**Improving Quality Monitoring**

**Processes Around Modules and**

**Coursework**Final report

Submitted for the BSc in

Computer Science

April 2021

by

James Anthony Cairns

Word count: 7467

Abstract

An inefficient and outdated system is currently in place at the University of Hull that is designed to quality monitor the processes around modules and coursework (Systems Analysis and Design, 2019). This means the system monitors the content of a module and its coursework to ensure that the content and coursework is relevant and not outdated or irrelevant to the subject.

An efficient system should contain good UI/UX design for the system to help encourage user interaction as well as help users to feel like technology in the workplace is not daunting. It should be able to inform of upcoming deadlines with a traffic light system to ensure work is done on time and keep the users time aware.

Although there are options available there is not one tailored to this specific workplace/environment that is not messy and overflowing causing the user to be stressed from over stimulation.

The process of this will be covered in the report as a new structure is formed to replace the current system and provide options in which the new system can be built in. There will also be discussion about how UI/UX design can help to encourage and motivate user interaction and how technology is impacting the workplace.

Contents

[Abstract i](#_Toc70275767)

[1 Introduction 3](#_Toc70275768)

[1.1 Background to the project 3](#_Toc70275769)

[1.2 Aims and objectives 4](#_Toc70275770)

[1.2.1 Understanding of the Current System 4](#_Toc70275771)

[1.2.2 Understanding of UI/UX Design 4](#_Toc70275772)

[1.2.3 Understanding of Technology and its Impact on the Workplace 4](#_Toc70275773)

[1.3 Research question 4](#_Toc70275774)

[2 Requirements 5](#_Toc70275775)

[2.1 Product requirements 5](#_Toc70275776)

[2.1.1 Current System 5](#_Toc70275777)

[2.1.2 Epics (Systems, Analysis and Design, 2019) 7](#_Toc70275778)

[2.2 Functional requirements 8](#_Toc70275779)

[2.2.1 Functional Capabilities 8](#_Toc70275780)

[2.2.2 Data Structures/Elements 8](#_Toc70275781)

[2.2.3 Constraints and Limitations 8](#_Toc70275782)

[3 Design 9](#_Toc70275783)

[3.1 Software design 9](#_Toc70275784)

[4 Implementation and testing 12](#_Toc70275785)

[4.1 Implementation Tools and Techniques Used 12](#_Toc70275786)

[4.1.1 Tools 12](#_Toc70275787)

[4.1.2 Techniques 12](#_Toc70275788)

[4.2 Iteration 1 12](#_Toc70275789)

[4.2.1 Features worked on 12](#_Toc70275790)

[4.2.2 Testing 13](#_Toc70275791)

[4.2.3 Evaluation for next iteration 13](#_Toc70275792)

[4.3 Iteration 2 13](#_Toc70275793)

[4.3.1 Features worked on 13](#_Toc70275794)

[4.3.2 Testing 14](#_Toc70275795)

[4.3.3 Evaluation for next iteration 14](#_Toc70275796)

[4.4 Iteration 3 14](#_Toc70275797)

[4.4.1 Features worked on 14](#_Toc70275798)

[4.4.2 Testing 14](#_Toc70275799)

[4.4.3 Evaluation for next iteration 14](#_Toc70275800)

[5 Evaluation 15](#_Toc70275801)

[6 Conclusion 16](#_Toc70275802)

[References 17](#_Toc70275803)

[Appendix A – Interesting but not vital material 18](#_Toc70275804)

# Introduction

The University of Hull currently has an inefficient and outdated system for quality monitoring the processes around modules and coursework (Systems Analysis and Design, 2019). This means the current system they have in place is causing issues such as inefficiency and timing when it comes to producing content and coursework for their modules.

## Background to the project

The university’s current QM software needs replacing due to it being something designed before large advancements have been made in technology within the workplace. This meant that the university had to deal with some very basic, inefficient software but now as time has come along there have been positive impacts of technology in the workplace.

Over the years, the world has been experiencing rapid advancements in technology. These advancements in technology have changed how people live, how businesses execute their core processes, and how people travel and communicate. The workplace has not been left behind. The adoption and integration of technology has changed how workers across industries perform their daily duties. In the workplace, technology has helped streamline business processes. As a result, workers’ productivity has significantly improved, leading to higher revenue generation in businesses. (Cascio & Montealegre 2016). The adoption and integration of technology in businesses is beneficial for the workplace in different ways. Some of the ways include the following: -

**Coordination and collaboration**

With technology, coordination and collaboration in the workplace is easier. There are online collaboration tools such as Zoom that organizations use to collaborate and discuss important business matters without physical meetings. (Cascio & Montealegre 2016). Additionally, there are numerous technology-driven tools, such as cloud-computing applications that allow employees to work remotely. Such tools make the workplace more efficient. The need for large office spaces has also reduced since workers can work remotely.

**Improved speed and efficiency in the workplace**

The adoption and integration of technology in the workplace has led to the automation of repetitive tasks. Automation helps improve the speed of business process execution. It also helps improve productivity among employees. They can perform their duties faster with minimal challenges. (Cascio & Montealegre 2016). They can also concentrate on core business processes since repetitive tasks get automated. As a result, workers become more efficient.

Something else that has helped to improve the speed and efficiency in the workplaces has been the advancement of UI/UX design. This has been a big impact to help encourage user interaction and help to motivate them into working more efficiently.

Motivation is the force that gives people motive, according to the Merriam-Webster dictionary; motive is something- such as need or desire- that causes a person to act (Definition of MOTIVE (2020)). Motivation appears on the crossroads of biological, cognitive, social and emotional factors that all together influence its strength and behaviour it stimulates. It is important for designers to understand the types and mechanisms of motivation because it shows the direct ways to make a product correspondent to users’ expectations and solving users’ pain. There are two major directions of how motivation evolves, and they deeply influence the decision-making process. Extrinsic motivation is reward driven behaviour, reward and other incentives- like praise, money or fame- are used as motivation for specific activities. Intrinsic motivation is doing something without any obvious external rewards; you do it because it is interesting rather than because of an outside incentive or pressure to do it. (Yalanska, 2017)

Several factors can contribute to the success of a platform or application, including the degree to which it entertains, informs, and teaches relevant information to the intended user. Design – both user interface (UI) design and, more broadly, user experience (UX) design – can be the deciding factor for the level of impact an application or platform has.

To maximize the effectiveness of applications and platforms, the user focus must also be extended to UX and UI design. For an application or platform’s UI to be effective, it must have met the following five criteria:

1. Easy for users to navigate

2. Provides continuous feedback to keep users encouraged and engaged

3. Evokes happy and playful emotions

4. Unique, attractive, and eye-catching visuals

5. Appropriately targeted to the intended age group

## Aims and objectives

### Understanding of the Current System

Gain an understanding of what the current system is like and generate suggestions on how to make the software more efficient.

### Understanding of UI/UX Design

Gain an understanding of how the UI/UX can help to increase user interaction by motivating the user in different ways.

### Understanding of Technology and its Impact on the Workplace

Gain an understanding of how technology has impacted the workplace. With things constantly evolving and more industries implement technology into their workplace, the more productivity is occurring (Cascio & Montealegre 2016).

## Research question

Speaking to students in their final year of their degree in computer science at the University of Hull, they have complained about overlapping of release dates and deadlines along with other things such as repetition over previous years.

# Requirements

UI and UX Design are important processes that should be considered in the development of usable and interactive software. User Interface Design aims at improving the appearance of a software. (Saha et al. 2015). For example, designers choose colour schemes, button shapes, and other visual elements during User Interface Design to ensure that the overall theme of the software matches its purpose. Additionally, designers must ensure that all visual elements of the software feel aesthetically united. On the other hand, User Experience Design focuses on how a user interacts with interface elements. It involves ensuring that a user interface is intuitive, friendly, easy, and enjoyable to use. (Harutyunyan & Riehle 2019). Therefore, how difficult or easy it is for a user to interact with the user interface of software determines User Experience.

UI and UX Design play a significant role in encouraging users to interact more with software. Software designed with the nature of User Interface and User Experience in mind is more engaging and visually attractive. (Dharti 2019) Additionally, the software provides an easy to use navigation with a continuous flow of information. Also, User Interface and User Experience Design processes contribute to the development of user-friendly software. (Dharti 2019). These factors make software flawless and perfectly designed to create an excellent experience that encourages users to interact more with software.

It is necessary to consider both UI and UX Design in software design since they are independent. (Dharti 2019). An application with an excellent user interface but poor User Experience is less attractive and less usable. If a software is designed with User Experience in mind but with poor User Interface, it will also be less usable. Therefore, considering both User Interface and User Experience ensures that software is usable, engaging, and visually attractive, encouraging users to interact more with the software.

## Product requirements

### Current System

This is a flowchart of the current system that explains the rather complex process of jumping through many hoops to get approval for coursework.

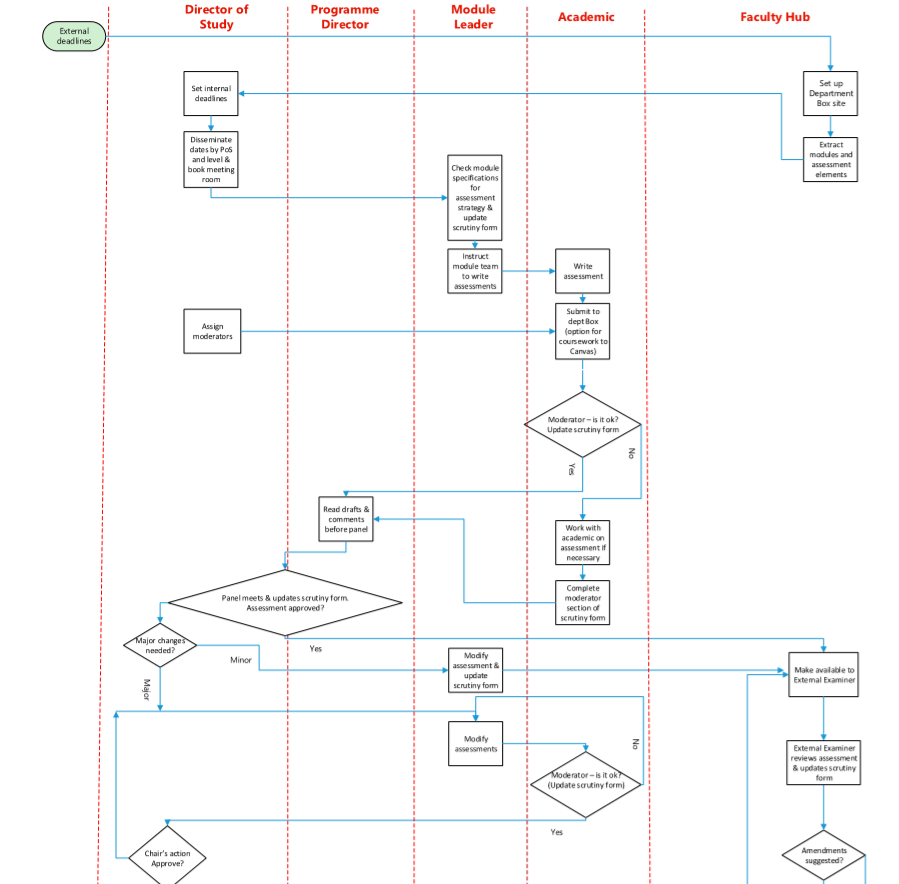


Figure - 1st Half of the flow chart (Systems Analysis and Design, 2019)

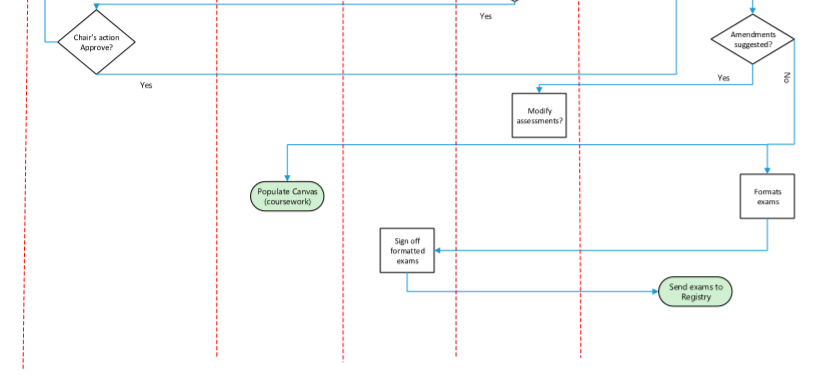


Figure - 2nd Half of the flow chart (Systems Analysis and Design, 2019)

### Epics (Systems, Analysis and Design, 2019)

**Director of Studies**

* Confirm that Moderators have been assigned
* set dates for:
  + the assessment scrutiny panel (either one per department or may be split by level if preferred)
  + deadlines for assessment setters to make drafts available to moderators on Box
  + deadline for ‘moderator approved’ assessment to be made available to the Panel on Box
  + deadlines for making amendments post-panel (where necessary)
* request support staff to book meeting room(s)
* request support staff to administer the panel
* post-panel, take Chair’s action on assessment tasks which required major amendment

**Module Leader**

* Instruct module teams to write assessments (having checked the assessment strategy on the module specification on SharePoint)
* complete Module Leader section of the scrutiny form

**Module Teams**

* Write the assessments in coordination with the Module Leader
* make corrections in coordination with moderators and Scrutiny Panel.

**Moderator**

* Review the draft assessment tasks adhering to the criteria in the scrutiny form
* complete Moderator section of the scrutiny form
* discuss any areas requiring amendment with the setters where necessary
* provide setters with suggestions for future years where appropriate (e.g. alternative assessment methods to consider)
* post-panel, review any amended assessment tasks following major modification

**Panel**

*Panels consist of Director of Studies (Chair), Programme Directors plus any additional staff as determined by the Director of Studies.*

*The Panel’s role is to consider all assessment tasks for the trimester for each level and:*

* check that assessments are appropriate for the module and programme
* check that moderation has been conducted
* review that assessment deadlines are appropriate for the programmes
* make any recommendations for review of assessment strategies for future sessions
* complete the panel section of the scrutiny form.

**Panel Administrator**

* As soon as the deadline for ‘moderator approved’ assessments has passed, inform the Director of Studies (cc Head of Department) of any missing assessments
* coordinate the outcomes of the Panel
* inform assessment setters where amendments are required (major/minor)
* coordinate Chair’s action post major amendment
* liaise with Hub regarding assessments ready for scrutiny by the External Examiner

## Functional requirements

### Functional Capabilities

A traffic light system to inform the user of which deadlines are currently outstanding and are nearing their submission date.

CRUD functionality to create, read, update, and delete coursework. This functionality will help to create a sort of modular way of programming.

### Data Structures/Elements

|  |  |
| --- | --- |
| **Views** | **Users** |
| * Module Leader * Moderator * Panel | * Module Leader * Module Checker |

**Classes**

* User\*
* Coursework\*
* ModuleLeader\*
* ModuleChecker\*
* CourseworkGroup
* Module
* CourseworkElement
* Review

|  |  |  |
| --- | --- | --- |
| **Coursework** | **User** | **Coursework** |
| * ID – int * Module – Module * Elements – List of type coursework element * ReleaseDate – DateTime * SubmissionDate – DateTime * Review - Review | * ID – int * Email – string * Password -string * Surname – string * Forenames - String | * ID – int * Module – Module * Elements – List of type coursework element * ReleaseDate – DateTime * SubmissionDate – DateTime * Review - Review |
| **Deliverable** | **Review** | **Module** |
| * ID – int * Type - int * SubmissionDate - DateTime | * ID – int * courseworkId - int * reviewerId – int | * ID - int * name – string * courseworkIds – List<int> |

### Constraints and Limitations

The biggest constraint was the time feasibility and meeting deadlines. This was a huge factor in some of the decision making, for example integration with canvas did not get achieved as there was no time to meet this requirement.

# Design

UI and UX Design are important processes that should be considered in the development of usable and interactive software. User Interface Design aims at improving the appearance of a software. (Saha et al. 2015). For example, designers choose colour schemes, button shapes, and other visual elements during User Interface Design to ensure that the overall theme of the software matches its purpose.

## Software design

|  |  |
| --- | --- |
| **Views** | **Users** |
| * Module Leader * Moderator * Panel | * Module Leader * Module Checker |

**Classes**

* User\*
* Coursework\*
* ModuleLeader\*
* ModuleChecker\*
* CourseworkGroup
* Module
* CourseworkElement
* Review

|  |  |  |
| --- | --- | --- |
| **Coursework** | **User** | **Coursework** |
| * ID – int * Module – Module * Elements – List of type coursework element * ReleaseDate – DateTime * SubmissionDate – DateTime * Review - Review | * ID – int * Email – string * Password -string * Surname – string * Forenames - String | * ID – int * Module – Module * Elements – List of type coursework element * ReleaseDate – DateTime * SubmissionDate – DateTime * Review - Review |
| **Deliverable** | **Review** | **Module** |
| * ID – int * Type - int * SubmissionDate - DateTime | * ID – int * courseworkId - int * reviewerId – int | * ID - int * name – string * courseworkIds – List<int> |

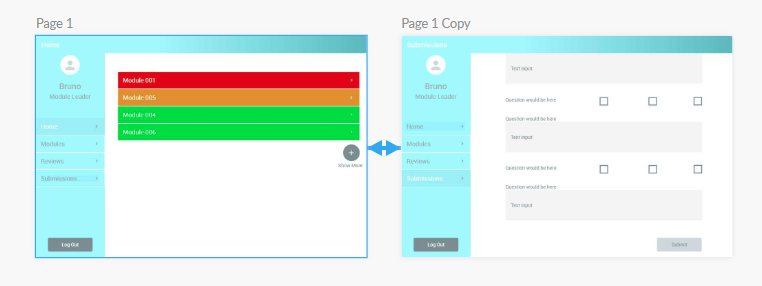


Figure - Small Prototype Design with Traffic Light System

Figure 3 shows a small prototype design made with simplicity for ease of mind as well as simple colours to help attract the user’s attention to where it needs to be. The blue should help to give a user a calm and relaxed feeling when the page is lacking other bright colours. I also chose the blue as it should match the one used with University of Hull’s latest evision software colour scheme.

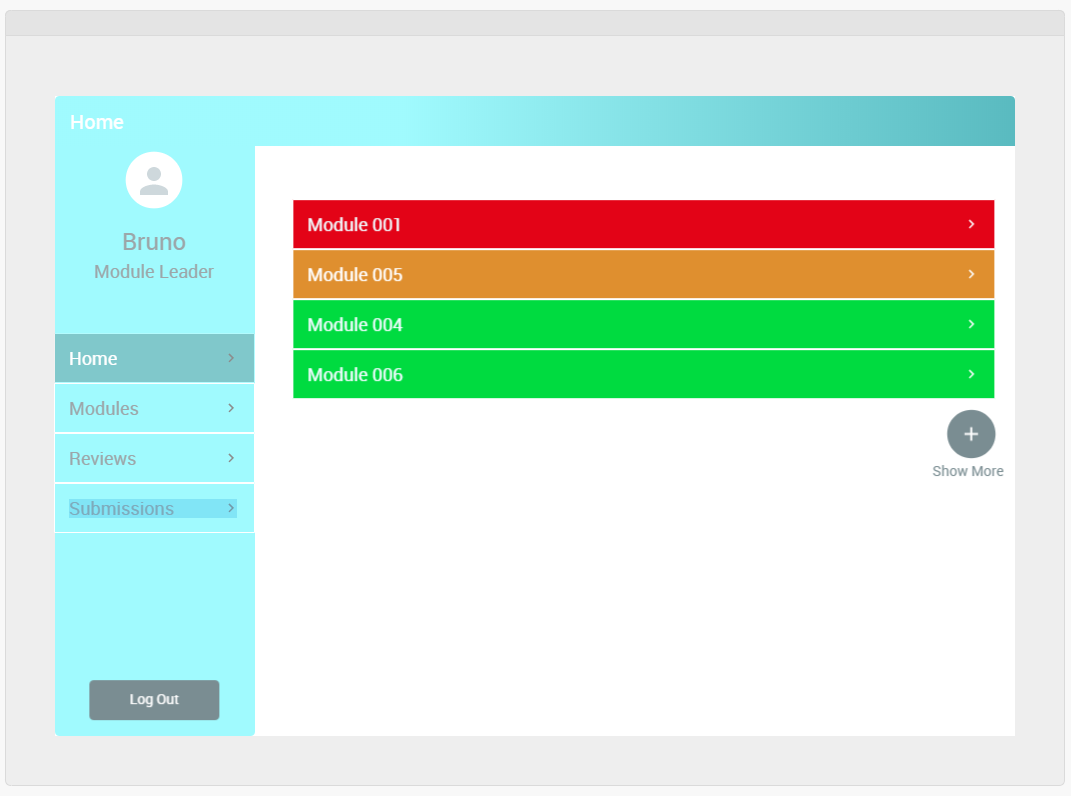


Figure - Home page of prototype

In figure 4 you can see the display of the traffic light system in place. A red module would signify the need for urgent attention whereas an orange/amber colour is used for one that is approaching but does not require urgent attention. The green modules signify that there is plenty of time left and there are no worries needed. The colour red is rather intense and is known to signal caution or danger which should interact with the user motivating them to pay more attention to the urgency.

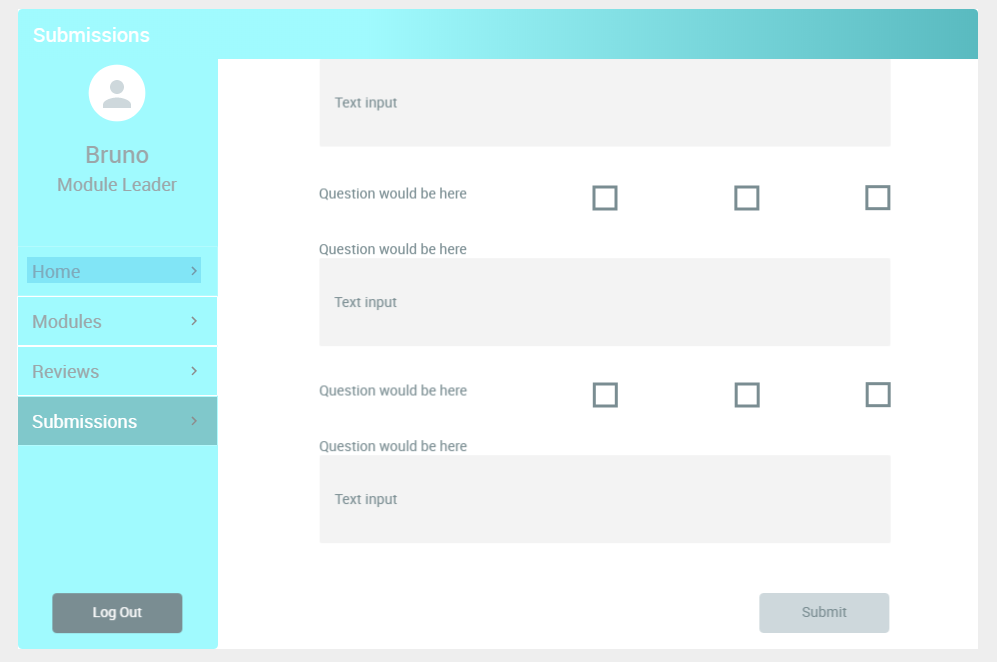


Figure - Submissions page of prototype

As Displayed in figure 5 there would be an online form rather than a separate document in which would be filled out and submitted for approval. This is what will form a submission for a reviewer to review and provide feedback on.

When it comes to UI design these factors must be considered:

1. Selecting a colour scheme is very important and so sampling these and testing how well they work together is important.

2. Look at other designs of similar systems to help inspire the design of the project.

3. Prototyping of designed screens and improve on them until satisfied with the result.

4. Implement the actual design into the placeholder for the website/system.

# Implementation and testing

Implementation needs to be done iteratively, testing with each iteration, and ensuring that each feature is done via the correct priority.

The process of the project is in 5 main steps.

1. Planning and Requirements.

2. Analysis and Design.

3. Implementation.

4. Testing.

5. Evaluation.

This is a rather efficient software development life cycle for this project as it allows a working copy to be shown to a stakeholder for feedback to assist in the changes for the next iteration as the cycle repeats. The life cycle helps to give rapid turnaround and easy adaptability which helps to have a working system a lot faster. (Iterative Model: What Is It And When Should You Use It? N.d)

## Implementation Tools and Techniques Used

### Tools

* + Visual Studio – this IDE is best as development is in C# using MVC ASP.NET.
  + MSSQL Server – this is where the database is hosted.
  + C# - Development language for the MVC ASP .NET Architecture.
  + GitHub – this is a version control system to help keep track of changes as well as to make workflow easier. It is a “system that records changes to a file or set of files over time so that you can recall specific versions later.” (Git - About Version Control, 2021).

### Techniques

* + Iterative development – an aspect of agile where each feature is completed then reviewed then move on to the next development and so on.
  + MVC ASP .NET architecture – this is the programming architecture used to develop the software system.
  + CRUD – (Create, Read, Update, Delete) “The CRUD paradigm is common in constructing web applications, because it provides a memorable framework for reminding developers of how to construct full, usable models.” (What is CRUD? | Codecademy, 2021).

## Iteration 1

In the first iteration of the project the important goal was to create a simple database and simple software system with CRUD functionality. Once this was implemented it provided a strong and solid foundation for the project.

### Features worked on

The features worked on in this iteration were the creation of the database (which would later be modified) as well as creating simple CRUD functionality to help give a strong base of what will be made later.

#### Database

With the software being done as database first over code first I needed to start by making a simple database. I decided to go for a very simple database which contained 1 table to start with (Staff) and then in future iterations the database can be expanded further.

To create the database, I opened Microsoft SQL Server and started a new database called ‘QualityMonitoringSystem’. The database was being kept as local for the purpose of creating a working demo and when everything was ready for publishing it would then be hosted elsewhere.

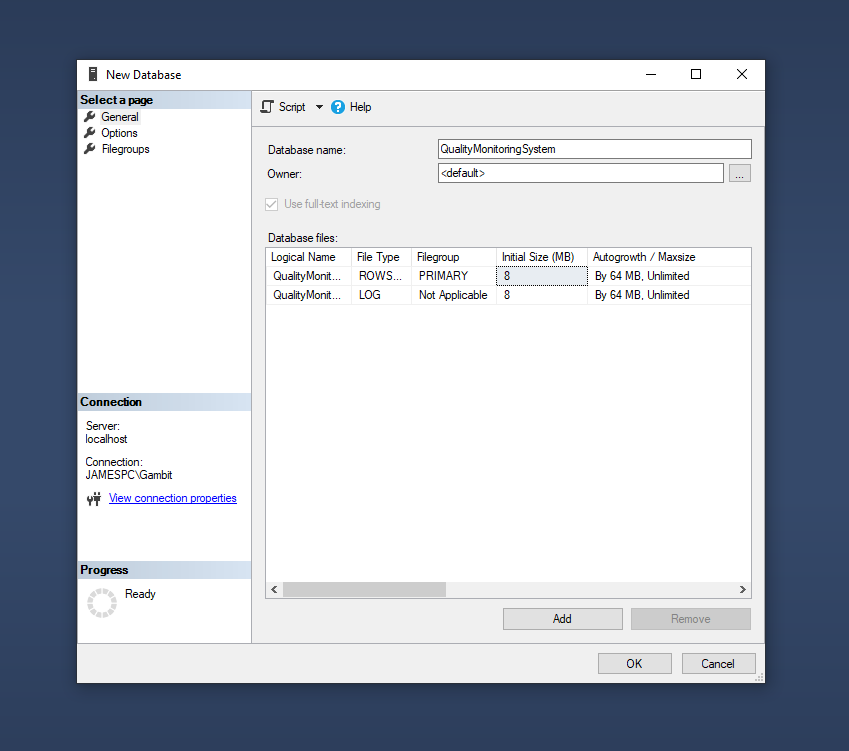


Figure Screen grab of the database being created

Once the database was set up, I then went on to create a table for the Staff by using the design for the User table shown in the design section. Staff seemed like a better name to set for the table over User as it made the system a bit more comprehendible.

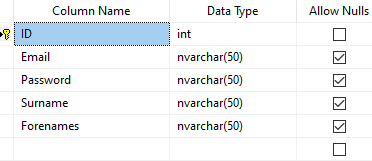


Figure Screen grab of the Staff table opened in design mode

Once the database was set up I added in a couple test entries to ensure it was working ok (as shown in figure 8 below).

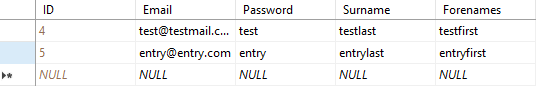


Figure Test entries in the Staff table

#### Software

Once I had a database with a table, I could then start creating a CRUD system for the user to interact with the database. Because I was working database first, I could automatically make the models for the system by creating an entity data model that connects to the database.

“The Entity Data Model (EDM) is a set of concepts that describe the structure of data, regardless of its stored form. The EDM borrows from the Entity-Relationship Model described by Peter Chen in 1976, but it also builds on the Entity-Relationship Model and extends its traditional uses.” (Entity Data Model - ADO.NET, 2021).

With the database set up with the Staff table I then created a new ASP.NET web application with the .NET Framework 4.7.2 (as shown in figure 9 below)

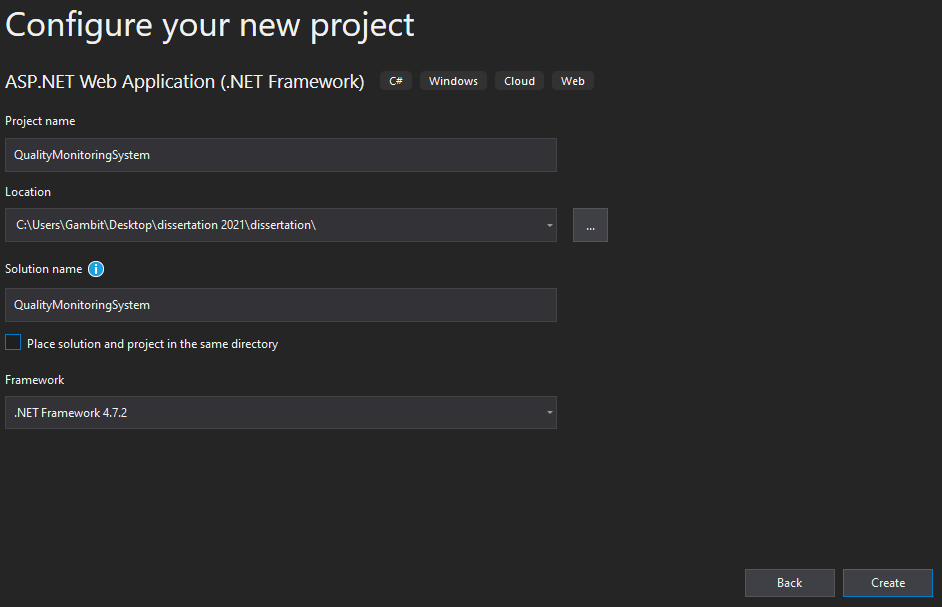


Figure Screen grab of the project being set up in Visual Studio 2019 Community Edition

With the project now set up and ready to start I then go on to creating an Entity Data Model which I connected to the database I had just made (as seen in figure 10). This was done to link the database and the software together.

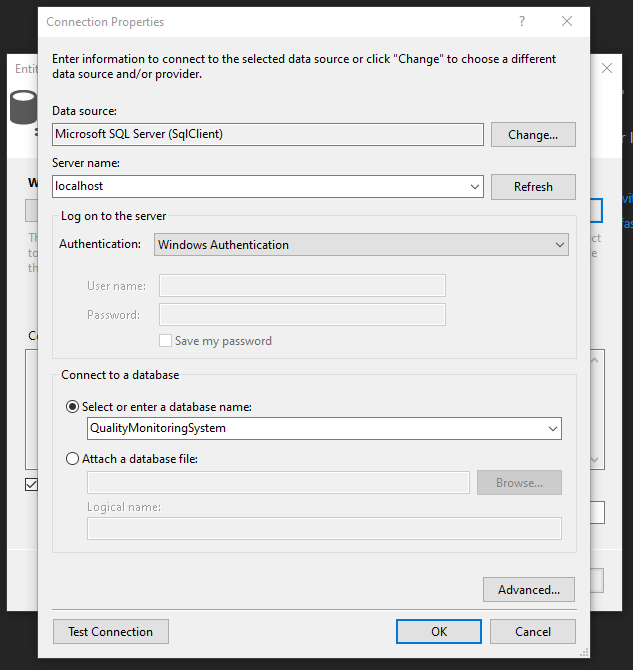


Figure Screen grab of the entity data model being linked to the database

Once the entity data model was linked it then generated a model file for the Staff table in the database. This made things easier for me as it meant I did not have to work on making the Staff model myself. This could have led to human error causing problems for the system’s production.

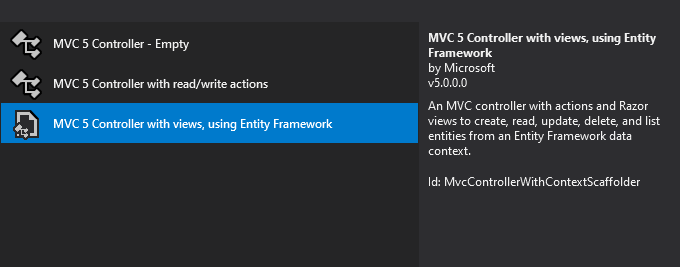


Figure Screen grab of a controller with context scaffolder

As shown in figure 11 above, when I created a controller for the Staff model it generates views as well as the CRUD ability within the controller itself. All the methods to load a page are handled with GET requests as they will be getting the views and loading them in. The POST requests will be sending the data taken from any input the user gives to the system (such as editing an entry) and will send the data to the database either creating, updating, or deleting records. An example of a GET and POST method can be seen in figure 12 below.

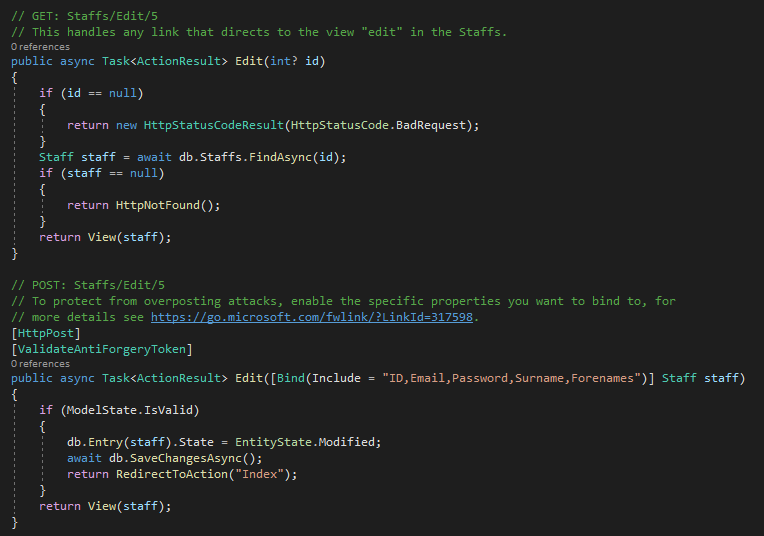


Figure An example of GET and POST methods that will handle the Update functionality out of the CRUD paradigm

### Testing

#### Software test

|  |  |  |  |
| --- | --- | --- | --- |
| **Hypothesis** | **Expected Result** | **Actual Result** | **Evaluation** |
| Using the CMS submit a new staff member. | New staff member is added to the database and is displayed on the index page | New staff member was added to the database and is displayed on the index page | This test passed. |
| Using the CMS edit the new staff member to have 1s in each field. | Staff member gets 1s put into each field and is saved to the database. | Staff member now contains 1s in each field and is save to the database | This test passed. |
| Using the CMS view the details of the staff member. | It opens the page to display the staff’s details. | The details page was opened and correctly displayed the staff’s details | This test passed. |
| Using the CMS delete the latest staff member from the database. | The latest staff member should be deleted from the database and should not be visible in the CMS table. | The latest staff member was deleted from the database and is no longer visible in the CMS table. | This test passed. |

### Evaluation for next iteration

The tests passed and acted as a great base foundation for both the database and the software to continue being worked on in the next iteration. Once it was all successfully tested and evaluated everything that was done was then committed and pushed to the main branch of the git repository.

## Iteration 2

In the second iteration of the project the new tasks to complete were:

* Additional tables are added to the database:
  + Coursework
  + Modules
* The entity data model (the edmx file) is updated
* CRUD functionality for both the coursework and module tables that are added to the database.

### Features worked on

The features worked on in this second iteration of the system were the expansion of the database so that it would have multiple tables and not just one. On top of this the web application was changed to take the database changes into consideration as well.

#### Database

In the first iteration of the project the database was created as simply as possible with just a staff table. It has now been updated to contain a staff, module, and coursework table. Once this was done the edmx file was updated and shows the new tables and how they link together.

When making the Module and Coursework tables I needed to use a Boolean data type to create a true and false field, however, there is no Boolean data type in MSSQL and so I had to use the bit data type. The bit data type can be 1, 0 or null, however, 1 can be conveyed as TRUE and 0 can be conveyed as FALSE.

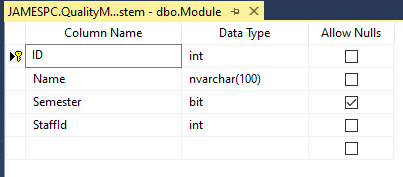
In the Module table I used the bit data type to handle the Semester field. I used TRUE to convey semester 1, FALSE to convey semester 2 and NULL to convey a module that spans over both semesters (This can be seen in figure 13).

Figure Design of the Module table

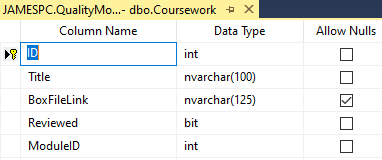
In the Coursework table I used the bit data to handle the field Reviewed. This field is set to not allow nulls so as to allow people to know if that coursework has been reviewed or not (This can be seen in figure 14).

Figure Design of the Coursework table

In both tables foreign keys were used to create relationships between two tables. For the Module table it linked with the Staff table. This allowed a 1(Staff) to many(Module) relationship be made. This links the 2 tables together allowing us to assign a particular staff to the module. The Coursework table it is linked with the Module table creating a 1(Module) to many(Coursework) relationship. An example of the foreign key relationship can be seen below in figure 15.

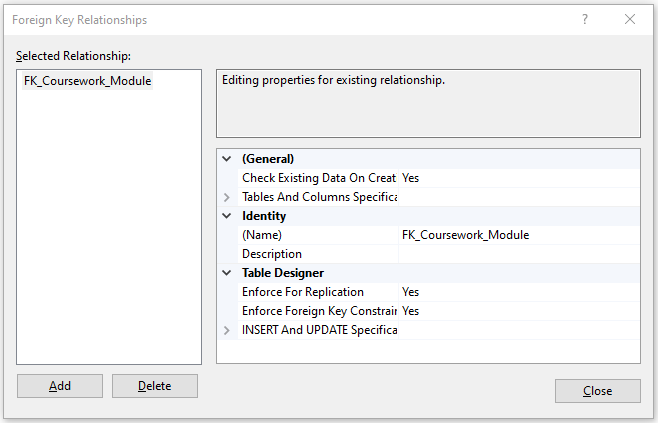


Figure Foreign key relationships in the coursework table

#### Software

Since the project was made as database first, I started to create the CRUD functionality for the new models. I could only start this once I had updated the edmx so that it would then generate the new models for me. This was all rather simple as it became just a factor of copy and pasting the modular functionalities to then fit the new models.  
Once the edmx was updated it brought in the other 2 models for the Coursework and Module tables. The edmx diagram can be seen in the figure below.

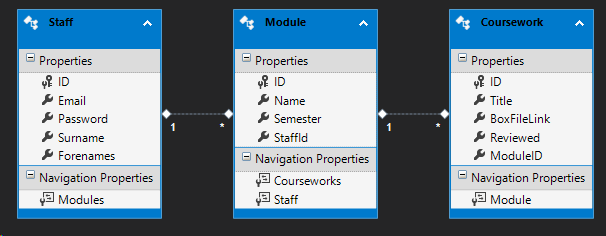


Figure EDMX diagram for the database

Once the edmx was updated I could create the controllers for the Coursework and Module models. I once again used the scaffold to help create the controllers and the views for the software. This helped to create the CRUD functionality for the software allowing us to create, edit and delete different coursework and modules just like with the staff. An example of one of the views can be seen in the figure below.

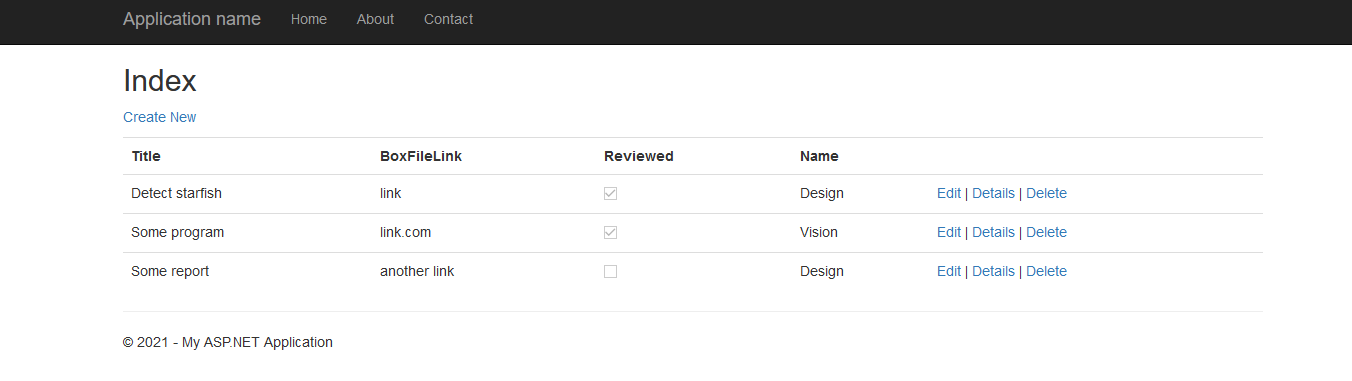


Figure Coursework view during iteration 2

The views that allow creating, editing, and deleting can be seen in the figure below. These show the views that people can see when they encounter the CRUD functionality. These views give the user easier control and readability to create, edit, and delete things in the database. This helps to make user interaction with the database through the system feel more natural and less complicated than entering directly into the database.

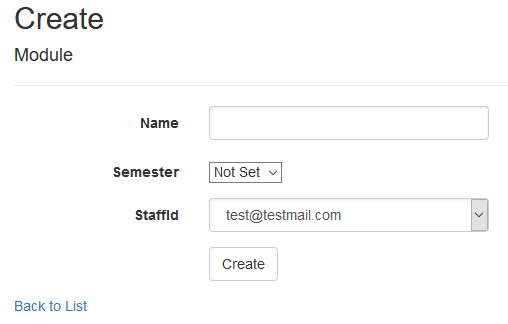
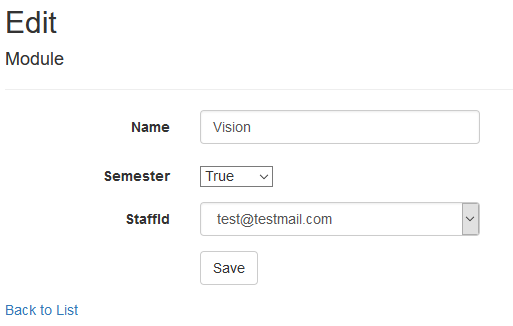
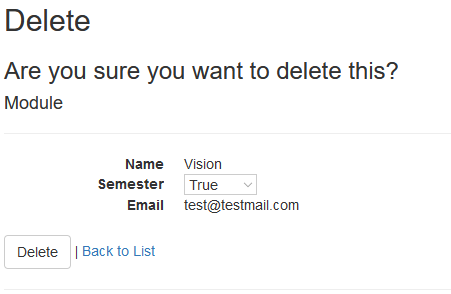


Figure Displaying the create, edit, and delete views for the Modules

### Testing

#### Software test

|  |  |  |  |
| --- | --- | --- | --- |
| **Hypothesis** | **Expected Result** | **Actual Result** | **Evaluation** |
| Using the CMS submit a new staff member. | New staff member is added to the database and is displayed on the index page | New staff member was added to the database and is displayed on the index page | This test passed again. |
| Using the CMS edit the new staff member to have 1s in each field. | Staff member gets 1s put into each field and is saved to the database. | Staff member now contains 1s in each field and is save to the database | This test passed again. |
| Using the CMS view the details of the staff member. | It opens the page to display the staff’s details. | The details page was opened and correctly displayed the staff’s details | This test passed again. |
| Using the CMS delete the latest staff member from the database. | The latest staff member should be deleted from the database and should not be visible in the CMS table. | The latest staff member was deleted from the database and is no longer visible in the CMS table. | This test passed again. |
| Using the CMS submit a new coursework. | New coursework is added to the database and is displayed on the index page | New coursework was added to the database and is displayed on the index page | This test passed. |
| Using the CMS edit the new coursework to have 1s in each field. | Coursework gets 1s put into each field and is saved to the database. | Coursework now contains 1s in each field and is save to the database | This test passed. |
| Using the CMS view the details of the coursework. | It opens the page to display the coursework’s details. | The details page was opened and correctly displayed the coursework’s details | This test passed. |
| Using the CMS delete the latest coursework from the database. | The latest coursework should be deleted from the database and should not be visible in the CMS table. | The latest coursework was deleted from the database and is no longer visible in the CMS table. | This test passed. |
| Using the CMS submit a new module. | New module is added to the database and is displayed on the index page | New module was added to the database and is displayed on the index page | This test passed. |
| Using the CMS edit the new module to have 1s in each field. | Module gets 1s put into each field and is saved to the database. | Module now contains 1s in each field and is save to the database | This test passed. |
| Using the CMS view the details of the module. | It opens the page to display the module’s details. | The details page was opened and correctly displayed the module’s details | This test passed. |
| Using the CMS delete the latest module from the database. | The latest module should be deleted from the database and should not be visible in the CMS table. | The latest module was deleted from the database and is no longer visible in the CMS table. | This test passed. |

### Evaluation for next iteration

The tests all passed with no changes required. The foundation of the project is now all laid out however the user interface is looking rather plain and boring and has nothing to entice the user to interact. With a strong foundation, a good design layout to use when moving forward with creating more pages would be in the best interest of the project. If the user is bored or struggling to navigate the web application, then the purpose of the software will not be achieved, and the productivity rate may faulter when the users interact meaning working on the front end may be best for iteration 3.

Everything that was done during this iteration was all committed to the git repository once it was all evaluated and tested. During the evaluation I noticed that the system is essentially a CMS (content management system). “A content management system (CMS) is an application that is used to manage web content, allowing multiple contributors to create, edit and publish. Content in a CMS is typically stored in a database and displayed in a presentation layer based on a set of templates.” (Content Management System, 2021).

## Iteration 3

The user interface and user experience (UI/UX) were the focus of the third iteration. This would include some work with CSS, HTML, and possible JavaScript to create a good-looking UI/UX that would help to encourage and motivate user interaction.

Here it would be best to avoid using some colours as well as look at taking some colours into consideration. Some colours to avoid may be colours such as pure black (#000000) as black is often too overpowering and a more subtle grey (#111111) would be easier on the eyes. Green and red would also be a bad combination as they would conflict with one another as well make distinguishing them apart for people who are a specific type of colour blind (deuteranopia). That is why green, and red are only really used for things like statuses or are commonly representing as “yes or no”.

Some colours to take into consideration of using are the university’s brand colours. If these are too simple then colours that can help create a calming work environment for the user, something that will not be too sharp on the eyes and something that will help to increase their interaction.

### Features worked on

The features worked on in this third iteration of the system were all front-end development for web application.

#### Front-end

To start with I changed the \_Layout file to change the navigation bar so that you can access the pages from all the pages, and you are not required to type in the URL to access the pages. The \_Layout file is used across all the pages that use the shared layout. This allows them to conform to the same layout and help to keep the simplicity when it comes to navigation and keeps the system’s UI rather uniform. The navigation bar is shown below with the retrospective code in the \_Layout file.

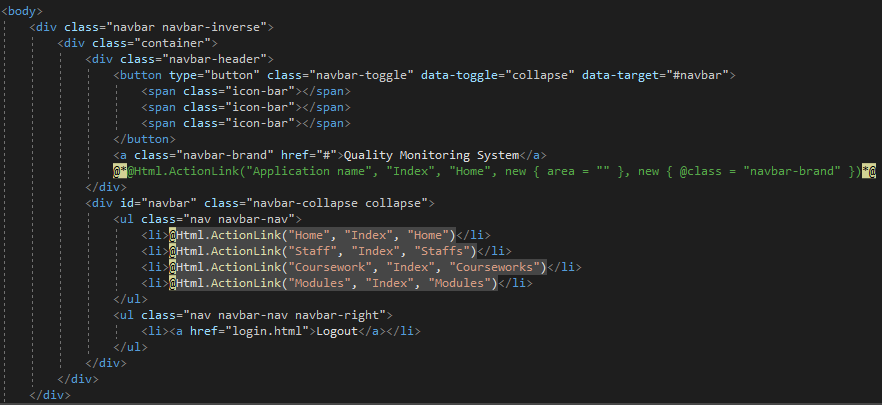


Figure Navbar displayed with the retrospective code

After looking online at the University of Hull’s websites and using an eye dropper tool I could get the hex values of the exact colours they use on their own website. This helped to set the colour theme of the system’s UI.



Figure The theme colours (going from left): #274065, #3D5475, #CE0058, #00828B, #FFFFFF

With the colours set up I went on to create a home page which also serves as to mock-up designs for future features. There will be logging in with user accounts and so a logout button is placed in the top right. On the home page I decided to create a dashboard for the CMS to display some statistics to help towards a more readable view. With the statistics of how many courseworks are ready to be reviewed being available on the first page along with other stats can help the use to see what needs doing. This will help with productivity and help the user be more effective when using the system.

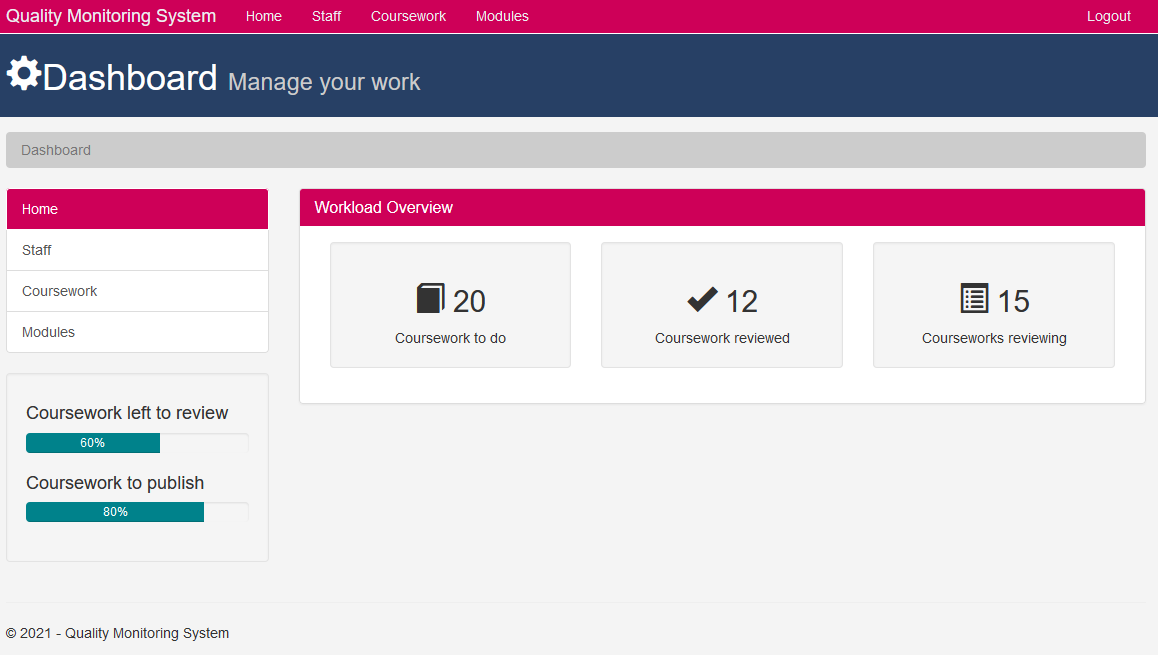


Figure Dashboard on the home page

As seen in figure 21 above, you can see progress bars and statistics of different content. To be able to achieve these features, in the next iteration, stored procedures will be utilised to bring back the current amount of coursework that are ticked as reviewed for this staff member logged in.  
All the views that do the CRUD functionality for the user also have been changed as well to act more user friendly and to help add better readability. Using icons to help accentuate buttons and using colouring to help comprehend the buttons as well, such as red for deleting to warn the user of losing data or green for letting the user know what will be safe to click. You can see this in the examples shown in figure 22 below.

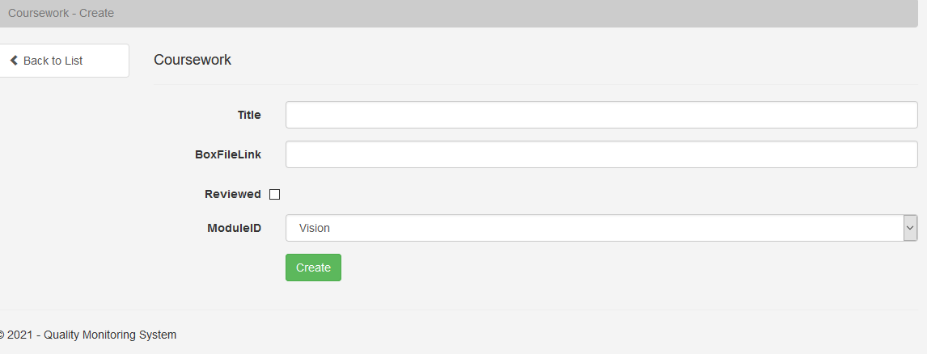
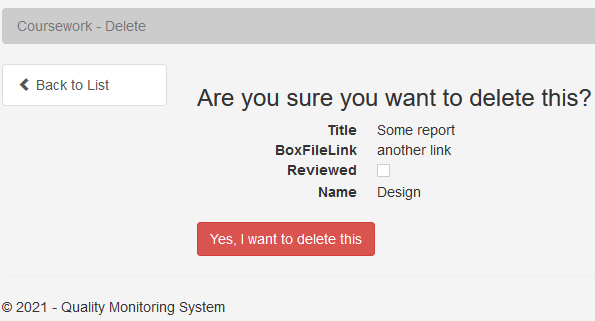
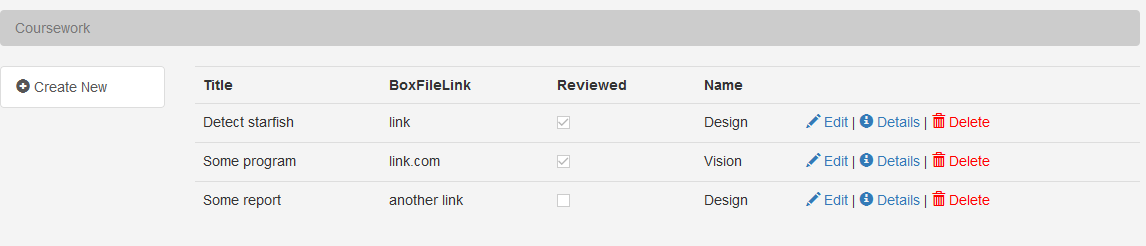


Figure Different examples of where colour is used to help user understanding of interaction

### Testing

#### Software test

|  |  |  |  |
| --- | --- | --- | --- |
| **Hypothesis** | **Expected Result** | **Actual Result** | **Evaluation** |
| Using the CMS submit a new staff member. | New staff member is added to the database and is displayed on the index page | New staff member was added to the database and is displayed on the index page | This test passed again. |
| Using the CMS edit the new staff member to have 1s in each field. | Staff member gets 1s put into each field and is saved to the database. | Staff member now contains 1s in each field and is save to the database | This test passed again. |
| Using the CMS view the details of the staff member. | It opens the page to display the staff’s details. | The details page was opened and correctly displayed the staff’s details | This test passed again. |
| Using the CMS delete the latest staff member from the database. | The latest staff member should be deleted from the database and should not be visible in the CMS table. | The latest staff member was deleted from the database and is no longer visible in the CMS table. | This test passed again. |
| Using the CMS submit a new coursework. | New coursework is added to the database and is displayed on the index page | New coursework was added to the database and is displayed on the index page | This test passed again. |
| Using the CMS edit the new coursework to have 1s in each field. | Coursework gets 1s put into each field and is saved to the database. | Coursework now contains 1s in each field and is save to the database | This test passed again. |
| Using the CMS view the details of the coursework. | It opens the page to display the coursework’s details. | The details page was opened and correctly displayed the coursework’s details | This test passed again. |
| Using the CMS delete the latest coursework from the database. | The latest coursework should be deleted from the database and should not be visible in the CMS table. | The latest coursework was deleted from the database and is no longer visible in the CMS table. | This test passed again. |
| Using the CMS submit a new module. | New module is added to the database and is displayed on the index page | New module was added to the database and is displayed on the index page | This test passed again. |
| Using the CMS edit the new module to have 1s in each field. | Module gets 1s put into each field and is saved to the database. | Module now contains 1s in each field and is save to the database | This test passed again. |
| Using the CMS view the details of the module. | It opens the page to display the module’s details. | The details page was opened and correctly displayed the module’s details | This test passed again. |
| Using the CMS delete the latest module from the database. | The latest module should be deleted from the database and should not be visible in the CMS table. | The latest module was deleted from the database and is no longer visible in the CMS table. | This test passed again. |

|  |  |  |
| --- | --- | --- |
| **Page to test display** | **What is displayed** | **Evaluation** |
| Home page is displayed and looks useable. Nothing should be incomprehensible. |  | The test passes well however there are a couple changes needing to be made such as something to fill the larger empty space still available on the screen. |
| Staff read view (index) displays the tables correctly and there are no issues displaying data. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Staff create view displays an easy to navigate form which can easily be backed out of. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Staff update view (edit) displays an easy to navigate form filled out already with the correct information which can easily be back out of. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Staff delete view displays an “are you sure” message to give the user a second chance before deleting the data entry from the database. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Coursework read view (index) displays the tables correctly and there are no issues displaying data. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Coursework create view displays an easy to navigate form which can easily be backed out of. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Coursework update view (edit) displays an easy to navigate form filled out already with the correct information which can easily be back out of. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Coursework delete view displays an “are you sure” message to give the user a second chance before deleting the data entry from the database. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Module read view (index) displays the tables correctly and there are no issues displaying data. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Module create view displays an easy to navigate form which can easily be backed out of. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Module update view (edit) displays an easy to navigate form filled out already with the correct information which can easily be back out of. |  | The test passes well and shows good use of UI/UX with no bugs found. |
| Module delete view displays an “are you sure” message to give the user a second chance before deleting the data entry from the database. |  | The test passes well and shows good use of UI/UX with no bugs found. |

### Evaluation for next iteration

In the next iteration I will tackle some stored procedures as well as iron out some of the issues with some of the UI as well as try to add more tables for things like the review page to individually review each coursework properly.

# Evaluation

Unfortunately, due to a pandemic any form of evaluation had to be placed on hold as there is no way to create a controlled environment to ensure that the data collected is not erroneous due to anomalies from uncontrolled sources. The only evaluation that could be done was self-evaluation.

Due to a lot of time constraints and other bumps in the road I was unable to complete the software on time to a standard that can be published and used, however, the iterations that have been made are ready working and can be built upon easily if the timeline got extended further.  
  
The tests run during the iterations all passed which shows that they were built stable and would be possible to expanded upon. Testing time could have been cut shorter if I were able to get unit testing to integrate with the CMS to ensure the software worked correctly and would not break from new additions or changes. This was not the case, however, and I believe this is where a time was lost. During each change throughout iterations, I had to change test things to ensure they worked so that I would not be wasting my time expanding on something broken.

# Conclusion

In conclusion there were many more issues that came up that were unexpected, and this task may have been largely out of scope for 1 person to do in the timeframe given (See Appendix A). The original time frame had things to be running smoothly until April, however issues kept interfering with the designing of the software and no development was started. In hindsight there should have been a clearer plan for the coursework and there should have been a more accurate estimate over the timeline given.

With given more appropriate testing and a better design plan I believe it could have had a lot more progress made throughout the project, however, due to my hindsight, the project took longer than expected. I do believe though that if the CMS was to still be worked on it would be possible to create a piece of software that will aide and increase productivity within the workplace, reducing more human errors and helping to encourage user interaction by having a more manageable work space.

References

Andrew Powell-Morse, n.d, Iterative Model: What Is It And When Should You Use It?, viewed 23 October 2019, <https://airbrake.io/blog/sdlc/iterative-model>

Cascio, W. and Montealegre, R. 2016. How Technology Is Changing Work and Organizations. Annual Review of Organizational Psychology and Organizational Behavior. 3. 349-375. 10.1146/annurev-orgpsych-041015-062352.

Codecademy. 2021. What is CRUD? | Codecademy. [online] Available at: <https://www.codecademy.com/articles/what-is-crud> [Accessed 25 April 2021].

Definition of MOTIVE (2020). (n.d.). Retrieved from Merriam-webster: https://www.merriam-webster.com/dictionary/motive (Accessed: August 1, 2020).

Docs.microsoft.com. 2021. *Entity Data Model - ADO.NET*. [online] Available at: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/entity-data-model> [Accessed 25 April 2021].

Git-scm.com. 2021. *Git - About Version Control*. [online] Available at: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control> [Accessed 25 April 2021].

Guest Author (2019) How User Design Can Impact Engagement and Learning for K-12 Students | Getting Smart, Getting Smart. Available at: https://www.gettingsmart.com/2019/04/how-user-design-can-impact-engagement-and-learning-for-k-12-students/ (Accessed: August 1, 2020).

Optimizely. 2021. Content Management System. [online] Available at: <https://www.optimizely.com/uk/optimization-glossary/content-management-system/#:~:text=A%20content%20management%20system%20(CMS,on%20a%20set%20of%20templates.> [Accessed 30 April 2021].

Systems Analysis and Design, Assessment Scrutiny Process, Faculty of Science and Engineering, The University of Hull, September, 2019.

Yalanska, M. (2017) Two Types of User Motivation: Design to Satisfy, Tubik Blog: Articles About Design. Tubik Studio. Available at: https://blog.tubikstudio.com/two-types-of-user-motivation-design-to-satisfy/ (Accessed: August 1, 2020).

Appendix A – Interesting but not vital material

