

Dissertation 2015/2016

School of Computing & Information Engineering

Ronan Lowry B00554185

[Personalised Timetable]

Supervisor: Simon Fraser

Second Marker: Gaye Lightbody



Contents

Abstract	3
Introduction	4
Analysis.....	8
Requirements Analysis	20
Design.....	22
Implementation.....	32
Testing and Evaluation	44
Conclusion	52
References and Appendix.....	57

"When submitting your work you are agreeing with the following statement: I declare that this is all my own work and that any material I have referred to has been accurately referenced. I have read the University's policy on plagiarism and understand the definition of plagiarism. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file."

Abstract

Every new semester, students are given out their list of modules by lecturers and are given a hard copy of their timetable. A hard copy is a simplistic way of distributing a timetable but it has disadvantages in that it can be lost or the timetable updates and new hard copies are required to be distributed. A larger problem arises for the student when they go online to find their timetable in the Ulster University website. The current system for looking for your module timetables requires time and effort to navigate through a series of menus asking for a school options, course options and academic year of study. Some of the outputs also require the user to build their own timetable as the website can only provide one module at a time when a student can take up to four modules a semester depending on the course.

What is required is a personalised timetable that each user can refer to throughout the semester improving on the system already in place. The proposed system will be designed to overcome the above issues by developing a separate website, in which users can log on and instantly see not only each week of the semester already populated with module information, but also create notes. The notes could be used to remind users about which notes to bring to lectures or important dates in the semester. Lecturers will have a separate level of access allowing them to see not only the modules they teach but also the entire course timetable they teach on giving them a broader view of the semester.

This report will follow through the steps taken to produce the above solution. Beginning with a detailed analysis of the problem and solution, requirements lists will be drawn up in order to design a fully working website. A prototype will be implemented and improved upon during the project after which a final system will be tested and evaluated.

Introduction

The problem

Over the course of 2014 / 2015, over 17,000 students were enrolled in full time courses across five campuses in Ulster University. A full breakdown of these numbers can be seen in table one below.

Actual enrolments 2014/15		Summary Table by Campus and Level					
University of Ulster							
		Belfast	Coleraine	Jordanstown	Magee	London / Birmingham	Total
Full Time	Degree	1537	3769	7347	3644	1371	17668
	PG Taught	65	320	393	134	162	1074
	PG Research	27	169	202	55		453
	Full time total	1629	4258	7942	3833	1533	19195
Part Time	Degree	248	396	2554	923		4121
	PG Taught	41	640	2457	309		3447
	PG Research	3	50	120	33		206
	Part time total	292	1086	5131	1265	0	7774
Total		1921	5344	13073	5098	1533	26969

Source: Internal University Statistics

Table one taken from http://www.ulster.ac.uk/finance/planning/information_development/devnos2014-15.pdf

This report will look into the problem that spans across all campuses in the Ulster University. Each of the students enrolled have a specific timetable of lectures and tutorials. It has been identified however that the way in which this timetable can be accessed is not efficient or time effective. The current model for accessing the timetable online can be confusing to navigate through, it is not personalised to each student, and is generated from the online portal via the following categories;

- School
- Institution
- Module
- Master Module
- Programmes

In essence, time and patience are required to use this system correctly and to produce a timetable which can be trusted by each student. The information provided can be either too much or either too little depending on the categories by which the timetable is generated from above.

When looking at a university or a student, it is the timetable which provides this organisation, which should allow students to properly plan their semester around classwork, assessments and revision, coursework and additional reading.

At present the Ulster University also provides a single sheet, hard copy timetable based on week one of each semester to every student. Included information is the time and room for each lecture and available tutorials, along with the lecturer's name and module codes. While this information is adequate for the first week of teaching, other useful information is lacking. For example, given that this is a hard copy timetable it is difficult to amend, i.e. if a lecturer moves a class an entire new set of timetables must be printed for each student or, if a lecturer sets a deadline the student must note this in either a diary or a calendar online. Currently all available tutorials are on the timetable provided, however if the class is broken into groups, half the class may visit one tutorial while the other class visits the next, leaving some of the information on the timetable now irrelevant. In short, the timetable system that is already in place does not meet the needs of today's students in Ulster University. The following problem statement will identify this problem and then will begin to outline the possible solution.

Problem Statement

"To create a personalised timetable for Ulster University students which can be accessed on a separate website. A database will be linked to the web page, which will contain all the relevant information for each student including their modules and lecture times. The timetable will be automatically generated in order to overcome the flaws in the system currently in place and therefore will act to replace the system the Ulster University has already in place".

Proposed Solution

The proposed solution is a website which will generate a timetable that can then be personalised by each individual user. The stakeholders involved will include and lecturers who will be able to see not only the modules they are teaching but also the entire list of modules in their course and the list of students enrolled in that course.

In today's mobile market there are a number of competing devices, including iOS and Android. None of these devices have a complete market share and neither do they share software, in that an app developed for one device could not operate on the other. Therefore creating one app would not target a wide enough audience in the University. It was decided then that a web page similar to that already on the Ulster University portal should be created.

Aims and Objectives

The aim of this project is to create a website, which will populate automatically a timetable for the user logging in based on the course they are registered to study. This website will provide other features to aid in the user to personalise their timetable by adding notes. The objectives of this report are outlined below;

1. Conduct research into the problem to understand the issues involved with finding a timetable and come up with a solution which fits the needs of users
2. Create a list of functional and non-functional requirements for the end system which are fit for purpose to the users
3. Design an appropriate system for Ulster University students and code the website based on the research carried out and ensure appropriate and accurate information is generated
4. Test the system thoroughly in house and with a focus group to ensure that the build is of a high standard and overcomes the original problem.

Summary

The remaining chapters of this report will present background research into the problem, followed by a detailed description of functional and non-functional requirements of the system being created. After these Analysis chapters, the report will document the design process and the implementation process explaining how the developer built the end system looking at the tools used and any new technologies which were employed before introducing the approach which was taken to create the system. Significant code will also be demonstrated in the implementation chapter along with what the developer has learned from creating the website. The report will detail the testing and evaluation of the website through the focus group and the results of this testing phase will be analysed. The conclusion will summarise the work which has been completed and critically examine the resulting system and state if the aims and objectives have been achieved.

Analysis

Introduction

The following analysis chapter sets out to find the source of the problem with users finding their timetable online and looks at possible already available solutions to creating their own personalised timetable. Given this research, a solution will be devised and discussed, before developing a list of functional and non-functional requirements for the system as a whole.

Background Research

On the Ulster University site, a student can look at timetables through the following steps:

Initially, the student must use their log on details to access the student portal. Diagrams one to eleven show this process.

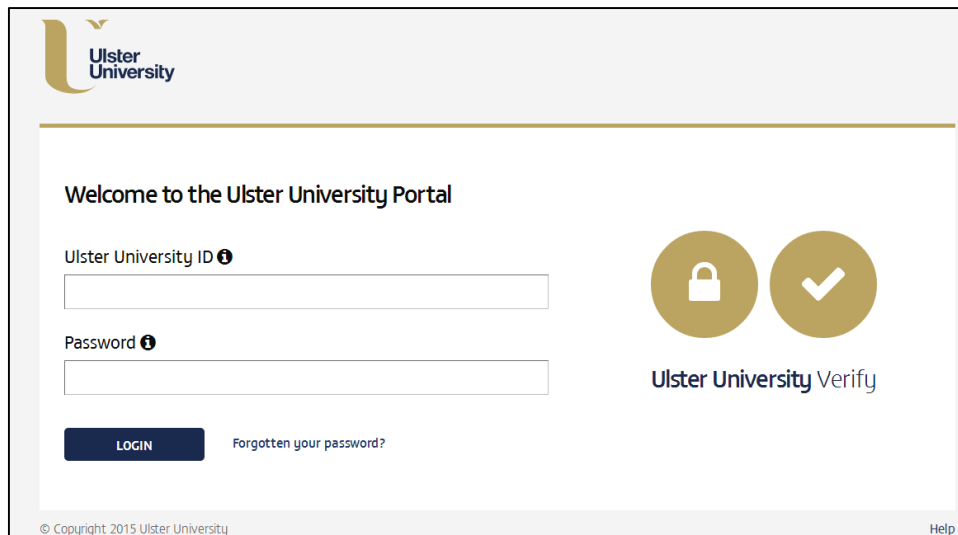


Diagram one

Once in the portal, the student can select the 'my studies' tab along the top of the page. Whilst on the 'my studies' tab, the following pane appears:

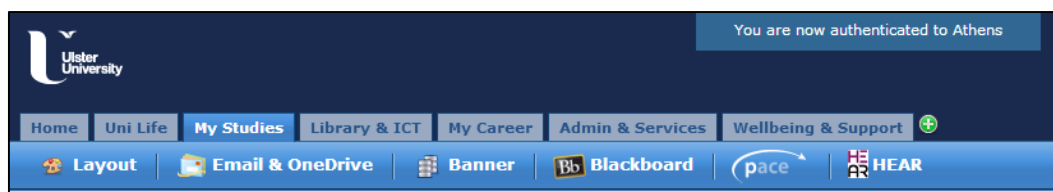


Diagram two

From here, the class timetable button is available. At this point the student can get access to which campus they are studying from.

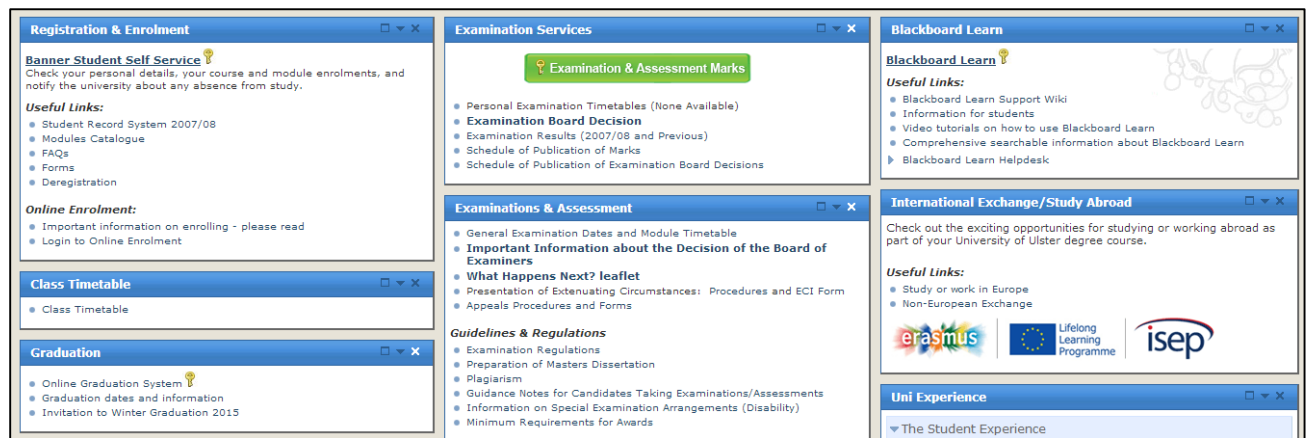


Diagram three



Diagram four

As stated above there are five categories from which a timetable can be generated.




Diagram five

If a student selects the first option, 'Institution', they will be able to select a teaching period across every hour, day and week number. Selecting these options will provide every lecture being taught at that time across all schools in that campus. This generator is very messy and not likely to provide a student with any meaningful information.

Coleraine

Semester Two - 2015/16



To View Institution Timetable Make Selections from the options below.

☐ Monday
 ☐ Tuesday
 ☐ Wednesday
 ☐ Thursday
 ☐ Friday
 ☐ Saturday
 ☐ Sunday

Teaching Periods

<input type="radio"/> 9.15	<input type="radio"/> 10.15	<input type="radio"/> 11.15	<input type="radio"/> 12.15	<input type="radio"/> 13.15
<input type="radio"/> 14.15	<input type="radio"/> 15.15	<input type="radio"/> 16.15	<input type="radio"/> 17.15	<input type="radio"/> 18.15
<input type="radio"/> 19.15	<input type="radio"/> 20.15	<input type="radio"/> 21.15	<input type="radio"/> 22.15	<input checked="" type="radio"/> All

Week No


1

Diagram six

Option two, 'School', shows every class that is available for that selected school. This time the timetable is narrowed down to a working week, Monday to Friday and has more meaningful information than before. However there is still some irrelevant information here

Coleraine

Semester Two 2015/16



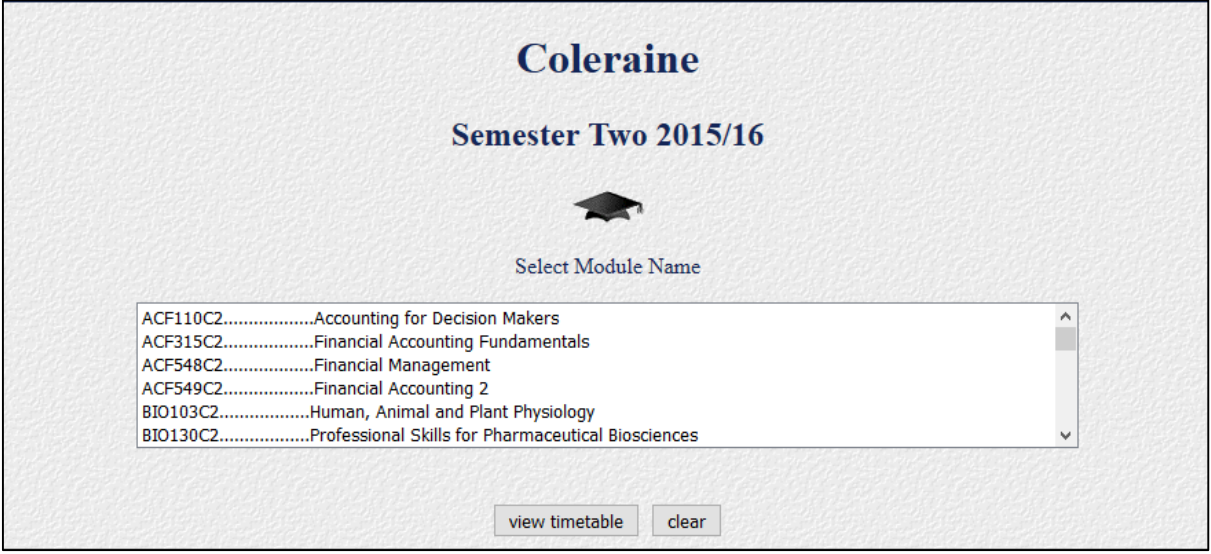
Please choose a School Timetable from the list below.

Faculty of Arts	English, History and Politics Modern Languages Media, Film and Journalism
Faculty of Business and Management	Business and Enterprise Hospitality & Tourism Management
Faculty of Computing and Engineering	Computing and Information Engineering
Faculty of Life and Health Sciences	Environmental Sciences Biomedical Sciences Nursing Pharmacy Psychology
Faculty of Social Sciences	Education
Department of Teaching and Learning	CELTI

and each class is shown only as a module code. A student will have to manually check off their module codes against those in the table which again can be time consuming and therefore inefficient.

Diagram seven

By selecting 'Module', a student can look at each individual module offered by that campus in a drop down menu. Looking through this menu can be difficult in that the list is quite large and again the student would need to know the code for every module which they are studying. This generator however will only show relevant information to each individual student although it is still quite a manually driven process. It should be noted that a student can select multiple modules at one time, but on viewing the timetable, each module will have a separate timetable.



The screenshot shows a web interface for Coleraine Semester Two 2015/16. At the top, the text "Coleraine" is in a large blue font, followed by "Semester Two 2015/16" in a smaller blue font. Below this is a graduation cap icon and the text "Select Module Name". A large text box contains a list of modules with their codes and names: ACF110C2.....Accounting for Decision Makers, ACF315C2.....Financial Accounting Fundamentals, ACF548C2.....Financial Management, ACF549C2.....Financial Accounting 2, BIO103C2.....Human, Animal and Plant Physiology, and BIO130C2.....Professional Skills for Pharmaceutical Biosciences. At the bottom of the text box are two buttons: "view timetable" and "clear".


Module Code	Module Name
ACF110C2	Accounting for Decision Makers
ACF315C2	Financial Accounting Fundamentals
ACF548C2	Financial Management
ACF549C2	Financial Accounting 2
BIO103C2	Human, Animal and Plant Physiology
BIO130C2	Professional Skills for Pharmaceutical Biosciences

Diagram eight

'Master Module' differs from 'Module' in that when selecting multiple options for modules, only one timetable is generated. Again however it still has the same drawbacks as 'Module' with a longwinded drop down menu and the student would have to be aware of all the modules in each semester they were studying.

Coleraine

Semester Two 2015/16



Select Module Name - Use Ctrl to select more than one module

ACF110C2.....Accounting for Decision Makers

ACF315C2.....Financial Accounting Fundamentals

ACF548C2.....Financial Management

ACF549C2.....Financial Accounting 2

BIO103C2.....Human, Animal and Plant Physiology

BIO130C2.....Professional Skills for Pharmaceutical Biosciences

BIO314C2.....Biotech

BIO316C2.....Professional practice

BIO317C2.....Molecular Biology

BIO385C2.....Innovation -LED Entrepreneurship

Diagram nine

The last option available, and the most useful is the “Programmes’ button. Selecting this allows the student to identify which course they are undertaking and then select a week on the timetable to view. This view will show only the modules on each specific course in any given week. However, there is still too much information to filter through to get the final timetable.

Faculty of Arts	
	English and History
	Modern Languages
	Media, Film and Journalism
Faculty of Computing and Engineering	
	Computing and Information Engineering
Faculty of Life and Health Sciences	
	Environmental Sciences
	Biomedical Sciences
	Food Nutrition & Dietetics
	Optometry
	Psychology
Faculty of Social Sciences	
	Education
Ulster Business School	
	Department of Business and Enterprise
	Department of Hospitality & Tourism Management
Centre for English Language Teaching	
	CEL
TESOL	TESOL

Diagram ten

Computing and Information Engineering Programmes

Coleraine Semester TWO 2015/16



Select Programme Name

BSc Hons Computing FT (2136) Year: 1

Make Teaching Week Selection (Default Value ALL Weeks (12 Teaching) of Semester)

☒ 1-8,11-14

Select Required Week -- Monday date for following seven days:

<input type="radio"/> 25/01/16	<input type="radio"/> 01/02/16	<input type="radio"/> 08/02/16	<input type="radio"/> 15/02/16	<input type="radio"/> 22/02/16	<input type="radio"/> 29/02/16	<input type="radio"/> 07/03/16
<input type="radio"/> 14/03/16	<input type="radio"/> 21/03/16	<input checked="" type="radio"/> 28/03/16	<input type="radio"/> 04/04/16	<input type="radio"/> 11/04/16	<input type="radio"/> 18/04/16	<input type="radio"/> 25/04/16

view timetable

clear

Diagram eleven

From this research carried out, it can be understood where the original problem arises. The menus are confusing to navigate and users if not properly shown will have to manually traverse each of the five options to figure out which will provide them with the most meaningful timetable. A better option would be a single option which shows the timetable for the user as they log on rather than selecting from the five menus and then manually inputting details of their course or modules or school of study. None of the above will allow a student to create notes either meaning the student will never truly own a personalised timetable.

Comparative Analysis

This report will now look at online options for users to generate their own personal timetables. It should be noted that by even using these options the user will first still have to either base their personal timetable on the hard copy timetable they receive at the beginning of the year or they will have to use the timetable they find on the university website. Users could also create their own timetable using Microsoft Excel or Microsoft Word but the same process will have to be taken i.e. the user will still need to access the Ulster University website to gather their course information.

Two websites have been chosen for this comparative analysis, getrevising.com which requires a user to create a login to use and gradetracker.com. Both these systems are free to use.

Initially, getrevising.com asks the user to sign in, provided the user has registered for the webpage. To create a personal timetable the user selects create from the header menu and then create a study planner. Diagrams twelve to fifteen outline this first online solution.

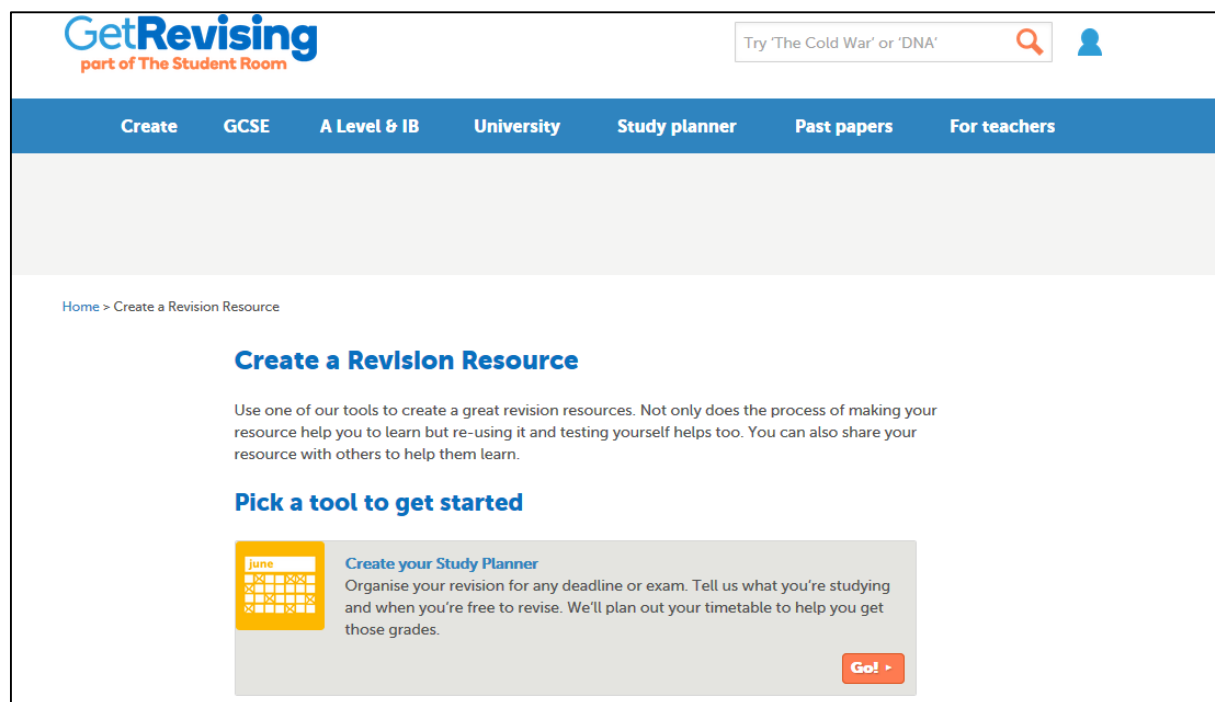


Diagram twelve

This is followed by the website asking questions about the users preferences to study times and the course that they are studying. After they have completed these questions, the following screen will appear

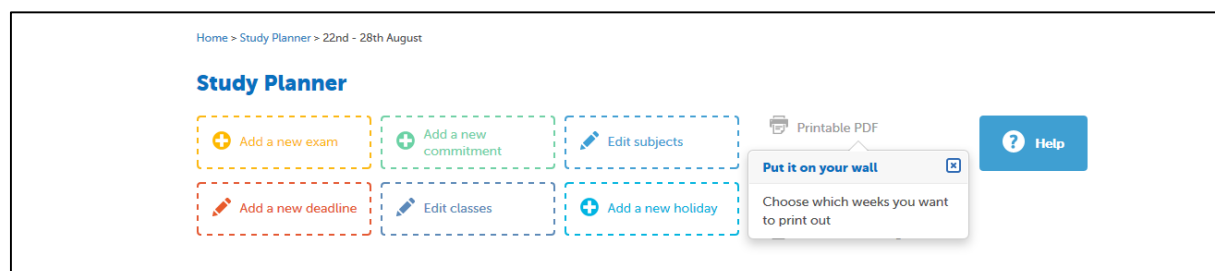


Diagram thirteen

	This week		◀ 22nd - 28th August ▶				
See earlier ▲	Mon 22nd	Tue 23rd	Wed 24th	Thu 25th	Fri 26th	Sat 27th	Sun 28th
						add activity	add activity
14:00 - 15:00						add activity	add activity
15:00 - 16:00						add activity	add activity
16:00 - 17:00					add activity	add activity	add activity
17:00 - 18:00					add activity	add activity	add activity
18:00 - 19:00					add activity	add activity	add activity
19:00 - 20:00					add activity	add activity	add activity
20:00 - 21:00					add activity	add activity	add activity
21:00 - 22:00					add activity	add activity	add activity

Diagram fourteen

The website works by splitting each week into times and dates, creating a table with cells. Options to the user include add holiday blocks and commitments blocks as well as the ability to edit the modules, or subjects they are wishing to display in the timetable. A user then selects the module they wish to add to the timetable and the times and day they are in class and the timetable will automatically update. The timetable itself is a full year calendar and the week displayed can be progressed. For example, ICT was added as a subject for a sample user. The user input that ICT ran from 1pm to 4.30pm on a Monday. The timetable then updates and the ICT class is displayed each Monday for the rest of the calendar year.

See earlier ^	Mon 19th	Tue 20th	Wed 21st	Thu 22nd	Fri 23rd	Sat 24th	Sun 25th
09:00 - 10:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity
10:00 - 11:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity
11:00 - 12:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity
12:00 - 13:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity
13:00 - 14:00	University ICT	add activity	add activity	add activity	add activity	add activity	add activity
14:00 - 15:00		add activity	add activity	add activity	add activity	add activity	add activity
15:00 - 16:00		add activity	add activity	add activity	add activity	add activity	add activity
16:00 - 17:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity
17:00 - 18:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity
18:00 - 19:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity
19:00 - 20:00	add activity	add activity	add activity	add activity	add activity	add activity	add activity

Diagram fifteen

A potential con of this website is that it does take time to complete as the user will have to insert each class individually each holiday individually on top of the fact the user will already have to go through the process of finding their timetable on the Ulster University website. The user will have to give an email address to allow access to this site.

Gradetracker.com is the second comparative website this report documents. This website allows a user to set up a single page timetable, breaking a week down to days and times as columns and rows like getrevising.com.

Class Schedule Maker

Create And Print Your Course Schedule In Just A Few Minutes!

If you are looking for a quick and simple schedule generator, then look no further. This free schedule maker uses a weekly calendar template to create your schedule of courses. You can use this customizable schedule planner to generate your timetable for your high school or college classes.

Class Name:

Class Color:

Math 101

	Mon	Tue	Wed	Thur	Fri
7:00 am					
8:00 am					
9:00 am					
10:00 am					
11:00 am					
12:00 pm					
1:00 pm					
2:00 pm					
3:00 pm					
4:00 pm					
5:00 pm					
6:00 pm					
7:00 pm					
8:00 pm					
9:00 pm					
10:00 pm					

Print Your Class Schedule

Diagram sixteen

Users input the name of the class or module they wish to display on the timetable and the colour which they wish the display cell to be. After this they click the cell they wish to update and are then given the option to print out the table. An example is seen below, showing ICT on a Monday from 1pm to 4.30pm.

Class Schedule Maker

Create And Print Your Course Schedule In Just A Few Minutes!

If you are looking for a quick and simple schedule generator, then look no further. This free schedule maker uses a weekly calendar template to create your schedule of courses. You can use this customizable schedule planner to generate your timetable for your high school or college classes.

Class Name:

Class Color: Aqua

ICT

	Mon	Tue	Wed	Thur	Fri
7:00 am					
8:00 am					
9:00 am					
10:00 am					
11:00 am					
12:00 pm					
1:00 pm	ICT				
2:00 pm					
3:00 pm					
4:00 pm					
5:00 pm					
6:00 pm					
7:00 pm					
8:00 pm					
9:00 pm					
10:00 pm					

Print Your Class Schedule

Diagram seventeen

This website is not based on a calendar year, only displaying a single week in a timetable. For Ulster University Students the print out feature may be useful to add notes but the cons are still the same as those of the hard copy issued at the beginning of each year.

Solution

A new way in which users can access their timetable is the proposed solution of this report. The developer has identified the issues rests with how the users find information on the Ulster University website. The webpage will require users to log into the system which will automatically generate a timetable for that user. This will act to mitigate the original problem highlighted through this chapter.

The targeted audience for this system will be students and lecturers as they are both stakeholders required to have to access their timetables for each semester. As there are two main competitive businesses in the mobile phones market, Android and iOS the

developer has decided not to build an app as this could restrict the capabilities of the system or only reach a certain corner of the market depending on which operating system to design an app for. The developer also noted that the current system was online and to keep the new system familiar to existing users of the system, the developer has decided to build a website.

The website will be designed so that a user will only have to log into the system to see their timetable for the coming semester, as the university only produces timetables a semester in advance. The website will be easy to use and provide the student a place to create small notes to allow their timetable to be personalised. Lecturers will also be able to see their timetable once they log in, but should also have access to the students enrolled in the course they lecture in and the entire timetable for the course they lecture in, not just those lectures they must teach, giving them a better view of the course as a whole.

The web page will offer the following technical benefits:

- Students will have easy access to their entire semester timetable, containing information on their modules, lectures and tutorials without navigating through separate menus.
- Students will be able to create notes on the web page to track work completed and organise work in the coming weeks.
- Lecturers will be able to log onto the system to see the entire timetable for courses they are involved in.
- Lecturers will be able to see class lists for the students enrolled in their course.

Summary

This chapter has defined the problem of users accessing their timetable on the Ulster University website and the issues of creating a personalised timetable on separate websites. A solution has been provided which sets out to circumvent the original problem to allow users some freedom in creating notes in a timetable which is automatically generated after a user logs in.

Requirements Analysis

Functional Requirements

The functional requirements are those which require the system to work properly. In the case of the timetable, there are two students and lecturers. Students and lecturers will have different levels of access to the website which are detailed in the functional requirements below.

- Users should be able to log into the system based on the information already held on the Ulster University database.
- Students should have a separate log in from lecturers as the features the website offers each user will be different
- Automatically, a timetable will be generated for the user once logged in without any further menus or manual input from the user
- A student user should be able to leave a note about modules they are studying which will be saved next time they access the system.
- Lecturers will have a timetable showing the modules they teach but also on a separate page they should have access to entire timetables for courses they lecture in.
- Lecturers will also have a class register to identify students in their classes.

Non- Functional Requirements

The non-functional requirements listed out below show how the website will execute the functional requirements above. The following will describe the performance of the website and will be taken into account for the design of the system.

- In order for the system to overcome the original problem it needs to be efficient. The user must log in and see their timetable with minimal time being wasted.
- The website must be available, meaning that it must be operational for when users require viewing their timetable. If for example the website required back up or upgrades which required downtime this should be at a time when the users won't need to be using the system
- The information displayed in the webpage needs to be reliable. The information the users views when they log on to the system should be accurate and not contain

unnecessary information the user either does not need or information that is incorrect.

- The system must be intuitive to use. The system will be built with the end user in mind and to overcome the original problem that is the current system is difficult to navigate and the buttons directing users to new pages are ambiguous. The new system should be intuitive for the user to overcome this.
- The system must also be secure. A user must only be able to log in with the correct username password combination. Without the correct log in details the user should not be able to access the system.

Design

Introduction

This chapter will focus on the design of the website that is the user interface the architecture of the website including mock up screen shots of how the web page will look. A map of the site will show how users can navigate through the web site and a description of the database will also be included along with an Entity – Relationship diagram.

User Interface Design

A well designed User Interface will determine how frequently the web page will be used. The end goal when designing the web page is to create a completely accessible tool for the entire target audience, with a consistent pallet of colours and logos. Given that the web page will be used by the Ulster University it will follow the same colour scheme throughout.

The flow of the website will also be intuitive for the user, with a main function of displaying the timetable for their course. With the original problem in mind and therefore the aim of this project, the website will have minimal user input in order to have their timetable displayed. Once the user logs in, the details of their timetable will be the main screen that the user sees. This information will be kept in a database and php code and SQL statements will allow the website to access this database.

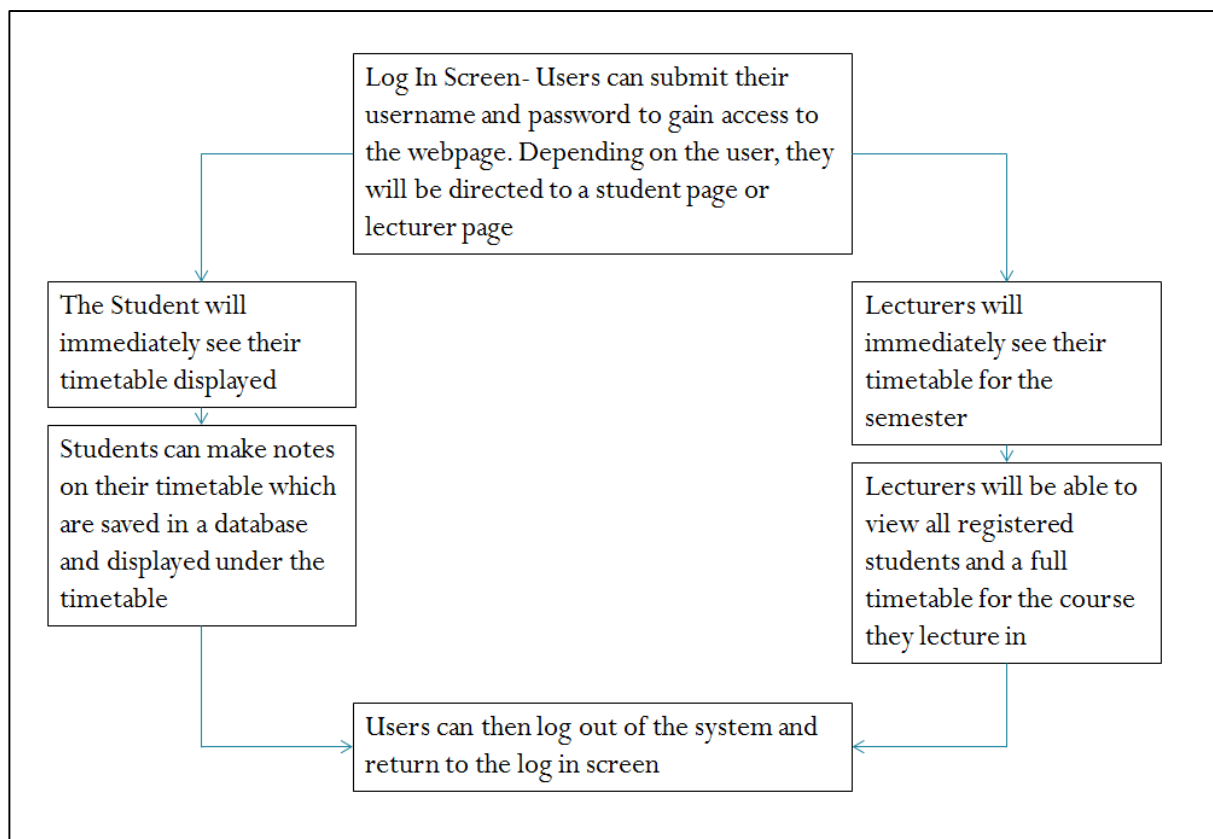
For designing the user interface, Shneiderman's eight golden rules of interface design can be applied. Shneiderman created these rules in 1987 after research into Human Computer Interaction (HCI) though the technology has changed drastically since then, the rules are still very much applicable. They are

1. Strive for consistency
2. Enable frequent users to shortcuts
3. Offer informative feedback
4. Design dialogue to yield closure
5. Offer simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load

Website Map

From an end user point of view the system must be simple to use in order to overcome the original problem that is it is currently a complex system in place to display a personal timetable. In order to achieve this, the developer has decided that the user input should be minimal, and the webpage should not be cluttered or difficult to navigate through.

Diagram eighteen




As can be seen from the map of the website on diagram eighteen, each user will see their timetable immediately. This will overcome the awkward menu system already in place, and users can quickly see what their timetable is once they log on to the system. Following from the requirements the developer has designed the website to allow student users to keep notes in their timetable and lecturer users will have a full view of the course they teach, not just the modules that they are lecturing.

Given that the current system in place is difficult to navigate through, the developer will use simple buttons to allow the user to access the separate functions in their website. The buttons will appear on the header of the website, clear for the user to see, and will not be ambiguously titled as the buttons in the Ulster University website.

Mock up Screens

Using the website map and the requirements as a guide the developer has created several screens to show what is anticipated from the final system. Diagrams nineteen to twenty two show this.




Welcome

Please log in to view your timetable

Username

Password

[LOGIN](#)



Welcome

You are registered for Course Name

Username

[Click to Create or View Notes](#)

Your timetable is:


Time /Day	0915	1015	1115	1215	1315	1415	1515	1615
Mon								
Tues								
Wed								
Thurs								
Fri								

[LOGOUT](#)

Diagram nineteen & twenty above. Diagram twenty one & twenty two below

Welcome
Username


You are registered for Course Name



Click to Return
to Timetable

Create your note here

Notes displayed here


Module 

Please enter your note here

LOGOUT

Welcome
Username

You are registered for Course Name



Click to Return
to Timetable

Here is the full course
timetable

Notes displayed here

Time /Day	0915	1015	1115	1215	1315	1415	1515	1615
Mon								
Tues								
Wed								
Thurs								
Fri								

LOGOUT

The above figures display how the developer expects the system will look after completion. The first screen shows the login / welcome screen user will initially see. Here they will be prompted for their username and password. Once entered they can select submit to move onto the next screen if the system accepts the username and password as correct.

The second screen users will see will be similar for both students and lecturers. They will immediately see their timetable in the main section of the page. The header of the screen will display a welcome message with the users name and the course they are studying indicating they have logged in successfully. A student user, shown above will then see a button to create or view the notes they have made. A lecturer will see a similar button which will allow them to view more information on the course they are lecturing.

The students note screen can be seen above. This screen will allow users to both create notes and to view notes they have already made. The welcome header will still be shown along with their name and course. A button will allow the user to return to view their timetable. Once a note has been created it should automatically appear on the left panel of the screen.

The lecturer will be able to view a timetable for the entire course they lecture in not just the modules they teach. On the left panel of the main screen they will also have a full class register. Again, as with the student screen the lecturer will still see the welcome message with their name and course.

All through these screens the layout and colour pallet will be consistent and the user will be able to log out. The Ulster University logo will also appear on each screen keeping consistency through the website.

Colour

As can be seen from the screen mock ups diagrams nineteen to twenty two, the colour pallet the developer has chosen is a light blue and dark blue to separate the header and main section of the screen. The background will be mostly white with black writing as to not distract the user. The timetable will have alternate coloured rows to stand out at quick glances from the user also.

Architectural Design

This system will be created using a variety of tools and languages working together for the final website allowing students and lecturers to view their timetables. To ensure that this is possible the following section will detail the server and client scripting languages which interact together to produce the website. Server side languages allow code to be executed on a server. The client receives the results of the running code although they are unaware of the code producing these results. On the client side, a website is accessed through a browser. The server side receives the request from the browser and replies with the running server side code.

Relational Database Design

In order to store all information required to produce a timetable and the information on the intended users of the system, a relational database will be created. This will be carried out by the following steps:

- Decide what information to gather on users in order to fully populate the database
- Use this information as a basis to create relationships in different entities
- Create an entity relationship diagram as a way to display this information
- Create attributes and primary and foreign keys to link the information stored in the database
- Apply normalisation to ensure a proper relational database design is achieved.

Diagram twenty three below shows the prototype database design which was deemed inaccurate during the latter half of the report.

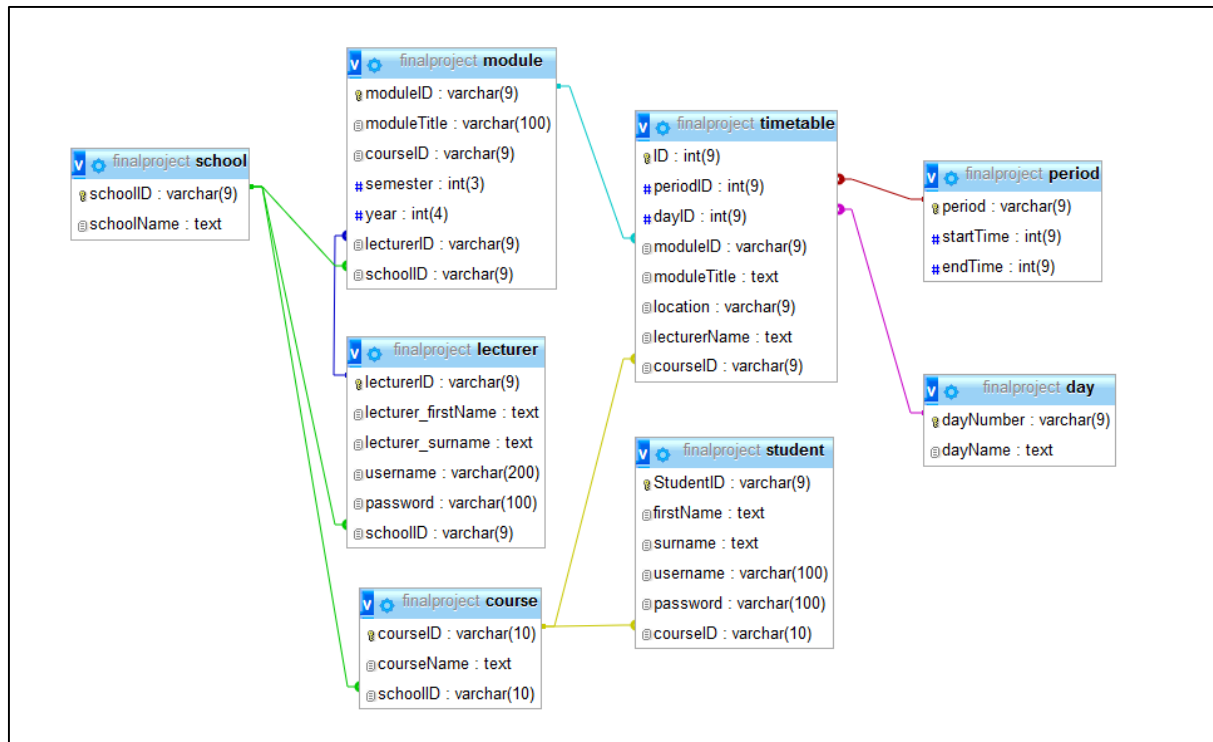


Diagram twenty three

The developer understood that after the prototype was built, a separate lecturer and student table was unnecessary as well as the school table. If the log in information was to be held in one table, then a user type column would be required to distinguish between students and lecturers. A new database was created based on what was learned from the prototype. The following ER diagram, diagram twenty four represents this new database.

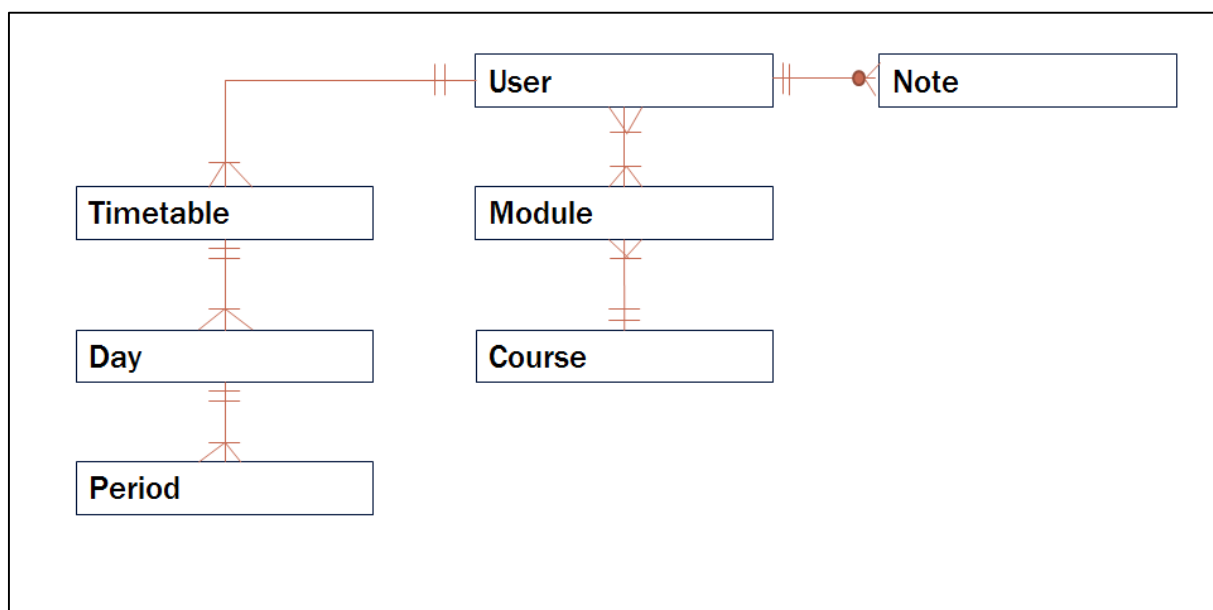


Diagram twenty four

The current database has seven tables, and holds the following information.

- User Table: User ID, first name, surname, login, password, user type, course ID*
- Module table: Module ID, module name, course ID*, user ID*
- Course table: Course ID, course name
- Note table: Note ID, user ID*, module ID*, note
- Timetable table: ID, period ID*, day ID*, module ID*, module Title, location, lecturer name, course ID*
- Day table: Day ID, day name
- Period table: Period ID, start time, end time

Diagram twenty five shows this database after it has been built:

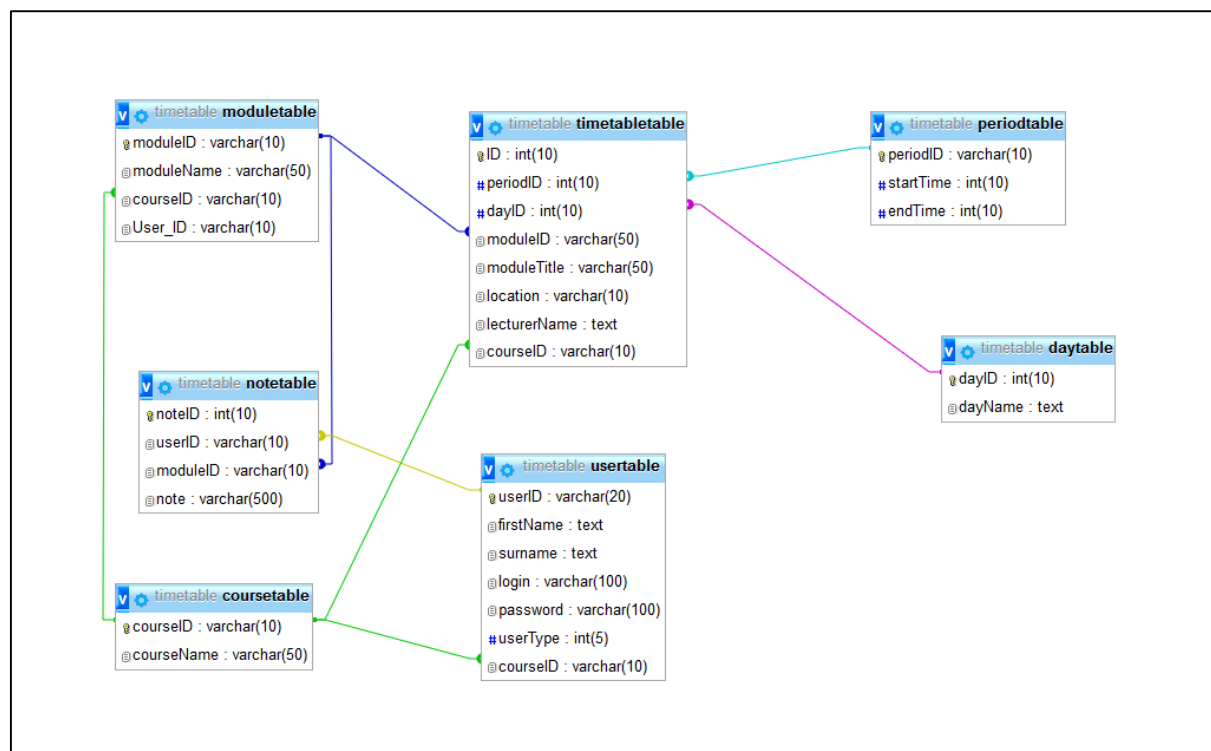


Diagram twenty five

As can be seen above, seven individual tables are included in the timetable database which will be discussed in greater detail below.

User Table

The user table has been created to hold the user details such as name, username, and password and user type. The username and password will be used to allow the user to login into the system. The user type defines if the user is a student or a lecturer. The user ID acts as the primary key for this table and it is linked to the course table via the course ID which is the foreign key in this table.

Module Table

The module table holds data for each module in the database. The module ID is the primary key. This table is linked to the course table again by using the course ID as a foreign key. The user ID is also a foreign key here. This creates a relationship allowing the developer to assign a lecturer to a module, in other words the user ID here shows which lecturer teaches this module. The module name is also included in this table.

Course Table

The course table contains information on the course name and course ID. This table has no foreign keys and the course ID acts as a primary key.

Note Table

The note table is used to store the notes that students create. In it, the note created is assigned a note ID which is automatically ascending in the database. The user ID and module ID act as foreign keys to link this table to the user table and module table. The user will be able to identify which module they are creating a note about when the notes are displayed on the webpage.

Timetable Table

Included in the timetable table is information regarding where each user should be at a given time. To allow each entry in this table to be unique, an ID is placed for each row. Period ID, day ID, module ID and course ID act as foreign keys to link the timetable table into the period, day, module and course tables respectively. Other information stored here includes the module title, location and lecturer name which are to be displayed on the webpage. By using a period ID and a day ID, the developer is able to allow the modules to be scheduled through different times of the day and on different days.

Period and Day Table

To allow for different days and periods both these tables were included. The day table sets each day as a number in the Day ID from 1 to 5, while the period ID sets out times as periods from 1 to 8 from 0915 to 1615 start times and 1015 to 1715 end times. Lectures work in Ulster University in one hour blocks starting at quarter past the hour which is why the developer has defined the start times and end times in this fashion.

Implementation

Introduction

This chapter deals with how the developer used an Agile development strategy to complete the project. The details of this strategy are discussed and following from this, sections of significant code are displayed. The developer will also discuss what was learned from the prototype system which was created.

Development Tools

XAMPP: phpMyAdmin and Apache

The system will be developed through XAMPP, utilising phpMyAdmin to create a database and Apache to act as a local host for the webpage. Apache is an open source web server which will enable the developer to create a web page to be delivered upon client request while phpMyAdmin is a web based database development tool allowing the developer to create a database to hold information required to make the timetable. Both Apache and phpMyAdmin are available to use once XAMPP has been downloaded.

Programming Languages – PHP, SQL, HTML, CSS

PHP is a server side scripting language used to create websites where the integration of a database is required. As the PHP code is executed on the server and the result is returned in the browser as HTML. By using PHP, the developer can create a link between the website and database allowing information stored on the database to be displayed in the webpage and information entered into the webpage to be stored in the database. This will be useful in displaying information regarding modules once a user has logged into the webpage and allows new users to be registered for the webpage. PHP is also available to run in different operating systems, is free to download and is compatible with most server applications such as Apache which is being used to develop the webpage.

SQL (Structured Query Language) is a programming language for allowing developers to access and manipulate databases. Using SQL queries, a developer can read and write data held in a database or alter the database itself by creating new tables or dropping existing tables. Incorporating SQL into the code for this webpage, the developer can show tailored information depending on which user has logged in. For example, a specific list of modules

can be displayed for each individual username depending on the information held in the database.

HTML (Hyper Text Mark-up Language) is a language used to describe how a web page is displayed. Consisting of a series of tags the code is surrounded by, a developer can use HTML to define how the web page is built. A web browser can then read and interpret the tags to display the webpage. HTML can be developed using a simple text editor and saving the file type as HTML. For this webpage, notepad++ will be used as an editor as it is freely available to download while HTML is being written in conjunction with PHP and SQL.

HTML was designed to describe the content of the webpage, whereas CSS (Cascading Style Sheets) formats the HTML page, allowing the layout of multiple web pages to be controlled at once. The colour and style of the web page can be defined using CSS, and minimises the amount of code written to achieve desired effects in the web page and as such is advantageous to use.

Agile Development

This project will be completed using Agile Development. By using this Agile method, the web page can be created in an organised manner and subject to constant testing as is the standard practice in Agile Development. This is in contrast to the Waterfall method in which testing is only completed in the penultimate phase.

One of the key drivers in the Agile methodology is gaining feedback on the work that has been completed, to ensure that the build is properly tested and done so within short iterations. By focusing on short development cycles in which the product is tested, it is ensured that the product created is to a high standard. It is also of note that all aspects of development in Agile are continuously revisited leaving room to move or change pieces of the project while the process as a whole constantly moves forward. These short iterations of development are referred to as Sprints. By looking at this project in an Agile Methodology, instead of one large deliverable which would be the case in a Waterfall process, there are smaller deliverables, which are created through the sprints. Below is a diagram highlighting the Agile process around Sprints with deliverables launched after each cycle.

By using this agile approach the developer can create sprints out of each feature being created for the website and for the database. Each feature can be tested in house as it is being built over the course of a sprint. This development strategy also allows the user to continue to test each feature as the build progresses through the sprints. It is noted that Agile development is better suited for team builds but it is preferable to continuously test this website over the incremental style of building which suggests the testing should be completed at the end of the methodology.

Diagram twenty six shows Waterfall versus Agile methodologies.

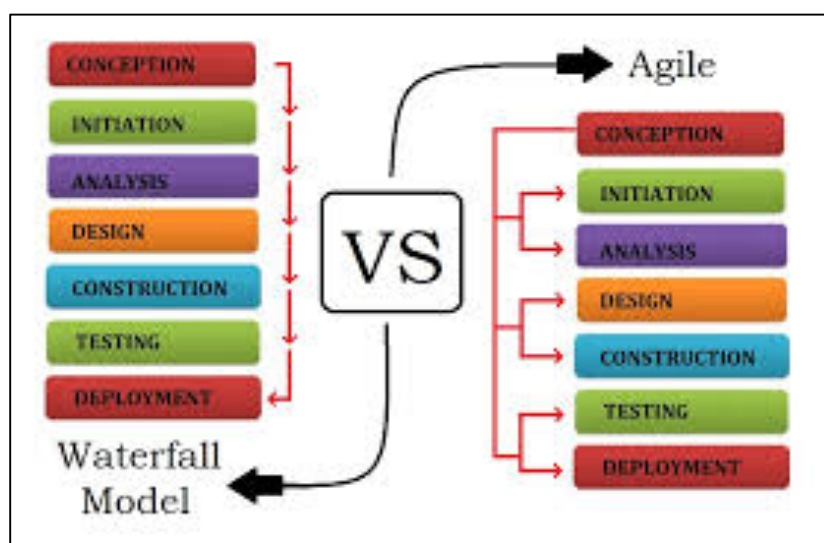


Diagram twenty six

Based on the Agile methodology, the developer has created a list of sprints through the course of the project in order to complete by the given deadline of September 1st. Table two below summarises these sprints. In total the developer has broken the implementation of the website into ten distinct stages and will break these stages down further in the table below.

Sprint Number	Objective
1	Create a database to hold user information and timetable information
2	Create a login and log out system
3	Recreate Database to allow combine a user table
4	Separate the login for a student and lecturer
5	Display the user information on the webpage
6	For Student, display timetable table.
7	For Student, create webpage to add notes for modules
8	Create new timetable format separating out each day of week for student and lecturer
9	For Lecturer, display module specific timetable
10	For Lecturer, display entire timetable and register of students

Table two

Sprint 1

Sprint one of this project involved the lecturer creating a database which would hold the information for students and lecturers to log into the website and also information on the timetable itself. The database was built using phpMyAdmin downloaded by the developer as part of XAMPP. Sprint one lasted from 18/04 to 30/05. With the database seven tables were created.

Sprint 2

The initial code that the developer created to was have a login and logout for the users in the database. Code was saved in the htdocs folder from the XAMPP folder in the C drive of the developer's computer. This allowed the developer to user local host to display the code created. The code was written in notepad++ and was saved as a PHP file. The code was written in PHP language and HTML language. CSS files were used throughout as a means to style the website. Code one below shows part of the login and log out sequences.

```
$query = mysql_query("select * from usertable where password='$password' AND login ='$login'", $connection);  
$rows = mysql_num_rows($query);  
if ($rows == 1) {  
    $_SESSION['login_user']=$login;  
    header("location: profile2.php"); // Redirecting To profile Page  
} else {  
    $error = "Username or Password is invalid";  
}  
mysql_close($connection); // Closing Connection  
}
```

Code one

From the code above, an SQL statement is used to select all information from the database where the username and password rows are equal to that of the data entered by the user. If there is only one row, meaning the information entered is correct, the code will progress to the next screen. If the information is incorrect the error will appear indicating that the Username or Password is incorrect. Sprint two lasted from 02/05 to 15/05.

Sprint 3

After the original prototype was created, the developer realised that the database had to be altered to allow separate log in's for users. The issue was resolved by creating an all-encompassing table for students and lecturers indicating the user type as either one or two. This sprint lasted from 20/06 to 26/06. The database was still built using phpMyAdmin.

Sprint 4

This sprint was used to create a separate login for the lecturer and the student user. The code was written using PHP, SQL and HTML. The code as shown below, Code two, uses an if statement. If the user type is one, the website progresses to the student screen. If the user type is 2 the website progresses to the lecturer screen. The fourth sprint lasted between 27/06 and 03/07.

```

if ($row['userType']==1){
    $userID=$row['userID'];
    $firstName=$row['firstName'];
    $surname=$row['surname'];
    $userType=$row['userType'];

    $_SESSION['login_user']=$login;
    $_SESSION['userID']=$userID;
    $_SESSION['firstName']=$firstName;
    $_SESSION['surname']=$surname;
    $_SESSION['userType']=$userType;

    //direct to the user homepage because they are number 1 which is a student

    header("Location: profile2.php");
}
else if ($row['userType']==2){
    $userID=$row['userID'];
    $firstName=$row['firstName'];
    $surname=$row['surname'];
    $userType=$row['userType'];

    $_SESSION['login_user']=$login;
    $_SESSION['userID']=$userID;
    $_SESSION['firstName']=$firstName;
    $_SESSION['surname']=$surname;
    $_SESSION['userType']=$userType;

    //direct to the user homepage because they are number 2 which is a lecturer

    $_SESSION['login_lecturer']=$login;
    header("Location: lecturer.php");
}

```

Code two

Sprint 5

Sprint five was used to show user information on the header of the webpage. This showed the developer that the login was working as the correct user information was being displayed on the webpage. This code used a session variable from the user's login and HTML code to display a welcome message and which course the user is registered for. Code three below highlights this. This sprint took the developer from 04/07 to 12/07.

```

<b id="welcome">Welcome: <i><?php echo $login_session; ?></i></b>
<b id= "heading"> You are registered for: <?php echo $show_course; ?></b>

```

Code three

Sprint 6

The sixth sprint was used to display the timetable as a table. The developer at this time was unable to fully replicate this using HTML, PHP and SQL. This sprint lasted from 13/07 to

18/07. The developer decided to create an additional sprint to devise a new way to display the user's timetable.

Sprint 7

For students to add notes to modules, the developer created code in PHP, SQL and HTML. The developer decided that the user should have a drop down menu which would allow the user to select their module from a drop down menu. This menu would be populated with modules only from the user's course list. The note would then have to be saved in the database and subsequently displayed automatically on the users screen. This was completed with the following segments of code.

```
<?php
$mysqli = new mysqli("localhost", "root", "stetson12", "timetable");

$query="SELECT moduleID
        from moduletable
        INNER JOIN coursetable
        ON moduletable.courseID = coursetable.courseID
        INNER JOIN usertable
        ON coursetable.courseID = usertable.courseID
        WHERE userID = '$show_userID'";
$result = $mysqli->query($query);
?>
<select name="moduleID" >
    <?php
    while ($row = mysqli_fetch_assoc($result)) {
        echo '<option value="'. $row['moduleID'] .'">' . $row['moduleID'] . '</option>';
    }
?>
```

Code four

This section of code, Code four, shows how the drop down menu was displayed. The SQL statement selects the appropriate module code from the database and the while statement takes the module IDs and inserts them into the drop down menu as clickable options.

```
if(isset($_POST['submit'])) {
    $moduleID = $_POST['moduleID'];
    $note = $_POST['note'];
    if($note != '' || $moduleID != '') {
        //Insert Query of SQL
        $query = mysqli_query("INSERT INTO notetable(userID, moduleID, note) VALUES ('$show_userID', '$moduleID', '$note')");
        echo "<br/><br/><span>Data Inserted successfully...!!</span>";
    }
}
```

Code five

The above segment of code, Code five, shows how the note written by the user is then imported into the database after the user clicks the submit button. This sprint lasted from 19/07 to 24/07.

Sprint 8

Sprint 8 set out to find a new way to display the timetable for students and by extension lecturers. The developer decided that a list of the times of the day would resemble a diary approach which should still be beneficial to the user. With this in mind, the developer would also have to separate out the days of the week in the website. The following code demonstrates how the developer created this using HTML, PHP and SQL.

```
$sql = "SELECT *  
FROM timetabletable  
INNER JOIN coursetable  
ON timetabletable.courseID = coursetable.courseID  
WHERE courseName = '$show_course';  
$result = mysqli_query($conn, $sql);
```

Code six

Initially SQL code is used to select the module information for the logged in user.

```
//create an array here that holds the days of the week. Have array index 0 blank and match up the rest to the value of d  
$days = array("nothing", "<tr><th>Monday:</th>", "<tr><th>Tuesday:</th>", "<tr><th>Wednesday:</th>", "<tr><th>Thursday:</th>", "<tr><th>Friday:</th>");  
  
//Print out the top of the array to display the times  
$periods = array("<tr><th>Day</th>", "<th><br>09:15 \n</th>", "<th><br>10:15 \n</th>", "<th><br>11:15 \n</th>", "<th><br>12:15 \n</th>",  
"<th><br>13:15 \n</th>", "<th><br>14:15 \n</th>", "<th><br>15:15 \n</th>", "<th><br>16:15 \n<br><br><br><br>\n</th>");
```

Code seven

Arrays are then created to hold the days of the week and the times of day.

```

//for days 1-5 (mon-fri)
for($d = 1; $d < 2; $d++){

//echo the value of $d in the days array (1=mon 2=tues etc etc)
    $printday = $days[$d];

    echo "<tr><td>" . $printday . "</td>";

    for($p = 1; $p < 9; $p++){

        $printperiod = $periods[$p];
        echo "<tr><td>" . $printperiod . "</td>";

        unset($course_row);
        foreach($courses as $course){
            if($course['dayID'] == $d && $course['periodID'] == $p){
                //if the course matches that day and the specific time then print the module name and the location
                $course_row = $course['moduleTitle'] . " " . $course['location'] . " " . $course['lecturerName'];
            } //if
        }

        if(isset($course_row)){
            echo "<td>" . $course_row . "</td>";
        } else {
            echo "<td>&nbsp;</td>";
        } //else
    } //for 2
}

```

Code eight

A for loop then uses the arrays created to pick out the relevant module information for each day of the week. If the day of the week and the period of time match the information held in the database, the website will display the module title, the location the module is being taught and the lecturer name. To account for only one day at a time, the for loop counts only one day at a time, from \$d = 1, \$d < 2, then \$d = 2, \$d < 3 and so on until \$d=5, \$d <=5. This sprint lasted from 25/07 to 28/07.

Although different from the original design plan the developer feels that this approach still fits the original aims and objectives and meets the functional requirements. The information that is displayed still is accurate for the user and the timetable is automatically generated. The only user input now for retrieving the timetable is selecting the day which they wish to view. The navigational bar in the header will allow student to move seamlessly between the days of the week which still is in scope of fixing the original problem that there are too many menus to sort through before the correct timetable information can be found by the user.

Sprint 9

Lecturers differ from students in that they only require knowledge of the modules they teach and not the entire list of modules in the course. Although it is a later sprint to show the entire list this sprint was concerned in only showing specific modules which the lecturer teaches. To create this, the developer uses SQL, PHP and HTML code saved in a PHP file. The following SQL code shows how the developer selected only the information required for the lecturer to see.

```
$sql = "SELECT * FROM timetabletable  
INNER JOIN moduletable  
ON timetabletable.moduleID = moduletable.moduleID  
INNER JOIN usertable  
ON moduletable.user_ID = usertable.userID  
WHERE userID = '$show_userID'";
```

Code nine

Depending on the user logged in, the SQL statement will select only the modules which the lecturer teaches, i.e. the relevant user ID in the module table in the database. Below Code ten displays the same process to show the lecturer timetable as to show the student timetable as described above. Below the difference can be seen as the lecturer will only need to know the module location and the module title. This sprint lasted from 29/07 to 03/08.

```

//for days 1-5 (mon-fri)
for($d = 1; $d < 2; $d++){

    //echo the value of $d in the days array (1=mon 2=tues etc etc)
    $printday = $days[$d];
    echo "<tr><td>" . $printday . "</td>";

    for($p = 1; $p < 9; $p++){

        $printperiod = $periods[$p];
        echo "<tr><td>" . $printperiod . "</td>";

        unset($course_row);
        foreach($courses as $course){
            if($course['dayID'] == $d && $course['periodID'] == $p){
                //if the course matches that day and the specific time then print the module name and the location
                $course_row = $course['moduleTitle'] . " " . $course['location'];
            } //if
        }

        if(isset($course_row)){
            echo "<td>" . $course_row . "</td>";
        } else {
            echo "<td>&nbsp;</td>";
        } //else
    } //for 2
}

```

Code ten

Sprint 10

The final sprint was used to display the additional information for the lecturer. The information displayed was the entire timetable and the list of all students registered for the same course as the lecturer. As described above, the same process was carried out to display the entire timetable as for students, the main difference being with the SQL statement leading into the for loop and arrays.

```

$sql = "SELECT *
FROM timetabletable
INNER JOIN coursetable
ON timetabletable.courseID = coursetable.courseID
WHERE courseName = '$show_course'";
$result = mysqli_query($conn, $sql);

```

Code eleven

The SQL statement does not take into account the user ID, instead pulling all the information for the course the user logged in is registered for.

To show the complete list of students the developer used the following code.

```
$sql = "SELECT *
FROM usertable
INNER JOIN coursetable
ON usertable.courseID = courseTable.courseID
WHERE userType = '1'
AND courseName = '$show_course'";

$result = $conn->query($sql);

echo "Here are the students listed for $show_course <br>";
if ($result->num_rows > 0) {

while($row = $result->fetch_assoc()) {

    echo    $row['firstName']." ". $row['surname']. "<br>";

}

}
```

Code twelve

The SQL statement takes all the user type one users from the user table that is the student users. If the number is more than zero, the code above will continue to output all student users which have the same course as the logged in lecturer. This sprint lasted from the 08/08 to 12/08.

Summary

This implementation shows how the developer used their time to build the system. The report breaks down how each sprint created a new feature of the system. The developer also showed how they learned to adapt the design and still is able to meet the original aims and objectives set out at the beginning of the project.

Testing and Evaluation

Introduction

Testing was carried out throughout the project by the developer to ensure that each function was fit for purpose as and when it was created. Thorough testing ensured that each function was efficient in its purpose and designed to be intuitive for the end user. This phase included also cross browser compatibility testing to show that each function operated as expected in the overall website. Testing was also carried out in a focus group in order to receive feedback on the finalised project and giving the developer indication if the final project was fit for purpose.

This chapter focuses on the testing that was carried out during the project and aims to demonstrate how the testing was appropriate, thorough and ensured the end system was in line with the original specification.

Testing

Testing was carried out by the developer as each function of the website was developed. This in house testing was carried out determining if each function completed the task it was intended for. All functions and testing was recorded in the tables below, allowing the developer to track the progress of the project and also catalogue issues which arose during development.

In the development of this project, two user types have been identified. A user can either be a Student or a Lecturer. Testing was completed for each of these users under separate tables as their functions differ slightly. A student for example requires visibility to an entire timetable, which can include three modules per semester. On the other hand a lecturer only requires seeing the modules which they teach not the entire course they lecture for. Consequently, if the timetable shows three modules being taught on a Monday, the student will need to know the times of each class, whereas the lecturer only requires the times for the module they teach. The tables can be seen in the figures below.

It should be noted that the prototype database created in June was not fit for purpose. Originally a table was created to be used for the Lecturer and Student individually. After further analysis since June, the developer has created a new database using instead a user

table in which the users are defined as either a Student or Lecturer by assigning their type as either one or two. This ensured that when logging into the system an SQL statement could find this user type and direct the user to the correct page.

Functionality Testing

Student Functionality	Does it meet Specification?	Issues, if any	Solutions	Notes
Login	Yes	No	N/A	Username and passwords are stored in database. SQL statements match the username and password to allow student to login
Logout	Yes	No	N/A	PHP file closes connection with server login out the current user when log out button is clicked
Display student Information	Yes	No	N/A	When the student user logs in, their name and course is displayed in the header section, indicating to the user they have logged on successfully
Display Timetable	No	Timetable does not display in table form	Create a new way of displaying students timetable, see display weekdays	As each individual day is displayed as a list rather than a table, the developer decided that it would be more appropriate to separate

				out each day as an individual page rather than a complete list
Display Weekdays	Yes	No	N/A	Each day is now shown as an individual page, allowing the student to see their timetable broken down by each day of the week. Information provided includes the day, time and location of each module as well as the module title and the lecturer name
Student Create Note	Yes	No	N/A	Each student is able to create a note about a module which is saved in a table in the database
Student Note displayed	Yes	No	N/A	Notes created by the student are displayed along with the module the note corresponds to

Table three

Lecturer Functionality	Does it meet Specification?	Issues, if any	Solutions	Notes
Login	Yes	No	N/A	Usernames and passwords are stored in database. SQL statements find the lecturer username and password and identify the user type as 2, and proceeds to login as lecturer
Logout	Yes	No	N/A	The server connection is closed and the user is brought back to the login page
Display Lecturer Information	Yes	No	N/A	When the lecturer logs in, their name and course is displayed in the header
Display Timetable	No	Timetable is not displaying in table format as expected	Create a new way of displaying their timetable, see display weekdays	The timetable is being displayed as a list rather than a table as expected. Instead of a table, the developer has decided to create separate pages as days of the week to reflect a diary more than a timetable
Display Weekdays	Yes	No	N/A	Days of the week are now separated out into individual pages, allowing the lecturer to select the

				day of the week and view the modules they are scheduled to teach that day
Lecturer Show Entire Timetable	Yes	No	N/A	If a lecturer requires knowledge information about the course they teach, then they can view a full display of the timetable. This view will show the day, time and location of each module being taught and the lecturer and location of each module
Lecturer Show Students Registered For Course	Yes	No	N/A	A lecturer may require a full listing of who is registered for the course in which they teach. This system allows the lecturer to see the complete list of students who are registered for their course and subsequently the modules they teach

Table four

Browser Compatibility Testing

It should be noted that the developer used Mozilla Firefox to test the project throughout the build. After the website was completed, the developer then tested that the functions for both users were compatible across Internet Explorer, Google Chrome for Windows and Safari for iOS. The results are shown in the table below.

Function / Browser	Firefox	Internet Explorer	Google Chrome	Safari
Login	✓	✓	✓	✓
Log out	✓	✓	✓	✓
Layout consistent	✓	✓	✓	✓
Colour consistent	✓	✓	✓	✓
Navigation consistent, buttons continue to work as expected	✓	✓	✓	✓
Display of information correct	✓	✓	✓	✓
Student can create and view comments	✓	✓	✓	✓
Lecturer can view class list and full timetable	✓	✓	✓	✓

Table five

As can be seen above, all the criteria have been met during the browser testing phase. This means that the website which was built would be suitable for students or lecturers using any of the above browsers.

Focus Group Testing

After the developer had completed the in house testing for functionality of features, it was necessary to complete a focus group test. The developer created a simple questionnaire and provided it to a sample of students and lecturers to test the usability and intuitiveness of the system. In total, ten users tried the system, three lecturers and seven students. The responses received would be used to critically evaluate the system as a whole and indicated

whether the system would be usable during a university semester. The questionnaire which was given to the focus group can be seen as appendix one at the end of this report.

The first three questions were asked on a scale from one to five, one being the best and five being the worst. The next six questions were yes or no answers testing various criteria in the system.

Results

From the focus group testing, all of the users agreed that the design and layout of the page was attractive and consistent with seven scoring *one* in the category and three scoring *two*. All users also agreed that the website was simple to navigate through, with eight scoring *one* and two scoring *two*. All users scored the information provided again between *one* and *two*, this time nine scoring *one* and one scoring *two*. From these first three questions, the developer can establish that users can freely navigate through the system and were able to identify which classes should be attended. This addresses one of the main issues with the system already in place, which was time consuming to use and difficult to navigate through.

The next questions were designed to receive comments from the users to get their feedback on the system. An overall consensus was that the information provided was clear and concise and the additional features for students and lecturers were useful and intuitive. One lecturer user noted

"I like knowing the students full timetable as I can schedule meetings in between classes they have"

A student user commented

"I'm happy to see a new way of getting my timetable from uni, the older system on blackboard was too difficult to use"

Another student user commented

“The note feature is simple to use and updates immediately although I’m not sure how often I would use this feature”. The same student user commentated that this feature may be a place for further development, perhaps to allow the student to delete old notes.

Another user commentated

“The website does provide me with my timetable but I would like to have some way of getting lecturers information as well, perhaps their email address or where their office is”

After the results were analysed, all participants reported no errors while testing and that all of them would use the system throughout a semester. This is a good indication of the usefulness of the system. Based on the focus group testing and from the testing during development the developer can say that the functional requirements of the system have been met in that the user can log onto the system and have their timetable for the semester displayed, and the system is more intuitive than before based on the comments from the testers. The website will automatically generate the timetable based on the user who logs in without having to filter dropdown menus or select a course which they are studying, or a school they are studying at.

Conclusion

This chapter aims to summarise the project and critically examine the final product. In doing so, this chapter will look at how the project developed and how the in house testing affected the development of the product. A set of recommendations will also be outlined for future development and lessons learned while developing the product will be discussed and clarification on how these lessons would affect development of a similar product in future.

Project Summary

Initially, a problem was found with the current system in place for students to find their timetable on the Ulster University website. Research was conducted into the Ulster University website to understand how the current system works and found that it required users to filter through various menus and select their course or modules, along with the department of study they belonged to and the course they had been registered for. This process could be time consuming and awkward to use. Obviously students require the correct timetable to undertake their studies and attend lectures, so it is paramount that the student navigates these menus correctly to gain their own timetable. The solution was aimed at creating a website that would be user friendly, but also generate a timetable for the user with minimal input.

At the beginning of each semester a paper copy of this timetable is also given out to students, although this means is deemed unreliable as the sheet can be lost or the timetable could be updated after the sheet has been given out and will need to be replaced. A copy of the timetable online was deemed as the best approach as this can be updated easily with minimal hassle for students as once they log on to their timetable it will have been automatically updated with the new information.

The website was designed to follow the current Ulster University format, including colours and images in order to feel like it was part of the Ulster University's website. This was to give the user a sense of familiarity and also as this system was created in order to deal with the existing problem within Ulster University. Screenshots were created in the design to

visualise how each screen should look, including colour schemes and positioning of images and text boxes. It was originally designed with a table layout in mind, but through development this was altered to provide a view more in line with a diary separating out each day as a separate page in the website. A database layout was comprised to show where the timetable and user information would be held. This database was subsequently updated to allow both lecturers and students log into their timetable later in the development cycle. Story boards were included in the design phase to show how a user might navigate through the website. An overall theme of the website in the design phase was usability and intuitiveness, giving this system the edge over the system currently in place.

To achieve this final design an agile framework was used, with each feature acting as a sprint according to the frameworks methodologies. The advantages of this allowed the developer to continue to build the website and test as each feature of the website was completed. These sprints included building the database, followed by creating a login for each user and a log out function. Then the information which was to be displayed for each individual user was added to the website. A sprint was used to create the timetable that the user would be able to see, followed by sprints to create and show notes for student users and show the class register for the lecturers and the complete timetable for the course the lecturer belongs to. Given the use of an Agile framework, the developer was able to change the database after the initial prototype was built to accommodate both lecturers and students to login with only minimal impact on the timeline of the project.

The developer used the storyboards and mock up sheets to create the webpage in the steps outlined above and used the following tools to develop each feature. To build the system, the developer used phpMyAdmin to house the relational database and allow local host testing of the website. This was downloaded as part of the XAMPP application. PHP files were used to write html, php code and SQL statements along with css files to style the webpage. These languages allowed the developer to create a connection between the details of users held on the database and the information on the timetable to the website the user would log on to.

Once the website was built, the developer produced a questionnaire for a focus group of ten users to allow for a thorough test of the system. The results returned were positive indicating that the system built was fit for purpose and was a solution to the original problem that was a time consuming and difficult to navigate web page existing already on the Ulster University website. In line with the Agile methodology, testing was also carried out in house by the developer after Sprints to ensure each feature created was fit for purpose. Testing was also carried out across several browsers to ensure that when a user logged in with different browsers the experience of the website would be consistent.

Objectives and Achievements

The project was original broken down into development steps which the developer feels have been achieved. These include:

5. Research was carried out at the beginning of the project to understand the problem and in order to devise a solution which would be fit for purpose and to be used by both lecturers and students in Ulster University.
6. Functional and non-functional requirements were developed to fit the needs of the end users.
7. A solution was coded in line with the requirements set out in the design of the project, ensuring intuitive navigation and accurate information presented.
8. The system was appropriately tested both in house and through a focus group ensuring the system is fit for purpose and meets the needs of the user overcoming the original problem.

Recommendations

Based on the testing from the focus groups and over time as the development was being completed, the developer has created a list of recommendations that could be used for future development.

- One point made during testing was the further development of the students note making feature. This could be expanded upon to allow students to delete notes that have been made and perhaps allow for notes to be date stamped or have reminders which are sent to the users email address.

- A feature that could also be created for future is a calendar function. With this function users could see further ahead in the year rather than see a solitary week in the semester. With this function holiday dates and reading weeks could also be incorporated into the website allowing better planning for lectures and revision.
- A feature also recommended for the system would be for the lecturer to communicate through the website to students in the class. With this feature lecturers could set deadlines for the students on the website which may alert students of coursework due to be finalised or a lecturer cancelling a class on a particular day.
- A further point is the layout of the timetable. Originally planned to be completed as a simple table, the website now contains distinct days and times. The website in further could build in this table as this is the preferred format of the system already in place which might feel familiar to students and lecturers.
- Some development could be included on the database to allow lecturers to teach in more than one course. At present lecturers can lecture across several modules although can only be involved in one course. An example of this is a lecture who teaches in Business can be part of composite courses such as Business with IT or Business with Finance, but this wouldn't be acceptable in the current database layout.

Summary

The personalised timetable was designed for Ulster University Students to retrieve their timetable in a timely manner without having to navigate through webpages of menus, selecting courses, campuses and schools but instead have a timetable generated just by logging into the system. This report documents the developer's process to overcome this issue by firstly researching the underlying problem, analysing the solution and designing a webpage which offers users a more intuitive system to use. The system was successfully developed and tested in various manners throughout its life cycle.

The project achieved the goals it set out with originally as a user, either student or lecturer can log on and have their timetable for the semester displayed without going first through separate web pages to select which course, year or campus the user is based at. The developer deems this as a success based on the feedback gained from the focus group testing where all users agreed they would use the system throughout a semester.

References and Appendix

Broadcast O'Reilly [ONLINE] <http://broadcast.oreilly.com/2010/02/nonfunctional-requirements-how.html> [Accessed 12/01/2016]

Euphemia Wong. 2016. *Shneiderman's Eight Golden Rules Will Help You Design Better Interfaces*. [ONLINE] Available at: <https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces>. [Accessed 1 August 2016].

Form Get. 2016. *PHP GET and POST Method – Tutorial*. [ONLINE] Available at: <https://www.formget.com/form-validation-using-php/>. [Accessed 6 July 2016].

Get Revising. 2016. *Get Revising*. [ONLINE] Available at: <https://getrevising.co.uk/>. [Accessed 20 July 2016].

Grade tracker. 2016. *Class Schedule Maker*. [ONLINE] Available at: <http://gradetracker.com/lab/class-schedule-maker>. [Accessed 19 July 2016].

IT Business Edge. 2016. What is SQL?. [ONLINE] Available at: <http://www.sqlcourse.com/intro.html>. [Accessed 16 June 2016].

php. 2016. What is PHP?. [ONLINE] Available at: <http://php.net/manual/en/intro-whatis.php>. [Accessed 16 June 2016].

Php [ONLINE] <http://php.net/manual/en/intro-whatis.php> [Accessed 14/01/2016]

Sdlc [ONLINE] <https://www.sdlc.ws/agile-vs-waterfall/> [Accessed 13/01/2016]

Software Testing Help. 2016. *How can a Web site be tested?*. [ONLINE] Available at: <http://www.softwaretestinghelp.com/how-can-a-web-site-be-tested/>. [Accessed 8 August 2016].

Ulster University [ONLINE] http://www.ulster.ac.uk/finance/planning/information_development/devnos2014-15.pdf [Accessed 08/01/2016]

W3Schools [ONLINE] http://www.w3schools.com/php/php_mysql_intro.asp [Accessed 13/01/2016]

w3 Schools. 2016. The language for building web pages. [ONLINE] Available at:
<http://www.w3schools.com/>. [Accessed 16 June 2016].

W3 Schools. 2016. *HTML(5) Tutorial*. [ONLINE] Available at: <http://www.w3schools.com/html/>.
[Accessed 7 July 2016].

W3C. 2016. HTML & CSS. [ONLINE] Available at:
<https://www.w3.org/standards/webdesign/htmlcss>. [Accessed 17 June 2016].

Appendix One

Tested Criteria	1	2	3	4	5
Was the design and format of the website consistent and attractive throughout?					
Was the website simple and intuitive to navigate through?					
How useful was the information provided?					

Tested Criteria	Response Yes / No	Comments
Is the information provided accurate and complete?		
If Student - Did you find it useful to be able to create notes for each module?		
If Lecturer - Did you find it useful to know the entire timetable and the list of students registered for the course		
Were there any errors during testing?		
Any additional comments or ways to improve the system?		
Would you use this system through a semester?		