Abstract – The road to effective science, technology, engineering and mathematics (STEM) instruction starts with a well-conceived and constructed plan or curriculum. STEM educators, who typically come from STEM backgrounds and have little or no STEM education training, can benefit from the use of an information technology (IT) tool that guides them through the complex task of designing an instructional module; i.e., a semester-long unit of teaching. The Instructional Module Development (IMoD) System is an example of one such tool. Currently under development, it is expected to provide users with a framework-informed by the scholarship on curriculum design-for conceptualizing and representing an instructional module, particularly in the areas of engineering and computing; and scaffold users through the design process. This paper presents the development of the client- and server-side of the web-based IMoD system, and discusses the user-centered design principles employed.

Index Terms- curriculum design, instructional module development tool, user interface, web application

I. INTRODUCTION

To ensure that future generations of engineering, science and other technological practitioners are equipped with the required knowledge and skills to continue to innovate solutions to solve societal challenges, courses or instructional modules (i-mods) that present relevant content using best pedagogical and assessment practices must be developed and delivered. Science Technology, Engineering and Mathematics (STEM) educators typically come from STEM backgrounds and have little or no STEM education training. Their approaches to learning, instruction, and assessment mimic the experiences they were exposed to as students and are not necessarily informed by scholarship in the area of how people learn. The road to effective STEM instruction starts with a well-conceived and constructed plan often referred to as the curriculum. An information technology (IT) tool that can guide STEM educators through the complex task complex task of i-mod curriculum development and provide relevant information about research-based pedagogical and assessment principles and how to implement them in their course will be of great value. Although IT tools such as, Electronic Performance Support Systems (EPSSs), Knowledge Management Systems (KMSs), and Repositories for reuse [1], have been used to support some aspects of the complex task of designing and developing curriculum, none of them currently provide this functionality.

A. Problem Statement:

Developing a well-conceived curriculum is a labor intensive process and is often only achieved after numerous iterations of teaching the course. The aim of the web-based Instructional Module Development (IMoD) system is to present a framework-informed by the scholarship on curriculum design- for conceptualizing and representing the i-mod, particularly in the areas of engineering and computing. This system will provide a scaffold via various help and support features to guide the users through the i-mod development process. The scope of the project presented in this paper is to build the client-side web user interface using user-centered design
principles such as: simplicity, visibility and consistency; and setting up a database at the backend. The expected outcome is a tool that provides a rich user experience and lessens the labor intensity typically involved in design of a curriculum.

II. RELATED WORK

The following section provides a review of 3 categories [1] of IT tools that currently support instructional curriculum design activities.

1) Electronic Performance Support System (EPSS):

EPSS is a category of software systems that helps users in improving performance. In Electronic Performance Support Systems [2], published in 1991, Gloria Gery defined EPSS as: “an integrated electronic environment that is available to and easily accessible by each employee and is structured to provide immediate, individualized on-line access to the full range of information, software, guidance, advice and assistance, data, images, tools, and assessment and monitoring systems to permit job performance with minimal support and intervention by others”. When applied to curriculum development, an EPSS performs tasks such as planning, drafting and designing curriculum materials, creating evaluation strategies. CASCADE-SEA is an example of an EPSS. CASCADE-SEA [1] stands for Computer ASSisted Curriculum Analysis, Design and Evaluation for Science Education in Africa. It is used to create teacher guides which offer commendable examples of how curriculum innovations can be implemented in practice, can be used by other users. This tool helps user to achieve the purpose of the material, analyze, design and evaluate the material. Another example of EPSS, is UbD [6] Exchange by ASCD (formerly the Association for Supervision and Curriculum Development). The tool’s main objective is to design curriculum, assessments, and instruction that help students to understand content more deeply. Using UbD exchange unit designers can create units using templates which are based on the backward design process, they can search existing units available in database, they can have review process by other Exchange subscribers and also they can get their unit reviews by Design experts.

2) Knowledge Management System (KMS):

KMS is an IT based program which manages the creation and retrieval of knowledge [3]. KMS can be document-based where technology such as Java, PHP etc. can be used to implement modules which can create/manage/share the knowledge. TeleTop [4] is a web-based environment which is used to design and deliver the courses. The purpose of this tool is to support faculty in planning and managing their courses and use telematics applications while teaching. Using Teletop, faculty can upload course materials and share it with students, can create assignments/quizzes and also can add learning goals. Blackboard [7] and Moodle [8] are also examples of KMS where instructors can create course modules and share it with students.

3) Repositories for Reuse:

Repository refers to the location of storage. Connexions [5] fall under the category of repositories where authors and instructors can create modules which are small knowledge chunks that contain educational materials, books, etc. so that students can access those modules to gain the knowledge.

Table 1 lists the categories of IT tools and corresponding links for existing curriculum development systems. IMoD spans over all three categories KMS, EPSS and Repositories.

Using IMoD, an instructor can design an i-mod with help of the step-by-step guidance provided by the tool (EPSS), manage, edit, evaluate, and administrate i-mods (KMS) and store, and share i-mods (Repositories).
Table 1: Classification of curriculum development tools

<table>
<thead>
<tr>
<th>Category</th>
<th>Tools</th>
<th>Reference link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UbD Exchange</td>
<td><a href="http://www.ubdexchange.org">http://www.ubdexchange.org</a></td>
</tr>
<tr>
<td>Knowledge management Systems</td>
<td>TELETOP</td>
<td><a href="http://knol.google.com/k/the-teletop-cms-as-a-research-tool#">http://knol.google.com/k/the-teletop-cms-as-a-research-tool#</a></td>
</tr>
<tr>
<td></td>
<td>Moodle</td>
<td><a href="http://moodle.org/">http://moodle.org/</a></td>
</tr>
<tr>
<td></td>
<td>Blackboard</td>
<td><a href="http://www.blackboard.com/">http://www.blackboard.com/</a></td>
</tr>
<tr>
<td>Repositories for reuse</td>
<td>Connexions</td>
<td><a href="http://cnx.org/content/">http://cnx.org/content/</a></td>
</tr>
<tr>
<td></td>
<td>GEM</td>
<td><a href="http://www.learningcommons.org/educator">http://www.learningcommons.org/educator</a></td>
</tr>
<tr>
<td></td>
<td>NEEDS</td>
<td><a href="http://www.needs.org/needs">http://www.needs.org/needs</a></td>
</tr>
</tbody>
</table>

III. IMPLEMENTATION

IMoD system requirements, system design and architecture, database design and implementation are discussed in this section.¹

A. System Requirements:

IMoD tool is a web-based semantic instructional module development tool where user can login, create profile, create, edit, save modules and load i-mods.

Figure 1 shows the use case diagram of the IMoD system.

Below is the list of the important functionalities of the IMoD system.

*Login* – To use the system a user has to create an account or if the user already has an account he/she has to login to system. Figure 2 shows the expected behavior when a user attempts to login to the IMoD system.

*Create i-mod* – User will login and create an i-mod by entering course details, instructor details, policies and prerequisite information. [12]

*Open i-mod* – Upon login into system user can see the existing i-mod created by him/her.

Figure 2: Flow Diagram for login process

*Edit i-mod* – User can edit the existing information of the i-mod and can save it to database.

*Generate syllabus* - User can use this functionality to create the course syllabus in pdf file format using all the course details specified in an i-mod.

¹ These system requirements, system design, architecture, database design and implementation are only related to the scope of the project presented in the paper. The IMoD system has other requirements which are not discussed here.
B. System Design

The IMoD system involves 3 main components as shown in figure 3, Web browser/client, Server and Database.

Web browser/client – User interface used by the user/instructor to interact with the system. User enters instructional module data and retrieves saved information using this module.

Server – Software module that responds to requests sent by the client and sends back results to client for display. Server is the intermediate communicator between client and database.

Database – Software module where data is saved.

User-centered Design Principles:

The following section gives a brief summary of the user-centered design principles that were used in the development of the user interface for the IMoD system and their implementation. [9]

Principle of metaphor:

Some of the user interface design ideas are borrowed from the existing popular web applications. The IMoD system’s toolbar as shown in figure 4 is borrowed from Gmail and Facebook where similar types of toolbars are provided to navigate the user to different pages of the applications. Also, the IMoD system’s welcome page as shown in figure 5- is inspired from the user interface of Prezi presentation software system [13] that displays a list of users existing presentations so that the user can navigate to them easily.
Principle of feature exposure:-

The user should clearly see what functions are available on the application. The menu or navigation options should be straight forward and should not be overloaded with information. They should be arranged in a logical manner. This is achieved by using straight forward navigation. As shown in figure 7, users can navigate easily using the toolbar. Also, help is provided for each content area on the application where user input is expected so that the user knows what type of data to enter.

Principle of coherence:-

The behavior of the program should be internally and externally consistent. The websites externally should be consistent with the websites from the same category and internally the content should be structured consistently. The panels in IMoD are designed considering their consistencies. The forms are arranged according to their categories.

Principle of help:-

User interface should provide necessary help to the user as and when they need it. As shown in figure 6, help is provided for each textbox in the IMoD system where user needs to enter data.

Principle of state visualization:-

Changes in the behavior should be reflected in the appearance of the program. As shown in figure 7, at any point of time in the IMoD system, the user can see which tab is activated via color-coding of the selected tab.

C. Technical Details:

This section explains the technical implementation details.

1) System Details:

a) Client – There are a multiple JavaScript frameworks available today for programming rich client-side interactions in web applications. With many different options, it is important to choose a framework that not only serves your current requirements but can also be compatible with long-term requirements. Based on the comparison between five JavaScript frameworks [10] as shown in table 2, ExtJS is the best suitable for implementing interactive User interfaces. ExtJS [11] is JavaScript library for building interactive web applications using techniques such as Asynchronous JavaScript [14] (Ajax), Dynamic HyperText Markup Language [15] (DHTML), and Document Object Model [16] (DOM) scripting. It is used to build user interface for rich web applications. The user interface for IMoD system is build using ExtJS framework.

b) Server/programming language – PHP 5.3.8 [17]

c) Database – MYSQL 5.5.16 [18]
2. Coding Details:

The IMoD system was built from scratch as part of this project. This involves design, implementation, and testing of the system. Total of 25 files were created which contains main functionality such as allow user to login, create i-mods, and load existing i-mods. Below is a list of the important code files and the functionality provided by the files:

1) **Welcome.php** – After a user logs into the system, the user is directed to the Welcome page where a user can see the i-mods created by him/her and can choose to create a new i-mod. It also contains JavaScript to render User Interface (UI) on the client-side that is implemented using ExtJS library functions. Table 3 shows the sample code to create a grid using ExtJS framework.

2) **Index.php** – This page has all the necessary JavaScript and PHP code to displays existing i-mod information.

3) **Register.php** – This page has all the necessary JavaScript and PHP code that allows a user to register in to the system and create new i-mods.

---

**Table 3: Sample code of ExtJS to create a grid [10]**

```javascript
Ext.require([  
    'Ext.grid.*',
    'Ext.data.*',
    'Ext.panel.*'
]);
Ext.onReady(function(){
    Ext.define('ImageModel', {  
        extend: 'Ext.data.Model',  
        fields: [  
            {name: 'name', type: 'string'},  
            {name: 'url', type: 'string'},  
            {name: 'size', type: 'float'},  
            {name: 'lastmod', type: 'date',  
                dateFormat: 'timestamp'}
        ]
    });
    var store = Ext.create('Ext.data.JsonStore', {
        model: 'ImageModel',  
        proxy: {
            type: 'ajax',  
            url: 'get-images.php',
        reader: {
            type: 'json',
            root: 'images'
        }
    });
    var listView = Ext.create('Ext.grid.Panel', {
        width: 425,
        height: 250,
        collapsible: true,
        title: 'Simple ListView <i>(0 items selected)</i>',
        renderTo: Ext.getBody(),
        store: store,
        multiSelect: true,
        viewConfig: {  
            emptyText: 'No images to display'    
        },
        columns: [{  
            text: 'File',
            dataIndex: 'name',
            flex: 50
        }, {  
            text: 'Last Modified',
            xtype: 'datecolumn',
            format: 'm-d h:i a',
            dataIndex: 'lastmod',
            flex: 35
        }]
    });
    store.load();
    var listView = Ext.create('Ext.grid.Panel', {  
        width: 425,
        height: 250,
        collapsible: true,
        title: 'Simple ListView <i>(0 items selected)</i>',
        renderTo: Ext.getBody(),
        store: store,
        multiSelect: true,
        viewConfig: {  
            emptyText: 'No images to display'    
        },
        columns: [{  
            text: 'File',
            flex: 50,
            dataIndex: 'name'
        }, {  
            text: 'Last Modified',
            xtype: 'datecolumn',
            format: 'm-d h:i a',
            flex: 35,
            dataIndex: 'lastmod'
        }]
    });
    // little bit of feedback
    listView.on('selectionchange', function(view, nodes){
        var l = nodes.length;
    });
});
```
4) newCourse.php – This page has all the necessary JavaScript and PHP code that allows a user to enter new i-mod information and save it to his/her profile.

5) saveCourseDetails.php – This page has all the necessary JavaScript and PHP code to save i-mod information entered by user into the database.

6) Logout/login.php – This page has all the necessary JavaScript and PHP code that allows a user to login and logout from the system.

7) JavaScript files - ExtJS is embedded in php files to render user interface to the user. Grids and form panels are used for this purpose.

3. Database Design

This section explains the database design implemented to store IMoD data. Database stores user information and i-mod data. There are total of 12 tables.

Below is the list of some of the important tables:

1) user – User information is saved in this table such as username, password, personal information.

2) IMoDContext – course module information is saved in this table. Below is the relation between user and IMoDContext tables.

3) Instructor – Instructor data is stored in this table.
4) **Imodinstructor** – i-mod ID and the corresponding instructor ID’s are saved in this table. Figure 8 shows the relationship between Instructor and Imodinstructor.

5) **CoursePrereqs** – Course prerequisites are stored in this table. Figure 9 shows the relationship between Imodcontext and Courseprereqs table.

6) **Policy category and policy table** – Figure 10 shows the relationship between Policy category, policy and IMoDContext table.

E. Deployment

This application is deployed on an Apache server on a windows machine. The path to the home page of the application is: [http://imod.poly.asu.edu/imod](http://imod.poly.asu.edu/imod).

IV. VERIFICATION AND TESTING

A. Verification:

The IMoD system was developed under the supervision of Dr. Srividya Bansal, Assistant Professor, CTI Department of Engineering at Arizona State University and Dr. Odesma Dalrymple, Assistant Professor, CTI Department of Engineering at Arizona State University. The tool was verified by Dr. Bansal and Dr. Dalrymple. The intent of verification was to confirm if the product of a given development phase satisfies the requirements imposed at the start of the phase.

B. Testing:

The IMoD application has been tested and found to meet all the requirements mentioned in the section (III-A). The application has been confirmed to work in multiple browsers like Mozilla Firefox 3.5, Internet Explorer 8, and Google Chrome 7.0. The functionality of the system is tested by registering the new user, creating new i-mod then performs operation on i-mod like edit, save and generate syllabus. APPENDIX A explains test cases in more detail.

V. SUMMARY AND FUTURE WORK

IMoD system helps to simplify the complex task of curriculum design and development. IMoD user interface is designed based on User-Centered design principles which helps user to complete complex tasks in a simple way and provides a rich user experience. This phase of the project included system architecture design, database design and creation of a simple user interface. While developing this project I learnt new technologies like ExtJS, PHP. It also helped me to improve my communication skills. This tool has very vast scope of future developments. Currently only Learning Context tab is developed where instructor can enter basic course details and can generate syllabus based on the contents of the Learning Context tab In the future, using this current architecture, some other functionality like creating assessments, learning objectives, content, and learning activities can be developed easily and back-end intelligence can be provided to scaffold the user through the design process. This tool can be given to end users,
e.g. faculty at ASU for evaluation and based on their feedback, further improvements can be made. Some of the possible areas of improvement are theme and/or color of the user interface. More impressive and attractive user interface themes can be implemented and integrated into the current system. Additional User-Centered design principles that can be implemented are - principle of shortcuts, where user should be able reach required information directly or indirectly; principle of focus, by adding attracting elements/images to the user interface; principle of safety, by using secure pages and handling confidential information securely; principle of user testing, by giving application to end user and improve the UI based on their feedback.

ACKNOWLEDGEMENT:

I would like to thank Dr. Srividya Bansal and Dr. Odesma Dalrymple for providing their valuable guidance and advice throughout the work and giving me the opportunity to work on this project where I got a chance to learn and apply new technologies. I also extend my thanks to Dr. Timothy Lindquist for serving as a committee member.

REFERENCES:

[5] “Connexions” Internet: www.cnx.org/content
[8] “Moodle” Internet: www.moodle.org/

APPENDIX A:

1) Instructor test case –
This test case is verify that an instructor is able to register & login to the system, create, edit, & save i-mods, and generates a syllabus.

2) Login – User can login to the system by entering username and password in login window as shown in below.

3) Welcome to IMoD system – User will be redirected to the welcome screen where user can see the existing IMoD created by him/her and also can create new IMoD.
4) **Create/Edit/Save i-mod** – Below is one of the panels from the IMoD system where the user can view/edit the details of the course.

Grid panel is used to add instructors.

![Course Details](image)

5) **Toolbar** – User can go to home page from the toolbar available on each page by clicking on the Home button. User can generate syllabus and also can change the background color of the web page.

![Toolbar](image)

6) **Edit Profile** - User can edit user information by clicking menu ‘Edit profile’ under username, as shown in above figure.

7) **Logout** - User can logout by clicking on menu ‘Logout’ under username, as shown in below.

![Logout](image)