School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

WebCMS3 Maintenance

by

Sunny Tang

Thesis submitted as a requirement for the degree of

Bachelor of Engineering in Software Engineering

Submitted: 28/04/2024

Supervisor: Prof. John Sheppard Student ID: z5255035

Abstract

WebCMS3 plays an integral role for the large majority of CSE courses at UNSW, simplifying the teaching experience for staff and enhancing the learning experience for students. In this thesis, we aim to improve the current system to address existing issues and further better the experience for its users. By examining other related systems, we will add or modify existing functionality within the quiz and forum modules, and investigate software bugs.

Abbreviations

CSE Computer Science and Engineering

LMS Learning Management System

UNSW University of New South Wales

UI User Interface

UX User Experience

KISS Keep It Simple, Stupid

SiMS Keep It Simple, Stupid

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Establishing a robust, intuitive and efficient website application is essential in streamlining the delivery of services to people. In the context of an educational institution, Learning Management Systems serve to act as a content delivery platform for students. They often also perform other duties, such as keeping records of assessment marks or student enrollments, distributing learning material, and providing a place for staff and students to communicate.

WebCMS3 is a Learning Management System currently in use by the University of New South Wales School of Computer Science and Engineering. It offers a classic appearance and the standard functionality of a LMS, incorporating quizzes, forums, formal documentation and learning material all in one central location. However, it is not without its faults.

In this thesis, we aim to address present student frustration and improve the user experience by implementing new functionality and addressing existing problematic issues. In particular, we wish to improve the quiz module by adding more functionality, investigate some of the software bugs across the site, and also improve the overall user interface and user experience with the forums.

In order to gain a better understanding of the structure of WebCMS3 and make improvements to it, we must first examine similarly implemented Learning Management Systems. With reference to this research, we will outline the proposed changes, then review the implementation

and the end product, particularly the design decisions made and changes to the system architecture. Lastly, we conclude with a summary and examine the possibility of future work to be continued on.

Chapter 2

Background

2.1 General Background

Learning Management Systems were originally conceived to enable autonomous student assessment and facilitate learning in absence of a teacher. These notions were brought to life early in the 20th century but were continually iterated and improved upon. MIT's Project Athena [Fre], which began in the May of 1983, marked the beginning of the need for a systematic method of managing learning material within education. By its completion in 1991, it had successfully realised the first distributed computing environment implemented at an educational institution, allowing staff and students to access their files, submit assignments and the like from anywhere on the campus. A decade later and Australian computer scientist Martin Dougiamas would release his open-source online LMS, Moodle, which eventually allowed users to create a personalised learning experience [Pra20], [QGa21].

With the rapid expansion of the internet and the increasing global dependency upon it, various systems have since been introduced to the world but have gone underutilised. Learning Management Systems acted more as a secondary or tertiary method of teaching, supplementing in-person classes. However, the COVID-19 pandemic in 2020 which eventually resulted in a global lockdown and prevented in-person learning acted as an ultimatum - education needed to transition online. Institutions came to realise the importance of incorporating LMSs into their infrastructure

to streamline the process of content delivery and management [QE21]. This drastic change led to a rapid evolution for LMSs, improving the overall quality of their systems.

In the current day, they play an integral role in the field of education. LMSs are used as a primary platform to enhance in-person learning, providing a central location for staff-student discussion and learning material to be efficiently distributed. Having a system that manages all these functionalities seamlessly is essential in ensuring a high quality education and a pleasant experience for teachers and students alike. Most of the modern day systems have become standardised, offering a similar set of functionalities in terms of quizzing and forums.

The third iteration of CSE's web-based course management system, WebCMS3, is the culmination of the work of many past staff and students. Built to be the successor to WebCMS2, it currently functions as the backbone for many CSE courses, integrating with other key systems such as *give*, CSE's own means of tracking assignment submission and marks and the Student Management System (SMS).

Originally introduced sometime around 2002, the original system provided various tools for lecturers and students to use [She02]. As a staff member, course web pages could be customised through an interface, a notice board gave the ability to make announcements, a calendar existed to schedule events, and there was the capability to manage course content, along with students and groups. Students, on the other hand, were given a place to access records of their assessment marks and course statistics, talk to peers and manage their own groups. A standard course page is shown in Figure 2.1.

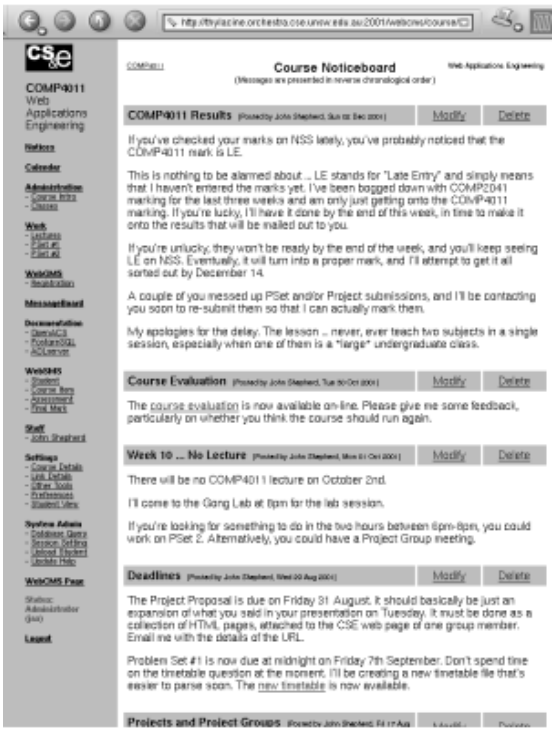

Figure 2.1: Typical Course Homepage

2.2 Related Systems

A number of other existing systems have been considered in order to improve our understanding of the state of the art.

Firstly, we will examine the quiz regrading functionality that each system implements. Currently, staff have raised issues regarding the lack of functionality when regrading quizzes in WebCMS3.

Next, we will examine the forum modules of each system. WebCMS3s existing forum module has been the cause of concern and frustration from both staff and student alike, who have voiced their discontentment with both the lack of functionality and usability.

These following systems were chosen due to their widespread usage within educational institutions and overall positive user experience. While all of the systems implement similar regrading functionality and a forum module at a primitive level, the semantics of their implementations

and the features they offer vary from system to system.

2.2.1 Quiz Regrading

The quiz regrading functionality offers the ability to amend an incorrectly selected answer in a quiz and modify the students marks where necessary. Generally, this functionality operates as follows:

1. The offending question is to be changed
2. The regrading is run either for the question itself, or the whole quiz

Moodle

Moodle is an open-source LMS with just over four hundred million registered users, and the current LMS of choice for the majority of UNSW.

It implements an very extensive quiz module, due to its intention to be sufficient for holding a variety of assessment types. There are numerous quiz creation options, as seen in Figure 2.13, enabling a quiz to be completely tailored to any specific situation.

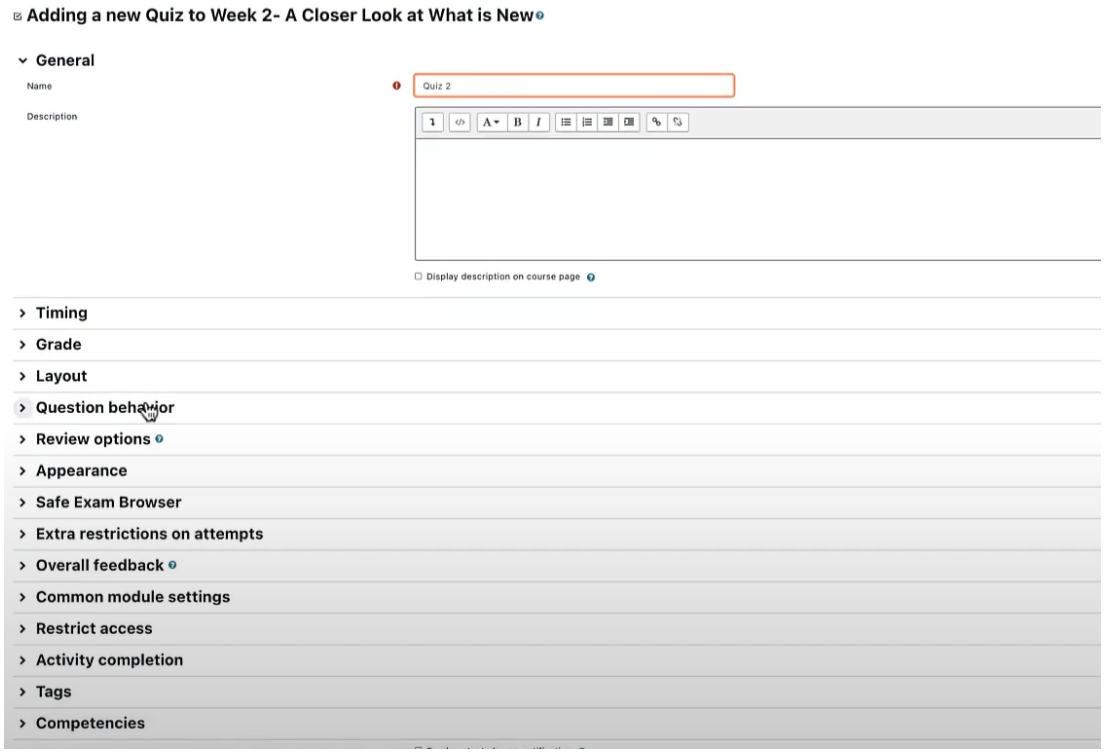

Figure 2.2: Quiz Creation Page

Another example of its vast quizzing capability can be seen in Figure 2.3, where fifteen different question types are able to chosen as a template. Questions themselves can be created as separate objects and added to question banks, enabling them to exist independently of a quiz. This creates flexibility for the user by allowing questions to be re-used in any future quizzes. Users are also able to view the history of a question, showing the edits and changes of previous versions [Moo24c].

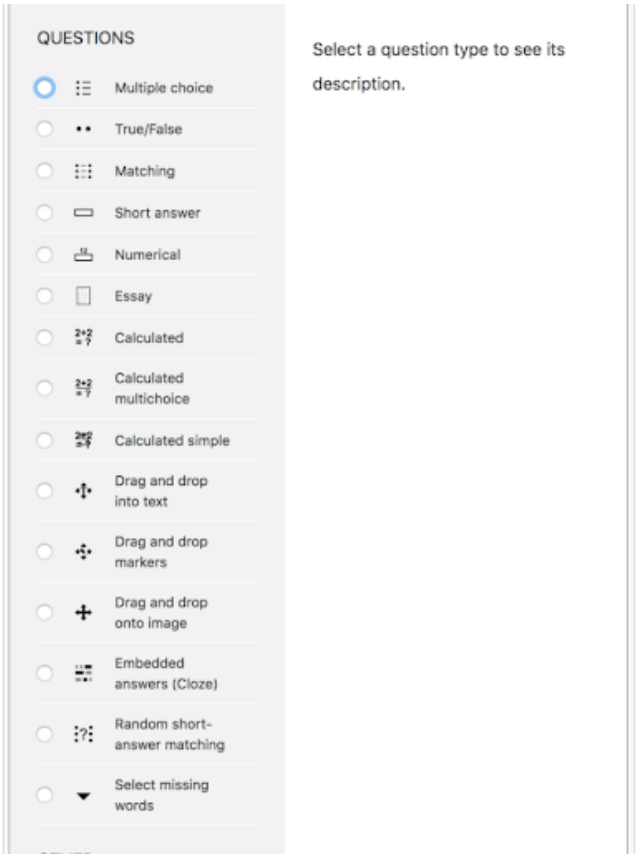

Figure 2.3: Different Types of Question Presets

The regrading ability implemented by Moodle functions by recalculating a mark according to the latest version of the questions in a quiz. This can be changed so that a specific version of a question is used instead. A dry-run option also provides the user with a preview of the updated marks, enabling a user to spot any areas of concern.

Whilst the abundance of customisation options for the quiz may seem like a undoubtedly useful feature with only positive implications, new users may often feel overwhelmed by the sheer number of settings. As a result, they may need to trawl through pages of documentation, a hard task for those who are less technologically adept. This is only hindered by Moodle's confusing documentation [Moo24b], which often seems to contradict itself and/or the actual system itself. As a result, users may spend more time on understanding how to use the system instead of producing learning materials.

Another shortcoming of the regrading feature is that despite the platforms overwhelming number of general customisation options, it lacks options when it comes to the adjustment of marks. In certain cases, we may wish to invalidate the question completely, give everyone full marks for the question, grant marks for only the updated correct answer or grant marks for both the original correct and new correct answer. Moodle does not support this capability, reducing the overall flexibility of the regrading functionality. Moodle also does not allow for a question to be regraded if a question has the number of answers modified. A question which originally contained five answers and is modified to only contain four poses an issue and prevents the regrading of the specific question.

Canvas

Canvas takes a different approach in designing their system, choosing to adopt the KISS principle. It currently offers two main methods of constructing a quiz, either using their old 'Classic Quizzes' engine or 'New Quizzes'. 'Classic Quizzes' can be migrated to 'New Quizzes', but 'New Quizzes' are not backwards compatible. We will be primarily focusing on the 'New Quizzes' engine as for almost all use cases, 'New Quizzes' is superior to its predecessor. Figure 2.4 shows the entry point for creating a quiz. On this page quizzes can be assigned points, categories, date and a submission type, which include text entry or media upload/recordings.

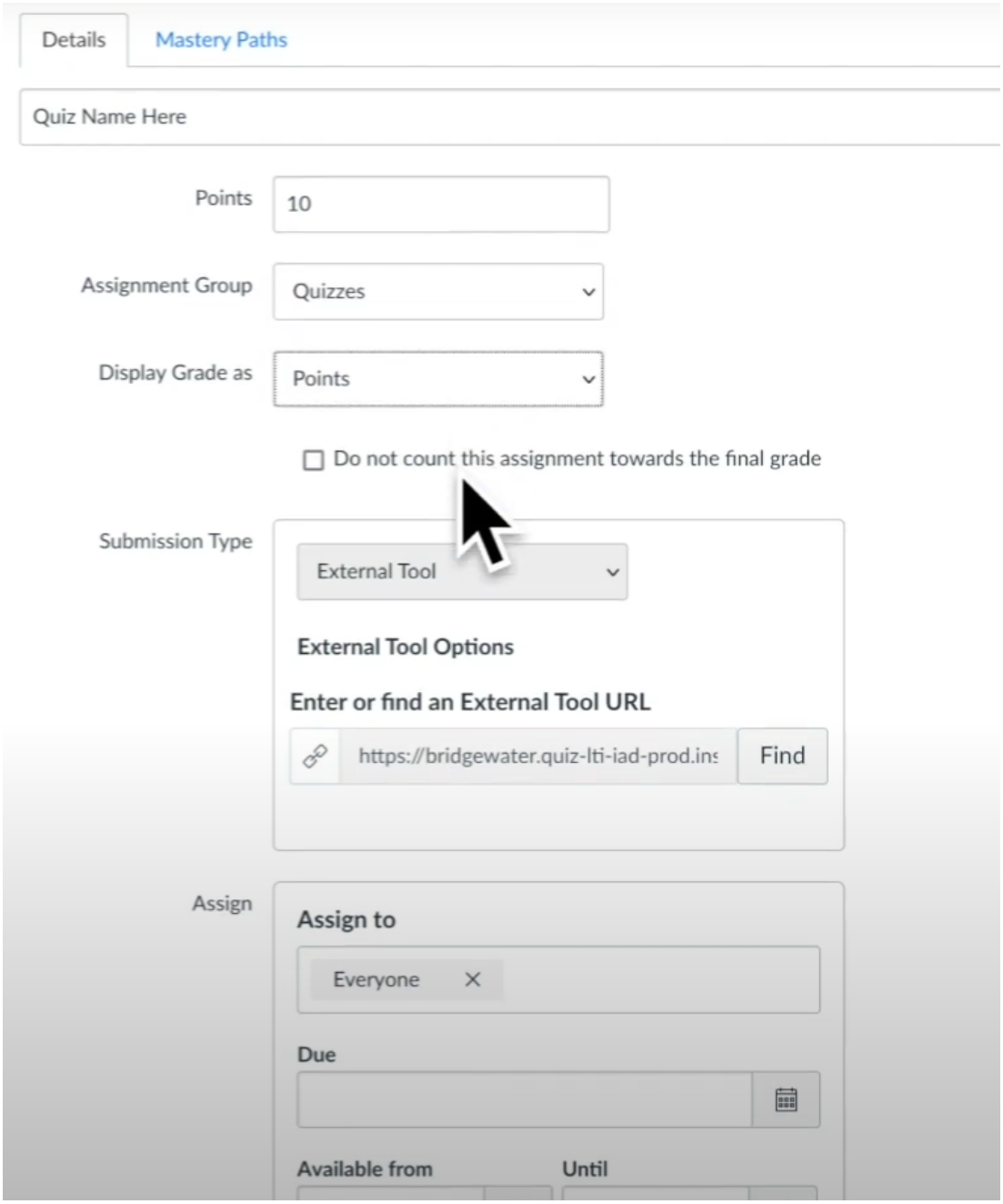

Figure 2.4: Enter Caption

With twelve question types available, which can be seen in Figure 2.5, Canvas also offers the ability to create questions independent from Quizzes by adding them to a question bank.

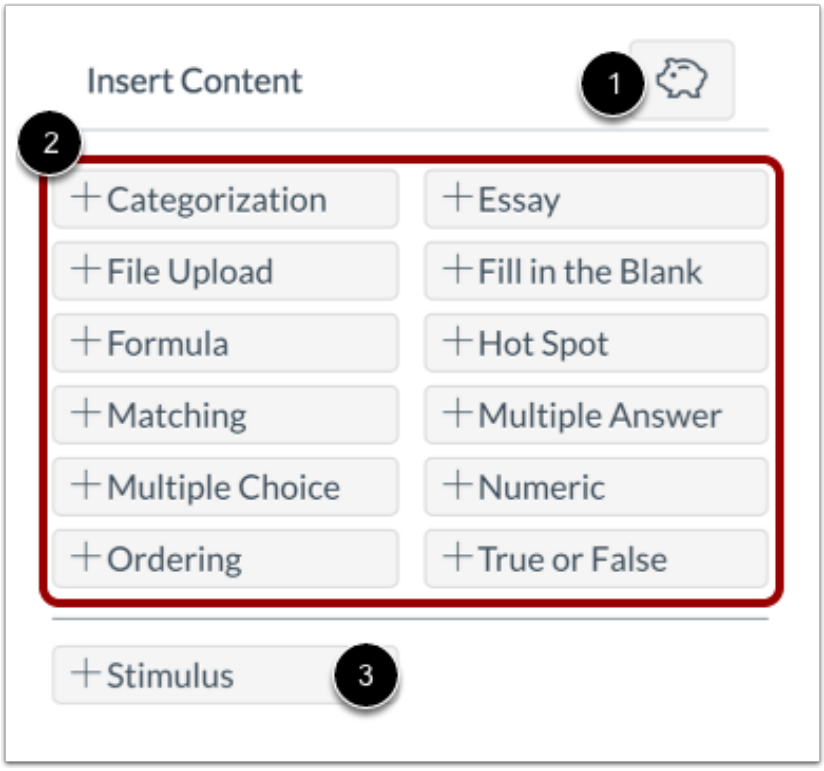

Figure 2.5: Question Types of Canvas

When regrading, Canvas regrades each question upon change, rather than the whole quiz itself [Can24b]. After an answer for a question has been changed, it presents the user with four options, shown in Figure 2.6. These four options provide some much needed flexibility with how marks are adjusted. Furthermore, Canvas has a substantial amount of documentation on regrading in addition to its intuitive interface, making it simple and easy to use for new users.

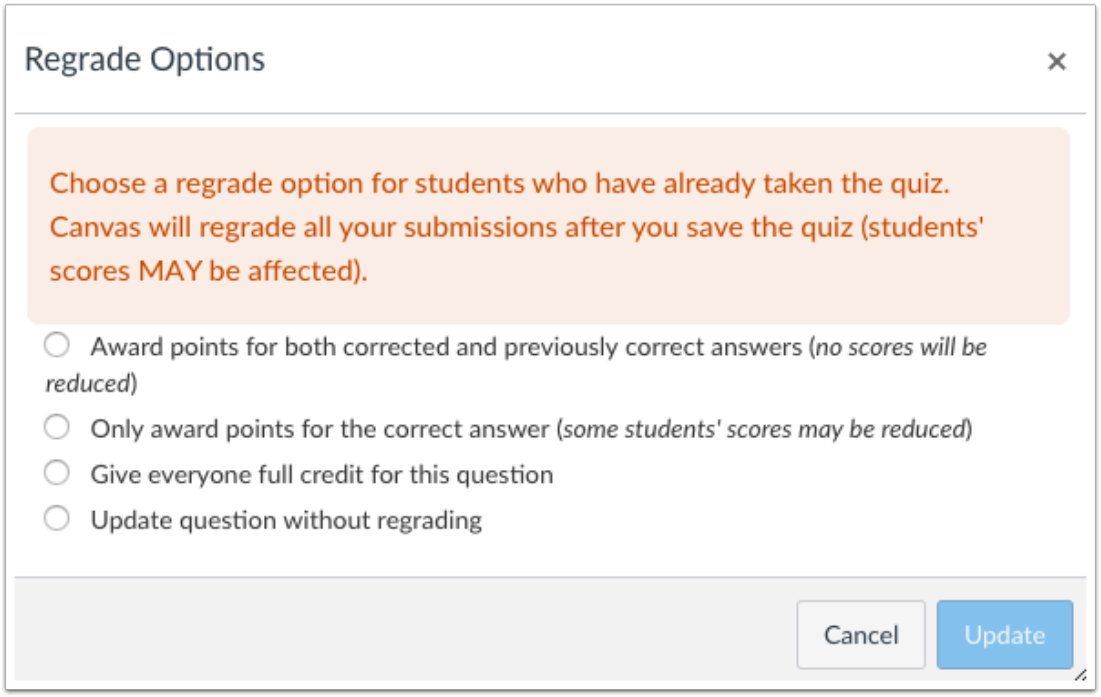

Figure 2.6: Regrading Options for Canvas

Whilst Canvas outclasses Moodle when it comes to overall flexibility and usability, it is still lacking on some areas. It fails to provide the dry run option that Moodle implements, and quiz regrading does not apply for questions drawn from a question bank. Much like Moodle, altering the number of answers in a question also prevents the option to regrade. Furthermore, a user cannot choose the ”Award points for both...answers” option when regrading if the question is one where multiple answers are enabled.

D2L Brightspace

Brightspace LMS offers similar regrading functionality to Canvas, but on a more limited level. The regrading process is also similar to Canvas, where regrading is run for an individual question rather than the whole quiz [Col]. Brightspace provides the user with two options, as can be seen in Figure 2.7.

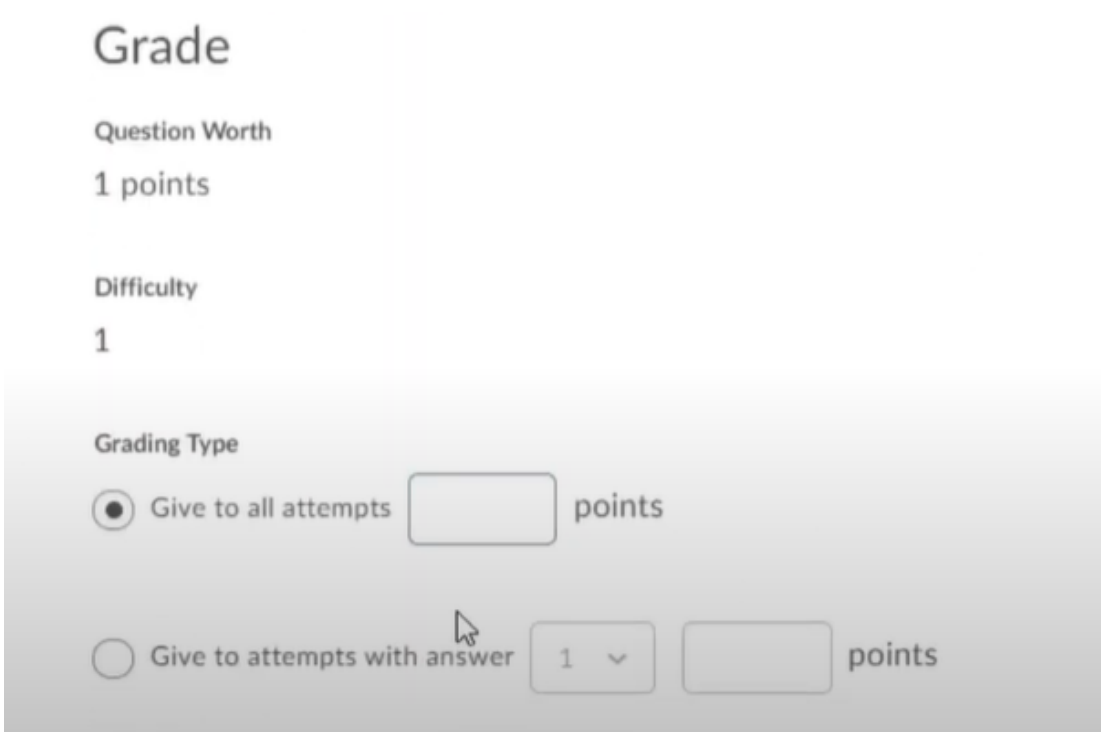

Figure 2.7: Regrading Options for Brightspace

Whilst these options do provide some degree of flexibility for a user, it is incomparable to what Canvas offers. In particular, the use cases that the second option caters for is only small subset of the use cases that Canvas covers. In addition, Brightspace also lacks the ability to add multiple correct answers during regrading, much like its competitors.

WebCMS3

WebCMS3 contains very limited regrading functionality, as can be seen in 2.8. Whilst the process is intuitive, this is more so due to the lack of options rather than the design, although one could argue that its intended design was to be simple. To regrade, a user simply changes the correct answer for a question and saves the quiz.

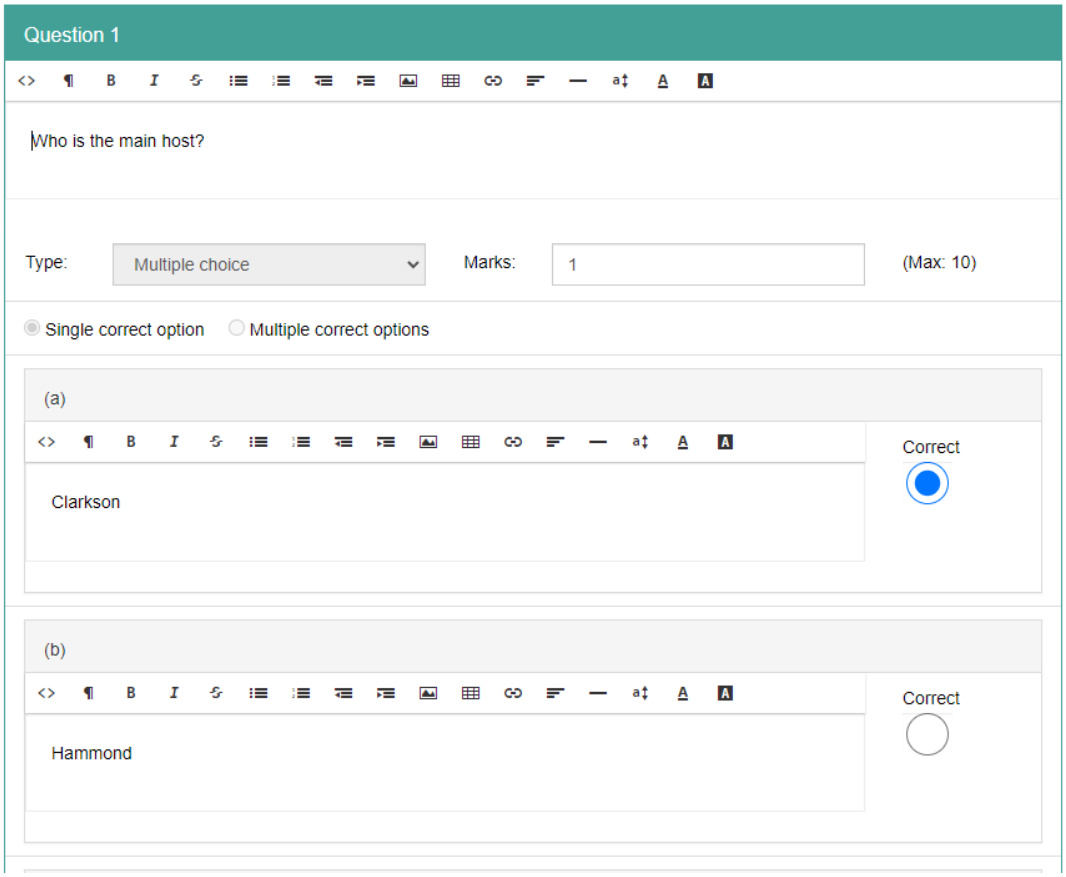

Figure 2.8: Regrading Screen for WebCMS3

WebCMS3 clearly lacks many features in comparison to the other LMSs that we have just examined. It lacks the dry run feature that Moodle offers and any sort of mark adjustment options like the ones provided by Canvas or Brightspace.

Another issue, albeit unrelated to quiz regrading, is the unintended ability of students to access hidden quizzes. While a hidden quiz does not appear on the 'Home' or 'Activities' pages, students are still able to access the quiz given a direct URL link. Clearly, this is not intentional.

2.2.2 Forums

The forum modules offer the ability to manage asynchronous communication between staff and students. Generally, they implement at minimum the ability to:

1. Make a categorised post
2. Comment on others posts

Moodle

As can be seen in Figure 2.9, the Moodle forum overview contains the groups under which individual posts are categorised.

General forums

Forum	Description	Discussions	Unread posts	Track	Subscribed	Email digest type	RSS
Announcements	General news and course announcements	17	0	Yes	Yes	Default (No digest)	

Learning forums

Section	Forum	Description	Discussions	Unread posts	Track	Subscribed	Email digest type	RSS
General Course Resources and Feedback Hub	Course Admin Q&A	This forum is for questions related to course administration, for example about accessing lectures/tutorials or using Numbas, etc.	51	0	Yes	No	Default (No digest)	
	Course Content Q&A	This forum is for questions related to the course content, for example clarifying content from lectures or discussing problems from the tutorials or Numbas lessons. For questions specifically about assessment content, use the Assessment Forum in the Assessment Hub.	178	0	Yes	No	Default (No digest)	
Assessments Hub	Assessments Forum	Post any questions relating to course assessments here.	117	0	Yes	No	Default (No digest)	

Figure 2.9: Moodle Forum Overview

Information regarding each topic is displayed, along with the option to receive notifications about new posts within that topic and the ability to track read posts. The user is given the ability to create different forum types, or categories, seen in Figure 2.10.

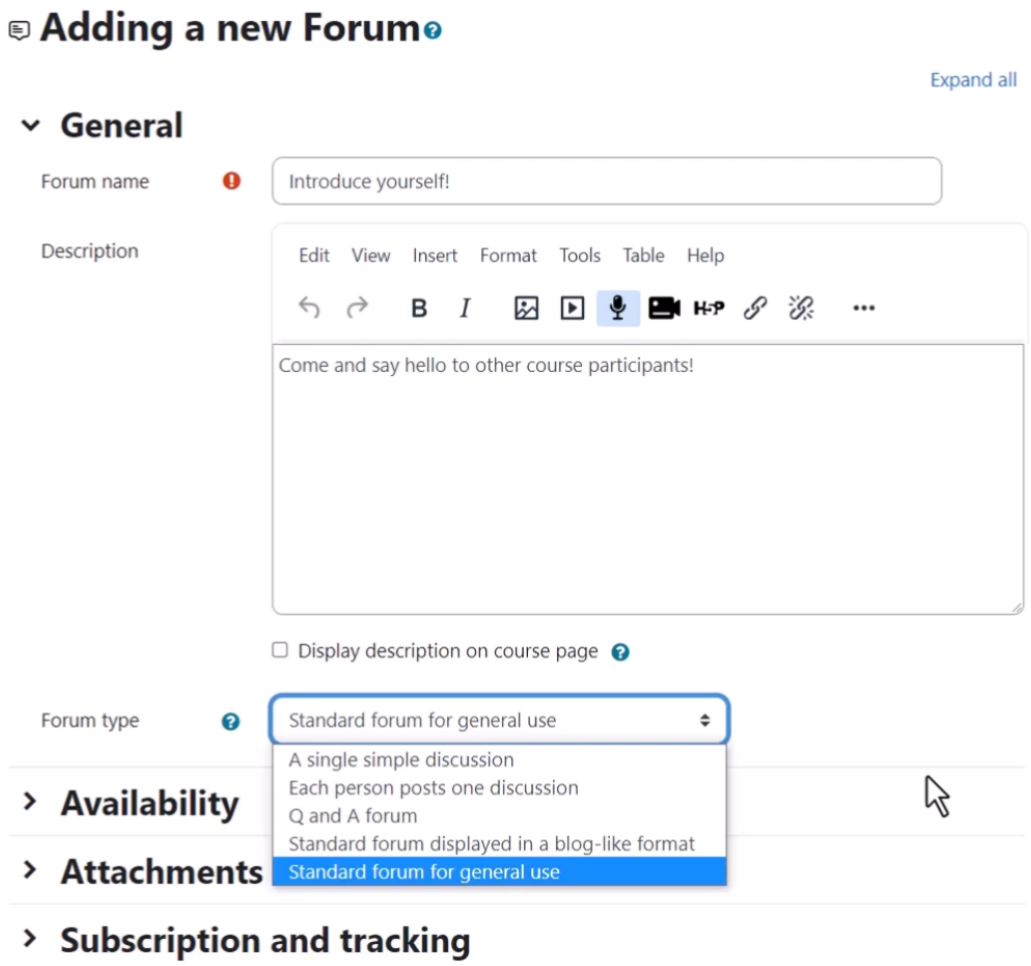

Figure 2.10: Creating a Category in Moodle Forums

Basic formatting options and media input support are provided by the editor, and much like the quiz module, a wide range of customisation options enable the user to tailor a forum to their needs. Most notably, Moodle settings aim to provide support for an assessable forum [Moo24a], where posts and replies can be marked with direct integration with the gradebook.

Navigating into the forum categories takes us to new page shown in Figure 2.11, where posts are by default, displayed in chronological order from most recent to oldest by default. Other sorting options include the ability sort posts alphabetically, reverse chronological order, the most recent reply, or the number of replies.

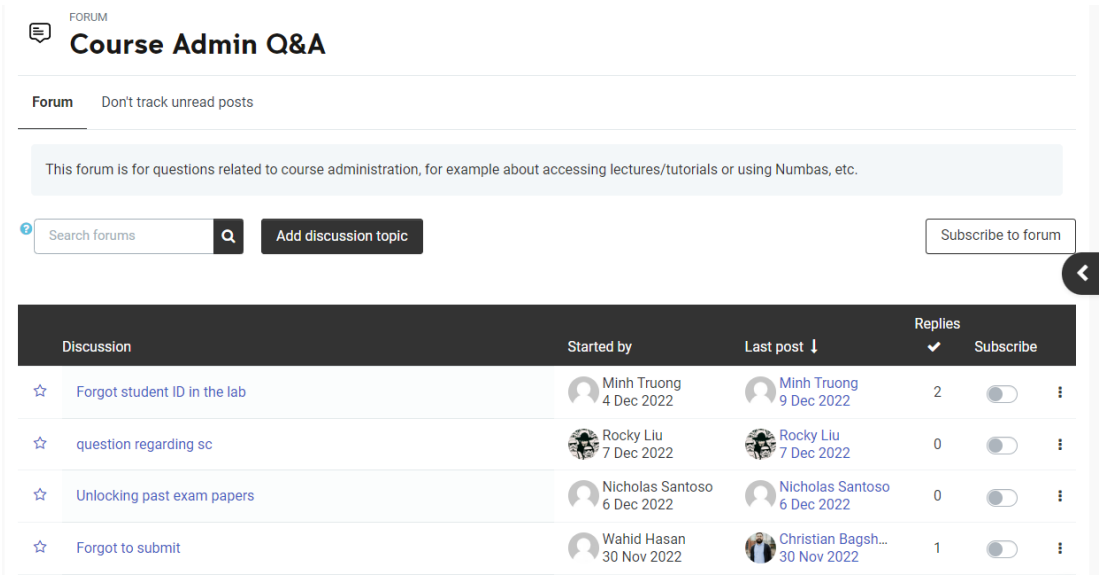

Figure 2.11: List of Posts

Above the main forum body exists a search functionality that allows the user to look for related posts or replies. As can be seen in Figure 2.12, the query string is highlighted in a separate colour to increase usability and accessibility.

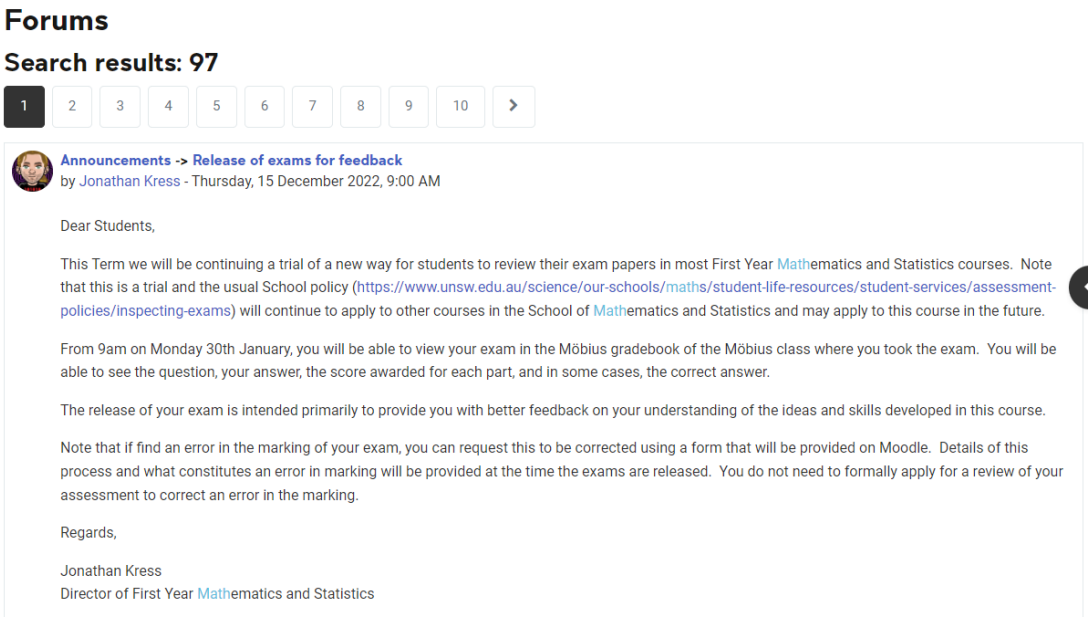

Figure 2.12: Search Results

Adjacent to the search bar lies the ability to create a post. Clicking on this button expands the page downwards, leaving us with what can be seen in Figure 2.13. This container also implements some basic formatting functionality and supports image, video and audio upload.

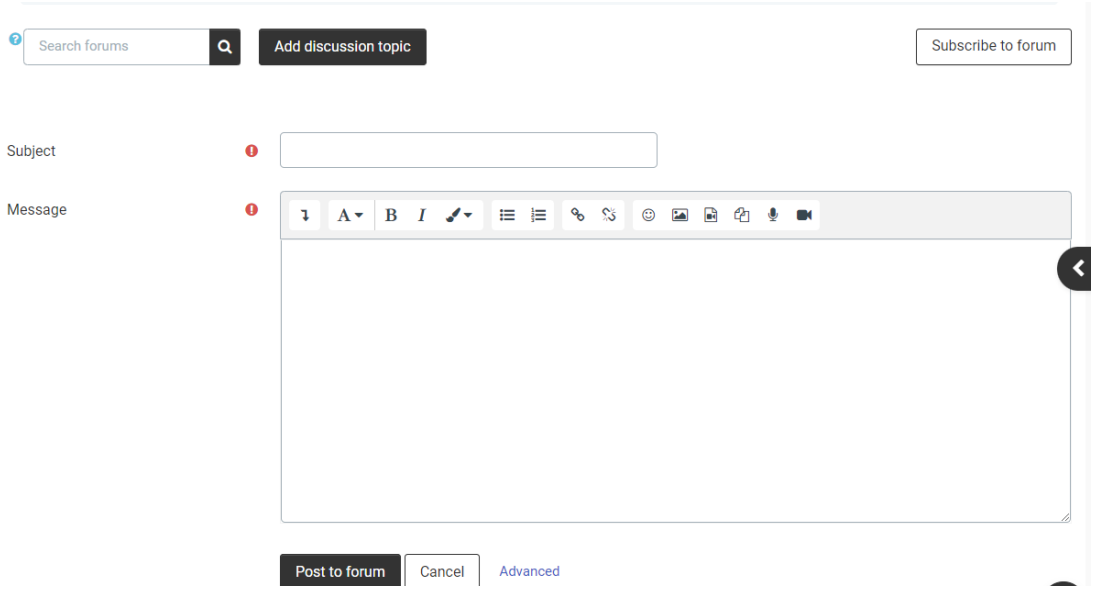

Figure 2.13: Create Post Page

An individual forum post exists on a different page from the forum overview and is shown in Figure 2.14.

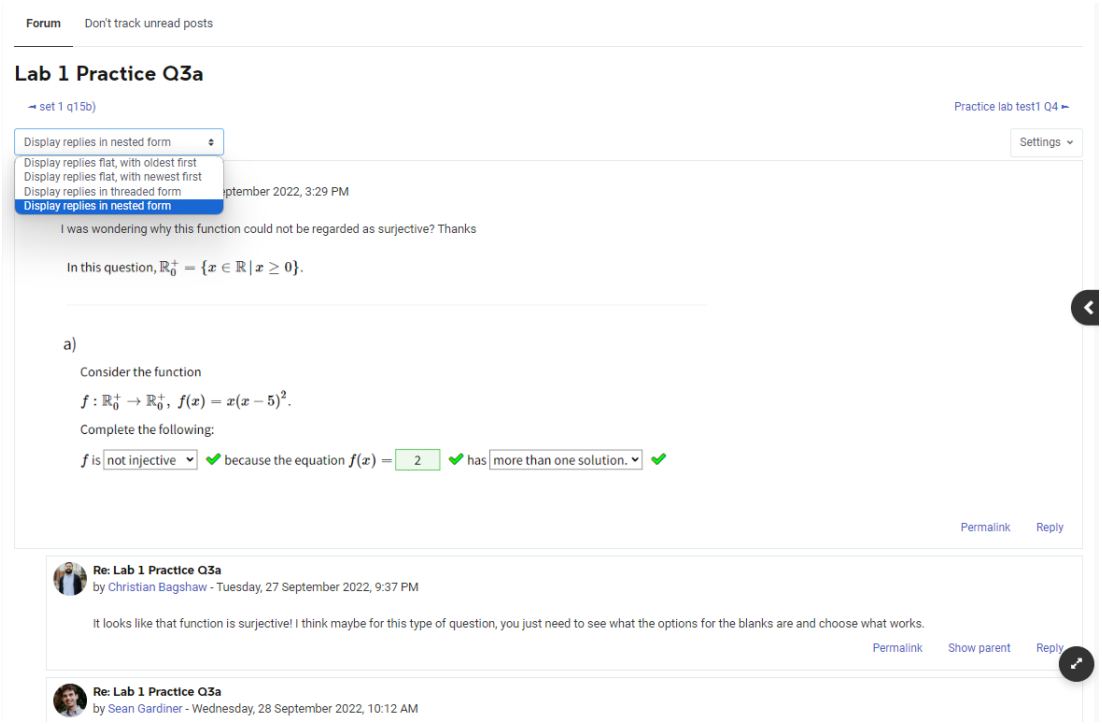

Figure 2.14: Standard Forum Post

The main body of the post occupies a portion of the screen, with replies being displayed in a nested format by default. The user can adjust the way replies are displayed, allowing for the nesting to be removed, or for the replies to be collapsed. By default, notifications are configured such that the original poster receives notifications via email whenever another user comments on their post.

Overall, the forums suffice. While they do not offer any noteworthy features, they are intuitive to navigate, offer almost enough tools to cover the majority of use cases and do not appear to have any underlying issues. However, Moodle forums lack the ability to make private posts, a special category of posts which are only visible to staff, or the upvoting/liking functionality offered by other LMSs.

Canvas

Canvas adopts a slightly different organisation hierarchy in comparison to Moodle, as can be seen from 2.15. Forum posts, or 'discussions' as Canvas calls them, are not categorised on the main forum page [Can24a].

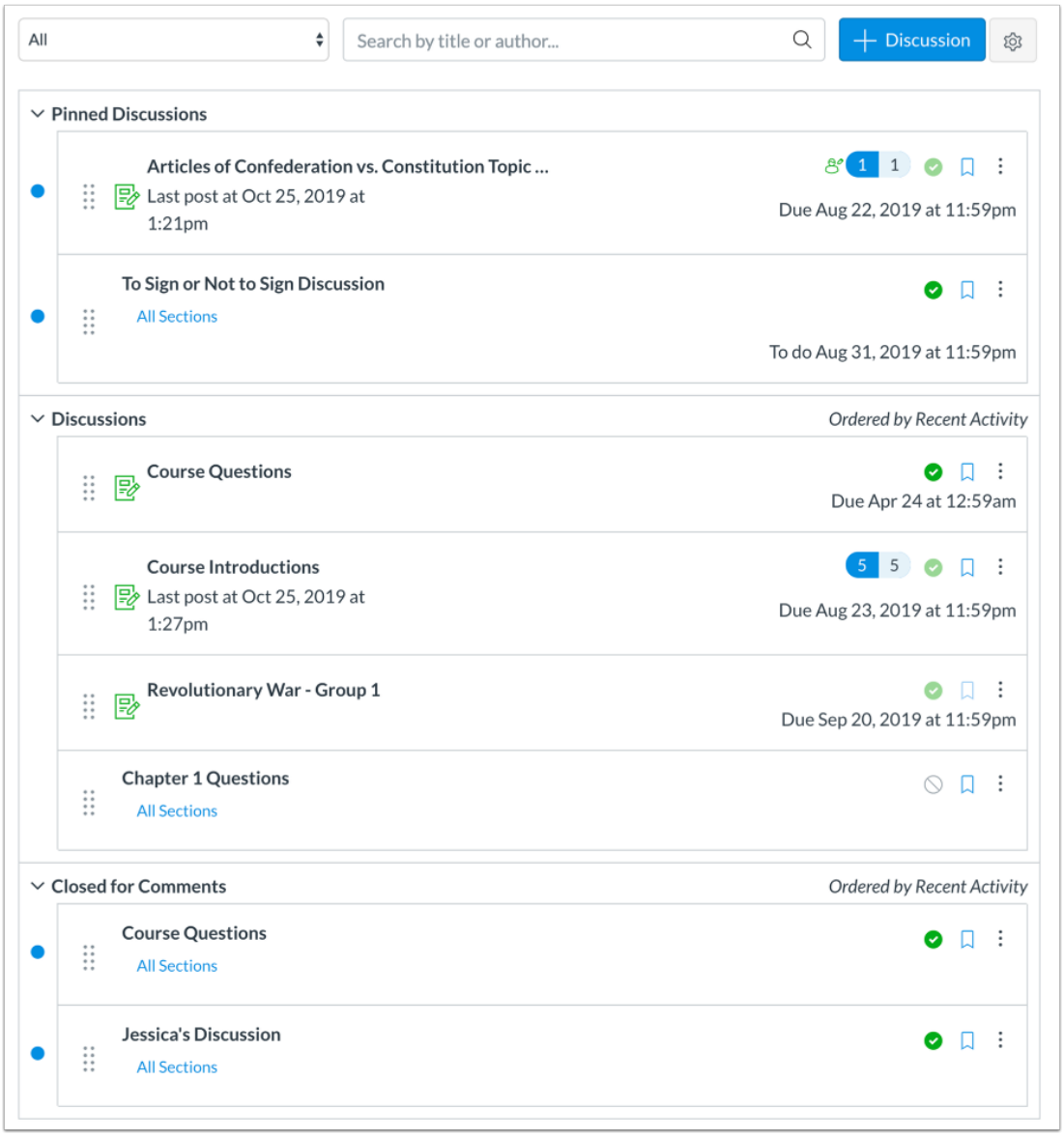

Figure 2.15: An Overview of the Canvas Discussions page

Whilst basic functionality is provided to highlight pinned posts, the main categorisation of

posts is done on the 'Modules' page, shown in 2.16.

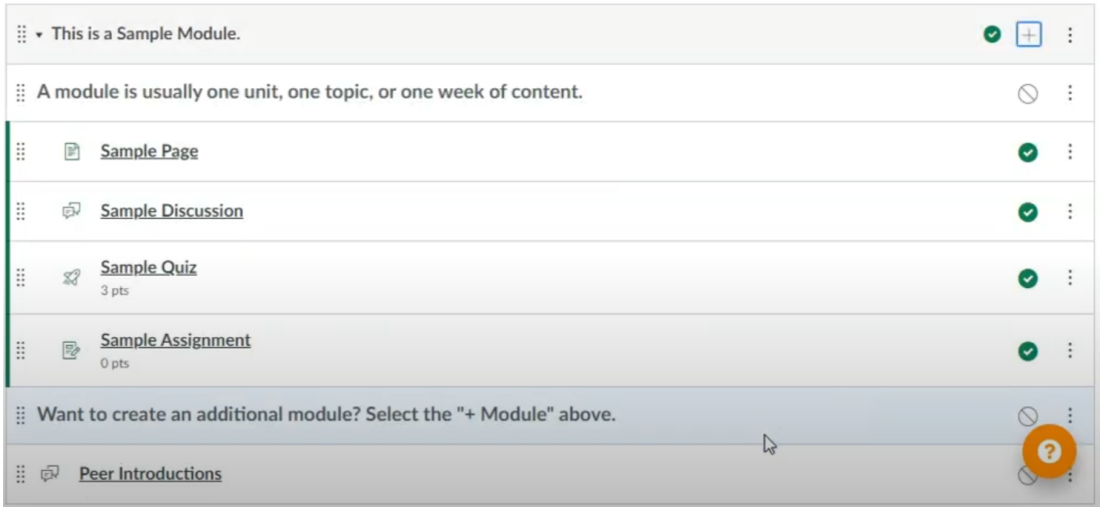

Figure 2.16: An Overview of the Canvas Modules page

Standard formatting options and media input types are supported when creating a post, as well as a small range of settings.

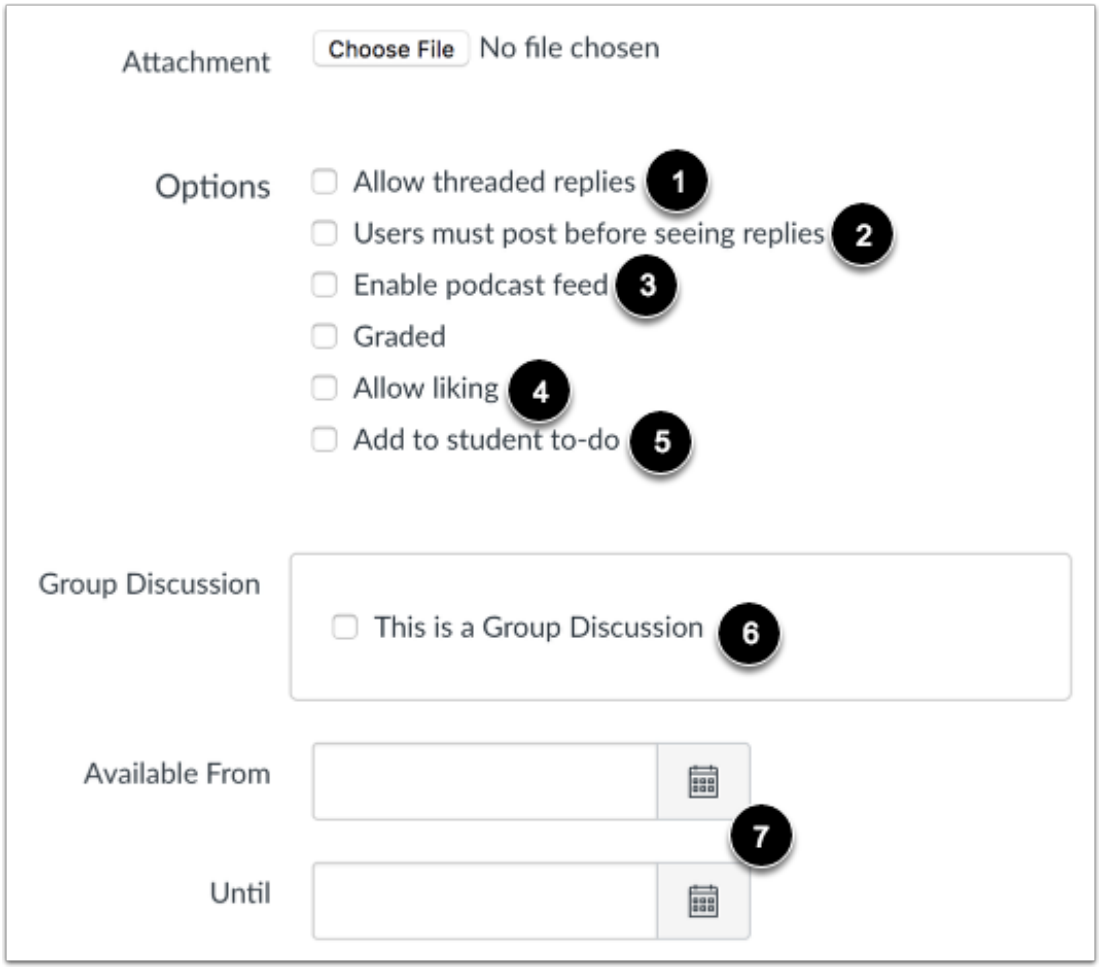

Figure 2.17: Settings when creating a Canvas Discussion

These settings, much like Moodle, provide the functionality for assessable forum posts. A standout feature is the 'Group Discussion' component, which enables a forum post to be limited to specific members of the course.

The content of a Canvas discussion post can be seen below in Figure 2.18, existing on a separate page to the post overview.

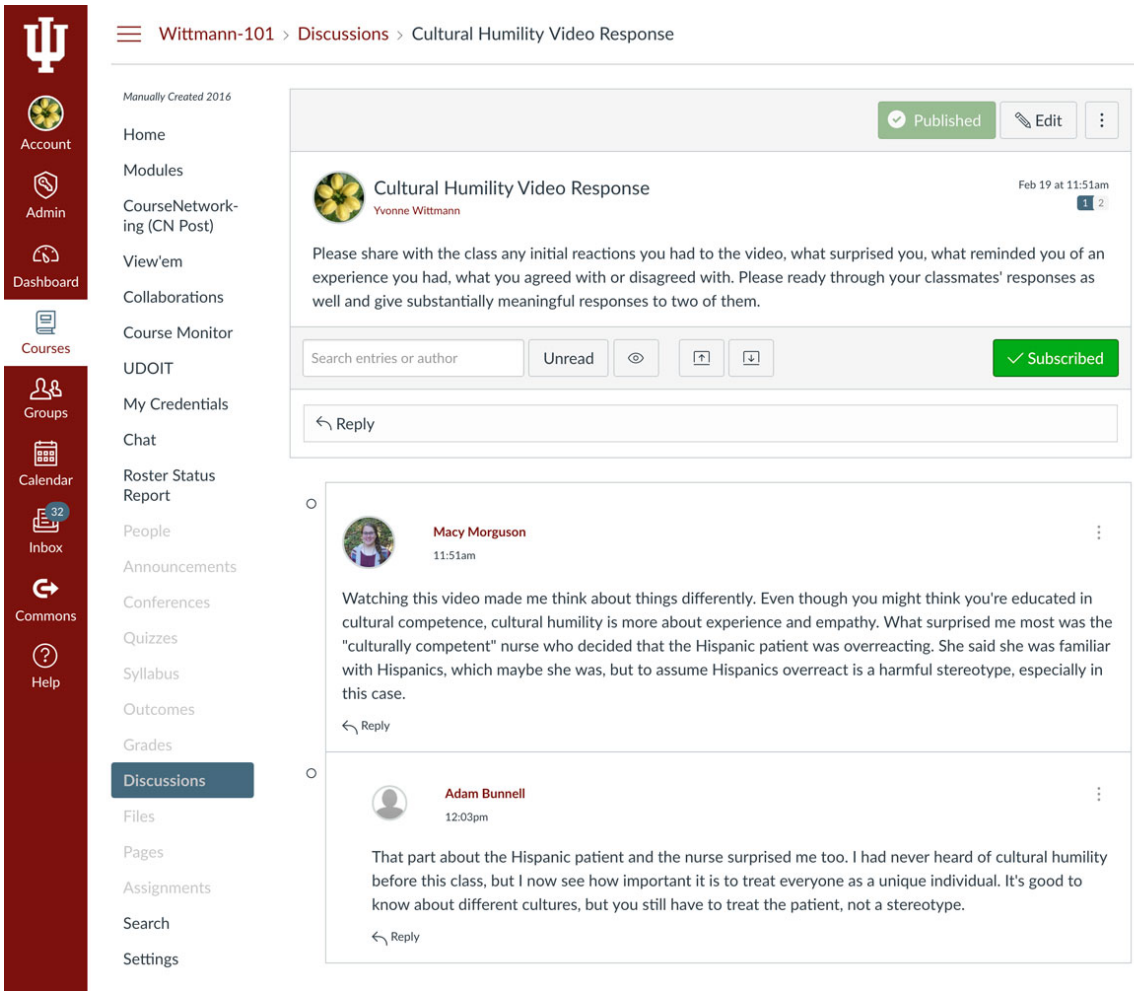

Figure 2.18: A Standard Canvas Discussion

Here, students are able to search and sort replies, and opt in and out of subscribing to posts. The post content takes up the majority of the screen, with replies expanding down the page. The buttons and reply text fields are placed in obvious locations and easily visible, promoting high usability.

While Canvas appears to provide adequate functionality for a forum, the overall structure of the it caters more towards being an assessment platform. This is evidenced by discussions being integrated into the 'Modules' page and the deliberate choice for them to be organised by learning topics rather than abstract categories.

A more relevant limitation of Canvas is the narrow searching functionality. Unlike Moodle,

the searching functionality only applies to a specific post and not the whole forum board. As a consequence, a user may need to go through the time consuming process of searching all the existing posts to get the relevant search results. The inefficiency of the searching functionality is only further propagated by the lack of emphasis on the search terms. Unlike Moodle, Canvas makes no effort to highlight the search query string in the results, making it far more difficult for users to identify the areas of interest. Furthermore, the returned results are by default collapsed, potentially hiding the relevant result.

Ed Discussion

Ed Discussion is not a LMS but rather an application designed with the sole purpose of being used as a forum. The dashboard for Ed is shown below in Figure 2.19, and contains a list of courses a user is currently enrolled in, as well as archived courses from previous periods.

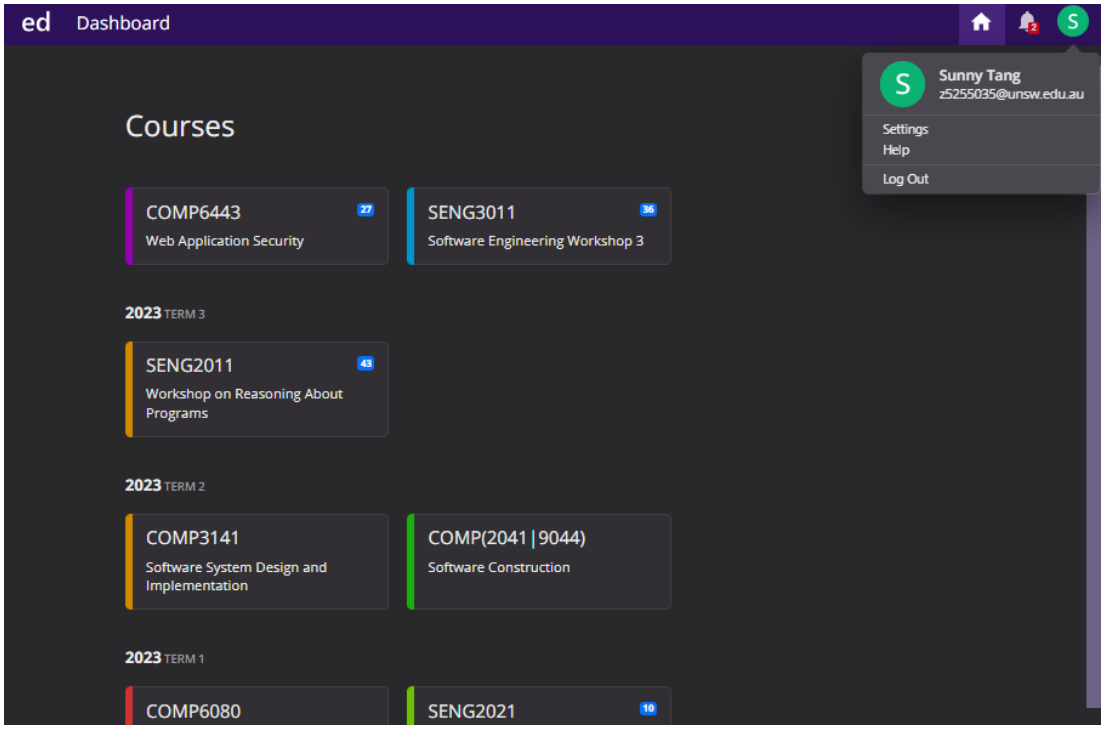

Figure 2.19: Ed Discussion Home Page

In the settings page, an option to customise the appearance for post previews is provided.

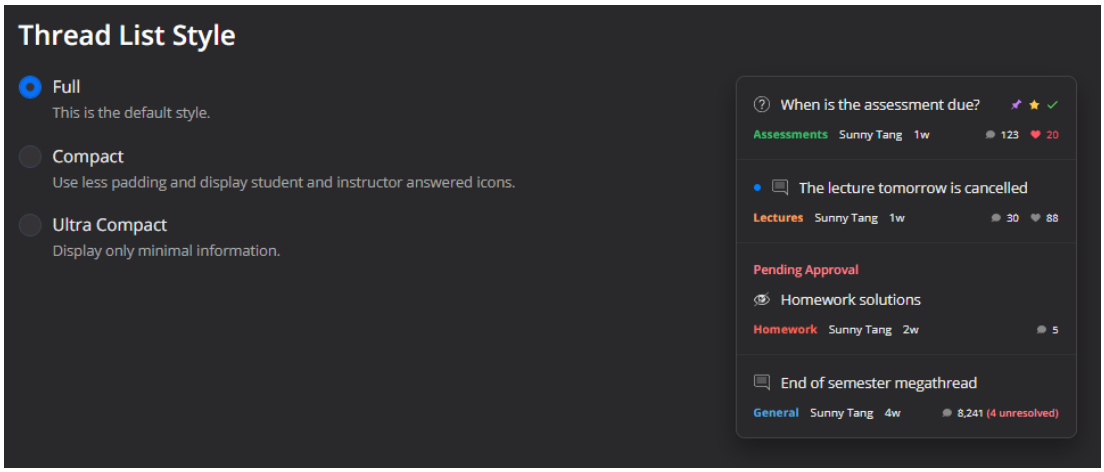

Figure 2.20: Ed Discussion Settings

Users are also able to view a collection of all of their posts and basic statistics, as seen in Figure2.21, and the posts can be filtered according to courses.

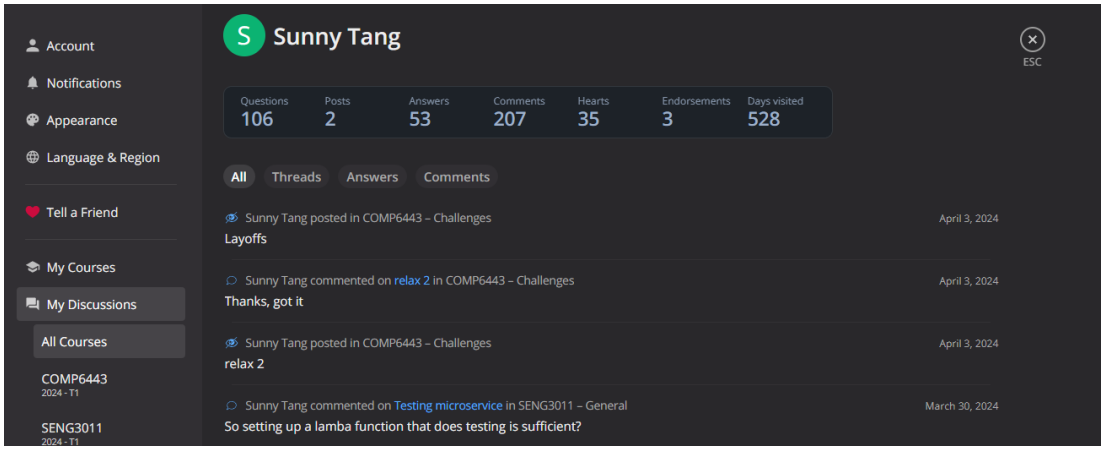

Figure 2.21: Basic information about a user

Ed chooses to implement a single-page application, adopting a flat minimalist design that results in a horizontal cascade rather than the tree-like structure of Moodle and Canvas.

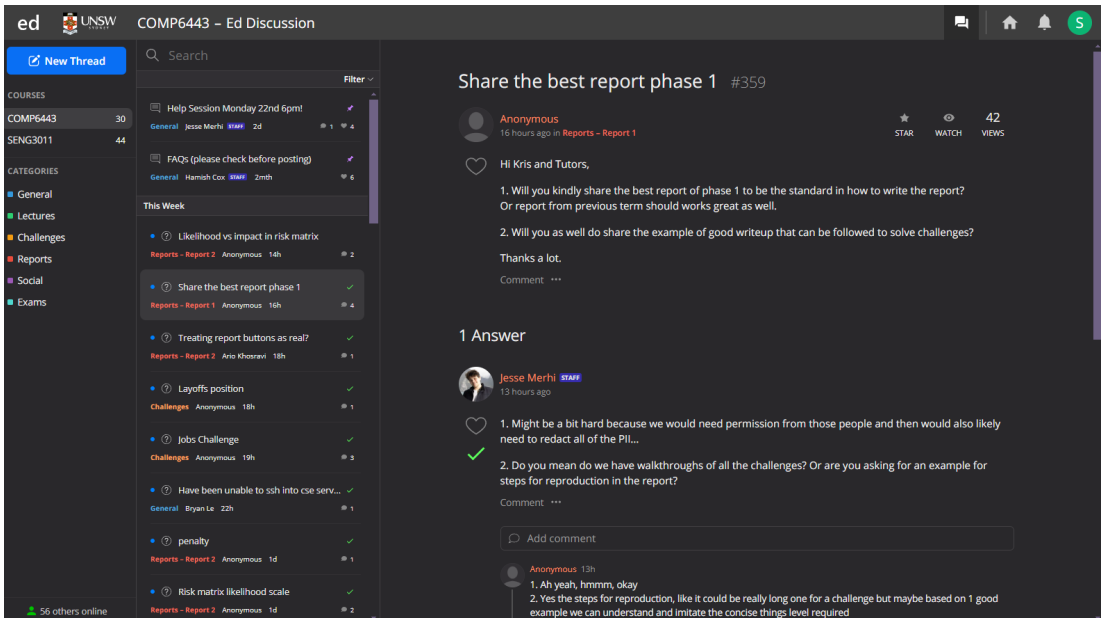

Figure 2.22: An Ed Discussion forum for a course

The page is divided up into three parts, with the main body of a post taking up the large majority of the screen. Basic but crucial information about posts are displayed in the preview, making it easier for users to locate helpful posts. A distinguishing feature of Ed is the indicator on a post that marks whether a post has been successfully answered, which most importantly, appears in the preview. This allows students to easily navigate to relevant answered questions, and staff to identify unanswered posts, reducing the overhead for a user. The search functionality is of great speed and search results are highlighted where relevant. New replies to a post are also highlighted with a green bar, shown below.

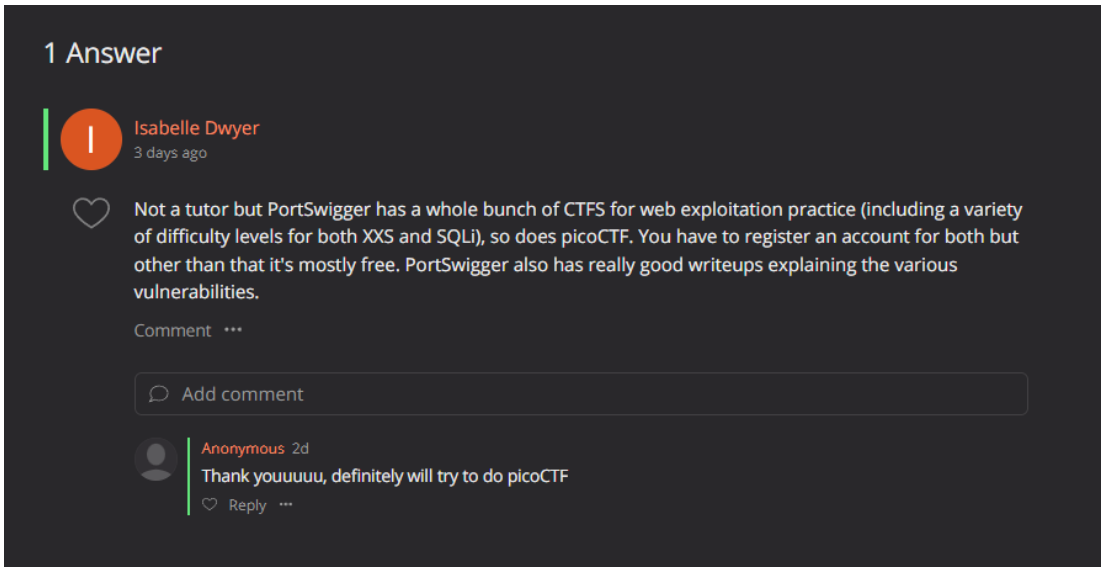

Figure 2.23: Highlighted New Replies

Ed provides a slightly more comprehensive formatting suite, with additional options on top of the standard options that Moodle and Canvas provides such as callouts and code snippets. These can be seen in Figure 2.24 and have a visual representation that clearly differentiates them from normal text.

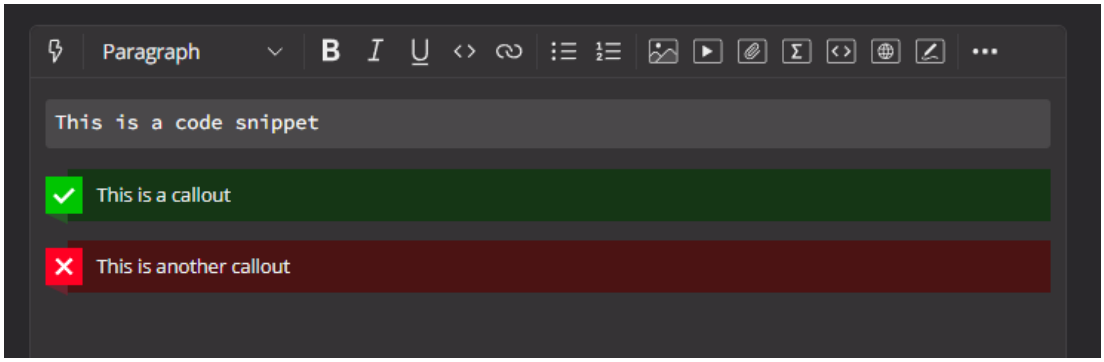

Figure 2.24: Callout formatting options provided by Ed

Ed also offers some unique features - most notably a code compiler with support for numerous languages, and a 'megathread' type post where each reply is counted as a question and can be individually marked as resolved. Furthermore, Ed implements the ability to make 'private' posts - posts that are only visible to the course staff. These type of posts allow students to ask

sensitive questions and reduce the overall load of the main course contact.

By implementing an interface that is easier to navigate compared to other systems, as well as providing more functionality, Ed distinguishes itself apart. By contrast, the forums that Canvas and Moodle implement seem to be designed to act as an assessment component rather than a standalone forum. Ed still lacks some functionality, such as the ability to anonymously reply to posts. While anonymous posting exists, the anonymity only applies to the original poster and their replies. Other users are not able to anonymously reply to any posts.

WebCMS3

The forum module of WebCMS3 exists on a single page, much like Ed Discussion.

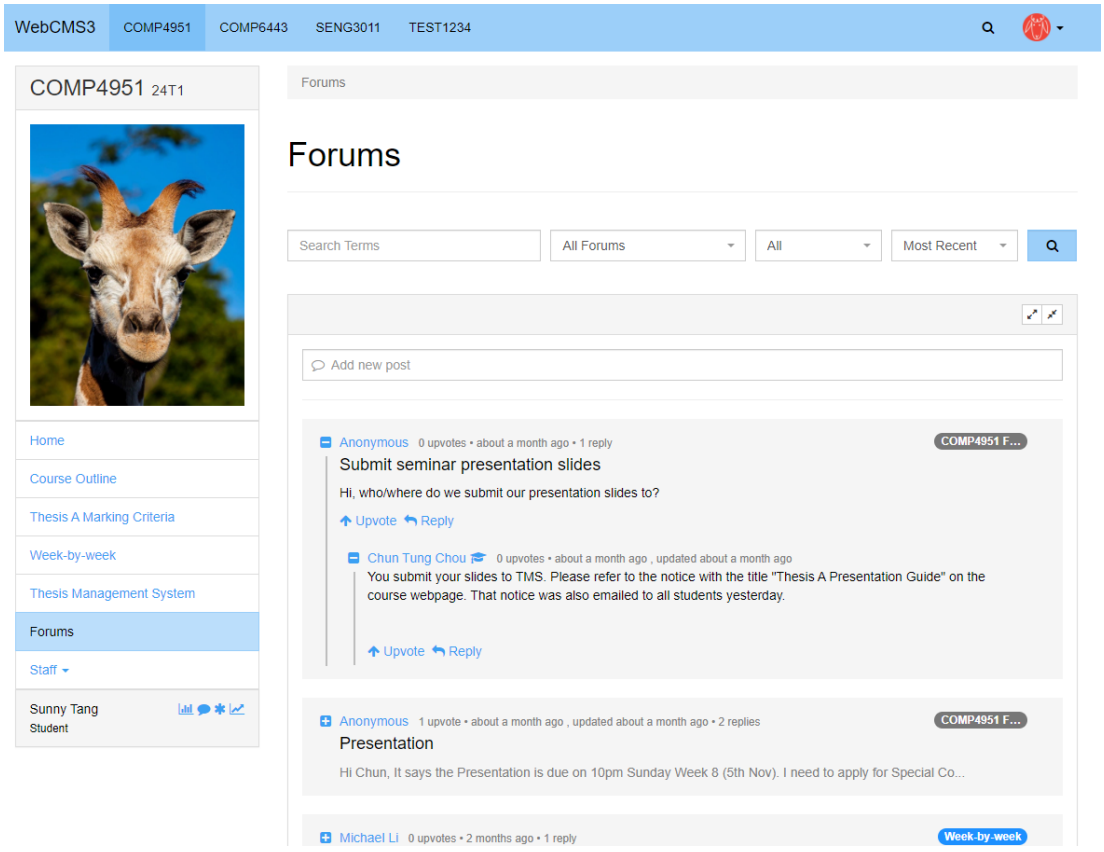

Figure 2.25: WebCMS3 forum overview for a course

At the top lies the ability to filter posts based on a search query or by a category, or sort posts

either chronologically or by popularity. The relevant queries in the search result are highlighted, improving the overall usability of the feature. The text editor provides sufficient formatting options, supporting HTML and providing some niche options. Posts cascade vertically, and expand out in the same direction, with vertical lines to represent the different threads of conversation. WebCMS3 also features an 'endorsement' feature, which can be seen below in Figure 2.26

Figure 2.26: Endorsement feature

This feature enables staff to highlight a students answer, allowing other students to easier identify answers verified by staff. The upvoting functionality allows other users to promote a post, once again making it easier for students to find relevant posts.

However, the endorsement feature does not appear in the post overview, unless the post itself is endorsed. Consequently, if a student replies to a post with an answer which gets endorsed, students scrolling through the list of posts will be oblivious to this fact. Further issues include:

- The lack of a direct reference to an individual post
- The lack of anonymous/private posts
- The slow searching speed

Lastly, one of the main areas of concern raised by students pertains to the design of the forums. As mentioned above, the list of posts within a forum flow downwards. To add to this, opening a post expands its content downwards, further pushing other posts down the page. As a result, a substantial amount of scrolling is required to browse the forums. As WebCMS3 does not implement fixed scrolling like Ed Discussion, this can make scrolling feel like a chore and in conjunction with having the press the "Load More Posts" button to display more posts, the forums provide a slightly frustrating experience.

2.2.3 Daily Updates

The daily updates module of WebCMS3 is unique and even though other systems may have similar implementations may exist, the current issues relate to bug fixing, rather than introducing new features.

WebCMS3's daily updates module involves the timetabling of classes and the record-keeping of student enrollments. As can be seen from Figure 2.27, duplicate entries for the same class occasionally appear.

Figure 2.27: Duplicate class times

Another example of such a case can be seen below.

Lectures

Name	Staff	Day	Start Time	End Time	Weeks	Room
1PGA	Zac Kologlu	Mon	18:00	20:00	1-5,7	ColomboThA
		Tue	18:00	20:00	1-5	ColomboThA
		Mon	18:00	20:00	9-10	ColomboThA
		Tue	18:00	20:00	7-10	ColomboThA
1UGA	Zac Kologlu	Mon	18:00	20:00	1-5,7,9-10	ColomboThA
		Tue	18:00	20:00	1-5	ColomboThA
		Tue	18:00	20:00	7-10	ColomboThA
WEB	Zac Kologlu	-			-	-

Laboratories

Name	Staff	Day	Start Time	End Time	Weeks	Room
W16A		Wed	16:00	18:00	1-5,7-10	H13-W 4034
		Wed	16:00	18:00	1-5,7-10	H13-W 4034

Figure 2.28: Another example for duplicate class times

The next issue pertains to the student enrollment data. On a daily basis, WebCMS3 receives a list of student enrollments from the SiMS and updates its own records for every course accordingly. Within a course, three categories for students exist:

- Enrolled Students - Enrolled students are in the records received from SiMS. These students have access to the course materials.
- Unofficial Students - Unofficial students are not in the latest version of the records sent by SIMS, but have previously been enrolled. Towards the beginning of a term, students who claim to have enrolled in the course are also tentatively put in this category. This category exists to allow students to have access to the course materials on WebCMS3 without being officially enrolled.

- Ex-students - This category contains students who are not in the records received from SiMS and have not been for some time. These students do not have access to course materials.

Occasionally, WebCMS3 does not correctly update its own records of student enrollments, causing headaches for both staff and students alike. The offending student(s) must be manually added or removed, increasing the workload for staff. As we only have access to one source of data for the student enrollment records, we are unable to verify the correctness of the data given to us. The problem may either be caused by bad data being fed into WebCMS3 or the parsing of the data performed by WebCMS3.

2.3 Proposed Solutions

As a standard for all the changes that should be made, the following functional requirements have been defined.

- The changes must retain all existing functionality

The following non-functional requirements have been defined:

- A consistent user interface and user experience

Module specific requirements have been defined below in each of their relevant sections.

2.3.1 Quiz Module

To address the issues raised regarding the quiz module - namely the inability to record multiple submissions and the lack of a timer functionality for quizzes, two solutions should be implemented into the existing infrastructure. They are as follows:

1. A quiz submission viewing system
2. A server-side timer for quizzes

Submission Viewing System

To enable WebCMS3 to keep accurate records of student quiz submissions, the ability to store and render multiple submissions should be added.

The following functional requirements have been defined:

- Backwards compatibility
- Accurate records of all student quiz responses
- Correct rendering of student responses
- An interface for navigating between responses

The following non-functional requirements have been defined:

- Fast performance when loading/rendering responses

Timer

To expand the functionality of the quiz module, the ability to add a time limit to quizzes should be added.

The following functional requirements have been defined:

- A timer rendered on the page showing the students the remaining time
- Automatic submission of the quiz upon timer completion
- Acceptance of submissions before the due time

- Rejection of submissions after the due time
- Secure timer implementation

The following non-functional requirements have been defined:

- Warning should be displayed before commencing a timed quiz
- Warning when attempting to navigate away during a timed quiz
- Automatic saving of answers

2.3.2 Forum Module

To address the issues raised with with the forum module, a new forums web application should be developed, which would provide a necessary user interface overhaul, drastically increase the searching speed, and provide both functionality and extensibility.

The following functional requirements have been defined:

- Staff should be able to create and remove categories
- Users should be able to create posts in a category
- Users should be able to view and comment on posts

The following non-functional requirements have been defined:

- A new, more modern interface
- Users should be able to see a 'like' and comment count, indicating relevance
- Categories should have a colour associated with it for easy differentiation
- Users should be able to create private posts which are only visible to course staff

- Users should be see when a post was created
- Users should be able to see an an indicator of if a post has been answered
- Forum theme should match the course theme
- Staff should be able to delete or private forum posts
- Users should be able to see a overview of posts

2.3.3 Other

To address the issue of students being able to access quizzes directly through the URL prior to the access date, correct user level permission checks should be performed.

The following functional requirements have been defined:

- Students should not be able to view a quiz prior due its active date

Chapter 3

Implementation

3.1 Quiz Module

3.1.1 Submission Viewing System

The changes made to the quiz module to address the lack of flexibility in quiz submissions satisfy the following requirements:

- Backwards compatibility
- Accurate records of all student quiz responses
- Correct rendering of student responses
- An interface for navigating between responses
- Fast performance when loading/rendering responses

System Architecture and Design

The new system has made no changes to the database. A simple interface for navigating between the different responses is rendered through Jinja, the templating engine already integrated

into the existing infrastructure. A class method that previously would only fetch the single response for a quiz by a student now fetches all responses by a student and all uses of the old function have been replaced and/or modified.

End Product

The new system now records all submissions for a quiz and provides a simple interface for navigating between the responses, an example of which can be seen in Figure 3.1.

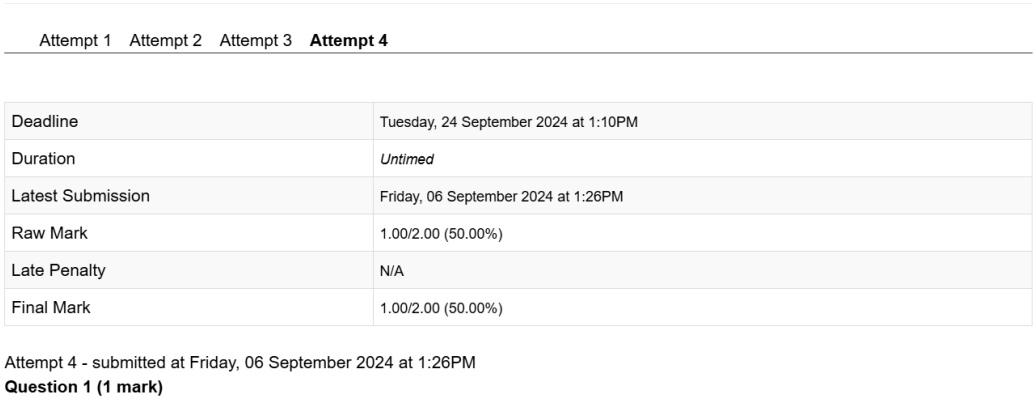

Figure 3.1: New Navigation Interface

The navigation bar sits above the quis table summary, creating an intuitive and streamlined user experience. The attempts are sorted by chronological order, with ”Attempt 1” referencing a student’s earliest submission. By default, the latest attempt is shown. Hovering over each attempt in the navbar changes the standard cursor to a pointer cursor, indicating to the user that the attempt texts are interactive. Clicking on another attempt text seamlessly transitions the body of the page to the display the respective quiz response and bolds the selected attempt text to improve accessibility, shown in Figure 3.2.

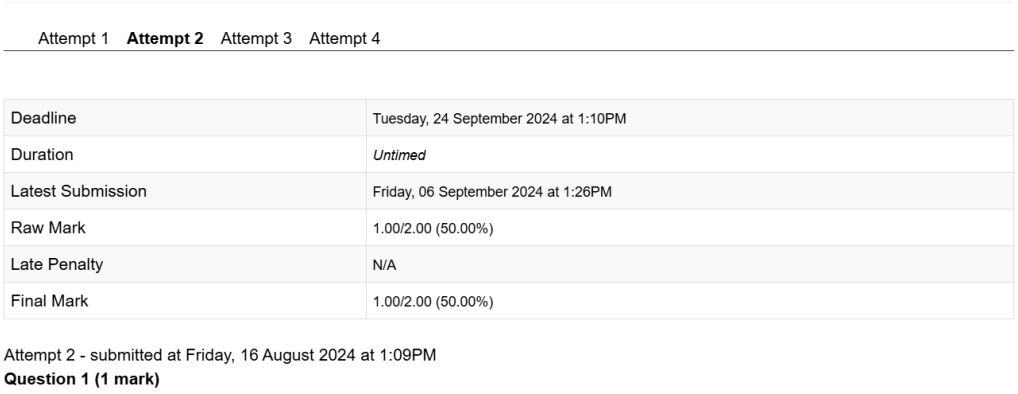

Figure 3.2: Selecting Another Response

The table summary has been modified to display information related to the latest submission. In particular, only the ”Raw Mark” and ”Final Mark” fields were changed. Below this, a simple text sentence displays which attempt the user is viewing, as well as submission time and date of that particular attempt.

Whilst all attempts are marked, only the latest submission is counted when a lecturer chooses to export the marks to SMS through the results page. A new function in the quiz *controller.py* was added to support this, which may also be useful in the future should the most recent quiz responses ever be needed.

3.1.2 Timer

The implementation of the timer in the quiz module satisfies the following requirements:

- A timer rendered on the page showing the students the remaining time
- Automatic submission of the quiz upon timer completion
- Acceptance of submissions before the due time
- Rejection of submissions after the due time
- Secure timer implementation

- Warning should be displayed before commencing a timed quiz
- Warning when attempting to navigate away during a timed quiz

System Architecture and Design

In order to accommodate for the introduction of the timer, changes have been made to the persistence layer. The Quiz class and its corresponding database table have been modified to include a new 'duration' field, which is nullable for quizzes that do not have a time limit. The UserQuizResponse class and its corresponding database table have also been modified with a new 'date_started' field, which records the datetime of when a student first accesses a timed quiz. When this event occurs, the 'date_submitted' field is calculated by summing the quiz duration and the 'date_started' field, with some buffer to account for slow page loading. By default, this buffer time is set to six seconds. Initially, this field is used to store the latest possible submission time, and in the event where a submission before this time is received, it replaces the field with when the submission was received. Furthermore, a students raw and final mark is initially set to zero to ensure backwards compatibility and to account for the case when a student begins a quiz, but no submission is received. If a submission is successfully received, these raw and final mark fields are updated to accurately reflect the real marks a student would receive. Even if a submission is not received, there will still be a record that the student started the attempt.

While answer fields in the UserQuestionResponse class and its corresponding database table may allow for nullable entries, WebCMS3 currently operates on the assumption that for every response to a quiz, if a submission exists for a quiz then there is a well-formed answer to every question for its respective question type. In order to accommodate for the scenario where the quiz is submitted with empty or partial answer fields due to the timer running out, much of the logic to do with marking and rendering of the answers have also been changed.

Lastly, the quiz countdown timer shown on the quiz page not only serves to display the remaining time a student has left, but also to automatically submit the current state of the quiz form when it reaches zero. When a timed quiz page is loaded, the timer begins counting down

from a value communicated to it by the server, as calculated below:

Listing 3.1: Timer Calculation

```
time_remaining = (datetime_started + timedelta(minutes=quiz.duration))
                 - datetime.now()
```

This allows for a timer that persists across refreshes and ensures that even in the case where a user accidentally navigates away, an accurate remaining time is shown.

End Product

The quiz creation page features the addition of a new numeric input field, which can be seen in Figure 3.3.

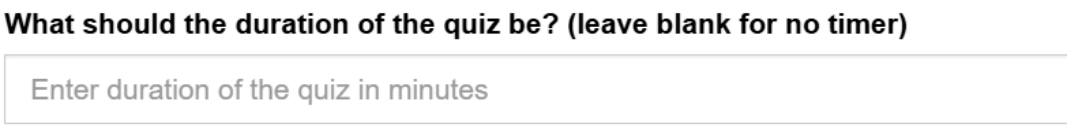

Figure 3.3: Input Field for Time Limit

While negative values may be entered into this field, attempting to create/save a quiz brings up an error message through Flask's Message Flashing, shown below.

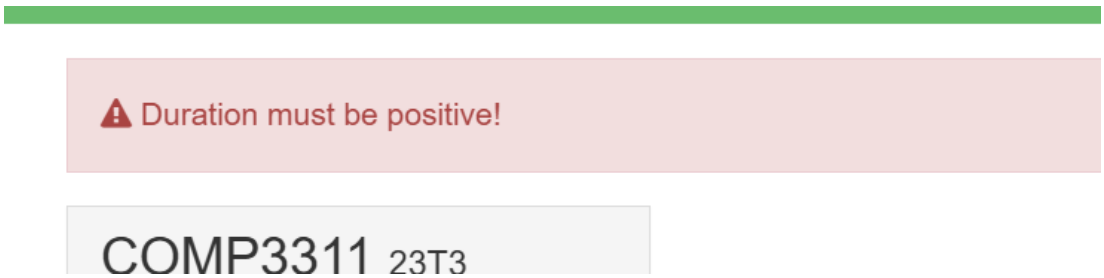

Figure 3.4: Negative Duration Error

A design decision that was made was to prohibit the ability to create a re-submittable timed quiz. Attempting to create/save one brings up the following error message.

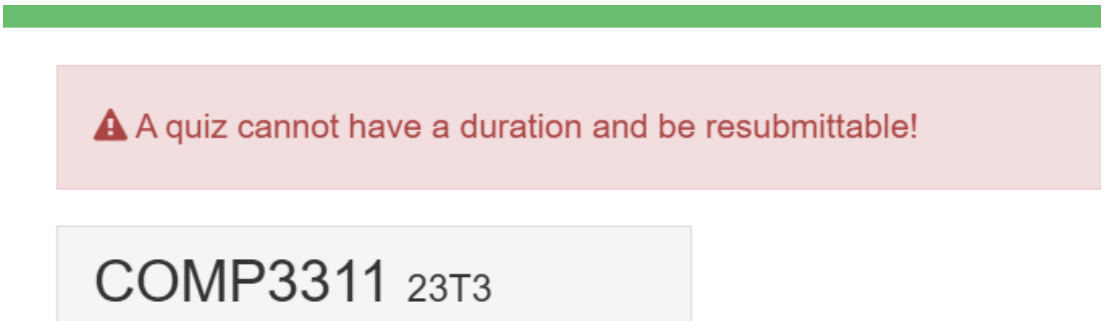

Figure 3.5: Attempting to Create a Re-Submittable and Timed Quiz

In the activity overview page, if a link to a timed quiz is clicked, a confirmation pop-up is brought up and displayed in the middle of the screen, requiring the user's acknowledgment before permitting access to the quiz. This is done in order to avoid any cases of grievances where students accidentally commence a quiz page before they are ready.

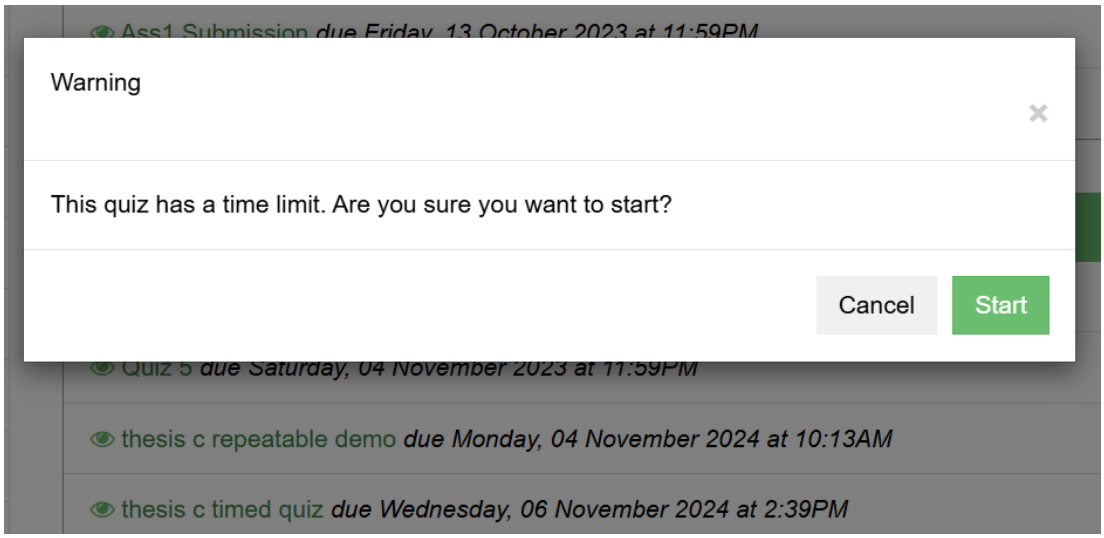

Figure 3.6: Confirmation Pop-Up

The pop-up itself is a Bootstrap Modal and as such, can be easily customised.

When a student enters the quiz page, the timer is displayed at the top right, in line with the title of the quiz in HHh MMm SSs format, shown below in Figures 3.7 and 3.8.

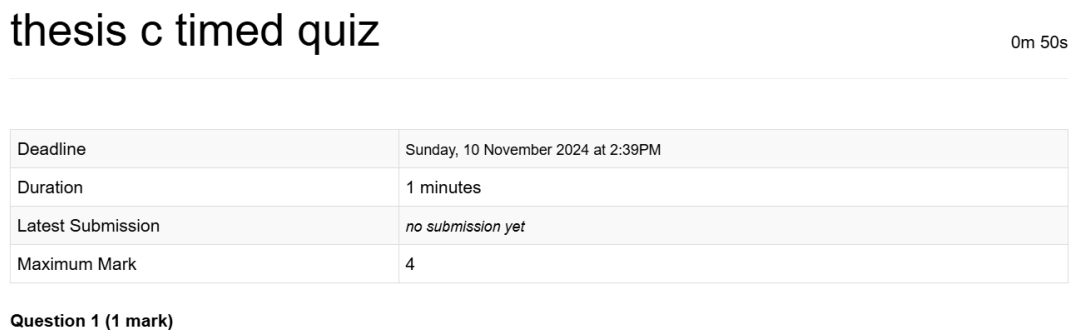

Figure 3.7: Quiz Timer with Minutes and Seconds

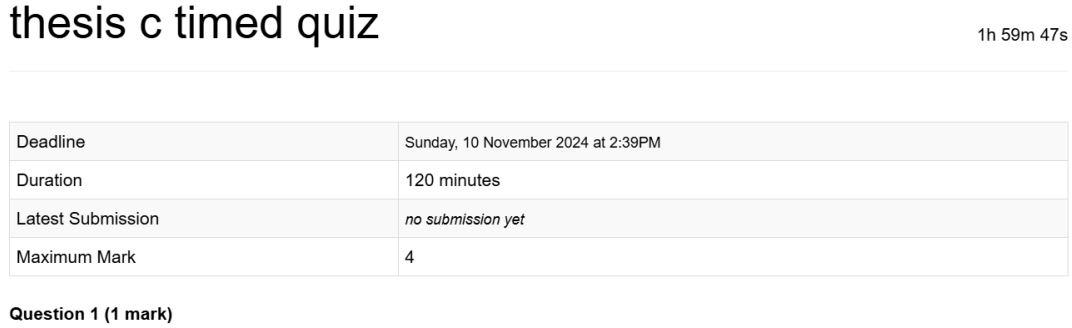

Figure 3.8: Quiz Timer with Hours

When the timer runs out, the system automatically submits the state of the quiz, collecting the user's responses and redirects the user back to the activity overview page with the standard message acknowledging the submitted response.

There is no timer for staff members and staff will not be automatically redirected back the activity overview page.

3.2 Forum Module

To address the issues raised with with the forum module, the foundations for a new system has been put in place. Whilst largely uncompleted, much of the groundwork has been laid, making the road ahead easier for future work to be done. The following requirements have

been implemented:

- A new, more modern interface
- Users should be able to create posts in a category
- Staff should be able to create and remove categories
- Users should be able to see a overview of posts

Changes to System Architecture

In order to support the introduction of a completely new forum system, several changes have been made to the current architecture.

Firstly, the forums are now rendered with the front-end framework ReactJS, replacing the current forums endpoint at '/forums'. React offers the ability to build client-side rendered applications, as opposed to Jinja's server-side rendered templates, simplifying the process of developing a more interactive application and smoother user experience.

Secondly, a new *model.py* file has been added to the forum directory, which contains a *ForumPost* class and a *ForumCategory* class. The current forums make use of a table shared by many other modules of WebCMS3, resulting in slow loading times as the system needs to query through around eighteen thousand database records. The two new classes and their respective tables aim to facilitate much faster loading and searching times and thus improve the user experience. The *ForumCategory* class and its corresponding database table stores the different possible categories for a course forum, as created by the staff. Each category has its own colour, title, and a reference to a parent category, enabling subcategories to be created. The *ForumPost* class and its corresponding database table consists of fields to store information such as:

- Title
- Post Content

- Poster
- Number of likes and replies
- Private status, indicating whether a post is only visible to staff
- A reference to a parent post, in the case it is a comment
- Answered Status, indicating whether a post has been answered
- The category it belongs to
- Time the post was made.

Both of these classes reference an entry in the course offering table, enabling every course offering to have their own forum.

Lastly, the *controller.py* file has been modified to support the front-end, with multiple routes added to it such as, adding categories and posts, and fetching categories and posts.

End Product

Upon navigating the '/forum' endpoint, the user is presented with the following screen:

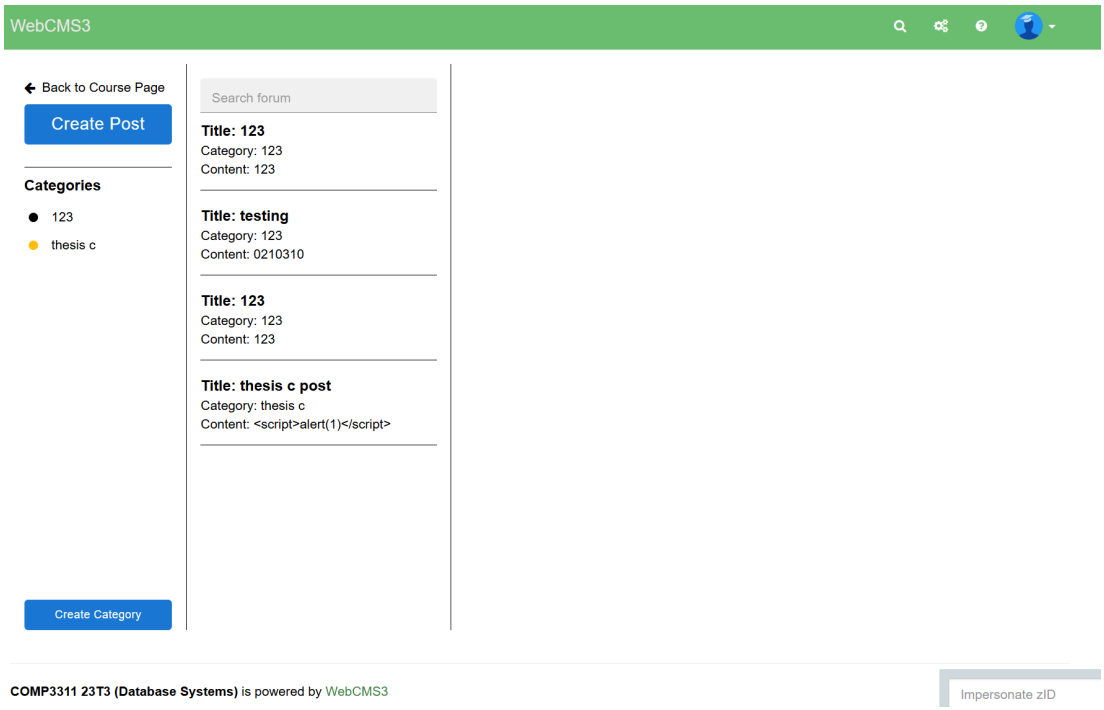

Figure 3.9: Overview of New WebCMS3 Forums

The sidebar, which normally would take up about a quarter of the width of the screen, has been retracted and enables the forums to use the screen space more effectively. The forums follow the same layout as Ed Forums, being divided up into three main sections - one for categories, one for an overview of the posts in the course, and the largest and rightmost section being used to display the main content of a post and other relevant information.

Clicking on the 'Create Category' button replaces rightmost section with a form to create a category, which can be seen below in Figure 3.10,

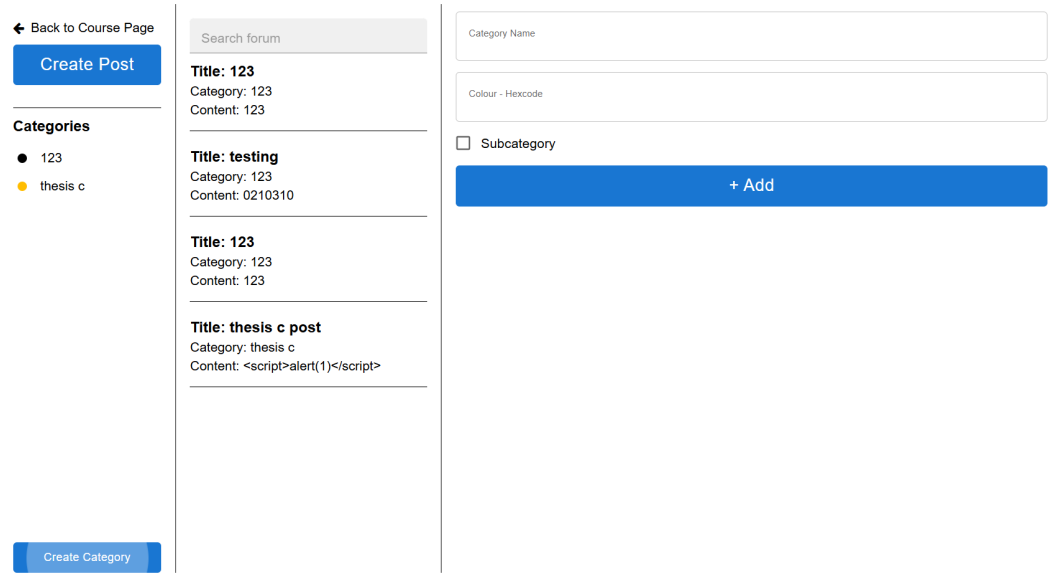

Figure 3.10: Category Creation

Importantly, it should be noted that the value inputted into the field should be a hexcode of the desired colour, and that the subcategory checkbox provides no functionality.

The post creation form, brought up when users click 'Create Post' is shown below:

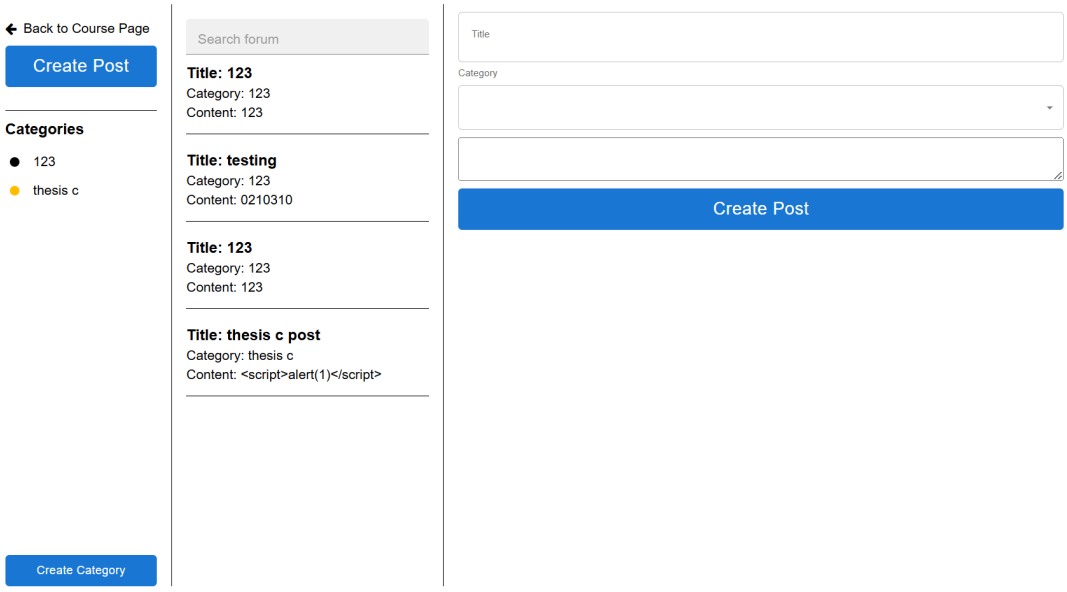

Figure 3.11: Post Creation

Here, users can input a title and their desired post content. A category must also be selected from a dropdown menu, shown in Figure 3.12

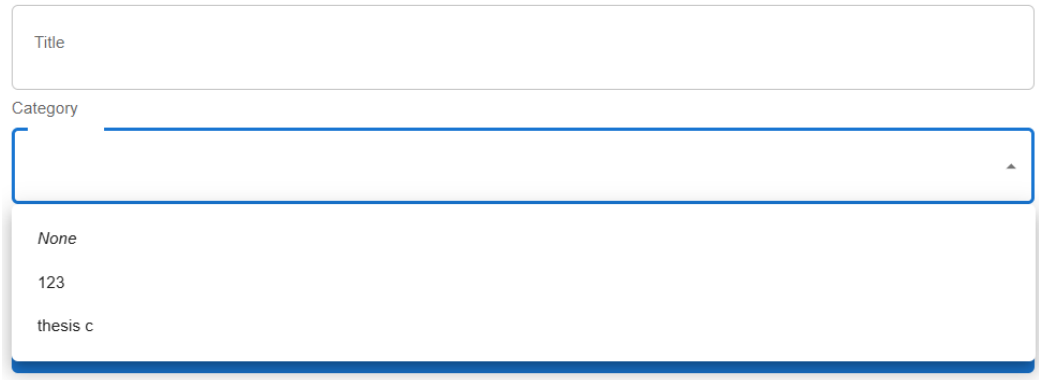

Figure 3.12: Post Creation Category Selection

Once a post or category is created, users need to refresh the screen before seeing new posts and categories. Whilst users may create standalone posts that do not go into any category, this is not the intended behavior.

Hovering over posts in the middle column also highlights them, improving visibility and accessibility.

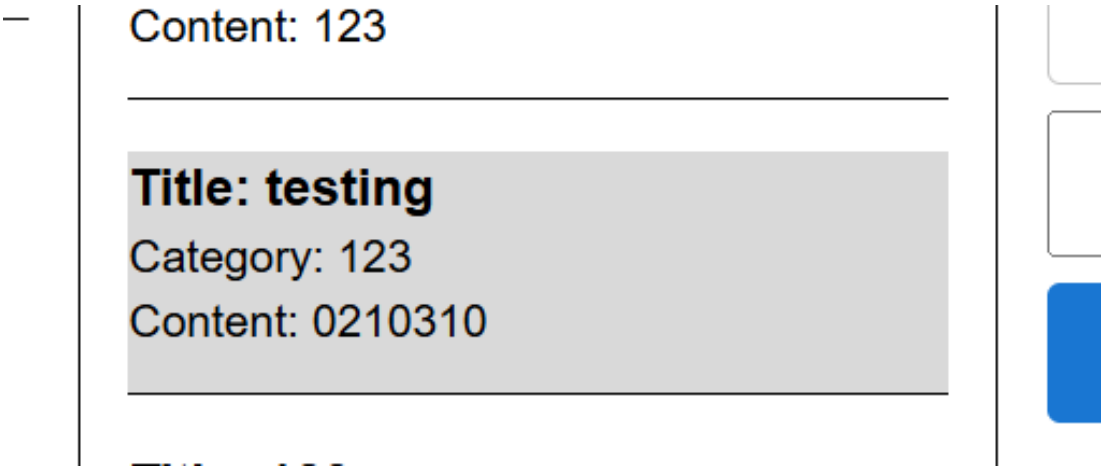

Figure 3.13: Post Highlighting

3.3 Other

The issue where students were able to access a quiz directly through its URL has been addressed.

The following requirements were completed:

- Students should not be able to access a quiz through the URL before the visibility date

System Architecture and Design

No changes to system architecture and/or design was needed.

End Product

Students are no longer able to access a quiz directly through its URL before the visibility date. Doing so redirects them to the 'no permission' page.

The following requirements were completed:

- Students should not be able to view a quiz prior due its active date

Chapter 4

Evaluation

4.1 Usability

For the development process, comments were written where deemed necessary to aid any future work.

4.1.1 Quiz Module

Submission Viewing System

Minimal changes were made to the original designs, namely the removal of the box surrounding the selected attempt. This was mainly done due to the CSS issues, as implementing it well would involve more time. While this may reduce usability in some scenarios, the bolding of the selected attempt should still provide a pleasant user experience. Otherwise, the other components of the navbar follow the original design.

Quiz Timer

Whilst an original design for the timer was not constructed, discussions were held with Dr. John Sheppard in order to implement a usable and accessible design, which have been followed.

4.1.2 Forum Module

As the forums module is largely uncompleted, much of the user interface is subject to change. However, one key aspect of usability that played an important factor of the design was the decision to forums not require consistent vertical scrolling, as well as the selected forum post filling up the large majority of the screen. Whilst the overall design of the forum prototype was followed, some changes were made during the development process. Most notably, the sidebar is not collapsible, but is not accessible at all on the forum page. While this may be detrimental to the overall usability, this was implemented fully due to a lack of time and/or the finicky aspects of UI design. Similarly, most components of the prototype were not implemented due to a lack of time.

4.2 Testing & User Feedback

No unit tests have been written, mainly due to a lack of time. However, some user testing has been conducted by attempting to explore all possible use cases of the implemented features. Most notably, the quiz timer implementation required significant time to fix and debug issues that arose from nullable answer fields.

A few students were asked for their opinion on the forum module. Their opinions, sumarised, were that while certain elements of the forums lacked visual depth, such as the border lines that separated the columns, the overall presentation and user interface looked promising.

4.3 Perfomance & Security

4.3.1 Quiz Module

Submission Viewing System

In order to provide a smooth user experience, all the responses are fetched upon the first render of the page, but are simply all hidden bar the most recent response. Changing the selected response simply hides all the other responses and displays whichever is selected. An alternative solution proposed was to dynamically fetch each attempt whenever the user selected it, but this would only prove advantageous if the user had submitted a particularly large number of responses as the wait between each selection would be inconvenient. For the large majority of use cases, the implemented solution is both faster and more efficient.

No particular security measures have been implemented into the submission viewing system as the changes made were related to rendering objects created by the existing system.

Quiz Timer

The implementation of the timer should have minimal impact on the quiz loading times or any performance.

Initially, the tracking of if and when a student had attempted a quiz was done through storing and read a field in the Flask session. However, this method did not guarantee security as there was the potential for users to manipulate the session token. In particular, if a user deleted their session token, there would be no record that they had started an attempt, enabling them to make unlimited attempts. Later in the development process, all the verification and record keeping was moved server-side to avoid these issues. The last area that needed attention to security was to do with the actual timer itself. As the client could not be trusted, after some discussion with some members of the COMP6443 (Web Application Security and Testing) teaching staff, the decision was made to simply pass a value to the client which the timer would begin counting down from. Then, when the timer reached zero, it would simply submit the form for the client.

In this scenario, the manipulation of the client-side JavaScript timer would have no effects as the server references its own datetime before validating submissions, explained in Chapter 3.

4.3.2 Forum Module

No particular efforts have been made to improve the perfomance of the new forums module. However, as stated in Chapter 3, the new forums should offer much faster performance over the current forums due to the separation from the messages table. Queries should simply be able to filter by course to fetch all the relevant information needed.

No particular security measures have been implemented into the forum module due a lack of time.

4.3.3 Other

User testing has been conducted out to ensure the issue of accessing a quiz through Insecure Direct Object Referencing before the visibility date has been rectified. Due to it only having one use case, this was possible.

4.4 Known Issues

4.4.1 Quiz Module

Quiz Timer

There are a few known issues with the implementation of the quiz timer. Firstly, navigating away during a timed quiz does not save the user's responses. This is currently bandaged with an *onbeforeunload* event, which prevents the user from accidentally losing all their progress.

Another issue lies with the blanks question type. With the current implementation of it, the blanks question type simply fills in the blanks left to right. In the case where a blanks question

contains two blanks and a user only fills in the second blank, the system moves the response to the first blank. In the case where both are filled out, or the first blank is filled out, no issues occur.

Lastly, the statistic graphs on the quiz results page do not correctly display the statistics with multiple submissions, counting all of a students submission as the pool size. Some graphs have had this issue addressed and resolved, but due to the numerous API calls being made when visiting that page, identifying the correct execution path proved challenging and was not able to be done. However, it should be noted that the graphs do not have any effect on the functionality of the quizzes and it is merely a visual loss. As previously mentioned in Chapter 3, a function has been written to obtain only the most response for a student, which may prove useful in addressing this issue in the future.

4.4.2 Forum Module

The only known issue with the forum modules is the possibility of cross site scripting. While ReactJS and the current system some basic protection against this, users are still able to input script tags or other malicious payloads directly into the system.

4.4.3 Other

User Blogs

Members of the COMP6443 Web Application Security and Testing course have reported the potential for XSS on user blogs in WebCMS3. While the exact steps to replicate this have not been found, and the level of execution have not been made clear, this issue should be addressed soon.

Timetables

The issue with duplicated classes appearing on the timetables page has not been fixed. Whilst significant time was spent into identifying the issue, the changes made to the timetable whilst debugging resulted in permanent changes to the database which greatly slowed progress. It is believed that one of the causes is when a tutorial/tutorial-lab changes the location of one of their classes, which may result in duplicate records. Attempts to define an class equality method to assist removing duplicates before rendering proved ineffective, as for some unknown reason all instances of that duplicate class would be removed. For example, in the list where the numbers represent a *ClassSession* object,

Listing 4.1: Sample List

```
[1, 2, 3, 1]
```

attempts to clean up the list would result in every instance of the '1' being removed, whereas the desired result would be only retaining the first instance of the '1'. Changing the definition of the equality method and the duplicate removal logic had no effect,

Chapter 5

Conclusion

5.1 Conclusion

This report details the motivation, requirements and work done to improve the current WebCMS3 system. The changes to each module were discussed with Dr. John Sheppard to identify the scope, and the work done has been explained in detail and evaluated, analysing the design decisions made, as well as a critical review of areas for improvement.

5.2 Uncompleted Work

Unfortunately, due to a look of foresight and underestimation, the ability to regrade quizzes and implement a functioning forum module were not completed. Whilst warning was given by Dr. Jake Renzella with regards to the ambitious schedule planned, the amount of work involved in developing an entire web application in the form of a forum was taken too lightly. The amount of work to implement quiz regrading functionality was also underestimated, as it is believed to require significant database/ORM changes, and ensuring compatibility with the entire quiz module would also be a lengthy task. For example, if we focus on enabling regrading for multiple choice questions, WebCMS3 currently has two question types - a single correct MCQ and a multiple correct MCQ. the quiz question type cannot be modified once a

student submits a response. This preventative measure could be removed, but this is undesirable. One possible solution would be to remove the question type of single correct MCQ, but then extensive work would be needed to ensure backwards compatibility. Lastly, the overall time and effort required to understand an existing codebase also led to delays in the schedule and thus a lack of time to implement all desired features.

5.3 Future Work

Following on from this project, there are a few possibilities for extension.

5.3.1 Quiz Module

Submission Viewing System

- The text that displays the attempt number, as well as the date and time of the attempt could be improved to a more user friendly design, rather than just a text field.

Timer

- The issue with the blanks question type raised in Chapter 4, Section 4.4.1 should be amended before the changes are rolled out. Some form of indexing would be needed address this issue, which may require the blanks question type structure to be modified.
- Ideally, responses to the quiz should be saved as the student works through it. Design decisions would need to be made, such as to when the answers would be saved, or how often they are saved.

5.3.2 Forum Module

There is much to be improved on in the forums. Due to a lack of time, the intention changed to setting up the boilerplate for the forums, as well as implementing some basic functionality.

This took the form rerouting the forums endpoint to React instead of Jinja, working around grid system currently in use, and setting up a basic UI template.

- Implement functionality mentioned in Thesis A
- Maintain a consistent colour theme in accordance with the course colour
- Ensure mobile compliance
- Implement security protections
- Migrating posts from old forums to new forums. Some pseudocode has been written up to assist future endeavours, shown below.

Listing 5.1: Old Forum Migration Pseudocode

```
INSERT INTO forum_post (course_id, title, content, num_likes, poster,
parent_id, time_posted)
SELECT course_id, title, message, likes, user_id, parent_id, create_time
FROM message;
```

Additional filtering by adding a *where* clause to only target forum posts is necessary.

5.3.3 Other

Other areas for future work:

- Address XSS issue in User Blogs
- Address duplicate timetabling issues

5.4 Acknowledgments

I would like to thank Dr. John Sheppard and Dr. Jake Renzella for their continued guidance and assistance throughout this thesis. I would also like to thank the course staff of COMP6443 for their help on certain parts of the work.

Bibliography

[Can24a] Canvas. What are discussions? https://community.canvaslms.com/t5/Canvas-Basics-Guide/What-are-Discussions/ta-p/3 , accessed 13/04/2024, 2024.

[Can24b] Canvas. What options can i use to regrade a quiz in a course? https://community.canvaslms.com/t5/Instructor-Guide/What-options-can-I-use-to-regrade-a-quiz-in-a-course/ta-p/1093 , accessed 15/03/2024, 2024.

[Col] Portland Community College. Regrade a quiz after changing a question. https://www.pcc.edu/online/faculty/technical/regrade-a-quiz-after-changing-a-question, accessed 20/03/2024.

[Fre] Eva Frederick. Looking back at project athena. https://.mit.edu/018/-looking-back-project-athena-distributed-computing-for-students-1111 , accessed 2/03/2024.

[Moo24a] Moodle. Forum settings. "https://docs.moodle.org/403/en/Forum_settings , accessed 4/04/2024", 2024.

[Moo24b] Moodle. Quiz grades report. https://.moodle.org/03/Quiz_grades_report , accessed 12/03/2024, 2024.

[Moo24c] Moodle. Quiz settings. https://docs.moodle.org/403/en/Quiz_settings , accessed 12/03/2024, 2024.

[Pra20] RK Prasad. A brief history of the lms. https://elearningindustry.com/brief-lms-history, accessed 1/03/2024, 2020.

[QE21] Rebecca Esi Quansah and Charles Essiam. The use of learning management system (lms) moodle in the midst of covid19 pandemic: Students' perspective. *Journal of Educational Technology and Online Learning*, 2021.

[QGa21] QGate. A brief history of learning management systems. `http://elearnhub.org/the-history-of-distance-learning-and-the-lms/`, accessed 1/03/2024, 2021.

[She02] John Shepherd. *WebCMS: a Web-based course management system*. PhD thesis, School of Comp. Sci. and Eng., UNSW, 2002.