**A tool for supporting "Attendance and Engagement" monitoring procedures**

**Ryan Trainer**

**April 2018**

**Dissertation submitted in partial fulfilment for the degree of   
Bachelor of Science with Honours in Business Computing**

**Division of Computing Science and Mathematics  
University of Stirling**

Abstract

Summary of the dissertation ***within one page***. Unnumbered chapter headings, as above, are entered using the *Unnumbered 1* paragraph style. The *Unnumbered 1* style automatically starts a new page.

This template starts the page numbering at the foot of this page. While you are printing drafts, you might find it useful to add the printing date and time into the footer – to help you, and your supervisor, tell which version is most current.

**Note: You are required to submit one extra copy of your title page and Abstract.**

It is suggested that the abstract be structured as follows:

* Problem: What you tackled, and why this needed a solution
* Objectives: What you set out to achieve, and how this addressed the problem
* Methodology: How you went about solving the problem
* Achievements: What you managed to achieve, and how far it meets your objectives.

Attestation

I understand the nature of plagiarism, and I am aware of the University’s policy on this.

I certify that this dissertation reports original work by me during my University project except for the following (*adjust according to the circumstances*):

* The technology review in Section 2.5 was largely taken from [17].
* The code discussed in Section 3.1 was created by Acme Corporation ([*www.acme-corp.com*](http://www.acme-corp.com)*/JavaExpert*) and was used in accordance with the licence supplied.
* The code discussed in Section 3.5 was written by my supervisor.
* The code discussed in Section 4.2 was developed by me during a vacation placement with the collaborating company. In addition, this used ideas I had already developed in my own time.

**Signature** *(you must delete this, then sign and date this page)* **Date**

Acknowledgements

I would like to thank my friends and family for the support they have provided throughout the process of this Honours project. I would especially like to thank Dr Simon Jones who gave me a great deal of focus and motivation from the first line of code to the last paragraph.

Table of Contents

The table of contents below is automatically generated from the paragraphs of style *Heading N* and *Unnumbered N*. To update this after revisions, right-click in the table and choose *Update Field* for the entire table.

Abstract i

Attestation ii

Acknowledgements iii

Table of Contents iv

List of Figures vi

1 Introduction 1

1.1 Background and Context 1

1.2 Scope and Objectives 1

1.3 Achievements 2

1.4 Overview of Dissertation 3

2 State-of-The-Art 4

2.1 RollCall by PowerVista 4

2.2 GradeLink 5

2.3 Einstein by Orbund 7

2.4 Canvas 8

3 Requirements/ Technologies 10

3.1 Requirements 10

3.1.1 Keep attendance records for each student/module 10

3.1.2 Make it as easy as possible for coordinators to record attendance at sessions 10

3.1.2.1 Hierarchical task analysis 11

3.1.3 Be configurable to alert coordinators when students miss contact points 11

3.1.4 Cross check attendance records between modules 11

3.1.4.1 Hierarchical task analysis 12

3.1.5 Be able to produce readable and archivable attendance record reports 12

3.2 Technologies 12

3.2.1 HTML 12

3.2.2 CSS 13

3.2.3 JavaScript 13

3.2.4 Bootstrap 13

3.2.5 PHP 14

3.2.5.1 PHP vs Python 14

3.2.6 MYSQL 15

4 High Level Design 16

4.1 Front end design 16

4.1.1 Schneiderman’s “Eight Golden Rules of Interface Design” 16

4.1.2 Login 17

4.1.3 Navigation 17

4.1.4 Portal 18

4.1.5 Report pages 18

4.2 Database design 19

4.2.1 Jargon Breakdown 19

4.2.2 Individual tables 20

5 Implementation 25

5.1 Front-end implementation 25

5.1.1 Login page 25

5.1.2 Portal 25

5.1.3 Take attendance page 25

5.1.4 At risk students page 25

5.2 Back-end implementation 25

6 Testing 26

7 Conclusion 27

7.1 Summary 27

7.2 Evaluation 27

7.3 Future Work 27

References 28

Appendix 1 29

Appendix 2 – User guide 30

Appendix 3 – Installation guide 31

List of Figures

Similarly, you can automatically generate a list of figures from paragraphs of style *Figure*. To update this after revisions, right-click in the table and choose *Update Field* for the entire table.

[Figure 1. RollCall by PowerVista Attendance Report 5](#_Toc4162420)

[Figure 2. GradeLink Attendance Report 6](#_Toc4162421)

[Figure 3. GradeLink Attendance screen 7](#_Toc4162422)

[Figure 4. Einstein by Orbund Home screen 8](#_Toc4162423)

[Figure 5. Canvas home page 9](#_Toc4162424)

[Figure 6. Take attendance hierarchal task analysis 11](#_Toc4162425)

[Figure 7. Cross check students hierarchal task analysis 12](#_Toc4162426)

[Figure 8. Simple HTML page 13](#_Toc4162427)

[Figure 9. PHP example 14](#_Toc4162428)

[Figure 10. Navigation bar design 17](#_Toc4162429)

[Figure 11. Portal Design 18](#_Toc4162430)

[Figure 12. Report Pages Design 19](#_Toc4162431)

[Figure 13. Users table 20](#_Toc4162432)

[Figure 14. Students table 21](#_Toc4162433)

[Figure 15. Module table 21](#_Toc4162434)

[Figure 16. Class table 22](#_Toc4162435)

[Figure 17. Module enrolment table 22](#_Toc4162436)

[Figure 18. Class enrolment table 23](#_Toc4162437)

[Figure 19. At risk students table 23](#_Toc4162438)

[Figure 20. Login Page 25](#_Toc4162439)

# Introduction

The aim of this project is to develop a solution for the staff of Stirling University to monitor attendance of students and identify those who are at risk of dropping below the accepted threshold. This monitoring is a duty set by the University for staff who coordinate modules. This is particularly relevant when students are studying under an Education Visa. There can be strict rules set for attendance and engagement during the study period and failure to comply with these rules could risk their immigration status being put at risk.

## Background and Context

As previously mentioned most faculties require a log of attendance to be taken for prescribed classes. Currently there is no unified way that teaching staff in Stirling University can monitor a student’s attendance to a particular class or module. There is a variety of methods used for example, Canvas quizzes, the use of student cards to digitally log attendance, physical attendance sheets that are passed around the class or a traditional roll call setup. The format that these records are saved in are also not unified, there is Word, Excel, Textpad documents, hand written registers and scanned copies physical attendance sheets. Also, different classes have different rules for attendance and engagement. Where in one class you are given grace with one to three missed classes, others may have a zero-tolerance policy for unexplained absences. This makes attendance monitoring difficult and time consuming. As there is no uniform way that the attendance records are collated, to begin analysing the data it must be parsed into a central format by hand.

As previously mentioned, there is a second issue, that there are different rules for different classes. To put this into context CSCU9A1, a first-year computing module has a practical that students are required to attend each week. However, there are three sessions each week that students are free to attend at their convivence. However, in certain other modules there are set sessions that students must attend. So, at the moment when all class attendance records are analysed by hand the staff member must remember what particular classes were mandatory and which were optional. This can result in clerical errors and students may be allowed to be missed.

## [Scope](http://www.cs.stir.ac.uk/~kjt/research/conformed.html) and Objectives

I aim to solve this by developing a Web interface that can be used by all Teaching Staff to monitor and record Student’s attendance and engagement. They will be able to track students over several modules and the system will also alert the staff to students who are starting to fall behind. Each module will have a configurable rule for attendance so that Teaching staff can set a specific threshold for expectations. The only administration details that will fall on staff every semester is updating the enrolled students that are in the university. This will allow the system to seamlessly continue as long as the student list will be updated. This can be done by parsing a simple CSV into the database management suite, most likely PhpMyAdmin.

As with any new system or idea implemented there will be resistance to change from the desired users. Currently, different staff have their own specific way of monitoring attendance of their classes. The success of this system is adoption, the issue of no uniformity can only be solved with the adoption of a central system. To solve this the system needs to be simple enough for anyone to be able to operate successfully with limited to no coaching. This will be achieved with simple user interface design, informative feedback and easy to follow navigation.

## Achievements

The objectives outlined in section 1.2 have been achieved. The system enables full monitoring of attendance records for any module that is live on the database. In the future there would need to be integration into the University student records to enable any module or student to be analysed with the tools available. Currently the system works on a local database level with any modules and students being required to be manually entered into a database. This is a proof of concept and with the correct permissions from the University could be integrated over very easily.

## Overview of Dissertation

The dissertation will analyse the current options available both within the university and those commercially available. There will be a clear description of what they are and how they operate to determine what they offer to the context of the problem at hand. The strengths and weakness will be explored to give scope to what the project can work on. The dissertation will then be broken down into chapters:

* Chapter 3- Requirements/ Technologies
  + A breakdown of exactly what the project must be able to accomplish and who will be using the final system.
  + A brief overview of the technologies that will be used throughout the project and what part they will play in the final design
* Chapter 4 – High Level Design
  + The design of the final system’s user interface and the stylistic choices implemented.
* Chapter 5 – Implementation
  + How the final project comes together and how the discussed technologies were used.
* Chapter 6 – Testing
  + An analysis of the testing for the system and they key takeaways from feedback.

# State-of-The-Art

There is a great deal of existing solutions available to Universities and Colleges already with most offering a vast range of features and platforms that will manage attendance. I am going to undertake a critical analysis of a selection of these solutions to understand what is already being offered and determine why my solution would be implemented over one of these. I am going to look at:

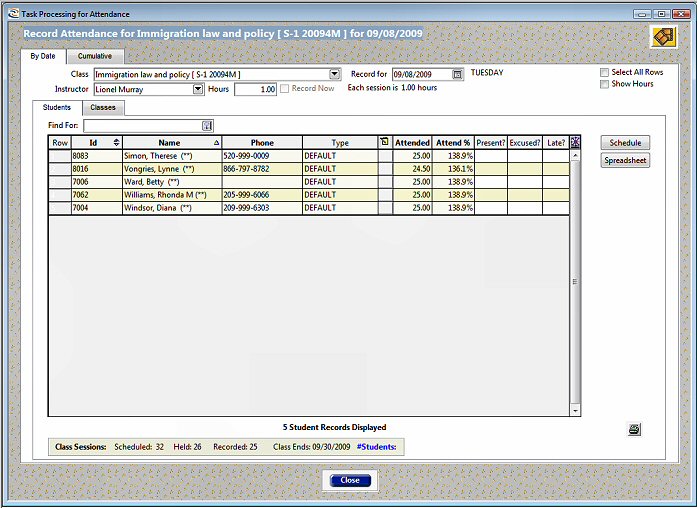
* RollCall
* GradeLink
* Einstein
* Canvas

## RollCall by PowerVista

Firstly, Rollcall by PowerVista much like many of the other solutions is a coverall solution for an institution [1].This means that the package includes everything necessary to run and operate a University. From financial management, merchandising, events, exams, syllabus management and student management. This is a lot more than the remit of the problems entails as there are already established systems in place to take control of most of these queries.

RollCall is a portal-based solution offering multiple connections to a student database which is comprised of several tables to hold the necessary information [1]. The Portal approach is very relevant to today’s standard as it is much easier to implement and maintain. A portal can be accessed from anywhere on or off the campus with an internet connection and the correct permissions. This means that lecturers do not need to be sitting at a desk to perform the administration tasks associated with managing a class. It can be done on the move. It also makes updates much easier as it is simply a case of updating the portal files without any need for downloading a new version as it is accessed dynamically every time there is a log in.

The user interface is very important especially when it will be several different users will be accessing the system throughout the day, it has to be something that suits the mass market.



1. RollCall by PowerVista Attendance Report

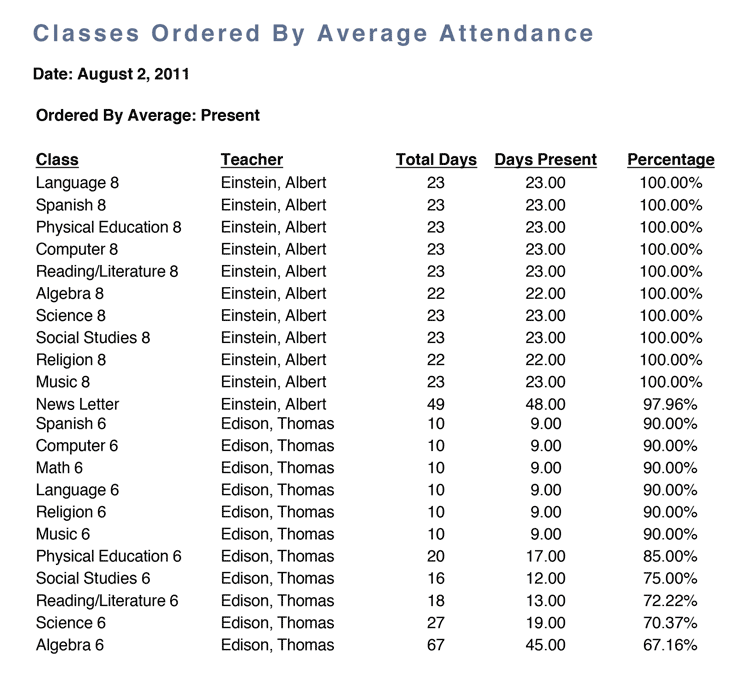
As is shown in Figure 1 the interface is dated by today’s standards, it is cluttered and hard to comprehend at first glance. The colours are very bland and un appealing to the eye. I believe this would put many users off using the system. I believe with a refresh it would bring the package up to a very high standard. By simplifying the interface and introducing some block colour I believe that this would create a much more appealing experience for the user.

Figure 1 shows that this solution does record attendance and also percentage of classes attended which is exactly what the project requires however it does nothing with this information. A large part of the problem is targeted at visa students who need to engage to meet the requirements of their residency. To make this more viable to the problem it would need to separate students who have these visas and have some form of alert system to allow lecturers to know if students are at risk of breaching their terms and conditions of stay.

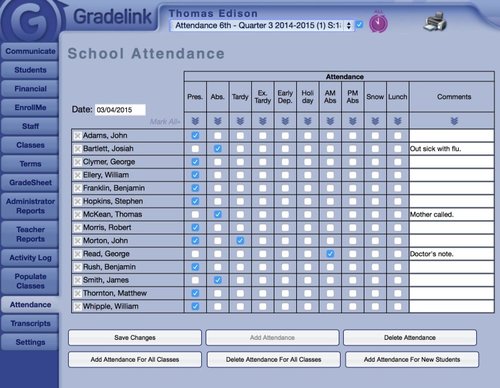
## GradeLink

GradeLink is a much more comprehensive system designed for nurseries right up to small university departments [2]. It once again offers a large array of features similar to that of RollCall and options that can help run an institution. Much like RollCall these features fall outside the scope of the problem at hand.

GradeLink makes use of desktop software to deliver their solution [2]. This is due to the sheer amount of options available to the user, this simply would not be efficient to have in a web- based context. This will slow initial set up due to installations and updates will be harder to manage and roll out. However, it is a small price to pay for the number of features that are available to the end user.

GradeLink’s most stand out feature is the reporting capabilities that is offers. It can generate full attendance reports for given students or classes with full breakdowns available with configurable filters as shown in Figure 2. This would be very useful for those students here on education visas as you could very quickly generate a report and filter the students down to see if any are slipping behind. However, it still does not offer the automatic function which means that lecturers would need to regularly generate the reports themselves to make sure no students are falling behind.

1. GradeLink Attendance Report



1. GradeLink Attendance screen

The User Interface is more visually appealing than that of RollCall, but it is still slightly dated. As Figure 3 shows this is a move in the right direction in terms of design as it is much clearer and has a better colour scheme. The only feature I would set out to change is the amount of options that is listed on the left-hand side of the screen. I would opt for a simpler scheme, grouped into sections that drop down when selected so the user can quickly find what they need as supposed to searching through all the options available.

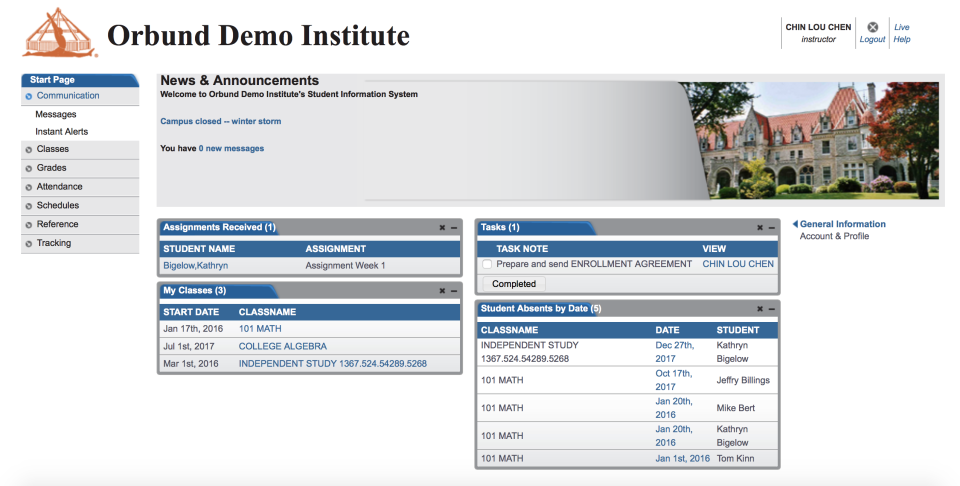
Overall GradeLink is a more modern example of what is available on the market today, it has excellent reporting capabilities and with some light customisation this could slot well into the use case of visa students. There is so many features that are outside of scope that I feel that the cost and time to install the software is not worth it due to how little of the capabilities will actually be used.

## Einstein by Orbund

Einstein is among the more modern adaptations of this type of package. It much like GradeLink is scalable in terms of size of institution that it is implemented into [3]. Also, much like RollCall it offers a large quantity of administrative features that would be very useful although out of scope for the problem. Where Einstein differs is the branding element, with Einstein you can alter the branding to suit your institution [3].

This feature would be very useful when considering using a commercial platform. To keep consistency between all University related products being able to later the appearance of the package will aid in this process. It is by no way a necessity, but it gives some of the control back to the user that is lost when choosing to use a pre-developed solution. In terms of the proposed project this step would be negated as it is being built from the ground up. Any specific design requirements that the University would like or require can be implemented as the package moves into the implementation stage and edited as necessary.

The general user interface is something I believe is a stand out feature for this package.



1. Einstein by Orbund Home screen

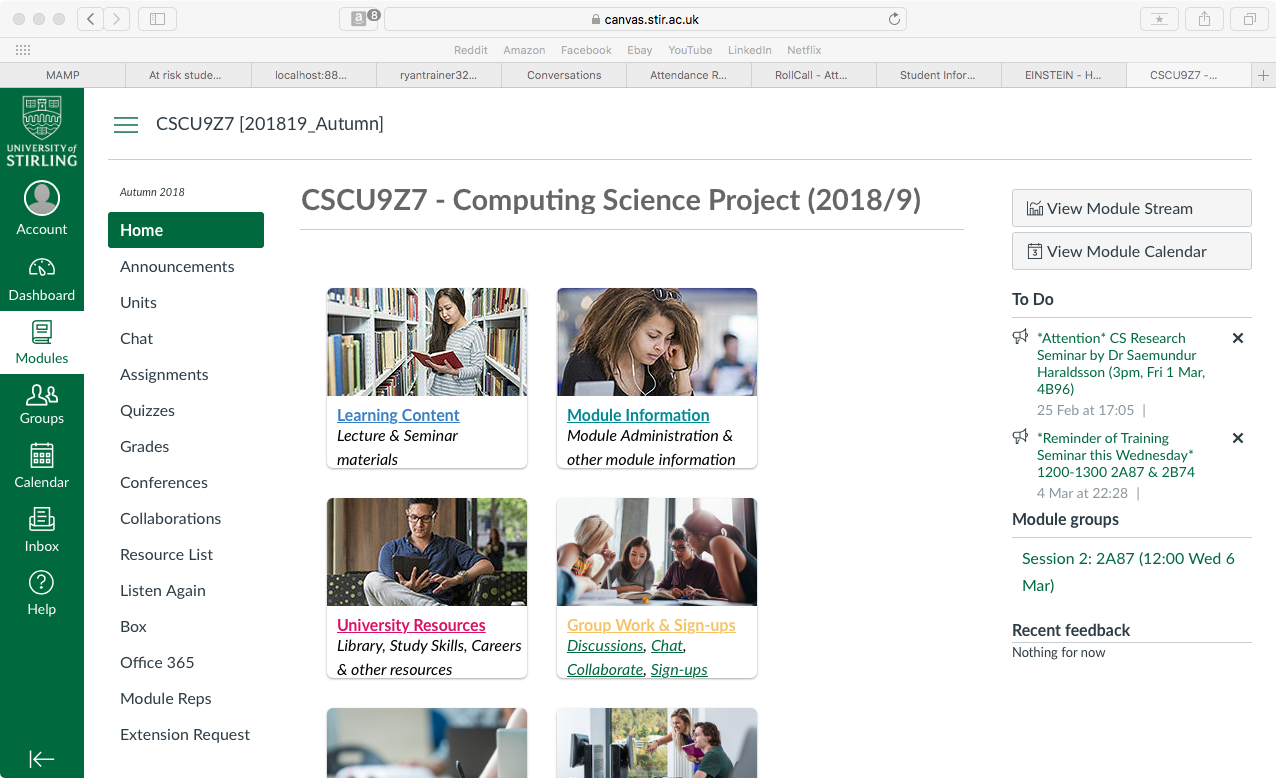
As Figure 4 shows it is clean, simple and looks professional. There is slightly more information on display than this project requires however it looks modern and very useable. Navigation is easy with the grouped options as I previously referred to with GradeLink. Figure 4 shows a demo home page that can be edited to suit your institution, colours and images can be altered [3].

Overall Einstein is the most modern package I found in my research, it offers several options that the University could use. However, this project has a very specific scope, the main focus is the uniformity which all packages would deliver but the configurable rules for each module is another main focus which none of the packages offer.

## Canvas

All analysed solutions offer a wide array of options that would be useful for any institution. However, Stirling already has services in place to handle these options with integration with Canvas for an example. The University is heavily integrated with the Canvas VLE which takes care of many of the tasks associated with these analysed packages for example marking class tests, module management and group allocation [4]. One of the main drawbacks with Canvas is the lack of attendance recording ability. There is currently no way to take a class register on Canvas and the main way at the moment [4] is to create a quiz with only one question and an answer that is given out only in the class. This is not practical and involves more admin than is needed.

The visual element to Canvas is refreshing as shown in Figure 5 when compared to the other analysed options. The colours are clear and can be configured to suit the module/institution. Navigation is simple and it is always clear where you are on the page. From a user standpoint Canvas works great for the day to day tasks. However as mentioned from an administrative point of view it does not offer the core ability to monitor and track student attendance and engagement in a module.



1. Canvas home page

# Requirements/ Technologies

This chapter will be broken down into two main sections, 3.1 will cover the main requirements as stipulated in the brief and where relevant there will be a use hierarchical task analysis to illustrate this point. Section 3.2 will cover the technologies that will be utilised to achieve this.

## Requirements

The outlined requirements below come from the CSCU9Z7 available projects page [5] specifically in reference to this particular project (“A tool for monitoring “attendance and engagement” monitoring procedures”).

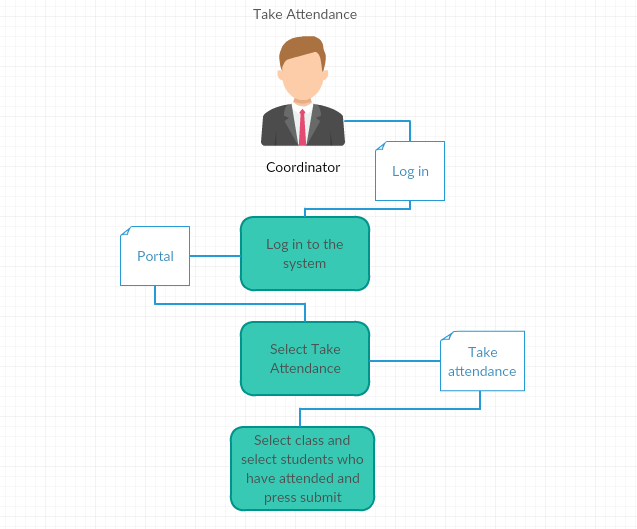
### Keep attendance records for each student/module

The ability to store the records is pivotal in monitoring attendance. This gives coordinators the ability to check attendance at any time throughout the academic year. Any recorded attendance data needs to be backed up securely in a central database to make retrievable easy at a later date.

### Make it as easy as possible for coordinators to record attendance at sessions

The physical taking of attendance is the most important factor in the project. Without attendance data there can be no analysis that can be made. The attendance data is recorded by the coordinators so making this step as easy as possible is important. This would mean as little clicks to the goal as possible.

#### Hierarchical task analysis



1. Take attendance hierarchal task analysis

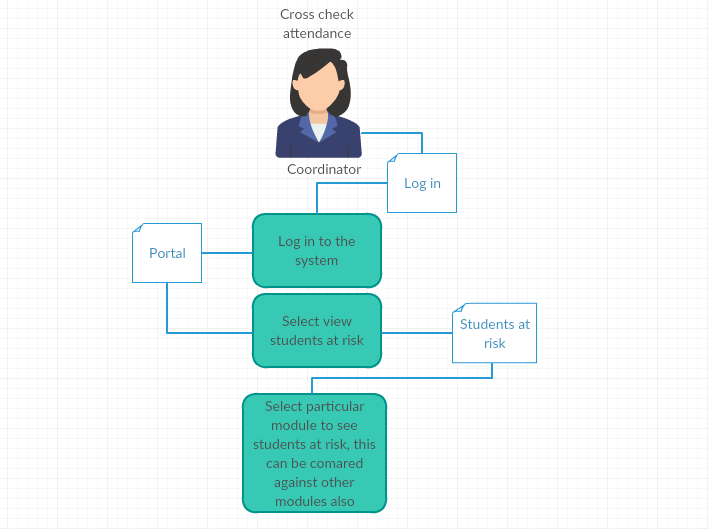
### Be configurable to alert coordinators when students miss contact points

The configuration element to the project covers the points made about modules having different rules for engagement as previously mentioned in section 1.2. This would mean involve the coordinator being able to stipulate the rules for engagement for each module to produce viable data for at risk students.

### Cross check attendance records between modules

Students can be undertaking a number of different modules at any one time. This means that they will have multiple sets of attendance records between all enrolled modules. The ability to be able to retrieve this data is important when trying to recognise students who are in need of assistance. It would also help with the identification of particular modules that are causing attendance issues for students.

#### Hierarchical task analysis



1. Cross check students hierarchal task analysis

### Be able to produce readable and archivable attendance record reports

The report function is important for record keeping as it can show clear patterns of attendance overtime. It is also key in disciplinary standards as it would enable one party to present tangible evidence to the party who has not been attending.

## Technologies

There were a number of technologies that were utilised in the final version of the project as it is web-based HTML, CSS and JavaScript were utilised to create the base pages for the user to interact with. PHP and MYSQL were used to handle the server-side scripting and databases element to the project.

### HTML

HTML forms the base of the web pages. It is universally accepted and is considered standard for building web pages [6]. To create a web page tags are used and give context to what will be enclosed in them, for example <body> and </body> represents anything that will be displayed within the moan body of the web page. Figure 8 below demonstrates this, all information is enclosed within the respective tags showing clearly what will be displayed. This form of code will display a very simplistic web page with no stylistic elements to it. Generally, HTML requires CSS to transform it into the web pages we are accustomed to. HTML forms the structure for the project and allows a platform for the scripts that will perform the functions required to be ran.



1. Simple HTML page

### CSS

CSS provides the stylistic elements that were mentioned in section 3.2.1, it allows for colour, font and layout changes to a HTML page [7]. CSS can take a very simple and hard to read web page and easily change how it is delivered to the end user. This is performed by allocating elements in the HTML “IDs” that can be called on inside a CSS file. This gives a target for the style injection.

### JavaScript

JavaScript is a scripting language commonly used in conjunction within HTML to allow client-side scripts to be performed [8]. There is a number of different libraries available for JavaScript to utilise depending on the use case. This project used core JavaScript and AJAX library. Ajax allows data to be read from a web server while the page is loaded and perform an update without the need for a reload [10]. This is particularly important in this project as there could be a user who is checking the status of multiple students and refreshing every time would cause unnecessary inconvenience and resource usage. So, the ability to limit the number of refreshes in an action is very important.

### Bootstrap

Bootstrap is a CSS/JavaScript framework that is popular amongst web developers for the flexibility and ability to integrate well into mobile environments. Bootstrap is essentially a large library of pre-programmed CSS and JavaScript that developers can take advantage of [10]. By providing “containers” which act through tags that are paced inside of the HTML code a user can dictate the structure of anything within this tag. This is used in the project to ensure elements are scalable when a window is resized for increased usability and to increase the stylistic appeal to the site.

### PHP

PHP is an open source scripting language heavily used in web development. It can be embedded into HTML [11] making it an appropriate technology to be used in this project. Any code written in PHP is placed inside tags within a HTML file. This enables the code to be contained within separate sections of the page. Figure 9 gives a simple but clear explanation of this. As shown below there is a simple HTML page with the relevant tags and the PHP script is encapsulated within independent tags inside the HTML body.



1. PHP example

Any code within these PHP tags are executed server side. The script returns the result to the user not the underlying code. This is great from a security point of view because the actual execution of the script takes place on the server as opposed to client side. This leaves the script less exposed to interception [11].

PHP is supported over all major operating systems [11]. This allows for flexibility in terms of implementation. As the project will be running on multiple different platforms this is key.

#### PHP vs Python

Python was a considered choice with regards to the server-side scripting that would be necessary for this project. While Python is not inheritably a server-side scripting language is it very versatile and with the correct frameworks can operate server-side scripting very efficiently [12]. A very common argument for the use of Python over PHP is the ease of access, Python is regarded as easier to learn than PHP. However, I had more exposure to PHP and thus had a strong confidence with the language.

PHP in later versions has also experiences significant speed increases to Python based frameworks [13]. This is critical when the project is query based and delay would only tarnish the overall experience if it there was significant delays between button presses.

Python does have the advantage when it comes to library availability. There are extensive libraries available for an array of different tasks[13]. PHP still has a great deal available to the user however they are not as extensive as the ones available in Python. This is no cause for concern with this particular project. The PHP used in the project will not need to utilise external libraries to function, it will be mainly query based and interpreting user input. This is something that PHP can handle very well, and it ultimately made for.

Overall Python is more popular in the community for its ease of access and extensive library input however for this project PHP is more appropriate. PHP is tailor made to handle this type of problem and because of that it can perform very effectively within the set tasks. Python can do much more, but this is far outside the scope of the project and would be dramatically underutilised for the performance drop.

### MYSQL

MYSQL is a database management system that utilises SQL queries to interact with databases and webservers [14]. These queries can change elements in a database table. MYSQL operates in conjunction with PHP. In this project the PHP ultimately gains the user inputted information (attendance status, class selection etc) and MYSQL forms the query that will allow this information to interact with the database. This could be updating a status or retrieving the students who belong to the respective module.

# High Level Design

The high-level design of this project consists of a number of core components:

* Front end design
* Database design

## Front end design

The front-end design is what they user will be interacting with to control the system. As such it is important that it is designed well. Before the front end is broken down into individual elements there is a Schneiderman’s analyses of the “eight golden rules of interface design” [15] that has been undertaken for the project.

### Schneiderman’s “Eight Golden Rules of Interface Design”

1. Strive for consistency:

* There should be a consistent colour scheme across all pages.
* Fonts and headings remain the same style across all pages.

2. Enable frequent users to use shortcuts:

* The navigation bar across all pages should allow users to navigate to any part of the site at will.

3. Offer informative feedback:

* When the user updates an attendance record there should be some form of dialog to inform them of the change.
* The navigation bar should reflect what page the user is on.

4. Design dialog to yield closure:

* To take or check attendance there should be a uniform way of selecting the module and class.

5. Offer simple error handling:

* The use of dynamic dropdowns limits the amount of user input so decreases the amount of possible errors.
* Limit the amount of user entered data and make use of selection techniques (check boxes etc)

6. Permit easy reversal of actions:

* Pages can be re-accessed at will and the attendance can be taken as many times as the user desires.

7. Support internal locus of control:

* The use of buttons to submit attendance records and drop downs to select class/modules gives the user more control.

8. Reduce short-term memory load:

* Elements within the web page are simple and do not require long load times.
* The use of AJAX technology allows a number of processes to be done within the same page without the need for a full refresh.

### Login

There will need to be the element of security with regards to the access to the project. As there is access to sensitive student data and records there needs to be a login page that would only give access to those who are authorised to access said information.

### Navigation

There will be a consistent navigation bar across the project. This satisfies rule one and two of Schneiderman’s eight golden rules [15]. This is important for the user as it allows them to stay grounded within the project and at all times can get back to a page that they need to access.

Home

Take attendance

Show at risk students

Create rule

Logout

1. Navigation bar design

Figure 10 above shows a very simple mock up design of the consistent navigation bar. This will be present across all pages and will allow the user to navigate to any of the major pages at will. This is important because it gives users control of their experience while using the project [16]. Whatever page the user is currently on will be highlighted with the use of highlighting the name of the page on the navigation bar. This is important for the user to be able to see where they are at any one time without the aid of page clues [16].

### Portal

When the user logs into the system they will be greeted with a portal screen. This will act as a landing page. The logic behind this is due to the sensitive nature of the information that the project provides there needs to be a buffer between this and the human eye. The portal would be a safe page to log into as on start-up there is no class or student information on display. This means that no students or staff who happen to be in eyeline are able to see any sensitive information.

Take attendance

Create Rule

Show students at risk

Hi User

1. Portal Design

Figure 11 above shows a mock-up of what the portal will look like. It gives access to all the major elements of the project but also provides a safe landing space for use in classrooms.

### Report pages

The report pages (Take attendance and Show students at risk) will hold a similar format with drop down selections at the top of the page and a table that will be populated with the information. This will allow the user to select the relevant module and class then view the relevant data for the page. By keeping these pages similar in layout then it means that it gives the user less of a learning curve and will breed familiarity with the project.

1. Report Pages Design

Navigation Bar

Module Dropdown

Attendance status

Class Dropdown

## Database design

The use of a database is critical to the operation of this project. It allows all the attendance information to be store stored from the various classes. It also allows records to be recalled for later analyses or clarification. The project used phpMyAdmin to perform any database management that was required. This database management suite is used throughout the University so there was a level of comfort with the general operation of the suite. phpMyAdmin allows the creation and manipulation of the operational databases [17]. It also allows a user to parse a csv document into the database which will populate it with the current students and classes for the given year. This would be the only element of user maintenance that a user would need to perform. As long as it is formatted in the respective fashion for the database tables.

### Jargon Breakdown

With regards to database design there is some key phrases that will be used heavily throughout the explanation. This sub section will very briefly introduce them for context.

1. Table –A table consists of rows each with columns, but column values vary. [18].
2. Primary key – A column or group of columns in a given table that uniquely identifies each row of the table [18].
3. Foreign key - One or more columns in a table that contains that values match the related tables’ primary key. This acts as a link between the two tables in the form of a column value [18].
4. Referential Integrity - A condition in which all foreign key values in a given table has a corresponding primary key value in another table. This is a key principle of database design and without it the system is open for malicious changes that would affect the entire project. [18].

### Individual tables

Below are the tables used in the project with fictitious entries for context. Each heading will be explained to give an insight into the logic of the database design.

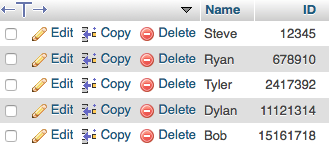
The entries below are a work of fiction. Names, ids and classes are either the products of the imagination or used in a fictitious manner. Any resemblance to actual persons, living or dead, or actual events is purely coincidental.



1. Users table

Figure 13 above shows the table that houses the user information for the project. As the project should only be accessed by authorised personnel there needs to be security in place.

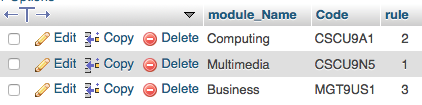
1. ID – A unique ID that is applied to each individual user. This number auto-increments as a new user is added and acts as the primary key. this is useful in database admin and is common practice to give entries with no necessary unique field to have an ID.
2. Username – A name that is chosen by each user to act as part of their login process.
3. Email – An email entered by the user that can be used in password reset process.
4. Password – A password set by the user that is hashed for increased security and allows the user to login when combined with their username.



1. Students table

Figure 14 above stores the current enrolment of students who will be monitored using the project.

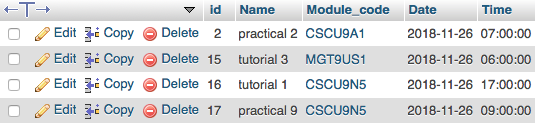
1. Name – Simply the name of the student for ease of attendance taking.
2. ID – The unique ID related to that student. This column entry must be unique and acts as the primary key.



1. Module table

Figure 15 above shows the tables that holds the current modules that will be used within the project.

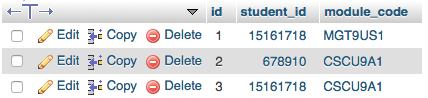
1. Module\_name – This simply acts as a display name for the module to give context to the user.
2. Code – This field is the unique code that each module is given by the University. It must be unique and also acts as the primary key.



1. Class table

Figure 16 above shows the table that holds the individual classes and the modules they are associated with.

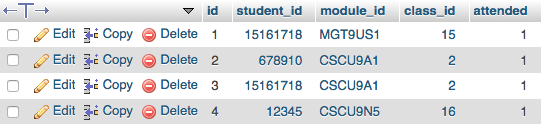
1. ID - A unique ID that is applied to each individual class. This number auto-increments as a new user is added and acts as the primary key. This is useful in database admin and is common practice to give entries with no necessary unique field to have an ID.
2. Name – A display name for the class.
3. Module\_code – The individual module code that the class is a part of. This acts as a foreign key referencing the module code field in Figure 15.
4. Date – When the class takes place in reference to the day.
5. Time - When the class takes place in reference to the time.



1. Module enrolment table

Figure 17 above shows the students and what individual modules they are enrolled in. This is useful for gaining full class lists of a particular module.

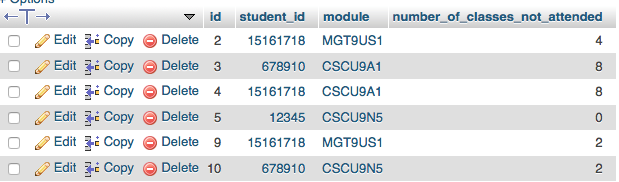
1. ID – A unique ID that is applied to each individual combination. This number auto-increments as a new user is added and acts as the primary key. This is useful in database admin and is common practice to give entries with no necessary unique field to have an ID.
2. Student\_ID – The individual code for the student. This acts as a foreign key referencing the student code field in Figure 14.
3. Module\_code – The individual code for the class. This acts as a foreign key referencing the student code field in Figure 14.



1. Class enrolment table

Figure 18 above shows the enrolment status of the students with regards to specific classes.

1. ID - A unique ID that is applied to each individual combination. This number auto-increments as a new user is added and acts as the primary key. This is useful in database admin and is common practice to give entries with no necessary unique field to have an ID.
2. Student\_ID – The individual code for the student. This acts as a foreign key referencing the student code field in Figure 14.
3. Module\_ID – The individual module code that the class is a part of. This acts as a foreign key referencing the module code field in Figure 15.
4. Class\_ID – The unique ID for the class. This acts as a foreign key referencing the ID field in Figure 16.
5. Attended – This field holds the attendance status of each student to each class they are enrolled in. This is a Boolean field with 1 representing attendance and 0 not attended. This allows simple collation which to determine attendance status.



1. At risk students table

Figure 19 above represents the collated number of classes not attended by each student for each module. This is critical in identifying students who are at risk of slipping in terms of attendance status.

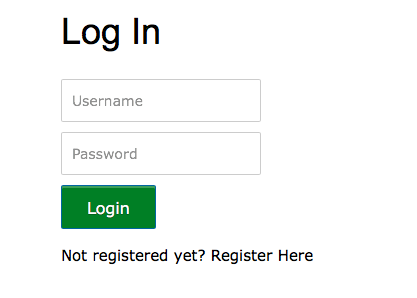
1. ID – A unique ID that is applied to each individual combination. This number auto-increments as a new user is added and acts as the primary key. This is useful in database admin and is common practice to give entries with no necessary unique field to have an ID.
2. Student\_ID – The individual code for the student. This acts as a foreign key referencing the student code field in Figure 14.
3. Module – The individual module code that the class is a part of. This acts as a foreign key referencing the module code field in Figure 15.
4. Number\_of\_classes\_not\_attended – The collated total of classes not attended by each student.

# Implementation

## Front-end implementation

The front-end implementation concerns everything that the user will interact with. This subsection will break down all these elements and compare them to the original design.

### Login page



1. Login Page

### Portal

### Take attendance page

### At risk students page

## Back-end implementation

# Testing

# Conclusion

## Summary

Summarise what you have achieved.

## Evaluation

Stand back and evaluate what you have achieved and how well you have met the objectives. Evaluate your achievements against your objectives in section 0. Demonstrate that you have tackled the project in a professional manner.

(The previous paragraph demonstrates the use of automatic cross-references: The “0” is a *Cross-reference* to the text in a numbered item of the document, it is *not* literal text but a *field.* The number that appears here will change automatically if the number on the referred-to section is altered, for example if a chapter or section is added or deleted before it. Cross-references are entered using Word's **Insert** menu. Cross-references are set to update automatically when printed but may not do so on-screen beforehand; you can update a field manually on-screen by right-clicking on it and selecting Update field from the pop-up menu.)

## Future Work

Explain any limitations in your results and how things might be improved. Discuss how your work might be developed further. Reflect on your results in isolation and in relation to what others have achieved in the same field. This self-analysis is particularly important. You should give a critical evaluation of what went well, and what might be improved.

References

1. RollCall - School Administration Software. Available: [http://www.powervista.com](http://www.powervista.com/)
2. GradeLink - School Success Software. Available: <https://www.gradelink.com>
3. Einstein College management system. Available: <http://www.orbund.com/einstein-college-management-system.html>
4. Canvas. Available: <https://www.canvasvle.co.uk>.
5. CSCU9Z7 - Available projects Available: <http://www.cs.stir.ac.uk/courses/CSCU9Z7/2018-19/available-projects.php>
6. What is HTML? Available: <https://www.w3schools.com/whatis/whatis_html.asp>
7. What is CSS? Available: <https://www.w3schools.com/css/css_intro.asp>
8. What is JavaScript? Available: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript>
9. What is Bootstrap? Available: https://www.w3schools.com/whatis/whatis\_bootstrap.asp
10. What is Ajax? Available: https://www.w3schools.com/whatis/whatis\_ajax.asp
11. What can PHP do? Available: <http://php.net/manual/en/intro-whatcando.php>
12. What is Python? Available: <https://www.python.org/doc/essays/blurb/>
13. Python vs PHP in 2019. Available: <https://hackr.io/blog/python-vs-php-in-2019>
14. MYSQL. Available: <https://searchoracle.techtarget.com/definition/MySQL>
15. Schneiderman’s Golden Rules. Available: <https://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html>
16. Implement global navigation to improve website usability. Available: <https://www.interaction-design.org/literature/article/implement-global-navigation-to-improve-website-usability>
17. About phpMyAdmin Available: <https://www.phpmyadmin.net>
18. Database Terminology Available: <https://raima.com/database-terminology/>

Appendix 1

You may have one or more appendices containing detail, bulky or reference material that is relevant though supplementary to the main text: perhaps additional specifications, tables or diagrams that would distract the reader if placed in the main part of the dissertation. Make sure that you place appropriate cross-references in the main text to direct the reader to the relevant appendices.

*Note that you must* ***not*** *include your program listings as an appendix or appendices*. You should submit such material to the project *digital repository*.

Appendix 2 – User guide

If you produced software that is intended for others to use, or that others may wish to extend/improve, then it is advisable to include user guide and installation guide appendices.

Appendix 3 – Installation guide

If you produced software that is intended for others to use, or that others may wish to extend/improve, then it is advisable to include user guide and installation guide appendices.