



UNSW
A U S T R A L I A

School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

Requirements to theses submitted in the Faculty of Engineering

Thesis submitted as a requirement for the degree of
Bachelor of Engineering

Rason Chia	z5084566	Software Engineering
Rebekah Chow	z5160152	Software Engineering
Daniel Ferraro	z5204902	Software Engineering
Emily Ngo	z5164090	Software Engineering
David Nguyen	z5166106	Bioinformatics Engineering
Edward Webb	z5207215	Software Engineering
Allen Wu	z5205003	Software Engineering

Abstract

More and more educational institutions depend on online learning management systems for distributing resources to students, which has been accelerated by the coronavirus pandemic, and so there are a huge number of learning management systems available. These systems have a wide variety of features such as quizzes, blogs, assessment management, integrations with third party platforms such as Zoom and TurnItIn, and more.

However, many of these systems do not have an efficient or useful way to reuse content and resources used in other courses. Time is wasted organising and uploading content for students. Many of these learning management systems are also difficult to use for both teachers and students and are missing key features such as forums and teachers may look to other platforms such as Piazza for these features in conjunction with the learning management system.

Therefore, we will implement a learning management system with an improved UI that is more accessible and easier to use for both teachers and students, and adopts a system where teachers can upload content to a specific topic. This topic then can be reused for other courses, allowing for easier course creation and management.

Acknowledgements

The authors of this thesis would like to extend their deepest gratitude to Professor Maurice Pagnucco for supervision, support and assistance to this thesis, and Dr John Shepherd for his feedback and suggestions as well.

(Expand acknowledgements in the future)

Abbreviations

BE Bachelor of Engineering

UNSW University of New South Wales

LMS Learning Management System

Contents

1	Introduction	1
2	Background	2
2.1	Comparison table	2
2.2	Google Classroom	2
2.2.1	Features	3
2.2.2	Review	4
2.3	OpenLearning	5
2.3.1	Overview	5
2.3.2	Home	6
2.3.3	Course Home Page	6
2.3.4	Performance	6
2.3.5	Forum and Commenting	7
2.3.6	Data Migration	8
2.3.7	Conclusion	8
2.4	Moodle	8
2.4.1	Overview	8
2.4.2	Dashboard	9
2.4.3	Course Pages	10

2.4.4	Forums	10
2.4.5	Performance and Accessibility	12
2.4.6	Data Migration	12
2.4.7	Conclusion	12
2.5	Edmodo	12
2.5.1	Overview	12
2.5.2	Dashboard	13
2.5.3	Assignments and quizzes	14
2.5.4	Data migration	14
2.5.5	Conclusion	15
2.6	WebCMS	15
2.6.1	Enrolments	15
2.6.2	Course Page Feed	16
2.6.3	Email Notifications	17
2.6.4	Quiz and Assignment System	17
2.6.5	Display Content	17
2.6.6	Gradebook and Profile	18
2.6.7	Admin Functionalities	18
2.6.8	Conclusion	18
3	Project Plan	19
3.1	Topic Tree	19
3.1.1	Overview	19
3.1.2	UI/UX	20
3.1.3	Requirements	22
3.1.4	Functional Requirements	22

3.1.5	Timeline	23
3.1.6	Milestones	26
3.1.7	Evaluation	26
3.2	Forums	26
3.2.1	Overview	26
3.2.2	Requirements	30
3.2.3	Backend Assumptions	30
3.2.4	Evaluation	31
3.2.5	Project Timeline	32
3.3	Assessments	34
3.3.1	Overview	34
3.3.2	Stakeholders	37
3.3.3	Functional Requirements	37
3.3.4	Non-functional Requirements	39
3.4	Required Training/Upskilling	40
3.4.1	Back-end Assumptions	40
3.4.2	Front-end Assumptions	40
3.5	Timeline	41
3.5.1	Term 1 (Thesis A)	41
3.5.2	Term 2 (Thesis B)	41
3.5.3	Term 3 (Thesis C)	42
3.5.4	Evaluation	43
4	Conclusion	44
4.1	Future Work	44

Chapter 1

Introduction

Having a set of clear requirements to their thesis is important to student finalising their BE, or other, degree. Such requirements are both in relation to the physical appearance of the thesis, as well as the writing style and organisation. The present document tries to concisely state the theses requirements while appearing in layout and structure as a thesis itself.

Chapter 2 explains the background for this document. Chapter 3 states the style and submission related requirements to theses submitted at the school. Chapter ?? explains content related requirements to theses. Chapter ?? evaluates the thesis requirements template. Finally, Chapter 4 draws up conclusions and suggest ways to further improve the thesis requirements template.

Chapter 2

Background

2.1 Comparison table

Categories	Canvas	WebCMS	Moodle	D2L Brightspace	OpenLearning	Edmodo	Google Classroom
Topic Tree			No		No	No	Yes
Account			Yes		Yes	Yes	Yes
Course Pages			Yes		Yes	No	Yes
Assignments			Yes		Yes	Yes	Yes
Dashboard			Yes		No	Yes	Yes
Quizzes/Exams/Tests/Polls			Yes		Yes	Yes	Yes
Forums			Yes		No (comments system only)	No	No
Multiplatform Access			Yes (Has separate mobile application)		Yes	Yes	Yes
Accessibility			Yes		Yes	No	Yes
Grading			Yes		Yes	Yes	Yes
Attendance			Yes (Requires a plugin)		No	Yes	No
Calendar			Yes		Yes	Yes	Yes (Provided by Google Calendar)
Enrolment			Yes		Yes	Yes	Yes
Blogs/Wikis/Discussions			Yes		Yes	Yes (Discussion only)	No
Notifications			Yes		Yes	Yes	Yes
Lectures and Tutorials			Yes		Yes	No	No
Third Party Integration			Yes		No (API is provided)	Yes	Yes
Inbox/Messaging			Yes		Yes	Yes	No
Gamification/Karma System			No		Yes	Yes	No
High Quality User Interface			Yes		No	Yes	Yes
Open Source			Yes		No	No	No
Data Migration			Yes		No	No	Yes
Performance (out of 10)			6		1		8

2.2 Google Classroom

Google Classroom is an online learning management tool provided by Google which offers a free and easy tool for helping educators manage and assess progress of students. It has become one of the most popular learning management platforms adopted in schools across Australia. The main selling points of this platform are:

- The ability to easily manage students learning
 - Allowing students join classes directly or by sharing a code or link
 - Setup a class quickly and create class work that is displayed on students' calendars
 - Providing communication with students' guardians and automatically send them updates

- The ability to easily measure students progress
 - Storing frequently used feedback to be used for fast and personalised responses
 - Allowing teachers to grade consistently and transparently with rubric integrated student work
 - Providing students with a plagiarism checker to help them produce their own original work
- The ability to provide collaboration between students and teachers
 - Connecting students and teachers online in virtual classes
 - Allowing communication of announcements on the Stream page
 - Enabling face to face connections with students using Google Meet
- Keeping data secured
 - Authenticating users through a login feature
 - Restricting classroom activities to members
 - Assuring data is never used for advertising purposes
- Allowing for third party apps to be integrated in learning for example:
 - Schoology: Gives educators and teachers generate insights on student learning
 - Hapara: Help students build executive functioning skills

2.2.1 Features

Google classroom provides the all of the basic functionalities and some unique features of a learning management system.

- Allows teachers to create classes/groups which students can join through a link or code
- Allows users to create announcements in which others can comment within the class
- Allows teachers to create classwork by creating questions, assignments, quizzes or class material
- Classwork can be scheduled and the due dates of the classwork will show up on students' calendars
- Classwork, posts and announcements can be reused
- Other Google applications are integrated within Google Classroom, offering more utility for all users
- Allows teachers to input grades which are viewable by students

2.2.2 Review

In terms of the aspect of ease of use, flexibility and usability Google Classroom offers teachers to reuse questions or sets of questions for classwork. Figure 2.1 highlights the functionality that allows teachers to reuse classwork. While reusing classwork teachers are able to then set separate parameters such as set the due date, add more questions, change the topic and change mark allocation.

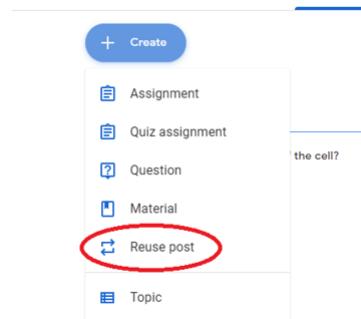


Figure 2.1: Teachers can reuse posts

Topics can also be created in Google Classroom. Topics allow for categorisation of class work. An example of the topics is shown in figure 2.2. The example displays 2 topics, each with 1 question. One functionality that is not implemented is the use of prerequisites for topics. For example, if topic1 was a prerequisite for topic2, then students who have not completed topic1 cannot access topic2. This design does not show how each topic is linked with each other which could be improved on.

A screenshot of the Google Classroom 'Topics' section. It shows two topics: 'topic2' and 'topic1'. 'topic2' contains a single question: 'What is a UML diagram' posted at 9:40 PM. 'topic1' contains a single question: 'What is the powerhouse of the cell?' due by Mar 31.

Figure 2.2: List of topics

One interesting feature within Google Classroom is the integration of other first party applications such as Google Calendar and Google Drive. Google Calendar is a very popular task reminder and calendar application and would benefit greatly with the incorporation of displaying due dates for classwork from Google Classroom. Google Calendar allows for the easy visualisation of important dates and thus is suitable for Google classroom. Google Drive is also a very popular file hosting server where files can be easily shared with other people. For the case of Google Classroom, it can be utilised to share classwork with students.

Due to Google Classroom being affiliated with Google many of the other platforms

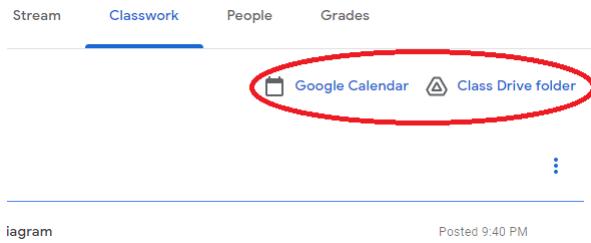


Figure 2.3: Google Drive and Google Calendars are integrated

related to Google are integrated very well. Announcements can be created which have Google Drives and YouTube videos attached which further improve the usability of Google Classroom.

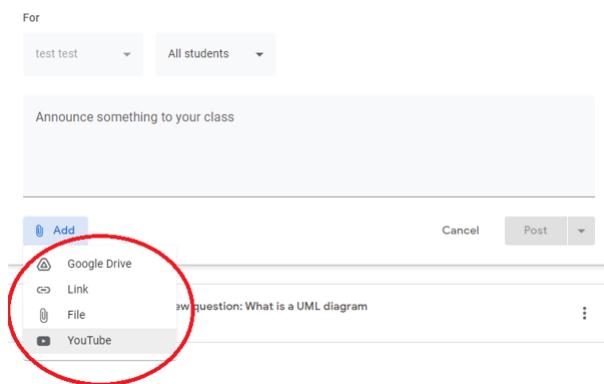


Figure 2.4: Google Drive and Youtube are integrated

Overall Google Classroom does not have many unique features that make it an outstanding learning management platform; however, it does provide the integration of other Google platforms such as Google Drive, Google Calendars and YouTube.

2.3 OpenLearning

2.3.1 Overview

OpenLearning is a ASX listed company that provides an LMS for educational providers. It was founded by Adam Brino, Richard Buckland (COMP6441 lecturer) and David Collien and is used in security courses at UNSW.

It works with universities such as UNSW and Taylor's University (based in Malaysia) to deliver MOOCs (massive open online courses). It also features an LMS that educational institutions can use. Some courses at UNSW currently use it such as COMP6441 (Security), and many universities such as UNSW, UTS, ACU, Charles Sturt University and more also use the platform.

Unlike other content management systems such as Moodle, OpenLearning is not free

or open source and can be quite costly.

2.3.2 Home

When logged in, the home page of OpenLearning is quite confusing.

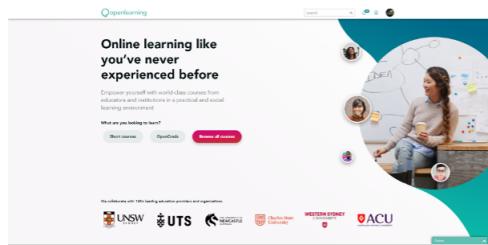


Figure 2.5: OpenLearning Homepage

For students, it can be quite confusing how to reach their courses. The home page clearly directs you to search for a new course offered by OpenLearning, not a course you are already enrolled in.

The balloon icon in the top right should be clicked to reach my courses.



Figure 2.6: This icon is confusing and it is not clear how to get to my educational resources.

2.3.3 Course Home Page

The course home page UI is good, with clear links to information and important announcements being easily viewable.

2.3.4 Performance

The overall performance of OpenLearning is poor, with pages feeling slow and sluggish with Google Lighthouse, a performance testing tool, giving 14/100 for performance. However, accessibility seems otherwise quite good, with a 90/100 score.



Figure 2.7: OpenLearning Course Home



Figure 2.8: Google Lighthouse Result

2.3.5 Forum and Commenting

Timothy Charles Brunette 10 months ago

Another lot of security breaches: <https://www.bleepingcomputer.com/news/security/hacker-group-floods-dark-web-with-data-stolen-from-11-companies/>

What's interesting is that the data isn't even worth that much \$500 - \$5000 per database.

Hacker group floods dark web with data stolen from 11 companies

A hacking group has started to flood a dark web hacking marketplace with databases containing a combined total of 73.2 million user records over 11 different companies.

BleepingComputer / Lawrence Abrams / May 10, 2020

Cary Guan 10 months ago

Sums up the entirety of week3!

Figure 2.9: OpenLearning comments system

OpenLearning makes it quite difficult to communicate with other students, with a com-

ment feature available but not searchable or sortable at all. There does not seem to be a forum feature either, and so in courses such as COMP6441, it can be very difficult to collaborate and communicate with classmates.

2.3.6 Data Migration

Course Setup > Exports

1. Student administration data, such as enrolments and payments
2. Student engagement data, such as posts and course completion summaries
3. Course pages

How do I export course information?

Last Exported	Export Name	Format	Size	Re-export	Download
22 January 2019	Class A: Posts	csv	4798	Run Export	Download
22 January 2019	Class A: Payment summary	csv	228	Run Export	Download
22 January 2019	Class A: Module completion summary	csv	2708	Run Export	Download

New Export

Figure 2.10: Exporting data in OpenLearning

Data migration seems quite poor in OpenLearning. There are no support articles available on how to import data from another course, and so it seems quite difficult to create a new course. There is a basic system to export data into a csv file, but there does not seem to be a way to import data either.

2.3.7 Conclusion

OpenLearning is a good platform for instructors and students, with a huge feature set with blogs, commenting, and a very flexible course setup. However, the UI, performance and stability can be significantly improved to be more competitive with other CMS platforms. A searchable forum can also be implemented, and little data migration features can hinder courses that use the platform, as time is wasted uploading and curating content for courses.

2.4 Moodle

2.4.1 Overview

Modular Object-Oriented Dynamic Learning Environment, more commonly known as Moodle, is a learning management system founded in 2002 by Martin Dougiamas. Supporting over 60% of higher education institutions around the world, including UNSW,

Moodle provides an open-source, personalised learning platform aimed at supporting both students and educators. Moodle's modular design is centered around providing a personalised experience, giving users' the flexibility to build and access courses in a way that suits them. With a vast library of plugins, and the ability for developers to design their own, Moodle really provides users with unlimited functionality.

2.4.2 Dashboard

When a logged-in user first accesses Moodle, they are directed to the Dashboard which consists of content blocks for each of the user's courses. User's are provided with numerous ways to customise the dashboard, including:

- Hiding and showing individual courses
- Filtering courses by past, present and future
- Sorting courses alphabetically or by date
- Displaying courses as cards or in a list

Users can use these customisation tools to organise the dashboard based on their needs and as a result, easily find a desired course.

The screenshot shows the Moodle Student Dashboard. On the left, there's a sidebar with sections like 'Moodle 3.9 Upgrade Features' (including 'Major Upgrades' like 'Activity Stream Overhaul' and 'Forum Grading and Improvements'), 'Private files' (empty), 'Upcoming events' (empty), and a 'Calendar' for March 2021. The main area is titled 'Course overview' and displays a grid of course cards. Each card includes the course name, a thumbnail, and a brief description. Some cards have custom backgrounds with text like 'Engineer', 'T3 2020', '9814 LICENCE', 'FAT MYT', and 'BADS - School of...'. The courses shown include 'COMP9321-Computer Systems Fundamentals - S2/17', 'COMP9321-Software Eng Fundamentals - S2/17', 'COMP9321-Database Systems T3 2019', 'COMP9331-Computer Networks & ...', 'COMP9321-Software Engineering T1 2020', 'COMP9321 Data Services Engineering T1-2020', 'COMP9341-Artificial Intelligence Term ...', 'GENS111-Big Fat Myths T3, 2020', 'MATH1231-Mathematics 1B (S2_2017)', 'MATH1241-MATH1231-Maple Lab Test Booking (2017 ...)', 'MATH1241-Information, Codes and Ciphers (2020_T3)', and 'SENG2021-Software Eng Workshop 2B - S1, 2018'.

Figure 2.11: Moodle’s Student Dashboard

2.4.3 Course Pages

A course page consists of various links to resources. These links can be separated into collapsible subsections which help with the organisation of a course. Course admins can customise the subsections based on how they would like to structure a course. Course admins can also customise the way in which students work through course content by blocking access to resources until certain criteria have been met.

The course page sidebar can also be edited to include different sections like course contacts, calendars, announcements and useful links.

Within a course page, students can access announcements, discussions, forums, quizzes, polls and course resources like lecture notes and assessment notifications. Course admins can open assignment submissions and provide students with assessment feedback and grades.

2.4.4 Forums

Moodle provides course admins with the ability to create one or more forums attached to a course page. Students and admins are able to post to the forum, reply to others, star posts and subscribe to conversations. There is no easy way to search through

Figure 2.12: Moodle's Student Course Page

forum posts which can make it difficult to find information, especially when the post list is long. There is also no functionality for filtering or sorting posts. Overall, the forum functionality in Moodle is pretty basic and could use a lot of improvement when it comes to the organisation of information and the overall experience.

Discussion	Started by	Last post	Replies	Subscribe
Literature Survey for Thesis Seminar			1	<input type="checkbox"/>
Extension			0	<input type="checkbox"/>
Presentation list			1	<input type="checkbox"/>
Extension			4	<input type="checkbox"/>
Due Dates/Submissions page			1	<input type="checkbox"/>
Modify thesis details			1	<input type="checkbox"/>
Thesis Confirmation			1	<input type="checkbox"/>
Recording of Thesis Introduction Lecture			1	<input type="checkbox"/>

Figure 2.13: Moodle's Forums

2.4.5 Performance and Accessibility

The performance of Moodle is average, with Google's Lighthouse performance tool giving it a score between 60% - 70%, depending on the amount of content on the page. A similar score was achieved for the accessibility of the website, with the biggest issues being lack of contrast between background and foreground colours, as well as HTML elements missing the appropriate attributes required for assistive technology.

2.4.6 Data Migration

Moodle provides users with the option to backup and restore data associated with a course. Automated backups for courses can be set up to reduce the risk of losing data. These backups can also be used to migrate data to a different platform. Admins are able to easily select which parts of the course they would like to backup in the backup settings. The backup files can then be easily saved and restored at any time.

2.4.7 Conclusion

While the user interface, performance and accessibility could use some improvements, Moodle successfully delivers all the required functionality need to support students and educators.

2.5 Edmodo

2.5.1 Overview

Edmodo is a learning management system founded by Nick Borg, Jeff O'Hara and Crystal Hutter in 2008. Edmodo offers an "all-in-one LMS" which includes essential management tools for teachers and a platform for which teachers and their students can collaborate and communicate outside of class. The learning management system serves as a complementary tool catered towards primary and secondary education with a focus on classes rather than stand-alone courses (such as those in tertiary education). Edmodo offers the following features:

- Assessment tracking: ability for teachers to see remaining submitted assessments to review, reviewed assessments, scheduled assessments to release
- Grading and recording attendance: ability for teachers to set grades for a student's assessment and fill in attendance for a class

- Collaboration and communication: ability to create a post, discussion, poll and direct messaging
- Assessment options and interactive activities: ability to create assignments, quizzes and gamified activities
- Unlimited storage space: ability to create files in a single click (Word Document, PowerPoint Presentation and Excel Worksheet) and also upload any files
- Social media platform: students and teachers can interact and view others' posts and shared resources, allowing them to learn more about specific topics or develop skills
- Calendar: shows due dates of upcoming assessments, quizzes and events created by the teacher

2.5.2 Dashboard

The dashboard is similar to dashboard of social media platforms, encouraging discussions amongst the students and their teachers as well as keeping students informed of homework and assessments assigned by the teacher.

Figure 2.14: Teacher's view of the dashboard on Edmodo

2.5.3 Assignments and quizzes

The creation of assignments and quizzes in Edmodo is very simplistic, however provides limited options and features in which the teacher can use. There is no ability to customise the assessment and teachers are forced to provide extra details in the attachment section in the form of a file.

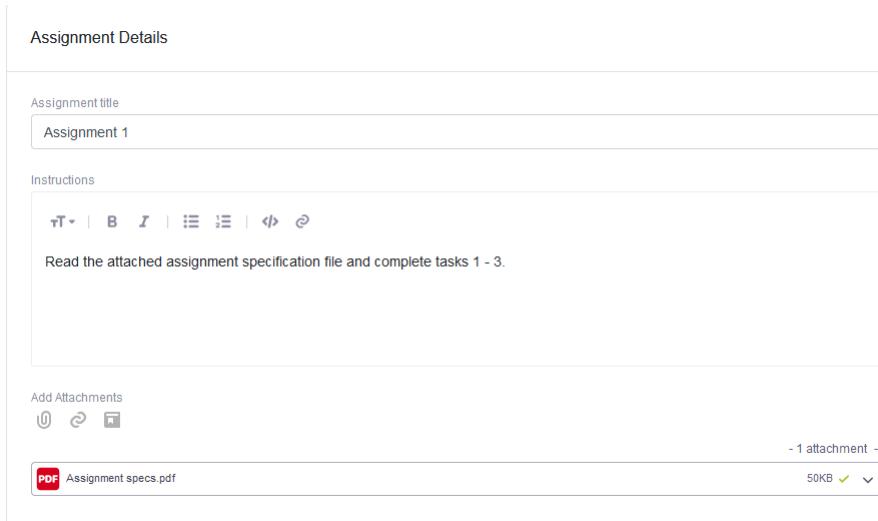


Figure 2.15: Creating an assignment on Edmodo

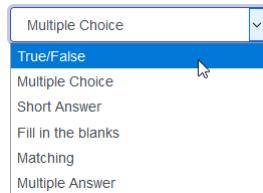


Figure 2.16: Options offered for a quiz question on Edmodo

2.5.4 Data migration

Edmodo does not provide any features to import packages or courses from other learning management systems and is not SCORM compliant. This is due to its nature of class content and assessments being in a fixed structure. Teachers are only offered the ability to copy from existing assignments and quizzes made on the Edmodo platform. New teachers transitioning from another learning management system to Edmodo will have no option to reuse their past course content or assignments.

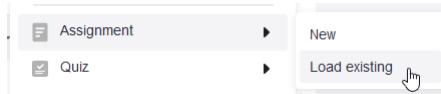


Figure 2.17: Loading an existing assignment on Edmodo

2.5.5 Conclusion

Edmodo offers a clean, social-media style user interface familiar with younger users and provides the essential management tools for teachers, however is one of the weakest learning management systems in terms of re-usability and flexibility for its features.

2.6 WebCMS

WebCMS is a LMS designed and used internally for UNSW CSE courses. WebCMS is a browser based LMS that allows unauthenticated users to only see course announcement page and other pages require an authenticated user with correct enrolment access to the course.

2.6.1 Enrolments

WebCMS has a robust enrolment system that is managed by lecturers and course admins. Course Staff creates a course for a specific term and enrolls students from enrolment list. Students can be enrolled into multiple courses for each term.

- WebCMS has multiple roles that can be assigned to individuals for specific courses (Lecturers, Students, Tutors, Course Admin). Each page in the course can be customised to be accessible by users with specific roles.
- Users enrolled into multiple courses will be able to navigate through the courses via the nav bar at top of page. Users can have different roles for each course. (Student in course A and Tutor in course B)

The screenshot shows the WebCMS student course page for COMP9315 21T1. The top navigation bar includes links for WebCMS3, COMP9315, and COMP9321, along with search and user icons. The sidebar on the left contains links for Home, Course Outline, Q and A, Course Work, Videos and Slides, Prac Exercises, Theory Exercises, Assignments, PostgreSQL, Activities, Forums, Moodle, Timetable, Staff, Rason Chia (Ex-Student), and a notifications icon. The main content area has three main sections: Notices (containing posts like 'Week 9 Videos/Slides' and 'Assignment 2 Testing Page'), Upcoming Due Dates (listing 'Signature Indexes - Specification' with a due date of '5 days from now'), and Assignment details (including 'Assignment 2 Testing' with a note about rounding). The overall theme is green and white.

Figure 2.18: WebCMS's Student Course Page

2.6.2 Course Page Feed

Each Course has a homepage that users will land on when navigating to the course. This main page consists of 3 main components, a static navbar to navigate course content, Course Notices / Announcements and due dates for deliverables.

- Course Notices / Announcements can be posted by lecturers or staff member. These posts will automatically generate an email notification to all enrolled users of the course. Posts are visible to all users and provide course wide information like release of assignments and exam information.
- Due dates of deliverables are retrieved from quizzes and assignments that have a due date set. The panel will show the number of days remaining as well as a link to the specific deliverable.
- The static sidebar provides an indexed view of course content. Content links can be grouped under a heading. This allows users to quickly navigate through to the content they are looking for. This sidebar also display the authenticated user's name, role in the course, shortcut links to user's gradebook, wiki and blog.
- Course Admins can set course theme color which is propagated through all the above elements as well as throughout all pages for the specific course. Each course can have a different theme color and this helps users differentiate which course they are looking at.

2.6.3 Email Notifications

WebCMS has a email notification feature which can be triggered automatically on various conditions. By default, any course wide notices always triggers email notifications.

- Users have the ability to subscribe to their posts (comments, replies, forum posts) which will automatically trigger notifications if there has been a new update to their posts. This can be toggled in the settings menu.

2.6.4 Quiz and Assignment System

WebCMS allows multiple quizzes and assignments to be created for each course. These deliverables have a due date and can be fully customised, automarked and results displayed directly into the user's gradebook.

- Quizzes can comprise of only multiple choice answers or have a mix with short & long response answers. Correct answers can be set for each quiz, once quiz is due, users are not allowed to make anymore submissions and quizzes can be automatically marked by WebCMS.
- Assignments can be created with multiple tabs (Specification, Make Submission, Check Submission, Collect Submission). Specification can be formatted or displayed through embedded pdf. Make, Check and Collect Submission functionalities interacts with CSE give command line tools for specific assignment.

2.6.5 Display Content

WebCMS has a robust page formatting toolset which allows the user to format content with headers, create lists and embed content (videos, images).

- Course Setting can restrict that only lecturers or staff members can create or edit the page.
- Pages created can be pinned into the static navbar on the platform for easy access
- Pages can be accessed by direct link to its content resource number that is unique to each page. e.g.<https://webcms3.cse.unsw.edu.au/COMP9321/21T1/resources/59281>
- Pages can be embedded with pdf files which provides a basic pdf viewer component so users can view pdfs while remaining on the platform

2.6.6 Gradebook and Profile

WebCMS gradebook stores marks from deliverables like quizzes, labs, assignments. User Profiles also has a blog and wiki feature where users can document notes as well as store a blurb about themselves which is visible to other users.

- Quizes which are set to automark can record their scores automatically into the gradebook.
- Marks for Quizes, Labs and Assignments can be manually inputted into the gradebook for each user. This feature can be restricted to be used by staff members only.
- Gradebook allows for total of specific marks to be displayed on the gradebook. This is done to provide a summary view of a student's score over all the deliverables.

2.6.7 Admin Functionalities

WebCMS administrative functionalities can be restricted to certain user groups like staff members. Admin Functions allow staff members to create and manage course content & structures.

- Course Creation Tools: Creating Brand New Courses and Stetup, Cloning previous courses, Link Course to myunsw enrolment course student lists
- Course Management Tools: Setup deliverables for the course (Quizes, Assignments, Labs), Setup Course Content (Lecture Slides, Labs, Assignment Pages, Quiz Questions),
- Student Interaction Tools: Setup pages to have commenting functionality, Setup forums, Setup Notification Settings

2.6.8 Conclusion

WebCMS is a comprehensive and robust Learning Management System, however it does fall short of some other LMS in the market. WebCMS is not an open source platform that can easily recieve external data or export its own data, it is built as a platform to store / share content but it does not engage students directly with any self serve content like gamification features or self serve modules.

Chapter 3

Project Plan

3.1 Topic Tree

3.1.1 Overview

Most learning management systems do not have a simple and easy method to import course material and resources from other courses. Some LMS platforms such as OpenLearning have no method of importing any data at all, and other platforms such as Canvas allows you to import crowd sourced material into your own course, however still does not allow you to import topics of resources.

This new LMS will have a new topic tree feature, which will allow teachers and academics to add course material under a specific topic instead.

Each topic will have prerequisites of other topics, for example the topic Graphs will be a prerequisite for the topic Depth first Search.

Topics A topic is a collection of course material that is related to a particular subject. For example, in UNSW's Introduction to Programming course (COMP1511), one of the topics in the course is "Pointers", which contains all course material related to pointers. Instructors can choose what constitutes as a topic, but a topic in this case is part of the course material of an entire course - it does not constitute all the course material of a single course. For example, there is no topic called "Introduction to Programming" or "COMP1511" as that is virtually unusable as not all of this content would be imported into another course.

In the above example, pointers, structs and discrete mathematics must be learned in order to learn graphs. Likewise, graphs and recursion is required to learn depth first search.

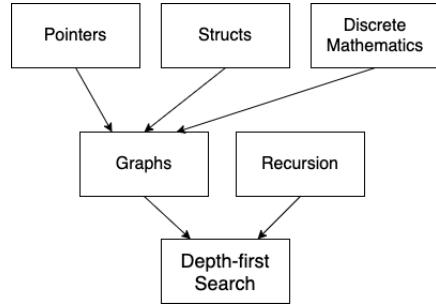


Figure 3.1: Example of a topic tree with Depth first search

Specific courses in this new LMS would be created by choosing the topics this new course has, and all course materials under each topic would be imported into the respective topic. This allows for faster creation of courses, and course materials can be reused more easily.

3.1.2 UI/UX

Instead of only a graph view, we opted for a more traditional UI for the topic tree as well. This is because a graph based UI may be unstable and difficult to use compared to a more traditional UI.

The following is a mockup of a possible UI, and will be changed in the future.

Each group of topics is listed in card format. The topics in each group can be viewed by clicking on the topic.

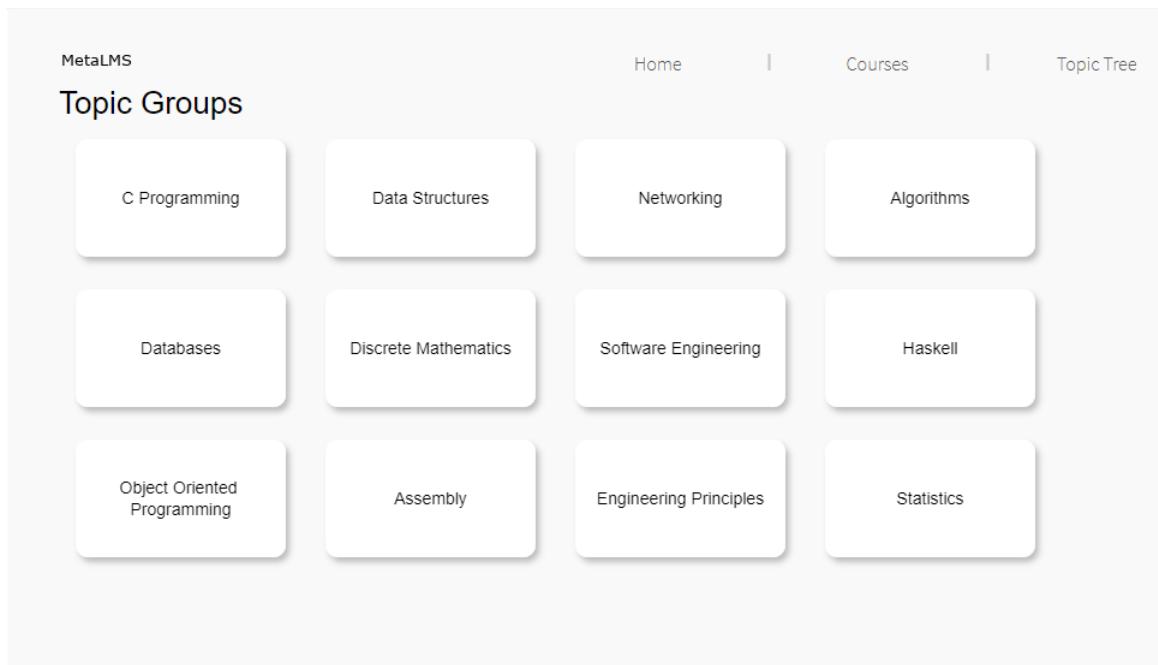


Figure 3.2: Topic Groups UI

A screenshot of the C Programming topic group within the MetaLMS system. The page title is 'C Programming', and the breadcrumb navigation shows 'Topic Groups > C Programming'. The content area displays a 2x4 grid of sub-topics: Pointers, Conditionals, Loops, Structs; Strings, Debugging, Memory.

Figure 3.3: Topic Group - C Programming Example

Inside a topic group, the full list of topics inside this group is listed. Prerequisites are NOT shown here.

All resources for this topic are listed, and prerequisites are also listed on this page. This view is modelled after the UNSW handbook, where course prerequisites are listed as links. This topic can then be imported into a course with the import button at the bottom.

Pointers
Topic Groups > C Programming > Pointers

Intro to Pointers	Download PDF
Null Pointers	Download PDF
Pointer Arithmetic	Download PDF
Pointer of pointers	Download PDF
Returning pointers from functions	Download PDF

Prerequisites

Structs	Go to topic
Loops	Go to topic
Strings	Go to topic
Conditionals	Go to topic
Debugging	Go to topic

Import

Figure 3.4: Topic - Pointers Example

3.1.3 Requirements

The requirements below are draft requirements, and will become more specific throughout the next stages of the thesis.

Each requirement will have a priority: High, Medium or Low. High priority requirements will be completed first, and then medium and low priority requirements respectively.

3.1.4 Functional Requirements

Viewing Topics

1. Users can view groups of topics **Medium**
2. Users can view topics within each group **High**
3. Users can search for a topic **High**
4. Users can search for specific resources **High**
5. Users can view topic prerequisites **Medium**

6. Users can view a graph of topics and their prerequisites **Low**

Adding Topics and resources

1. Users can add a new topic **High**
2. Users can upload course material **High**
3. Users can add a new topic group **Medium**

Deleting Topics

1. Users can delete a topic that they've created **High**
2. Users can delete a topic group that they've created **Medium**
3. Users can remove course material from a topic **High**

Topic Prerequisites

1. Users can add a topic prerequisite **Medium**
2. Users can delete a topic prerequisite **Medium**

Integration

1. Users can import course material by selecting topics for a course **High**
2. Users can remove topics from a course **High**

Exporting

1. Users can export data from topics and course material **Low**

3.1.5 Timeline

This timeline details what is planned over the course of the year to achieve a working topic tree feature in the meta learning management system. Milestones are featured throughout this timeline.

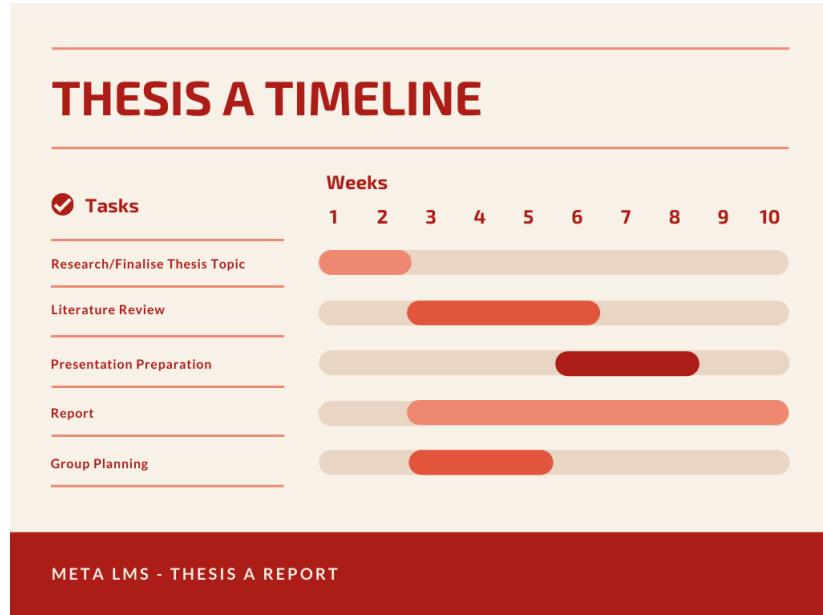


Figure 3.5: Thesis A Timeline

This term, the focus is literature review and planning features for implementation in Thesis B and C.

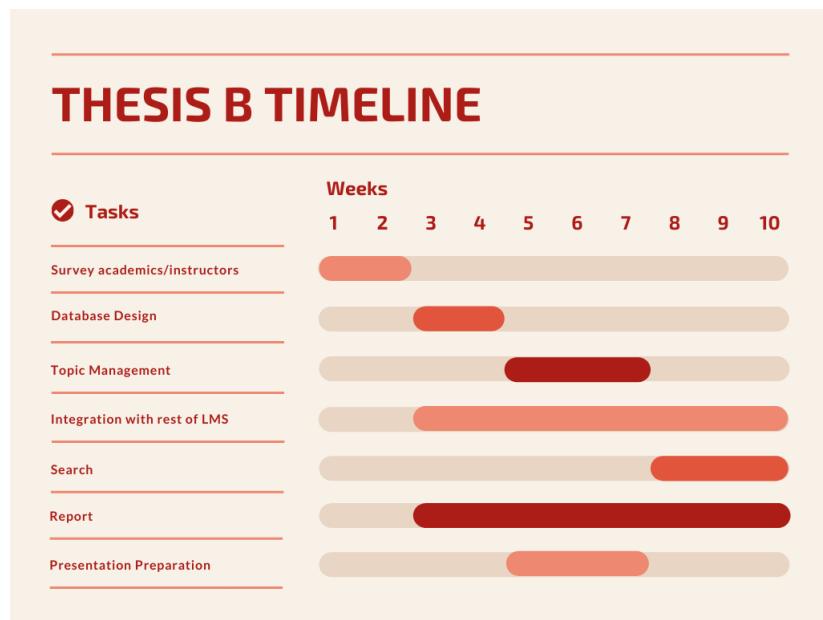


Figure 3.6: Thesis B Timeline

In Thesis B, the focus is to start implementing the topic tree feature, with core features implemented and mostly usable, and a working database design.

Database Design will involve designing a schema for the database to store topics and their dependencies. This will most likely involve a graph relationship between topics. Topic Management involves designing the interface for adding, removing and managing topics and helping develop a backend for this. Integration with the rest of the system involves using the topic tree to import content into a course, and export content into the topic tree, etc. Search involves searching for content, and is not as important as developing the topic management feature.

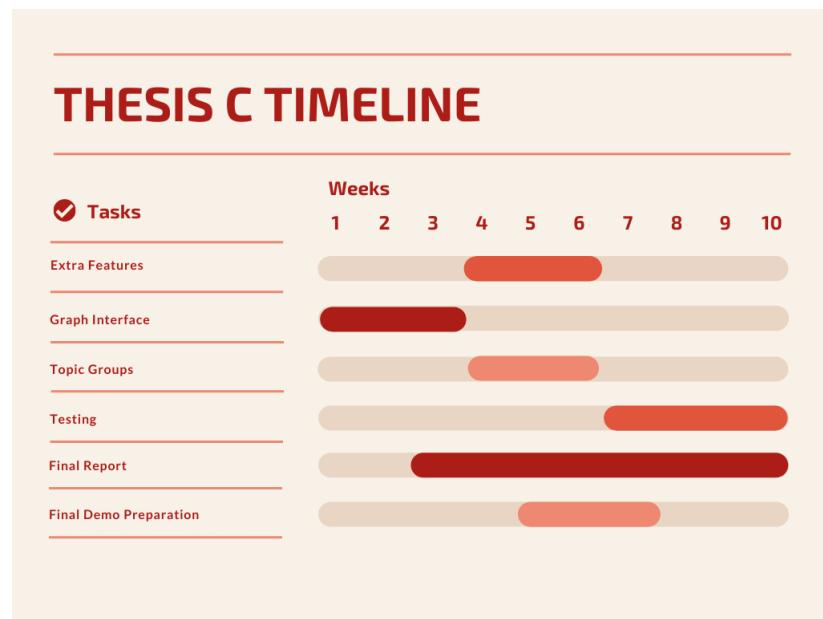


Figure 3.7: Thesis C Timeline

In Thesis C, the main focus is to finalise features, fix any bugs that arise, test the feature and its integration with the rest of the system, implement a graph interface and allow topics to be grouped with each other.

A graph interface would make it much easier to visualise dependencies between topics, but is not a high priority so the plan is to implement this during Term 3 2021.

3.1.6 Milestones

Major milestones for the topic tree feature include:

1. Week 6 Term 2 2021 - Implement a database schema for storing topics and their prerequisites, and set up an interface for topic management.
2. Week 11 Term 2 2021 - Adding and deleting topic prerequisites/dependencies, a search function and proper integration with the rest of the learning management system
3. Week 6 Term 3 2021 - Graph interface to easily view prerequisites between topics and topic groups
4. Week 11 Term 4 2021 - Final Testing and Analysis of the learning management system

3.1.7 Evaluation

In order to evaluate how well the feature has met its requirements and achieved its purpose, a criteria will be proposed. In addition to meeting its requirements, the feature will also be assessed in several areas:

1. Performance - Whether the topic tree feature is fast and responsive
2. Accessibility - Can a wide array of users use the topic tree easily (including people with disabilities, etc.)
3. UI/UX - Is the feature easy to use and attractive
4. Errors - Is the feature bug and error free

3.2 Forums

3.2.1 Overview

Forums are a common feature in learning management systems as it provides a community environment where students can ask questions and discuss content with educators and other students. Many of these built-in forums, however, are very basic and lack sufficient search, sorting and filtering functionality. In many cases, educators are turning to external, third-party forums in order to take advantage of the more advanced features that they offer.

The aim of this feature is to develop a built-in forum that meets students' and educators' needs so that they no longer need to use an external application.

The forum interface consists of the overview page and the individual post pages.

Forum Overview Page

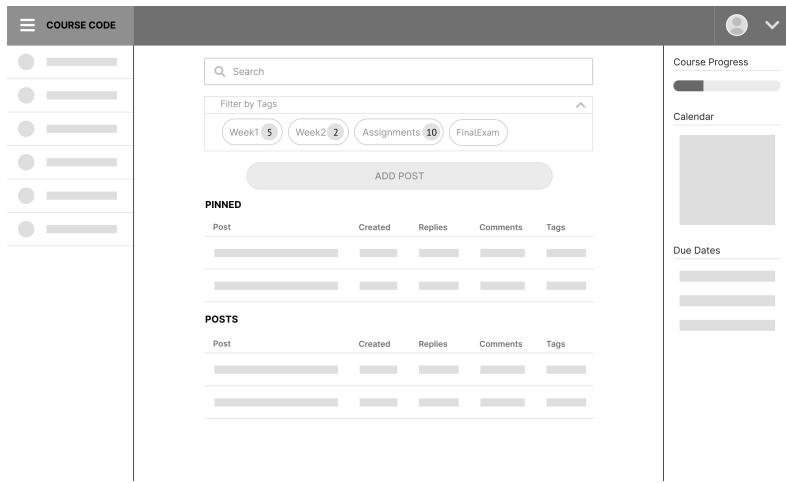


Figure 3.8: Forum overview page for a student

In general, the forum overview page consists of a list of posts, as well as search and filtering mechanisms. The collapsible filter menu allows users to filter the forum post based on pre-defined tags. Forum posts are ordered such that the list of pinned posts are at the top, followed by the remaining posts in chronological order. Each row in the table includes the post title, date created, number of replies, number of comments and the associated tags.

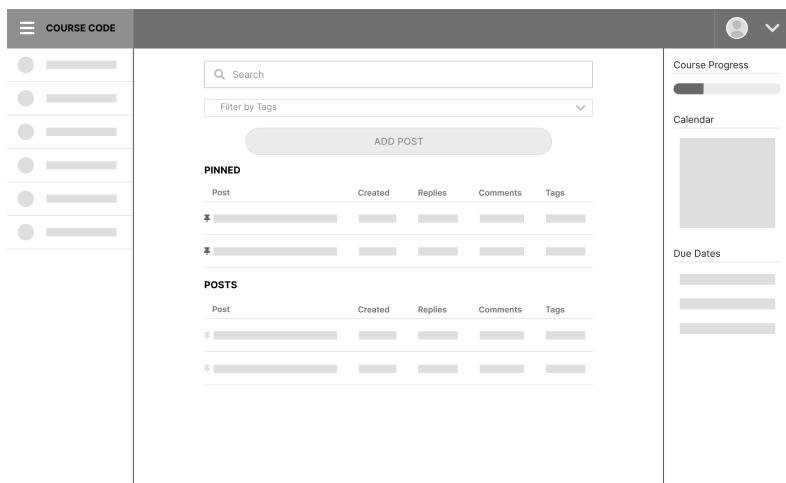


Figure 3.9: Forum overview page for an admin

Admins have an additional button that allows them to pin and unpin posts.

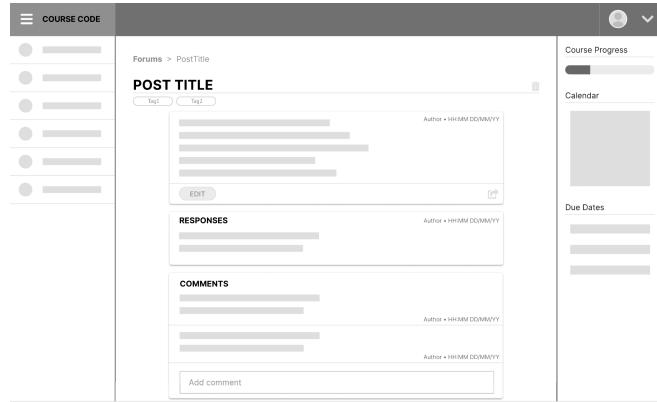


Figure 3.10: Forum post page for a student

Post Page

Each forum post has its own post page which contains the post details, responses and comments. Students are able to edit their posts from the post page if required. They can easily view the instructor's response, if any, in the responses section. The comments section allows the author to view and leave any additional comments. It also gives other students a place to write a response or ask questions based on that forum post.

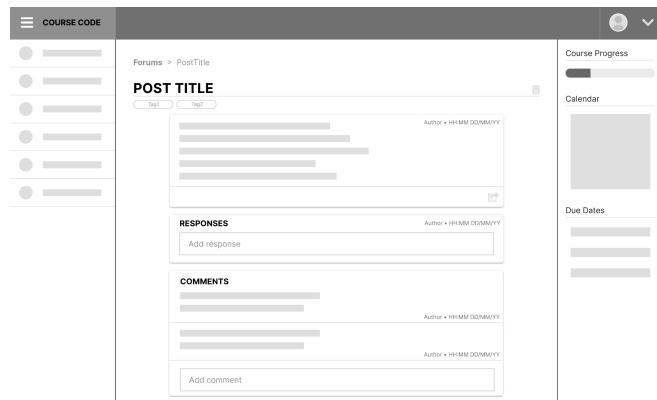


Figure 3.11: Forum post page for an admin

If a forum post is unanswered, an admin is able to leave a response. The current idea is to restrict this so that only admins can write responses to ensure that all responses are verified. It also ensures that forum posts aren't left unanswered if a student accidentally leaves a follow-up question in the responses area, instead of the comments section. This method of implementation may be reassessed if time allows.

3.2.2 Requirements

The following features are prioritised using the MoSCoW method which assists in identifying the order in which to implement the requirements. It contains the following categories

- **Must have** - vital features that are critical to the basic functionality of a project
- **Should have** - important features that aren't critical but add to the basic functionality of a project
- **Could have** - desired features that aren't necessary to the overall project but can provide a better user experience
- **Won't have** - low-priority features that likely won't be able to be completed in the given time-frame

Functional Requirements

1. Users can view a list of forum posts (Must have)
2. Users can make posts to the forum (Must have)
3. Users can reply to forum posts (Must have)
4. Users can embed images and links in their posts (Should have)
5. Users can share forum posts (Should have)
6. Users can categorise forum posts (Should have)
7. Users can search forum posts (Should have)
8. Users can clearly see which posts have been read and actioned (Could have)
9. Admins can pin important forum posts (Should have)
10. Admins can link/embed materials to forum posts (Could have)
11. Admins can curate forum questions into collections (Won't have)

3.2.3 Backend Assumptions

Since the backend for the Meta LMS is being built out independently to the individual features, the following contains a description of the ideal backend design for the forum component.

Database

The main database table required for this component is one to store all the forum posts. This would include all the post details including author, title, description, date created and tags. It would also need to have a way to keep track of the replies and comments left on the forum post. This could either be done by having separate tables for replies and comments, or by storing a list of replies and comments within each forum post entry. For each of the forum posts, the author would need to be linked to a user in the user table of the database.

API

In terms of the backend API, the following functionality would be required in order to store, retrieve and update the appropriate data from the database.

1. Retrieve a list of all forum posts
2. Retrieve a list of the pinned posts
3. Retrieve a list of posts related to search term
4. Retrieve a list of posts related to filter
5. Retrieve individual post details for post page (including comments and responses)
6. Store new post details
7. Store post responses
8. Store post comments
9. Update post details when author edits post
10. Update response when author edits response
11. Update pinned post list when admin pins/unpins posts

3.2.4 Evaluation

In order to ensure that the needs of students and educators are met for this forum component, the following evaluation methods will be used.

Functional Requirements

The first form of analysis that will be used to evaluate the forum component is to compare the completed product with the functional requirements defined during Thesis A. This will ensure that all of the essential requirements are covered. If there is enough time, it will also highlight any lower priority requirements that may be added to the component.

Usability Testing

To test the usability of the system, usability tests will be run with various students and educators. A usability test typically consists of having participants who have never used the system before run through a scenario. In this case, it may be appropriate to test that a student is able to create and view a post. Similarly, an admin might be tested to see if they can easily find an unanswered post and respond to it. Different measures like number of errors and time it takes to complete a task can be used to quantitatively assess the usability of the system. As a result, usability testing will help to identify any areas of the system that may be confusing or difficult to use.

Google Lighthouse

In order to test the performance and accessibility of the forum component, Google Lighthouse will be used. This runs tests on a site and produces a score out of 100 for each category tested. In this case, a score will be produced for the performance of the site and the accessibility. In addition to the score, Google Lighthouse also produces a report that gives more detail on ways the site could be improved to produce a higher score. These scores can be used to ensure that the forum component is accessible to all types of users. It can also be used to compare this forum with forums from other LMS.

3.2.5 Project Timeline

Below is a rough guideline of some milestones that should be met throughout the year in order to complete this feature on time.

Thesis A

The focus of Thesis A, Term 1 2021, is solely based around researching current learning management systems to get a good understanding of the desired features required for a meta LMS. It also includes more in-depth research of forums and planning out what this forum component should consist of.

THESIS A

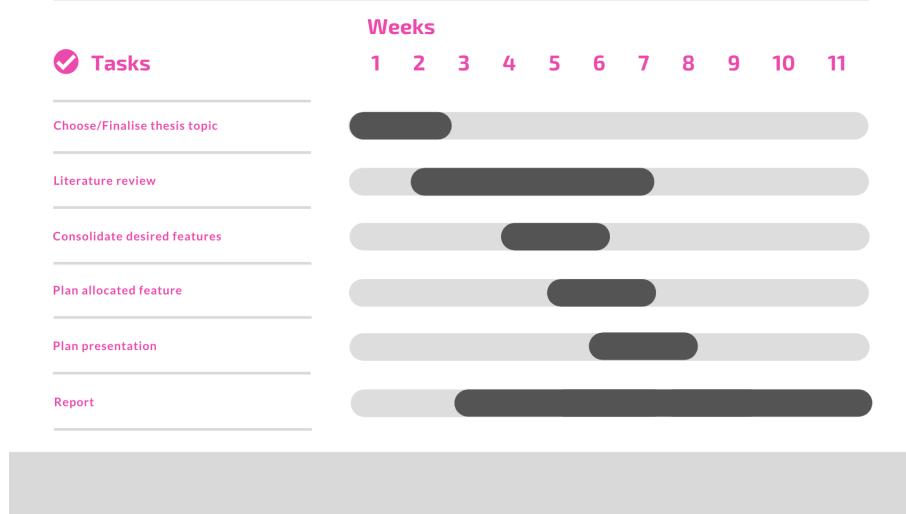


Figure 3.12: Thesis A Timeline

Thesis B

Thesis B, Term 2 2021, will hopefully consist of most of the implementation work for forums. It will start with some market research to get a good understanding of the wants and needs to students and educators. Analysis of these results will help to finalise the features and designs of the forum component.

Once the frontend and backend designs have been finalised, the forum overview page and post page will be implemented. This includes viewing posts, making posts and replying to posts.

The next focus is the searching and filtering functionalities. This includes searching the forum and filtering based on pre-defined tags.

Finally, in Thesis B, the ability to pin and share posts will be added. This should conclude the implementation of all the main features of the forum component.

Thesis C

Thesis C, Term 3 2021, is mainly centred around finalising the implementation and evaluating the solution. The first few weeks will consist of completing any unfinished features, cleaning up the UI and debugging any problems. If there is time, extra features

THESIS B

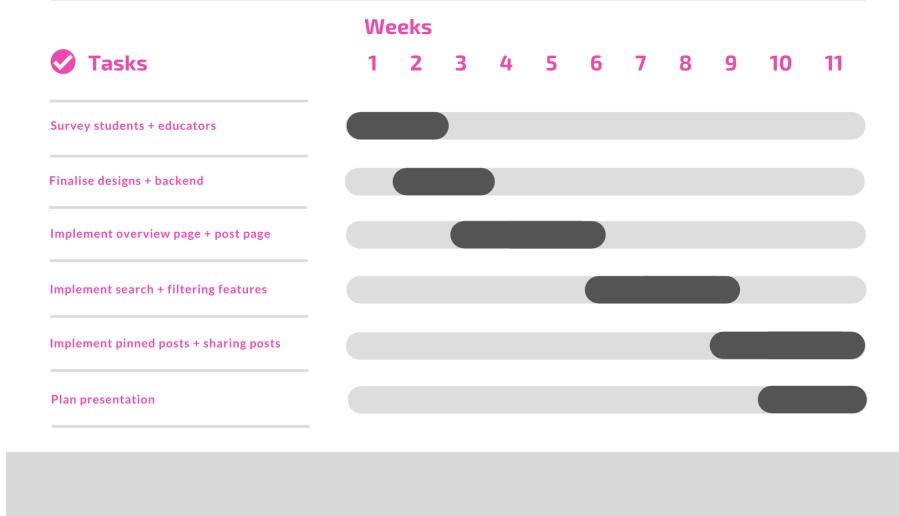


Figure 3.13: Thesis B Timeline

could be implemented. Time will also be allocated to integrating the forum with the rest of the LMS.

The middle weeks of the term will focus on analysis and usability testing. This will help see if the users' needs have been met by the implemented solution.

The final few weeks will consist of any final improvements and testing. The results from the analysis will help to fix any problems with the forum component.

3.3 Assessments

3.3.1 Overview

The Assessment feature consists of a quiz and poll system where the poll can be used as a supplementary tool for the quiz or to be used as a general poll system. Quizzes are a widely used form of assessing students' knowledge about a particular topic or course. It is beneficial to students as it gives a good indication of their understanding of a topic and whether they need to revise more in a particular area. Likewise, it is beneficial to course lecturers as they'll be able to gauge the students' understanding and whether they should change their teaching methods or revise the content with students to resolve difficulties with concepts. Therefore it is an essential in any learning management system to provide quizzes, which most, if not all, learning management

THESIS C



Figure 3.14: Thesis C Timeline

systems do. However, some existing quiz systems lack in providing an aesthetic user interface and useful feedback for both parties. Therefore, the Assessment feature aims at:

- allowing quiz submissions and grading to be performed on the same platform for convenience
- providing additional useful statistics as feedback to help improve the teachings of a course instructor and the learnings of a student enrolled in the course
- having an appealing, readable interface for both course instructors and students
- providing a poll system for general poll use or as a means for students to communicate what topic they'd like the course lecturer to revise

The core features for quizzes include being able to:

- create/remove/edit a question
- answer/edit an answer to a question
- choose between three types of questions to create:
 - multiple choice - select one answer only
 - short answer

- checkboxes - select none, one or multiple answers
- auto-save and manual save answers during quiz attempt
- timer that runs during quiz attempt

Question 1

Question type:

Question:

Answers:

Add to question bank

Figure 3.15: Quiz creation (Instructor's view)

Question 1

Answers:

Figure 3.16: Quiz usage (Student's view)

The core features for polls include being able to:

- create/remove a poll
- create/remove/edit a poll option

- select poll options/s to vote and change your vote
- choose between two types of polls to create:
 - restricted - can only vote for 1 option
 - open - can vote for multiple options

3.3.2 Stakeholders

- Lecturer/course instructor
- Admins
- Students

3.3.3 Functional Requirements

The list of functional requirements' priority level will be determined using the MoSCoW method to clearly outline what needs to be implemented throughout the thesis and what the minimum viable product should have.

The MoSCoW method splits requirements based on 4 categories:

- [Must Have]
- [Should Have]
- [Could Have]
- [Won't Have]

A requirements survey was conducted and completed by me and other Thesis A students to gauge what was the priorities for each requirement. The results for the Assessment feature from the survey and my own ideas were combined to form the following functional requirements:

Quiz creation

1. Course lecturers can create a quiz [Must Have]
2. Course lecturers can add questions to a quiz [Must Have]
3. Course lecturers can create different types of quiz questions (multiple-choice, short answer, checkboxes) [Must Have]
4. Course lecturers can set up a timer or due date for a quiz [Must Have]

5. Course lecturers can add "drag and drop" type questions [Could Have]
6. Course lecturers can add "connect the pairs" type questions [Could Have]
7. Course lecturers can add media (audio or video) into a question as the entire question or as a supplementary material to the question [Could Have]
8. Course lecturers can create a question bank [Should Have]
9. Course lecturers can add a question to a question bank [Should Have]
10. Course lecturers can import a question from a question bank [Should Have]

Quiz modification/removal

1. Course lecturers can edit/remove a quiz [Must Have]
2. Course lecturers can edit/remove a question [Must Have]
3. Course lecturers can remove a question bank [Should Have]
4. Course lecturers can remove a question from a question bank [Should Have]

Quiz usage

1. Students can answer a quiz question [Must Have]
2. Students can edit any of their answers during the quiz [Must Have]
3. Students can submit a quiz attempt [Must Have]
4. Students can manually save their progress at any time [Should Have]

Poll creation

1. Course lecturers can create different types of polls (can vote either 1 option only or multiple options) [Must Have]
2. Course lecturers can add an option to a poll [Must Have]

Poll modification/closing

1. Course lecturers can edit/remove options [Must Have]
2. Course lecturers can close polls [Must Have]
3. Course lecturers can set a poll closing date [Should Have]
4. Course lecturers can close a poll but make results viewable [Could Have]

Poll usage

1. Students can vote for one or multiple options (depending on poll type) [Must Have]
2. Students can add an option (if course lecturer enables the option for it) [Could Have]

Viewing quiz results and feedback

1. Students can view results against their answers [Should Have]
2. Students, course lecturers and admins can see how many students selected each answer for a question [Should Have]
3. Students can view topic or lecture the question derives from [Should Have]
4. Students can view an explanation of the correct answer if the question is answered incorrectly [Could Have]

Re-usability

1. Course lecturers can add a question to a general question bank [Should Have]
2. Course lecturers can import/export quizzes [Should Have]
3. Course lecturers can import/export questions [Should Have]

3.3.4 Non-functional Requirements

The non-functional requirements used is heavily inspired by the Jakob Nielsen's 10 usability heuristics that's used to evaluate the usability of user interfaces.

1. **Efficiency** - users are able to perform tasks without taking too many steps
2. **Learnability** - new and returning users are able to quickly learn how to use and interact with the feature on the go
3. **Performance** - operations within the feature are completed within a timely manner, resulting in a smooth process
4. **Consistency** - components, colour themes and formats are consistent across pages
5. **Aesthetic and minimalist design** - the user interface is appealing and simple while providing the expected features
6. **Re-usability** - users are able to re-use components and content they've previously made

3.4 Required Training/Upskilling

Training across multiple areas will be required in order to develop the Learning Management System:

- Create technical diagrams using Miro (diagramming tool)
- Develop React skills
- Explore and experiment with graph visualisation libraries (to display statistics)
- Research SCORM and other standards used in learning management systems

3.4.1 Back-end Assumptions

The following back-end assumptions contains details of the final backend technologies used, as discussed with other Thesis A students, and general assumption of the database which data will be retrieved from via API calls.

Technologies Used

The technologies that will be used are:

- Back-end runtime environment: Node.js
- Web application framework: Express.js
- Database management system: PostgreSQL

Database

The use of a relational database will be ideal as it'll be more organised and querying for particular data related to each other will be easier during the implementation of the Assessment feature.

Potential tables involved would include a Question, Poll, Quiz and User table.

3.4.2 Front-end Assumptions

The following front-end assumptions contains details of the final front-end technologies used, as discussed with other Thesis A students, and a breakdown of the Assessment feature to give a better insight of the implementation process.

Technologies Used

The technologies that will be used are:

- Web application framework: React
- Component library for React: Material UI

Feature Breakdown

Since the React frameworks structures content into re-usable, isolated components, the Assessment feature will need to be broken down into such components. Below are the main components that will likely be implemented during Term 2:

- Question
- Quiz
- Poll Option
- Poll
- Timer
- Question Bank
- Quiz Results
- Answer Explanation

3.5 Timeline

3.5.1 Term 1 (Thesis A)

Term 1 will mainly focus on literature research, outlining the main and potential features required to achieve the goals of this thesis.

3.5.2 Term 2 (Thesis B)

Term 2 will mainly focus on implementation of the core features and ensuring a good design overall before adding any extra features.



Figure 3.17: Thesis A Timeline

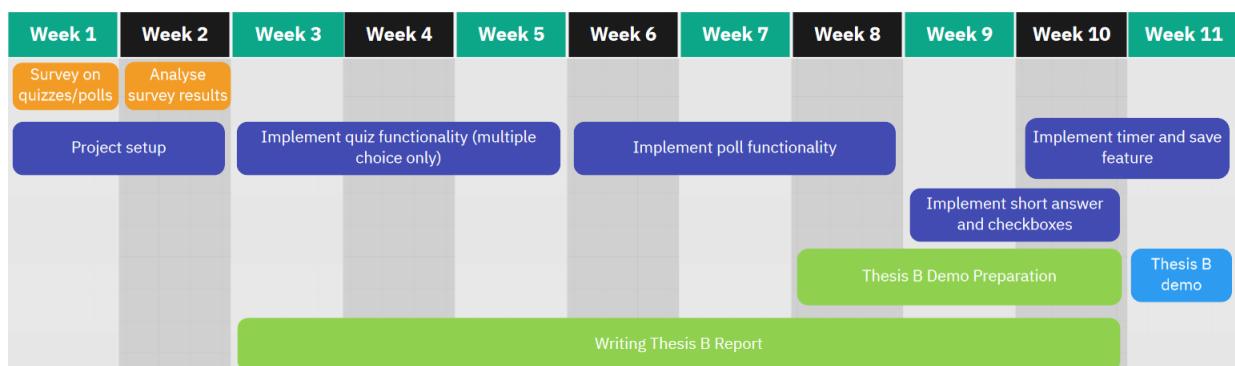


Figure 3.18: Thesis B Timeline

3.5.3 Term 3 (Thesis C)

Term 3 will focus on improving the system based on feedback, implementing additional features and doing a final testing on the system to ensure it satisfies requirements.

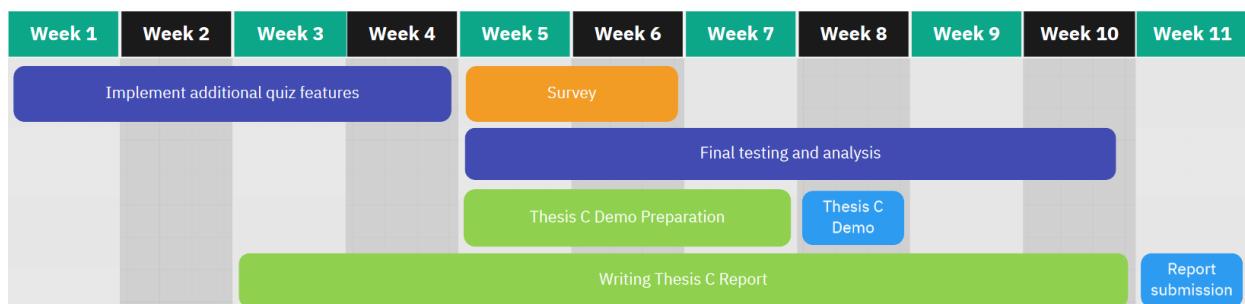


Figure 3.19: Thesis C Timeline

3.5.4 Evaluation

The system will be evaluated based on:

- **Functional requirements** - what was completed, with the satisfactory being all "Must Haves" and some "Should Haves"
- **Non-functional requirements** - whether it satisfies the goals of the feature and performs operations in a timely manner
- **Usability tests** - if users can navigate through the feature and be able to complete operations within the feature

Chapter 4

Conclusion

A thesis requirements/template document has been created. This serves the dual purposes of giving students specific requirements to their theses — both style and content related — while providing a typical thesis structure in a L^AT_EX template.

4.1 Future Work

Extract the requirements from the template in order to have very concise requirements.