct.php](http://www.techonthenet.com/sql/distinct.php). Accessed 8 Feb. 2020.

## Image reading and writing:

In order to show the parsing of data streams, the student is able to set a profile image that is saved on the server. The alternative to this would be the use of blob files within the database, however such a solution often slows down the database/site. Thus, only the file path and name is saved in the *users* table under the image column.

```
// POST to attain image
if(isset($_POST['submitimage'])){
    echo "Profile picture updated!";
// $_FILES is a super global variable which can be used to upload files
    $picture_name=$_FILES['image']['name'];
    $image_info = explode(".", $picture_name);
//Creation of random string
    $rand = uniqid(true);
    $fileExt = strtolower(end($image_info));
// Parsing of unique name
    $picture_new_name =$rand."_ ". $_user_check.".".$fileExt;
// Setting file path
    $filepath="./user_uploads/".$picture_new_name;
    $insert="update users set image='$picture_new_name' where email='$_user_check'";
// Insertion into database
    $stmtinsert2 = $db->prepare($insert);
    $result2 = $stmtinsert2->execute([$picture_new_name]);
// Moving of uploaded files.
    move_uploaded_file($_FILES['image']['tmp_name'], "$filepath");
}
?>
```

## Form submission (\$\_POST):<sup>11</sup>

The \$\_POST variable is also used to collect data from **forms**, the data that the \$\_POST gets, is invisible to users, unlike the \$\_GET.

In the example below, the code below is for the login page on the site.

```
<!--Initialisation of the login-form using the http post method --->
<form action="login.php" method="post">
  <!--Input windows with labels, required tag will not allow posting without selection--->
  <label for="email"><b>Account Email</b></label>
  <input class="form-control" type="email" id="email" name="email" required>
  <label for="password"><b>Password</b></label>
  <input class="form-control" type="password" id="password" name="password" required>
  <!--Button for form submission --->
  <div style="text-align:center">
    <input class="btn btn-outline-success" class="form-control" type="submit" id="LOGIN"
name="create" value="LOGIN" >
  </div>
</form>
<!--Execution of the 'Alert'--->
if(isset($_POST['email'])) {
  //Posting form submissions into data to be posted to database while cleansing sql
injection.
  $email = sanitizeEmail(mysqli_real_escape_string($conn,$_POST['email']));
  $password = sanitizeInput(sha1(mysqli_real_escape_string($conn,$_POST['password'])));
  //Checking whether the email entered and password match existing accounts inside the db.
  $sql = "SELECT * FROM users WHERE email='$email' and password='$password'";
  $result=mysqli_query($conn,$sql);
  //If statement to make decision to login or provide error message
  if (is_object($result) && $result->num_rows == 1) {
    // Register variables and redirect to file "profile.php"
    session_start();
    //starting of a session as logged in user using email.
    $_SESSION['login_user'] = $_POST['email'];
    //redirect('profile.php');
    header('location:profile.php');
  }else{
  //if authentication fails or no db element is found, then the user is prompted with an
error message.
    $login=$login+1;
```

---

<sup>11</sup> ProgrammingKnowledge. "How To Create a Register Form Using PHP, MySQL And Bootstrap." YouTube, 2020, [www.youtube.com/watch?v=TCLTL7xveGc](https://www.youtube.com/watch?v=TCLTL7xveGc). Accessed 12 July. 2019.

## Transmission of data across the server (\_GET):<sup>12</sup>

The following code is used in the class viewer site of teacher account. The results of all the tasks that have been marked as completed by the students are shown in table form using the HTML table format. Initially, I used the post method to allow the teacher to accept or reject submission requests. However, during the testing stage, I found that the inherent limitation of this method is that no matter on which task an action was done, the most recently generated row would be affected rather than the selected row. To overcome this problem, I opted to use the http GET method which eliminated this problem using data collected from the URL.

```
<!--Creation of the table --->
<tbody>
  <tr>
    <td <?= $row3['firstname']?> <td>
    <td <?= $row3['task_name']?> <td>
    <td <?= $row3['date_due']?> <td>
    <td <?= $row3['classname']?> <td>
    <?php $id=$row3['id']?>
    <?php $taskid=$row3['taskid']?>
    <td <?= $status?> <td>
<!--Buttons initiaised using bootstrap, each button has its own href with the corresponding
task id and student id so that the selection can be made and the information is set to the
url --->
    <td style="width:15%">
    <a href="viewclass.php?id_accept=<?php echo $id; ?>&taskid_accept=<?php echo
$taskid; ?>" class="btn btn-success" aria-pressed="true">Accept</a>

<div class="mx-auto" style="height: 2px;"></div>

    <a href="viewclass.php?id_reject=<?php echo $id; ?>&taskid_reject=<?php echo
$taskid; ?>" class="btn btn-warning" aria-pressed="true">Refuse</a>
    </td>

  </tr>
</tbody>

<?php
if (isset($_GET['id_accept'],$_GET['taskid_accept'])) {
    $id2 = $_GET['id_accept'];
    $complete = $_GET['taskid_accept'];
    $completeq = "UPDATE student_task SET status=2 WHERE taskid=$complete AND id=$id2";
    $stmtinsert1 = $db->prepare($completeq);
    $result1 = $stmtinsert1->execute();
    echo"Changes applied, Please refresh the page!";
}
if (isset($_GET['id_reject'],$_GET['taskid_reject'])) {
    $id2 = $_GET['id_reject'];
    $failure = $_GET['taskid_reject'];
    $discardq = "UPDATE student_task SET status=0 WHERE taskid=$failure AND id=$id2";
    $stmtinsert2 = $db->prepare($discardq);
    $result2 = $stmtinsert2->execute();
    echo"Changes applied, Please refresh the page!";
}
?>
```

---

<sup>12</sup> "PHP: \$\_GET - Manual." Php.Net, 2020, www.php.net/manual/en/reserved.variables.get.php. Accessed 1 Feb. 2020.

## Visualisation of data using Javascript and Google Chart API :<sup>13</sup>

The Google Chart API is an interactive Web service that creates graphical charts from user-supplied data<sup>14</sup>. I integrated the Javascript library with elements from the database using php. This attains to the criteria 8, as it allows the client to produce reports visually, and see overall performance of the global classes. This code was taken from the Google sites but adapted and changed to my needs.

```
<form method="POST">
<!--Select of student using html $_POST method --->
  <SELECT class= "form-control" name="classid">
    <?php
//Query to display all students from database.
$result1 = mysqli_query($conn,"SELECT * FROM users INNER JOIN class on
class.classid=users.classid where (class.classid!=0)");
while($rowf = mysqli_fetch_array($result1))
{
    $s_name=$rowf["firstname"];
    $id=$rowf["id"];
    $s_lname=$rowf["lastname"];
    $c_name=$rowf["classname"];
//Drop down selector of accounts.
    echo '<OPTION VALUE="' . $id . '">'. $id . " " . $s_name . "
".$s_lname."||".$c_name.'</OPTION>';
}
?>
</select>
<hr class="mb-3">
<div style="text-align:center">
    <input class="btn btn-outline-info" class="form-control" type="submit" id="submit"
name="submit" value="Create Report" >
</div>
</form>
//Integer retrieval to be used in data projection.
$student_complete = mysqli_num_rows(mysqli_query($conn, QUERY))
$student_failed = mysqli_num_rows(mysqli_query($conn, QUERY))
$student_inprogress = mysqli_num_rows(mysqli_query($conn, QUERY))
$student_awaiting = mysqli_num_rows(mysqli_query($conn, QUERY))
$resultname mysqli_query($conn,"SELECT firstname,lastname from users where id='$id'");
<div>
    <head>
//Loading of API functions.
    <script type="text/javascript">
        google.charts.load("current", {packages:["corechart"]});
        google.charts.setOnLoadCallback(drawChart);
//Chart function initialisation.
        function drawChart() {
//Conversion of array data into chart elements.
            var data = google.visualization.arrayToDataTable([
                ['Task', 'How much done'],
//Presentation of php variables(data) in JS scripture.

                ['Completed (<?php echo$student_complete?>)',    <?php
echo$student_complete?>],
                ['Failed (<?php echo$student_failed?>)',    <?php
echo$student_failed?>],
                ['Undetermined (<?php echo$student_inprogress?>)',    <?php
echo$student_inprogress?>],
                ['Awaiting (<?php echo$student_awaiting?>)',    <?php
echo$student_awaiting?>]
```

<sup>13</sup> "Code Examples | Charts | Google Developers." Google Developers, 2019, developers.google.com/chart/interactive/docs/examples. Accessed 18 Jan. 2020.

<sup>14</sup> "Frequently Asked Questions | Charts | Google Developers." Google Developers, 2020, developers.google.com/chart/interactive/faq. Accessed 17 Feb. 2020.

```
]);
//Pie chart settings.
var options = {
    title: 'Report for: <?php echo$sendname." ".$sendsurname, "
(".$id.)"?>',
    pieHole: 0.4,
};
//Drawing of chart element.
var chart = new
google.visualization.PieChart(document.getElementById('donutchart'));
chart.draw(data, options);
}
</script>
</head>
<body>
//Sizing and centering.
<div id="donutchart" style="width: 900px; height: 500px;"></div>
</body>
</div>
```

## Example of SQL logic

In order to show a complex procedure that has been planned for in criterion B, this being the task creation procedure. This code below shows the algorithmic complexity in relation to the tasks. For this function to happen three queries must be undertaken that retrieve data from multiple tables, this task had to be logically complex. This task corresponds to success criteria 4.

```
//Selection of class that the task should be set to.
<label for="classid"><b>Select Class</b></label>
<SELECT class="form-control" name="classid">
  <?php
//Retrieval of all classes found in the database
$result1 = mysqli_query($conn,"SELECT * FROM class");
while($row = mysqli_fetch_array($result1))
{
    $class_id=$row["classname"];
    $classid=$row["classid"];
    echo '<OPTION VALUE="' . $classid . '">'.$class_id.'</OPTION>';
}
?>
</select>

<?php
$end=0;
//Submission of elements from the form.
if(isset($_POST['submit'])){
    $task_name=$_POST['task_name'];
    $date_set=$_POST['date_set'];
    $date_due=$_POST['date_due'];
    $task_desc=$_POST['task_desc'];
    $classid=$_POST['classid'];
//Insertion of the data into the task table in the database.
$stmtinsert1 = $db->prepare("INSERT INTO task (date_set, date_due,task_desc,
task_name)values (?, ?, ?, ?)");
$result = $stmtinsert1->execute([$date_set, $date_due, $task_desc, $task_name]);
//Retrieval of the most recent taskid from the task table.
$t_id = mysqli_query($conn,"SELECT * FROM task ORDER BY taskid DESC LIMIT 1");
while($row = mysqli_fetch_array($t_id))
{
    $taskid=$row["taskid"];
}
//Then all the students within the class that the task has been set to are retrieved.
$inclass = mysqli_query($conn,"SELECT id FROM users WHERE classid='$classid'");
while($row = mysqli_fetch_array($inclass))
{
    $recipient=$row['id'];
//Posting into database the task id and the corresponding student id's into student_task
table.
    $stmtinsert1 = $db->prepare("INSERT INTO student_task (id,taskid)values (?,?)");
    $result = $stmtinsert1->execute([$recipient, $taskid]);
}
echo "TASK SET";
$end=$end+1;
}
?>
```



## Bibliography:

- 1) Otto, Mark, and Jacob Thornton. "Bootstrap." · The Most Popular HTML, CSS, and JS Library in the World., [getbootstrap.com/](https://getbootstrap.com/)
- 2) Ouellette, Alexandre. "What Is Bootstrap: A Beginners Guide." What Is Bootstrap? An Awesome Beginners Guide, CareerFoundry, 20 Sept. 2017, [careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/](https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/).
- 3) "SweetAlert2." SweetAlert2, [sweetalert2.github.io/](https://sweetalert2.github.io/).
- 4) PHPMailer. "PHPMailer/PHPMailer." GitHub, 28 Dec. 2019, [github.com/PHPMailer/PHPMailer](https://github.com/PHPMailer/PHPMailer). Accessed 28 Jan. 2020.
- 5) Reece Kenney. "Sending Reset Password Emails - PHP Tutorial | Part 1 of 5." YouTube, 2020, [www.youtube.com/watch?v=BLwhJYEOVE4&list=PLy9OqUw9V30b4YiPEng75T6jd0WBA6UEv](https://www.youtube.com/watch?v=BLwhJYEOVE4&list=PLy9OqUw9V30b4YiPEng75T6jd0WBA6UEv). Accessed 12 Jan. 2020. .
- 6) "PHP Sessions." W3schools.Com, 2020, [www.w3schools.com/php/php\\_sessions.asp](https://www.w3schools.com/php/php_sessions.asp). Accessed 17 Feb. 2020.
- 7) "SQL Keywords Reference." W3schools.Com, 2019, [www.w3schools.com/sql/sql\\_ref\\_keywords.asp](https://www.w3schools.com/sql/sql_ref_keywords.asp). Accessed 20 May. 2019.
- 8) "SQL INNER JOIN Keyword." W3schools.Com, 2020, [www.w3schools.com/sql/sql\\_join\\_inner.asp](https://www.w3schools.com/sql/sql_join_inner.asp). Accessed 7 Feb. 2020.
- 9) [www.techonthenet.com](http://www.techonthenet.com). "SQL: DISTINCT Clause." Techonthenet.Com, 2020, [www.techonthenet.com/sql/distinct.php](http://www.techonthenet.com/sql/distinct.php). Accessed 8 Feb. 2020.
- 10) ProgrammingKnowledge. "How To Create a Register Form Using PHP, MySQL And Bootstrap." YouTube, 2020, [www.youtube.com/watch?v=TCLTL7xveGc](https://www.youtube.com/watch?v=TCLTL7xveGc). Accessed 12 July. 2019.
- 11) "PHP: \$\_GET - Manual." Php.Net, 2020, [www.php.net/manual/en/reserved.variables.get.php](https://www.php.net/manual/en/reserved.variables.get.php). Accessed 1 Feb. 2020.
- 12) "Code Examples | Charts | Google Developers." Google Developers, 2019, [developers.google.com/chart/interactive/docs/examples](https://developers.google.com/chart/interactive/docs/examples). Accessed 18 Jan. 2020.
- 13) "Frequently Asked Questions | Charts | Google Developers." Google Developers, 2020, [developers.google.com/chart/interactive/faq](https://developers.google.com/chart/interactive/faq). Accessed 17 Feb. 2020.

## Used on many occasions:

"Stack Overflow - Where Developers Learn, Share, & Build Careers." Stack Overflow, 2020, [stackoverflow.com/](https://stackoverflow.com/). Accessed 18 Feb. 2020.

## Appendix for Criterion C:

[1]

```
<?php
//Retrieval of the users id
$id_query = mysqli_query($conn,"SELECT id FROM users WHERE email='$user_check'");
while($row = mysqli_fetch_array($id_query)):
{
    $id=$row['id'];
}endwhile;

//Retrieval of the tasks set based on the id attained above.
$tasks = "SELECT * FROM student_task RIGHT JOIN task on student_task.taskid = task.taskid
WHERE id='$id' AND (status=(0) OR status=(1)) group by student_task.taskid DESC";
//The code below checks the state of the task and displays it.
$result = $conn->query($tasks);
$num = mysqli_num_rows($result);
if (0<$num){
while($row3 = $result->fetch_assoc()){
    if($row3['status'] == 2){
        $status="Completed";
    }elseif($row3['status'] == 1){
        $status="Confirmation pending";
    }else{
        $status="Not Completed";
    }
}
?>
```

[2]

```
<?php
//Query
$tasks = "SELECT * FROM student_task INNER JOIN users on users.id=student_task.id INNER
JOIN class on class.classid = users.classid INNER JOIN task on
task.taskid=student_task.taskid where (student_task.status = 1) order by task.taskid ";
$result = $conn->query($tasks);
while($row3 = $result->fetch_assoc()){
    if($row3['status'] == 2){
        $status="Completed";
    }elseif($row3['status'] == 1){
        $status="Confirmation pending";
    }else{
        $status="Not Completed";
    }
} ?>
//Output of query
<tbody>
<tr>
<td> <?= $row3['firstname']?> <td>
<td> <?= $row3['task_name']?> <td>
<td> <?= $row3['date_due']?> <td>
<td> <?= $row3['classname']?> <td>
<?php $id=$row3['id']?>
<?php $taskid=$row3['taskid']?>
<td> <?= $status?> <td>

//Buttons for selection of user response

<td style="width:15%"> <a href="viewclass.php?id_accept=<?php echo $id;
?>&taskid_accept=<?php echo $taskid; ?>" class="btn btn-success" aria-
pressed="true">Accept</a><div class="mx-auto" style="height: 2px;"></div>
<a href="viewclass.php?id_reject=<?php echo $id; ?>&taskid_reject=<?php echo
```

```

$taskid; ?>" class="btn btn-warning" aria-pressed="true">Refuse</a> </td>

    </tr>
</tbody>
<?php } ?>

```

[3]

Code relates to the Google Charts API section.

[4]

```

<form method="post">
    <SELECT class="form-control" name="taskid" required>
        <?php
            $taskid=null;
//Query
            $result2 = mysqli_query($conn,"SELECT DISTINCT * FROM student_task INNER JOIN users
on users.id=student_task.id INNER JOIN class on class.classid = users.classid INNER JOIN
task on task.taskid=student_task.taskid group by student_task.taskid DESC");
            while($row1 = mysqli_fetch_array($result2))
            {
                $taskid=$row1["taskid"];
                $task_name=$row1["task_name"];
                $date_due=$row1["date_due"];
                $class=$row1["classname"];
//Output of retrieved potential options.
                echo '<OPTION VALUE="' . $taskid . '">'.$task_name.'" || Date Due: ".$date_due.'"
|| ".$class.'" || Id: (".$taskid.")'.</OPTION>';
            }
        ?>
    </select>
</form>

```

# CRIT D VIDEO

<https://youtu.be/FAZbCSngokQ>

## Criterion E: Evaluation

Word count : 483

### Evaluation of the product

- 1 Students will be able to register and login their own unique accounts.  
✓ Met, students can create unique accounts with hashed passwords.
- 2 A password recovery procedure for students  
✓ Met, unique password recovery link is sent to requestants email.
- 3 An ability for students to change classes and upload a profile image.  
✓ Met, students can edit said fields.
- 4 Allow the client account to set tasks to students within certain classes.  
✓ Met, the client can set tasks to any existing class.
- 5 Ability for students to see all the tasks assigned  
✓ Met, all active tasks are displayed on the student's main page.
- 6 A task approval page for the client.  
✓ Met, the client can accept or reject submission requests.
- 7 Allow the client to access records with all past notices of students.  
✓ Met, all past submissions and records are accessible.
- 8 Allow the client to create reports.  
✓ Met, custom reports can be created with data from a selected student.
- 9 Allow the client to activate and deactivate tasks.  
✓ Met, Tasks can be deactivated/reactivated by the client.
- 10 The teacher portal cannot not be accessible by a student account  
✓ Met, when a student attempts to access the teacher portal, they are redirected away.
- 11 The website design should be simple and elegant.  
✓ Met, design is simple and intuitive.

The official response from the client can be found in the appendix.

The product works well however it could be prone to hacking as protection is limited.

## Recommendations for Further Development

### - Minor Improvements

- Include an email verification when an account is created in order to counteract excessive account creation with an intent to overload the database.
- Include a change of password within the settings of the account for users that are already logged in, as changing password through email confirmation can be time consuming.

### - Major Improvements

- Include a possibility for file submissions. This would be done using FileZilla (ftp server). This means that the teacher would be able to request files from students. This however would be quite time consuming an additional level of complexity.
- Automation of certain processes. Currently for a task to be marked under failing condition, it must be done by the teacher themselves. An improvement would be the system to mark certain submissions as failed if certain criteria are not met (if overdue).

## Extensibility

The client showed a lot of enthusiasm for the email verification feature as manual alteration of passwords for students is very time consuming and inefficient. I found the development and planning of this application as very interesting and it facilitated to me the necessity of extensive planning as due to my comprehensive plan the design process was swift.

Overall the success criteria set out at the beginning of the project were all successfully attained. The client was very pleased by the final solution and voiced that the product will be used in the next academic term. Evidence of success can be reflected by the test plan results that can be found in appendix.

## Appendix: Evidence of correspondence

Initial interaction and discussion with the client:

**Me:**

Hello Mr Mcaleavy,

Firstly I would like to thank you for taking the time to meet me regarding this potential product I would like to develop for you.

I have noticed that you are not able to come to the club as often as before due to the constant meeting you have to be part of and therefore cannot be present during the club. Thus, I thought in order to help you with your busy schedule and running of the club that I would develop a website/portal for you to be able to set tasks for the students. If you could please specify some requirements you would expect for such a site and what features and permissions the students should be able to have.

Thank you in advance and kind regards,

[REDACTED]

**Client:**

Hi [REDACTED].

Thank you for taking the time to do such a project and I expect it will come very helpful to my very busy schedule as I want to still be connected with the club even if I am not able to attend.

The requirements and features I would like to see from the solution are:

1. Students will be able to register and login their own unique accounts. The accounts should be secured.
2. An intuitive password recovery procedure for students (One that does not need input from my end).
3. An ability for students to change classes and upload a profile image manually.
4. Allow my account to set tasks to students within certain classes.
5. Ability for students to see all the tasks assigned to them and the ability to mark a task as completed.
6. A page for me to be able to approve submissions by students.
7. Allow me to see all records of all students and the status of their past activity.
8. Allow me to create reports based on specific students' performances.
9. Allow me to activate and deactivate tasks.
10. The teacher portal cannot not be accessible by a student account
11. The website design should be simple and elegant.

Thank you and please email me if you have any further questions,  
Best regards,  
Stephen

#### Email (After first design sketches were shown)

Good morning,

Thank you for sending these mock designs to me, they are very useful in the visualisation of the potential site. I really like the idea of the navigation bar as it often helps in fast and intuitive navigation across pages. I have compiled a few potential things I would like to see on the site, I have set them out in sections.

#### **Main Page**

I would like to potentially add a slideshow of images that can be uploaded for showcase of some of the students projects.

#### **Register Page:**

For this page I do not really like that students can potentially create a teacher account, therefore as I would like for the admin to create the teacher accounts through the back end. I also have students that often change what club they go to; therefore, I would like the ability to change the club later.

#### **Login Page:**

This page seems perfect. Except I would like potentially to have an ability for students to change password using an email system (Potentially?). As students are prone to losing or forgetting login details!

#### **Student Page:**

I like the idea of tasks showing up for the students. I would like the following criteria to show: *Task Name, Task Description, Date Set, Date Due*. I would like an option of them to mark something as 'completed' which then on my personal page I can either accept or reject.

#### **Task viewer page:**

I think this page is redundant, it should be enough for the tasks to show on the student page.

#### **Task setting page:**

I like the overall idea of this, I would like to see more options.

#### **Other pages I would like to see:**

*Task deactivation:* I would like the ability to quickly deactivate or reactivate tasks.

*Report generation:* Parents often email me how well their students are doing, with completing tasks and homework. I often spend a lot of time looking at my personal excel spreadsheet, I would really like a report generating system.

*Changing of classes for students:* As i mentioned in the register page comment, I would like the ability for the students to change their class themselves.

Overall, I find your solution very appealing!  
Please get back to me with an overview and a timeline!



Kind regards,  
Stephen

#### Final correspondence with the client

**Me:**

Good morning,

I am happy to announce that the final solution for the project is complete. I have spent quite a long-time doing bug fixes and I believe that the “EE Hub” site is ready for use. Students should however note any problems or bugs that they potentially may find as they can be the solutions ‘beta testers’. I am very happy that you have allowed me to have this opportunity to work on such an interesting project, as I hope the solution can be used for the near future.

I have attached the project in the download link below.

Thank you and with kind regards,

[REDACTED]

**Client:**

Good morning [REDACTED]

I would like to thank you for the product, and I am looking to start using it in the next academic term.

First I would like to tell you about the features and parts that I really enjoy and think are great, then I will discuss potential improvements for the future which would make this product even better and maybe potentially useful for other parts of the school and not only the EE club.

The features I really enjoy are the intuitive and simple design of the entire website as it is simple and easy to navigate through all the pages (Thanks to the nav-bar). I also enjoyed the ability to create reports with pie charts as parents often ask me how well their students are doing, and this is a quick and easy way for me to show them. The password recovery is also very interesting and intuitive as it is done automatically without my input which is very respectful to my lack of time for administrative tasks.

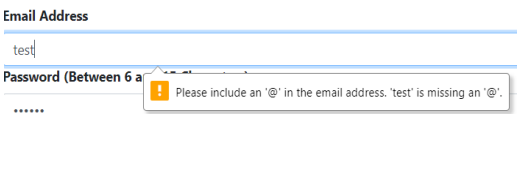

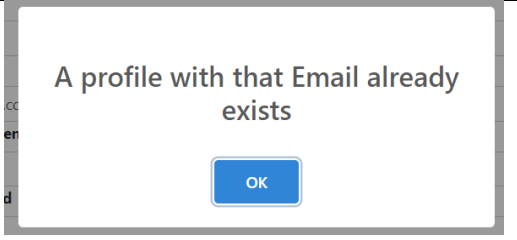
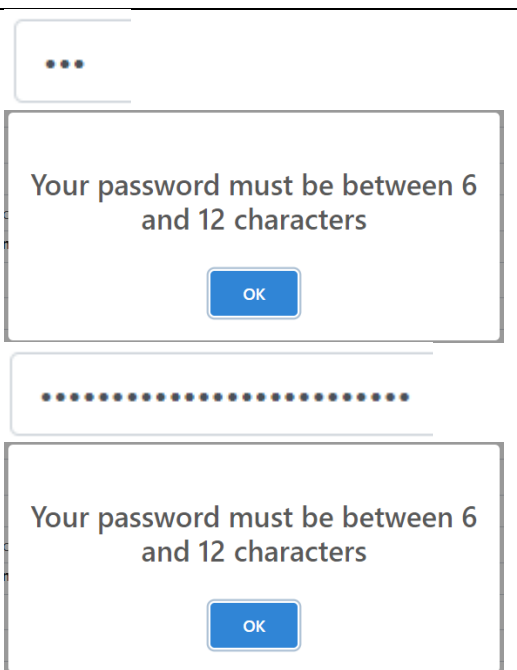
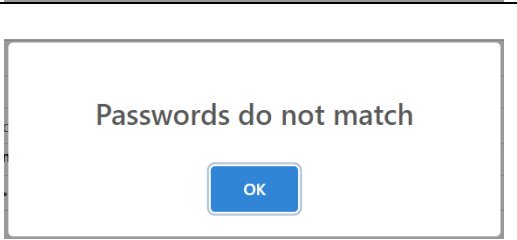
The parts I would like to see developed are:

- 1) Include a change of password within the settings of the account for users that are already logged in.
- 2) Include a possibility for file submissions, in response to tasks that have been set.
- 3) Automatically set tasks overdue as failed.

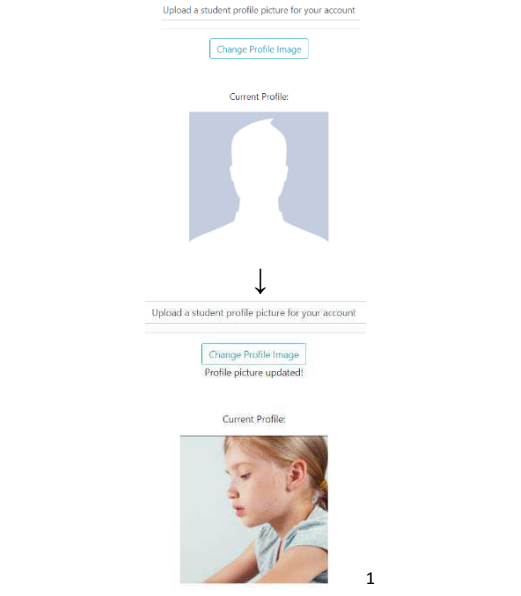
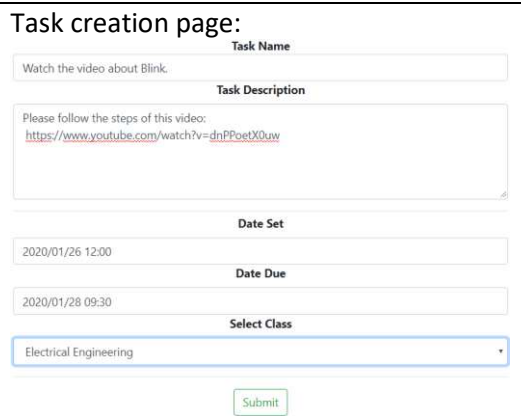
Overall, I really appreciate all the work that you have put into this project and I think that it is a sound product and I will be using it for the years to come.

Thank you and best regards,  
Stephen

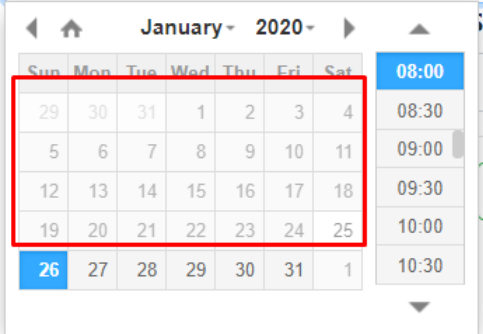
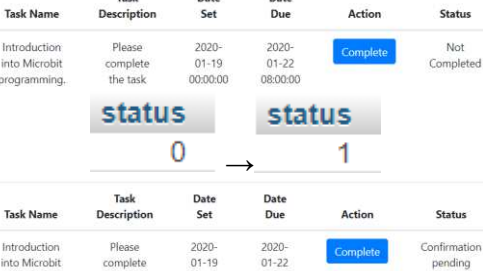

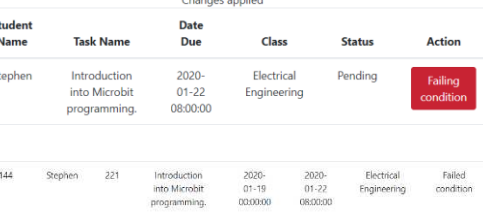
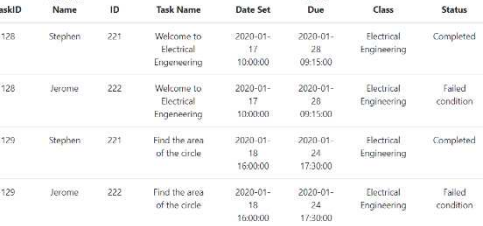
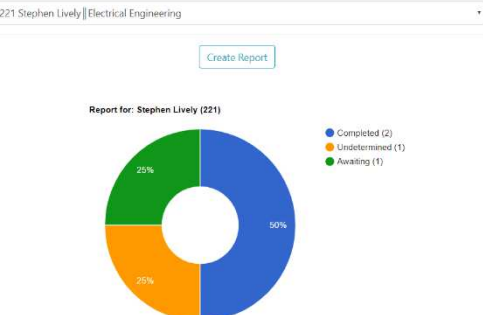
## Completed Tasks

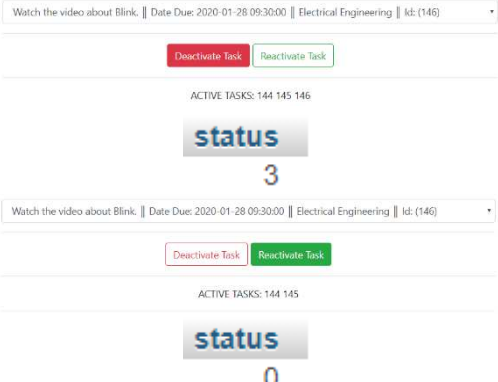
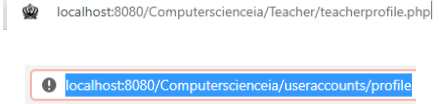
Test Number	Test description	Test data	Expected Outcome	Actual Outcome	Pass/Fail
1	Requirement 1: Email field in the registration page must contain an '@' sign.	Press submit button with an email that doesn't contain an '@' sign.	Submission failure, an error shows up displaying a need for at '@' sign in field.		Pass
2	Requirement 1: All fields within the register page must be filled.	Press submit button without all the fields being filled out.	Submission failure, an error shows up displaying a need for all fields to be filled.		Pass
3	Requirement 1: There can only be one account per email address.	Press submit button with an email that already has an account connected to it.	Submission failure, an error shows up displaying that an account already exists for that email address.		Pass
4	Requirement 1: Password must be between 6 and 15 characters long.	Type a password that is less than 6 then more than 15 characters long.	Submission failure, an error shows up displaying that the password requirements are not met.		Pass
5	Requirement 1: Password and confirm password must match.	Type a different password into the <i>password</i> and <i>confirm password</i> fields.	Submission failure, an error shows up displaying that the passwords do not match.		Pass

5	Requirement 1: All passwords are hashed.	When an account is created the password must be encrypted.	An encrypted <i>password</i> string is stored in the database	<pre>password f98ac97fe4ad0e246a26be01f55c53b9a525fdd9</pre>	Pass
6	Requirement 2: Password recovery system will only work for accounts that exist	A forgotten password link will only be sent to emails that exist in the database.	An email will not be sent to an email that does not have a corresponding account on the site.	No account under that email exists!	Pass
7	Requirement 2: Password recovery link is sent to destined email.	A password recovery link will be sent to an email that is found in the database.	A forgotten password email is sent to the email filed in a request.	<p>Your password reset request</p> <p>EEHub &lt;pbis.ee.mailer@gmail.com&gt; to me ▾</p> <p><b>You requested a password reset</b></p> <p>Click this <a href="#">link</a> to proceed.</p> <p>Reply Forward</p>	Pass
8	Requirement 2: Password recovery system will contain a unique authentication token for each request.	When a password change is requested, each request will have a unique token.	No two requests will have the same authentication token.	<pre>token          email 15e2cc6aa91225 stroxvfx@gmail.com 15e2cc6ca624d7 bruh123@gmail.com</pre>	Pass
9	Requirement 2: The password can successfully be changed.	The new password is processed and hashed.	The password will successfully change. (As shown by two differing hashes)	<p><b>Hash before:</b> 2509c4d5594bf7295cebd097e0ec4b5653abded6</p> <p><b>Hash after:</b> a469f2ef9f0e6ce0008666414e3c02705080c88c</p>	Pass
10	Requirement 3: The class can successfully be changed.	The student can change their class within their settings page.	The student's class will change after the designated class is selected from a dropdown menu.	<p>Current class: No class selected</p> <p>Select your class Select your class Electrical Engineering Robotics club</p> <p>classid 0</p> <p>After submission the <i>current class</i> has changed:</p> <p>Current class: <b>Electrical Engineering</b></p> <p>classid 1</p>	Pass

11	Requirement 3: The profile picture can be changed.	The student can change their profile picture.	The default profile picture is changed to the new custom one that is uploaded by the user.		Pass																	
12	Requirement 4: Allow the client account to set tasks to students within certain classes.	The <i>teacher</i> account is used to create a new task.	The task created should show up in the <i>task</i> table, then all of the students that are assigned to the selected class should receive the tasks within the <i>student_task</i> table.	<p>Task creation page:</p>  <p><i>task</i> table:</p> <table border="1" data-bbox="957 1276 1479 1422"> <thead> <tr> <th>taskid</th> <th>date_set</th> <th>date_due</th> <th>task_desc</th> </tr> </thead> <tbody> <tr> <td>146</td> <td>2020-01-26 12:00:00</td> <td>2020-01-28 09:30:00</td> <td>Please follow the steps of this video: <a href="https://www.youtube.com/watch?v=dnPpoetX0uw">https://w...</a></td> </tr> </tbody> </table> <p><i>student_task</i> table:</p> <table border="1" data-bbox="957 1489 1479 1612"> <tbody> <tr> <td>221</td> <td>146</td> <td>0</td> </tr> <tr> <td>222</td> <td>146</td> <td>0</td> </tr> <tr> <td>239</td> <td>146</td> <td>0</td> </tr> </tbody> </table>	taskid	date_set	date_due	task_desc	146	2020-01-26 12:00:00	2020-01-28 09:30:00	Please follow the steps of this video: <a href="https://www.youtube.com/watch?v=dnPpoetX0uw">https://w...</a>	221	146	0	222	146	0	239	146	0	Pass
taskid	date_set	date_due	task_desc																			
146	2020-01-26 12:00:00	2020-01-28 09:30:00	Please follow the steps of this video: <a href="https://www.youtube.com/watch?v=dnPpoetX0uw">https://w...</a>																			
221	146	0																				
222	146	0																				
239	146	0																				

<sup>1</sup> "Close up Portrait Profile Girl Stock Footage Video (100% Royalty-Free) 22535554." Shutterstock, [www.shutterstock.com/video/clip-22535554-close-portrait-profile-girl-child-7-8-years](http://www.shutterstock.com/video/clip-22535554-close-portrait-profile-girl-child-7-8-years).

13	Requirement 4: Make sure that the due date cannot be earlier than the current time.	Using a drop down, an attempt to select an earlier date is attempted.	All previous times should become greyed (unclickable) out.		Pass
14	Requirement 5: Ability for students to see all the tasks assigned to them. (Option to mark a task as completed)	One task is set for the class that the test student is in.	The task should show up on the student index page. When the completed button is clicked the status of the task should change from 0 to 1.		Pass
15	Requirement 6: A task approval page, allowing the client to approve or reject submission requests.	The approval request from the student should appear on the conformation page.	The teacher should be able to either <i>accept</i> or <i>reject</i> the submission. If it is accepted the <i>status</i> should change to <b>2</b> , if it is rejected then it should change the <i>status</i> to <b>0</b> .		Pass
16	Requirement 6: The client should be able to apply a failing condition when a task is not submitted on time.	The teacher applies a failing condition to a task.	The <i>status</i> of the task should turn from <b>Pending</b> into <b>Failed condition</b> .		Pass
17	Requirement 7: Allow the client to access records with all past notices of students.	The client should see all past records when logged into the <i>teacher</i> account.	The teacher should be able to see all the past tasks with the status shown clearly.		Pass
18	Requirement 8: Allow the client to create reports based on specific students' performances.	The client generates a student report while logged into the <i>teacher</i> account.	The client should generate reports for a selected student.		Pass

19	Requirement 9: Allow the client to activate and deactivate tasks.	The <i>teacher</i> account will deactivate then reactivate a select task.	The <i>status</i> of the task when deactivated should change to <b>3</b> , then when reactivated the <i>status</i> should change to <b>0</b> .	 <p>The screenshot shows a task management interface. At the top, there is a header with the text "Watch the video about Blink.   Date Due: 2020-01-28 09:30:00   Electrical Engineering   Id: (146)". Below this, there are two buttons: "Deactivate Task" (red) and "Reactivate Task" (green). Underneath the buttons, it says "ACTIVE TASKS: 144 145 146". In the center, there is a large grey box with the word "status" in blue, and the number "3" below it. At the bottom, there is another header with the same text as the top one. Below that, there are two buttons: "Deactivate Task" (red) and "Reactivate Task" (green). Underneath, it says "ACTIVE TASKS: 144 145". In the center, there is a large grey box with the word "status" in blue, and the number "0" below it.</p>	Pass
20	Requirement 10: The teacher portal cannot not be accessible by a student account	The teacher page URL is typed into the address bar.	The student should be redirected to their profile page if such request is made using a student.	 <p>The screenshot shows a browser address bar. The top part of the address bar shows the URL "localhost:8080/Computerscienceia/Teacher/teacherprofile.php". Below this, there is a red box with a warning icon and the URL "localhost:8080/Computerscienceia/useraccounts/profile".</p>	Pass