Optimising Mathematical Content Creation and Management Systems for Dig-it

Ethan Marrs
University of Cape Town
18 University Avenue
Cape Town, South Africa
mrreth002@myuct.ac.za

Nathan Begbie
University of Cape Town
18 University Avenue
Cape Town, South Africa
bgbnat001@myuct.ac.za

1. PROJECT DESCRIPTION

Dig-it is a mathematics mobi-site aimed at high school students. It supplies them with questions on a weekly basis and rewards them with airtime if a certain percentage of questions are answered correctly. Dig-it currently operates on a small scale with roughly 1500 students from Grades 10 to 12. Dig-it is built and managed by the Praekelt Foundation and is funded by Investec. Dig-it is built with Python and Django, a web framework. In the next year, Dig-it will be scaling to a national level of over 100 000 students in 3 provinces. Various stakeholders engage in the content creation process through various platforms, with multiple stages of the process managed by different systems. As Dig-it scales, it requires tools to manage this scaling effectively, through better content creation and management systems. Dig-it needs to be an inclusive platform that allows for full control of content creation and content management in one place. It currently faces a number of issues.

On the content creation side, the underlying mathematical data is not stored within the platform. Content is created using Microsoft Word and the MathType equation editing tool. These documents are emailed to a moderator who edits and returns the content, which is then sent to a content uploader, who takes screenshots of the equations and copies the text across to the platform. This process should be internalised within the platform to reduce organisational complexity.

From the perspective of content management, Dig-it is facing a number of problems. The Content Management System (CMS) is distributed and unwieldy, making use of external platforms; Asana, email, online documents and spreadsheets keep track of content changes and moderation. This also means that content cannot be automatically released – the platform manager needs to manually make new content available to students. Managing the addition of new questions and modules is brittle, with a number of steps and many potential points of failure.

Lastly, Dig-it needs to optimise content delivery. Dig-it's differentiating factor as an online education platform is that it provides a valuable experience for those accessing the content on feature phone, as well as smartphones. It needs to reduce the size of the content that it serves to learners.

2. PROBLEM STATEMENT

2.1 Aims

The aims of this project are:

- Allow content creators to create content within the platform in order to store the underlying data as LaTeX and text rather than images and text, and to optimize the content delivery system.
- Internalize the syllabus and content management within the platform as much as possible, for intuitive use by stakeholders, to manage the content as the platform scales.
- Automate parts of content management system as identified by the client.
- To reduce the load size of the pages that are sent to learners.

2.2 Research Questions

While not a research project, we wish to answer the following questions as the focus of this software engineering project:

- To what extent can the mathematical content be created by the platform, specifically with regards to equations by introducing a WYSIWYG equation editor?
- To what degree can a content management system reduce the reliance on external tools for coordinating content workflows?
- By what percentage can we decrease the average page size of questions and answers sent to students?

2.3 Software Engineering Project Requirements

Supervisor:	Melissa Densmore
Client:	Lauren Kotze, Project Manager of Dig-it
Users:	Project Manager
	Platform Manager
	Syllabus Creator
	Content Creator
	Content Uploader
	Learners

This project has been requested by the Dig-it team as a prototype, to demonstrate whether a large-scale upgrade is worthwhile. The current system is quite complicated, as indicated by **Figure 1.** The client's initial requirements are to streamline content creation and management and optimise content for use on feature phones. The aims stated above should adequately meet these requirements and are in line with the vision for the platform. **Figure 2** represents a high level overview of the proposed system that the client requires. The requirements can be broken down into 4 parts.

2.3.1 Models Schema

The models need to be changed to reflect better management and creation of content. This will be done by investigating the changes that are needed for the current models in order to improve content management.

2.3.2 Content Creation

Currently, content is created in MS Word documents and a content uploader manually copies the text and screenshots of equations to the Dig-it CMS. Dig-it would like to internalise the content creation process, remove the need for a content uploader and remove the dependency on MS Word.

2.3.3 Content Management

The current content management system involves a number of manual interactions and external tools. As many of these business processes need to be incorporated into the Dig-it platform as possible if the platform is expected to scale effectively in future. This includes managing questions, modules and classes, as well as moderating questions.

2.3.4 Content Distribution

The web pages containing content sent by the Dig-it platform need to be reduced in size, which will be done by implementing an automated optimisation process on the data and by modifying what content a device requests, depending on the device's capabilities.

3. PROCEDURES AND METHODS

This section discusses the software development practices and methods that we will use throughout the project.

3.1 Developmental Procedures

3.1.1 Database Schema

The database is created using Django models. The current schema will need to be analysed and modified to allow for changes to the content creation system and content management system. Design and iteration of the models will be done in collaboration with the Project Manager. Designed changes will be made to the Django models and where possible, automatic migrations will be applied. Unit tests will then be written to test the model functionality.

3.1.2 Content Creation

A What You See Is What You Get (WYSIWYG) equation editor will be used to create the math content, and store the underlying data as LaTeX. A WYSIWYG equation editor will need to be created that allows the content creator to intuitively create equations on the platform. These equations would need to be saved in LaTeX. A javascript/HTML/CSS WYSIWYG editor will be created as a plugin for the Dig-it CMS, that will allow content creators of Dig-it to create equations within the Dig-it platform. This will be created by forking and adapting the open source "Equation Editor" [1]. Once the plugin is implemented, it will be manually tested to determine whether these equations can be created using the plugin and that this is a viable alternative to MS Word in terms of functionality. If possible, these front-end tests will be automated for repeated testing. A sample of questions currently on the Dig-it platform will be selected to use as testing questions to confirm that the equation editor has adequate functionality.

We will then iteratively improve the equation editor's capabilities by ensuring that an increasing number of the sample questions can be implemented. Throughout this process, we will iteratively receive feedback on design with the Project Manager, content creator and uploader to optimise design and achieve client satisfaction. We will also schedule meetings to conduct informal user testing with content managers and content creators, to optimise the interface.

3.1.3 Content Management

Currently, content is managed through a customised administration interface provided by the Django framework. The process of managing content involves workflows that are external to the Dig-it platform. **Figure 1** indicates the current system, which involves Asana for tracking content creation, Google Sheets for managing the syllabus and email for coordinating the moderation of questions. To improve this system, changes to the workflows, interfaces and data structures will need to be made.

Modifications to the system will start off with detailed analysis of the current processes and implementation. Once the current system is fully understood, relevant database schema changes will be made to allow for improvements. From here, regular scrum iterations will focus on building an improved content management system. The improvements to the content management system will focus on:

- Allowing for automatic or semi-automatic content availability, which involves making questions available to students at certain points in the syllabus
- Improved arrangement and selection of questions for modules
- Reducing dependencies on external tools
- An inclusive system that manages the moderation of questions before release

To allow for the most usable content management system, visual methods may be used to help display modules, questions and courses. This would involve the use of the D3.js library, or other suitable open source libraries that fit into the Django stack. Stakeholder feedback will be incorporated into the scrum iterations, with feedback from the project manager and system administrators informing design decisions.

3.1.4 Content Delivery

In order to reduce content size, we will need to incorporate processes that will automatically optimise the data once it has been saved. In order to do this, we will evaluate OSS libraries that convert LaTeX equations to images and image compression libraries to losslessly compress images. We will then implement an automation process that creates the images of equations of the LaTeX and compresses them. We will then create 2 web-page templates that contain the same content, one with optimised images, automatically created from the LaTeX equations and another using a client-side equation rendering library, to display the LaTeX equations. As a means of testing how effective the implementation will be, we will select 50 questions currently stored in the Dig-it database at random and measure the page size of those questions. We will then compare the optimised pages and record the average reduction in page size for each method. In order to serve these images to the correct device, we will implement a progressive enhancement technique to serve images to feature phones and LaTeX rendered on the server side with KaTeX to smart phones. Throughout this process, we will iteratively receive feedback from Project Manager on levels of performance for optimisation and whether more time must be spent optimising the pages sent to learners.

All of the system elements listed above will be written in Python or Javascript.

3.2 Development Practices and Methods

In this project we will be making use of Scrum, the agile software development methodology. Given that there are only two team members, we will not be able to follow a formal Scrum implementation. Rather, we will focus on using Scrum to coordinate work, manage time and involve stakeholders.

Throughout the project we will participate in:

- Structured development iterations
- Coordinated backlog grooming
- Sprint planning
- Sprint reviews
- Regular standup meetings
- Client product reviews and feedback sessions

We will also be making use of Test Driven Development (TDD) in order to better structure our progress and ensure the reliability of the Dig-it platform. This will involve using the “red, green, refactor” process of writing tests for features before the actual code is written.

3.2.1 Tools

All of the tools listed below are free and open source software. The collaboration and testing infrastructure will be provided by Dig-it.

3.2.2 Software and libraries

Python

Django

Celery

KaTeX

D3.js

3.2.3 Testing software

Pytest

Phantom.js

3.2.4 Collaboration and Testing

Github repositories

Travis CI for automated testing

3.3 Methods for Measuring Success

We will be measuring the success of our platform through the following

- Client satisfaction – Dig-it will evaluate our system compared to their legacy system and the project will be determined a success if our system provides increased performance and optimises workflow.
- Quantitative measure – Measuring the degree to which we have reduced page sizes for learners
- Our success will also be measured by the degree to which our improvements can replace Dig-it’s dependency on multiple platforms

Details on the specific metrics that we will be using for success are listed in Evaluation under Anticipated Outcomes.

4. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

A local copy of the database will be required for development. To avoid issues with the leaking of personal data, we will only be provided with the relevant production database info from the Praekelt Site Reliability Engineering team. The databases required for development will be secured with password protection. Software used in the creation of this project will be open source and correctly credited. Software created by this project will be released as Open Source Software under Praekelt Foundation's’ license which is derived from the MIT license.

5. RELATED WORK

The importance of reducing page size is two-fold. It reduces the data costs that a user faces and reduces slow page load times. Chen et al. [2] note that if page load times are high under low bandwidth conditions, users are more likely to abandon the task they set out to do.

Miner [3] points out the difficulty in using equation in an online context. Various different standards and methods mean that there is no universal equation language for the web. While there are various methods for structuring mathematical content, it is far more intuitive for users to interact with equations in a natural way. In discussing WYSIWG editors, Van der Hoeven [4] point out the steeper learning curve with purely text based structured text and lists controllability, transparency and readability as advantages of the WYSIWG editor. By using a WYSIWG editor, a person can intuitively create correctly typeset equations, without the need for further specialised knowledge and training. Only one open source web based WYSIWG editor was identified by Dig-it as a viable solution to the problem [1]. Dig-it has also identified a library that is able to render LaTeX equations in most browsers [5].

The current process of managing questions, modules and courses involves multiple platforms and workflows. This is depicted in **Figure 1**. As discussed, the current process is not ideal, and would benefit from increased automation and unification. This process is described in the literature as “workflow automation”. This involves the integration of heterogeneous systems, increasing efficiency of workflows and migrating manual tasks to electronic platforms [6]. For Dig-it, this is precisely the intervention that is required. Specifically, Dig-it needs to make use of a time-constrained workflow system, which is a system that includes deadlines in the workflow definition [7]. Essentially, this would allow for the workflows to have an understanding of the two-week content availability deadline, and incorporate administrators as required.

6. ANTICIPATED OUTCOMES

6.1 System

This project aims to build a prototype that demonstrates a number of capabilities. Generally, the features and modifications to the platform should work towards Dig-it’s goal of scaling to thousands of users in South Africa.

This project will implement a WYSIWYG equation editor that operates within the browser and stores the underlying data as LaTeX. This will allow both content creators and content moderators to work on the content within the platform.

We will implement an automated content transformation and compression system that makes use of the new data format to reduce load size for learners. This will mostly focus on delivering the best format based on the user's device, which will involve image compression for certain users.

With respect to the content management aspect of the project, an improved management system is anticipated to streamline current processes and allow for all tasks to be carried out in one place on the Dig-it platform. The improvements to the content management system would also allow for questions to be automatically released to students, rather than the current method which requires the platform manager to manually release content every two weeks. This is technically challenging based on the fact that the representation needs to allow for delivery to be automated using a scheduled queue, as well as improving the workflows of administrators.

6.2 Project Impact

We expect our prototypes to demonstrate drastic improvements to the usability and sustainability of the Dig-it platform. Primarily, we expect our project to improve both the processes of creating content and managing this content on the platform. We also expect our project to demonstrate reduced load times for users and improved database schemas going forward.

6.3 Evaluation

The following metrics will be used for measuring the success of the project:

- Reduce the page size for feature phones by at least 20%.
- Reduce the page size for smart phones by 40%.
- Replace the creation of mathematical content with internal processes rather than MS Word.
- Remove the need for a content uploader.
- Reduce the reliance on external tools such as Asana, Google Docs, Google Sheets and email correspondence for managing content.
- Ensure as much content management tracking information as possible is kept on the Dig-it platform.
- Allowing for automated or semi-automated release of questions on a schedule.
- Stakeholder satisfaction with changes to the workflow of the platform.
- Achieving test coverage of 80% for all code written.

7. PROJECT PLAN

7.1 Work Allocation

The project can be divided into two main sections. The first section focuses on content creation and delivery. This involves incorporating an open-source WYSIWYG equation editor to allow non-technical content creators to set mathematics questions, as well as reducing the load size of web pages delivered to learners on the platform. Nathan will be working on the content creation and delivery aspect of the project. The second section focusses on content management. This involves designing and enhancing the current content management system to allow syllabus creators, content moderators and content creators to easily understand and organise course content. Ethan will be working on the content management aspect of the project. Other work required for the project, such as designing and modifying database schemas, will be shared between Nathan and Ethan.

7.2 RISKS

Risk	Likelihood	Impact	Significance	Mitigation Strategy
Timeline is too short for proposed amount of work	5	8	40	Start working on the project as early as possible, plan out a schedule with slack time
Time estimations are inaccurate and lead to project delays	5	6	30	Consult with project stakeholders on task time estimations
Understanding the Dig-it platform delays progress on the project	7	6	42	Consult with engineers and project stakeholders, allocate time for understanding the platform
Making changes to the Dig-it platform proves to be difficult	5	5	25	Consult with project stakeholders, make use of familiar frameworks and technologies
Project scope is too large	4	7	28	Consult with project stakeholders, set some of the ambitious tasks as stretch goals
Equation Editor Library lacks needed functionality	3	8	24	Reduce scope of content optimisation in order to allocate more time to adding functionality
Satisfying the requirements of all stakeholders involved in the content management process is not possible	3	8	24	Focus on particular stakeholders' needs, consult with stakeholders on how the current system could be improved

7.3 Timeline

See **Figure 4** in the Appendix.

7.4 Milestones

See **Figure 4** in the Appendix.

7.5 Resources

- Time of the project manager and team members of Dig-it
- Dig-it code base
- Dig-it documentation
- Dig-it teaching materials
- Anonymized copy of Dig-it database

7.6 Deliverables

- Documentation
    - Collation of business rules
    - All system changes and new software
    - Analysis of current tools and functionality compared to prototype
- How-to’s for stakeholders
- An outline of the final project report.
- A complete draft of the final project report.
- The final project report.
- A project poster.
- A website for the project.
- A reflection paper based the project.

8. REFERENCES

[1] Camdenre/equation-editor: 2016. https://github.com/camdenre/equation-editor. Accessed: 2016- 06- 05.

[2] J. Chen, L. Subramanian, and K. Toyama. Web search and browsing behavior under poor connectivity. In CHI'09 Extended Abstracts on Human Factors in Computing Systems, pages 3473-3478. ACM, 2009

[3] R. Miner. The importance of mathml to mathematics communication. _Notices of the AMS_, 52(5):532–538,2005.

[4] Van der Hoeven, J. 2001. GNU TeXmacs, A free, structured, wysiwyg and technical text editor. Cahiers Gutenberg. 39-40 (2001), 39-50.

[5] Khan/KaTeX: 2016. https://github.com/Khan/KaTeX. Accessed: 2016- 06- 05.

[6] Stohr, E. and Zhao, J. 2001. Workflow Automation: Overview and Research Issues. Information Systems Frontiers. 3, 3 (2001), 281-296.

[7] Hyun Son, J. and Ho Kim, M. 2001. Improving the performance of time-constrained workflow processing. Journal of Systems and Software. 58, 3 (2001), 211-219.

9. APPENDIX

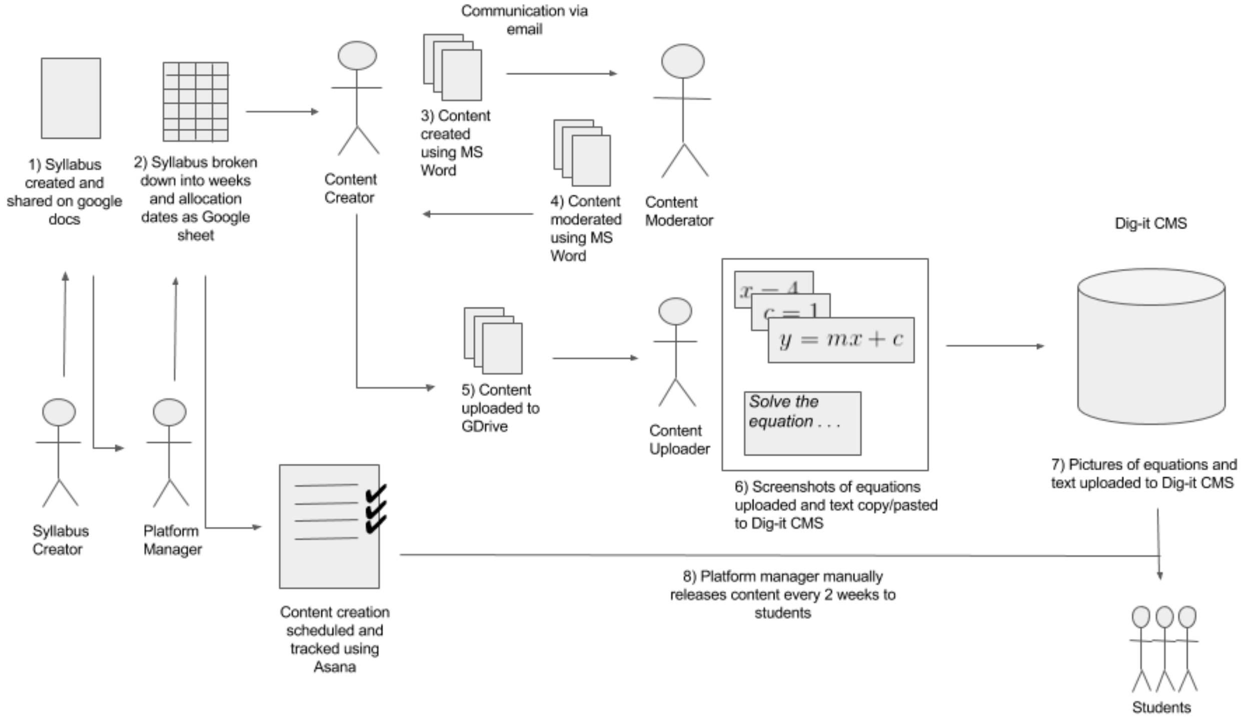

Communication via email
1) Syllabus created and shared on google docs
2) Syllabus broken down into weeks and allocation dates as Google sheet
Content Creator
3) Content created using MS Word
4) Content moderated using MS Word
Content Moderator
Dig-it CMS
$x = 4$
$c = 1$
$y = mx + c$
Solve the equation . . .
5) Content uploaded to GDrive
Content Uploader
6) Screenshots of equations uploaded and text copy/pasted to Dig-it CMS
7) Pictures of equations and text uploaded to Dig-it CMS
Syllabus Creator
Platform Manager
8) Platform manager manually releases content every 2 weeks to students
Content creation scheduled and tracked using Asana
Students

Figure 1: Current Dig-it Processes

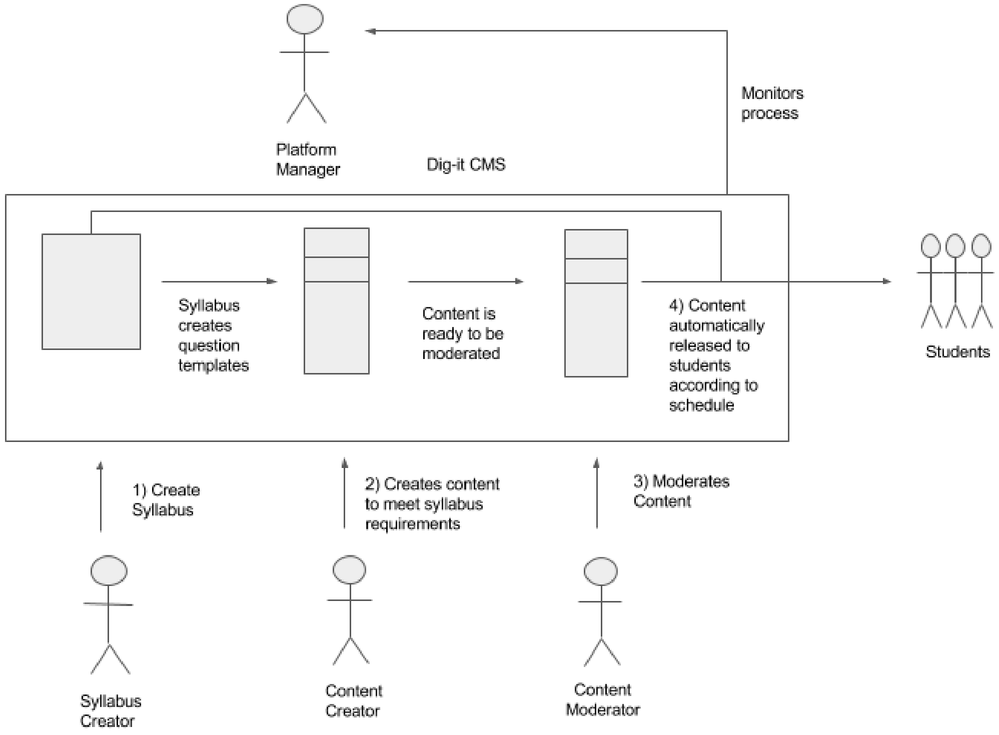
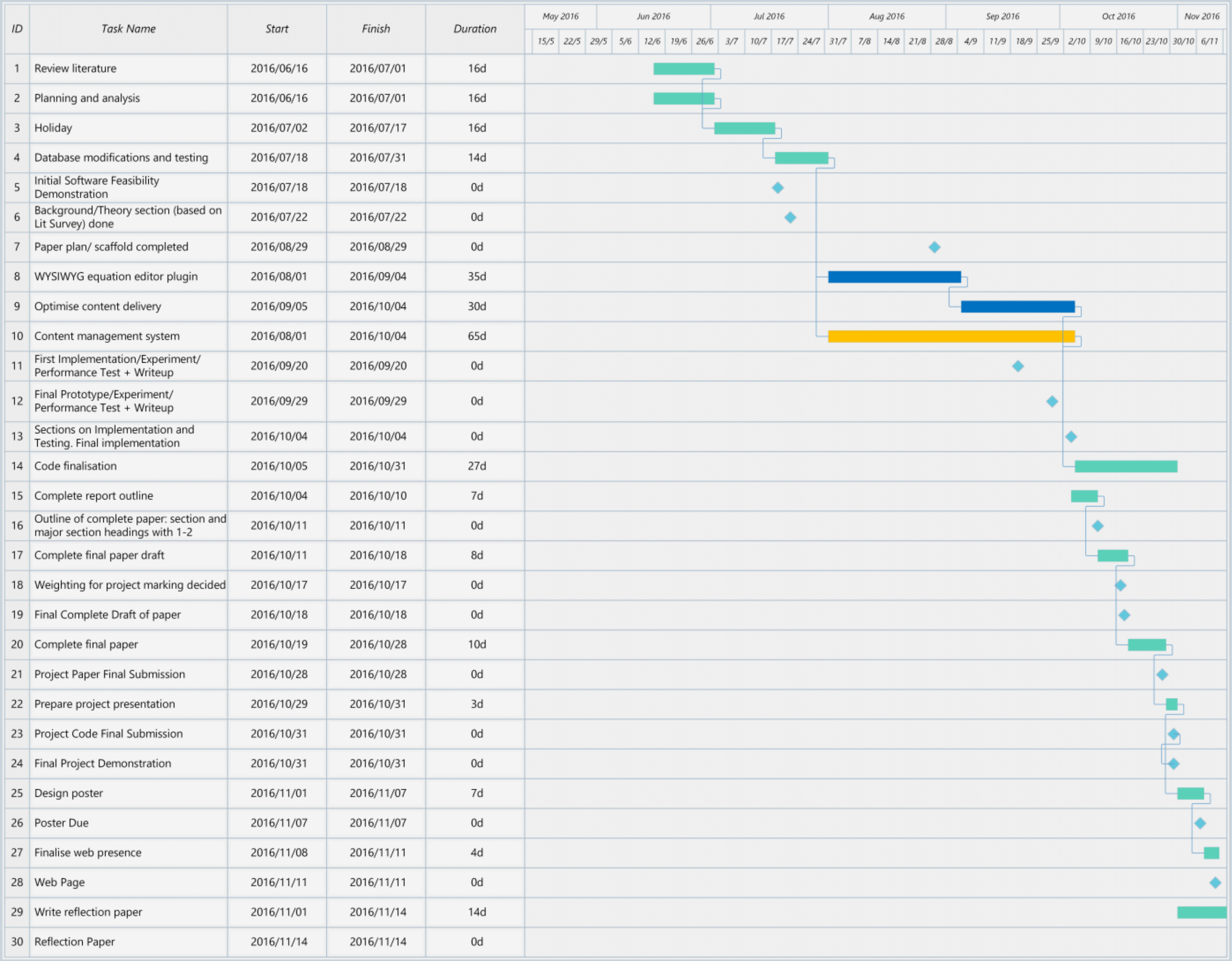

Platform Manager
Dig-it CMS
Monitors process
Syllabus creates question templates
Content is ready to be moderated
4) Content automatically released to students according to schedule
Students
1) Create Syllabus
2) Creates content to meet syllabus requirements
3) Moderates Content
Syllabus Creator
Content Creator
Content Moderator

Figure 2: Proposed Dig-it Processes



ID	Task Name	Start	Finish	Duration
1	Review literature	2016/06/16	2016/07/01	16d
2	Planning and analysis	2016/06/16	2016/07/01	16d
3	Holiday	2016/07/02	2016/07/17	16d
4	Database modifications and testing	2016/07/18	2016/07/31	14d
5	Initial Software Feasibility Demonstration	2016/07/18	2016/07/18	0d
6	Background/Theory section (based on Lit Survey) done	2016/07/22	2016/07/22	0d
7	Paper plan/ scaffold completed	2016/08/29	2016/08/29	0d
8	WYSIWYG equation editor plugin	2016/08/01	2016/09/04	35d
9	Optimise content delivery	2016/09/05	2016/10/04	30d
10	Content management system	2016/08/01	2016/10/04	65d
11	First Implementation/Experiment/ Performance Test + Writeup	2016/09/20	2016/09/20	0d
12	Final Prototype/Experiment/ Performance Test + Writeup	2016/09/29	2016/09/29	0d
13	Sections on Implementation and Testing. Final implementation	2016/10/04	2016/10/04	0d
14	Code finalisation	2016/10/05	2016/10/31	27d
15	Complete report outline	2016/10/04	2016/10/10	7d
16	Outline of complete paper: section and major section headings with 1-2	2016/10/11	2016/10/11	0d
17	Complete final paper draft	2016/10/11	2016/10/18	8d
18	Weighting for project marking decided	2016/10/17	2016/10/17	0d
19	Final Complete Draft of paper	2016/10/18	2016/10/18	0d
20	Complete final paper	2016/10/19	2016/10/28	10d
21	Project Paper Final Submission	2016/10/28	2016/10/28	0d
22	Prepare project presentation	2016/10/29	2016/10/31	3d
23	Project Code Final Submission	2016/10/31	2016/10/31	0d
24	Final Project Demonstration	2016/10/31	2016/10/31	0d
25	Design poster	2016/11/01	2016/11/07	7d
26	Poster Due	2016/11/07	2016/11/07	0d
27	Finalise web presence	2016/11/08	2016/11/11	4d
28	Web Page	2016/11/11	2016/11/11	0d
29	Write reflection paper	2016/11/01	2016/11/14	14d
30	Reflection Paper	2016/11/14	2016/11/14	0d

Figure 3: Gantt Chart