CS3216 Final Project Report

Group 1 - Coursemology

Jerome Cheng (A0080860H)
Chen Minqi (A0099314Y)
Joel Low (A0097630B)
Fazli Sapuan (A0096836M)

Application Links

The application is deployed at the following locations:

- Coursemology.org
- Demo Sandbox
    - Username: `jfdi@academy.com`
    - Password: `SuperSecretpaSS`

Application Description

Coursemology is an open source gamification platform which makes academic courses more fun and engaging, adding elements found in typical role-playing games to them. Students are given "Missions" instead of assignments, and are rewarded with experience points and achievements for completing them. In addition, participation in discussions with one another is incentivised through rewarding them with additional experience points for insightful or useful posts. Students can also see the progress (or "level") of their peers, which further motivates them to keep up with their coursework.

Though the gamification elements were complete and working quite well, Coursemology was lacking several features which would be a boon for both teachers and students alike. Though the system allowed assignments to be released and submitted through it, there was no way for teachers to dispense other course materials such as lecture slides. Coursemology also didn't have an in-built discussion forum; the lack of these components meant that at least two systems had to be used in parallel: Coursemology, and some other service.

Our project focused on adding three components to Coursemology to make up for this deficiency: a discussion forum, lesson plan, and materials bin (or workbin). With these additions, Coursemology no longer requires the use of another service, and can serve as a "self-sufficient" platform for courses that choose to use it.

Similar Alternatives

The closest application to Coursemology which is currently in use is the JFDI Academy[1], used by CS1101S Programming Methodology in the School of Computing. The JFDI Academy can be seen as the forerunner of Coursemology, having perhaps even more gamification elements than Coursemology does. However, it was designed specifically for the module, and is unable to be used by other courses.

On the other hand, Coursemology was designed from the ground up to be a generalised online platform for course gamification. It has thus lost some of the specificity and automation that JFDI has in exchange for this goal.

A platform which also aims to assist in pedagogy is the IVLE system used within NUS. IVLE has been in service for a long time, making it a relatively stable platform which both educators and students alike know how to use. However, IVLE has many quirks in its behaviour and while *useable* is not pleasant when used for extended periods of time. This is because of the way it scales with increased content -- something which becomes apparent near the end of the semester. Before our project, both Asst. Prof Ben Leong and Dr. Steven Halim both used Coursemology in concert with IVLE, relying on it where Coursemology was lacking.

Blackboard Learn[2] and Moodle[3] are platforms similar to IVLE, and can also be used similarly. Still, they lack the gamification elements that characterise Coursemology, and would thus result in the same problem faced with IVLE: users having to rely on both Coursemology and another system.

Differentiating Factors

Coursemology

The largest differentiating factor that Coursemology has vis-a-vis the competing products which we have mentioned above is that Coursemology has gamification built right within the platform. Gamification has shown to increase the performance of students - both at the top end of the spectrum, as well as students who might be struggling with their coursework. This is demonstrated through a project at Sembawang Secondary School in which students using Coursemology performed more than 85% better than their peers who were not.

Coursemology also has features which help Computer Science courses specifically: Missions can have code submissions with inline code commenting for direct feedback between educators and students, reducing the attrition in highlighting the deficiencies in students' code.

Another significant difference between Coursemology and IVLE as well as Blackboard is that it is open-source. This has significant benefits when it comes to the long-term sustainability and viability

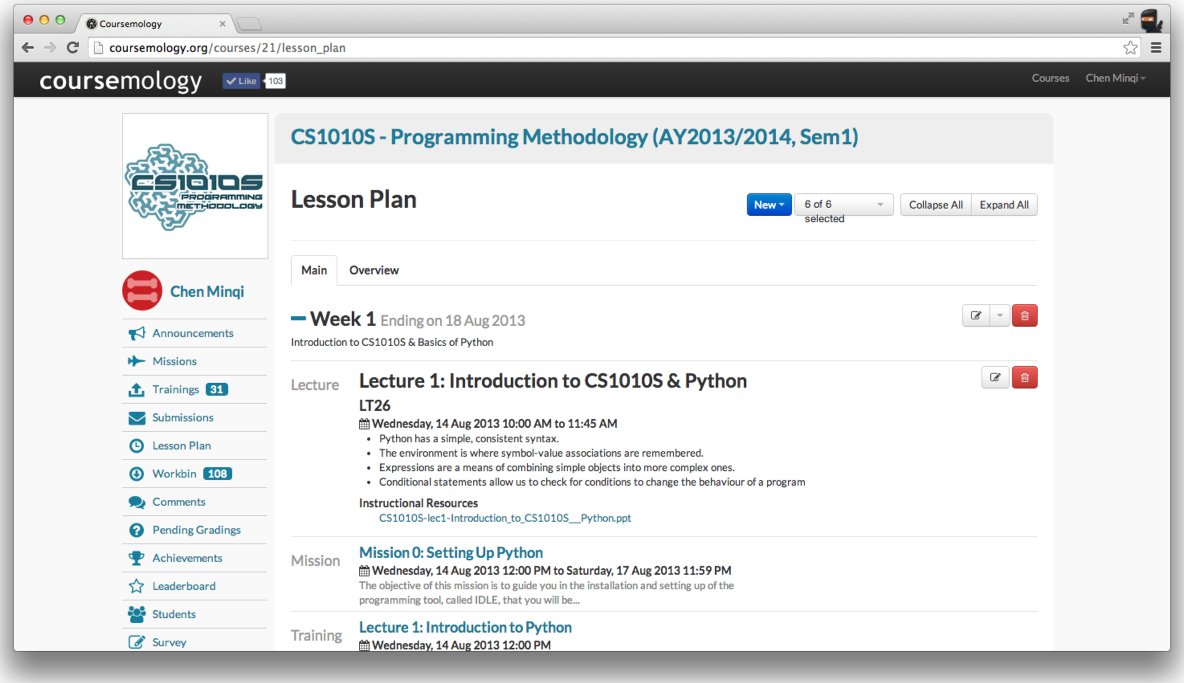
[1] http://jedi.ddns.comp.nus.edu.sg
[2] www.blackboard.com/Platforms/Learn/Overview.aspx
[3] www.moodle.com

of the product; commercial products might see vendor support being abandoned, or vendors might prevent customers from moving to a better platform should one arise.

Lesson Plan

The main benefit of our Lesson Plan implementation is the ease with which teaching staff can work with it. Our interface allows events to be created in much less time than before: to replicate the CS1010S Lesson Plan on Coursemology took approximately half an hour, while doing the same on IVLE takes several hours (according to Asst. Prof. Ben Leong.) There are also much fewer mouse clicks required to do so: creating a thirteen-week Lesson Plan on IVLE takes eighty-four clicks, while only fifty-four are needed on Coursemology[4].

From a students' perspective, the Coursemology Lesson Plan is a great convenience. Assignments created through the Mission or Training systems automatically appear on the Lesson Plan, and the entire plan can be filtered by type (i.e. students can show or hide lectures, tutorials, and assignments individually.) Each individual period of time on the Lesson Plan can also be collapsed or expanded individually, allowing students to view only information which is currently relevant to them; this is unlike other systems which always display all events, resulting in information overload.



CS1010S - Programming Methodology (AY2013/2014, Sem1)

Lesson Plan

Week 1 Ending on 18 Aug 2013

Introduction to CS1010S & Basics of Python

Lecture

Lecture 1: Introduction to CS1010S & Python

LT26

Wednesday, 14 Aug 2013 10:00 AM to 11:45 AM

- Python has a simple, consistent syntax.
- The environment is where symbol-value associations are remembered.
- Expressions are a means of combining simple objects into more complex ones.
- Conditional statements allow us to check for conditions to change the behaviour of a program

Instructional Resources

CS1010S-lec1-Introduction_to_CS1010S__Python.ppt

Mission

Mission 0: Setting Up Python

Wednesday, 14 Aug 2013 12:00 PM to Saturday, 17 Aug 2013 11:59 PM

The objective of this mission is to guide you in the installation and setting up of the programming tool, called IDLE, that you will be...

Training

Lecture 1: Introduction to Python

Wednesday, 14 Aug 2013 12:00 PM

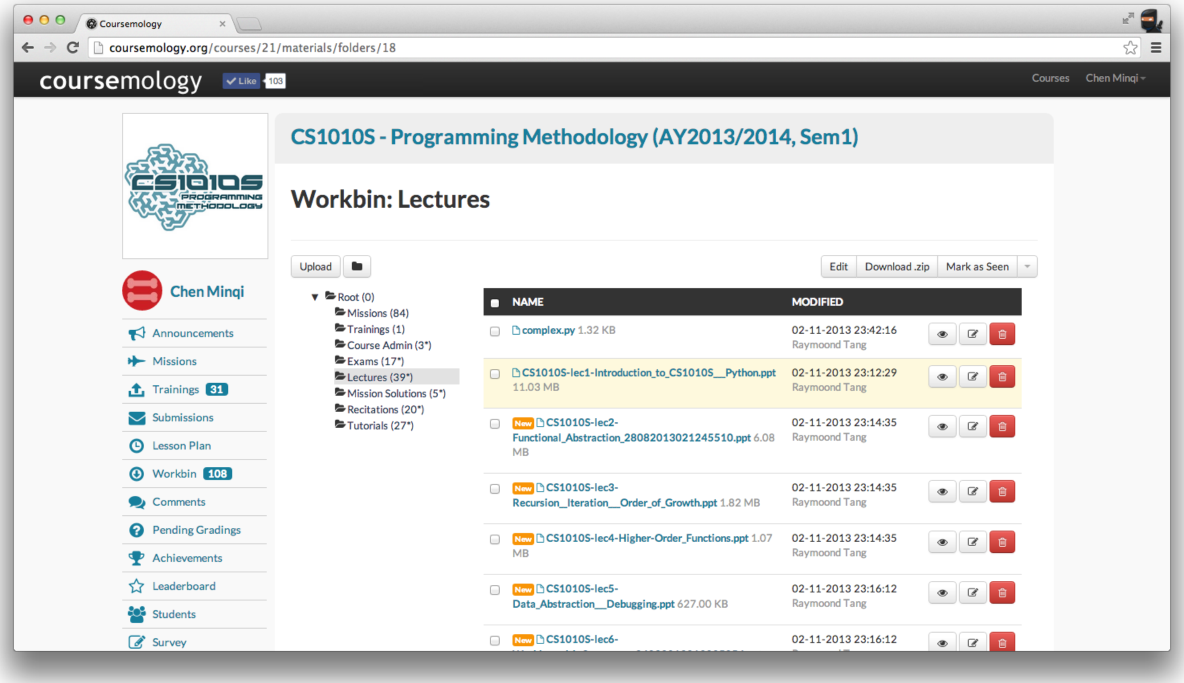
Workbin

The workbin implementation on IVLE is comparatively painless, and there were no major improvements we made in this area. We did, however, work on the usability and discoverability of new content, to ensure that students are never left second-guessing when new content is published:

[4] Assuming two events per week, and excluding clicks needed to switch between fields on the input form.

a simple indicator appears next to the file in question, and can be dismissed either by downloading the file or specifically marking it as "seen".

There is also a workflow improvement when uploading large numbers of files: IVLE has a hard limit of uploading no more than four files at a time, while we allow the upload of as many files as required in one pass.



CS1010S - Programming Methodology (AY2013/2014, Sem1)

Workbin: Lectures

□	NAME	MODIFIED
□	complex.py 1.32 KB	02-11-2013 23:42:16 Raymoond Tang
□	CS1010S-lec1-Introduction_to_CS1010S__Python.ppt 11.03 MB	02-11-2013 23:12:29 Raymoond Tang
□	New CS1010S-lec2-Functional_Abstraction_28082013021245510.ppt 6.08 MB	02-11-2013 23:14:35 Raymoond Tang
□	New CS1010S-lec3-Recursion__Iteration__Order_of_Growth.ppt 1.82 MB	02-11-2013 23:14:35 Raymoond Tang
□	New CS1010S-lec4-Higher-Order_Functions.ppt 1.07 MB	02-11-2013 23:14:35 Raymoond Tang
□	New CS1010S-lec5-Data_Abstraction__Debugging.ppt 627.00 KB	02-11-2013 23:16:12 Raymoond Tang
□	New CS1010S-lec6-	02-11-2013 23:16:12

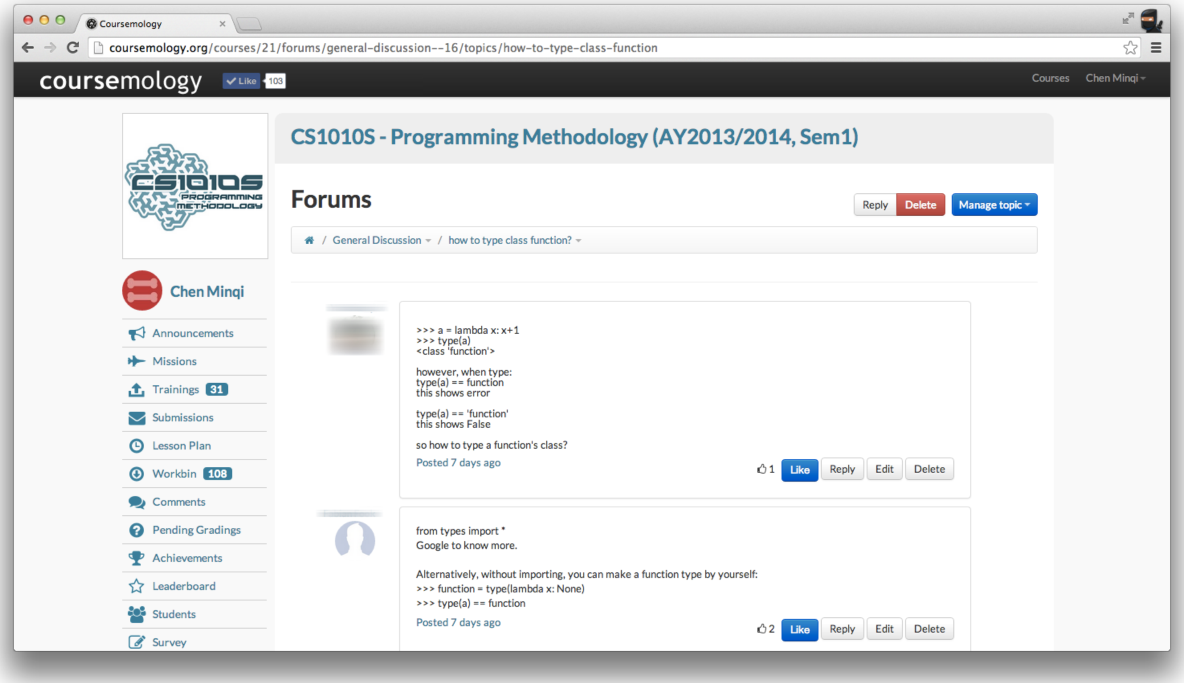
Discussion Forum

Having to host discussion forum on a separate platform is in itself an imperfection. Without a built-in portal for discussion, students and teaching staff are forced to have their conversation elsewhere, disjoint from the rest of the teaching activity.

IVLE's forum tended not to scale in large courses, especially when forum participation is a course requirement. Students are required to scroll through long lists, made up of a mixture of both read and unread items due to the way IVLE groups and orders posts. Though there is a function to collapse read posts, the post listing is still cluttered and the users would have to manually scroll down the list to look for unread posts. Using this function also results in the posts losing their original context. This is a problem especially since most students tend to not quote the post they are replying to, resulting in needing to switch back and forth between posts and the unread posts view. This is disruptive and requires the reader to reconstruct their train of thought.

Furthermore, the integration of a forum into Coursemology itself greatly eases the burden of teaching staff who need to award experience points for discussion participation. Prior to this, tutors would need to manually retrieve (or scrape) every post made on the course's forum, and tally the number of posts made by each student. Our forum implementation now automates this task for them, counting the amount of posts made and suggesting a corresponding amount of experience points.

In addition, our implementation of “likes” on forum posts helps alleviate the subjectivity associated with “grading” forum posts. Prior to this system, course coordinators would assign a tutor to review all posts in the forum and award experience to posts deemed insightful or helpful. With large enrolments, the review process becomes very subjective and includes an element of chance in grading due to the varying stress and patience levels of the tutor. By allowing other forum users to “like” posts, the burden of judging quality is shifted away from a single person and indirectly spread out across the entire student cohort instead (without this being explicit!) - the same system mentioned above automatically takes the amount of “likes” gained into consideration when suggesting the amount of experience points to be awarded to each students[5].



CS1010S - Programming Methodology (AY2013/2014, Sem1)

Forums

```
>>> a = lambda x: x+1
>>> type(a)
<class 'function'>

however, when type:
type(a) == function
this shows error

type(a) == 'function'
this shows False

so how to type a function's class?
```

Posted 7 days ago

1 Like Reply Edit Delete

```
from types import *
Google to know more.

Alternatively, without importing, you can make a function type by yourself:
>>> function = type(lambda x: None)
>>> type(a) == function
```

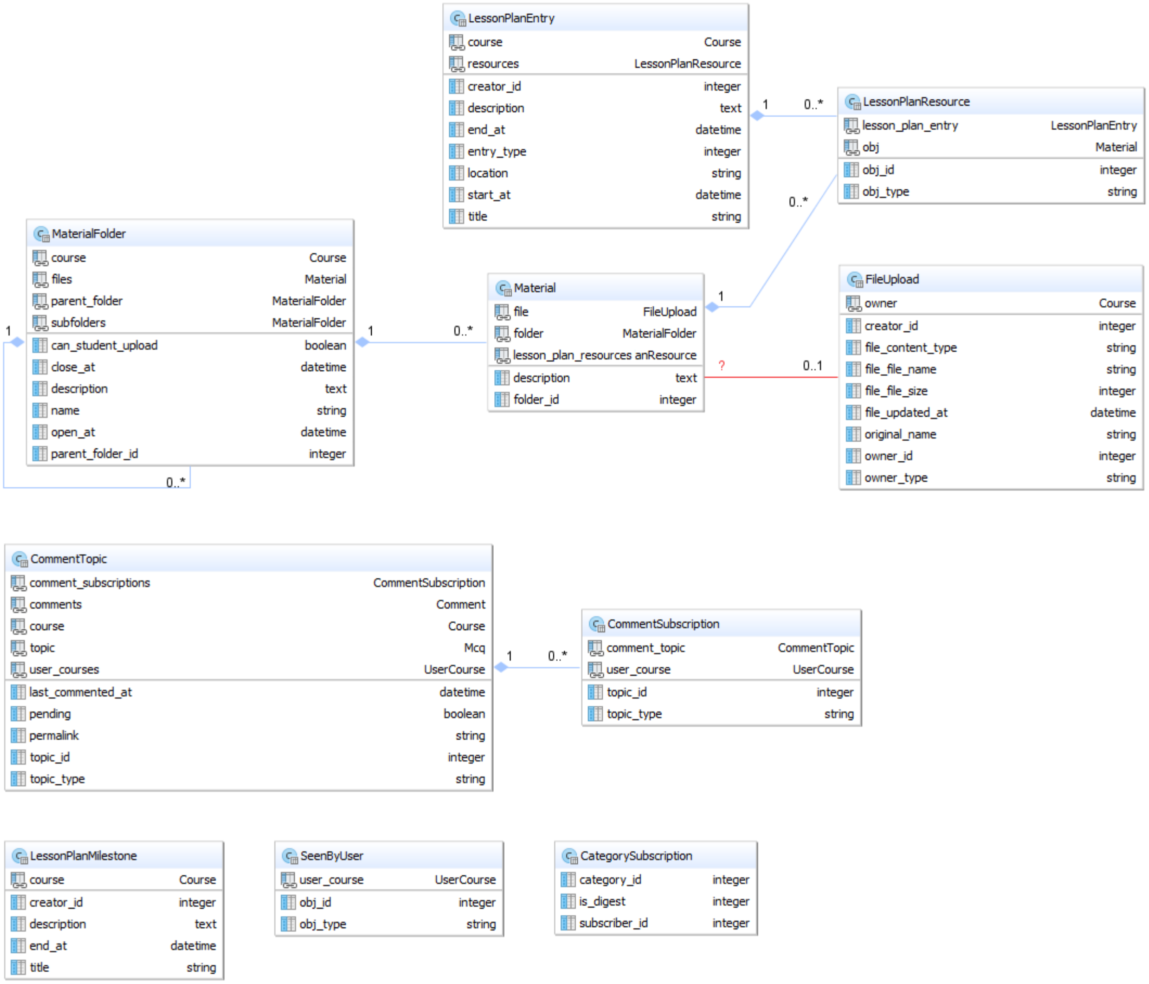
Posted 7 days ago

2 Like Reply Edit Delete

[5] To prevent students from gaming this system, teaching staff are also given the opportunity to easily review each student’s posts, and manually adjust the amount of experience each student is given. The suggested experience amount made by the system is just that: a suggestion.

Application Design

Database Schema



LessonPlanEntry	
course	Course
resources	LessonPlanResource
creator_id	integer
description	text
end_at	datetime
entry_type	integer
location	string
start_at	datetime
title	string

LessonPlanResource	
lesson_plan_entry	LessonPlanEntry
obj	Material
obj_id	integer
obj_type	string

MaterialFolder	
course	Course
files	Material
parent_folder	MaterialFolder
subfolders	MaterialFolder
can_student_upload	boolean
close_at	datetime
description	text
name	string
open_at	datetime
parent_folder_id	integer

Material	
file	FileUpload
folder	MaterialFolder
lesson_plan_resources anResource	
description	text
folder_id	integer

FileUpload	
owner	Course
creator_id	integer
file_content_type	string
file_file_name	string
file_file_size	integer
file_updated_at	datetime
original_name	string
owner_id	integer
owner_type	string

CommentTopic	
comment_subscriptions	CommentSubscription
comments	Comment
course	Course
topic	Mcq
user_courses	UserCourse
last_commented_at	datetime
pending	boolean
permalink	string
topic_id	integer
topic_type	string

CommentSubscription	
comment_topic	CommentTopic
user_course	UserCourse
topic_id	integer
topic_type	string

LessonPlanMilestone	
course	Course
creator_id	integer
description	text
end_at	datetime
title	string

SeenByUser	
user_course	UserCourse
obj_id	integer
obj_type	string

CategorySubscription	
category_id	integer
is_digest	integer
subscriber_id	integer

There are additional tables that are used which are not covered in this diagram; the ones shown here are those which are directly used within our code. Notably, the Forum's basic tables (including forum sections, topics, posts and their relevant metadata) are not generated because that functionality is provided by a plugin ("Gem") and as such does not appear in our ERD.

For the Workbin and Lesson Plan, there are some record tables missing as well because there needs to be some virtual objects (like missions and trainings) which do not physically exist in the database as a downloadable file or as a lesson plan event.

Code Structure

Coursemology is built on Ruby on Rails 3, and so the code follows a typical MVC organisation. We have defined models for the tables shown in the ERD above, with controllers handling individual requests and views as presentation. One thing which caught us unaware was that the assets used by the application was stored separately and needed to be precompiled into one production source file. This led to a 1.2MB JavaScript file in production.

The forum part of Coursemology is slightly different. It is also MVC, but the model and the controller parts are provided by the Forem gem. We supplied our own views and we extended the models and controllers with our desired functionality using decorators.

Project Timeline Review

Our original project plan (specified in the initial proposal) was no longer relevant after we shifted goals -- we had initially intended to port the JFDI Academy (mentioned above) to Coursemology, but decided augmenting Coursemology was a more worthwhile objective after a discussion with Prof. Ben. Even after this pivot, progress was still not as quick as desired due to several technical issues.

The Workbin component was the first to reach the working prototype stage, during the third week of the project, followed one week later by the Lesson Plan. These two components were deployed to production in week four, and data migrated from IVLE.

The Forum took slightly longer due to unexpected difficulties when attempting to integrate with existing features in Coursemology, and was deployed in week six of the project - as with the Workbin and Lesson Plan, the existing data on IVLE was also migrated to Coursemology as part of the deployment.

Insights Gained

Live data should be handled with extreme care. While this sounds like a trivially obvious point, it was only through a (rather stupid) mistake that a hard lesson was taught. When testing the forum with a live database dump, we neglected to turn off the mailer in the background. However, there were delayed mailing jobs queued in the database and as a result, duplicate emails were sent to CS1010S students. While the actual disturbance caused by this oversight was minimal, the potential damage could have been far worse.

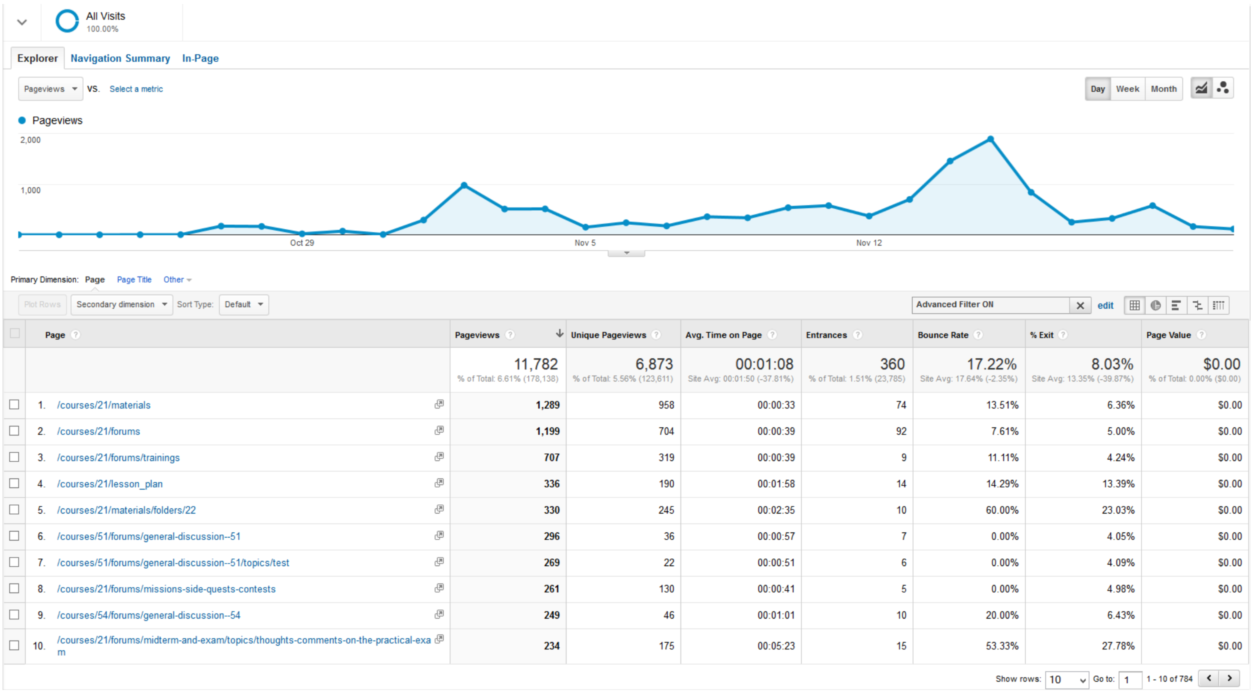
More thoughts should be put into scalability issue in production. For some of us, Coursemology's user base is the largest that we have interacted with. While we have been taught well to understand how scaling issues may occur, this wasn't initially a large concern during the development of the forum. As a direct result, extremely inefficient database queries were pushed into production and issued multiple times concurrently. This drained all server CPU time and eventually brought down the production server for around two hours. This issue was never noticed during development since the queries were fine if they were triggered sequentially.

Understanding the needs of clients is never an easy experience. Clients always assume that you deeply understand their requirements and will produce a product that is exactly what is in their mind, even without them explicitly specifying what that product is. Often we found ourselves implementing an idea only to find out later that the client had a vastly different vision of how that component should work. This would have not been as great a problem if not for the fact that we took a long time to develop our first forum prototype, preventing us from being able to iterate on it sooner.

User Analytics

The Coursemology platform is currently used in two courses within NUS, CS1010S Programming Methodology, and CS2010 Data Structures and Algorithms II, as well as at Sembawang Secondary School. The features we have added are available platform-wide, and so the students and teaching staff of all courses on Coursemology have access to them.

The chart below demonstrates the use of the features which we integrated with Coursemology since the first deployment. There is an upward trend, with a spike of usage near the time when CS1010S held its practical examination, followed by a tapering of traffic during Reading Week.



	Page	Pageviews	Unique Pageviews	Avg. Time on Page	Entrances	Bounce Rate	% Exit	Page Value
		11,782 % of Total: 6.61% (178,138)	6,873 % of Total: 5.56% (123,611)	00:01:08 Site Avg: 00:01:50 (-37.81%)	360 % of Total: 1.51% (23,785)	17.22% Site Avg: 17.64% (-2.35%)	8.03% Site Avg: 13.35% (-39.87%)	$0.00 % of Total: 0.00% ($0.00)
1.	/courses/21/materials	1,289	958	00:00:33	74	13.51%	6.36%	$0.00
2.	/courses/21/forums	1,199	704	00:00:39	92	7.61%	5.00%	$0.00
3.	/courses/21/forums/trainings	707	319	00:00:39	9	11.11%	4.24%	$0.00
4.	/courses/21/lesson_plan	336	190	00:01:58	14	14.29%	13.39%	$0.00
5.	/courses/21/materials/folders/22	330	245	00:02:35	10	60.00%	23.03%	$0.00
6.	/courses/51/forums/general-discussion--51	296	36	00:00:57	7	0.00%	4.05%	$0.00
7.	/courses/51/forums/general-discussion--51/topics/test	269	22	00:00:51	6	0.00%	4.09%	$0.00
8.	/courses/21/forums/missions-side-quests-contests	261	130	00:00:41	5	0.00%	4.98%	$0.00
9.	/courses/54/forums/general-discussion--54	249	46	00:01:01	10	20.00%	6.43%	$0.00
10.	/courses/21/forums/midterm-and-exam/topics/thoughts-comments-on-the-practical-exam	234	175	00:05:23	15	53.33%	27.78%	$0.00

Show rows: 10 Go to: 1 1 - 10 of 784

Coursemology’s userbase will also be seeing an increase in the future. Prof Ben Leong will be taking using the platform with a course of close to a thousand students via a MOOC next semester. In addition, Asst. Prof. Seth Gilbert intends to utilise Coursemology for his module CS2020 Data Structures and Algorithms Accelerated, and A/Prof Martin Henz intends to replace the JFDI Academy with Coursemology for CS1101S Programming Methodology.

Future Improvements

Since deployment to the live site, we have iterated a few times based on feedback from users. Though the overall feedback we have received at this point is positive, there are still some areas which can be improved upon.

For example, we have received a feature suggestion relating to the forum’s navigation. Instead of rendering each thread on a separate page, we could asynchronously load the thread content whenever a user clicks on the thread title, to achieve an ’expand - collapse’ effect. Another area for improvement is the structuring of replies. While in our current linear presentation, the ‘in reply to’ link maintains the information about who replied to what, this piece of information is not very well presented. Users have suggested an indented layout (similar to the layout found on Reddit and Hacker News) to better present such relationship.

In the long run, we do intend to roll Coursemology out to a greater number of modules so that both teaching staff and students alike can benefit from our work. We have to further examine the usefulness of Coursemology for non-Computer Science courses, however.

Finally, we also hope that at some point Coursemology will be able to supersede and replace IVLE.

Contributions

Minqi and Fazli were responsible for the implementation of the Discussion Forum. In particular, Fazli was responsible for the early integration of the Forem engine into Coursemology as well as implementing add-on features including likes and unread support. Minqi worked on features including subscription, post aggregation as well as automatic suggestion of forum participation experience points.

Jerome and Joel were responsible for the Workbin and the Lesson Plan. Joel was responsible for implementing the necessary back-end components for both features, as well as laying out the initial skeleton views. Jerome was responsible for front-end development and refinement.

We would like to express our gratitude to Raymoond Tang, the current developer and maintainer of Coursemology, for merging our code and deploying it to production, even when courses were actively using the platform. This was not a simple task, as we introduced quite a large number of potentially breaking changes to the system - some of which did end up breaking the entire platform.

Many students from CS1010S Programming Methodology, including Jiang Jing, Hong Xiangyu, Zhou Yichen, Gao Yuan, Cui Xiaoning and Liu Siyuan, have provided valuable feedback as users.

We also wish to express our thanks to Asst. Prof. Ben Leong and the CS3216 teaching staff for their guidance and feedback throughout the project.