

Curriculum Mapping Database



Sarah Loughran

Supervisor: Helen R Phillips
Moderator: Stefano Zappala

Module Title: One Semester Individual Project
Module Number: CM3203
Module Credits: 40

School of Computer Science and Informatics

Abstract

Cardiff University School of Computer Science and Informatics currently partake in two yearly review processes which occur manually and therefore are inefficient and laborious. The aim of this project is to create a system for the School that automates this process and provides an easily accessible interface to enable a cross-curricular view for all staff members to view module information as well as related Accrediting Body Requirements (from the BCS: The Chartered Institute of IT), assisting in the curriculum mapping process.

This paper explores why the module review and curriculum mapping processes are so vital in establishing constructive alignment between module and learning outcomes to enable the School to offer the best level of education possible to students. This paper also explains the approach taken, using the agile methodology to justify the design and implementation of a SQL Server Integration Services project that integrates with a web front-end interface to provide a dynamic and fully automated solution.

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance. I would firstly like to thank my supervisor Helen Phillips, whose guidance and insightful feedback has been invaluable throughout.

A particular thanks to my parents for their love and support throughout my degree, I am indebted to you both. Also, a big thank you to all my friends and family who have kept me sane during what has been a period of hard work!

A special thanks to my partner Alex, for constantly motivating and inspiring me to work harder than I thought possible and for providing me with endless cups of coffee for the duration of this project. And to Mary Thomas, for taking time out of her gardening to proofread my every word.

Table of Contents

TABLE OF FIGURES	5
INTRODUCTION	6
WHAT IS CURRICULUM MAPPING?	6
THE CURRENT PROCESS	7
BACKGROUND	9
PROJECT AIMS AND OBJECTIVES	11
APPROACH	12
SYSTEM SPECIFICATION.....	15
SYSTEM REQUIREMENTS	15
USER STORIES.....	15
NON-FUNCTIONAL REQUIREMENTS.....	21
DESIGN	22
IMPLEMENTATION	26
UNFORESEEN ISSUES.....	33
RESULTS	34
TEST CASES	34
<i>Functional Requirements</i>	35
<i>Non-Functional Requirements</i>	37
USER TESTING	38
EVALUATION.....	41
HEURISTIC EVALUATION	41
SYSTEM EVALUATION.....	43
FUTURE WORK.....	44
CONCLUSIONS	45
REFLECTION	46
APPENDICES	48
REFERENCES	50

Table of Figures

FIGURE 1: ASSESSING STUDENTS' LEARNING OUTCOMES. (BIGGS, J.B., 2003).	7
FIGURE 2: VIEW OF COMPAS WEBSITE HOME SCREEN (KIDD, 2010)	10
FIGURE 3: THE SCRUM LIFECYCLE (BOER, 2014)	12
FIGURE 4: A SIMPLE ERD OF MY PROJECT APPROACH	13
FIGURE 5: FINAL DATA MODELLING DESIGN FOR MYSQL DATABASE	22
FIGURE 6: PROJECT CODE EXTRACT	27
FIGURE 7: PROJECT CODE EXTRACT	28
FIGURE 8: PROJECT CODE EXTRACT	28
FIGURE 9: PROJECT CODE EXTRACT	29
FIGURE 10: PROJECT CODE EXTRACT	30
FIGURE 11: VISUAL STUDIO SCREENSHOT OF 'EXPORT DATA TO SQL SERVER' DATA FLOW	30
FIGURE 12: PROJECT CODE EXTRACT	32
FIGURE 13: PROJECT CODE EXTRACT	32
FIGURE 14: PROJECT CODE EXTRACT	32

Introduction

The following is a project I am currently developing for a client who is the Assessment and Feedback Lead in the School of Computer Science and Informatics. I have been asked to help develop a system that assists my client in the module review process as a mechanism to enable the easier view of curriculum planning and cross-modular review, as well as tracking changes at a degree programme level and administration of criteria mapping at professional body accreditation level. In this introduction I will take you through what curriculum mapping is along with the two current processes in use by the School of Computer Science and Informatics. These current processes will be compared with similar processes that already exist whilst justifying why this problem is one that requires the solution I am providing.

What is curriculum mapping?

Curriculum Mapping is defined as the process “of indexing or diagramming a curriculum to identify and address academic gaps, redundancies and misalignments for the purposes of improving the overall coherence of a course of study and, by extension, it’s effectiveness” (The Glossary of Education Reform, 2014).

Why is it needed?

Every year, the School of Computer Science and Informatics at Cardiff University engage in a module review. Recently the University have encouraged staff towards considering review of degree programmes as a whole by year, rather than by module. This allows the school to look holistically at the current design. This is to mirror for lecturers the fact that students are able to view all modules, while lecturers can just view their module. Therefore, this would enable lecturers to have a cross-modular view to compare assessments. This would also enable the Assessment and Feedback Lead, my client - along with the head of year to get an idea of the type of assessments that are being handed out and what curricula is covered during the year cross-modularly.

There is currently no software in the School of Computer Science and Informatics that carries out this process, which is why I am proposing this idea which would enable a cross-modular view. This would also potentially fill the gap that the client feels is present in facilitating a whole review across year group level. Enabling both the client and the year head for that group to look at where there have been a number of small tweaks to individual modules which through a ripple effect could cause quite a considerable change to a year group.

The Assessment and Feedback lead also carries out a separate review every four years. This is mainly the outline description and resultant learning outcomes that are required to be met to ensure accreditation by the BCS: The Chartered Institute for IT – the professional governing body for IT related degrees at higher education. Currently there is no software process that does this for the school, my project aims to create a mechanism for all methods of review within the school.

Accreditation for degrees offered by the School of Computer Science and Informatics must meet specific accreditation criteria, highlighted in the Accreditation Guidelines (BCS, The Chartered Institute for IT, 2020). This accreditation criteria is essential to “establish and maintain standards of competence, conduct and ethical practice for information systems professionals” (BCS, The Chartered Institute for IT, 2020).

The Current Process

The School of Computer Science and Informatics currently partakes in two review processes. The first is an annual review carried out by the client, as Assessment and Feedback Lead within the school. This is to allow the administrative side of a modular review ensuring that individual modules cover essential topics and ensure that assessments are currently matching intended learning outcomes. This is similar to the concept of constructive alignment in education. This is a holistic educational idea in which it is “key that components in the teaching system, especially teaching methods and the assessment tasks are aligned with the learning activities assumed in the intended outcomes” (Biggs, J.B, 2003).

However, Biggs (2003), also states that misalignment of assessments within the curriculum can do a lot of damage for education. This concept is further supported by the idea that “an assessment is the curriculum, as far as the students are concerned” as most students will learn what they think they will be assessed on - not what is stated in the curriculum (Ramsden, P., 1992). This can be further explained that if the curriculum is reflected in the assessment, both the teaching and learning outcomes are directed to the same goal. This is preparing students for assessments within the modules and therefore students will be learning the curriculum, as shown in Figure 1 below.

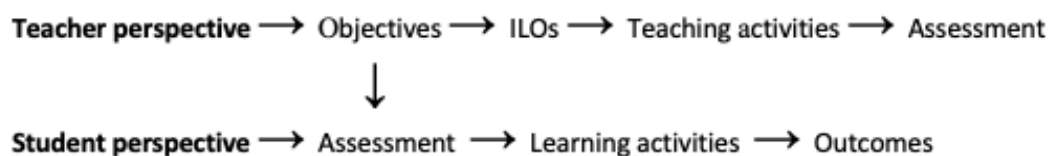


Figure 1: *Assessing students' learning outcomes.* (Biggs, J.B., 2003).

This review is carried out currently as part of a long-winded and laborious manual process which involves the creation of review word document for each specific module by that module leader. When making any changes to modules, the Assessment and Feedback Lead as well as Directors of Teaching must abide by university rules. These state that if there is a “significant change to 1/12th of the programme at a year level” the staff need university permission to carry out these changes – this permission is normally sought through an approval panel. This panel evaluates the changes and may also ask the question on whether these changes allow the year and degree to still meet accrediting body requirements. If the university grants permission for the significant changes the school is then required to inform the accrediting body, in this case the BCS: The Chartered Institute for IT. This is due to the fact that significant changes in modules can have a huge effect on the degree and the students.

The second review process happens every four years and is for the BCS: The Chartered Institute for IT. This review occurs to ensure that the school are ensuring their degree programmes are meeting the Accreditation Criteria highlighted in the Accreditation Guidelines. These review documents are developed to align with an excel spreadsheet of accreditation criteria distinct to each particular degree level e.g. undergraduate, postgraduate taught and postgraduate research. Currently the alignments between these two documents are compared manually every four years which is an arduous process; especially when the myriad of modules offered and inconsistency within the word documents created by different lecturers is taken into account. This inconsistency could be in spelling variations or deviation from the template layout that had been provided.

As such, the module review process has always been seen as an area of struggle for the key stakeholder who acts as lead module reviewer for Cardiff University School of Computer Science and Informatics. One of the main reasons for this is the fact the BCS: The Chartered Institute for IT accreditation criteria is worded very precisely and often this does not correlate with the wording used in the review assessment documents produced by module leaders. The key stakeholder of the project has also explained that this issue has been a challenge to try and resolve as major reviews only happen every four years – not occurring often enough to warrant a big change but nevertheless causing problems and time-consuming issues on an annual basis.

Background

Curriculum mapping is generally regarded as an electronic repository of a curriculum (Mattern, W. D., et al., 1992), based on the work of Dr Heide Hayes Jacobs' specifically "Curriculum 21: Essential Education for a Changing World" (Jacobs', 2010). The School of Computer Science and Informatics will only be granted accreditation for the BCS: The Chartered Institute of IT if the school meets the criteria as referenced in Accreditation Guidelines (British Computer Society, 2020). This criterion is made to "establish and maintain standards of competence, conduct and ethical practice for information systems professionals" (British Computer Society, 2020).

During research I was able to observe that curriculum mapping to professional bodies currently transpires at intraorganizational levels mainly within the health professions of various universities in America and Canada including University of Calgary and University of California, Los Angeles. This allowed me to ascertain what products were already on the market and in use.

The main focus thus far in what has been developed was specifically focused on the health professions such as medicine and dentistry in higher education. It is believed that this could be due to the increased rigidity of medical accreditation versus non-medical accreditations, as evidenced by a conceptual model that suggests that accreditation drives medical schools to implement and strengthen processes that support quality in medical education (Bloudin, et al., 2018).

Applications designed to date include CurrMIT, a curriculum management and information tool designed by the Association of American Medical Colleges which is described as being designed to accommodate data from virtually any medical curriculum throughout America and Canada (Salas, et al., 2003). This is a relational database that provides what could be compared to a digital library, for documentation and management of information related to curricula for medical schools including learning outcome objectives, similar to this project.

Another platform created was LiftUpp, developed by the University of Liverpool as a platform to map their School of Dentistry. Similarly, to CurrMIT, it was developed to support quality assured assessment and feedback however it also developed this idea further including a mapping aspect – this is regarded as a cutting-edge approach to Artificial Intelligence in Education (AIED) and is an aspect this project aims to accomplish. This app has recently been acquired by the US exam software company, ExamSoft (University of Liverpool, 2019). This shows that the concept behind this project is currently commercially profitable and in demand. It is evident that developing upon these pre-existing ideas whilst expanding beyond the field of the medical and dental health professions to encompass other degrees with accrediting bodies e.g. Engineering, Law or in this case Computer Science will

Other platforms that exist with similar concepts such as Elentra Consortium ME or Curriculum 21, the former which is used to manage and deliver curricular information

and the latter of which provides electronic tools to aid implementation of curriculum mapping. These platforms therefore are seen as a basis from which my project can develop further.

During my first project meeting with my supervisor and the key stakeholder in this project we discussed the concept of Constructive Alignment. This outcomes-based approach is based on the ideas of John Biggs - where teachers define the outcomes they intend for students to learn, then align teaching and assessments to these outcomes (Biggs, 2014). As a base for this, Biggs insists on the use of a verb during the creation of intended learning outcomes (ILOs). This verb then becomes the common link that establishes alignment between the ILOs, teaching/learning activities and assessment tasks. In turn, a comprehensive review carried out by McMahon and Thakore (2007), found that with exposure of constructively aligned teaching, students' ratings for "appropriate assessments" and "clear goals and standards" progressively increased (Biggs, 2014).

In this first meeting the key stakeholder also directed me to a curriculum mapping website produced by BARTS School of Medicine and Dentistry in the UK whom developed the system, COMPAS (Kidd, 2010). The key stakeholder in the project made it clear they wanted an end product with a web front-end of similar stature to COMPAS, as shown below in Figure 2. However, this project will differ as it will branch out to cover the accreditation of the BCS: the chartered institute of IT rather than the British Medical Association and also the project unlike COMPAS will not be available for public use. The web front-end will be hosted on an internal university server meaning only staff will have access.

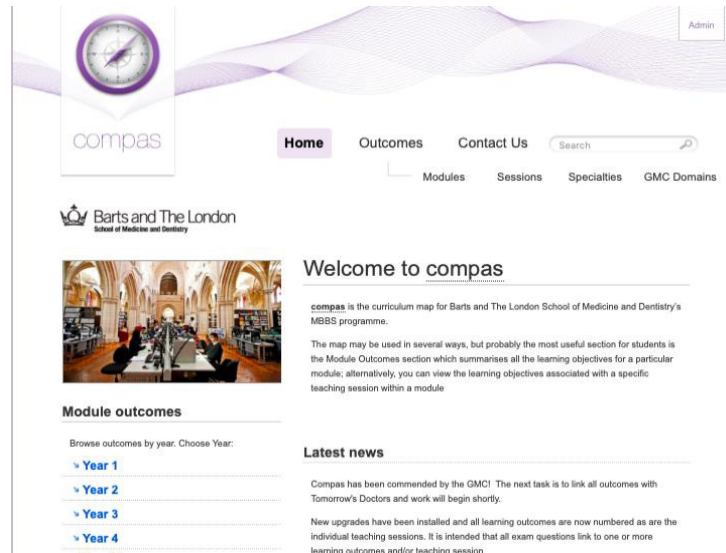


Figure 2: View of COMPAS website home screen (Kidd, 2010)

In order to design and develop a dynamic automated curriculum mapping database and web front-end, this project will identify curriculum applications and platforms currently in use as well as expanding to cover a different accrediting body, specifically the BCS: the chartered institute for IT. This project will evaluate and determine whether this mapping is possible with the layout of the current review documents and adjust the documents accordingly if a need is proven. It will also result in the creation of a user-friendly front end that will be user tested to judge the effectiveness of the solution.

Project Aims and Objectives

The aim of this project is to design a dynamic and automated curriculum mapping system that makes relationships between and maps specific module learning outcomes to specific accreditation criteria allowing for a cross-modular view of the curricula at degree and year group levels, thus overcoming the current inefficient processes associated with this long and laborious manual process.

The product aims to act as a mechanism to enable constructive alignment within the School of Computer Science and Informatics. This is a holistic educational idea in which it is “key that components in the teaching system, especially teaching methods and the assessment tasks are aligned with the learning activities assumed in the intended outcomes” (Biggs, J.B, 2003).

The current process causes many issues for enabling the centralisation of curriculum planning on a year level as module leaders can only view the modules they teach and cannot see any others. Recent staff feedback has even highlighted that lecturers believe it would be a huge benefit to be able to view what is taught in other modules – particularly when assuming a prior knowledge base for students. There is currently no software that carries out this function with the only place staff can view their module outcomes currently stored on SIMS. From staff feedback this is also very hard to access when not stored more locally. Therefore, as with this project it seems justified to have learning outcomes and assessment criteria openly available for all staff to observe and interact with on a web interface – even if they do not teach the specific modules.

This project will provide several benefits in assiduously improving the efficiency of the current process and will particularly benefit my project supervisor, as Assessment and Feedback lead and the main target user within the school. However, it will provide benefits to every module leader who will be able to easily access the website, facilitating the tracking of a particular learning outcome or module and allowing evaluation.

Another feature within this project will be the addition of an option for module leaders to edit and/or add modules during the review period within the year. This review period will aim to take into account when the university is closed, and staff are on holiday. This will in turn expedite the school’s ability to ensure all modules are being assessed accurately and meeting the correct standards as well as increasing coordination and centralization of curriculum planning and development.

Approach

The beginning of the planning and implementation of this project required careful consideration of many factors including but not limited to the framework, layout, entity relationship diagrams (ERD) and the integration between the database backend and the web front-end.

When approaching the initial development of this project, due to the potential likelihood of evolving requirements and solutions in the project the agile methodology was deemed to be the more suitable methodology to follow. This methodology is a continuous iterative approach of development and testing. Working by enabling faster development by allowing teams to deliver work in smaller, more practical increments. This is different from the classic waterfall method of development as the waterfall model is a sequential life cycle approach that only works sufficiently when the scope of the project will not change and is known prior to the start of development.

Agile development is valuable for this project as it allows for partial success during iterations as the more valuable features are prioritised and implemented first which decreases the risk of failure within the project. For this project using an agile approach enables testing to occur throughout the development cycle providing flexibility and allows the key stakeholder to be available throughout the project versus just at major milestones in the waterfall model, enabling for a more consistent feedback.

By using the agile methodology, I will aim to follow the lifecycle as shown in Figure 3 below. More specifically my sprints of length will be four days taking in to account the need for a daily 'scrum meeting' used to plan what will be developed on the day. This also ensures there is time for sprint planning (usually a period of a few hours a week), allowing for prioritisation of key features. As well as one hour a week being spent on the sprint review which takes place during weekly meetings with my client/supervisor starting in person but being moved remotely to skype in line with Coronavirus regulations.

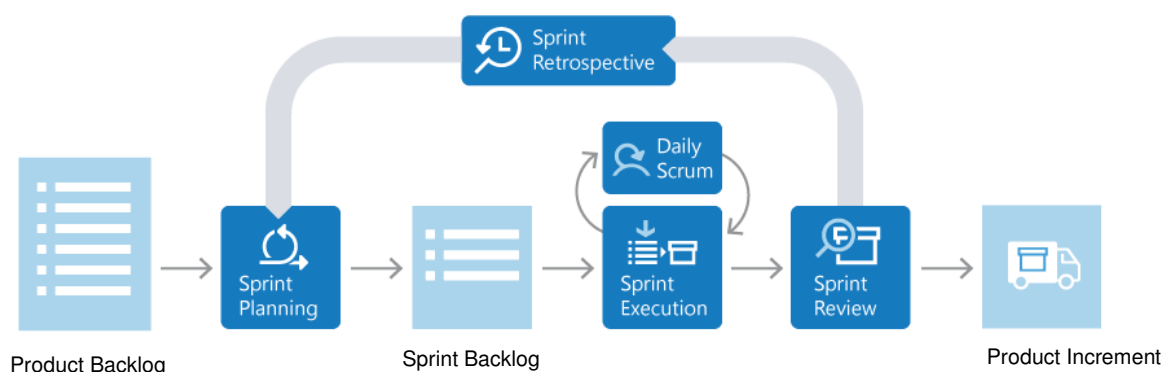


Figure 3: The Scrum Lifecycle (Boer, 2014)

In my approach, I will develop systems to facilitate the automatic retrieval of specific data input of the module review documents and accreditation criteria documents into different tables within a MySQL Database using SSIS (SQL Server Integration Services) on a specific date every year. I will utilise databases as the preferred solution as they are capable of organising data into fields and records and are designed for data management (Masters, 2018). Once this database has been populated with all module review documents and accreditation criteria, I will then use PhpMyAdmin to pull data from the database and display it, along with mappings on a Curriculum Mapping web-interface as shown in Figure 1 below.

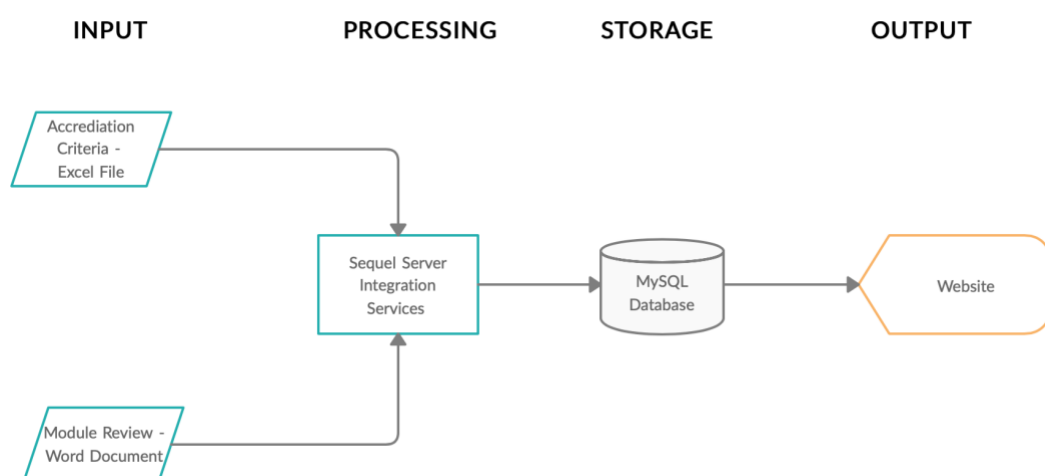


Figure 4: A simple ERD of my project approach

By beginning my approach, I researched into different databases I could use. To assess which one was best to store the information, but also which would be easiest to link with the SQL Server Integration Services extension for Visual Studio – as this was always the basis of my automation mechanism for the project. I also had to consider how the database chosen would fit in with the web front end.

After research my choice came down to MySQL and SQL to stay away from NoSQL databases. With MySQL, it is a database that is more popular when used for web-based applications this functionality seemed like a fit for integration with the web front end of the project. As a free database engine, this also has built in connectors on SSIS through visual studio.

The other option was SQL Server, which is a database management engine that runs Cloud based as well as on local servers. This is a fast and stable database which is guaranteed to work well with other Microsoft products and is easiest to integrate with SQL Server Integration Services.

Ultimately in the end, I chose to host my database using both SQL and MySQL. SQL is used to first store the data in both the excel and word documents are being imported and

transformed to fit in to my specific database structure. SSIS will then export the structure to a MySQL table which can be linked to the web-fronted end to populate the website.

The School of Computer Science and Informatics use a plethora of different platforms therefore there weren't any obvious restricts. With this in mind, I chose Visual Studio which I had some previous experience of using. However, this was not a service that was openly available to students in the School of Computer Science. I therefore took it upon myself to arrange meetings with the IT Service department and present a business case to demonstrate a need for a premium version of Visual Studio. After I presented this, the IT Service department gave me Enterprise access to Visual Studio to begin my project. I then added the SQL Server Integration Services extension to be able to incorporate my database.

For this project, using the SQL Server Integration Services extension on Visual Studio allowed me to automate as much of the review process as possible. SQL Server Integration Services is a platform for building high-performance, enterprise-level data integration solutions – mainly through extraction, transformation and loading packages (ETL) for data warehousing (Microsoft, 2018).

A major advantage and the main reason I chose SQL Server Integration Services for the migration was the fact that SSIS can handle data from heterogeneous data sources within the same package. As the data sources in this project are loaded in from excel and word documents it ensures that they can be loaded in parallel allowing structured handling for multi-step operations. For inserting data from SSIS into SQL this also removes any potential bottlenecks, speeding this process up. This allowed for the facilitation of automatic extraction of review documents which were in turn loaded in to populate the corresponding database tables every year.

System Specification

System Requirements

During my initial meeting with the client for this project, I began the process of requirements gathering by asking questions to gather a basic idea of what the client was aiming to be developed. This basic idea enabled me to begin researching how I would approach this for potential solutions to this problem, I also began research on the BCS: The Chartered Institute of IT accreditation criteria to give an idea of what details would need to be mapped.


In my second meeting I carried out an informal and in person interview asking questions to the key stakeholder on what the current module review process entailed as well as viewing documents that had been produced by the process in previous years. Questions from this interview also covered what key objectives the client was looking to be developed during this project and highlighted websites, mainly COMPAS (Kidd, 2010) that are similar to the idea the client had in mind. I then myself went to compose a list of refined functional and non-functional requirements which I went through with the stakeholder for confirmation before any part of the development process was started to ensure that there was no opportunity for miscommunication. The key outlined requirements from this gathering are detailed below including acceptance criteria which demonstrate what the system does based on the requirement being met.


User Stories


In agile software development, it is easiest to demonstrate understanding from many of the different end users' perspectives through User Stories. This allows me to accurately describe what the user wants and why. This allows a look at the features of the project solution from each end-user's requirements, to represent the needs and wants of the full client base of users, rather than a single individual (Parker Software, 2019). I have also chosen to detail the acceptance criteria for each user story, making it easier to test and determine that the specific user story is completed and working. This is done by setting expectations to be met before me as the developer can mark the user story as 'done'. The user stories for each of the full intended client base are below.


It is important to note that there is alternate to every flow with the user typing in the specific module, learning outcome or assessment name into the search bar at the top right-hand corner of the website. The search will return results based on how strongly they match the term that has been searched, this will assist users who are unsure of where specific modular or accreditation information may be accessed on the website.


Though not highlighted below in the event of system failure the web app front-end will still appear to work for the first few pages. However, when the user selects more specific criteria e.g. a particular module or learning outcome the end data may not appear. This would occur if the back-end MySQL database is down.


Story ID: #01	Story Title: Module Leader User Story	
User Story: <div> <p>As a: Module Leader</p> <p>I want to be able to: click in to view every module along with the subsequent module description, learning outcomes and assessments</p> <p>So that I can: review my module as well as other modules whilst also establishing a prior knowledge base for students.</p> </div>		Importance: <div>  HIGH </div>
Acceptance Criteria: <div> <ul style="list-style-type: none"> • The module leader is able to click on the module option on the home page of the website. • The module page will contain a list of all the degree levels offered by the school. • The module leader will then be able to choose the degree level and course title for the specific module he is looking for. • The module leader will choose the module then the page will display full information for each module including module description, learning outcomes and assessments. </div>		The School is encouraging development of cross-modular yearly review process

Story ID: #02	Story Title: Assessment and Feedback Lead User Story	
User Story: <div> <p>As: Assessment and Feedback Lead</p> <p>I want to be able to: view the assessments that are assigned in each module</p> <p>So that I can: carry out a cross-modular assessment review across a particular year group.</p> </div>		Importance: <div>  HIGH </div>
Acceptance Criteria: <div> <ul style="list-style-type: none"> • The Assessment and Feedback Lead is able to click on the assessment option on the home page of the website. • The assessment page will contain a list of all degree levels within the school. The user will then be able to choose the specific degree level and year group. • When a specific year group is chosen it lists all assessments assigned across all modules offered in that chosen year. • The user can click in to view more information on assessments. </div>		The Assessment and Feedback Lead is the project client.

Story ID: #03	Story Title: Assessment and Feedback Lead User Story	
User Story: <div> As: Assessment and Feedback Lead I want to be able to: view module learning outcomes and BCS accreditation criteria So that I can: carry out reviews to see if the two align. </div>		Importance: <div>  HIGH </div>
Acceptance Criteria: <div> <ul style="list-style-type: none"> • The Assessment and Feedback Lead is able to click on the learning outcomes option on the home page of the website. • The learning outcomes page will contain a list of all degree levels within the school. • The user will then be able to choose the specific degree level and then year group for the learning outcomes they wish to see. • The user will then be shown a list of all module learning outcomes sorted by module. • The user will also be shown the specific accreditation each learning outcome aligns with. </div>		The Assessment and Feedback Lead is the project client

Story ID: #04	Story Title: Year Tutor User Story	
User Story: <div> As: Year Tutor I want to be able to: view information on all modules in a year So that: I can cross-modularly review a year group. </div>		Importance: <div>  MODERATE </div>
Acceptance Criteria: <div> <ul style="list-style-type: none"> • The Year Tutor is able to click on the modules option on the home page of the website. • The modules pages will contain a list of all degree levels within the school. • The user will then choose the year group of which they are the tutor. • When the user has chosen the year group the page shows a list of all modules within that year. • The user can click into a specific module which will show further information about the module. </div>		A year tutor is not deemed a main user of the website.

Story ID: #05	Story Title: Student User Story	
User Story: <div> <p>As a: Student</p> <p>I want to be able to: view the modules offered in each year along with the modules I have chosen to study</p> <p>So that: I can see the assessments and their criteria that I will have to carry out.</p> </div> Acceptance Criteria: <div> <ul style="list-style-type: none"> • The student is able to click on the modules option on the home page of the website. • This opens the modules page which will display all modules offered in the student's current degree. • If the student is undergraduate this page will show options for different years within the degree. • The student will select the year of the modules they would like to see. • The student will be brought to a page which shows all modules within that year. • The student can click on a specific module to get further information on it. </div>		Importance: <div>  LOW </div> <p>This website is not currently available to students, adjustments may be made to allow student access in the future.</p>

Story ID: #06	Story Title: External Examiner User Story	
User Story: <div> <p>As an: External Examiner</p> <p>I want to be able to: view all assessments across a degree programme</p> <p>So that: I can review the degree programme.</p> </div> Acceptance Criteria: <div> <ul style="list-style-type: none"> • The external examiner is able to click on the modules option on the home page of the website. • The modules pages will contain a list of all degree levels within the school. • The external examiner, depending on what degree and year they are examining will choose one. • They will then be able to see all modules offered by the school and can click in to see information such as module description, assessments and module learning outcomes. </div>		Importance: <div>  MODERATE </div> <p>An external examiner is not deemed a main user of the website.</p>

Story ID: #07	Story Title: Academic from Another University User Story	
<div>User Story:</div> <div><div>As an: Academic from another University</div><div>I want to be able to: view information on all modules offered</div><div>So that: I can review the degree programme.</div></div>		<div>Importance:</div> <div><div></div>MODERATE</div> <div>An academic from another university is not deemed a main user of the website.</div>
<div>Acceptance Criteria:</div> <div><ul style="list-style-type: none">The external examiner is able to click on the modules option on the home page of the website.The learning outcomes page will contain a list of all degree levels within the school.The user will then be able to choose the specific degree level and then year group for the learning outcomes they wish to see.The user will then be shown a list of all module learning outcomes sorted by moduleThe user will be shown this along with the accreditation criteria each learning outcome and assignment aligns with.</div>		

Story ID: #08	Story Title: (Deputy) Director of Teaching User Story	
<div>User Story:</div> <div><div>As: (Deputy) Director of Teaching</div><div>I want to be able to: have an be able to view all modules and learning outcomes in a year group</div><div>So that: I can review how cross-modular changes may affect a year group.</div></div>		<div>Importance:</div> <div><div></div>HIGH</div> <div>The Director of Teaching may review any substantial module changes via the website.</div>
<div>Acceptance Criteria:</div> <div><ul style="list-style-type: none">• The student is able to click on the modules option on the home page of the website.• This opens the modules page which will display all modules offered in the student’s current degree.• If the student is undergraduate this page will show options for different years within the degree.• The student will select the year of the modules they would like to see.• The student will be brought to a page which shows all modules within that year.• The student can click on a specific module to get further information on anv changes that occurred within it.</div>		

Story ID: #09	Story Title: Lecturer User Story	
<p>User Story:</p> <div data-bbox="225 409 1098 607" style="border: 1px solid black; padding: 10px;"> <p>As a: Lecturer I want to be able to: view information such as module description and learning outcomes about the module I teach So that: I can review and update my module</p> </div> <p>Acceptance Criteria:</p> <div data-bbox="233 678 1101 1032" style="border: 1px solid black; padding: 10px;"> <ul style="list-style-type: none"> • The lecturer is able to click on the module option on the home page of the website. • The module page will contain a list of all the degree levels offered by the school. • The lecturer will then be able to choose the degree level and course title for the specific module he is looking for. • The module leader will choose the module then the page will display full information for each module including module description, learning outcomes and assessments. </div>		<p>Importance:</p> <div data-bbox="1193 546 1267 613" style="display: inline-block; width: 20px; height: 20px; background-color: #FFD700; margin-right: 10px;"></div> <p>MODERATE</p> <hr style="width: 100%; border: 0.5px solid #ccc; margin: 10px 0;"/> <p>This website can make it easier for lecturers to see module information, but this information can also be viewed on sims.</p>

Non-functional Requirements

Requirement 1: Reliability

The new database must be able to be backed up on the university server.

Acceptance Criteria:

- In visual studio when the project is run the database tables are stored on the university MySQL server
 - When the project is run the database tables are also stored on the university Oracle SQL server
 - The project only needs the data to be stored on MySQL server, however the system writes to Oracle SQL server as a back-up.
-

Requirement 2: Reliability

The new database must be able to maintain performance over time when modules are added, updated or removed.

Acceptance Criteria:

- When a new module is added or removed the module review file is added/removed from the source file.
 - The next time the program is run the module will be added/removed from the database.
 - The modules on the web application will reflect these changes.
-

Requirement 3: Usability

The web front-end interface must be intuitive to use and require minimal to no training.

Acceptance Criteria:

- The web front-end interface has a clearly labelled navigational menu
 - The web front-end also has a search bar that users can utilize if they are unsure of what they are trying to find.
-

Requirement 4: Performance

The web front-end must have a quick response time when queried.

Acceptance Criteria:

- The web front-end interface will be able to be navigated with new pages and more information options loading in 3 seconds.
- These new pages and more information options will be queried from the end database and therefore the response time is based on the response time for the database.

Design

Data Modelling

With my research, I determined the MySQL relational database management system was most the suitable for storage and retrieval by the web app. Using this system also ensured I could integrate and query the data easily and efficiently into the web app using PHP, specifically through the software tool PhpMyAdmin.

Using MySQL as a database, I had to ensure that the most appropriate table layout was selected. To do this I decided on creating one table for the parsed word document data to be stored called 'Module', and four tables for the parsed excel document data to be stored. These four tables were therefore named after the degree level mapping e.g. 'UndergradMapping' for BCS accreditation requirements for undergraduate degrees. Whilst designing I also had to ensure that these tables were normalised to minimise redundancy which can eliminate issues stemming from any database modifications (Wenzel, 2014). This normalisation ensures that only related data is stored in each table. Figure 5 below shows the final data modelling design for data entities included in the tables for this project, this also shows the relationships between entities within the tables.

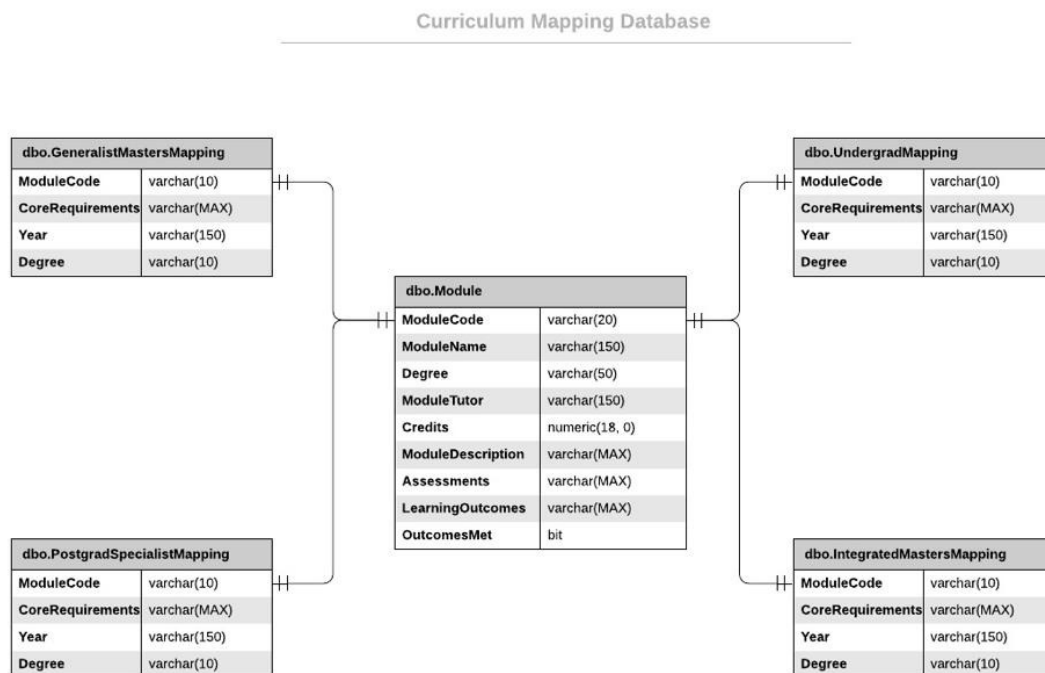


Figure 5: Final data modelling design for MySQL database

Data Flow & Automation

My aim for this project was to create a solution that was easy to maintain and as automated as possible for the client. To facilitate these base requirements, I designed the automation part of the system using Visual Studio. By using this programme, I am able to connect to and import the source documents into VS (word documents for the module reviews and excel documents for the BCS accreditation requirements).

Once the documents have been imported, I can proceed to parse both sets of documents using a Visual Studio scripting component. I felt that programming a Visual studio data tools scripting component was easiest as it meant I could create one parse script for the word documents and another parse script for the excel documents. This would therefore save time as all the word and excel documents shared a common format.

Using Visual Studio to provide the most automated features for the end user ensures that the information pulled from the parsed scripts can then be exported to be stored in my SQL and MySQL database tables. By ensuring that this is all automated this means the whole process can be scheduled to occur on a particular date every year after it is deployed. This can only occur on condition that the files are deposited into two specifically provided folders on the university network (the 'BCSAccredCrit' folder for the excel documents and 'ModuleReview' folder for the word documents). This is folder specific as I have parameterised the Visual Studio project specifically to look only in this folder for importing of the source documents.

Choosing a MySQL database as one of the output sources for the data ensures that it's easier to retrieve the data and use it to populate the web application on curriculum mapping. During my design phase I realised it would be easier to include a pre-parsing step in the Visual Studio project. This step would run code ensuring each time the project is run that if the tables exist in the database they would be dropped, cleared and then re-populated. This guarantees that only data from the most recent project deployment would be stored. This clearing and repopulating also ensures each time the web app is accessed by an end user, they are seeing the most up to date information being queried.

System Architecture

In this design, there are three main types of module within the system. The first being group of modules which handle the automated pulling of data from source module review word files and excel BCS accreditation files into the visual studio project. Within this group of modules, the data is also processed – mainly through the parsing of the documents by a visual studio script component. This scripting component would be programmed by myself to detect and parse particular strings matching certain criteria within the documents and gathering this data. This also flags the fields for processing as well as holding the data detected by the scripting component until it is exported out to the SQL and MySQL databases.

The second module within the project is the module which populates and stores the data into both the MySQL and SQL servers. This involves the use of specific connection managers

containing the server name along with any usernames and passwords required for entry to the databases. Each time the program is run this module has the ability to check if the table name already exists and if so, will drop the table from both of the servers. This results in the tables being cleared and repopulated each time the program is run to ensuring that the data is the most up to date.

The third type of module is the output modules for the web application side of the project. By using PHP scripts within the HTML and CSS of the webpage it allows for the data stored in the MySQL database tables to be queried and pulled on to the web page. PHP is deemed a low-level scripting language and therefore the data is pulled by the programming of several MySQLi queries. This is regarded as the output method of the intended data; it is where the information becomes accessible to the client and visible to end users.

User Interface

To ensure the interface of the web app allows for easy navigation by the end user I created a simple system using HTML and CSS. This aided the creation of a visually functional system. Using both HTML and CSS together ensures the modules are highly extensible. This means they can be adjusted to fit any change in or additional end user requirements whilst minimising impact to the existing system. Essentially this means that the code is easier to maintain.

I began by designing a mock-up of how I intended the user interface to look and I have included these under appendix 1. I was able to provide these to the stakeholder to ensure the layout met visual requirements and was aesthetically pleasing.

Deployment Plan

Prior to the start of development of the project I already had an idea of the basic steps that would be required for the deployment process. Initially I had planned to have deployment fully operational for all staff within the university – however with the coronavirus pandemic it has not been possible to create the shared folders needed and test this way. The current deployment process therefore takes account of this.

Firstly, during deployment I am parameterising the visual studio project to ensure that the MySQL and SQL database servers are university servers. This means they can be accessed from within the university but also outside the university using the university VPN. This VPN has enabled me to access and test these database servers during isolation.

Currently on the university computer system there are shared directories for lecturers in the School of Computer Science and Informatics, in which the client and other lecturers would be able to drop the source word review and excel BCS accreditation files for the visual studio project. As a student I do not have access to these, I have therefore parameterised the visual studio project to look at folders currently stored on my computers desktop for the source word and excel document files. During development to aid automation and ease for the client I have created an instructional step-by-step document which details how the client can change document file paths and deploy the visual studio project as well as

changing DLL file paths. I have ensured this is as easy as possible by making certain the parameterisation for the SSIS project can be done outside of visual studio itself – by deploying the SSIS project in SQL Server and controlling parameters through SQL Server.

Implementation

Due to using an agile iterative approach to the implementation of this project I used iterations to reach milestones in a sequence of repeated cycles. These repeated cycles are what Larman (2003), calls the functional model of the ADTC wheel - “Analysis, Design, Code, Test” - more technically referred to as the PDCA cycle. This more flexible alternative to the long-established and rigid waterfall model ensured client feedback was always encouraged throughout the design and implementation process. The five iterations I carried out were roughly two weeks each and are briefly detailed below.

<p style="text-align: center;"><u>1st ITERATION</u> February 4th – 20th February</p>
<ul style="list-style-type: none">• Create database structure in SQL/MySQL• Create table structure for BCS table• Write visual studio C# code to parse BCS excel file and pull out necessary fields• Create SSIS flow to get parsed data stored into SQL/MySQL tables• Test with different BCS excel files <p>Upon successful testing move on to iteration 2</p>
<p style="text-align: center;"><u>2nd ITERATION</u> February 21th – 8th March</p>
<ul style="list-style-type: none">• Create table structure for University table• Write visual studio code C# to parse word document source and pull out necessary fields• Create SSIS flow to get a word document into SQL/MySQL tables• Test with different module review word documents <p>Upon successful testing move on to iteration 3</p>
<p style="text-align: center;"><u>3rd ITERATION</u> March 9th – 25th March</p>
<ul style="list-style-type: none">• Expand visual studio to bring all word documents for each module into flow• Expand visual studio to bring all excel documents for each degree level into flow• Populate database table University• Test that database tables are fully populated <p>Upon successful testing move on to iteration 4</p>
<p style="text-align: center;"><u>4th ITERATION</u> 26th March – 7th April</p>
<ul style="list-style-type: none">• Development of web front-end mock-up• Check with client to see if they approve page design mock-up• Make page design using HTML, CSS and PHP – make future pages using template• Test that web-front end <p>Upon successful testing move on to iteration 5</p>
<p style="text-align: center;"><u>5th ITERATION</u> 7th April – End of April</p>
<ul style="list-style-type: none">• Aim to try and get method for matching feature working• Test method

In the previous design section of this report I gave a high-level overview of the architectural and modular design of the project. In this section I will begin to talk about this in a finer, low level detail. Any screenshots and code described in this section are the result of the completion of the final iteration.

The final version of my system is comprised of a visual studio integration services package and a visual studio solution package – these also contain the scripting component using the visual studio tools for applications (VSTA environment). As well as MySQL database files – this does not include the files for the source documents as these are already parsed and populated in the MySQL database tables. In this implementation section I will be highlighting the various methods of code I have used throughout as well as depicting the reasoning behind my decisions to implement it the way I have.

The first module of this program deals with the automation of project. This was a key feature of the program as my client specified their need for this to be as automated and as easy to use as possible. For this I used Microsoft Visual Studio programs SQL Server Integration Services extension and the Visual Studio Tools for Applications (VSTA) extension environment. I chose this as I was already familiar with how to use the Visual Studio program having gained quite a bit of experience on creating an automation and transformative project whilst on my Year in Industry. I had an original outline plan of what I wanted to achieve and a vague idea of how I wanted to achieve it. I came across many additional features and capabilities offered by the Visual Studio IDE during the research stages. These additional features included the VSTA environment and ensured I would be able to carry out all project automation and parsing using the one program which made the choice to implement with the Visual Studio IDE an extremely easy one.

By first using a script component as a data flow option in my visual studio project I was able to parameterise the script to look for specific variables in the project called “DLLPath”, “WordDocRootFolder” and “ExcelRootFolder” in specific locations on my computer desktop. As both documents are formatted to only allow specific input this ensures that the data being supplied to the system is consistent in both content and format. Therefore, this meant if the documents needed were in the folders and were parameterised when the project ran, the program would import these documents. However, if there is no document in that folder that matches the criteria of the parameters then the program will not be able to run and will error as it will have no source documents. To enable ease of use for the client, during implementation I created a word document that contained step-by-step instructions on how to change parameters for the project. This ensures if in the future the locations need changing or updating the client is able to do this easily.

```
public ScriptMain()  
{  
    AppDomain.CurrentDomain.AssemblyResolve += new ResolveEventHandler(CurrentDomain_AssemblyResolve);  
}
```

Figure 6: Project code extract

The code snippet in Figure 6 above, is from the main.cs file and shows how I coded the loading and handling of the DLL (dynamic-link library) path files necessary to instruct the scripting component to carry out various instructions. By including this in the ScriptMain

class this allows the client to be able to parameterise and change the DLL paths in the future.

This module also therefore includes the scripting component which I have used to parse the module review word document files and the BCS accreditation requirement excel files. This scripting component enabled my package to run custom script code for the parsing. This VSTA environment extension supported the use of the C# programming language, which I was unfamiliar with and had no previous experience of using. In order to familiarise myself with this language I sourced the internet for online resources. This research taught me that this language was object-oriented, and class based, component-oriented programming language that was more flexible to use than another programming language such as Java. The C# code below relies on the DocumentFormat.OpenXml.Packaging and DocumentFormat.OpenXml.Spreadsheet DLL files.

```
private static void ProcessSheet(ISheet sheet)
{
    //get the degree first
    string degreeCellAddress = "Q1";
    var cr = new CellReference(degreeCellAddress);
    var row = sheet.GetRow(cr.Row);
    var cell = row.GetCell(cr.Col);
    string degree = cell.StringCellValue;
```

Figure 7: Project code extract

The code shown above in Figure 7 was used to process and find the degree level value from within the XLS spreadsheet by first converting the cell description from excel (Q1). Then using the variables 'row' and 'col' with the 'GetRow' and 'GetCell' method to look up the actual cell in the excel file. The value that is in this cell reference then gets converted to a string named 'degree'. This value is then held until the code for the rest of the excel data is run. This is run for every data value in the excel, however in my implementation I will only follow and explain the code relating to the one value, the 'Degree' level value. When all the data is converted to strings the excel data is then able to be exported to be stored in the exOut object of the class ExcelDataOut as shown in Figure 8 below.

```
else
{
    ExcelDataOut exOut = new ExcelDataOut();
    exOut.Degree = degree;
    exOut.CoreRequirements = coreRequirements;
    exOut.Year = yearData;
    exOut.ModuleCode = sheet.GetRow(rowNum).Cells[i].StringCellValue;
    excellOutData.Add(exOut);
}
```

Figure 8: Project code extract

Due to this coding being in the C# programming language, this coding was beyond course learning and was also vital to the successful operation of the system. This is due to the fact that the parsing code ensured that the values taken from both sheets corresponded with the values I had set the database tables to hold. (These can be seen in the entity relationship diagram I have included in the previous design section).

These values were then transferred to the ExcelDataOutBuffer which was added to the class CreateNewOutputRows() as shown below. In the main.cs file I after I created the new class, I added the data buffer to output the data from the parsed module review word document which I included in a for loop but have omitted in the code shown in Figure 9 below.

```
public override void CreateNewOutputRows()
{

    /*module review word document output code omitted*/

    foreach (var item in excelOutData)
    {
        ExcelDataOutBuffer.AddRow();
        ExcelDataOutBuffer.CoreRequirements = item.CoreRequirements;
        ExcelDataOutBuffer.Degree = item.Degree;
        ExcelDataOutBuffer.ModuleCode = item.ModuleCode;
        ExcelDataOutBuffer.Year = item.Year;
    }
}
```

Figure 9: Project code extract

Throughout the coding snippets I have highlighted in this implementation section so far, it is easy to see that C# is similar to the Java programming language. However, Java was not a coding language that worked with the .NET Framework and compiler I would be using through Visual Studio. Therefore, using the C# programming language during this part of the development made the most sense and was easiest for me. To help with the day to day coding using the C# language, I also included help instructions within the comments of my code to assist me in the event I got stuck or became confused while working on the development. These can be viewed throughout the main.cs file that has been submitted alongside this report.

The next module in the design and implementation of this project was the automated creation and population of the SQL and MySQL database tables. This occurred within the main Visual Studio project (not within the VSTA like the previous module) as the data “excelOutData” and “DataOut” flow to be stored in separate SQL tables. When I first began implementation on this module, I had just connected the script component outputs to the SQL tables in my database using the OLE DB Destination feature in the SQL Server Integrated Services toolbox on Visual Studio. This connection was established from visual studio to the SQL database using an OLE DB connection manager. However, during the first point of testing I realised that every time the program ran it was adding the data in to the database tables without any data being cleared or deleted. This meant the database was holding a large amount of duplicate data.

To resolve this issue, I made the decision to add an OLE DB Command toolbox feature between the VSTA scripting component and the connection to the SQL database for both the 'Module' and 'UndergradMapping' tables. This additional OLE DB Command feature meant that every time the program was run the tables from past program runs were dropped and recreated ensuring only the most recently pulled through data was populating the tables. The code for the dropping and creating of the 'Undergrad' table using the 'Clear SQL Undergrad' OLE DB Command in Visual Studio is shown in Figure 10 below.

```
SET ANSI_NULLS ON
```

```
SET QUOTED_IDENTIFIER ON
```

```
DROP TABLE IF EXISTS [dbo].[UndergradMapping]
CREATE TABLE [dbo].[UndergradMapping](
    [MODULECODE] [varchar](10) NULL,
    [CORERQUIREMENTS] [VARCHAR](max) NULL,
    [DEGREE] [varchar](150) NULL,
    [Year] [varchar](10) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY];
```

Figure 10: Project code extract

Figure 11 below also shows the 'Clear SQL Undergrad' OLE DB Command positioned in the project between the scripting component and the export to the SQL database tables.

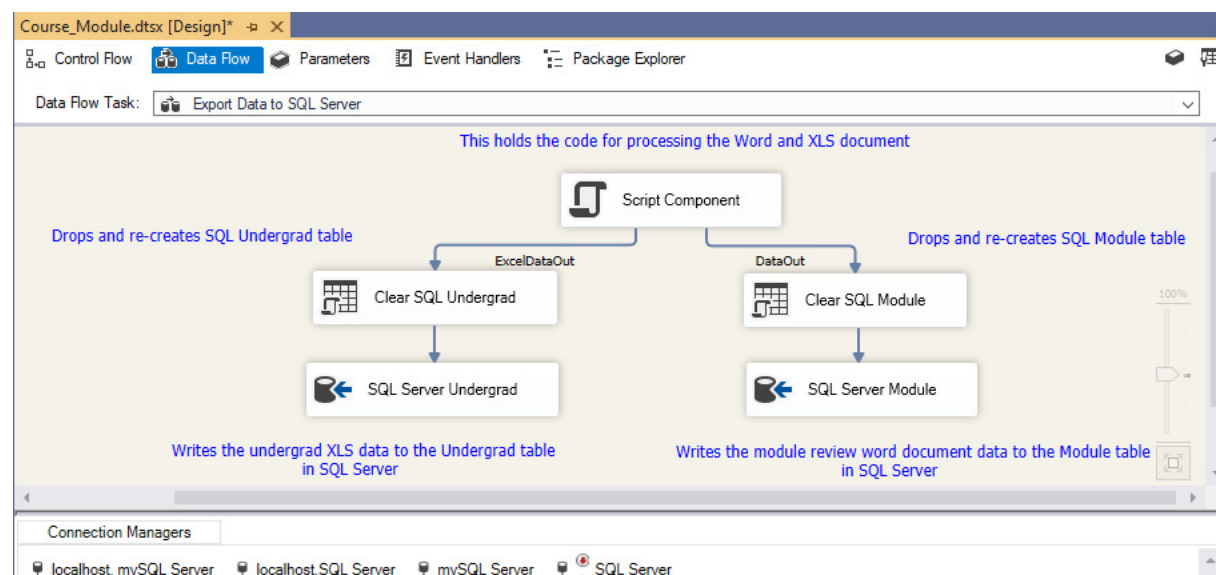


Figure 11: Visual studio screenshot of 'Export Data to SQL Server' data flow

When connecting to the database I have used an OLE DB connection manager which stores the server name of the MySQL database and the username and password needed to reach them. On the connection, the tables are visible, and mapping can occur both manually and automatically to match data coming into columns in the specific database tables. For this,

the mapping was automatic as the data coming in matched the names of the columns within the database table.

To ensure the utmost automation for the system, I have ensured that when the programme is run that this populates the MySQL tables connected to it. To view these, I can log on to the desktop PhpMyAdmin website via the university VPN as the databases and tables are stored on the university oracle server. I have included a screenshot of a specific query run on the table to show the layout and included this as appendix 2 in this section. To aid the automation for the end user further I have also programmed that the 'job' (in this case the SSIS project part of the system) will run on a particular day every year within SQL Server Agent. By enabling an email profile for the end user on the alert system within the SQL Server Agent. With this the end user will then receive a confirmation e-mail after the system has successfully run every year, or a warning email if the run has been unsuccessful - along with error messages explaining what went wrong.

As a result of not storing any confidential data I was able to assess that persistent data storage in the system was not a problem. However, in the future encryption of the system could be introduced. I will discuss this idea further in the future work section of this report.

The third and final module of the project deals with the web-app front end/interface of the project. For this I aimed to design and implement a simple and intuitive user interface for the system. This was mainly done using HTML and CSS to provide an extensible, fast and perceptible visual functionality for the system. This HTML and CSS had the same main code used as a template throughout to ensure that the pages all maintained a high level of consistency.

To enable integration of the MySQL database in with this HTML and CSS, I incorporated the embedding of PHP – a low level general-purpose scripting language. These PHP scripts were used to generate dynamic page content that accurately reflects the data currently stored in the MySQL database tables. The inclusion of PHP ensured that the data was capable of being queried to pull through specific information to be displayed across the interface for the end user. To aid the addition of PHP integration after the HTML I moved to MAMP as my chosen PHP development environment, this enabled me to run a live localhost MySQL server and also aided in local testing when it came to conducting this.

```
// Checking mysql connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Writing a mysql query to retrieve data
$query = "SELECT MODULECODE, MODULENAME FROM `module` WHERE MODULECODE LIKE
'CM1%' AND DEGREE = 'BSc Computer Science'";
$result = $conn->query($query);

if ($result->num_rows > 0) {
    //Show each data returned by mysql
    while($row = $result->fetch_assoc()) {
```

```

echo '<li><a href="moreinformation.php?ID='.$row["MODULECODE"] .' class="mod- link"></li>
.$row["MODULECODE"] . " : " . $row["MODULENAME"]; '</a>'?'>
<?php
}
} else {
    echo "0 results";
}

```

Figure 12: Project code extract

The snippet of code in Figure 12 above shows how I was able to integrate the PHP code with the HTML code of the web page to execute a specific MySQL select query to return the module codes and module names currently in the database where the module code contains 'CM1' (making this a 1st year module) and the degree course being 'BSc Computer Science'. To enable multiple queries to be executed on the same page, it was essential to change the \$query variable name, in my case I numbered my queries. This is shown in Figure 13 below.

```

// Writing a mysql query to retrieve data
$query1 = "SELECT MODULECODE, MODULENAME FROM `module` WHERE DEGREE = 'BSc Applied
Software Engineering' AND MODULECODE LIKE 'CM6%'";
$result1 = $conn->query($query1);

if ($result1->num_rows > 0) {
    //Show each data returned by mysql
    while($row1 = $result1->fetch_assoc()) {
        ?>

        echo '<li><a href="moreinformation.php?ID='.$row["MODULECODE"] .' class="mod-
link"></li> .$row["MODULECODE"] . " : " . $row["MODULENAME"]; '</a>'?'>

```

Figure 13: Project code extract

By clicking on the particular module or learning outcome, this in turn brings the user on to a new web page which displays more data on the particular module or learning outcome that was selected including the BCS requirements that it meets. To enable this to happen most efficiently in the module section of the web front end, I named the new web page 'moreinformation.html' and passed the Module Code through as an ID parameter. This ensured that when the new page opened, the \$ID_Parameter was getting the module code the user had selected from the last page. This \$ID_Parameter is then used in the page query to pull through further information. The full snippet for this code is shown in Figure 14 below.

```

$ID_Parameter = $_GET['MODULECODE'];

$query2 = "SELECT * FROM `module`";
$result2 = $conn->query($query2);

if ($result2 = $ID_Parameter)
    echo '<h2 class="heading-39291"><b>Module Code:</b></h2> '.$row2["MODULECODE"] .'<br>';

```

Figure 14: Project code extract

Unforeseen Issues

Outsourcing

In this project, following extensive research I decided that the option of Visual Studio was the program that would be most beneficial to use for automation, transformation, loading and extraction of data. This was based on utilising and importing from Excel and Word document sources exporting into the MySQL database. Yet my primary laptop was a MacBook. With the Visual Studio for Mac application new it consequently does not contain the same features as the Windows version. Using university windows labs ensured I was able to use the windows version of Visual Studio. However, with the coronavirus pandemic occurring and university closing due to self-isolation rules being enforced this meant my access to labs was impeded. Therefore, I had to resort to outsourcing an old windows laptop from a friend to be able to continue development, which took time. The computer itself has been very slow when running a program as big in size as Visual Studio with the necessary extensions and this has been very frustrating.

University VPN

When I eventually outsourced the windows computer, I needed to be able to VPN on to the school server to enable the continuation of the work I had carried out in the university windows labs. Upon contacting the university IT team, the VPN was up and ready to run. However, before using the OpenVPN the documentation told me I must obtain a .ovpn file from the Undergraduate Certificate Manager portal which was required to be downloaded from within the university campus. This wasn't possible as I had left Cardiff to self-isolate at home. Therefore, I had to find a course-mate still in Cardiff to go on campus and download my .ovpn file at the same time he was downloading his own. When I received this, I was then able to access the university server successfully from home.

Misinterpretation

At the start of the project, I received some module review word documents from the 2018/2019 academic year from the client. I also received the four different BCS accreditation mapping documents based on each degree level e.g. Undergraduate, Integrated Masters etc. However, the excel spreadsheet document for the accreditation mapping had not been filled out. I based the parsing on this document to pull out specific information that had been highlighted by the client. During the later stages of the project when I requested a populated version for testing purposes, I realised the document was filled in differently by the client than I had anticipated. This therefore meant I had to adjust my parsing visual studio script component to take account for this.

Going back to review the parsing module of the project also provided its own difficulties as it had been nearly two months since I had begun coding the parsing script. Due to this time lapse and the fact I had not annotated my code as well as first thought I found it harder to follow back through and make the adjustments necessary. This encouraged me to carry out a full review of my code and increase my standard of source-code annotation. As this difference in anticipated against actual data only became apparent during testing, towards

the end of development, resolving this parsing problem fixed was a pressing and time-sensitive issue which I managed to over-come in the end.

Results

In this section I will test the current system including the interface that I have implemented over the duration of this project. By testing the system against the functional and non-functional requirements I established during preliminary meetings with the client, I am able to evaluate the extent to which these have been met.

To ensure that testing was carried out to reflect real-world conditions, I added both completed module review documents for BSc and MSc Computer Science along with BCS guideline mapping documents from a previous year into the source folders on my desktop. From here I was able to run the 'Course_Module_Import.sln' SSIS project which when run in turn populated the SQL and MySQL databases with this information. I then carried out my tests based on this.

(It is worth noting that Use Case #05 was deemed additional functionality and was not part of the main requirements for this project. During a discussion with the client, we therefore decided that we would not include this in the implementation and is therefore not included in the test cases below. This enabled the idea to be kept for potential further work that may be carried out in the future - this will be further discussed in the Further Work section later in this project.)

Test Cases

Below are the test cases reflective of the requirements and aims of this project that are detailed in the system specification section above. Each test case ID also matches the User Story ID/non-functional requirement ID in the system specification section of this project. If two use cases were similar but from different end users, I have merged them in to the one test case.

Functional Requirements

	Test Case ID: #01		
Summary:	The system must allow all users to view a module along with more detailed information including the description and learning outcomes. Module review information must be specific to the module chosen.		
Preconditions:	The user has the GUI open. The user is on the university campus or on the university VPN. The 'Course_Module_Import.sln' file has successfully run in the system at least once over the past year.		
	Test Case Steps		
Test Procedure	Expected Result	Actual Result	Test Outcome
1. The user navigates to the module page of the website using the navigation bar. 2. The user selects the undergraduate module option. 3. The user selects a 1 st year computer science module. 4. The user selects the CM1102 module code.	The system will display the following information specific to module CM1102: <ul style="list-style-type: none">- Module Code- Module Name- Degree- Module Tutor- Credits- Module Description- Learning Outcomes- Outcomes Met	All of the information for module CM1102 is shown apart from whether the learning outcomes have been met. This is due to them not being met.	Pass

Test Case ID: #02/04/#07/#09			
Summary:	The system must allow the user to view all modules and assessments offered by the school across all years and degrees to allow for cross-modular review.		
Preconditions:	The user has the GUI open. The user is on the university campus or on the university VPN. The 'Course_Module_Import.sln' file has successfully run in the system at least once over the past year.		
Test Case Steps			
Test Procedure	Expected Result	Actual Result	Test Outcome
1. The user navigates to the module page of the website using the navigation bar. 2. The user selects the undergraduate option on the module page. 3. The user goes back to the module page of the website. 4. The user selects the postgraduate option on the module page.	- The system will display all undergraduate and postgraduate courses offered in the school along with all the core modules and assessments.	The system displays a list of all the undergraduate and postgraduate courses offered by the school on the corresponding pages alongside a full list of the core modules for each degree and year.	Pass

Test Case ID: #03			
Summary:	The system must allow the user to view module learning outcomes and BCS accreditation criteria.		
Preconditions:	The user has the GUI open. The user is on the university campus or on the university VPN. The 'Course_Module_Import.sln' file has successfully run in the system at least once over the past year.		
Test Case Steps			
Test Procedure	Expected Result	Actual Result	Test Outcome
1. The user navigates to the learning outcomes page of the website using the navigation bar. 2. The user selects the undergraduate module option. 3. The user selects the year of the module containing the learning outcome they are looking for. 4. The user selects the learning outcome “Understand the knowledge management lifecycle and activities”.	The system will display module information for CM3107 Knowledge Management as this learning outcome is from this module as well as the BCS accreditation criteria this module meets.	The system displays all module information for CM3107 Knowledge Management including the BCS accreditation guidelines that the module meets.	Pass

Test Case ID: #08			
Summary:	The system must allow the user to view modules and learning outcomes for all degrees offered by the school.		
Preconditions:	The user has the GUI open. The user is on the university campus or on the university VPN. The 'Course_Module_Import.sln' file has successfully run in the system at least once over the past year.		
Test Case Steps			
Test Procedure	Expected Result	Actual Result	Test Outcome
1. The user navigates to the modules page of the website. 2. The user selects the undergraduate option or postgraduate option. 3. The website displays all the core modules from the degree in these degree levels. 4. The user navigates to the learning outcomes page of the website. 5. The user selects the undergraduate option or postgraduate option. 6. The website brings up all the learning outcomes from modules in the courses in these degree levels.	- The system displays all the core modules from the courses in these degree levels - The system displays all the learning outcomes from the modules in the courses offered by the school.	The system displayed all the core modules and learning outcomes from courses offered in all degree levels within the school of computer science and informatics.	Pass

Non-Functional Requirements

Test Case ID: #1.1			
Summary:	The database must be able to be backed up on the school server.		
Preconditions:	The user is on the university campus or on the university VPN. The module review and BCS mapping documents are in the source folders. The 'Course_Module_Import.sln' file has successfully run in the system at least once over the past year.		
Test Case Steps			
Test Procedure	Expected Result	Actual Result	Test Outcome
1. The user opens the 'Course_Module_Import.sln' file in the Visual Studio IDE. 2. The user runs the SSIS project. 3. The user logs on to the university's Oracle SQL server database. 4. The user queries the tables using the query 'SELECT * FROM MODULE' and 'SELECT * FROM UNDERGRADMAPPING'.	The SQL database is present and the tables are fully populated.	The SQL database is present and the corresponding tables are populated on the schools Oracle SQL server.	Pass

Test Case ID: #1.2			
Summary:	The SQL and MySQL databases must be able to maintain performance over time when modules are added, updated or removed.		
Preconditions:	The user is on the university campus or on the university VPN. The new files to be updated/removed from the database are added/removed from the corresponding source folder.		
Test Case Steps			
Test Procedure	Expected Result	Actual Result	Test Outcome
1. The user would go to the source folder path. 2. The user would add/remove a module review file or add updated file and remove old file. 3. The user would open the 'Course_Module_Import.sln' file in Visual Studio. 4.The user would click run.	The MySQL and SQL database system shows it is populated with the most recent information, the changes in modules being added/adjusted or removed are visible.	The MySQL and SQL database tables are all up to date with any changes, adjustments or deletions visible.	Pass
Alternative Procedure			
1. User would go to the source folder path. 2. The user would add/remove a module review file or add updated file and remove old file. 3. The user would wait for assigned day for the system to automatically run.			

Test Case ID: #1.4			
Summary:	The web-front end must have a quick response time performing queries.		
Preconditions:	The user is on the university campus or on the university VPN.		
	The module review and BCS mapping documents are present in the source folders.		
	The 'Course_Module_Import.sln' file has successfully run in the system at least once over the past year.		
	The user is on the web-front interface.		
Test Case Steps			
Test Procedure	Expected Result	Actual Result	Test Outcome
1. The user navigates to the module or learning outcomes page using the navigation bar of the website. 2. The user chooses from the undergraduate or postgraduate options. 3. The user clicks on a particular learning outcome or module from the list provided by the interface to see more information.	The interface will display more information on the user's choice in less than 10 seconds.	There is less than a 10 second wait for the user before information is displayed on the interface.	Pass

User Testing

When detailing my requirements earlier on in the project, I made a plan of how each would be tested. With this I realised that using test cases for certain requirements, predominantly the usability of the system could not be accurately tested this way. The International Organization for Standardization (2013), defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Taking this into consideration when making my requirements, I decided that user testing through observation and listening would be more beneficial for testing purposes to allow me to evaluate the extent to which the implemented system works in comparison with the project objectives.

At the beginning of the project, I had planned to conduct user testing with the client on the university campus. However, a different approach than originally planned was decided upon to accommodate COVID-19 quarantine guidelines as I could not observe in person. As the web front end was locally hosted, I conducted user testing with people in my household.

For this I gave them an initial idea of what curriculum mapping was and what aspects I had mapped in this project. As they have both attended and graduated from university, they were familiar with the pedagogy of teaching including module layout alongside assessments and learning outcomes. Both test users also completed degrees with accrediting bodies (Business and Marketing – Chartered Institute of Marketing and Psychology – British Psychological Society respectively). This ensured that the users were familiar with the concepts of this project.

By including the observational user testing I was able to ensure that any problems I may not have thought of can be addressed before the system is finalised. For this user testing, I observed the test users carrying out “blind” naturalistic observation involving the test user’s behaviour as they interacted with the web site as they would with any other website. Using this naturalistic observational approach is deemed more reliable as the test users are more likely to experience any problems navigating intuitively rather than following a set of instructions on how to navigate.

I kept a record of both test users’ actions whilst on the interface and noted down any feedback or questions they had. This is included below.

User Actions

Test User 1

1. User scrolls down home page to read through information the page contained.
2. The user clicked on to the modules page in the navigational menu.
3. The user clicked on the undergraduate photo to bring them on to the undergraduate module page.
4. The user quickly clicks a few of the drop-down menus.
5. The user goes back to the ‘Year in Industry/Year Abroad’ option and clicks on the ‘CM2500: Placement’ module.
6. The user quickly inspects the information presented on the module.
7. The user clicks on to the Cardiff University logo in the navigational menu.
8. The user clicks on the admin page using the navigational menu.
9. The user enters a random username and password, which was incorrect.
10. The user asks for a username and password to test this feature. When given the correct admin username and password this worked.
11. The user clicks the help page in the navigational menu.
12. The user reads through the FAQs section on this page.

Test User 2

1. The user firstly adjusted the browser size of the home page of the interface.
2. The user scrolled to the bottom of the home page and clicked on the Twitter and Facebook social widgets in the ‘Follow Us’ footer section.
3. The user used the back button on his browser to return to the home page.
4. The user clicked on the ‘burger’ menu of the reduced browser size to scroll down and click on the learning outcomes option.
5. The user then clicked on the postgraduate option on the page.
6. The user then clicked the specialist postgraduate option.
7. The user clicked the ‘(MSC) Cybersecurity’ option and observed the more information displayed on this module.
8. The user then clicked on the Contact page using the navigational menu.
9. The user filled out the ‘Contact Us’ form and submitted this.
10. The user then selected the Year page from the navigational menu.
11. The user expanded the years for each degree to see what they showed.

User Feedback/Questions

Test User 1

Test user 1 found that overall the website was both easy to navigate and responsive. They felt the design was simple and consistent, particularly referencing through feedback how the site being uncluttered ensured the user was able to focus on what was important within the website. The user identified that on the admin page whilst logging in, he could not log in by pressing the enter button after filling in the username and password and could only log in by using the 'Login' button. The user provided feedback that many other websites allow the use of the enter button to do the same thing as the login button as a shortcut. The user also suggested the use of the enter button to allow the user to move from entering the username to entering the password. Therefore, I adjusted the log in feature to allow the submission of the form by adding a JavaScript 'on key down' function to begin the username and password check just as the login button does. The user also agreed that all questions he would have had on the website would be answered through the FAQs or through the introduction information I had explained at the start. This brief introduction only contained information that would be prior knowledge to staff members of the School of Computer Science and Informatics when using the interface.

Test User 2

Found the site was extremely easy to navigate by not having unnecessary links or numerous pages to go through to find important information. This user found that the website was very easy to learn to navigate, as they needed only brief prior introduction to understand what information normal users would be looking for within the website itself. The user particularly highlighted that the page maintained visual aesthetic even when the browser was not full size. This user provided feedback that mostly entailed a few small changes. The first was the biggest change and was more important. This was that they noticed the first letter of every word in forms were capitalised, when this came to the log in feature this therefore could have a massive effect on users if they had a password created that started with a lowercase letter. I changed this to ensure all text was not auto capitalised. The next part the test user expressed feedback on was the 'Contact Us' form there was not a submission receipt or confirmation to confirm the form was sent. Consequently, I added a form validation function which when the form was completed and the 'Send message' button clicked that the user would receive confirmation that their form had been submitted. The user also enquired about where the information was sent on submission, which I clarified was emailed to the temporary created 'curriculummapping@outlook.com'.

Evaluation

With the creation of the web front-end using HTML, CSS and PHP to allow integration with the MySQL database it was essential to ensure the front end was simple to use. Whilst designing I had to take in to account the qualitative guidelines that it would be evaluated against; these being Jacob Nielsen's 10 Usability Heuristics for User Interface Design (Langmayer, 2019). A screenshot of a page within the interface that the heuristics were evaluated against is included in appendix 3 at the end of this report.

Heuristic Evaluation

Heuristic	Suitability
Visibility of System Status	Through the interface, when there is a choice for the user to make, the options change colour when hovered over. When the system errors at the user's choice which has not been carried out, the code will give an error message telling the user that the website can't connect to the database or that there are 0 results matching.
Match between System and Real World	The GUI front-end is written in English and uses formats that tie in with the module review document format that lecturers are familiar with. It is very clear and self-explanatory how the interface system is meant to be interacted with.
User Control and Freedom	With the front-end of the system, the user is free to navigate and choose any module or learning outcome that is displayed on the system to see more information on this. The user is also able to alter and delete input on the 'Contact us' form on the contact page of the web interface.
Consistency and Standards	Throughout the GUI there is a consistent layout, which is similar to other websites with a navigation bar etc so will be familiar with the user. The same fonts and font sizes are used throughout to ensure consistency through every page.

Error Prevention	As the system is scheduled to run as a job on the SQL Server Agent once every year, alerts within this are enabled so the end user gets an email confirming whether the job has run successfully or not.
Recognition Rather than Recall	The navigational menu along the top of the interface is clearly visible and each page contains instructions on how the user can access further options.
Flexibility and Efficiency of use	This interface relies on human interaction; however, it is flexible to allow for users of all skill levels as all the user has to do is navigate using knowledge they already have.
Aesthetic and Minimalist Design	The interface is minimal with no unnecessary clutter or information. The colour scheme is both pleasing to the eye whilst also maintaining consistency with the Cardiff University intranet and logo colours.
Helps user recognise, diagnose and recover from errors	If the user has anything to add they are directed to use the 'Contact Us' form on the Contact page of the navigational menu of the interface.
Help and Documentation	There is a help option on the navigation bar of the website. Here the user is brought to a section of Frequently Asked Questions, these will be updated over time as the website generates more use. Throughout implementation I have made a help document for the end user to instruct how to change various paths within the system and how to deploy the system.

System Evaluation

Through the test cases and user testing methods I have utilised in my testing section, I feel that this has accurately highlighted any alterations or improvements needed within the system. The results of this testing reflect that all bar one of my requirements were successfully met by the implemented system. I believe that this reflects that the system is successful and highly suitable for use by the client and other end users to assist their day to day work.

I feel that utilisation of user testing enabled an observational response on the system that highlighted any problems. This also benefited final development of the system as it allowed for the suggestion (and incorporation) of additional functionality, which provided invaluable feedback. Despite having to modify the methods used in testing which prevented direct client user observation taking place, I still achieved a level of client feedback on the system. This occurred mainly during the weekly meetings with the client, involving the layout and aesthetics of the interface.

Throughout the development and implementation of this project my main goal has been to build a system that was dynamic and automated what was formerly a laborious manual process for the client. I believe that my completed system fully embodies this. Whilst the system does not match learning outcomes to the particular accreditation requirements, due to the extended time period required to facilitate creating the matches. This would have involved having to re-format the source module review and accreditation guidelines documents and introduce this across the School of Computer Science and Informatics. However, I feel that the system is designed in a way that is easily extensible to accommodate any further implementations further down the line. In spite of this, the system I have created allows for easy automation by parsing and populating documents whilst presenting them through a functional and visually appealing interface benefiting not only the client, but all staff members throughout the School of Computer Science. Irrespective of the focus on automation, the system also supports manual input or manipulation of records within the database if necessary, for the client.

I believe that the implemented system reflects well on my methodology. Working with the agile methodology I knew there was possibility for less predictability of the full extent of required efforts. However, I feel that I balanced this well in the planning I undertook for the project. Using this methodology, the short feedback loops established end user feedback and communication that was constant throughout, enabling for true integration of the client in the development process.

The system also reflects well on my choice of development architecture. With the majority of the back-end system being developed in Visual Studio I discovered IDE extensions that when combined with online documentation and guides provided help to solve any problems or surpass difficulties efficiently. Visual Studio itself was a brilliant IDE to choose as during implementation the ease of integration with other parts of the system became apparent. I also believe that the structured nature of the data and data flow in the system works seamlessly with MySQL. This choice of database ensured ease of use and excellent query time within the web interface of the system through PhpMyAdmin.

Future Work

Despite the successful creation of a dynamic and automated system for the client, there is still room for future work or adjustments to the system to reflect any process changes within the School of Computer Science and Informatics.

Taking into account what I was able to achieve during the implementation and development of my project, I feel the next stage for the future would be making the web-based front end available to students. By making information such as module learning outcomes and accreditation criteria visible to students this would enable learning trajectories to become explicit. This could in turn help promote reflection and potentially enhance the quality of their learning processes through what Diamond, (1998) terms “transparent articulation”. This would also help students to better stimulate prior knowledge recall and relate it to new knowledge or meaningful context according to Merrill, (2002). Being open to students would enable tailoring to each student, allowing them to see the modules they have completed during their degree as well as the module they are currently studying. This would also be beneficial for students providing more information on modules when it came to choosing modules for the next academic year.

If future work were to continue on this project, I would predominantly like to ensure that the system offered the matching facility of learning outcomes to their subsequent BCS accreditation criteria. This would ensure that the interface could offer maximum benefit to the client as Assessment and Feedback lead within the School. Due to the easily extensible design of the system, the addition would not be hard to integrate into the system itself but the concept of high success rate matching itself would be complicated to successfully develop, as I found in my unsuccessful attempts during development.

In the future I would also aim for expansion to include other courses within Cardiff University that maintain accrediting bodies for their degree accreditation. These can include but are not limited to Law, Engineering and Optometry but can also expand to the medical professions like other similar solutions that I detailed in my background research have. This would enable the web front end to be another open Cardiff university intra-organisational website similar to the Intranet, Learning Central and SIMS which together form the primary hub for staff and students.

As the data stored in this database currently is not confidential in any way, persistent data storage is permitted. However, if the database was altered to store more confidential data in the future then data protection laws must be adhered to. For this I would suggest encryption of the system. This could potentially raise the cost of the system as it would involve the overhead processing cost of encrypting and decrypting the data and could decrease performance of the system. If this were to be introduced it would also take time to implement which would mean the front-end web app may need to be down temporarily.

Conclusions

The main objective of this project was to improve the way that the School of Computer Science and Informatics conducted their module review and curriculum mapping processes. The current process was laborious and inefficient, with the work occurring manually by the Assessment and Feedback lead within the School of Computer Science and Informatics. Due to the fact this occurred yearly and with data stored in such a consistent format I knew that the development of an automated system would be most advantageous for the client.

The majority of the current process involves no software which hinders the process. With the School of Computer Science and Informatics encouraging a change in the process to offer a more cross-curricular view of degree programmes, the development of an automated software system enables the client to establish this view. Successfully developing a system that automates from the module review and BCS accreditation criteria source documents to the interface website makes the information more accessible and massively speeds up the review process for the client. As the main objective I was very happy with how this had turned out.

While the interface is minimal and uncluttered to allow the user to focus on the important information throughout, it also maintains an aesthetical appeal with the use of the familiar Cardiff university colour scheme. I feel that the overall website design could have been more polished and professional looking. This was due to the fact I did not have much experience or confidence in the design of interfaces and websites.

From a technical view, the system created is of a good standard. It performs all the tasks needed automatically on the date it is scheduled to run and ensures that the client is kept in the loop on successful and unsuccessful runs through the SQL Server Agent emails. The data is also able to be queried efficiently by the user through the easy to navigate website. The database and system being designed to enable both automated and manual changing of data ensure that if the client doesn't want to run the full system again that they can make minor tweaks ensures that the system is uncomplicated and straightforward. I also feel that the development of instructional documentation on how to use and alter certain paths and code(s) within the system is particularly useful for the client. This also shows I have taken initiative to ensure all of the clients' needs are taken into consideration.

Overall, I feel this project provides the client and other staff within the School of Computer Science and Informatics with an automated process and an easy to use interface allowing the holistic cross-curricular view - which was not possible with the Schools current process. I feel that this project has been both challenging but extremely beneficial to myself. I also feel confident that the design decisions I have made through the course of the development and implementation are justified and demonstrated by how efficiently and effectively the system works. I am therefore looking forward to hearing from the client how the process improvement has been felt when the system is scheduled to run in the next academic year!

Reflection

Research has concluded there are many benefits of curriculum mapping. These include increasing student achievement, providing flexibility for hosting and creating a resource centre for staff members by enabling the website to be reached intra-organisationally to all staff at the same time.

Due to the recent outbreak of COVID-19, I felt it was harder to achieve feedback from testers. This was due to the fact I had planned to carry out partial testing by observing how users interacted with the web-front end. This therefore affected my understanding of the usability of the product especially with my client. However, this must also be contrasted with the Hawthorne Effect which dictates that the act of observing how someone does something can change their approach to carrying out the task. Nevertheless, to combat the lack of observation testing I was able to instead carry out heuristic evaluations using members of my household.

Contrarily, I also believe that the COVID-19 outbreak had a positive effect on how I have carried out work within this project. Despite working remotely, I learnt to overcome any distractions present in my household when during such times I would usually visit the university library to focus. This also developed other personal skills as being out of a normal academic routine compelled me to show higher levels of discipline and determination. These in turn allowed me to develop into a more self-motivated person which increased my productivity.

Having used many different coding languages throughout the development of this project including HTML, CSS, PHP, SQL and C#, has greatly increased my knowledge and proficiency in the languages. This has in turn greatly expanded my grasp on the whole software development process and increased my familiarity with the agile methodology which I feel will benefit me greatly in any future development processes.

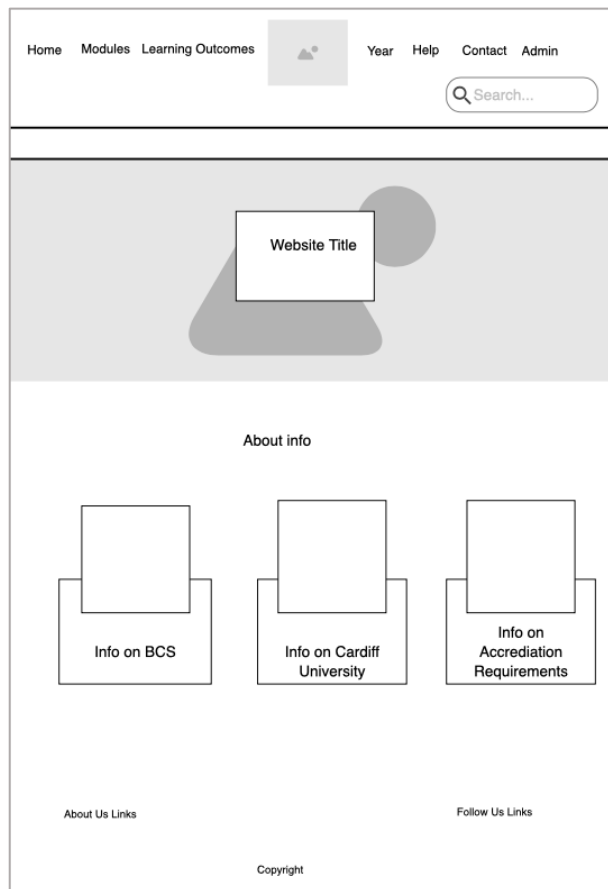
Relying on the university VPN for access to university servers has also been unpredictable, which caused certain workdays to be less productive due to slower internet speeds. However, despite this I felt I was still able to stick loosely to my intended work plan with each iteration taking around or just over 2 weeks to complete. While my time management skills were satisfactory before, I noticed over the course of my project that my time management skills greatly improved with working as a single developer rather than part of a team.

I feel I underestimated the part of the project involving matching the module learning outcomes with the accreditation criteria. As this couldn't be tested until I had the database tables populated from the document input and web front-end working – this left the matching until quite late in the project for such an important feature to the client. This feature was also hard to develop and therefore if I was to go back and change things about how I carried out this project I would schedule in more time to work on this part or begin development work on this specific part earlier. Unfortunately, this requirement was the one not met but this did not detract from the usefulness of the system.

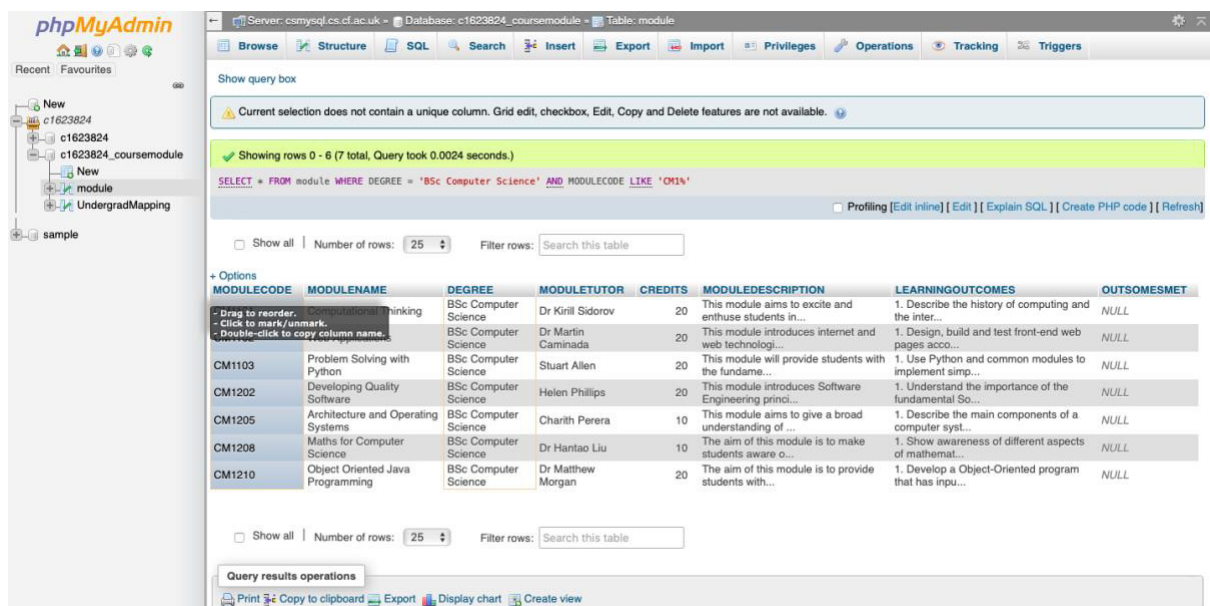
As a result of this project I feel I have developed my skills in eliciting the necessary information from a client to ascertain their exact requirements and other information. I found this challenging at times as my client used educational terminology which I was not familiar with, also they had domain knowledge that I did not have. By gaining knowledge and becoming familiar with the terminology I was able to gain a greater insight and understanding of the process and therefore feel I was able to offer a more optimal solution to the client. I am confident these skills will assist me throughout future projects.

I believe that this project has taught me the importance of communication with any stakeholders in a project. Making assumptions can often lead to solutions that may not necessarily fit the users' needs, so this project really made me aware the importance of clarifying anything that you're unsure about. Having all the information you need from the client at the start is also extremely important as it ensures that you start with a solid foundation and understanding.

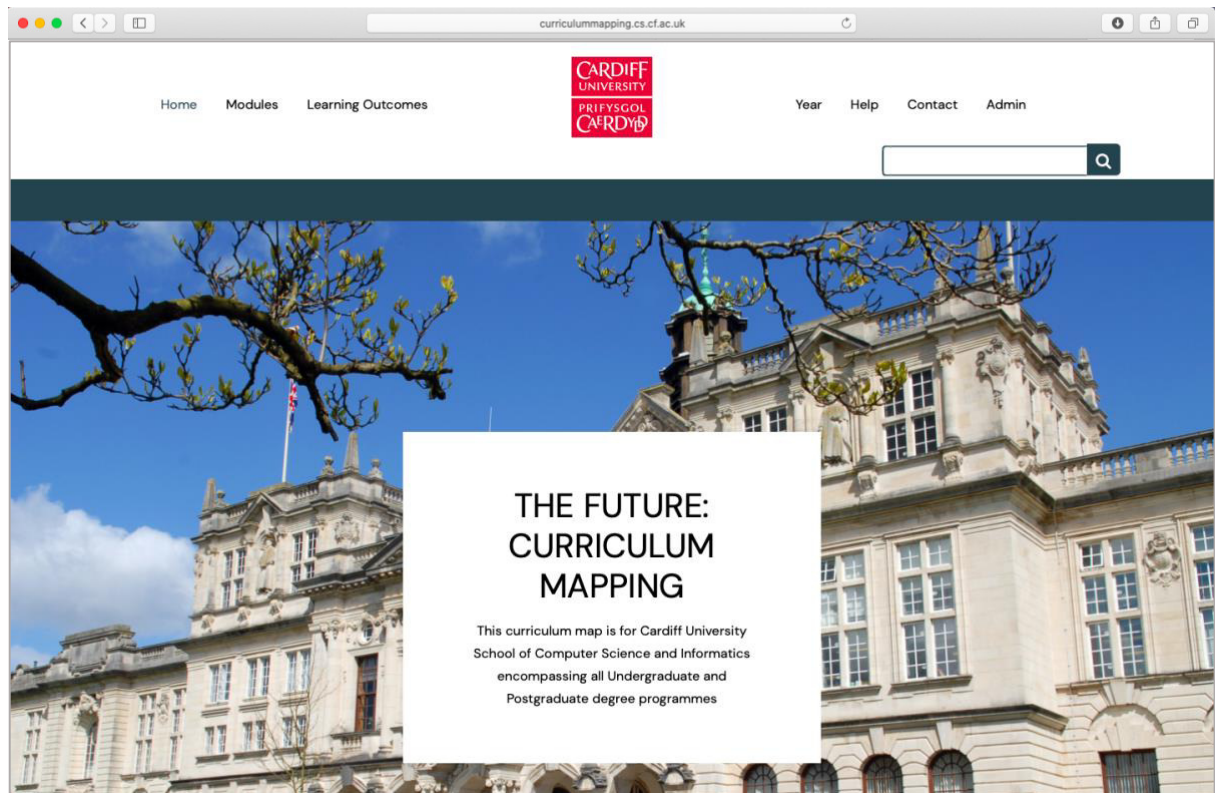
Appendices



Appendix 1: Web app home page mock-up



Appendix 2: PhpMyAdmin screenshot showing populated table



Appendix 3: Screenshot of home page of website/interface.

References

- Biggs, J.B. (2003). Teaching for Quality Learning at University. *The Higher Education Academy*. [Online]. pp1. Available at: https://www.researchgate.net/profile/John_Biggs3/publication/255583992_Aligning_Teaching_for_Constructing_Learning/links/5406ffe70cf2bba34c1e8153.pdf. [Accessed: 14th April 2020].
- Biggs, J.B. (2014). Constructive Alignment in University Teaching. HERDSA Review of Higher Education. [Online]. 1(7). pp5-22. Available at: https://tru.ca/_shared/assets/herdsa33493.pdf. [Accessed: 22nd April 2020].
- Bloudin, D., Tekian, A., Kamin, C. and Harris, I. B., (2018). "The impact of accreditation on medical schools' processes." *Academic Medicine*. [Online]. 52(2). pp182-191. Available at: <https://doi.org/10.1111/medu.13461> [Accessed: 24th March 2020].
- Boer, G. (2014). What is Scrum. *Microsoft*. [Online]. Available at: <https://docs.microsoft.com/en-us/azure/devops/learn/agile/what-is-scrum> [Accessed: 29th January 2020].
- British Computer Society. (2020). *Guidelines on course accreditation*. [Online] Available at: <https://www.bcs.org/deliver-and-teach-qualifications/university-accreditation/> [Accessed: 24th January 2020].
- Diamond, R.M. (1998) Designing and assessing courses and curricula: A practical guide. 2nd edn. San Francisco, CA: Jossey-Bass, Inc.
- Elentra. (2020). *Elentra Consortium*. Available at: <https://elentra.org> [Accessed: 26th March 2020].
- International Organization for Standardization. (2013). Usability of consumer products and products for public use – Part 2: Summative test method. *ISO Online Browsing Platform*. [Online]. Available at: <https://www.iso.org/obp/ui/#iso:std:iso:ts:20282:-2:ed-2:v1:en>. [Accessed: 25th March 2020].
- Jacobs, H. H. (2010). *Curriculum 21: Essential Education for a Changing World*. Alexandria, VA: ASCD.
- Kidd, B. (2010). *COMPAS*. Barts and The London, School of Medicine and Dentistry. Available at: <https://compas.smd.qmul.ac.uk> [Accessed: 23rd January 2020].
- Langmayer, M. (2019). 10 Usability Heuristics Every Designer Should Know. *UX Collective*. [Online]. Available at: <https://uxdesign.cc/10-usability-heuristics-every-designer-should-know-129b9779ac53>. [Accessed: 26th March 2020].
- Larman, C. (2013). *Agile and Iterative Development: A Manager's Guide*. 1st edn. Boston, MA: Addison-Wesley.

Masters, T. (2018) *Spreadsheets vs Databases*. [Online] Available at: <https://smallbusiness.chron.com/spreadsheets-vs-databases-53907.html> [Accessed: 23rd January 2020].

Mattern, W. D., Anderson M. B., Aune K. C., Carter D. E., Friedman, C. P., Kappelman, M. M. and O'Connell, M. T. (1992). "Computer Databases of Medical School Curricula". *Academic Medicine*. [Online]. 67(1). pp 12-16. Available at: <https://doi.org/10.1097/00001888-199201000-00003> [Accessed: 1st February 2020].

Merrill, M.D. (2002) First Principles of instruction. *Education Technology Research and Development*. [Online]. 50(3). pp43-59. Available at: <https://link.springer.com/article/10.1007/BF02505024>. [Accessed: 23rd January 2020].

Microsoft. (2018). SSIS How to Create an ETL Package. *Microsoft*. [Online] Available at: <https://docs.microsoft.com/en-us/sql/integration-services/ssis-how-to-create-an-etl-package?view=sql-server-ver15>. [Accessed: 27th March 2020].

Parker Software. (2019). What is a user story and why should you be writing them?. *Parker Software*. [Online]. Available at: <https://www.parkersoftware.com/blog/user-story-writing/>. [Accessed: 15th April 2020].

Ramsden, P. (1992). Learning to teach in higher education. *Studies in Higher Education*. [Online]. 18(1). pp 290. Available at: <https://doi.org/10.1080/03075079312331382498>. [Accessed: 15th March 2020].

Salas, A. A., Anderson, M. B., LaCourse, L., Allen, R., Candler, C. S., Cameron, T. and Lafferty, D. (2003). "CurrMIT: a tool for managing medical school curricula." *Academic Medicine*. [Online]. 78(3). pp275-279. Available at: <https://doi.org/10.1097/00001888-200303000-00009> [Accessed:24th March 2020].

Taylor, R. and Canfield, P. (2007). Learning to be a scholarly teaching faculty: cultural change through shared leadership. In A. Brew and J. Sachs (Eds.) *The Transformed University: Scholarship of Teaching and Learning in Action*. Pp233-247. Sydney: Sydney University Press.

The Glossary of Education Reform. (2014). *Curriculum Mapping*. [Online] Available at: <https://www.edglossary.org/curriculum-mapping/> [Accessed: 23rd March 2020].

University of Liverpool. (2019). University spin-out acquired by US exam software company. *University of Liverpool News*. [Online] Available at: <https://news.liverpool.ac.uk/2019/03/25/university-spin-out-acquired-by-us-exam-software-company/> [Accessed: 25th March 2020].