



**NCI** Placement 360



National  
College *of*  
Ireland

# NCI Placement 360

---

TECHNICAL REPORT

Robert Dillon – x12342551  
ROBERT.DILLON@STUDENT.NCIRL.IE

## Declaration Cover Sheet for Project Submission

### SECTION 1 *Student to complete*

|                                          |
|------------------------------------------|
| <b>Name:</b><br><b>Robert Dillon</b>     |
| <b>Student ID:</b><br><b>x12342551</b>   |
| <b>Supervisor:</b><br><b>Lisa Murphy</b> |

### SECTION 2 **Confirmation of Authorship**

*The acceptance of your work is subject to your signature on the following declaration:*

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

**Complete the sections above and attach it to the front of one of the copies of your assignment,**

#### **What constitutes plagiarism or cheating?**

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another

and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

### **Penalties for Plagiarism**

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend:

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work

**Executive Summary .....5**

**1. Introduction .....6**

**1.1 Background .....6**

**1.2 Aims.....7**

**1.3 Technologies .....8**

**1.4 Structure .....9**

**2. System .....10**

**2.1 Requirements.....10**

        2.1.1 Functional requirements .....10

        2.1.2 Data requirements.....18

        2.1.3 User requirements.....19

        2.1.4 Environmental requirements.....20

        2.1.5 Usability requirements .....20

**2.2 Design and Architecture.....20**

**2.3 Graphical User Interface (GUI) Layout .....21**

**3 Testing.....25**

**3.1 Unit Testing.....25**

**3.2 UI Testing .....25**

**3.3 Test Driven Development.....25**

**3.4 Iterative Testing.....26**

**3.5 Regression Testing.....26**

**3.6 User Acceptance Testing .....26**

**3.7 Trunk Testing.....26**

**4 Implementation.....27**

**4.1 Overview.....27**

**4.2 Role-based Access Control .....28**

**4.3 Gamification.....30**

**4.4 Security .....30**

        4.4.1 Passwords.....30

        4.4.2 User Account Verification .....31

        4.4.3 Incorrect Password Restriction .....31

**5. Appendix.....32**

**5.1 Project Plan .....32**

**5.2 Project Proposal .....32**

**5.3 Project Plan .....37**

**5.4 Monthly Journals.....38**

## EXECUTIVE SUMMARY

NCI Placement 360 is a responsive web application designed for National College of Ireland (NCI) students, faculty of NCI and employers who work within NCI to evaluate students during internships. It allows students and managers to enter their monthly reports, enter in skills used throughout their internship, complete final year evaluations and make contact with their contact point from NCI.

The application allows students and employers to enter data into the application, which is then shown in a visual chart dashboard. NCI can then use this data to update curriculums and adequately prepare students upon graduation. This data can be mined directly from the application and visualised using various chart frameworks dynamically.

In order to encourage users to enter skills into the web app, gamification is integrated directly into Placement 360 for students. Elements of gamification include a progress circle showing how many entries they've made towards 100 total entries, badges on their dashboard and a "trophy case" displaying the user's currently collected badges.

Recently, mobile internet traffic overtook desktop traffic. This is a big milestone and when you take just younger demographics into account, this percentage would be even higher.

With this trend in mind, NCI's student offerings were looked at. The online portal, MyNCIStudent, isn't mobile responsive. Unless it's viewed on Internet Explorer, users can't scroll down the page. This is ineffective given that the majority of users of this portal don't use Internet Explorer and a large number of students want to access this system on their smartphones.

If students and employers use Placement 360 to fill in their monthly reports instead of the paper method, it will save time for both parties but above all provide a visual representation in a 360 degree angle. The graphs section allows the School of Computing to stay as up to date as possible in its course content.

When looking at implementation, a responsive web app seemed like the most effective option. After researching Placement 360 as a mobile app, it seemed ineffective due to users having multiple different mobile operating systems, which would require multiple native apps for full coverage. Placement 360 also needs to be available on desktop so a responsive web app was the best option.

## 1. INTRODUCTION

### 1.1 Background

In many courses within the School of Computing in the National College of Ireland, a Work Placement module makes up the student's second semester in 3rd year. This is a six-month placement. Many students begin looking for placements between the end of 2nd year and the start of 3rd year. Most placements begin in January.

Internships are six months long, so if one starts in January, it will finish in July. During this time, there is supposed to be regular contact between the student and the college as well as the employer and the college. This includes an on-site visit from the college supervisor as well as a final report done by the student and a survey filled out by the employer for the on-site visit.

At the minute, there's no all-round system between the student, college and employer that handles all work placement requirements and my goal is to solve this. The application will be used from when students start looking for placements up until they complete their internship in the summer following 3rd year. The application will take this all into consideration.

The current method of completing monthly reports is inconvenient and time consuming for students. For monthly reports, students need to download the template every month and enter in my section in Microsoft Word. Then the student would email it to their manager or workplace supervisor and they would fill out their section. They would print it, sign it and return it to the student to complete. The student would sign and scan the document before uploading it to Moodle. This is a lengthy process and a system like NCI Placement 360 would make this system much more efficient and useful to the student and NCI.

Another area where students find the process to be lengthy is the organisation and execution of on-site visits by the academic supervisor. This is generally done over email and requires more Word documents to be completed. These documents are lengthy and the contents of the document are never used effectively even though the information filled out by the students and employers are valuable. With NCI Placement 360 this data could be used effectively in many ways to increase the quality of teaching in the School of Computing.

These are observations from a student point-of-view. When speaking with my supervisor, she made me realize there were many benefits from an NCI's perspective.

Currently there isn't a system for aggregating statistics from internships. This starts from early in the process such as job requirements. It would benefit NCI to have a NCI Placement 360 system that takes information from employers and aggregates them. The system could then generate dynamic graphs to display this information.

The most useful element of the application will be the feedback that it provides to teaching staff. For example, a lecturer will be provided with information about java and given clear examples of how it is used in the company. In return, providing the lecturer with up to date examples to use in the classroom. The system could also aggregate information from a student's final report and graphs could then be generated using data from all students.

This in return could provide information to the college where students are slipping during their internships and what areas require more attention. One complaint heard from students in NCI was that there wasn't enough contact between the college and students during their internship. This system could provide a way to keep both the college and student in contact— specifically their supervisor. Work Placement makes up quite a big part of a student's time in NCI and I think it's time there was a dedicated, end-to-end system for all three parties involved: students, faculty and employers.

## **1.2 Aims**

**The aims for this project are broken down into three areas:**

For **students**, NCI Placement 360 should be always available, secure and a responsive system. The purpose it serves for students is one that allows an upload facility for their monthly reports. These reports need to be entered on a monthly basis and the student should detail what they've done over the previous month.

Students should be able to log their skills within the system. A student can select a category, a title and a duration for their skill. An example would be a student on placement at a .NET company logging four hours spent writing C# code. This data is then viewable to the individual students and also to faculty.

Students should also be able to upload their final report – a document completed when the student is nearing the end of their placement – through the system to

be viewed by their academic supervisor. A student should also be able to contact NCI through the system by choosing their contact.

For **faculty**, Placement 360 should be a system that allows a portal to view the various students currently on work placement. Faculty should be able to view monthly reports by the students they are supervising. Faculty should be able to access the Skills Dashboard. This is similar to the dashboard accessible by students, but instead of a personal view, faculty can see data spanning the entire student body on placement. There is also a section to read employer feedback. This data is provided by employers on a feedback page and can be viewed by faculty through a graph dashboard and they can also read comments left by employers when they entered this information into the system.

For **employers**, Placement 360 should be a system that allows industry supervisors to submit their comments in regards to their placement student's monthly report. They can also leave feedback on where they feel students are lacking from both a technology and employability aspect. This data then goes to faculty who can view trends where multiple employers are saying the same thing.

### **1.3 Technologies**

The technology stack that will be used for Placement 360 is:

#### **Laravel:**

For this project, I chose to use an MVC framework. I chose the PHP-based framework Laravel. This framework is in regular development and has many resources online. I've never used it prior to this project but I have used PHP and MySQL, which should assist in getting to grips with the framework.

#### **Frontend:**

HTML/CSS: for the frontend, this will be the foundation of Placement 360. For CSS, Bootstrap will be used to make the app responsive and modern. The Laravel framework uses the Blade templating engine for the view.

#### **Backend:**

PHP: Laravel uses PHP as its primary language. All controller and model logic will be written in PHP and passed to the relevant views for functionality for the final user.



## Charts:

ChartJS, a powerful JavaScript-based charting framework is used to display various user charts within the web app. The JavaScript code pulls in the values from the database query and builds the chart.

The chart is then passed into the view and is rendered when the page is loaded. Below is a screenshot of the code used to create one of these graphs that is currently available within the web app. This graph returns a pie chart showing how the logged in student's time is spent on different skills/technologies.

```
//Declares and returns the individual technology names from the DB.
$tLabels = Technology::where('userid', $currentUser)->groupBy('techname')->pluck('techname');
//Declares an array for the values.
$tValues = [];

//For each loop getting the sum of the duration and returning it in an array.
foreach($tLabels as $tech) {
    // Getting the sum and assigning it to the $value variable.
    $tValue = Technology::where('techname', $tech)->where('userid', $currentUser)->get()->map(function ($item){
        return $item->duration;
    }->sum());

    // Adding the values to the previously declared array.
    array_push($tValues, $tValue);
}

//Creating the ChartJS pie chart with the data variables passed into the chart.
$techByDuration = Charts::create('pie', 'chartjs')
    ->title("Your Skills by Time (hours)")
    ->labels($tLabels)
    ->values($tValues);
```

Figure 1: PHP code to generate a graph using values from an Eloquent ORM database query.

## Data:

MySQL: for data, a relational database model using MySQL will be implemented to store and return data to the web app. All tables will be created using migrations within Laravel that will allow for easier production deployment and faster production updates as the app is developed.

## 1.4 Structure

This document is broken up into different sections. Below are some brief explanations of what each chapter covers.

- **Requirements**
  - This section outlines both functional and non-functional requirements of the system, including use case diagrams to aid in the development of the system.
- **Design and Architecture**
  - This section includes information on the technology stack, why it was chosen and diagrams of implementation; it is a high level overview of how the system will operate.
- **Graphical User Interface**
  - Screenshots of some of the more important pages within Placement 360 and explanations behind their design.
- **Testing**
  - Details on the different types of testing undertaken during the duration of the project.
  - Unit tests, UI tests and usability tests included.

## 2. SYSTEM

### 2.1 Requirements

#### 2.1.1 Functional requirements

##### Login Use Case

##### Description

The Login Use Case is essential to the application's use. All functionality requires a user to be logged into their account so this will be the first flow a user will complete when visiting the web app.

##### Use Case

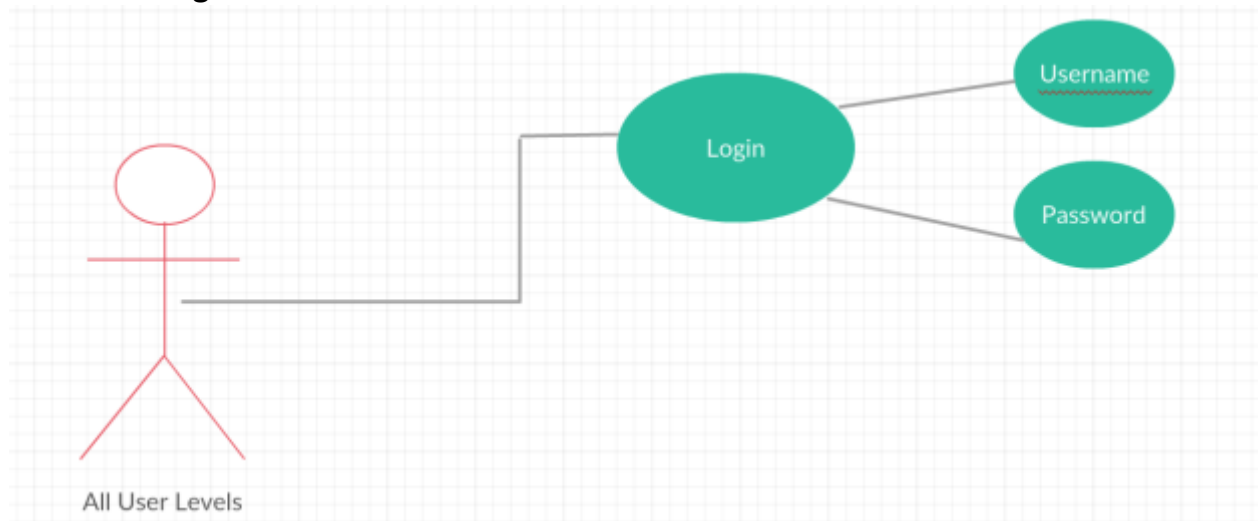
##### Scope

The Scope of the Login Use Case is to understand the way a user will use the login feature of the web app. This use case is applicable to all four levels of user across the system.

##### Description

The steps and process a user goes through to be able to log into the system and use subsequent features.

## Use Case Diagram



## Flow Description

### Precondition

The user needs to use a feature of the web app. For example, a monthly report may be due and a student must use the system to file the report.

### Activation

The user visits the login page of the web app, or they try to visit a page within the app logged out and they're brought to the login screen.

### Main Flow

1. The login screen loads with two required fields: username and password.
2. The user enters their correct username.
3. The user enters their correct password.
4. The user clicks the submit button.
5. The login is successful and the user is directed to their Home screen.

### Alternate Flow

1. The login screen loads with two required fields: username and password.
2. The user enters their correct username.
3. The user enters an incorrect password.
4. The user clicks the submit button.
5. The system rejects the credentials and asks the user to recheck their credentials.

### Termination

The system logs the user into the system and the user is presented with their Home screen.

**Postcondition**

The user now has access to continue to whatever feature they needed to access.

**Monthly Report Use Case**

**Description**

The Monthly Report Use Case details how a student and employer will use the system to submit each of their monthly reports for the past month of work placement.

**Use Case**

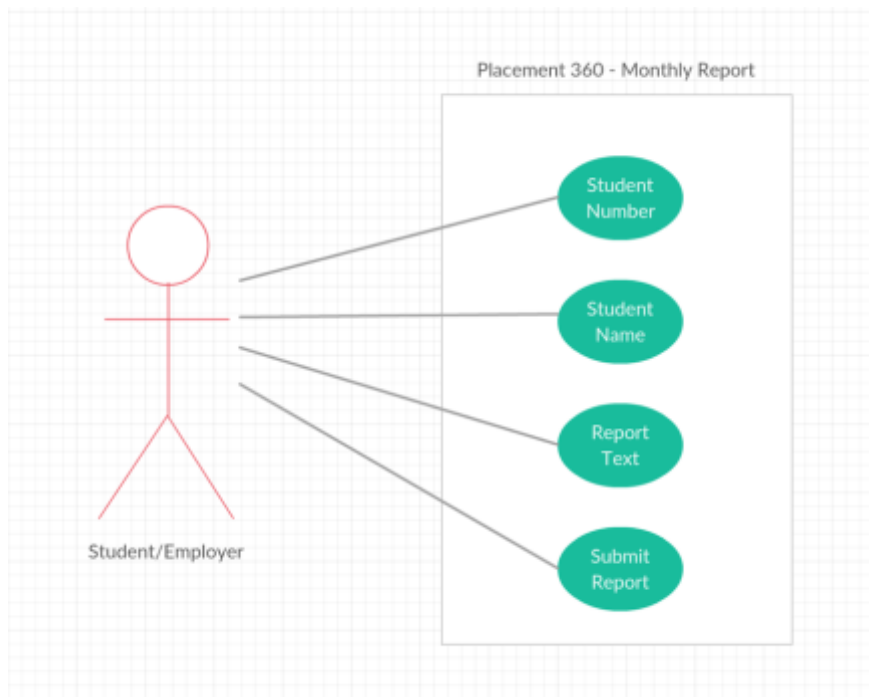
**Scope**

The Scope of this Use Case is to allow users to enter their section of the monthly report. This Use Case applies to two user levels: student and employer.

**Description**

The steps and processes a user goes through to input and save one of the six monthly reports in the system.

**Use Case Diagram**



## **Flow Description**

### **Precondition**

A user must be logged in and on the Home screen of the application.

### **Activation**

A user has clicked on the “Monthly Reports” link from the Home screen and chosen one of the six monthly reports available.

### **Main Flow**

1. The Monthly report screen loads.
2. The user inputs their name and student number.
3. The user inputs their update in a text area.
4. The user clicks the “Save” button.
5. The report is saved and the user is served with a success message.

### **Alternate Flow:**

1. The Monthly report screen loads.
2. The user leaves one of the fields empty.
3. The user clicks the “Save” button.
4. The report isn’t saved and the user is served with a message stating that all fields are required.

### **Termination:**

That month’s monthly report is saved and the user is returned to the main Monthly Reports screen.

### **Postcondition:**

The user has saved their monthly report successfully and can now logout.

## **Generate Graph Use Case**

### **Description**

The Generate Graph Use Case details the steps involved in generating and viewing a graph using the system.

### **Use Case**

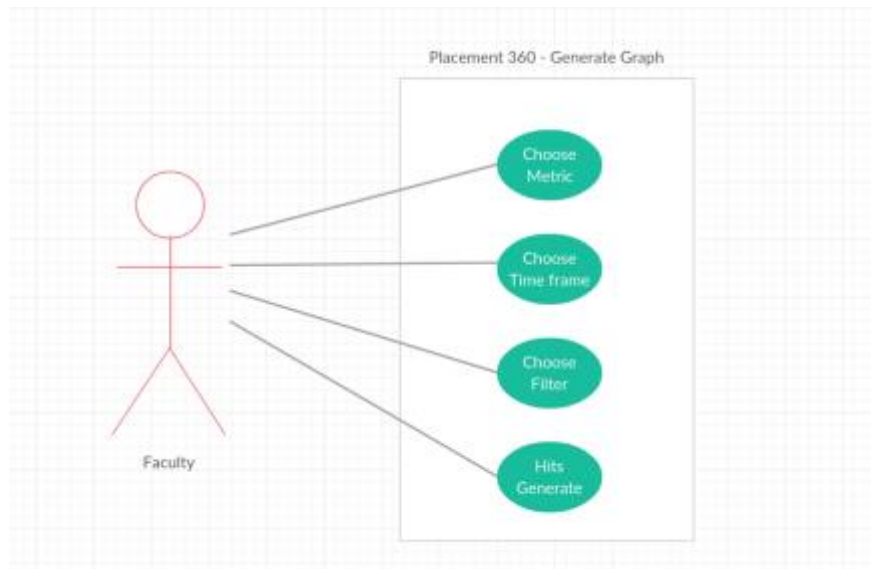
#### **Scope**

The scope of this Use Case allows the user to view the Graphs page and generate one dynamically. This Use Case applies to one user level: faculty.

#### **Description**

The steps involved in generating and viewing a graph.

## Use Case Diagram



### Flow Description

#### Precondition

The user is logged in. The database is already populated with data that can be used by the graphs feature.

#### Activation

The user has clicked on the “Graphs” link from the home screen.

#### Main Flow

1. The user chooses a metric in which to graph.
2. The user chooses a time frame.
3. The user chooses a filter to get a more precise result.
4. The user clicks the “Generate” button.
5. A graph is generated for the user.

#### Alternative Flow

1. The user doesn't choose a metric in which to graph.
2. The user chooses a time frame.
3. The user chooses to filter to get a more precise result.
4. The user clicks the “Generate” button.
5. The user gets an error, reminding them to choose a metric.

#### Termination:

The graph is generated and the user is finished viewing it. They can return to any page in the system or logout.

**Postcondition:**

The user has completed their task of viewing a specific graph and logs out.

**Enter Job Specification Use Case**

**Description**

The Enter Job Specification Use Case allows for users to enter industry information into the system for analysis and data mining.

**Use Case**

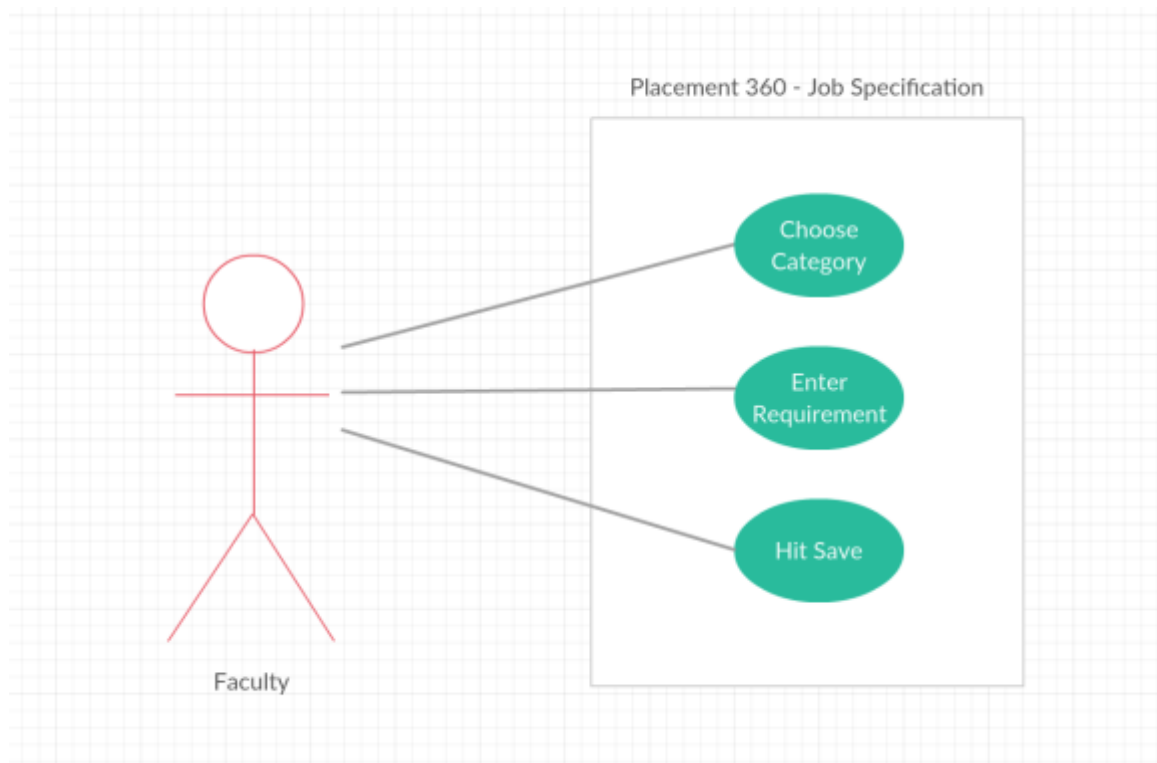
**Scope**

Allows users enter job specifications into the database permanently. This use case applies to one user level: faculty.

**Description**

The steps involved in entering job specification information into the system.

**Use Case Diagram**



**Flow Description**

**Precondition**

The user has logged into the system and is on the Home screen.

### **Activation**

The user clicks the “Job Specification” link.

### **Main Flow**

1. The user clicks “New Entry”
2. The user chooses what category of requirement they want to enter.
3. The user enters the requirement into a text field.
4. The user clicks the “Save” button.
5. The user is shown a success message.

### **Alternative Flow**

1. The user clicks “New Entry”
2. The user chooses what category of requirement they want to enter.
3. The user enters the requirement into a text field.
4. The user navigates away from the page without clicking the “Save” button.
5. The values are not saved.

### **Termination**

The user is returned to the main “Job Specification” page.

### **Postcondition**

The values have saved and the “Job Specification” page loads successfully for the user.

## **Send Email Use Case**

### **Description**

The Send Email Use Case details the steps involved in contacting someone through the system.

### **Use Case**

#### **Scope**

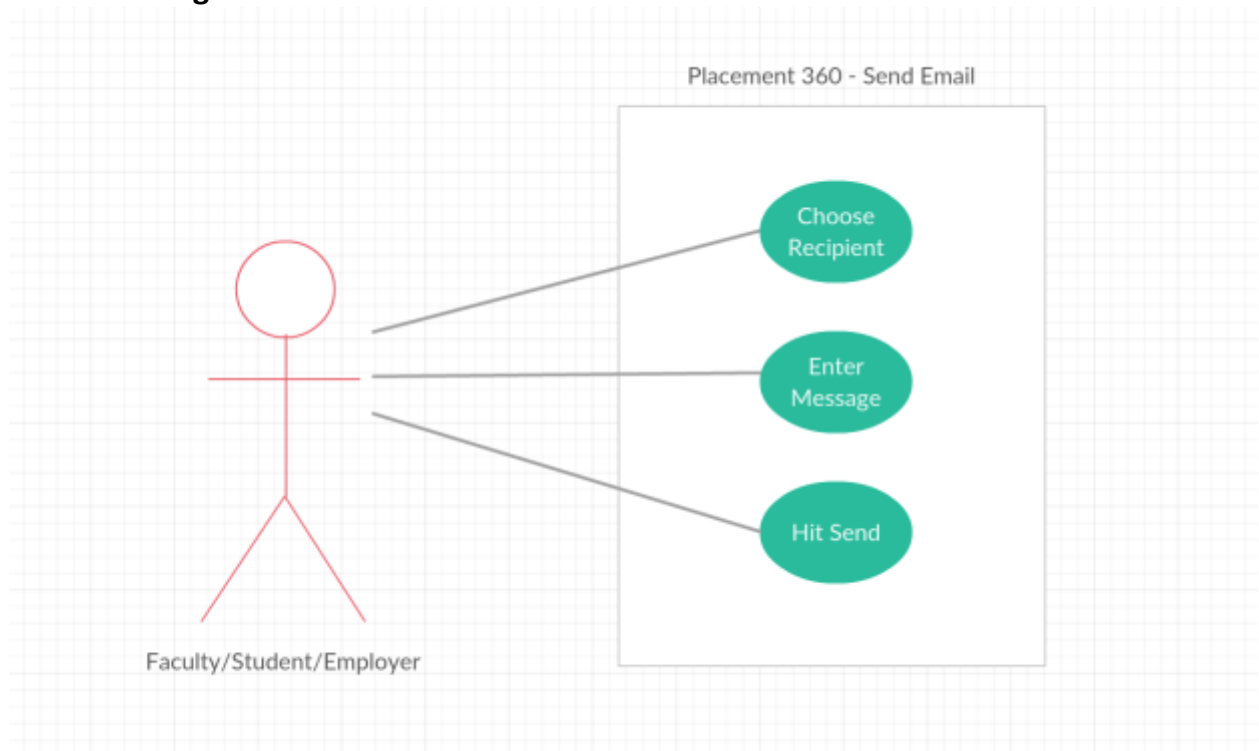
Allows a user to send an email to another relevant person within the system. This use case applies to three user levels: faculty, student and employer.

#### **Description**

The steps involved in sending an email.



## Use Case Diagram



### Flow Description

#### Precondition

The user has logged into the system and is on the Home screen.

#### Activation

The user clicks on the “Message” link from the Home screen.

#### Main Flow

1. The user chooses one of the relevant people from a dropdown menu.
2. The user enters their message into a text area.
3. The user clicks the “Send” button.
4. The user is shown a success message.

#### Alternative Flow

1. The user chooses one of the relevant people from a dropdown menu.
2. The user doesn’t enter anything into the message text area.
3. The user clicks the “Send” button.
4. The user is shown an error message and reminded that all fields are required.

#### Termination

The user is returned to the Home screen.

**Postcondition**

The message has sent successfully and the user is able to continue with other features in the system or logout.

**2.1.2 Data requirements**

NCI Placement 360 has 10 tables to hold data. These tables are broken down as follows:

- Users
  - **users** – this table contains all information regarding users and their accounts.
  - **roles** – this table contains the different user roles available upon user creation. These roles are assigned by an admin at account creation (eg. Student, faculty etc).
  - **role\_user** – this table ties the Role ID from the above table to the User ID of the user logging in. This relationship is how the system differentiates between the various users and roles.
  - **password\_resets** – this table details any password reset attempts made by the user.
- Reports
  - **monthlyreports** – this table contains all monthly reports submitted by the student/employer.
- Skills
  - **technologies** – this table contains the skills that a student enters into the system. This is the table in which the student and faculty dashboards draw their information from when using the system.
  - **employerfeedback** – this table contains all employer feedback regarding students and how they may be falling short. The table stores a priority of 1 – 5 and this is where the Employer Feedback Dashboard takes its data from.
- Data
  - **migrations** – The migrations table contains any migration information from files within the Laravel project.

All database tables were created using migrations. This route was chosen over manual creation due to easier creation and maintenance of the database as the

system evolves. Database migration files are stored within the project and a sample table migration looks like this:

```
Schema::create('users', function (Blueprint $table) {
    $table->increments('id');
    $table->string('name');
    $table->string('studentNumber');
    $table->string('email')->unique();
    $table->string('password');
    $table->rememberToken();
    $table->timestamps();
});
```

Figure 2: Migration code to create the Users database table.

There is also code to roll back the migrations and drop tables if required.

### 2.1.3 User requirements

#### Performance

- When the user loads the login page, the page should be loaded quickly (<5 seconds).
- When the user logs in, they will be authenticated and the Home section should be loaded quickly (<5 seconds).
- When generating a graph, the data should be pulled and loaded into a ChartJS chart quickly (<10 seconds).

#### Responsiveness

- If a user loads NCI Placement 360 on a mobile device on any operating system, the website should automatically load the optimised website.
- When the optimised website loads, it should be easy to login and navigate using touch on a smaller device.
- The user interface should be easy to navigate. To achieve this, at any point when a user is using Placement 360, a clear menu should be visible that can bring them to any relevant page.

## Security

- When an admin user sets up another user account, the password creation should have a minimum standard of eight characters.
- Limited login attempts: the system should only allow five invalid login attempts before suspending the login functionality.

### 2.1.4 Environmental requirements

#### Scalable

- The system should be able to withstand heavy load as a result of increased usage over time and should not crash or become unusable or unstable.
- The system should be capable of being updated easily in production with minimal loss of downtime or none at all.

### 2.1.5 Usability requirements

- The app interface should be minimal and focused to avoid confusion for users.
- Users should be informed of any issues with the system to avoid confusion. If a report fails to save, a message should be shown so the user knows to resubmit.

## 2.2 Design and Architecture

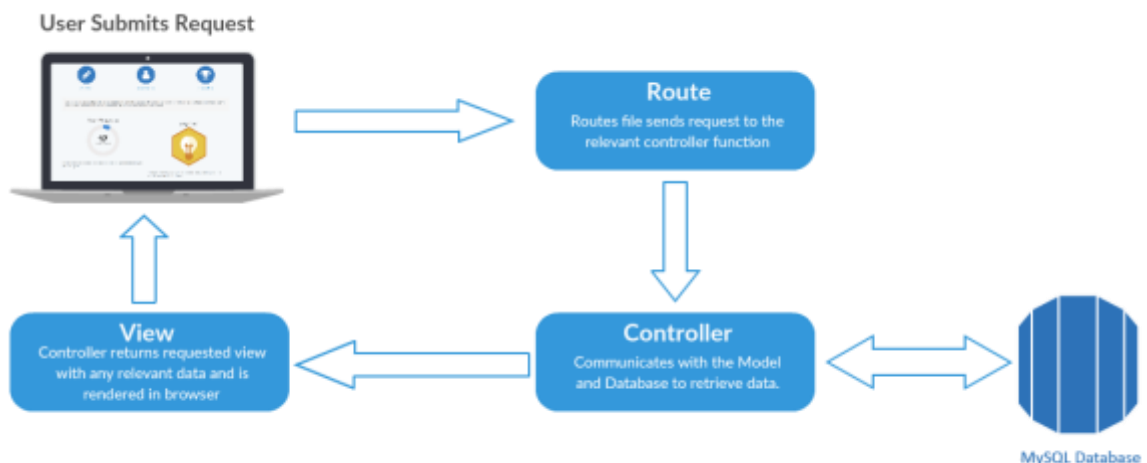


Figure 3: Diagram showing the process of a user request within the system.

Since Placement 360 is a web app, it is device agnostic and will work with all mobile devices. The GUI will render depending on device type size and will scale depending on device screen size.

## View Layer

The view layer of Placement 360 uses the Blade templating engine built into Laravel. This engine allows vanilla HTML5 to be written within the pages as well as both embedded and external CSS and JavaScript. Blade files are saved with the extension “.blade.php” and can be rendered by all browsers as HTML.

Blade can also include PHP logic passed from the controller within the view. The Placement 360 charts dashboard uses this functionality to display charts within Bootstrap <div> sections for styling purposes.

### 2.3 Graphical User Interface (GUI) Layout

The design of NCI Placement 360 is intentionally minimal. The web app is intended to be a means to complete a task, therefore users will want to complete their task as fast as possible. From a design perspective, there is an intentional focus on completing tasks with large text fields and buttons and very few distractions.

The navigational structure of the website is designed to have linked to the main sections at the top, which lead to “portals”. Each portal represents a different section of the website and has its own separate menu.

The logo was designed to represent the purpose of the system: a seamless web app that allows employers, students and faculty to keep in regular contact as well as always being able to use features of the system.

### Login Page

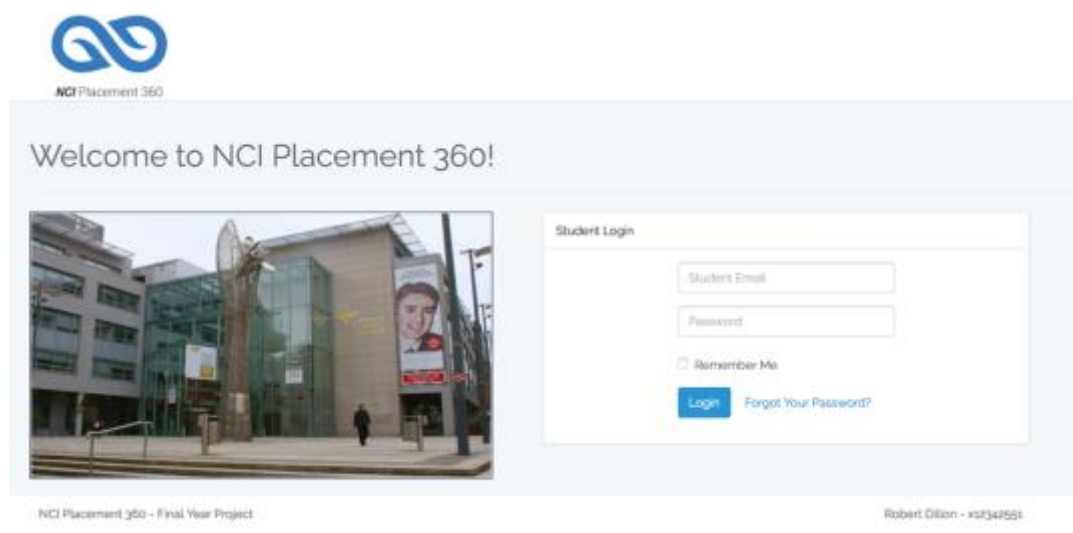


Figure 4: Login page displayed to all users.

The login screen of Placement 360 is simplistic, with the logo at the forefront and large username/password input fields at the forefront. A submit button is directly below. Along with the image of the NCI campus this makes up the entire page.

### Monthly Report Portal

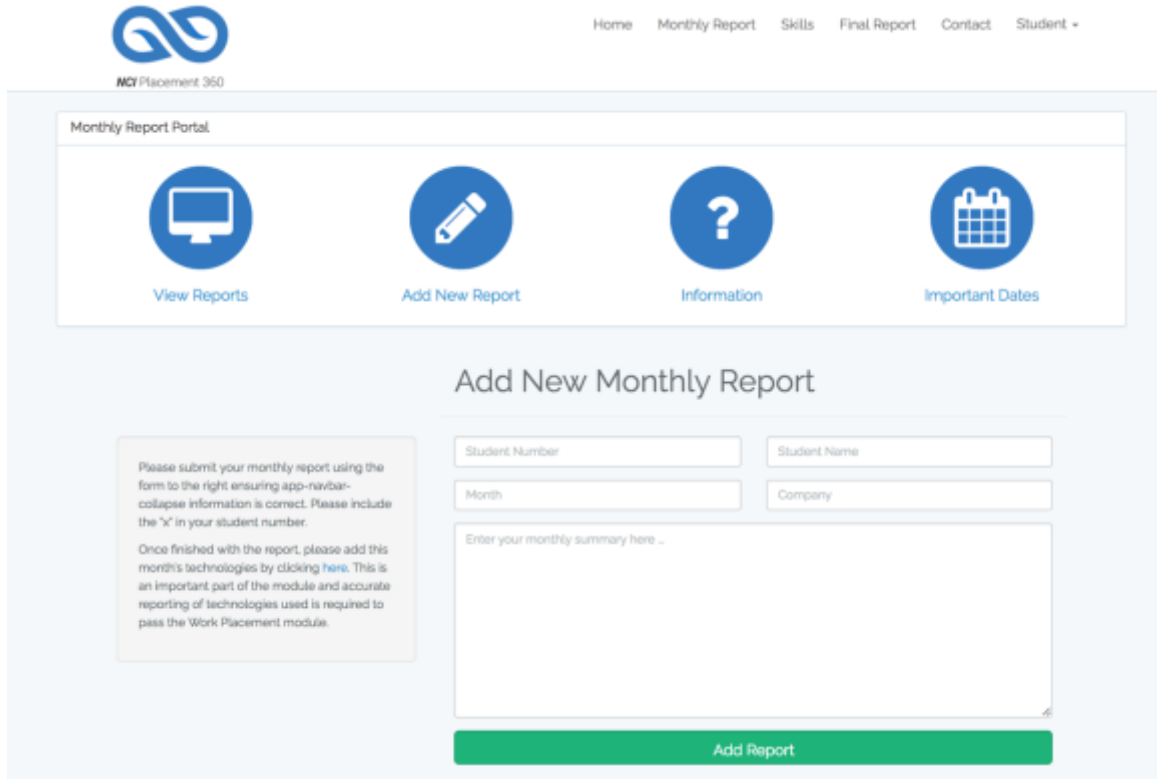


Figure 5: Student monthly report submission page.

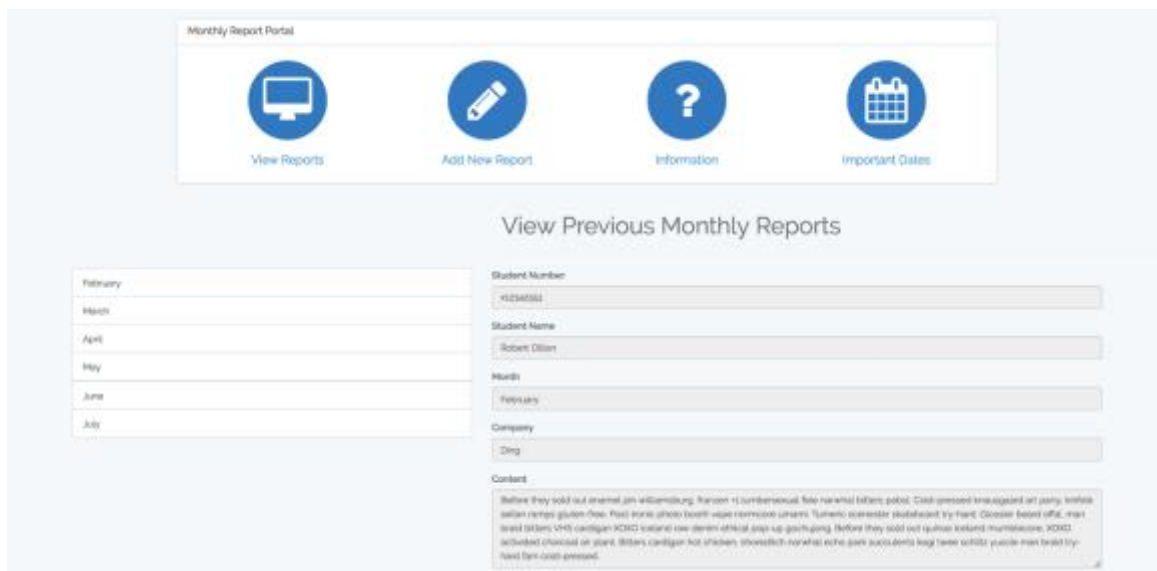


Figure 6: View previous monthly reports page.

The monthly report page will be used by students and managers. It allows entry of all fields currently on the paper method. Users can enter the fields and click Submit, navigate to another page within the system or log out using the menu.

The screenshot above shows how the user can browse into different elements of the monthly report portal.

### Student Skills Portal

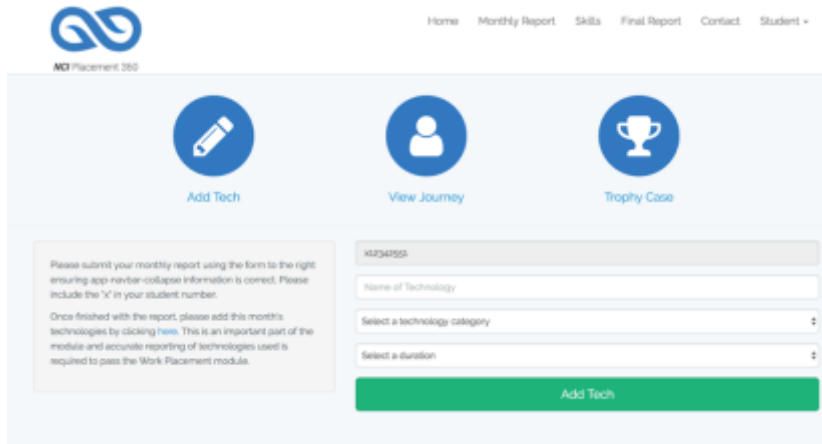


Figure 7: Student skill submission form.



Figure 8: Student skill visual dashboard.

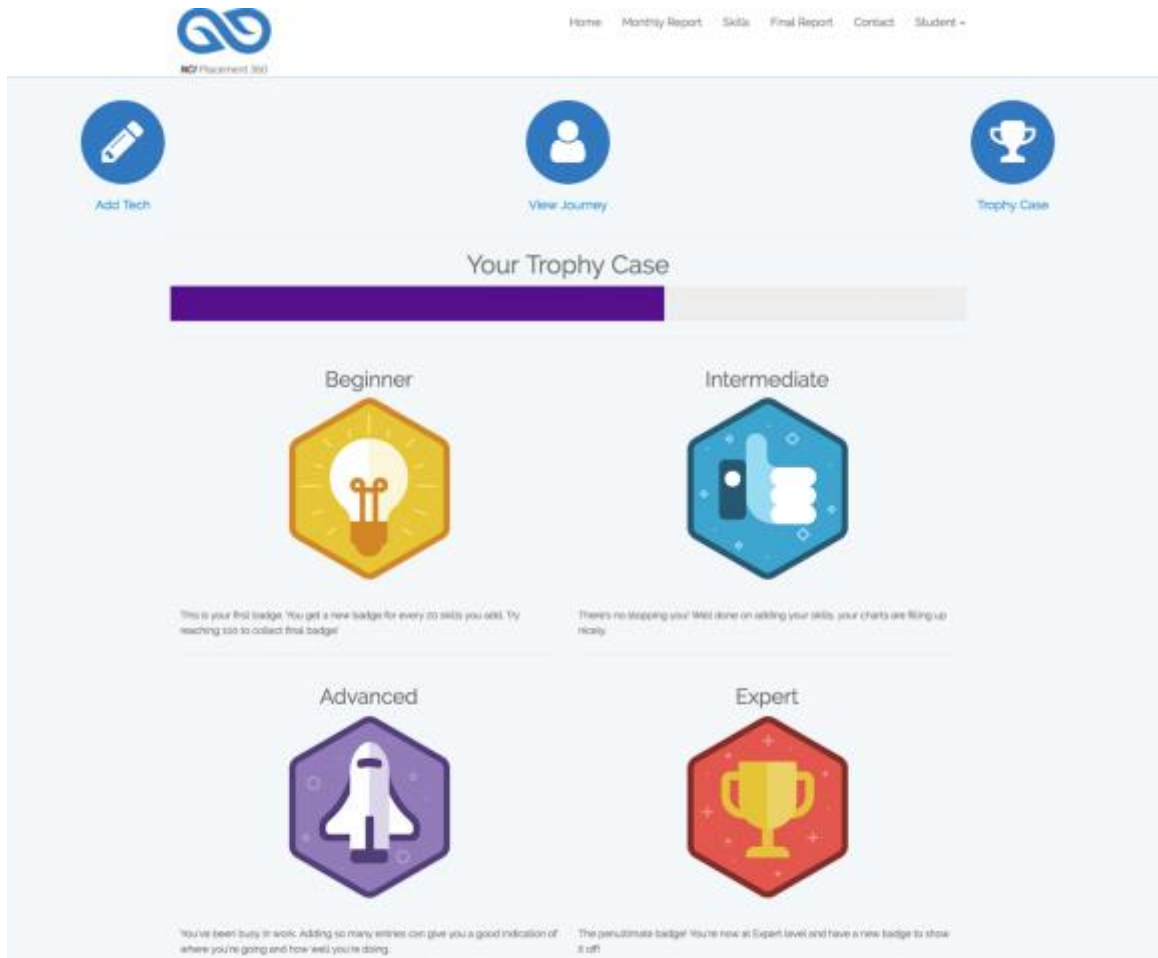


Figure 9: Student trophy case and progress dashboard.

The Skills portal for the student has the skill entry form. Once skills are entered, the student can visit their dashboard and see a visual image of the skills they’ve used during their placement. Gamification aspects can be seen within the dashboard as well as the user’s trophy case.

The Contact Supervisor page allows both students and managers to send a message to their NCI contact within the college. A common complaint from lecturers is that due to the volume of emails they receive, emails without adequate information takes up time.

With this page, all fields are required and in the forefront so when the email is sent to the supervisor, the important information will be at the forefront of the mail.



## 3 TESTING

### 3.1 Unit Testing

A suite of unit tests are included to test the app and its functionality. For some of the development process, Test Driven Development (TDD) was used as a way of ensuring code quality and functionality. These tests were written in PHPUnit and can be run easily. So far these tests have been passing using an open source test runner within the Atom text editor.

```
public function testNavUsername()  
{  
    $this->visit('/')  
        ->type('student@me.com', 'email')  
        ->type('sim2000', 'password')  
        ->press('Login')  
        ->see(Auth::user()->studentName);  
}
```

*Figure 10: Unit test checking for the user's correct username upon login.*

### 3.2 UI Testing

UI testing is a very effective way of testing the various functions of a website that unit testing cannot. UI tests include opening a browser and filling in the forms automatically. The code for these tests are more complicated than unit testing code but they offer a much higher test coverage percentage and hold the web app to a higher standard in regards to user interface bugs.

Selenium and Chrome Webdriver is used to write UI tests for NCI Placement 360.

### 3.3 Test Driven Development

Development of the project didn't completely follow a TDD approach but for certain elements, such as adding in monthly reports, a TDD approach was used. This initially failing test passed once the functionality was successfully added. This method gave valuable experience in writing quality code.

### 3.4 Iterative Testing

During the development period, time to test new features will be put aside as they're developed. For example, if the part of the system that allows the user to submit monthly reports has been developed and implemented, iterative testing of that feature will be completed immediately to make sure it's saving to the database, there's validation to make sure there's text being submitted and the supervisor can see the reports among other things.

### 3.5 Regression Testing

Iterative testing is useful for making sure a new feature isn't broken but time will be set aside for regression testing of previously introduced features. This is an important element of testing as newly developed features could potentially break older features. Regression testing will prevent such bugs from falling into production.

### 3.6 User Acceptance Testing

As features were complete, potential end users tested the system and gave valuable feedback. These potential end users consisted of classmates who were familiar with the method that NCI Placement 360 was replacing. Think Aloud techniques were used during these tests to get an accurate idea of issues that may arise if this application was used by work placement students.

### 3.7 Trunk Testing

I completed a trunk test with two separate NCI students to evaluate the ease of use and value of the application.

#### User 1:

| Question                   | Answer                                 | Correct/<br>Incorrect? | Time Taken |
|----------------------------|----------------------------------------|------------------------|------------|
| What type of site is this? | A site for internship students in NCI. | Correct                | 25 seconds |
| What page is this?         | A page to enter a monthly report.      | Correct                | 10 seconds |

|                                        |                                                                                  |                |                   |
|----------------------------------------|----------------------------------------------------------------------------------|----------------|-------------------|
| What are the main points of this page? | Information and a form for entering data.                                        | <b>Correct</b> | <b>10 seconds</b> |
| What are the navigation options?       | I can move around using the submenu or go to other sections using the main menu. | <b>Correct</b> | <b>5 seconds</b>  |
| Can you navigate home?                 | <i>User navigates home using the logo.</i>                                       | <b>Correct</b> | <b>5 seconds</b>  |

**User 2:**

| <b>Question</b>                        | <b>Answer</b>                                               | <b>Correct/<br/>Incorrect?</b> | <b>Time Taken</b> |
|----------------------------------------|-------------------------------------------------------------|--------------------------------|-------------------|
| What type of site is this?             | A web app for when I was on placement.                      | <b>Correct</b>                 | <b>20 seconds</b> |
| What page is this?                     | A dashboard with lots of charts.                            | <b>Correct</b>                 | <b>15 seconds</b> |
| What are the main points of this page? | A submenu that leads me to other sections.                  | <b>Incorrect</b>               | <b>20 seconds</b> |
| What are the navigation options?       | I can go home or to other sections using the header links.  | <b>Correct</b>                 | <b>10 seconds</b> |
| Can you navigate home?                 | <i>User navigates home using the home link in the menu.</i> | <b>Correct</b>                 | <b>5 seconds</b>  |

## 4 IMPLEMENTATION

### 4.1 Overview

NCI Placement 360 uses a number of different technologies that work with each other to give the end user a seamless experience. Implementing the MVC

framework, Laravel, was quick and efficient. The fast installation on a development machine meant that development of the project could be commenced very quickly.

The majority of development was completed within Apple's MacOS operating system. Once the Terminal was in the correct directory, the command "*laravel new Placement360*" created a directory with the bones of a Laravel application. All other dependencies for the project (MySQL, Apache etc.) had already been installed for other projects.

Adding new sections to the project followed the same process, which is similar to most MVC methods of development:

1. Create the controller.
2. Create a Blade view file.
3. Add a route to the *web.php* file that points to a function within the controller.
4. Write the application logic within the controller. Most of these controllers then return the view.

Development of a section would generally include working within both the controller and the view file simultaneously.

The majority of this process was replicated throughout development, with only the controller code changing to suit the different features. An example controller function is below.

```
public function store(Request $request)
{
    $userId = Request::all();
    $userId['userId'] = Auth::user()->id;

    $input = Request::all();
    Monthlyreport::create($userId, $input);
    return redirect('monthlyreport')->with('info', 'Monthly report added!');
}
```

Figure 11: Function to store a student's monthly report upon submission.

## 4.2 Role-based Access Control

As the system is designed for multiple different types of users (students, employers, faculty), it was important to have a specific user account type that shows only what

each user needs to see. The system currently has four different user levels; the three mentioned above and an admin user level.

To achieve this functionality I used a package for Laravel called Entrust. With this, I migrated new tables to the database:

- a table for the roles and the role information
- a table that associates a role ID with a user ID
- a table for permissions and permission information
- a table that associates a role ID with a permission

Permissions weren't used as the functionality didn't require it but the tables remain in case the system evolves and permissions become a requirement. Permissions would be a more articulated system than what was required but it could prove expensive to add this functionality in later and leaving the unused functionality doesn't negatively affect the app in any sort of performance way.

There are different menu items for different users to restrict access to navigating to a page they shouldn't have access to. Within the Blade templating engine, you can choose which role can see certain elements, like so:

```
@role('student')
<li><a href="{{ url('/') }}">Home</a></li>
<li><a href="{{ url('/monthlyreport') }}">Monthly Report</a></li>
<li><a href="{{ url('technology/add/addtech') }}">Skills</a></li>
<li><a href="{{ url('/finalreport') }}">Final Report</a></li>
<li><a href="{{ url('/') }}">Contact</a></li>
@endrole
```

Figure 12: Code determining if the user is a student and showing their specific menu.

If a user manually types a restricted address into the URL bar, they will get the following redirect with a message:

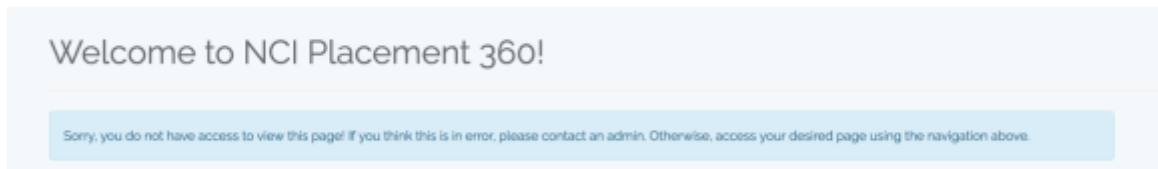


Figure 13: Secure redirect from a restricted page with an information message displayed.

This functionality is achieved using **middleware** that is called within the controller to check the user account type before returning the view.

### 4.3 Gamification

A key element in Placement 360's success is user participation, specifically that of the student. In order to encourage students to submit skills as often as possible, gamification was implemented to the skills section. The first graph shown on a student's dashboard is a progress circle showing how many entries they've made. This graph is dynamically populated and has a completed value of 100 (this is the number that is encouraged of students but isn't an enforced limit; there is no limit to the amount of entries a student can make).

```
$userCount = DB::table('technologies')->where('userid', $currentUser)->count();

$techProgress = Charts::create('percentage', 'justgage')
    ->title('Your Progress')
    ->elementLabel('Skills Added')
    ->values([$userCount,10,100])
    ->responsive(false)
    ->height(300)
    ->width(0);
```

Figure 14: Code to get a user's current entry number and displaying this in a progress circle.

On top of this, a student is given a badge. There are five badges to collect and the dashboard shows the most recent badge they've achieved. There are new badges for every 20 entries. A student can also view their trophy case, which shows all badges collected. Once all badges are collected, the student gets a congratulations message and is encouraged to continue adding entries to their log.

### 4.4 Security

Placement 360 contains sensitive information such as emails and passwords, student specific content regarding their course and faculty information and as a result, security was an important concern during development.

#### 4.4.1 Passwords

User passwords within the database are stored with bcrypt encryption. This password hashing method is considered to be highly secure, which is why I chose it.

#### **4.4.2 User Account Verification**

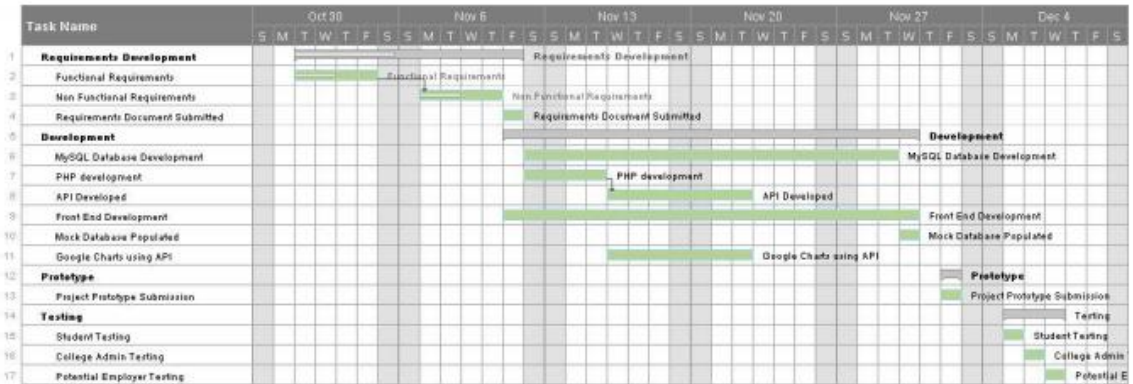
As mentioned within 4.2, user account types are implemented within the system and this is used to make sure pages are secure. If a student tries to access a restricted page for faculty, they will be redirected. If a logged out user tries to access a page that's restricted to any user level, they will be redirected to the login page.

#### **4.4.3 Incorrect Password Restriction**

To prevent brute force attacks, the Placement 360 system restricts incorrect logins to five attempts. After this, users must wait 60 seconds before trying again.

## 5. APPENDIX

### 5.1 Project Plan



### 5.2 Project Proposal

#### Objectives

The objective of the project is to provide a system for college staff, students and employers to interact with each other during the student Work Placement module. This system will have three areas broken down for the three main users: college faculty, students and employers. The objects for each of these users are as follows:

- College Faculty
  - Allows faculty to enter job specifications into the system, specifically technical requirements of the jobs. These will be saved in a database and staff will be able to generate dynamic graphs and statistics about what technologies are in-demand in the industry. This can then be used to evaluate what should be added to the curriculum for computing courses based on what the industry is currently looking for.
  - Allows faculty to see a student’s rating in their final report, which staff rate students from 1 – 5 on various areas of the internship. This will allow faculty to generate graphs about student performance on their placements.
  - Allows supervisors to view monthly reports submitted by the student and employer. It also allows them to receive and send messages to both the student and the employer. Email notifications will be sent to an email address of their choosing.
- Student



- Allows students to submit their monthly reports and final report through the system, as well as message their supervisor.
- Employer
  - Allows employer to submit their section of the monthly reports, message the supervisor assigned to the student and enter in job requirements to the system for future placements.

## **Background**

In many courses within the School of Computing in the National College of Ireland, a Work Placement module makes up the student's second semester in 3<sup>rd</sup> year. This is a six-month placement. Many students begin looking for placements between the end of 2<sup>nd</sup> year and the start of 3<sup>rd</sup> year. Most placements begin in January.

Internships are six months long, so if one starts in January, it will finish in July. During this time, there is supposed to be regular contact between the student and the college as well as the employer and the college. This includes an on-site visit from the college supervisor as well as a final report done by the student and a survey filled out by the employer for the on-site visit.

At the minute, there's no all-round system between the student, college and employer that handles all work placement requirements and my goal is to solve this. The application will be used from when students start looking for placements up until they complete their internship in the summer following 3<sup>rd</sup> year. The application will take this all into consideration.

During my work placement, I worked as a QA Engineer in Ding. For my monthly reports, I would need to download the template every month and enter in my section in Microsoft Word. Then I would email it to my manager and she would fill out her section. She would print it, sign it and give it to me to sign. I would sign and scan the document before uploading it to Moodle. This was a lengthy process and I think there should be an easier way to manage this.

I also noticed that when we were organizing the on-site visit, it took a while to finalize the details over email and then, when the time for the visit came, we had more paperwork to fill out. This paperwork included a lengthy rating of the student that included valuable information that could be mined. This is where I think the system would be a useful addition. For user experience, it makes more sense to do it online and we'd also have this information archived for future reference.

These are observations from a student point-of-view. When speaking with my supervisor, she made me realize there were many benefits from an NCI's perspective.

Currently there isn't a system for aggregating statistics from internships. This starts from early in the process such as job requirements. It would benefit NCI to have a system that takes information from employers and aggregates them. The system could then generate dynamic graphs to display this information. The most useful element of the application will be the feedback that it provides to teaching staff. For example, a lecturer will be provided with information about java and given clear examples of how it is used in the company. In return, providing the lecturer with up to date examples to use in the classroom.

The system could also aggregate information from a student's final report and graphs could then be generated using data from all students. This in return could provide information to the college where students are slipping during their internships and what areas require more attention.

One complaint I heard from students in my class was that there wasn't enough contact between the college and students during their internship. This system could provide a way to keep both the college and student in contact— specifically their supervisor.

Work Placement makes up quite a big part of a student's time in NCI and I think it's time there was a dedicated, end-to-end system for all three parties involved: students, faculty and employers.

## **Technical Approach**

### **Research**

My first step is to do research on the application. I've already spoken to my work placement manager and other students about their thoughts on the current process, specifically asking if they think my system will be an improvement and I have received a good response. I plan on fleshing this out to include a survey that students, managers and college faculty can fill out to give me an idea of what the most pressing needs are.

## **Preparation**

I've been researching the technologies I want to use during this project and luckily I'm familiar with all of them to an extent but I will still need to learn more. I'm going to be using Pluralsight to complete courses around the programming languages and databases design, I will be using during the project. I have also looked up the library catalogue and there are some good, recent books around the technologies I want to implement.

## **Requirements Specification**

Once I have a solid idea of what features I want to include in the project, I will lay out the functional and non-functional requirements of the project. For non-functional requirements, I'll design use case diagrams and the various flows that different users can go through using the system among other things. For functional requirements, I will look at performance, availability across multiple devices and security as well as other requirements.

## **Prototyping**

Once I have the non-functional requirements laid out, I want to begin working on a usable prototype that I can show in my mid-point presentation. During the prototyping phase, I will do regular testing and get regular feedback from potential users.

## **Development**

Once I'm happy with my prototype, I will flesh the system and develop it to include all required functionality from the functional and non-functional requirements.

## **Technical Details**

### **PHP & MySQL**

For the core functionality of my mobile friendly web app, I want to write most of the code in PHP and store information in a MySQL database. I've looked up language alternatives such as Python or Ruby but I'm most experienced with PHP and it ticks all the boxes. I've also looked up some NoSQL database options like MongoDB and Firebase but I'll stick with MySQL. I love working in phpMyAdmin so I'll be able to dive right in.

## **JavaScript**

I want to make sure the web app looks professionally designed and easy to use , so I want to incorporate a JavaScript framework into the app. I worked with KnockoutJS during my own internship and really liked some of its features such as automatic UI refresh. This particular framework is only 54kb when minified so it will help with my speed requirements.

## **Bootstrap**

I have used Bootstrap a lot during college projects to make sure my apps work across all devices so I will use it once again here. Having the website responsive is important so that it can work across multiple platforms. The idea of being able to work on my monthly report on the go during my own internship would've been ideal so I want to make the system as mobile friendly as possible.

## **Google Charts**

During my databases project last year, I used Google Charts to dynamically generate different types of mobile friendly charts and graphs using data from my database to visual information. This will be implemented for college faculty users to see what employers are looking for, how students are getting on etc.

## **Apache**

I plan on hosting my project on an Apache web server. It's the web server software I'm most familiar with and it ticks all the boxes so it seemed like the best option.

## **Evaluation**

### **UI Tests**

During my internship, I worked as an Automation Engineer. I helped develop a framework for one of our products and wrote many UI tests. I plan on doing the same for NCI Placement 360. I would like to include tests to verify login is working correctly, monthly reports are saving, graphs are generating, emails are sending correctly etc. These could be executed at the click of a button at any point and will run in the background. These would be dynamic and I will be notified of any failed tests so I can fix any bugs that may be introduced during development.

### Iterative Testing

During the development period, I will include time to test new features as they're developed. For example, if I've developed the part of the system that allows the user to submit monthly reports, I will test that feature immediately to make sure it's saving to the database, there's validation to make sure there's text being submitted and the supervisor can see the reports among other things.

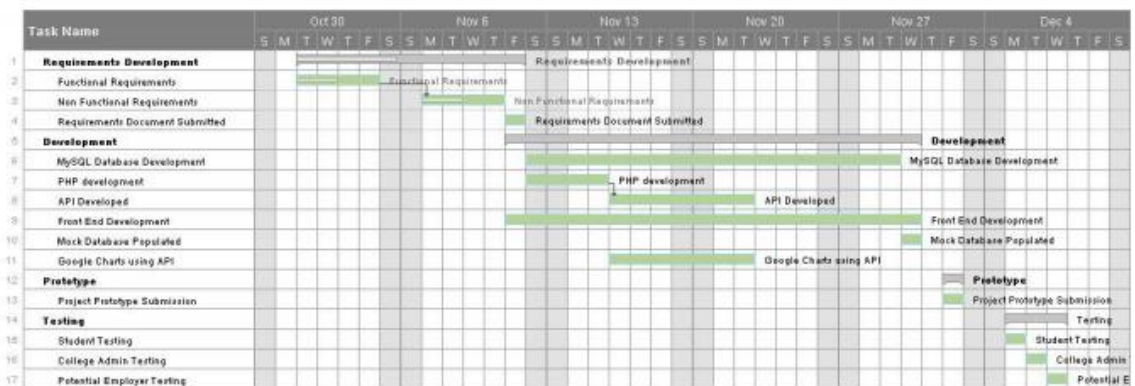
### Regression Testing

Iterative testing is great for making sure a new feature isn't broken but I also want to set aside some time for regression testing of previously introduced features. I won't do this as often as the other two testing methods but I think it's important to check regularly that all functionality not covered by a UI test is working as expected.

### End User Tests

Once the system is at a state where I think it's ready to be tested by outside people, I will hopefully find other students who can test it for me and give me feedback. I would also like to find college faculty to test the system too, particularly the graph functionality once I've populated the database with mock information.

## 5.3 Project Plan



## **5.4 Monthly Journals**

### **September**

September has mainly been spent working on my project idea and how I'll achieve it. My idea is an app for restaurants that lets the restaurant put tablets or phones on the table to allow people to order food and drinks from the tablet, request the bill etc. I'm hoping to make the app using Android Studio and I'm going to use Firebase to store my data. The app will need to have pretty good push notifications and Firebase has some great ways to incorporate this functionality into Android apps so it seems like a good choice. It's also free up to a certain point and from looking at the tiers they offer, I'll never need the paid tier during the project.

I had my project proposal meeting on Wednesday and I'm not sure it went too well. I'm not sure I got my pitch across too well. I haven't found out if I was approved or rejected just yet but it's anybody's guess. I'm just going to continue on working on my original idea and if it's rejected I'll deal with it then. I really like my idea and think it solves a problem that doesn't currently have an existing solution.

This weekend I'm going to continue looking at Pluralsight courses on Firebase and Android studio. Not much else happening this month and I haven't had any supervisor meetings because I haven't been assigned one just yet.

### **October**

#### **11/10/2016**

So I was told by Eamon yesterday my project was rejected. I thought the presentation went bad and I was a little annoyed but I'm not going to continue on with it. I'm happy enough to let it go. I looked at the Approved Projects List and came across one I liked. A feedback system for students who are on work placement. It's almost a 360 solution for students, NCI staff and employers. I met with Lisa Murphy and she's happy for me to go ahead with it so I'm going to crack on with my Project Proposal this week.

#### **13/10/2016**

Yesterday I spent a lot of the day working on my Chess Game project for Introduction to AI. I'm doing well but just need to work on getting my last piece done (as well as fix some bugs). This meant I didn't get a whole lot done on my project proposal. I've just been jotting down some notes for each heading and then

I'll narrow it down to what I think is manageable. There's a lot of ideas I have about expanding the features and functionality so I'll be focusing on this today after work.

### **15/10/2016**

I've spent the past couple of days researching technologies for my project and also doing some "unofficial" research about if it'd be useful or not by quizzing my manager in work and other students and the response was positive. So yesterday and today I spent a fair bit of time on my Project Proposal and was happy enough with what I have so far. I've sent it to Lisa to see if she has any thoughts on how to improve it. I've still to do my Gantt Chart but I didn't have access to Microsoft Project at home but I'll work on that this week. I've just spent two hours trying to add the last bit of functionality to my chess game for this deliverable but no luck just yet. Will be trying again tomorrow!

### **21/10/2016**

Lisa got back to me earlier this week with some changes to my proposal and also some tips on how to improve it. I've made these changes and also got my Gantt Chart done. I've just submitted it and am happy enough with the document. I kept looking over the proposal trying to make improvements but I knew there was nothing more I could do so I submitted it. I'm really happy with how the idea is fleshing out.

### **24/10/2016**

I haven't even looked at my Requirements Doc until this evening. I was busy working through tutorials for my App Development CA, which was this morning. Thankfully it went pretty well! With that out of the way I was able to take a look at the next Software Project deliverable and I began working on Use Cases and different user flows that can be completed with the system.

This is turning out to be useful, coming across some edge cases that I hadn't thought about so it's good to have these out in the open sooner rather than later!

### **30/10/2016**

I've been working on the diagrams for my requirements document when I had the time over the past couple of days. I don't enjoy diagramming requirements as much as I do writing about them but down the line I'm sure the visual aid will be helpful.

I'm almost finished my first draft of the requirements document so I'll probably upload it at some point next week once I review it fully.

## **November**

**2/11/2016**

I always find my productivity takes a dive during reading week but this one hasn't been so bad. I spent my day today looking at a Pluralsight course on JavaScript frameworks and I think I'm going to use React. This will be down the line of course and I'll definitely be developing my database and server side software first but I'm most unfamiliar with JavaScript and its various frameworks so being the control freak that I am, I thought it'd be a good idea to look this up anyway.

There's some really nice UI examples that can be easily implemented. I really want to find the time to make my project look really nice as well as be functional even if the functionality should be centre stage. Hopefully it works out!

**8/11/2016**

I just found out the Technical Requirements deliverable was extended, which is a big help. I'm confident with my use cases but this gives me some extra time to work on my diagrams. I know it's a living document but it's nice to have a good first upload!

**18/11/2016**

I've been busy with other modules over the past couple of weeks but I've been slowly chipping away at my requirements specification document and I've just uploaded it today. It looks good and I'm happy with the results, it'll give me a head start on my Technical Report.

**25/11/2016**

The past two days has been spent familiarising myself with ReactJS since I'll be using it for the frontend of my web app. This included blitzing the Pluralsight course and getting some hands on experience with some basic stuff - it's a really nice framework and since it's gaining popularity, it's a good arrow to have in my quiver once I graduate.



### **27/11/2016**

I've managed to set up a React app on my local machine and have been tinkering with the basics. It's not what I'm used to at all and it'll likely be a lengthier process than vanilla JavaScript but if it makes for a better project, I'm excited to go ahead with it!

### **28/11/2016**

I sent Lisa an email with an update today just to keep her in the loop. I haven't had any meetings since the initial one but I haven't really needed to have one since I've been working on documentation and taking baby steps with the technical side of things. This year's modules are very time intensive so I'm going to focus on getting a lot done over Christmas. I can't imagine much free time will become available before then because of my other projects.

### **November Summary:**

- Finished Technical Report and submitted it.
- Watched Pluralsight course on React.
- Installed React and began to play with it on my local machine.

### **December**

#### **3/12/2016**

I've been working on my technical report for the past while, getting little bits done here and there but I'm going to get a chunk of it done over the next few days I think and send it over to Lisa for feedback. Still not 100% set on my technology side of things even though I've already started a prototype but I definitely plan on nailing down the finer details by the end of the semester and full steam ahead on development over Christmas and the New Year.

#### **8/12/2016**

I've been working on my report and I've sent it over to Lisa today for some feedback, I think it's detailed enough and I'm assuming some of the chapters in the template can be removed because they're just not applicable yet. I can add them in as they become relevant since it's a working document.

### **9/12/2016**

Feedback from Lisa was really good, it's definitely improved the document and it's in a good state to be uploaded I think! It's good to get a lot of the technical stuff like user flows and architecture written down and drawn up respectively.

### **13/12/2016**

With the document submitted, I think it's time to start a presentation for my mid-point. Mine is on the 20th and a lot of other module stuff has been pushed back and is due on the 23rd so it'll be a very busy ten days ahead so it'll be nice to get this done soon. I've decided to use Prezi but I'm double checking with Lisa to make sure that's good.

### **16/12/2016**

I think I'm happy enough with the presentation and I've also done up some nice mobile mockups. I've also been working on the prototype and although it doesn't do a whole lot, I think it adequately shows the potential and how fast I can built on the barebones scaffolding. Since submitting my technical report I've decided to move from React to Angular so the report is already slightly out of date and Lisa and Anu, my mid-point judge will likely have read the old one but I can explain that in the mid-point.

### **20/12/2016**

I think that went well! I got my points across and Lisa had some good feedback both on my logo and on where the idea can be improved. It was her idea first so I'm keen to take her feedback on board about how things can be better executed. I think it's time to focus on submissions now!

### **29/12/2016**

I decided that the time after my final submissions and January will be project time and I've been doing some Pluralsight courses over the Christmas so it's not too intense. I'll be studying for exams from this point on but I have ten days at the end of my exams before term starts back to get some good work done!

### **December summary**

- Submitted my technical document.
- Changed my tech stack to Angular for the View.
- Had my mid-point presentation, which went well.
- Got more Pluralsight and development done.

## **January**

**10/1/2017**

No project work so far since I've been busy with exams but I began working again today. Finding my progress is very slow. Not entirely sure if moving to unfamiliar ground technology-wise was a clever move for the final year project.

**15/1/2017**

I spent the past few days looking at PHP and MySQL, specifically Laravel. I spent a fair amount of time last semester so moving to this stack means ditching my work but the code I yielded from the time spent last semester wasn't a lot. I think I can get back up to where I was with a couple of days solid work. I think I'll move over to PHP and MySQL.

**18/1/2017**

I've been using Laravel, a PHP framework and it's going quite well. Very happy with how fast I'm moving. It's an MVC framework for PHP so it's familiar territory since I'm familiar with PHP and MVC technologies.

**22/1/2017**

I'm back up to where I was before Christmas. I have user registration and login working as well as the monthly report functionality writing to the database. My next step is to add different user levels since the system will be used in various different contexts (student, lecturer, employer etc). This will be difficult.

**25/1/2017**

I had another meeting with Lisa today. Good advice from her to keep ploughing ahead with project work while the semester is slow. No doubt it'll be very busy in a few weeks so I want to get as much done as possible.

**30/1/2017**

My access control system is slowly getting there. I'm using a couple of new database tables. One to store different user levels and another that stores the different users to the role assigned to them. This will allow me to restrict different pages using middleware. I was worried this would bring my project to a standstill but it's nice to be moving faster. Laravel is a great framework and the template engine means I can build the UI very quickly. It's still responsive because I'm using bootstrap so I'm still working closely to my requirements.

### **January summary**

- Moved to PHP/MySQL using Laravel as a framework.
- Set up user registration and monthly report functionality.
- Role based action control is set up, which allows me to show/restrict different pages for certain users.

### **February**

**3/2/2017**

I've noticed my code is getting sloppy as I add new functionality to the project. I keep telling myself that I'll just refactor it all at the end since nobody is collaborating with me but I don't want to leave a mountain of work for myself in the final stages. So today I spent a chunk of time cleaning up my template files and I'm going to try put more time into this going forward.

**4/2/2017**

I've decided to include a portal on each of the main pages in the application, which allows easier navigation for the user. Most of my time recently has been writing functionality in PHP and doing minimal UI stuff so it was nice to break away and use Font Awesome and CSS to write a nice submenu portal.

**8/2/2017**

Back to working on functionality. I've added in the Technology section of the website and am working on getting the view for the student working. This'll mainly be for individual students to see their progress. Allowing faculty to see a whole view of all students will be tricky but I'll come back to this later.

**16/2/2017**

More template changes along with some access changes for users. Some slight additions to the technologies section, I'll work on this more in-depth later. I've been putting a big focus on the final project and my other modules have deadlines and CAs creeping up. I'll have to focus on these for a few weeks.

**24/2/2017**

Today was a busy day. I had a report for Usability Design due and a CA test in Distributed Systems. The report went really well and I'm happy enough although the test could've gone better.

**28/2/2017**

Sticking to my pledge from earlier this month and I spent a couple of hours refactoring code. A lot of the files now look much cleaner!

**February summary:**

- Refactoring code so the project is cleaner.
- Technology section set up and ready to be developed further.

**March**

**10/3/2017**

I had a meeting with Lisa today. She's happy with my progress so far but she had some good feedback on some terminology and design aspects that I agree with. We did a think aloud and I'll be implementing the results of it as soon as I can. They were mostly UX things, which isn't my strong point so I'm delighted with the feedback.

**15/3/2017**

I've been added some unit tests to my project. There's 10% going for testing so I'm hoping to get as good as possible a grade for this section. My QA Engineer experience should come in handy here; I might even get some UI tests done as well.

**23/3/2017**

I've only started to add more to my project over the past couple of days. College has been absolutely hectic and I'm finding less time to journal, let alone add to the project. Hoping to add some more!

**March summary:**

- Did a Think Aloud with Lisa
- Added more tests to my project