A Learning Management System for Experimenting with Different Tools and Methods for the User Interface

Jackson Xie

Abstract

Using a learning management system (LMS) is quite common for institutions. A learning management system provides services for teachers and students to interact with each other online, and a platform for sharing resources and information. Inspecting currently used LMSs shows that although current LMSs provide plenty of features for users, their outdated user interfaces are becoming a barrier for attracting more users and improving the effectiveness of the systems.

The purpose of this project is to explore how to build a prototype learning management system in the form of a website, and to try to experiment with a modern user interface. The content of the website includes both the client and the server side. The project can be divided into two parts. In the first part of the project, we implement basic features of online study, such as user management, assignment collecting, and forums for discussions. In the second part of the project, we use different tools and methods, such as Data-Drive Documents (D3), to improve the user experience by providing new ways for web-user interactions.

1 Introduction

Learning management systems came into existence when companies started to realize the potential in them. Currently widely used systems, such as Blackboard/WebCT appeared as early as 1997. It still has millions of users at the time of writing. A survey [6] shows that in the USA in 2011, 32% of higher education students were taking online courses, which means over 6.7 million students in 2011. In 2012, there was an increase of 570,000 students. Though the increase became small in 2014 [7], the number of participants in online courses still goes up each year.

A learning management system (LMS) is a software application for the administration, documentation, tracking, reporting and delivery of electronic educational technology (also called e-learning), education courses, or training programs. [9] The definition of learning management systems is based on that of content management systems. A content management system (CMS) is a computer application that allows publishing, editing and modifying content, in addition to, organizing and deleting, as well as maintenance, from a central interface. [10]

According to the basic definition of a CMS, it deals with storing and organizing files, and provides version-controlled access. When some features for online learning

have been added to a CMS, it often becomes known as an LMS. A learning management system often has features such as: course management, a forum, and provision for including online teaching media.

CMS and LMS have some similarity regarding the challenges that they have to deal with: How to provide a user interface? How to let developers do modifications? How to make the user interface extendable? They also share other challenges like support for massive access, security problems, and flexibility of programming structure.

Providing services for a large number of students, there are many platforms functioning as learning management systems available on the internet. Websites like Youtube, Udemy, Edx, Moodle, Blackboard, provide platforms, which academic leaders and institutions can use to start online courses, and teachers and students can use to interact with each other without meeting.

Youtube provides limited functionality for online study, but it is good for sharing and publishing videos of lectures. Udemy and Edx, similar to Youtube, provide a platform for publishing series of lectures in addition to other features such as course discussion, assignment, and course process. Those websites like Udemy and Edx are also classified into the category of MOOC (Massive Open Online Course). They mainly concentrate on providing courses for a large number of students online.

The system proposed in this project is more similar to Moodle or Blackboard. Moodle and Blackboard, as learning content management systems, are widely used for online learning. They provide blended study for institutions, giving platforms for students and teachers to interact with each other online. They have many features, are open source projects and tested and used by many. They perform as plug-ins to websites, and are easy to install and use. But one of the downsides of them is that their user interfaces are outdated. They remain unchanged in spite of the general improvement of user interfaces of websites around the world.

We have seen a need for user interface improvement and underline the potential in it. Flexibility, user experience, and structure of the UI can be improved. Improving the UI may not be the first priority of online learning websites, but it can keep people away from annoying situations, save their time and make the website more reliable, thus improving the overall utilization of the platform.

The proposed software provides a series of basic features to help teachers and students interact with each other online, and tries to improve the user interface at the same time. It has features such as user management, user privilege administration, file management, file submission and downloading, and a forum. It uses Data-Driven Documents to provide flexible web-user interactions.

2 Related Work

In order to choose some leading LMSs for deeper investigations, we first looked into the area of learning management systems for higher education. There are many learning management systems available online, either paid or free-to-use. Institutions can choose among those systems according to their requirements and financial

situations. According to Campus Computing Project's Campus Computing Survey 2010 [2], leading commercial learning management systems are Blackboard/WebCT, Desire2Learn, and Pearson's eCollege. Leading open source projects are Moodle, Sakai, and Canvas by Instructure. Commercial LMSs have the advantage of having come into the area earlier than open source LMSs. Blackboard came into the area as early as 2000. Many institutions have already installed an LMS like Blackboard. They will need a considerable amount of investigation and efforts if they want to change to another LMS. But the number of users of open source LMSs is growing faster than that of commercial LMSs. [4]

Since users of Blackboard and Moodle account for more than half of all the users in the area [4], we gathered general information about their feature lists and usage situations. Also, we compared Blackboard and Moodle by the feedback given by their clients, looked deeply into their feature differences, and the positive and negative feedback. We wanted to find out not only fundamental and necessary features that are needed, but also what users need and what things we should avoid.

In terms of the functionalities and features that LMSs provide, they have a lot in common. Usually, an LMS will provide tools for course management, homework collecting and grading, and forums for discussions and communications. Some more complex features are due-day reminders, event calendars, dividing students into groups for discussions, detailed information for students and teachers, online tests and projects. Different LMSs provide different sophisticated features, but their basic features are the same.

If we separate the basic feature into two types, knowledge features and interaction features, we can find that teachers and students use knowledge features more often than interaction features. According to a survey [3] based on an LMS called CEIBA, knowledge features such as course details, student information, FTP files, announcement boards, emails, have less than a 20% chance to be ignored by students or teachers. On the other hand, interaction features such as forums, assignments, resource sharing, grouping students, and chat rooms have a 30% to 70% chance to be ignored. The most used interaction features are forums and assignments. Although usage of features varies between different systems, we can conclude that knowledge features are used by more students and teachers because of their simplicity and clear objective.

After looking at the general features and usage of LMSs, we consider the feedback from students and teachers about two leading LMSs, Blackboard and Moodle.

Moodle provides activities, and classifies them by weeks or topics. Blackboard classifies its posts by types. The former approach is more appreciated by students and teachers. [1]

Blackboard has the advantage of sharing documents between students. [1] Students can easily share their drafts of assignments and comment on them, leading to more discussion between students. Moodle lacks this feature and has a limit on modifying posts, which leads to less communication.

Moodle is appreciated for its flexibility. [1] Teachers can easily change the theme

for their courses to provide a suitable site for students.

In terms of the user interface, Moodle has more advanced features and is easier to use than Blackboard. [1] Moodle provides editing tools for users to edit a variety of content. On the other hand, Blackboard provides a lot of links and often needs three clicks to complete an action. Moodle provides a quick switch between student role and teacher role, which is also a feature to make it do better in terms of the user interface.

Although Moodle does better than Blackboard on the user interface, they both share common drawbacks. Many useful functions are ignored by students. [1, 5] This might be because too many functions are shown to users at the same time, or are not clearly formatted or not targeting users well. A survey [5] has shown that art and management school students use less of LMSs than science and engineering students, which tells us that more things can be done about the user interface and user targeting to improve overall usage of LMSs.

3 Methods and Implementations

In order to experiment within the user interface, we need the standard features of a learning management system to start with. What we need is a simple prototype LMS system with features for user privilege management, assignment collecting and forums. We did not use current open source LMSs such as Moodle, because these systems have many features that we will not experiment with. We would like to make the system simple, so that we can perform the user interface experiment in a more flexible environment. Consequently, instead of using an existing LMS, we developed our own prototype LMS with those basic features we demanded.

3.1 General Tools for Implementations

We built our learning management system on a Windows machine using Eclipse J2EE for Web Developer as the IDE. We did not use any framework for web development such as Spring or Maven. After an approximate prediction of the size of the system, we started our system from scratch instead.

At the programming level, our tasks can be divided into tasks on the client side and tasks on the server side. The client side of our system is built by using standard web development tools: HTML, Ajax, and jQuery, in addition to Bootstrap and Data-Driven Documents (D3). Ajax and jQuery are used to build client-server communication. Bootstrap and D3 are used to deal with the user interface. The server side of our system is built by using Java, MySQL and Tomcat 7.0. MySQL is a more complex method than SQLite, but it provides better flexibility, and its documents and tutorials are easy to find. We chose Tomcat as our server, because it has been tested over many years. It has sufficient documentation and tutorials. It is flexible and also integrated well with Eclipse. We developed both the client and the server side in Eclipse and ran the client on IE, Firefox and Chrome.

3.2 Basic Features of the Learning Management System

Our learning management system contains features for privilege management, assignment collecting, and forums. These features are the most common ones and used by most teachers and students, according to our previous research.

For a user to gain access to any of the pages on the website, he will need to log in. If his account has not been created yet, he will be informed to sign up in order to go further. The sign up page and log-in page are on the same page. A user can sign up as a student or teacher. After signing up, he will be able to browse the website.

The content of the website will be different according to the role of the viewer. A student will be able to upload assignments and comments on forums. A teacher will be able to post assignments, add comments on forums and create articles.

3.3 User Interface

The design of the user interface of this project is inspired by different sources. We looked at famous websites like Pinterest and Facebook. We also checked newly established tools, such as D3 (Data-Driven Document) and HTML5.

Some of the design of the user interface of this project is inspired by D3 (Data-Driven Document) example projects. Some of the components are directly derived from D3 example projects. We tried to use D3 as much as possible to visualize data structures and to provide navigation.

D3 is a JavaScript library for manipulating documents based on data using HTML, SVG and CSS. [12] D3 is good at showing large amounts of data in graphics. Some of the examples are shown in Figure 1.

In the UI section, we only used D3 for the navigation bar. We did not use D3 for showing complex content such as posts, which may contain buttons, images and links. D3 is good at showing graphics and relationships of data, but it lacks the ability to provide complex interactive elements inside the graphics. This is because the graphic system of D3 is based on Scalable Vector Graphics (SVG), rather than HTML5. SVG is an XML-based vector image format for two-dimensional graphics with support for interactivity and animation. [13] Many of the D3 examples only have plain text inside the graphics or a single image.

There is another reason for us to use D3 to make the user interface: D3 provides transform animations based on position, scale, data, and color. Almost all the examples provided by D3, for example those examples in Figure 1, are animated, smoothly transforming through different states. This provides users a much better experience with user interfaces.

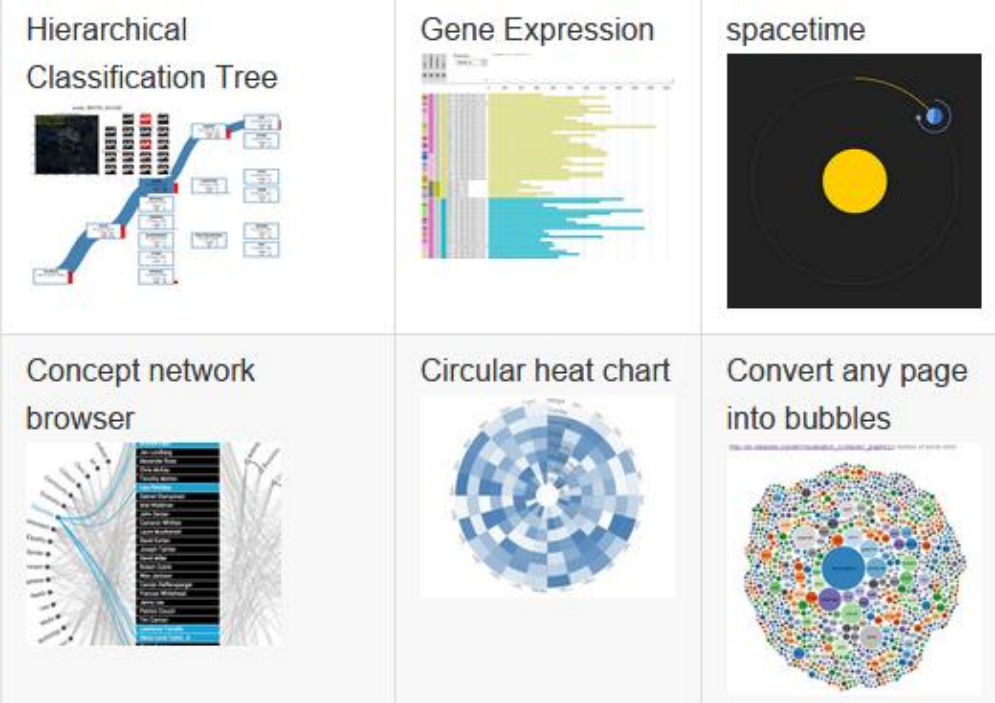

Figure 1. Some Examples of Data-Driven Documents projects (https://github.com/mbostock/d3/wiki/Gallery)

3.3.1 Navigation Bar

The navigation bar of the website is derived from one of the D3 example projects, Zoomable Partition Layout. [13] The original example shows an example of browsing data with a tree structure. A screenshot of the navigation bar is shown in Figure 2. The first column is the code of the course. The second column contains different weeks of the course. The last column shows different contents for each week, which are often created by teachers. Contents like posted assignments, forums for discussion, presentations of classes are in the last column.

The navigation bar will show up only when the user is viewing one of the courses. It will zoom in when the user clicks on a nested section, and zoom out when the user clicks on the left side. It shows leaf nodes, and contents like assignments or forums, in different colors. When the user clicks on a leaf node, a page about the node will open up and show the details.

This navigation bar provides quick navigation in a deep data structure. Students or teachers can directly check the assignments or forums in different weeks without clicking several times or loading extra pages. The navigation bar also gives users a sense of where they are by zooming in and out according to the user's currently selected section.

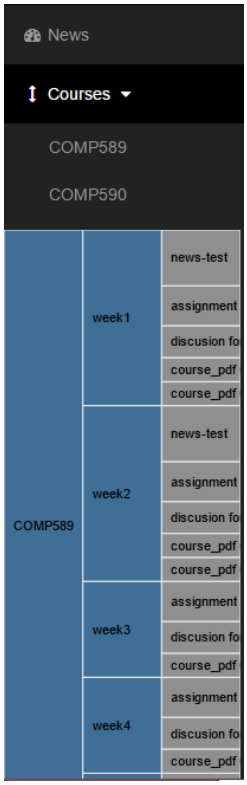

Figure 2. The navigation bar

3.3.2 Forums section

The design of the forum section is similar to the design in Facebook or Tumblr. It is essentially a list of posts. The users can scroll down the website and quickly scan across all the content. A screenshot of the forum section is shown in Figure 3.

All the comments on the posts are collapsed by default. When a user clicks on the reply or view comments button, the comments and a reply widget will show up. The process of expanding is animated by using CSS with HTML5, so it looks smooth. The list of posts will also respond to the resizing of the browsing window. It will try to reposition and resize itself to the center of the section with a proper size.

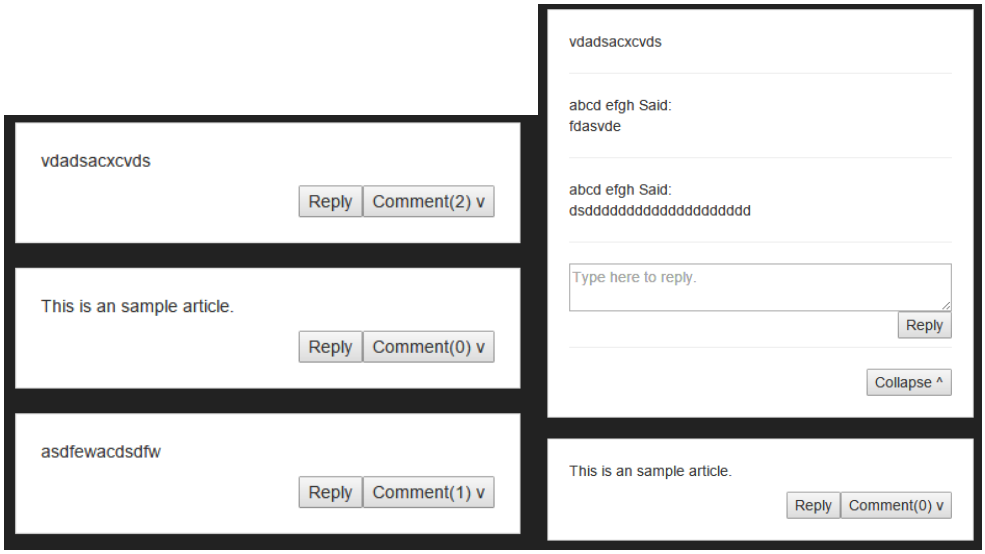

Figure 3. A screenshot of the forums section. Left side: the default and collapsed view of a forum. Right side: expanded view of a forum.

3.3.3 Assignments section (student's view)

The assignment section for students is similar to the forum section except that in addition to putting comments on the forum, students are able to submit assignments by submitting files. Figure 4 shows an example of an assignment section.

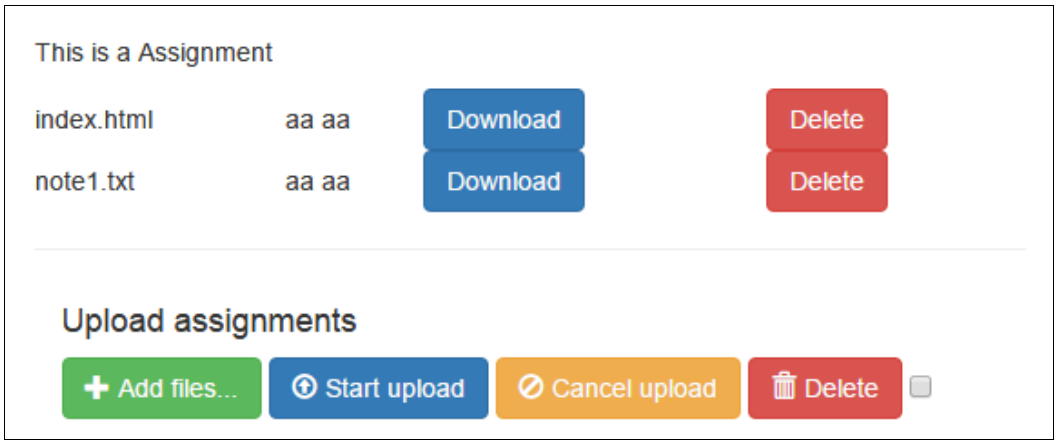

Figure 4. A screenshot of an assignment section for students. The upper part shows the uploaded files; The lower part shows a widget for students to upload assignments

A student's view of the assignment section contains two parts. The upper part is a summary or explanation of the assignment created by teachers, and a list of files that have already been uploaded by the user. The user can download or delete its own files by clicking on the corresponding buttons. The lower part of the student's view of the assignment section is an upload widget that can help students upload their files. The widget supports multiple file upload and drag & drop interaction.

3.3.4 Assignments section (teacher's view)

From a teacher's view of the assignment section, a list of submitted assignments can be found under the collapsible section of the article. Figure 5 shows an example of an assignment section.

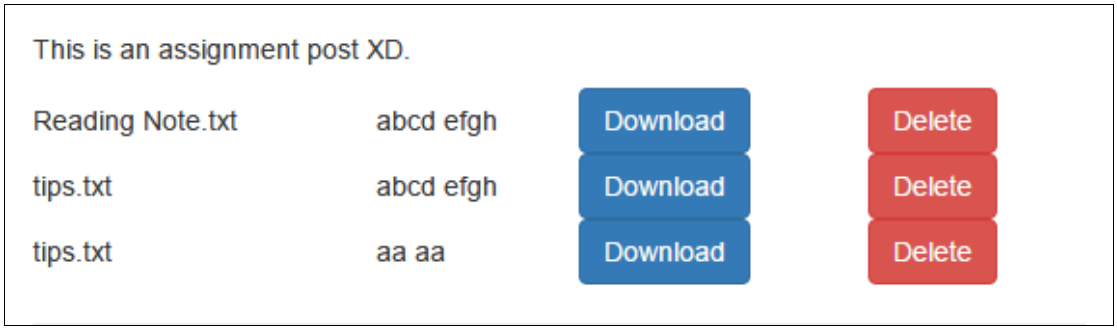

Figure 5. A screenshot of an assignment section for teachers. It shows a list of assignments that have been uploaded by all the students.

A teacher's view of the assignment section contains a message created by the teachers, usually a summary or explanation of the assignment, and a list of assignments that have already been uploaded by students. Teachers are able to see assignments uploaded by all students. They can download them or delete them by pressing the corresponding buttons.

3.3.5 Course page (teacher's view)

When a teacher wants to create a new article for the submission page of an assignment and detailed information about it, or he wants to create a new forum for discussion on a certain topic, he can use the create article widget shown in Figure 6.

Figure 6. A screenshot of the create article widget for teachers to create new articles.

The widget has a style which can be commonly found on websites like Twitter and Facebook. It appears at the top of the course page below the course summary section.

The create article widget is only available for teachers. Students will not see this widget on their webpage.

The content of the article can be edited in the input text area. The content supports HTML tags. After creating the article, the content of the article will be shown as HTML strings. This feature brings users a powerful ability to customize their content, but it also has security issues. Teachers are able to perform HTML injection, like adding advertisements or irrelevant log-in chat in their articles. Although there are security issues for this method, addressing them is beyond the scope of this project.

The style of the article is based on the type of the article. The type of the article will be determined by the selections provided by radio buttons on the create article widget. The default type of article is forum. A forum type article is an article that enables other users to reply or comment on the article. An assignment type article provides the function for students to submit their assignment. Teachers will be required to select the right type of article before they put them onto the website.

3.4 Implementations of Basic Features

At the programming level, basic features of our learning management system can be divided into different components. Each of them functions as an individual part, and contributes to the whole system. Those components are user identification, file management, and database design and usage.

The fundamental relationships between clients, the server, servlets and databases are shown in Figure 7. Each client goes through the server before it can communicate with the database. For instance, when clients want some data, they will send a request to the server, and the server will do the authentication. If it succeeds, it will retrieve the data from the database. This process gives more flexibility to the server side. It also makes the website more secure by preventing clients from directly accessing the database.

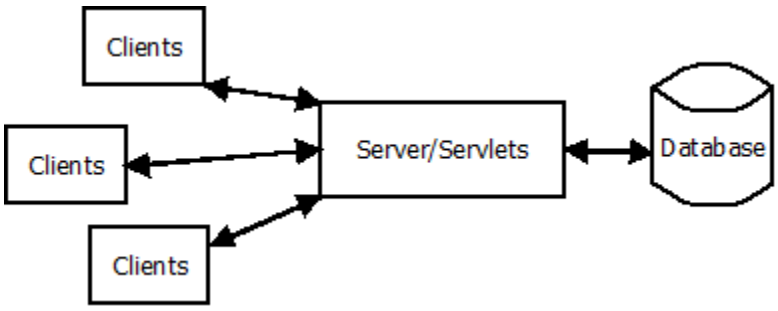

Figure 7. The relationships between clients, the server, servlets and databases in our implementation.

3.4.1 User Identification

The first page a user would see is the log-in page, shown in Figure 8. Users need to log-in or sign-up in order to maintain further usage of the system. The reason here is that all the information provided to the user is based on his or her profile. Teachers and students have different objectives when they use the system. Also, users involved

in different courses will see information based on their course selection.

Figure 8. Log in widget.

When a user presses the log-in button or signs up successfully, the system will record the user's log-in information and save it as a cookie of the website. To make it simple, we ignore the security problems that might be caused by cookies. For later access to the system, the browser will send the user ID as an additional parameter whenever it communicates with the server. The server, after receiving the information, will identify and classify the user ID. Then, it will give a response according to the user's request as well as the user information.

User information is saved in a MySQL database. Servlets set up in the server are responsible for retrieving user information when they receive a request from the client side. Servlets communicate with MySQL and send requests to it. After receiving data from MySQL, they process and forward that information to the client side.

3.4.2 File Management

Our learning management system has a file management component controlling file uploading and downloading as well as managing user privileges for files. On the server side, we use Tomcat and created several servlets to receive file requests. We also use MySQL as the database to save files, and to retrieve and maintain file information.

We used the open-source component "jQuery File Upload" [8] as a helper for downloading and uploading files on the client side. "jQuery File Upload" provides multiple file selection, drag and drop, a progress bar, and previews. A screenshot of the "jQuery File Upload" widget can be found in Figure 9.

There are two different widgets for file management. One is the file upload widget, which provides a user interface for the user to upload multiple files. Previews, names and size information are shown in the widget. Another one is the file manager. It can provide download and delete links and detailed information for files by sending a request with the user's name to the server.

The widgets communicate with our customized server when they need to modify or retrieve information from the server. When a user clicks on an upload or check button, the widget transforms the request into a JSON string and sends it to the servlets set up on our server side. Servlets will process the string, check user privilege, and manipulate the database according to the request. After the processing, they will give feedback to "jQuery File Upload", which will again transform the feedback and show it to the user.

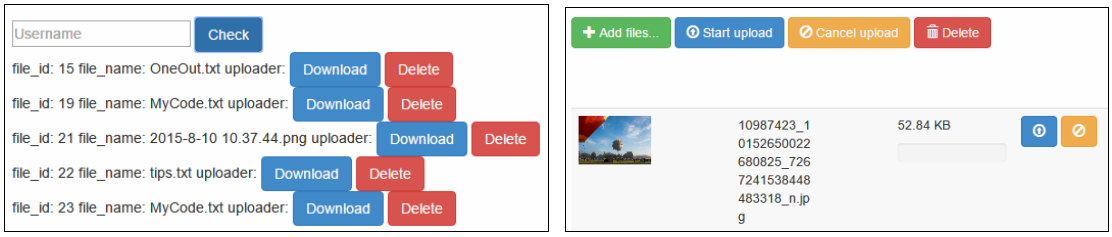

Figure 9. Left side: file manager checking all the files stored in the database. Right side: file upload widget trying to upload an image.

3.4.3 Database Design and Usage

The database is used in this project to manage data from user identification, file management, and forums. Each table in the database stores information for a different type of items, such as courses, files, users, articles, and posts. Tables also maintain relationships between each other according to their parent-child relationship. Most of the tables keep a reference to the users table to maintain information of the creator of an item. Some of the tables also keep IDs of their parents.

Courses, files, and users tables are used to store information for corresponding types of items. Articles and posts tables are used in the forums for posts and comments. Files, articles, and posts tables keep references to the users table to keep track of their creators. In the design, the courses table is the parent of the articles table, which is again the parent of the posts table.

The structure of the database is shown in Figure 10 below:

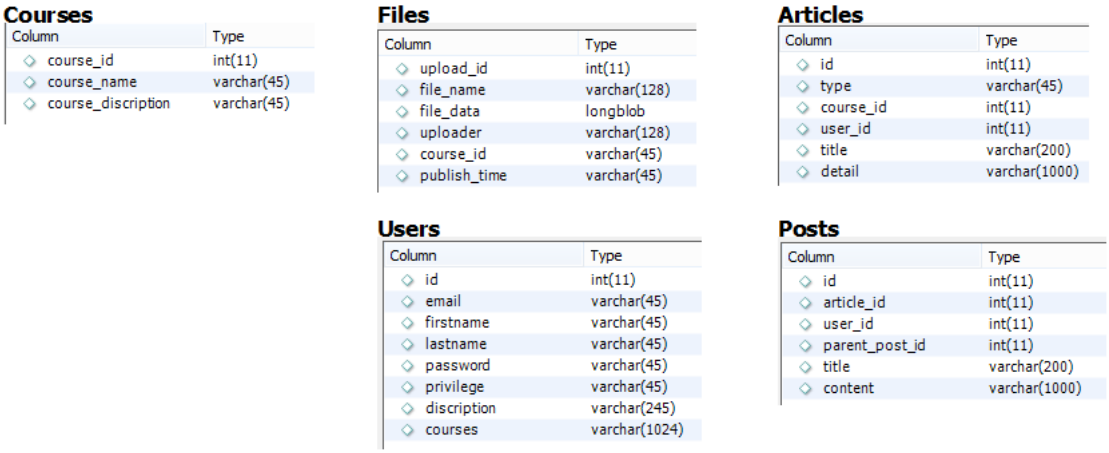

Figure 10. Tables of the database used in the project

When users try to open up a forum, all the posts and comments will be retrieved from the database and shown to the users. The data will first be assembled into a JSON format string with a tree structure and then be parsed to a JSON object on the client side. The tree structure has three layers in total. The root node of the tree is the information of a course. The tree has articles data as its children, and posts data as its leaves. After retrieving the JSON object, the client side will generate posts and comments in the forum by using the information in the JSON object.

4 Evaluation

At the time when this evaluation was written, we were at an early stage of developing the final user interface. We did not conduct user testing, because users would have been easily confused due to the very little content provided by the website. In this evaluation, we use Nielsen's heuristics [14] [15] to evaluate what we have done. We evaluate our system according to the guidelines provide by the evaluation method. We will discuss the advantages and disadvantages of our design.

Visibility of system status

The system has a navigation bar which can give a clue about users' current location. But the system does not have other mechanics to show user's current position and status clearly. Users might not know where they are when they browse the pages. The system gives feedback instantly when users interact with the web pages. When users create articles or comments, Ajax is used to dynamically modify the page contents.

Match between system and the real world

The system uses common terms like forums, assignments, posts, and comments. We do not use system-oriented terms, which would confuse users.

User control and freedom

The current system lacks the ability to modify content, and undo or redo an action. Uses do not have hull control of the created content. They also do not have the freedom to modify any content.

Consistency and standards

We do not use any specific standard for the system, but we keep the consistency of the words. We use the same words referring to the same situations or actions.

Error prevention

The system does not have an error prevention system currently. Users will not receive a confirmation dialogue before their commits to the system.

Recognition rather than recall

The system has a flat design structure. All the assignments and forums of the same course are on the same page. Users do not need to remember or recall what they were doing. Whenever users interact with a page, e.g., to upload an assignment or post a comment, they do not need to jump from one page to another, due to the usage of Ajax.

The system uses HTML5 to provide animation between transformations. For example, when a teacher creates a new article, the article will be instantly added on the page, and other articles on the same page will slide down gradually to give space to the new article. When users change the size of the browser window, the same effect

is applied to the whole page to adjust the size of different components. This also helps the users to identify what is happening.

Flexibility and efficiency of use

The flat design of the system makes the system easy to use for both experienced and inexperienced users. Articles and comments can be posted by performing a few clicks and users do not need to jump between pages.

The navigation bar of the system, created by using D3, is convenient for advanced users, but highly confusing to new users. Advanced users can jump to the content that they are looking for by clicking on the navigation bar. It saves their time by showing the structure of the page content. On the other hand, new users can find it quite confusing, because the structure of the navigation bar itself does not give any hint to the user about its usage, nor does it try to connect itself to a natural behaviour of the user.

Aesthetic and minimalist design

We do make collapsible widgets for comments of articles. Comments, reply and submit widgets will only appear when the user clicks on the expand buttons. Also, students and teachers have different sets of widgets, so irrelevant widgets regarding to different roles will not show up.

Help users recognize, diagnose, and recover from errors

The current system lacks the ability to modify content, and undo or redo an action. A teacher who has placed an incorrect assignment cannot delete it, nor could he edit the assignment to make it right. The same thing happens in other parts of the system, like when giving comments and creating articles. Hopefully, these features could be added in the next version of the system.

Help and documentation

We have not provided help or documentation for the system. But the system and design of the system are not too complex, so at the current stage, it might not be a problem.

5 Conclusions

In conclusion, we have developed a learning management system and performed experiments on the system to provide a better user interface.

First, we built a learning management system that has all the basic but necessary features for a learning management system, such as user privilege management, assignment management and forums. We implemented a database and used a client-server structure for the system.

During the later experiments with the user interface, we used HTML5, Ajax and Data-Driven documents. We developed a website with a flat design structure, which means that users do not need to jump between pages when they perform different

actions.

Looking back at our system and those learning management systems that are commonly used, we are surprised by how much a system relying on new technologies can make a difference. When we designed the structure of our system, we relied on Ajax to modify the web content dynamically. It comes naturally that by using Ajax, we tend to perform functions on the same page, which leads to a flat design structure. Without using Ajax, there would be many situations, where we need to either direct users to a new page or bring up a dialogue. Commonly used systems like Moodle and Blackboard use Ajax in some places, for example, when uploading assignments, but their structures remain the same because most of their content does not rely on Ajax.

6 References

[1] Beatty, Brian, and Connie Ulasewicz. "Faculty perspectives on moving from Blackboard to the Moodle learning management system." *TechTrends* 50.4 (2006): 36-45.

[2] Green, Kenneth C. "The 2010 campus computing survey." *Retrieved October* 6 (2010): 2012.

[3] Yueh, Hsiu-Ping, and Shihkuan Hsu. "Designing a learning management system to support instruction." *Communications of the ACM* 51.4 (2008): 59-63.

[4] Learning Management System (LMS) Evaluation 2011-2012 http://blogs.butler.edu/lms/files/2011/08/executive-summary.pdf *Retrieved on August 10 (2015)*

[5] Weaver, Debbi, Christine Spratt, and Chenicheri Sid Nair. "Academic and student use of a learning management system: Implications for quality."*Australasian Journal of Educational Technology* 24.1 (2008).

[6] 2012 – Changing Course: Ten Years of Tracking Online Education in the United States http://onlinelearningconsortium.org/survey_report/changing-course-ten-years-tracking-online-education-united-states/ *retrieved on August 10 (2015)*

[7] Grade Level: Tracking Online Education in the United States, 2014 http://onlinelearningconsortium.org/read/survey-reports-2014/ *retrieved on August 10 (2015)*

[8] blueimp/jQuery-File-Upload on GitHub https://github.com/blueimp/jQuery-File-Upload *retrieved on August 10 (2015)*

[9] Ellis, Ryann K. (2009), Field Guide to Learning Management Systems, ASTD Learning Circuits

[10] Paul Boag (2009-05-05). "10 Things To Consider When Choosing The Perfect CMS". SMASHING MAGAZINE. Archived from the original on 2009-05-05. Retrieved2014-07-07.

[11] D3 gallery https://github.com/mbostock/d3/wiki/Gallery *retrieved on August 10 (2015)*

[12] Data-Driven Documents http://d3js.org/ *retrieved on September 10 (2015)*

[13] Scalable Vector Graphics https://en.wikipedia.org/wiki/Scalable_Vector_Graphics *retrieved on September 10 (2015)*

[14] Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1–5 April), 249-256

[15] Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, Communications of the ACM 33, 3 (March), 338-348