Digital Lab Marking System

Heriot-Watt University

Deliverable 1: Final Year Dissertation

MEng Software Engineering

Lewis Francis McNeill
supervised by Peter J King

April 3, 2017

Declaration

I, Lewis Francis McNeill, confirm that this work submitted for assessment is my own and is expressed in my own words. Any references, made within it, of the works of other authors in any way (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Lewis McNeill

Date: April 3, 2017

Abstract

The aim of this dissertation project is to replace the current system for the marking of computer labs with a new digital system. This will enable lecturers to create a marking scheme online. Lab helpers will select the student they are marking and the marking scheme will then be loaded, marks will be entered and then made immediately available to both student and lecturers to view. It will also provide useful statistics for both student and lecturers.

Contents

7   Testing   23

8   Evaluation   24

9   Discussion   25

References   26

List of Figures

List of Tables

1   Introduction

The current system for marking of computing science labs is to use multiple lab helpers, each given a list of students and the marking scheme for them. Generally marking schemes are a selection of tasks students must have completed and lab helpers tick them off when this has been achieved. The biggest problem with this part of the marking is the length of time it takes lab helpers to locate the students on the list. This causes frustration with increased waiting times for students. Multiple other issues can also arise from this: students can be marked by two helpers and obtain different grades from both; the lab helper omits to tick off a completed task; they assign marks for the wrong student on the sheet or simply they misplace the actual marking sheet.

After the lab helpers have completed their marking, the sheets are provided to the lecturer who collates them together into one spreadsheet to calculate marks. After that it is entered it into vision. This too can cause its own set of problems-the chances of transcription errors are increased as it is possible for the lecturer to misread marks when they are transferring them across. The lecturer may not enter the marks immediately into the spreadsheet increasing the chance that a marking sheet goes missing, and finally this system means that students are having to wait even longer to receive their results.

The objective is to develop a system that will reduce and hopefully eliminate the problems of the current system. Along with this, it should hopefully reduce the amount of time taken to mark students work and therefore speed up labs in general. It should also enable students to see their grades immediately, allow lecturer to see the result of the assignments as they are being marked and make marking quicker for lab helpers.

2 Aims and Objectives

2.1 Aim

The aim of this dissertation is to design and implement a system for the digital marking and analysis of computer labs and to help improve the speed at which they are marked. The system will also provide useful statistics for both lecturers and students.

2.2 Objectives

- Simplify the way that labs marks are currently processed.

- Allow lecturers to create marking schemes on-line that lab helpers can access.

- Lab helpers can mark students in labs using marking schemes.

- Lab helpers able to mark labs using an on-line application.

- Allow students to see the mark they achieved from the lab instantly.

- Provide useful statistics and graphs for lecturers and students.

- Provide different views for student, lab helpers and lecturer.

2.3 Scope

3 Background Information

This section contains the summaries of literature relating to the topic and should help to create a context for the development of a digital marking system. It will cover what marking systems that are currently being used, what current digital marking systems actually exist and why they are an improvement. Along with this it will also cover how to control what users are allowed to see, as well as explaining systems for creation of custom website forms, and finally it will cover the graphical displaying of statistics.

3.1 Web Applications

3.2 HTML

Hypertext Markup Language (HTML) is a markup lanaguage that is used to create the structurer of a website through the use of elements which are represents by tags (<html>, <href>, <a>).

3.3 Cascading Style Sheets

Cascading Style sheets (CSS) is used to change the properties of elements created using HTML through the changing of these properties developers can change the look and response of a website.

The idea behind Cascading Style Sheets is that

Then it comes to websites there can be multiple different Cascading Style Sheets being used.

As part of this project I will be using CSS 3 which at time of write (2017) is the currently agreed standard [5].

3.4 Java-script

3.5 PHP

PHP stands for Hypertext Preprocessor and is an open source scripting language which can be embedded into HTML. Unlike HTML which is sent to the user and process by there computer, PHP scripts are processed by the server HTML is generated sent to the user.

It currently has two major version which are version 5.x and 7.x

For this project I will be using version 5.6.25 as version 5 is still the most common version of PHP and allow for easy installation of the lab marking system into the universities server.

3.6 Marking Systems

3.6.1 Lecturer Based

The way lecturer based marking works is that students complete their assignment, the lecturer or tutor marks it and provides result in a timely manner with useful feedback which can be majorly important in helping students improve their skills [20].

The advantage of this style of marking is that students can obtain useful feedback from their lecturer that can help improve their learning. An article study [12] found that of the students, when surveyed 82% agreed to the question "I pay close attention to the comments I get" in response to assignment feedback.

A downside to this style of marking is that as the number of students increases on courses the amount of time required to mark assignments consequently takes longer and in some cases this can actually cause marked assessments to be scrapped completely due to the amount of time taken to give feedback to students [3].

3.6.2 Peer Based

To cope with increasing class sizes some courses are beginning to move towards peer marking. Peer marking system works by having students assess each other and in some cases the students produced their own marking criteria [17].

This style allows students to gain experience in evaluating other people's work, which some graduates feel is a necessary skill to possess. [15]. Peer marking also deals with increasing amounts of students very well- this is because as the number of students increases the number of markers also increases!

Peer marking however has its own set of problems - for example, "Students may have a less well developed sense of the criteria compared to the lecturer which could lead to a lack of reliability of

student marking." [17].

3.7   Digital Marking Systems

3.7.1   Reasons for Digital Marking

Digital marking systems are designed to mirror the current paper based marking systems but with the advantage of the electronic environment [11]. These systems help to reduce the increasing workload caused by more and more students taking courses. Along with this it also allows administrative tasks associated with coursework to be automated enabling more time for other tasks.[14].

For students digital marking is great as it allows for quick feedback as the assessor is able provide students feedback immediately after they have written it up, instead of having to wait for a class to receive it. In one study[7] they found that 78% of students would like get their feedback electronically .

According to the highlighted article [8] plagiarism is on the rise amongst student. This is where digital marking can help to reduce plagiarism as the programme can do what a human marker cannot. They can compare a submission thousands of documents and judge if a person has plagiarised. They can also help to show patterns in assignments and marks that normally might go unnoticed.

3.7.2 TurnItIn

There currently exists an on-line electronic plagiarism system called TurnItIn, [22] currently being used by many universities around the world. It allows students to upload their essay assignments online. It then checks for plagiarism in the document by searching the internet and using a large database of documents. After it processes the document it assigns a plagiarism percentage and highlights any areas that were plagiarised. Lecturers can then login and view all the submitted documents and mark them .

Current research highlighted [7] conducted a questionnaire and found that students felt that the system was easy to use and more convenient than having to provide paper copies. It also found that 50% of students strongly agreed and 33.3% just agreed that they preferred to have their grade shown online rather than have a cover sheet.

3.7.3 BOSS System

The BOSS system was developed at the University of Warwick to help deal with their problem of having too many students for the number of staff and yet wanting students to have accurate and quickly available feedback [13]. It is an electronic submission and assessment system created to allow computing science students to submit their programming assignments and have them tested and marked online [14]. The system is not designed to remove human markers completely, instead simply "assist the instructor in achieving a quicker, more accurate and more consistent assessment of programming assignment"[13].

When a file is first submitted it is run through a plagiarism check to make sure that the submission is actually the students own work. It also checks that the submission passes pre-set tests to make sure it works. After this it goes into the evaluations stage, since evaluation attributes of code can be very subjective what the second step does is generate metrics about the submitted program. Some of these metric are a number of comments and percentage of methods declared [14], which will help human markers evaluate the submission quicker.

3.8　User Access Views

3.8.1　Social Media

Controlling the view that users have, based on their access level, is common practice. Social media websites for instance allow users to limit what others can see, through the use of a privacy setting [21]. This means that another user's view is determined by the access level they are given for example, a user that is a friend will have a higher access and be allowed to see their whole feed, while an other users access may only allow them see the profile name.

3.8.2　Database Controlled Access

The patent highlighted [19], describes a system of limiting user web page access through the use of relation databases. The system would work by using two databases; one would hold a list of all the url's and associated access level, while the second database would hold all the user id's along with their assigned access level. When a user requests a webpage, the access level for that webpage and the user are looked up. If the users do not have the appropriate access they are denied permission to load the page and depending on implementation may be redirected to another webpage. The design of this system is well suited for scalability since no matter how large the two datasets are only one piece of data is need from each database to confirm whether a user is allowed access.

3.9　Custom Input Forms

3.9.1　SurveyMonkey

Survey Monkey [10] is an example of custom web forms being created by users. Founded by Ryan Finley in 1999 Survey Monkey enables users to create their own surveys and easily distribute them. It builds the surveys by letting the user select the contents of the question and what the response type will be: The user can also decide if the responses are completely anonymous by default and the participants ip address is stored when they complete the survey. The users can continue to add as many questions as they would like, even after the survey is initially created. After designing the survey the user chooses how they would like to have their survey distributed. The available
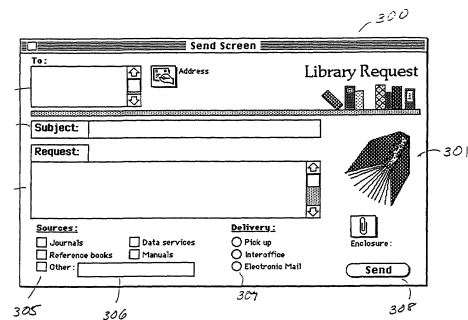
options that can be selected are a web link, social media, email or embeddable on a website [23] .

When participants complete the survey their results are immediately stored and the results of the survey are visible to the user by login into their account on surveymonkey. They can choose to look at the responses individually or look at metrics about how participants responded.

3.9.2 Customizing Forms In Electronic Mail Systems

A patent [6] describes a process for user-customisable forms in an e-mail system where the administrator selects custom field types and behaviours. For example current e-mail forms have

a field for address, subject and one for the the actual message to be sent. While an example of what the patent is suggesting can be seen in figure 1, it shows the adding of additional fields allowing for wide variety of form to be created and not limit the users to use the few forms that are precreated.

Figure 1: Example Form (Patent [6])

This increased flexibility in email forms would allow for easier interpretation of messages, making responding or providing information via email a lot simpler and quicker.

3.10 Development Tools

3.10.1 jQuery

jQuery is a JavaScript library that is designed to improve and enrich the

3.10.2 Twitter Bootstrap

Twitter bootstrap is a front-end framework [1] designed to help developers create dynamical responsive websites quickly and easily. It provides pre-created cascadding style sheets (CSS)

3.10.3 PHP Unit

3.10.4 D3

D3 [2] is a javascript library, which was designed for the creation of interactive visualisations of data and was first developed in 2011. D3 uses precreated Javascript functions to create scalable vector graphics (SVG's) which are embedded into the HTML of websites."SVG is a language for describing two-dimensional graphics in XML"[9] and can have displays changed using Cascading Style Sheet (CSS).
Data-sets can also be bound to an SVG allowing for a visual way to interpret the data-set, and as the data-set changes the SVG will be changed allowing for a dynamic display.

3.10.5 CakePHP

CakePHP is an open source framework for php, it is developed to help with rapid development of web application and make them simpler,fast and less complex to build [4]. It allows the development of well structured and robust web applications [18], and its latest version is version 3.3 called "Red Velvet" it can be downloaded from (https://cakephp.org/).

The usefulness of the framework come in the form of its predefined functions, it allows for more secure code, since you cannot accidental forget to escape character from input as the functions do this anyway. Cakephp helps to improve the maintainability of systems as it is easy to understand with plenty of comments.

A downside to cakephp is its testability, compared to other web frameworks it lacks in testing tools and does not contain a testing environment [18].

4 Requirements

4.1 Requirements Analysis

To help find out the requirements of the lab marking system I preformed a requirements analysis questionnaire(Full questionnaire in Appendix **??**), the questionnaire was given to lab helpers and lecturers to help understand what they would both want out of the system.

The results of the requirements analysis was used to develop and refine the the list of requirements displayed, functional requirements can be found in section 4.2.2 and non-functional requirements in section 4.2.3.

4.2 Requirements

Requirements for the system are each given an idea depending on the type of requirement: FR for functional requirements and NFR for non-functional requirements.

Along with this, each requirement has a description stating what the requirement is and a priority is assigned using the MoSCoW prioritization method discussed in the next section (4.2.1).

4.2.1 MoSCow Prioritisation

MoSCow Prioritisation [16] was designed to be used in project that utilise agile development and have a fixed time limit in which to be completed. The way that it works is that all the requirements of the project are prioritised into four categories, these are:

- **Must:** This category of requirements make up the "<u>M</u>inimum <u>U</u>sable <u>S</u>ubse<u>T</u>" of requirements that the project must implement to be successfully delivered.

- **Should:** This category contains requirements that are not vital to the delivery of the project, but their absence can effect the project overall.

- **Could:** This category contains requirements that are wanted in the system but are less important than the "should" priority. Reasons that a requirement would become a could

instead of a should is if the amount of time take to implement the requirement outweighs the effect if will have on project.

- **Won't:** This category contains all the requirements that will not be implemented in the delivered project, but could be implemented in later development.

For this project I will be attempting to implementing all of the must priority functional and nonfunctional requirements, I will also try and implement all of the should requirements. Finally I will implement as many of the could priority requirements that I can starting with ones that best improve the system.

4.2.2 Functional Requirements

Functional requirements also include an access column which defines what users should be able to use. Some items are restricted to lecturers as some requirements should only be be usable by lecturers and lab-helpers and not by students. The table is sorted first by access level starting with widest allowed access then sorted in access order 1 - 4, it is secondly sorted by priority.

The access levels are: 1-Admin, 2-Lecturers, 3-Lab Helpers and 4-Students

Table 1: Functional User Requirements

ID	Requirement	Access	Priority
FR-01	Login to view system	1,2,3,4	Must
FR-02	Accounts created for them	1,2,3,4	Must
FR-03	Change password	1,2,3,4	Must
FR-04	Login using university ID	1,2,3,4	Won't
FR-05	Logout	1,2,3,4	Must
FR-06	Remove students from courses	1, 2	Must

FR-07	Update student accounts	1,3	Won't
FR-08	Look up students in lab	2,3	Must
FR-09	Select students from lab list	2,3	Must
FR-10	Leave comments about students	2,3	Must
SR-11	Save marks	2,3	Must
SR-12	Update marks	2,3	Must
SR-13	Delete marks	2,3	Must
FR-14	Search for student by name	2,3	Medium
FR-15	Mark student even if they are not in the system	2,3	Could
FR-16	Assign students to courses	1	Should
FR-17	Assign lectures to courses	1	Should
FR-18	Create marking schemes	2	Must
FR-19	Display generated stats	2	Must
FR-20	See submitted marks	2	Must
FR-21	Generate end of year spread sheets	2	Should
FR-22	Editing of students in class	2	Should
FR-23	Create peer marking scheme	2	Could
FR-24	Look at students stats	2	Should
FR-25	Set what parts of the marking scheme students can see	2	Could
FR-26	Update marking scheme	2	Should

FR-27	Delete marking schemes	2	Should
FR-28	Able to assign students to set labs	2	Won't
FR-29	Set penalties for late marking	2	Won't
FR-30	Able to export to vision	2	Won't
FR-31	Access Marking Scheme	3	Must
FR-32	Enter selected students mark	3	Must
FR-33	Submit student mark	3	Must
FR-34	Select the lab they are helping in	3	Must
FR-35	See current mark	4	Must
FR-36	Show different displays depending on access level		Must
FR-37	Load students current lab mark scheme		Must
FR-38	Apply penalty for late lab completion		Could
FR-39	Create a set of useful stats based on lab		Must
FR-40	Store what class student belong too		Must
FR-41	List of all students in class		Must

4.2.3 Non-Functional Requirements

Table 2 lists all the non-functional requirements for the development of the system, they are ranked in order of priority.

Table 2: Non-Function Requirements

ID	Requirement	Priority
NFR-01	All person data encrypted	Must
NFR-02	Update stats as marks are entered	Must
NFR-03	Take less than 2 seconds to generate stats	Must
NFR-04	PHP Should use prepared statements	Must
NFR-05	Website dynamically designed	Must
NFR-06	HTML, CSS and Javascript should be validated	Must
NFR-07	Should make sure inputs are valid	Must
NFR-08	Should prevent SQL Injection	Must
NFR-09	Function on a wide variety of smart phones and tablets	Should
NFR-10	Handle a large number of users without any faults	Should
NFR-11	Check passwords contain alphanumerics and have a minimum and maximum length	Should
NFR-12	Auto save marks as they are entered	Could
NFR-13	Auto record what lab help marked what student	Could
NFR-14	List all students that did not attend the lab	Could
NFR-15	Track how long it takes to mark a student	Could

NFR-16	Disability options (Increase text size, colour layout)	Won't
NFR-17	Readable by screen readers	Won't
NFR-18	Take less than 2 second to load student marking scheme	Won't
NFR-19	Group marked people	Won't
NFR-20	Retrieve student images from university system	Won't
NFR-21	Backup database regularly	Won't

5   Design

This sections is to explain all the design decision take in developing the marking system using sketches, models and diagrams and thus assist in the understanding of how the system works as a whole. The design aspect that are cover are: the user interface, database design and the main functionality required to make the system work.

5.1   User Interface Design

For the system to be useful the user interface must be easy to use and understand, along with this it will have to be functional on mobile devices as well. To achieve this I have developed mock-ups for main web pages required to make the system work, including a design for desktop and mobile.

5.1.1   Lab Results

For the system to be useful students and lecturers both need to be able to see the results of a lab. For this apart of the application there needs to be two different views depending on the type of person signed in. This is because students need to only see their own results, while lecturers need to be able to select students and see their mark. To help develop the two different displays I have created a design for how lab results should displayed for students (figure [lfm]), and another one for lecturers (figure [lfm]).

5.1.2   Lab Marking

The lab marking page is designed to allow lab helpers and lecturers to easily select which course and lab they are marking, and then be provided with list of students who can be marked.

I expect this page to be used more on mobile devices, since it easier carry a mobile device than log-in to each computer this is reflected in the mobile design shown in figures([lfm]).

5.1.3   Lab Creation

The "Lab Creation" page will be very important for lecturers as they will have to use to to create all of their labs. Given that courses tend to have a fair number of labs it is likely lecturers will

have to use this part of the system fairly often, therefore the process for creating a lab should be fairly simple. These idea are reflected in the designs shown in figures([lfm]).

In the desktop design the side bar contents have been swapped for buttons one for each of the different types of question that a lab can have. Along with this the main content area has a drop down menu to allow lecturers to select the course they would like to create the lab for, and next too that a input box to allow them to provide the labs name.

The questions themselves are designed with simplicity in mind, lecturers select what type of question they want by clicking it on the left hand side and a new question tile appears in the main content area. The question tiles state what type of question they are what question number they will be and have remove button. To add the question text lecturers just fill out the question input box, and depending on the type of question select how many marks it is worth from the drop down menu.

The mobile version is the same but with the side bar being moved to the top and the question type buttons being underneath the navigation bar.

5.1.4   Lab Manager

The "Lab Manager" page is what lecturers will use to control labs that they have already created. Functionality available from here include: the ability to delete a lab, edit a lab, set whether lab-helpers are able to mark a lab and being able to export all the results for a selected lab. The desktop and mobile designs can be seen in the figure below ([lfm]).

5.2 Database Design

For the system to work there will need to be database that stores students, labs, and marks received. The database schema I will be using for this system is shown in figure(2) below.

Figure 2: Database Schema
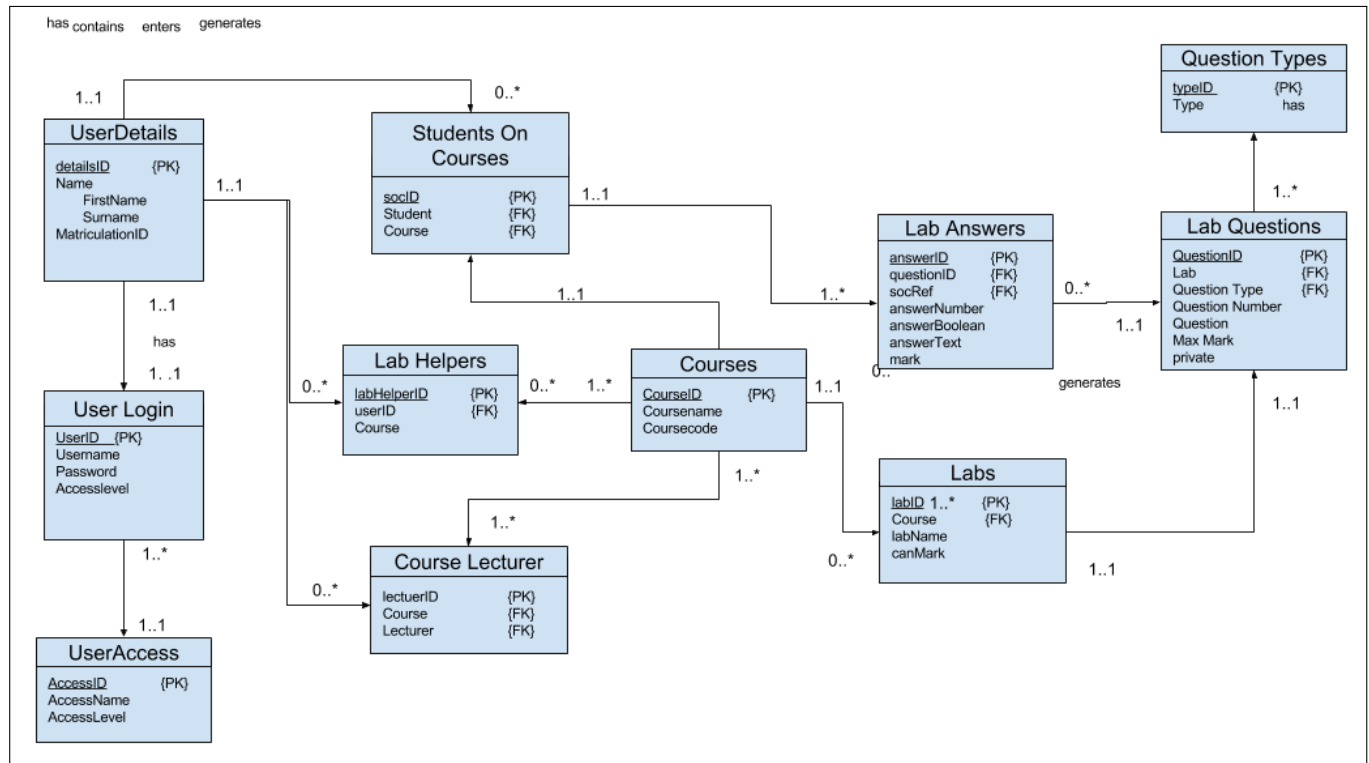
The database can be broken down into three section a shown in figure(3). The first sections ([lfm]) is designed for storing all the users of the database, it is designed to be easily changed to allow for easy integration into already existing databases. As long as users have unique ID's the rest of the database will be able to function with no changes being required.

Section two ([lfm]) is designed to storage information relating to users and courses. The Students on Courses table stores what students are on what course, The lab helpers table stores what courses a lab-helper is helping and finally the Course Lecturer tables stores what courses lectures are teaching.

The third section ([lfm]) is entirely to do with the storage of information relating labs.

Figure 3: Database Sections
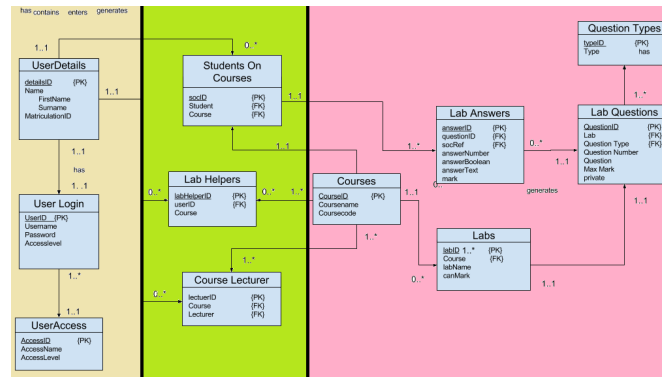
5.3   Functionality Design

6 Implementation

This section documents and explains how the lab marking system was developed and the key functionality works.

This is done through the use of screen shots showing the functionality, code snippets with explanations on what it does and how.

6.1 Database

The database schema discuss in the design section (5.2) was converted into a relational database using mysql. Figure 4 displayed below shows the database structure and the lines between tables shows their relations. There a no major changes from the original design, the only difference that occured during implementation was that some of the attribute names were changed different.

Figure 4: Implemented Database Schema
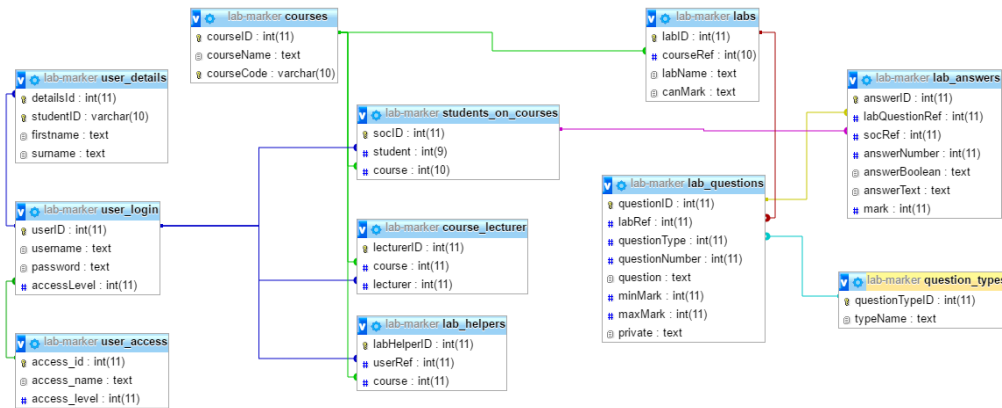
6.2 PHP Class Structure

All the PHP developed as part of this project was built on a class structure, each class contains the functionality for a part of the system. Many of the classes inherit functions from other classes and the relationship diagram can be seen in figure([lfm]).

When it came to the creation of classes and their inheritance there are limitation created by PHP. In PHP class are only able to extend one other class, the only way to inherit multiple classes

is to create an instance of it instead. This is usually done in the construct so that the class may be accessed through a private variable.

In my classes I do a combination of constructor declared instances, and instances declared in functions

6.3 Lab Creator

6.4 Marking Labs

6.5 Results Display

6.5.1 Students

6.5.2 Lecturers

6.6 Lab Management

6.7 Admin Panel

6.8 Background Functionality

This section is for the classes and functionality that are not dedicated to one part of the system, but are used by multiple different classes to provided key functionality for them to work.

6.8.1 Database Connection

The ConnectDB class is designed to create a connection to the database and allow other classes to retrieve information through the connection. The class also has a secondary function of checking if a session has been started and if not starting one, this is to allow access to sessions that may be needed for querying the databse. The full code listing can be seen below (1).

Listing 1: Database Connection Class

```
1  class ConnectDB
2  {
3      public $link;
4
```

```
5     function __construct()
6     {
7          $this->link = mysqli_connect('host', 'username','password')
              ;
8          if(!$this->link) {
9              die('Could_not_connect_to_MySQL:_' .
                 mysqli_connect_error());
10         }
11
12         mysqli_select_db($this->link,"lab-marker"); //Selects the
              Database
13
14         if (session_status() == PHP_SESSION_NONE)
15              session_start();
16     }
17 }
```

The way that the class works is, it contains one public variable called $link; when the class is initilised the constructor function called. The constructor makes a mysqli connection using the provided host, username and password details, it then sets the $link variable to the result of the connection. After this it check if the connection was successful, if it is not then a die command is thrown and an error message is shown. If it is a successful connect then is selects the "lab-maker" database. Checks if a session is already run, and starts one if it is not.

6.8.2   Security

7   Testing

This section states how the lab marking system was tested to assure the quality of the system and discover any unexpected bugs.

Testing of the system comprised of two stages, the first was the continues testing being done during development and the second being tests that were run when the system was completed to find any remaining bugs.

7.1   Development Testing

To help test functionality that I had developed I created Unit Tests usin

7.2   Final Testing

8 Evaluation

This section shows how I evaluated the lab marking system and help determine how successful I was at implementing the system, and what future improvements could be done to make the system more useful.

8.1 Usability Case Study

The way the marking system was evaluated was through the use of a usability case study (UCS), participants for the study were students, lab helpers and lecturers.

The usability case study consisted of four sections: student, lab helpers, lecturers and Mobile. Each section has specific tasks that participants were required to preform, they then evaluate how challenging the task was and provided comments about the design and any additional improvements that could be made.

In the first three sections the participant signed into the system as that particular type of user (student, lab helper and lectuer)

(Appendix **??**)

8.2 Feedback

8.3 Implementation Of Feedback

9 Discussion

9.1 Development

9.2 Limitations

9.3 Future Improvements

The electronic marking system has many improvements that can be implemented some are requirements that I was not able to implement

9.4 Conclusion

References

[1] "Bootstrap The world's most popular mobile-first and responsive front-end framework." In: (). bibtex: _bootstrap_????? URL: http://getbootstrap.com/ (visited on 03/28/2017).

[2] M. Bostock. *D3.js - Data-Driven Documents.* bibtex: bostock_d3.js_???? URL: https://d3js.org/ (visited on 11/20/2016).

[3] S. A. Brown and A. Glasner, eds. *Assessment matters in higher education: choosing and using diverse approaches.* bibtex: brown_assessment_1999. Buckingham [England] ; Philadelphia, PA: Society for Research into Higher Education & Open University Press, 1999. ISBN: 978-0-335-20243-0 978-0-335-20242-3. URL: https://books.google.co.uk/books?id=KVblAAAAQBAJ&pg=PA1&lr=&source=gbs_toc_r&cad=4#v=onepage&q&f=false.

[4] *CakePHP at a Glance.* bibtex: _cakephp_???? URL: http://book.cakephp.org/3.0/en/intro.html (visited on 11/24/2016).

[5] *CSS Current Status - W3C.* URL: https://www.w3.org/standards/techs/css#w3c_all (visited on 04/03/2017).

[6] "Customizing forms in an electronic mail system utilizing custom field behaviors and user defined operations". Pat. bibtex: holt_customizing_2006 bibtex[holder=Holt, Nick and Thomas, Steve]. May 2006. URL: http://www.google.com/patents/US7051273 (visited on 11/19/2016).

[7] S. Dahl. "Turnitin The student perspective on using plagiarism detection software". In: *Active Learning in Higher Education* 8.2 (July 2007). bibtex: dahl_turnitin_2007 bibtex[langid=english], pp. 173–191. ISSN: 1469-7874, 1741-2625. DOI: 10.1177/1469787407074110. URL: http://alh.sagepub.com/content/8/2/173 (visited on 11/21/2016).

[8] B. Derby. "Duplication and plagiarism increasing among students". In: *Nature* 452.7183 (Mar. 2008). bibtex: derby_duplication_2008 bibtex[rights= 2008 Nature Publishing Group;langid=englis pp. 29–29. ISSN: 0028-0836. DOI: 10.1038/452029c. URL: http://www.nature.com/nature/journal/v452/n7183/full/452029c.html (visited on 11/24/2016).

[9]   J. Ferraiolo, F. Jun, and D. Jackson. *Scalable vector graphics (SVG) 1.0 specification*. bibtex: ferraiolo_scalable_2000. iuniverse, 2000. URL: `https://www.w3.org/TR/2001/REC-SVG-20010904/REC-SVG-20010904.pdf` (visited on 11/21/2016).

[10]   R. Finley. *SurveyMonkey*. bibtex: finley_surveymonkey_1999. 1999. URL: `https://www.surveymonkey.com/home/` (visited on 11/04/2016).

[11]   E. Heinrich and Y. Wang. "Online marking of essay-type assignments". In: *In*. bibtex: heinrich_online_2003. 2003, p. 768772. URL: `http://www-ist.massey.ac.nz/MarkTool/Publications/EdMedia2003Onscreen.pdf`.

[12]   R. Higgins, P. Hartley, and A. Skelton. "The Conscientious Consumer: Reconsidering the role of assessment feedback in student learning". In: *Studies in Higher Education* 27.1 (Feb. 2002). bibtex: higgins_conscientious_2002 bibtex[langid=english], pp. 53–64. ISSN: 0307-5079, 1470-174X. DOI: `10.1080/03075070120099368`. URL: `http://www.tandfonline.com/doi/abs/10.1080/03075070120099368` (visited on 11/23/2016).

[13]   M. Joy, N. Griffiths, and R. Boyatt. "The Boss Online Submission and Assessment System". In: *J. Educ. Resour. Comput.* 5.3 (Sept. 2005). bibtex: joy_boss_2005. ISSN: 1531-4278. DOI: `10.1145/1163405.1163407`. URL: `http://doi.acm.org/10.1145/1163405.1163407` (visited on 11/23/2016).

[14]   M. Joy and M. Luck. "Effective electronic marking for on-line assessment". In: *ACM SIGCSE Bulletin* 30.3 (Sept. 1998). bibtex: joy_effective_1998 bibtex[langid=english], pp. 134–138. ISSN: 00978418. DOI: `10.1145/290320.283096`. URL: `http://portal.acm.org/citation.cfm?doid=290320.283096` (visited on 11/15/2016).

[15]   A Langan and C Wheater. *Some Insights into Peer Assessment \textbackslashtextbackslash LTiA Issue 4 \textbackslashtextbackslash CeLT \textbackslashtextbackslash MMU*. bibtex: langan_insights_????? URL: `http://www.celt.mmu.ac.uk/ltia/issue4/langanwheater.shtml` (visited on 11/21/2016).

[16]   *MoSCoW Prioritisation*. Nov. 2015. URL: `https://www.agilebusiness.org/content/moscow-prioritisation` (visited on 04/02/2017).

[17] P. Orsmond, S. Merry, and K. Reiling. "The Use of Student Derived Marking Criteria in Peer and Self-assessment". In: *Assessment & Evaluation in Higher Education* 25.1 (Mar. 2000). bibtex: orsmond_use_2000, pp. 23–38. ISSN: 0260-2938. DOI: 10.1080/02602930050025006. URL: http://dx.doi.org/10.1080/02602930050025006 (visited on 11/16/2016).

[18] J. Plekhanova. "Evaluating web development frameworks: Django, Ruby on Rails and CakePHP". In: *Institute for Business and Information Technology* (2009). bibtex: plekhanova_evaluating_2009. URL: https://ibit.temple.edu/wp-content/uploads/2011/03/IBITWebframeworks.pdf (visited on 11/23/2016).

[19] "System and method for restricting user access rights on the internet based on rating information stored in a relational database". Pat. bibtex: baker_system_1997 bibtex[holder=Baker, Brenda Sue and Grosse, Eric]. Oct. 1997. URL: http://www.google.com/patents/US5678041 (visited on 11/22/2016).

[20] J. Tang and C. Harrison. "Investigating university tutor perceptions of assessment feedback: three types of tutor beliefs". In: *Assessment & Evaluation in Higher Education* 36.5 (Aug. 2011). bibtex: tang_investigating_2011, pp. 583–604. ISSN: 0260-2938. DOI: 10.1080/02602931003632340. URL: http://dx.doi.org/10.1080/02602931003632340 (visited on 11/23/2016).

[21] Z. Tufekci. "Can You See Me Now? Audience and Disclosure Regulation in Online Social Network Sites". In: *Bulletin of Science, Technology & Society* 28.1 (Feb. 2008). bibtex: tufekci_can_2008 bibtex[langid=english], pp. 20–36. ISSN: 0270-4676, 1552-4183. DOI: 10.1177/0270467607311484. URL: http://bst.sagepub.com/content/28/1/20 (visited on 11/24/2016).

[22] *Turnitin - Home.* bibtex: _turnitin_???? URL: http://turnitin.com/en_us/home (visited on 11/21/2016).

[23] E. Waclawski. "How I Use It: Survey Monkey". In: *Occupational Medicine* 62.6 (Sept. 2012). bibtex: waclawski_how_2012 bibtex[langid=english], pp. 477–477. ISSN: 0962-7480, 1471-8405.

DOI: 10.1093/occmed/kqs075. URL: http://occmed.oxfordjournals.org/content/62/6/477 (visited on 11/22/2016).