6-6-2013

MUVE : a web tool to digitize books

Elysia Chu
Santa Clara University

Victoria Hall
Santa Clara University

Maya Hough
Santa Clara University

Urvashi Urvashi
Santa Clara University

Santa Clara University
DEPARTMENT of COMPUTER ENGINEERING

Date: June 5, 2013

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Elysia Chu, Victoria Hall, Maya Hough, and Urvashi Reddy

ENTITLED

MUVE: A Web Tool to Digitize Books

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

AND

BACHELOR OF SCIENCE IN WEB DESIGN AND ENGINEERING

THESIS ADVISOR

DEPARTMENT CHAIR

MUVE: A Web Tool to Digitize Books

by

Elysia Chu, Victoria Hall, Maya Hough, and Urvashi Reddy

SENIOR DESIGN PROJECT REPORT

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Engineering
and Bachelor of Science in Web Design & Engineering
School of Engineering
Santa Clara University

Santa Clara, California

June 6, 2013

Table of Contents

Acknowledgements

We would like to dedicate our work to the people who have helped us accomplish our goals. Professor Silvia Figueira and Radha Basu helped us to conceptualize the foundation of our system. Figueira helped to us to stay focused on the core functionalities of the site, while Basu consistently communicated with us and provided us with resources to complete our project. Through Basu, we connected with a technical team at Anudip and iMerit. We are grateful for their responsiveness and assistance in gathering research for our project. Lastly, we would like to thank the School of Engineering for their support and funding of our project.

Abstract

Many Santa Clara University students have the luxury of having access to devices such as iPads and eBook readers which support many applications that make our lives easier. The numerous ways to read digital books today has significantly lowered the cost of gaining access to educational material. However, in rural developing communities in countries like India, this is not the case. Currently in India, people in some developing communities have basic mobile phones called feature phones which can perform basic functionalities such as texting, calling, and browsing the web. Feature phones are affordable and very popular in developing communities. There is currently no set standard for reading ebooks on these phones and no applications specifically designed to view books on them. Through online books, people can have access to a wealth of information from which they can obtain educational resources. Anudip and iMerit, two organizations based in India that train and employ women to make online books, work with publishers who are readily digitizing books for expensive devices such as smart phones and eReaders. These companies currently do not have a tool that allows basic feature phones to gain access to online books. As a solution, we constructed a web tool customized for Anudip and iMerit which digitizes printed text into a format viewable on basic feature phones. Our web system, MUVE, successfully converts text into Extensible Markup Language, XML, a format understood by major browsers. While there are already existing efforts to bridge the education and employment gap in rural communities through companies such as biNu, which makes apps for feature phones and allows users to download books, there are no efforts that utilize XML, a much more readable, simple, and usable language that is much easier to interpret on feature phones. (Cave, 2013) For the final step of our tool, MUVE compresses the XML data into a file available for download. We took into consideration the importance of partitioning the data files into smaller pieces so that the information can be downloaded by users who have low bandwidth connections. The final output of our completed system is a folder of XML files containing the chapters of a book. This conversion tool will be delivered and installed; however, these files need to be filtered through a simple XML interpreter. The construction of this XML interpreter will happen simultaneously with the installation of our system on the Anudip and iMerit servers in India. Ultimately, our tool will hopefully grant better access to online books through mobile phones, thereby promoting better education, and will also provide more jobs for women through employment with Anudip and iMerit.

List of Tables and Figures

Section 2

Section 4

Section 9

Section 1 – Introduction

Employment and Education in the Developing World

Millions of people in the developing world often do not have access to fundamental resources such as education and employment. In many remote areas in underdeveloped countries like India, people do not have access to books and thus are denied the privilege to learn. This lack of education, among many other factors, contributes to the high unemployment rate in rural areas, which can make it difficult to sustain large families.

Due to financial and accessibility reasons, family members use the internet on their basic mobile phones to access information more often than purchasing printed educational materials. Through online books, individuals can gain education that they might not otherwise receive via textbooks in schools, which can often be outdated. However, these online books are not easily accessible for people in remote areas who face financial hardship because the few existing online books are not formatted correctly for their specific mobile devices. Companies such as biNu have been targeting these basic feature phones and creating means for its users to gain access to online books. (Cave, 2013) However, there currently is no streamlined and effective way for all feature phone users to access educational resources.

Secondly, many parties have proposed unemployment solutions by offering jobs outside the rural communities. According to the labour and employment minister Mallikarjun Kharge, the overall unemployment rate in India has been reduced from 8.3 percent (2004-05) to 6.6 percent (2009-10), despite issues in the global economy. ("India unemployment", 2012) However, analysts speculate these rates to be an inaccurate portrayal of joblessness in India, especially when considering female employment. Most of these jobs are unattainable to women in rural villages, due to their religion and social environment.

Our Solution: MUVE, A Web Tool to Digitize Books

We propose a solution that grants better access to online books and employment to women in rural villages in India. We will provide a web tool to digitize and format online books for the Anudip and iMerit Foundations. These non-profit organizations, based in India, offer local employment in the field of information technology to better the lives of marginalized women. Our solution consists of a website where employees of the Anudip and iMerit Foundation will be able to type text into a editor that will convert it into a readable format customized for their specific mobile devices. They will have the option to type, upload, or paste the text into the editor. Our tool will provide more jobs for women through employment with Anudip and iMerit Foundation. The now digital books will be made accessible for people throughout the remote areas in India on their basic mobile devices.

Objectives of MUVE

Our project required a multi-step process: research, design, construction, and implementation. Many factors required us to be efficient with our time and plan accordingly. This especially affected the earlier stages of our project development, causing the design phase to be the most critical and elaborate step in creating our system. Below is a step-by-step description of our creation process.

After solidifying the concept of our project with our sponsor organizations, Anudip and iMerit, we needed to investigate which kinds of phones are most commonly used in the developing communities. Once we researched the exact make and model, we had to investigate how to correctly format text, such as the chapters of a book, specific to these basic mobile phones.

Next, in order for the design phase of our project to begin, we initiated communications with the technical team at Anudip and iMerit to become familiar with their organization. Anudip and iMerit explained the details of their employee infrastructure, the organization of their database, and the technical specifics of their servers. With this information, we were able to begin designing our a customized system for their organizations.

Based on the structure of the employee hierarchy for these companies, we designed different levels of access for specific users--paralleling the management roles in the organizations. In order to accomplish this, we started by designing the basic high-level framework, then outlined the capabilities of each role, and finally designed the functional mock-ups of each page within the site. This user information, as well as mass content, needed to be stored in a highly organized way. Therefore, we carefully designed the database structure in order to efficiently manage the information.

Yet, the primary function of our site is to convert text. So, we enabled our site to allow users to copy and paste text into an editor, save, and then download that converted text. Since we anticipated the system to be used for numerous projects at a time, we included in our design functions to allow management of companies, users, and projects.

After creating highly-detailed prototypes and written plans, we began constructing our site and entered the development phase of our project. During this phase, we programmed to build the different functions for users, mentioned above, and developed the web software to convert the text into XML.

Lastly, we are documenting our system for installation and later use. This will enable those reading our manuals, read-me files, and setup guides to understand the code behind our site, as well as the technology involved in the conversion. Therefore, the code can be modified for further development. Additionally, the final result of the converted text from our tool must be processed, which requires a simple application to interpret the XML for the basic mobile phones for which the project is intended. This summer, during the development of this simple XML interpreter, our conversion tool will be installed on the Anudip and iMerit servers in India.

Section 2 – Research

Our Partners, Anudip and iMerit

Radha Basu is the director of Santa Clara University's Frugal Innovation Lab which focuses on developing sustainable technologies in underserved communities. She started Anudip, a non-profit based in India that trains rural women and youth in IT skills in places in and around Kolkata and in the state of Jarkhand. ("Anudip", 2013) IMerit, a sister company of Anudip, employs the women trained by Anudip and provides IT services for different clients. iMerit has relationships with publishers who are in need of digitizing their copyrighted books. They also have the employees who will type in text from these books into our tool MUVE to create digitized versions. Most of our research is based on primary research through interviews with employees of iMerit and Anudip.

Phones

Before we built our tool, we conducted field and user research, interface design research, as well as usability research. Through interviewing with Anindya Chattopadhyay, the Chief Delivery and Development Officer of iMerit, we learned that in the developing communities with which he works, the main mobile device used is a feature phone, a phone that "[occupies] the middle ground between basic phones that simply make, and receive calls and text messages, and smartphones". (Millar, 2013) It supports the most fundamental functionalities like listening to MP3 music, having the ability to run simple applications, texting, taking calls, viewing a calendar, and browsing the web. (Millar, 2013) According to Chattopadhyay, the specific make of the phones range from the brands Sony, Sony-Ericsson, ZTE, iNQ, Toshiba, Samsung, Spice, T-Mobile, Vodafone, and Nokia with specific models listed in Table 2.1. According to a BBC article, Gartner, a technology research company estimates worldwide feature phone sales totaled to approximately 264 million units in the 4th quarter of 2012, easily outstripping smartphone sales of around 207 million units. (Millar, 2013) Additionally, it was found through a data collection of 300m clickthroughs from mobile adverts that 77% of the Indian phone-using population are still using a feature phone. (Cave, 2013) Feature phones are more affordable and have gained significant popularity in developing communities and emerging economies. Currently, there are existing efforts in making books available on feature phones. Companies like biNu "makes apps for feature phones and has signed a partnership with Harlequin, making over 8700 titles available for purchase directly onto non-smartphones." (Cave, 2013) However, while there are already existing efforts to bridge the education and employment gap in rural communities through companies such biNu, there is currently no set XML standard for reading ebooks on these phones. XML is a markup language that allows for more simplicity, readability, and usability on the web.

Brand	**Model/Make**
Sony	Ericsson Xperia Arc S, Xperia acro S, Xperia go, Xperia ion, Xperia ion LTE, Xperia J, Xperia miro, Xperia neo L, Xperia S, Xperia SL, Xperia Sola, Xperia T, Xperia tipo, Xperia tipo dual, Xperia TL, Xperia U, Xperia V, Xperia Z, Xperia ZL
Sony-Ericsson	Xperia active, Xperia mini, XPERIA neo V, Xperia Play 4G, Xperia Ray, A8i, Live with Walkman, W8, WT18i, x10 mini, x10 mini Pro, Xperia Arc, Xperia Arc S, Xperia mini pro, Xperia Neo, Xperia neo V, Xperia PLAY, Xperia PLAY CDMA, Xperia pro, Xperia ray, Xperia X10, Xperia X10 mini, Xperia X10 mini pro, Xperia X8
Spice	Mi-270, Mi-300, Mi-310, Mi-350, Mi-410, Mi-720
T-Mobile	Arizona, Comet, G1, G2, G2 Touch, G2x, Garminfone, Move, Move Balance, myTouch,

	MyTouch 3G, myTouch 3G Fender Edition, myTouch 3G Slide, myTouch 4G, myTouch 4G Slide, myTouch Q, Pulse, Pulse Mini, Sidekick 4G, Vivacity
Toshiba	REGZA Phone IS04, REGZA Phone T-01C
Vodafone	845, 858 Smart, 945
ZTE	Mobile Concord, Amigo, Anthem 4G, Avid 4G, BLADE, FTV Phone, Fury, Groove, Libra, Merit, Racer, Racer II, Score, Score M, Skate, Sprint Flash, U880, U900, V821, V9, Warp
iNQ	Cloud Touch
Samsung	Galaxy Y S5360 , Galaxy Y Duos , Galaxy Ace S5830 , Galaxy Mini S5570 , Galaxy Ace Plus S7500 , Galaxy S II Epic 4G Touch , Galaxy W I8150 , Galaxy S 4G , Galaxy Gio S5660 , Galaxy S II 4G
Nokia	N8-00, E7-00, X7, C7, C6, E6 and S60v5 Touch (Nokia X6-00, Nokia C6-00, Nokia C5-03, Nokia 5800 XpressMusic, Nokia5530 XpressMusic, Nokia 5250, Nokia 5230 Nuron, Nokia N97.

Table 2.1: Brand, Make, Model of Mobile Devices

Target Users - Women of Anudip

Through working with Radha and her partners, we not only learned that feature phones are very commonly used in rural India, but also that employment among women is low. Therefore, the target users of our tool are the women trained by Anudip and hired by iMerit.

Last July, the Indian Labour Bureau conducted an employment survey which revealed that “while official unemployment in India is lower than in the developed world, female labour force participation rates are extremely low.” The Bureau's survey found that only 17.9 percent of women were working, as opposed to the 73.3 percent of employed males in both rural and urban communities. (Shrinivasan, 2012).

Through our tool, we want to promote education in the rural communities and villages in India while creating more jobs for these women who are trained in IT skills by Anudip and employed by iMerit. Ultimately, our tool will allow users to have a more streamlined and accessible way to view online books through mobile phones, thereby promoting better education. It will also provide more job opportunities for women through employment with Anudip and iMerit, therefore helping them better sustain their lives and families.

Information Organization

The system will be uploaded and used on iMerit’s servers therefore details about iMerit and Anudip’s information organization are crucial to the success and compatibility of our system Through further communication with Anindya, we gathered information on their database, server configuration details, operating systems in use, and the web browser versions. It was found that the version of mySQL database is 5.1.47, that they use PHP4, and that they use all the major browsers Firefox, Chrome, and Internet Explorer. Below is a list of some of the important information we were given:

Server configuration details:
Server IP
Server OS and the kernel version numbers
Database version and Database Server version
HTTP Server and the server version

Services access credential:

Information for File Transfer Protocol (FTP):

User ID

Password

MySql Database:

Database

Password

Other:

Browsers and respective versions

Screen resolution

EMP data table from its PostgreSQL server

HTTP path

MySQL UID

Section 3 – System Requirements and Technology

Requirements & Design Constraints

There are two types of requirements our system satisfies: functional and nonfunctional. Functional requirements define what must be done and can be answered or evaluated as true or false. Non-functional requirements define the manner in which the functional requirements need to be achieved. They are usually answered by a degree of satisfaction. Below is our breakdown of the two types of requirements.

Functional

The system will:

- Be a web-based system
- Be hosted online in a secure format (https).
- Work on low bandwidth connection
- Permit four kinds of users: Admin, Quality Lead, Project Lead, and Editor
- Convert text into a format that will make the text viewable on specific "functional" phones
- Allow users to download documents to desktop
- Store uploaded documents on a server in folders organized by companies and projects
- Permit a search feature limited by role for each user
- Upon login, display a welcome message that includes the user's name

Non-Functional

The system will be:

- Easy to use for all Anudip and iMerit users
- Easy to maintain
- Secure: it will have certain security features such as protection from DB attacks
- Scalable for large numbers of users, projects, and data

- We will strive for security in all areas of our website
- Easy to set-up folder organization
- Simple to install on the Anudip servers
- Enable employees to easily manage previous work
- Easy to login
- Friendly for users unfamiliar with computer and online technologies

Design Constraints

The system will use:

- The Anudip Foundation server platform
- Technology that enables the use of all major browsers: Chrome, Internet Explorer, Safari, and Mozilla Firefox
- A client-server model
- Anudip employee usernames for login

Technologies Used

Programming Languages & Database

- XHTML
- CSS3
- PHP4
- SQL for MySQL

Major Functionality

- XML

Software & Applications

- ADOBE Illustrator
- ADOBE Fireworks

Design Rationale

System

Some engineers focus on merely the elegance of their visual design and usability in order to sell in a competitive consumer market, especially like that of the western world. However, we are not catering to such monetary markets nor attempting to attract an influx of users or customers. Additionally, we are not solely designing for expert engineers or for a predominantly technical audience who have higher standards of efficiency and functionality. Instead, we are designing for a specific pool of users -- employees of Anudip -- who will need to use the system everyday. Therefore, we focused our efforts towards the design phase and maintaining a simple interface with a small number of core functionalities. Specifically, the login page and text editor had to be simple in order to prevent users from being confused by superfluous design elements and functionalities that are typically seen on other mainstream web pages. Our site utilizes international symbols and uses easily translatable text wherever possible since this system will be used by non-native English speakers.

Lastly, we maintained a balance between core functionality and aesthetics in order for our program to be successful. At Anudip and iMerit, companies and organizations will sponsor the digitization of various books, articles, magazines, and other materials. For example, a company called HoughBooks may have three textbooks. HoughBooks, therefore, indicates the "company," and its three textbooks are classified as "projects." This nomenclature falls into a hierarchical organization of with the other files stored on the server, as well as provides security and access-specification for projects. Lastly, we anticipated a high influx of projects under any one company requesting services from Anudip, so the hierarchical organization also provided the system scalability.

Users

Based on the employee structure of Anudip and iMerit, we have created four types of users including Admin, Project Lead, Quality Lead, and Editor. For security and managerial purposes, these user types have different capabilities and specified visibilities of the projects within the system.

Technology

The technologies we used to develop our system can be divided into two categories: Programming Languages & Database, Major Functionality, and Softwares & Applications.

Programming Languages & Database

The programming languages that we used to create our system were XHTML, CSS3, SQL for MySQl, and PHP4. We used XHTML to structure the pages in our system because it is stricter than HTML4 and fully supported by all major browsers. CSS3 was used to style our pages since it is the latest standard of CSS and is also supported by all major browsers. SQL is commonly used to create and access a database. We used SQL to design a mock database for the Anduip and iMerit technical team to mirror when installing our system. We specifically used SQL for MySQL since this is the version some of the Anudip and iMerit servers are using. Although the most recent version of PHP is PHP5, we designed our system using PHP4 since this is the language with which some of the Anudip and iMerite servers are operating.

Major Functionality

The major functionality of our system is to convert text into a format to make the text viewable on a basic feature phone. To the best of our knowledge, the most common basic feature phone in developing world is a Nokia mobile phone. Displaying text onto these phones required us to use XML which allowed us to convert the text into readable format viewable on most major web browsers.

Software & Applications

Since designing the interface was a critical component of the success of our system, we needed the appropriate prototyping tools to help visualize the placement of all the functionalities. We used ADOBE Fireworks and utilized its “hot spot” feature by creating a clickable mockup. Additionally, we used ADOBE Illustrator to create the icons and graphics throughout our site.

Section 4 – Design and Construction

User Interface

One of the primary steps in the design process is planning the basic, high-level framework for the site. Our team centered the basic structure of MUVE around the most important function of the site: to convert digital text into a format viewable on feature phones. We outlined primary tasks that users would perform, using our tool. Below in Figure 4.1 is a preliminary activity diagram, which shows some of the dynamic behavior of the system.

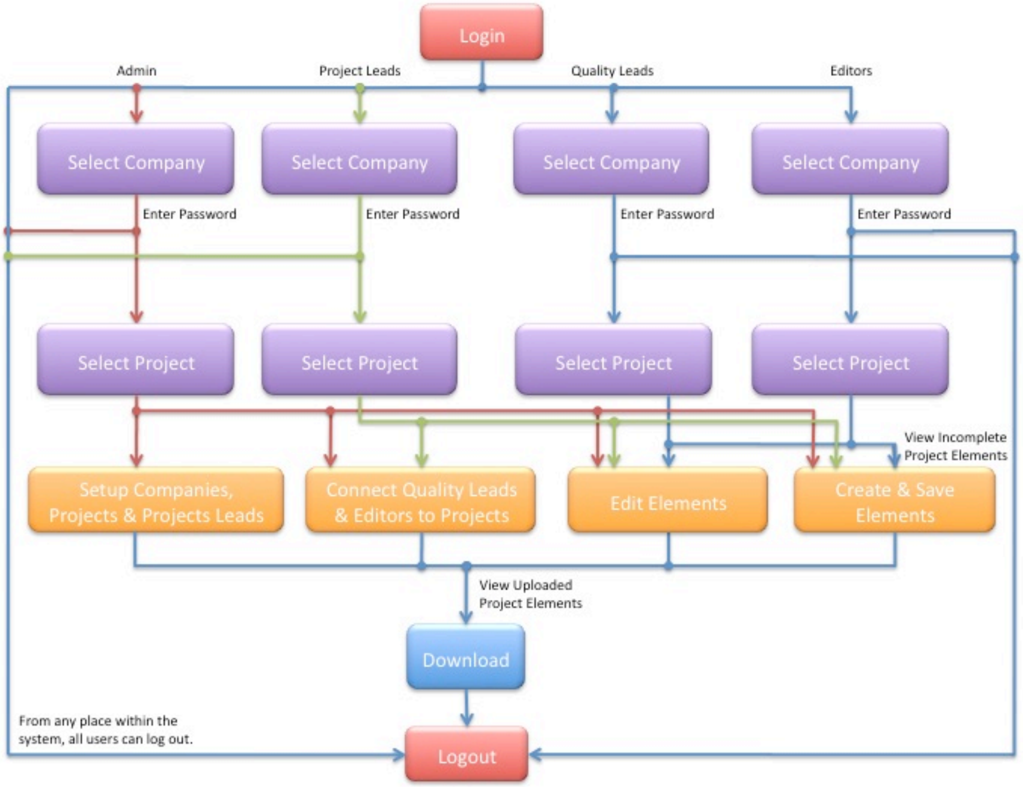
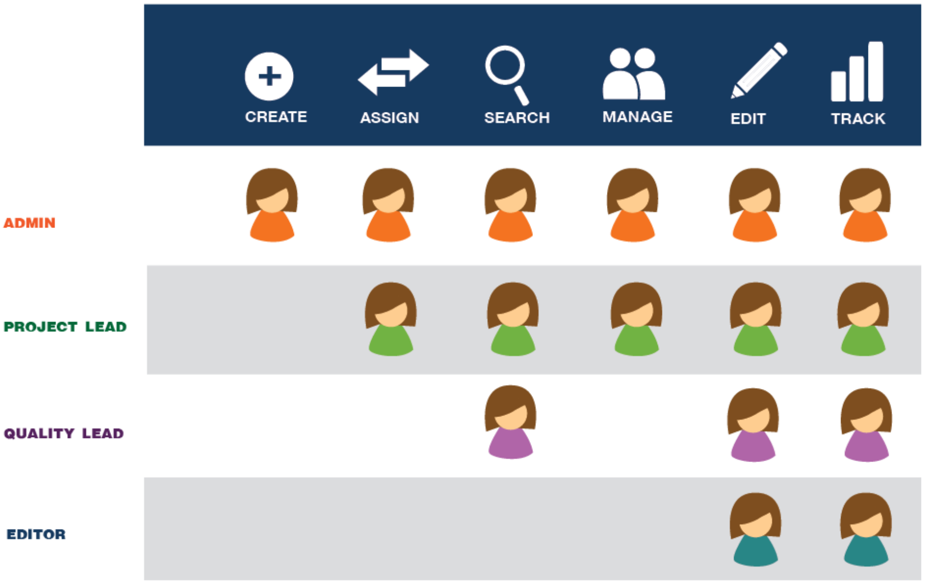


Figure 4.1: Activity Diagram

Further along in our design process, we organized the necessary tasks into six modules, described below, as well as the predominant users of the system.

The requirement that influenced our system design the most was the difference in the kinds of people working with our tool. Based on the structure of iMerit employee management, we have four users roles within our system: Admin, Project Lead, Quality Lead, and Editor. In Figure 4.2, we have illustrated the capabilities of each role by showing which modules they can access.



	CREATE	ASSIGN	SEARCH	MANAGE	EDIT	TRACK
ADMIN	●	●	●	●	●	●
PROJECT LEAD		●	●	●	●	●
QUALITY LEAD			●		●	●
EDITOR					●	●

Figure 4.2: User Capabilities

These modules, or major actions, include creating projects, assigning employees to projects, searching, managing users, editing projects, and tracking progress. We designed the roles to grant the admin full capability and access to every module within the site. As shown in Figure 4.2, the admin has all six major functionalities. Below is a Use Case diagram for the administrator, describing how the admin could move through the tool.

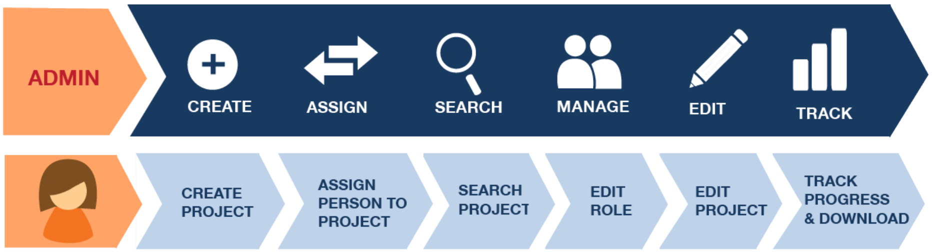


Figure 4.3: Administrator Use Case

Once an admin has a piece of text or a book to digitize, they need to create a project within our system, that will contain information about that book, such as the author and title, as well as the actual text. Then, he or she can assign people to that project to carry out the work necessary for completion. She can use the search module to view a basic overview of the users, projects, and companies. The admin can also manage the users in the system; for example, the admin can promote an Editor to Quality Lead. A user acting as admin can also edit a project's chapter text, track that project's progress, and download the final file of converted text. The reason we found it important for users acting as administrators to have full access to the modules is so they can take a project from start to finish singlehandedly. This is in the case of smaller projects, for example, where iMerit will only need to employ one individual to digitize a shorter book. The roles are designed hierarchically; so, the nearer the role is to the top of the hierarchy, shown at the top of the diagram in Figure 4.2, more inclusive its functionalities. As we implemented the user interface of our system, we came to the realization that we needed to adjust the page layouts slightly for each user accessing our system. We gave certain functionalities to some users but not others, for management and organizational purposes. As a result, we had to compensate for the empty spaces in a user's interface, especially when we designed for the editor's minimal functionality. Most importantly, the four user roles and their capabilities were designed to parallel the organizations for whom we are designing. We considered even the smallest task differences in order to verify that MUVE's design would imitate the iMerit employee management.

Conceptual

Having four different users of our web system meant that we needed to prioritize site organization in order to check for gaps in the user interfaces. In order to do this, we designed functional mock-ups of each page within the site. To design the prototype, we used an application called Fireworks. This is a tool from Adobe that has the ability to create "hotspots," which are links between specified pages you created. Using this prototype enabled us to navigate the system as a user and to easily discover missing steps, identify poor connections, and assess the general navigation logic. Pictured in the screenshots below are images from the Fireworks prototype. Here, the interface is very basic, as the prototype exists primarily for functionality purposes. The prototype also helped us arrange the pages' major elements in a logical way, so as to maximize the ease at which users navigate through the system. Creating a highly developed prototype, in which we structured the layout of the elements on every single page, empowered us to be more creative during the construction phase. The front-end development and planning--the user role explanations, the Fireworks mock-up--gave our team the ability to control all aspects of the user interface. The conceptual model is a rough representation of our system user interface. Our program is designed to be an intuitive interface for non-native English speakers, so we have chosen to minimize the stylistic attributes of our program in order to create a simple, uncluttered layout. The software will display the appropriate user experience based on the four roles outlined in the User Interface portion of this report: Admin, Project Lead, Quality Lead, and Editor. Below are a few images for the conceptual model of the site.

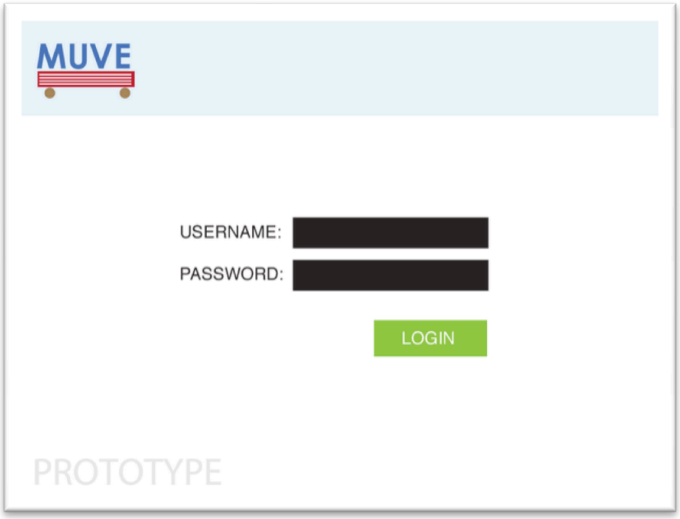


Figure 4.4: Login Screen

Here, Figure 4.4 depicts the login screen displayed to all users when they arrive on the site. If users are unsuccessful in logging into the system or do not have access to other pages within the site, they will be redirected to this page. The login screen also appears when the user logs off the system. Users type in their credentials and log into the system.

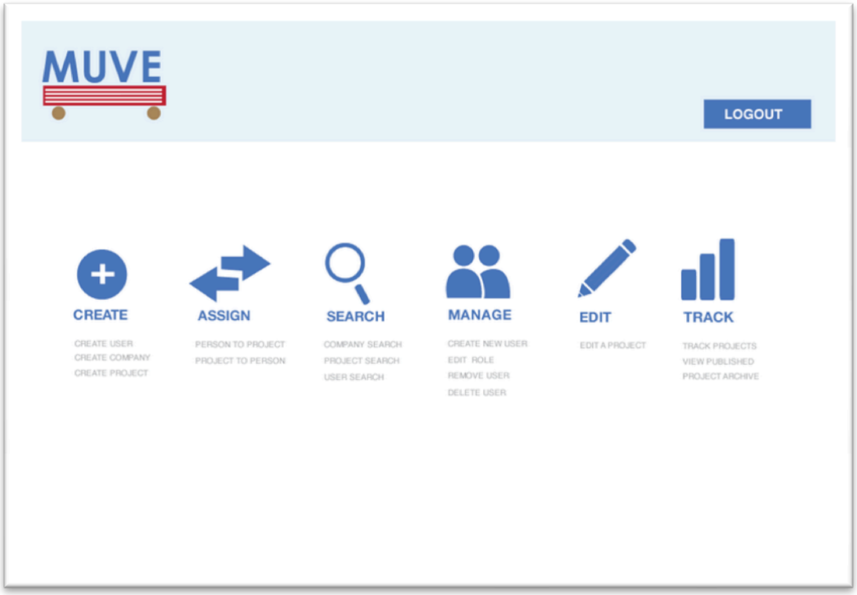
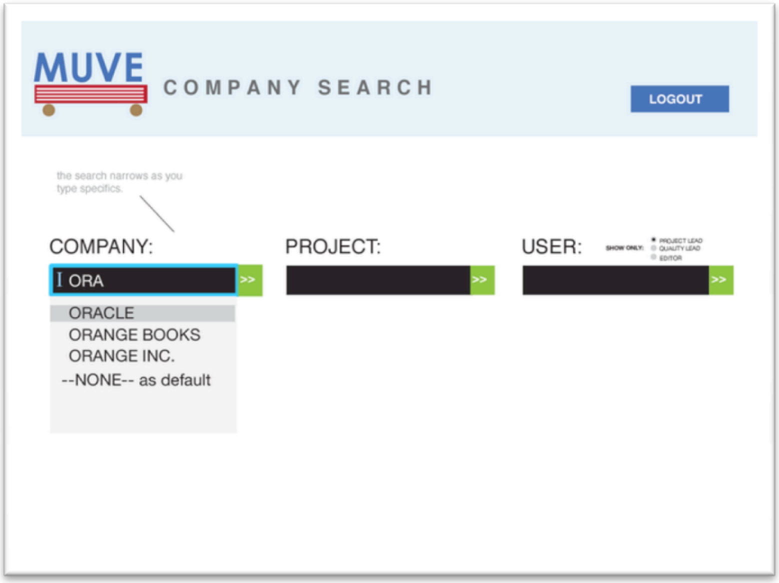


Figure 4.5: Splash Screen

Upon logging into the system, users will see the six different modules: Create, Assign, Search, Manage, Edit, and Track. Within each module shown in Figure 4.5, users can access more functions.



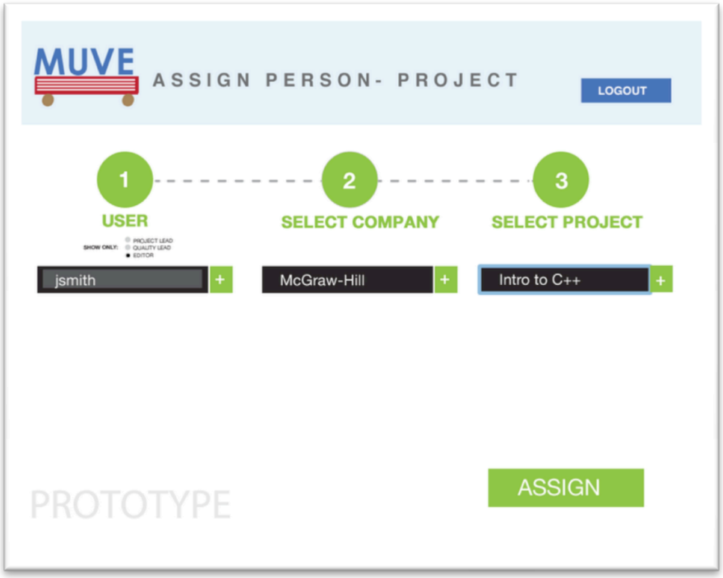
COMPANY: ORA

PROJECT:

USER: SHOW ONLY: ● PROJECT LEAD ○ QUALITY LEAD ○ EDITOR

Figure 4.6: Company, Project, and User Search Screen

If a user selects the “Search” module, they can search for a company, project, and user, as shown in Figure 4.6.



SHOW ONLY: ○ PROJECT LEAD ○ QUALITY LEAD ● EDITOR

jsmith

McGraw-Hill

Intro to C++

Figure 4.7: Assign Person to Project Screen

Once users, companies, and projects have been created, the Admin can assign a person to a project on a page similar to Figure 4.7.

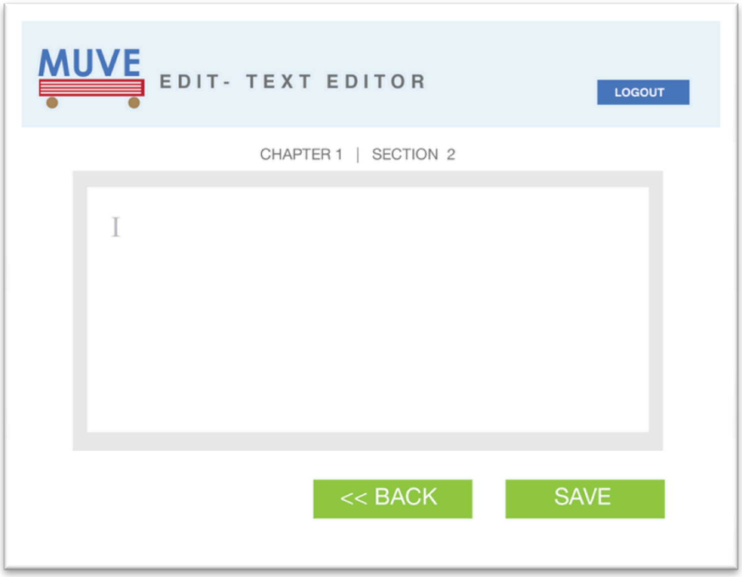
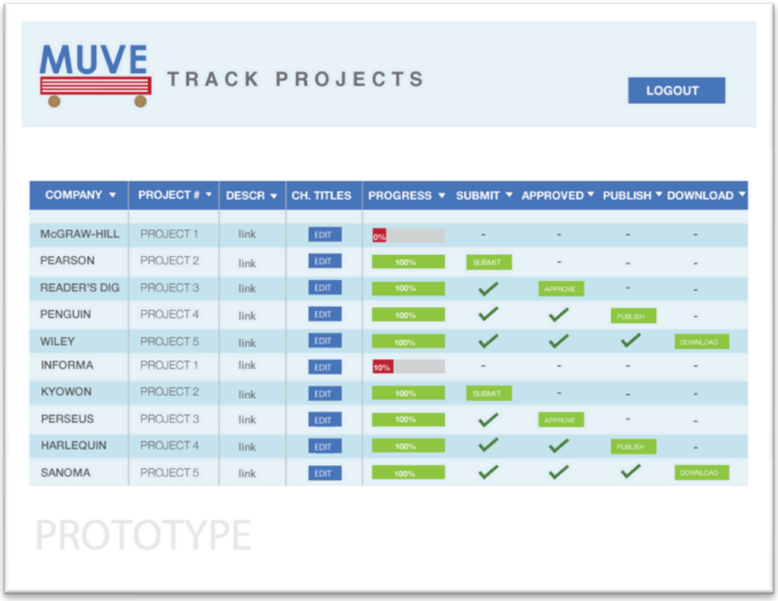


Figure 4.8: Text Editor Screen

As Figure 4.8 shows, once a user is assigned to a project, he or she can edit the chapter text of the project using the text editor.



COMPANY ▾	PROJECT # ▾	DESCR ▾	CH. TITLES	PROGRESS ▾	SUBMIT ▾	APPROVED ▾	PUBLISH ▾	DOWNLOAD ▾
McGRAW-HILL	PROJECT 1	link	EDIT	0%	-	-	-	-
PEARSON	PROJECT 2	link	EDIT	100%	SUBMIT	-	-	-
READER'S DIG	PROJECT 3	link	EDIT	100%	✓	APPROVE	-	-
PENGUIN	PROJECT 4	link	EDIT	100%	✓	✓	PUBLISH	-
WILEY	PROJECT 5	link	EDIT	100%	✓	✓	✓	DOWNLOAD
INFORMA	PROJECT 1	link	EDIT	10%	-	-	-	-
KYOWON	PROJECT 2	link	EDIT	100%	SUBMIT	-	-	-
PERSEUS	PROJECT 3	link	EDIT	100%	✓	APPROVE	-	-
HARLEQUIN	PROJECT 4	link	EDIT	100%	✓	✓	PUBLISH	-
SANOMA	PROJECT 5	link	EDIT	100%	✓	✓	✓	DOWNLOAD

Figure 4.9: Track Page

The Track page shown in Figure 4.9 allows users to track all the projects being worked on in the system. Users can view the company, project code, project description, chapter titles, and the different phases of development of each project.

When any user wishes to log out, he/she clicks the “Sign Out” button available on the top right hand corner of every screen (underneath where his or her name will appear). The Admin has access to all functionality and screens available to the Project Lead, Quality Lead, and Editor.

Information Management

All of the user data, as well as the functionality for every page, needed to be stored in an organized fashion. Moreover, the entire information management aspect of the system is important in order to efficiently access and store the data. So, we used a database to store information on the server; this allows for the functionality to log into the server, and store information pertaining to projects in an organized way. Below, Figure 4.10 is a model of how data is transformed as it moves through our system, to demonstrate how clients access the database.

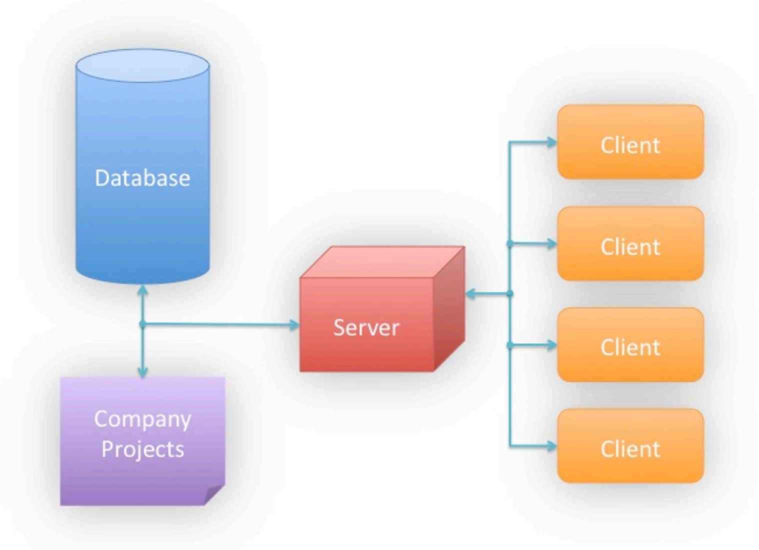


Figure 4.10: Architectural Diagram

In the database, we 'chunked' information into useful parts; for example, we generally put user information into separate tables than the project information. In Figure 4.11, the database tables, relationships, and attributes are shown in an Entity Relationship Diagram. Specifically, we organized the database tables to allow for scalability and ease of access. We sectioned the data by the kind of "object" it referred to, namely, chapter, project, and user. We gave each 'object' attributes; shown in Figure 4.11, users in the user table are assigned a user id and their user name is stored in the database. We connected elements using relationship tables and related unique ids; this allows for non-repeated relations between elements. Our system was designed to enable easy access of the database through PHP to take information stored in the system and display it to the users.

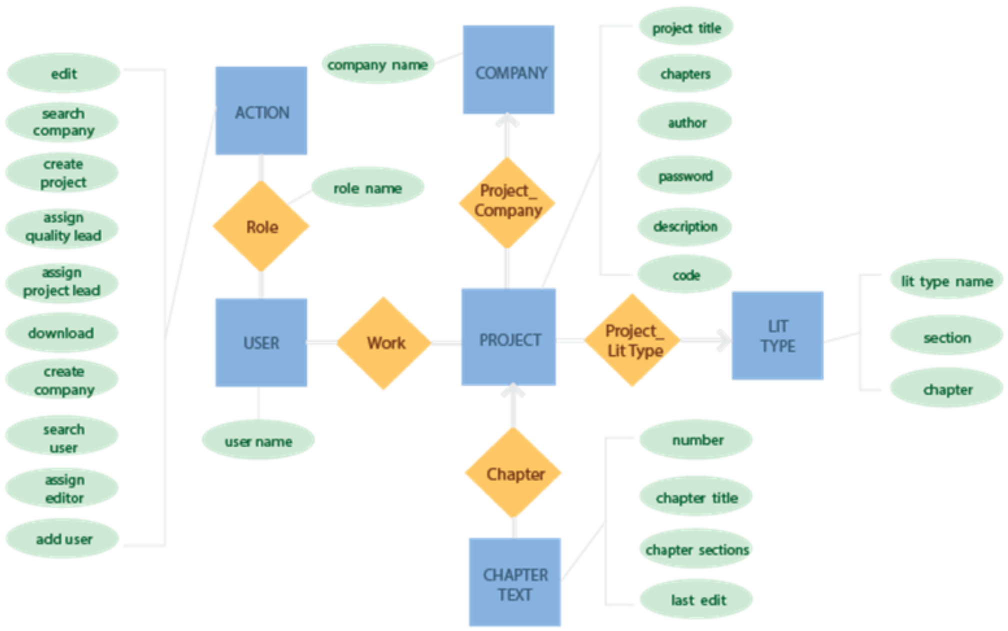


Figure 4.11: Entity Relationship Diagram

Generally, we designed the back-end of our site in three stages, which we titled: database, "filter," and "display" functions. We used few technologies to develop during these stages. (For examples of our system code, see the appendices section of this report). In general, most of these functions return an array containing a "Boolean" variable and a message. The booleans indicate with a zero or one value whether the function called was a general failure or a success. The message in the array is either another array of relevant information, or one or more error messages, depending upon the function's success. All of these functions are written in PHP, using MySQL, and most of the time the reasons for failure are: receiving improper input, having a problem connecting to the database, or gathering an unexpected result.

The database functions access the database and pull information to be used in other functions within the system code. The filter functions call database functions, process that information, and sort that data into the *format* needed for a specific kind of page element. The display functions use filter functions to get only the information needed for that particular page. The display code outputs that data -- and using CSS, with ids and classes -- these functions display the information in a neat, orderly way to the final front-end of the system. Table 4.12 below outlines the differences between the function types that comprise the backbone of our system.

Step	Kind	Attributes
1	Database Functions	Closest to the server/database
		Use built in MySQL functions in PHP to perform operations on the database (INSERT, UPDATE, SELECT)
		Query placed into an associative array and returned
		Called by filter functions
2	Filter Functions	Act as "in-between" functions; managing information passing between the database and the front-end
		Built in order to be page/function - specific
		Call one or more database functions to get information
		Perform any necessary calculations on the database information
		Sort the relevant information into an array which is returned to the caller display function
		Called by display functions
3	Display Functions	Closest to the user interface (UI)
		Called on the pages themselves
		Receive relevant information from filter functions
		Echo/print out the information to the screen
		Called on the pages themselves
		Use ids and classes to structure the CSS style sheet to make the webpage readable and user friendly

Table 4.12: Database to Interface Information Management

In summary, the user interface is created by the display functions, which use filter functions, which call database functions, which access and pull information from the database. We used a 'bottom-up' design methodology, meaning, we broke the system functionalities into smaller pieces, then placed the pieces on top of each other.

Conversion

The most important aspect of our site is the textual conversion to XML; however, the process did not require a highly complex design. We designed our system to enable users to simply paste text into an editor, save, and download that converted text. The back-end is more complicated, and organizes the information logically for viewing on feature phones. We chose to have our final output as XML for several reasons, including the facts that XML is universal and readable on browsers. All of the information for the projects is "chunked" and stored in the database in plain text, so the XML conversion happens upon download. XML is only created upon clicking the download button. At this step, the information is pulled from the database, translated into XML using PHP functions, written to files, compressed in a ZIP file, and finally downloaded to the browser. To prevent overwriting of old files, we made the system make a new file title every time the download button is clicked, for version control purposes.

We "chunked" the information because we anticipated the files working with feature phones running on bandwidth connections that are too low for large downloads and low memory. Due to the inability to predict who and where the digitized books will be distributed, we leave some of the "splicing" / "chunking" of information specifics up to the computer engineer managing and maintaining our system. These splicing decisions can be re-made later depending on results from further testing to determine how to be most efficient.

In order to convert the information into an organized folder of XML files, we designed different kinds of files to hold specific information. Our system creates a folder that contains two kinds of XML files: a "Shelf" file, and a "Chapter" file. The "Shelf" file contains general information about the project, while the chapter file holds information about the individual sections of the book. This conversion process is shown below in Figure 4.13.

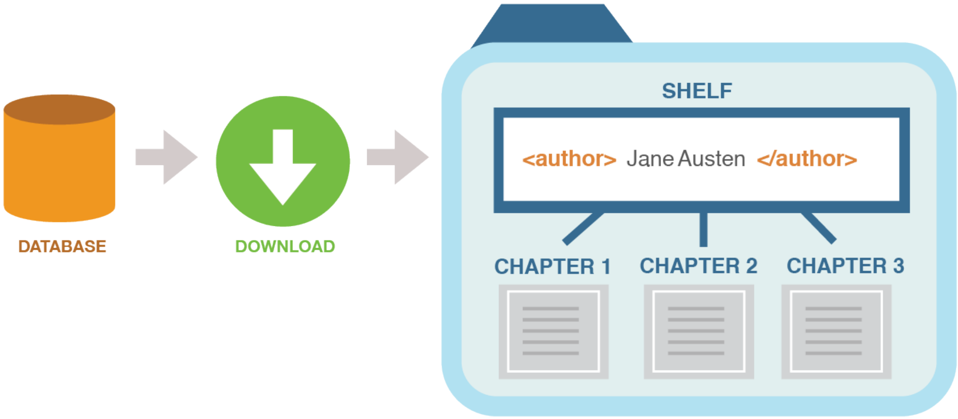


Figure 4.13: Conversion Process Flow

Both kinds of files, the Shelf files and Chapter files, files may be adapted to hold more information later. Specifically, we designed the Shelf file to be more of a "master" file, meaning that it can manage the storage of Chapter files corresponding to multiple projects.

The process for the conversion involves pulling information from the database, modifying it, compressing it, and making it available for download. Based on how we structured our system for the users, all information pertaining to a project that was entered by a user is stored in the database. For the conversion, our system uses PHP functions to pull all the information associated with one project. Then, it translates the text into XML in a series of steps. The tool uses information pulled from php functions to write a string for a file with two functions, titled, "write_shelf()" and "write_chapter()." These functions are simple parsers; they take in information and write a long string with proper XML tags. A temporary project folder is created on the server, in which the PHP files are written; then, the file is compressed in the ZIP format. Finally, the system uses PHP to force a download of the ZIP file to the computer.

Section 5 – Testing, Documentation, and Installation

Testing

Test Plan

Testing was continuously performed on our system during the design and development phases of our project. Testing during the design phase was strictly white box focused, while we shifted our focus to black box testing during the development phase of our project.

During the design phase of our project, we performed usability testing on our system. This was mostly done in the form of use cases as we increased the detail of our design using different mediums. Use cases are lists of instructions performed by a specific actor with a specific goal in mind. As we first began designing our system, we worked together to design the simplest way to categorize any action any of the four users could perform on our system, while also keeping scope of the users in mind. This resulted in our six modules. Then we created a more detailed paper model of our system. During this step, we performed many use cases for each of the four user roles to make sure any action the user might want to take from within their scope was as simplified as possible. At this step we decided upon each of the individual pages that would make up our system. Then lastly, we created the Fireworks model of our system to perform final use case testing to make sure our system would meet the needs of each of our four user types before we began to develop our system.

As we developed our system, we continued to perform white box testing mostly in the form of functionality testing. As we created our system using a bottom-up design methodology, we tested each basic function as it was created before moving onto the next design layer. This helped us to make sure that our system conformed to the design specifications given to us and to minimize bugs in our code later on. In addition, we continued to perform our own usability testing on the system as minor changes were made in the design during development.

As we began completion of our system, we shifted our usability testing from white box to black box testing. We wrote use cases for our system and presented them to a group of individuals who were not familiar with the code of our system. The results of this testing are shown below.

Test Results

When conducting our usability testing, we asked the following questions to our participants.

1. How easy was it to navigate through the system?
2. Was the interface simple or overwhelming?
3. Were the colors and overall look of the site appropriate?
4. Is the text on the site legible?
5. Does the logo fit within the context of the site?
6. Is the “Track Page” easy to comprehend?

We gave our participants the option to rank the above on a 1-5 scale with 1 being poor and 5 being excellent. Within the above criteria, our survey takers ranked our system to be either 4 or above. The highest ranked criterias were those concerning the design of the interface, such as the logo and color choices. About 46 percent of our survey participants rated the navigation through the system with a 4, making it the lowest rated criteria of the six. Additionally, we gave our participants the opportunity to provide written responses on the functionality of our site. this would allow us to know if they experienced any hiccups during their experience. From these results we learned that 38% of our users had some difficulty downloading the ZIP file. Other users reported some confusion on saving the yet most were able to use the system without any problems.

Documentation

The last phase of our project was to create documentation for the users of our system and the computer engineers who will be traveling to India to install the system on the Anudip and iMerit servers. Those reading our documentation files will gain a better understanding of the code behind our site, as well as the technology involved in the conversion. This can potentially enable the code to be modified for further development. We will hand over our documentation in a folder that will include a “readme” file, setup guide for installation, and a user manual. The “readme” file is a supplement included with most computer software to describe the contents of a directory. Our readme file contains an introduction to our system and directs users on where to find installation and setup instructions. The setup guides will mainly instruct the computer engineers and the technical team at Anudip and iMerit the additions they will have to include in their existing database. Since we created a mock database to test our system, the step of the installation will be fairly simple and only require them to mirror the database we have designed. Secondly, the technical team will have to integrate their current employee information in order to allow our system to access it; this will involve going into the system and setting up user accounts and passwords. Lastly, the user manual in the documentation outlines the capabilities of the four users of our system. The user capabilities diagram (Figure 4.2), the Admin Use Case diagram (Figure 4.3), and other useful diagrams illustrating various functionalities will be used to give users a better understanding of the system.

Installation

Documentation is one of many preparations needed to deliver a product. For our final deliverable, we prepared by organizing all important files, deleting unnecessary files on the existing server folders, and commenting our system code. Another preparation our team found important is giving the users of our system our contact information, should they have any questions in implementation. The process for installation of our system involves uploading our files to Anudip and iMerit’s system and verifying that all the permissions are correct. Installation also involves running SQL files on the database in order to create the tables, initialize the system, and do test projects. Then, the administrator of the system needs to verify that their users are connected to our system; this means they must query their tables to see if the usernames and passwords match. Finally, the administrator can use the user interface we designed and log in using the administrator account to set up the system with initial users and projects.

Section 6 – Societal Issues

Ethics and Web Systems

Many ethical questions are posed with our web tool MUVE. The digitization of books involves copyright issues, which means that in order to ethically digitize printed text, one must be given legal permission from the owners of that information. Otherwise, individuals would be digitizing books unethically and against copyright laws. Thus, our web tool, MUVE, gives users a technology that can be used for good -- working lawfully with publishing companies -- or for bad -- digitizing books, then freely and illegally distributing them. Throughout development, this question was raised many times: how are you ensuring that your product doesn't infringe on copyright laws? Midway through development, we were able to get a better understanding of how our product would be used when we spoke with the iMerit team. They informed us that their existing relationships with publishers meant that the users of our tool will be employed to digitize their books. This means that users will be granted the copyrights to their books, and thus working legally and ethically. However, our system MUVE is unable to store information about the various downloads of finished books, and track where those downloads go. Because digital files can be complicated to track, employees of iMerit will be facing an ethical decision every day when they choose not to download and distribute a publishing company's material. We know, however, that these employees have agreements with the publishing companies and that there is a trust relationship built in.

Social Impact

Our system will socially impact the two organizations we are working with and the communities surrounding them. Since the main users of our tool are the women being trained and employed by Anudip and iMerit, their livelihoods will change with incoming opportunities to digitize books. The communities around the foundations may be affected with the increase of digital text and decrease of printed material available to them. This may cause people in those areas to be more reliant on technology as well as reduce their amount of spending on printed literature.

Manufacturability

When we first designed our system, we considered using scanning technologies to expedite the digitization process. Yet since we are ultimately viewing the content on basic feature phones, we needed to convert the text into xml. Scanning in the text would not have allowed us to do perform this conversion so we improvised by having the Anudip and iMerit employees place text into an editor. Our system provides them the option to either type text into the editor or copy and paste the content in.

Along with the cooperation of the Anudip and iMerit employees, we had to work with the organizations' technical team to integrate and install our system. The best way to achieve this goal is to send computer engineers who are well versed in our system to India to work side by side with their technical team. Although our product does not cost money to manufacture, it will take time to integrate and will require the companies to employ staff to be trained on how to use the system.

Sustainability

In computer and web engineering, sustainability can mean many things; for our system, we can measure sustainability in terms of reusability and longevity of our system. We can also measure sustainability in a broader sense and consider the world resources our project requires. First, we designed our system such that engineers modifying the web software could reuse the code and modules we constructed, which saves time. The longevity of our system is a function of many things: technological relevance, upkeep, and maintenance. We considered the changing technologies in the environment around our system. For example, what if people in the developing communities stop using the "feature phones" for which these books are being digitized? Our system will no longer be relevant (unless it is modified for other phones or digital devices of the future). Second, as many web systems do, there is maintenance and upkeep involved in making our tool useful for the present and future. For instance, the system database needs to be updated for major changes within the organization by the system administrator. Lastly, we considered the resources our system requires to function. Ultimately, our system is not completely self-sustaining, because as web systems often do, MUVE requires energy to store information and access data.

Environmental Impact

Our project affects the environment in two major ways: it uses energy (as mentioned above in the section addressing sustainability), yet it has the potential to save paper. Specifically, when a user stores any information about a project onto the servers, energy is required, as well as whenever a user later accesses that information. Depending on the source of the energy used to power the servers Anudip and iMerit use, different environmental resources are being depleted. Perhaps more significantly than electricity usage in the world is the production of paper and the consequent waste generated. Our web tool, MUVE, participates in a global technological trend towards the digitizing of information. With respect to the environment, digitization of printed text can reduce the amount of paper production; when a customer wishes to purchase a book, she may buy the digital version from one of iMerits partner publishing companies, instead of purchasing the printed book. This means a lower demand for printed copies, thereby reducing paper usage. As the demand for non-recycled paper decreases, less trees will be murdered.

Web Usability

The main focus of our design was to ensure that users would be comfortable working with our system. We were well aware that our users would be non-native English speakers so we included graphical icons wherever possible to communicate the features of the site. We also took into consideration the 10 Usability Heuristics of User Interface Design when designing our system. Some of the more important guidelines we adhered to were: making sure our users knew where they were in our system, including examples to prevent errors, and staying consistent with our terminology. In order to be certain that our tool was user-friendly, we performed black box usability testing and created a manual for everyone who will be interacting with our system.

Political, Economic, Health & Safety

On the other hand political, economic, and health and safety issues did not influence our decisions greatly. Since we are working with two specific organizations, the outcome of our project only affected the their surrounding communities and not the general public. As a result, our project was not political in nature and did not interfere with the policies at the government level. Working with Anudip and iMerit, also allowed us to use their database and avoid costs in manufacturing our system. Additionally, our web tool is built using code and technology that is free to use so economic constraints were avoided. Lastly, our project is not a physical system and thus could not harm users physically.

Lifelong Learning

As a team of students, we have learned many things working during the creation of MUVE. We drafted our solutions ourselves, researched independently, managed our time and resources, and learned new skills. Coding for our site developed our understanding of technologies and scripting languages such as PHP, SQL, and XML. Collaboration was one of the primary tools we used to deepen our knowledge of these languages.

When working on a team, we learned several valuable methods to working and learning. Each of us, to some extent, attempted to round out our skills sets. We learned how to use our complementary strengths for the benefit of our project, as well as challenge ourselves by learning & working on certain aspects of the system. For example, often one group member would teach and instruct the others on a certain technology with which she was fluent. Also, we learned to appreciate increasingly appreciated [word choice?] the benefits of coming to group meetings having done independent research. As we planned out how to work together, we learned a lot about project management & strategizing. We were constantly assessing ours progress in order to learn how to complete our project on a deadline and prioritize our tasks.

One of the most important things we learned is the importance of putting forth serious effort in the design phase. We spent a lot of time designing the user roles, organizational structure for the information, and interface for the web system; actions which, during the construction phase, we grew to appreciate. In particular, we designed our system to be easily adaptable, which has enabled us to make plenty of changes (as is usually necessary during the construction phases of any project). We learned the necessity of merging the phases of software engineering; our project involved simultaneous and ongoing revision, modification, and re-designing across all functions of our system as we progressed. We learned the importance of making a system as flexible as possible throughout all phases of the system.

Compassion

When we were searching for a project to work on, we hoped to find a project that would benefit others. We were fortunate to be connected with Radha Basu through our engineering advisor, Silvia Figueira. Radha's company, the Anudip Foundation, is "dedicated to creating livelihood opportunities for impoverished people in rural areas and urban slums of India." ("Anudip", 2013) After conversations with Basu and Figueira, our project objectives emerged: to continue provide job opportunities for the women in Anudip and iMerit, as well as digitize educational materials. So, we were able to adopt the vision behind these non-profit organizations as our project's vision: to MUVE jobs and better education to change lives. Additionally, we have been inspired to work with a company that is motivated by greater causes for the world. According to their website, Anudip reveals one of their aspirations: *"It is our dream that our integrated model of training, placement, entrepreneurship and project services will become the archetype for international development."* ("Anudip", 2013)

Section 7 – Conclusion

For our senior project, we created a web tool for a nonprofit organization based in India. We learned how to create a site that not only converts text into XML, but manages employee information and securely stores files. Our team learned how to make a detailed system for a business, which was challenging, due to the complexity required by having the different users.

We were able to learn a lot from the multifaceted aspects of our project. The research, design, construction, and preparatory work we've had the opportunity to do has taught us how to manage our time efficiently for a long-term project, compared to weekly class assignments. The design phase easily became the longest, most detail-oriented phase of our project creation, and we grew to strongly appreciate prototyping. We also learned the importance of planning ahead realistically and leaving plenty of time for construction of the site. Collaboration gave us a reason to identify our personal strengths as well as individually challenge ourselves to work with unfamiliar technologies.

After installation of MUVE, the final result of the converted text from our tool must be processed, which will require a simple application to interpret the XML on the basic mobile phones for which the project is intended. Currently, our system successfully converts the text into XML files that can be viewed through a browser. However, this text includes the XML tags, so a basic application must be used to read the XML and output the text in a readable format on the various feature phones. The computer engineers traveling to India to help install our conversion tool will simultaneously be creating a basic XML interpreter.

In conclusion, we're excited that our system will in fact be installed and used by the intended users of the system, beginning this summer. We recently found out that publishers have already contacted Radha's organization and iMerit and are ready to employ Anudip women to digitize books using our tool. And, one of our visions for this tool is that it will be used to put better education resources--books created through our site--into the schools in rural areas nearby. So, we hope to be "MUVE-ing" better jobs and education to improve the lives of people in developing communities in India.

Section 8 – References

Field

Radha Basu, Founder of Anudip and CEO of iMerit
Anindya Chattopadhyay, the Chief Delivery and Development Officer of iMerit

Citations

Cave, Kathryn. (2013, May 10). India: How Feature Phone Apps Thrive in Emerging Markets... Now Spurred on by Romances! *IDG Connect* –. Retrieved from http://www.idgconnect.com/blog-abstract/1690/india-how-feature-phone-apps-thrive-emerging-markets-now-spurred-romances

Millar, Michael. (2013, May 6). Market for feature phone apps is low-tech goldmine. *BBC UK.* Retreived from http://www.bbc.co.uk/news/business-22390444

Shrinivasan, Rukmini. (2012, Jul 11). Unemployment lowest in Gujarat, highest in Goa. *Times of India.* Retrieved from http://articles.timesofindia.indiatimes.com/2012-07-11/india/32632097_1_unemployment-rate-labour-force-women-in-urban-areas

Unknown Author. (2012, Mar 1). India unemployment rate 'falling'. *BBC UK.* Retrieved from http://www.bbc.co.uk/news/world-asia-india-17216820

Unknown Author. (2013) Anudip. Retrived from http://anudip.org/about/

Section 9 – Appendices

Adobe Fireworks Prototype

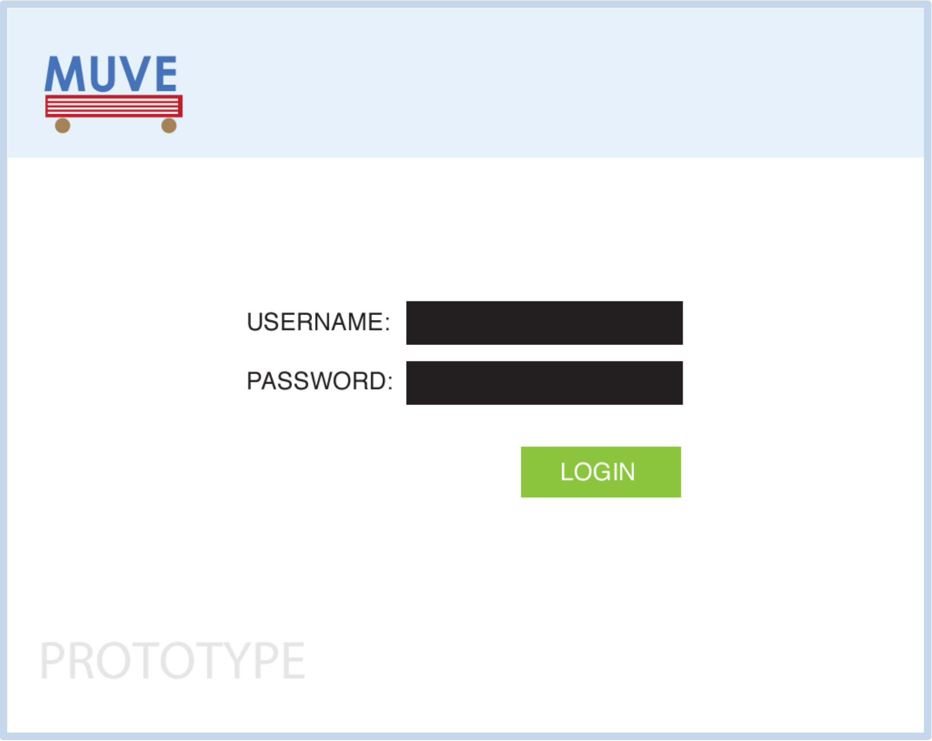


Figure 9.1 Prototype Login Page

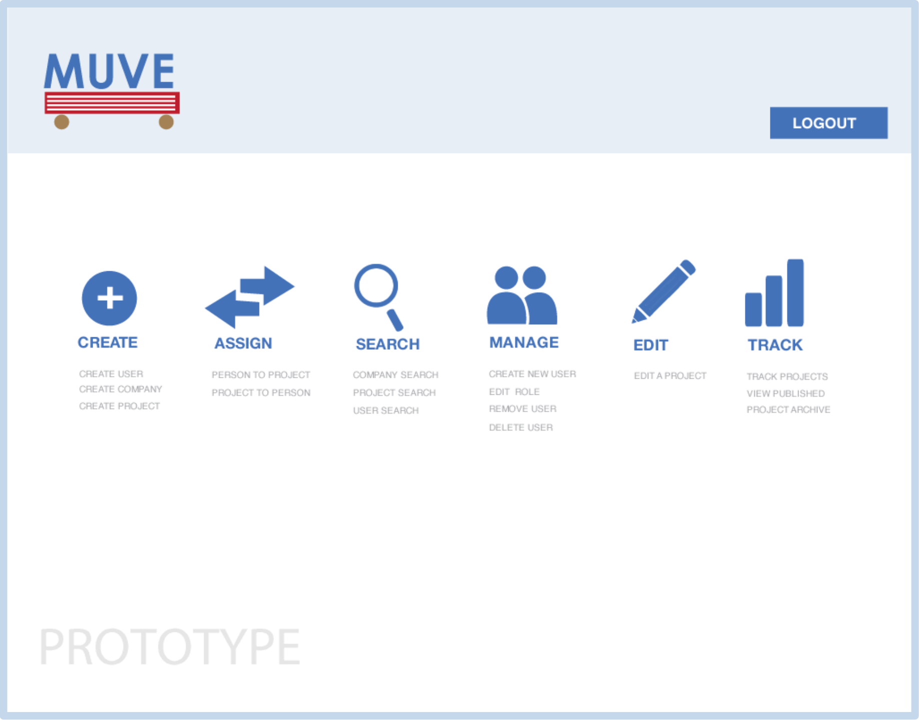


Figure 9.2 Prototype Home (Splash) Page

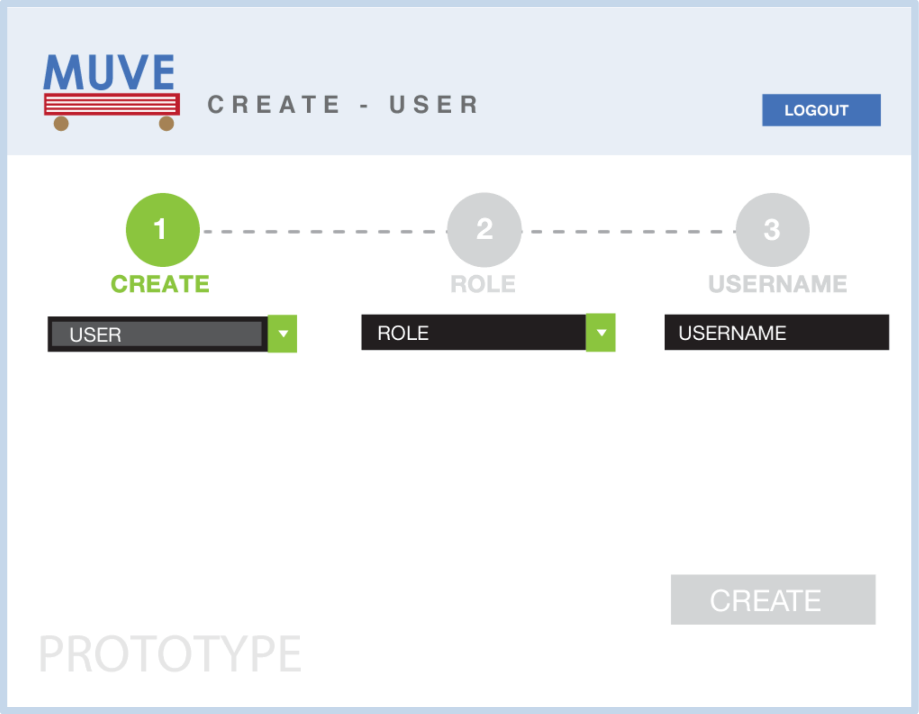
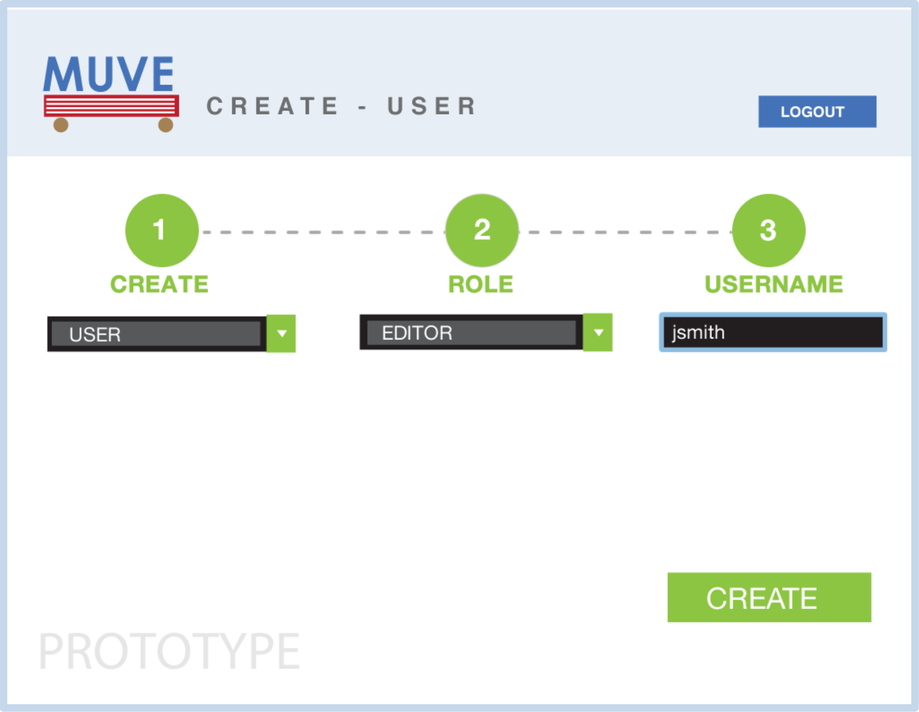


Figure 9.3 Prototype Create User Screen 1

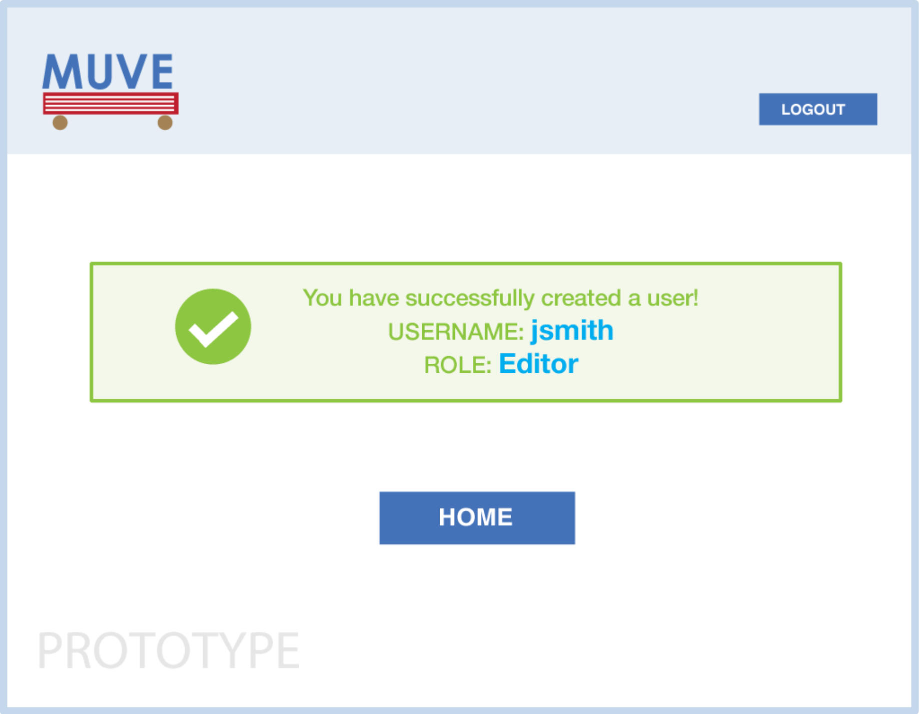


CREATE: USER

ROLE: EDITOR

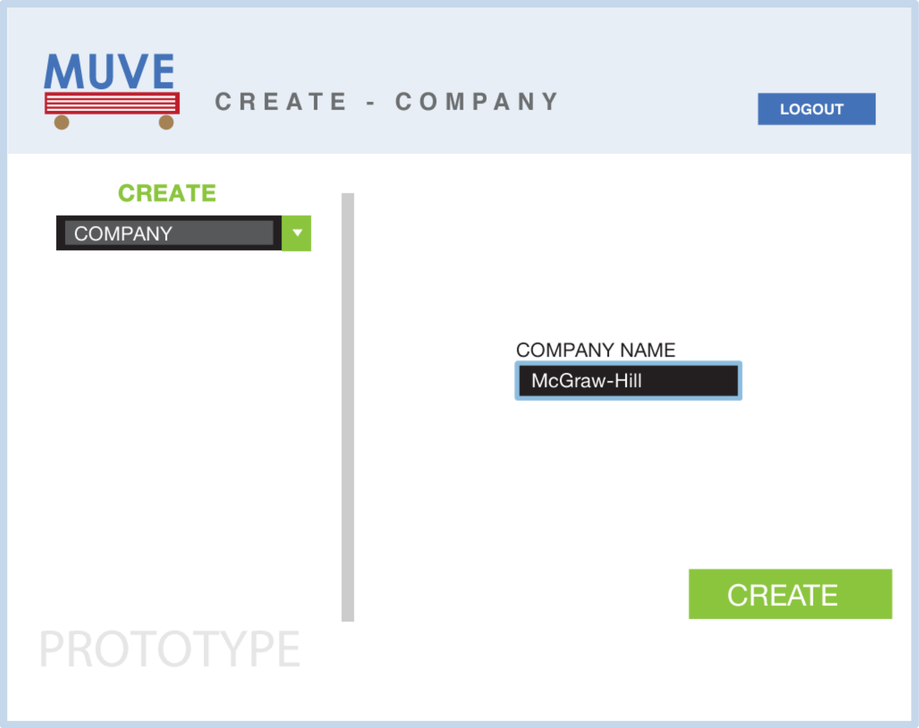
USERNAME: jsmith

Figure 9.4 Prototype Create User Screen 2



You have successfully created a user!
USERNAME: jsmith
ROLE: Editor

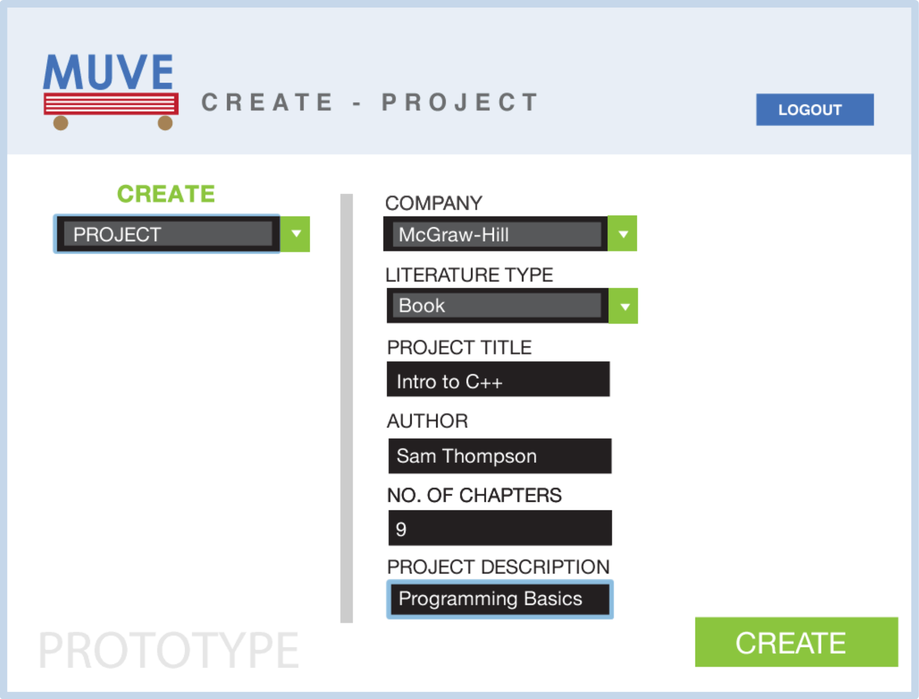
Figure 9.5 Prototype Create User Confirmation



COMPANY ▾

COMPANY NAME
McGraw-Hill

Figure 9.6 Prototype Create Company



PROJECT ▾

COMPANY
McGraw-Hill ▾

LITERATURE TYPE
Book ▾

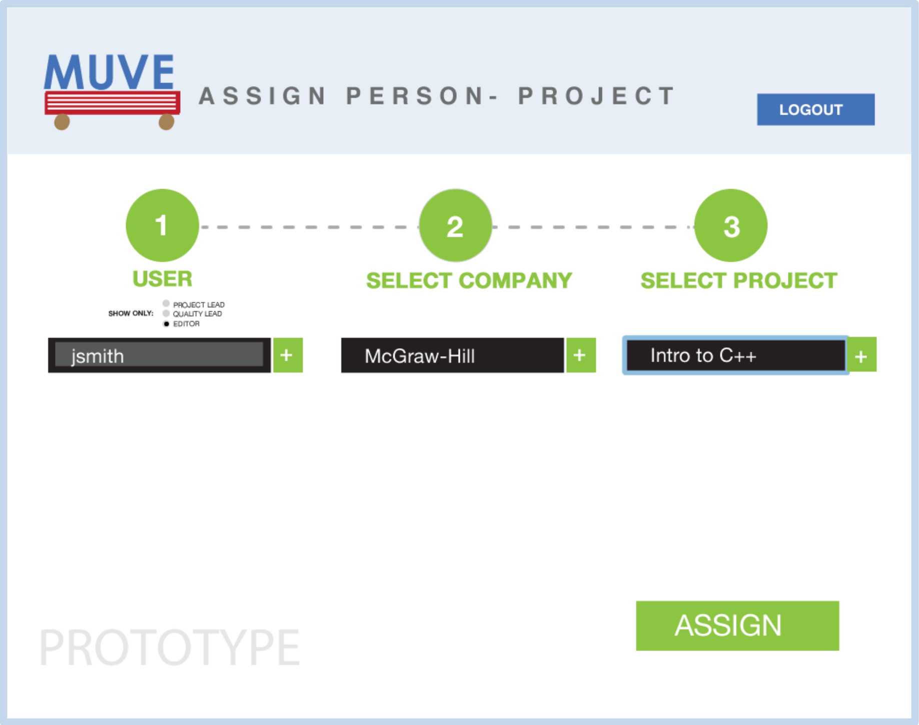
PROJECT TITLE
Intro to C++

AUTHOR
Sam Thompson

NO. OF CHAPTERS
9

PROJECT DESCRIPTION
Programming Basics

Figure 9.7 Prototype Create Project

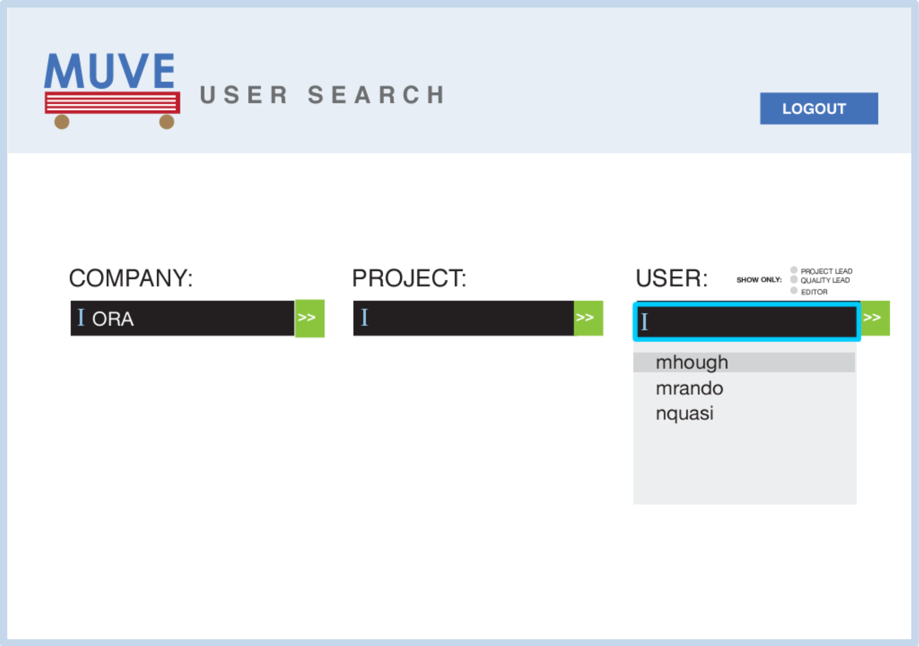


jsmith

McGraw-Hill

Intro to C++

Figure 9.8 Prototype Assign Project



COMPANY: ORA

PROJECT:

USER:

mhough
mrando
nquasi

Figure 9.9 Prototype Search for a User

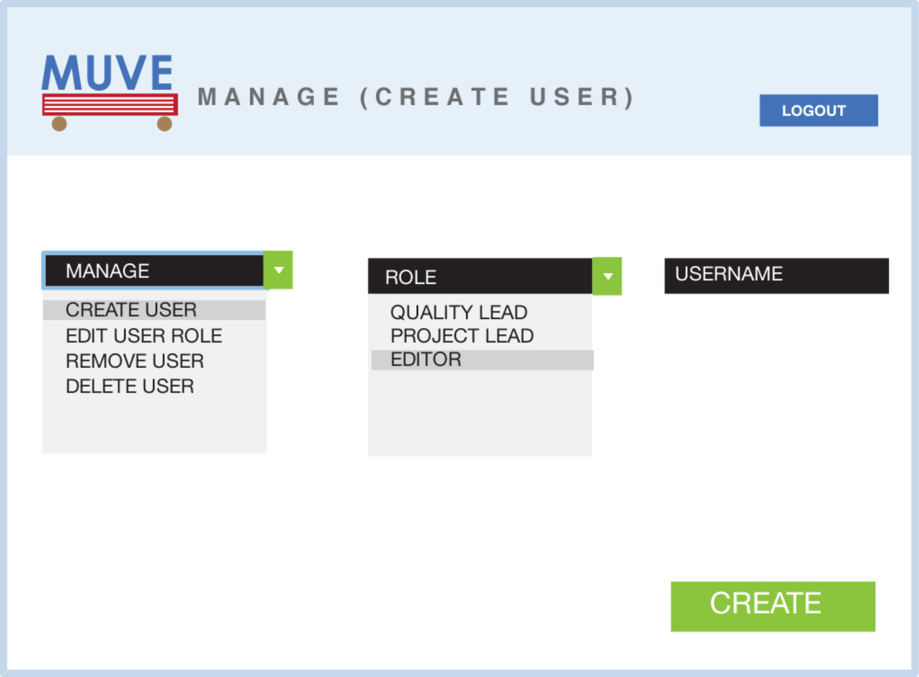
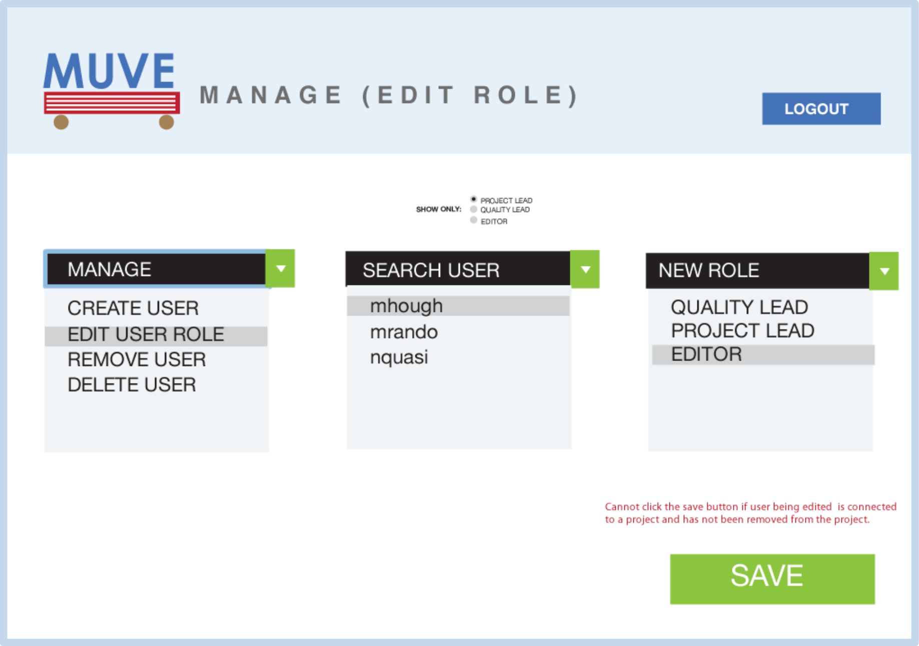


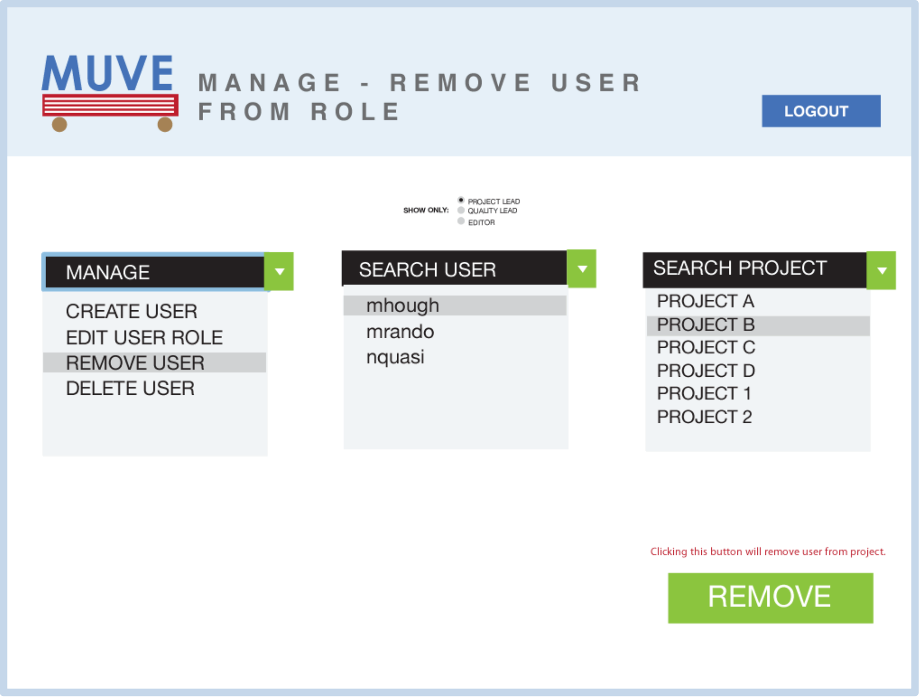
Figure 9.10 Prototype Manage Users



SHOW ONLY: ● PROJECT LEAD ○ QUALITY LEAD ○ EDITOR

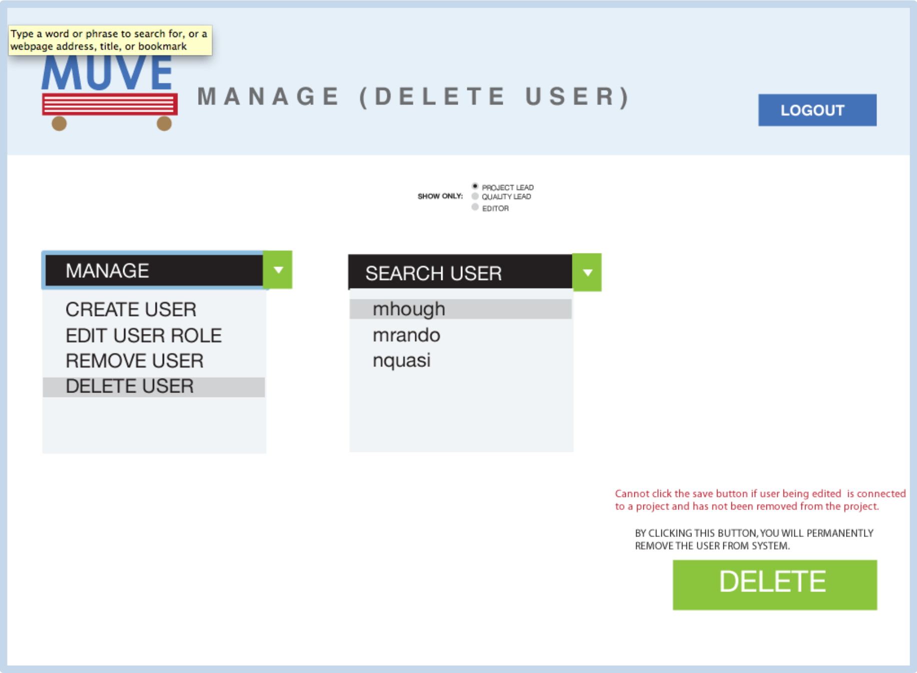
Cannot click the save button if user being edited is connected to a project and has not been removed from the project.

Figure 9.11 Prototype Manage User Roles



Clicking this button will remove user from project.

Figure 9.12 Prototype Remove User Role

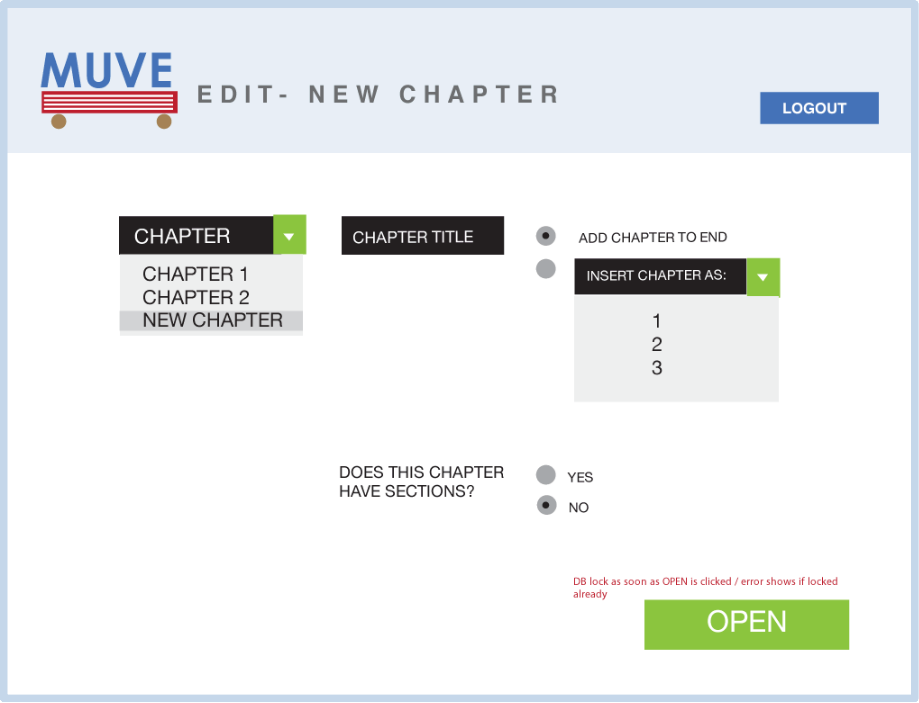


Type a word or phrase to search for, or a webpage address, title, or bookmark

Cannot click the save button if user being edited is connected to a project and has not been removed from the project.

BY CLICKING THIS BUTTON, YOU WILL PERMANENTLY REMOVE THE USER FROM SYSTEM.

Figure 9.13 Prototype Delete User



● ADD CHAPTER TO END
○ INSERT CHAPTER AS:

DOES THIS CHAPTER HAVE SECTIONS? ○ YES ● NO

DB lock as soon as OPEN is clicked / error shows if locked already

Figure 9.14 Prototype Edit New Chapter

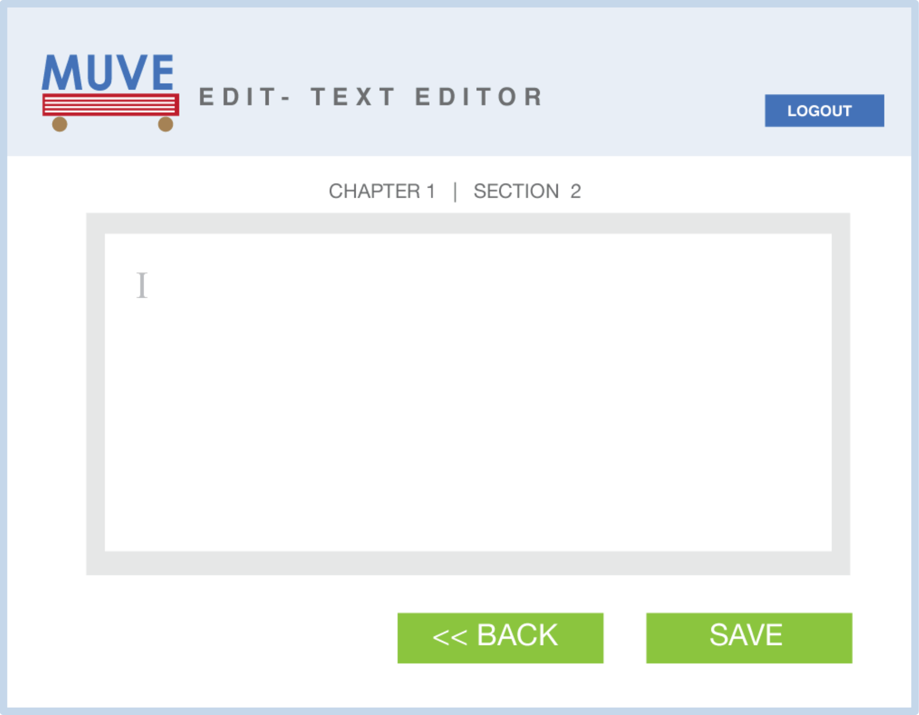
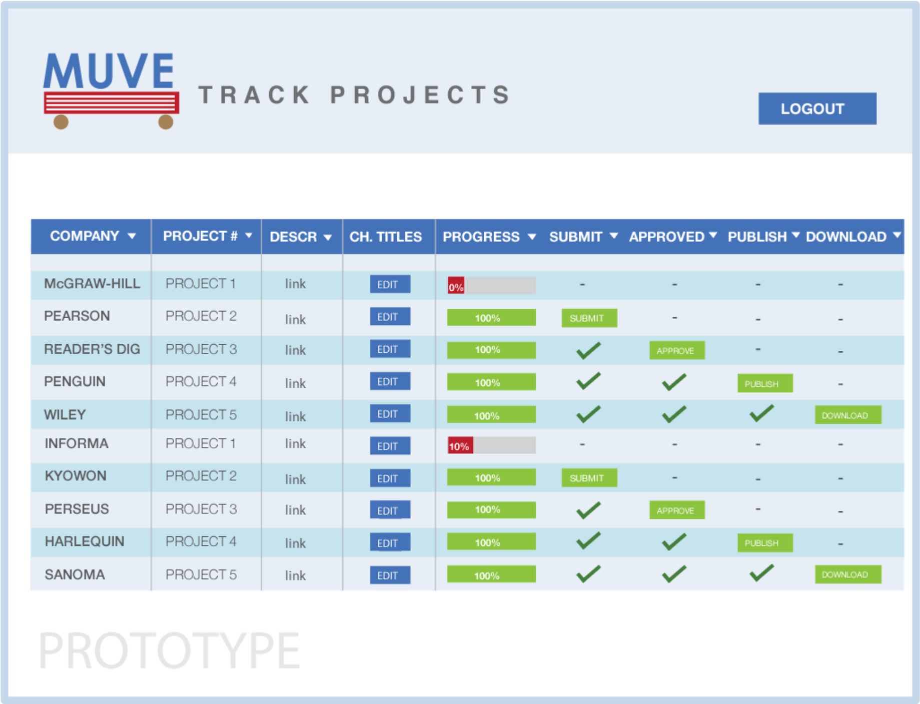


Figure 9.15 Prototype Text Editor



MUVE

TRACK PROJECTS

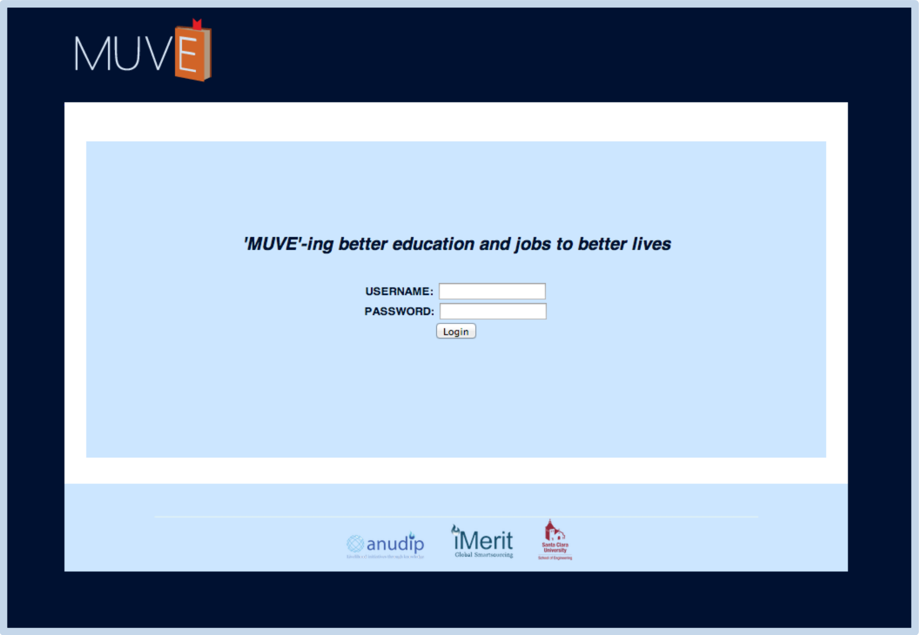
LOGOUT

COMPANY ▼	PROJECT # ▼	DESCR ▼	CH. TITLES	PROGRESS ▼	SUBMIT ▼	APPROVED ▼	PUBLISH ▼	DOWNLOAD ▼
McGRAW-HILL	PROJECT 1	link	EDIT	0%	-	-	-	-
PEARSON	PROJECT 2	link	EDIT	100%	SUBMIT	-	-	-
READER'S DIG	PROJECT 3	link	EDIT	100%	✓	APPROVE	-	-
PENGUIN	PROJECT 4	link	EDIT	100%	✓	✓	PUBLISH	-
WILEY	PROJECT 5	link	EDIT	100%	✓	✓	✓	DOWNLOAD
INFORMA	PROJECT 1	link	EDIT	10%	-	-	-	-
KYOWON	PROJECT 2	link	EDIT	100%	SUBMIT	-	-	-
PERSEUS	PROJECT 3	link	EDIT	100%	✓	APPROVE	-	-
HARLEQUIN	PROJECT 4	link	EDIT	100%	✓	✓	PUBLISH	-
SANOMA	PROJECT 5	link	EDIT	100%	✓	✓	✓	DOWNLOAD

PROTOTYPE

Figure 9.16 Prototype Track Page

MUVE Screenshots



'MUVE'-ing better education and jobs to better lives

Figure 9.17 Login Page

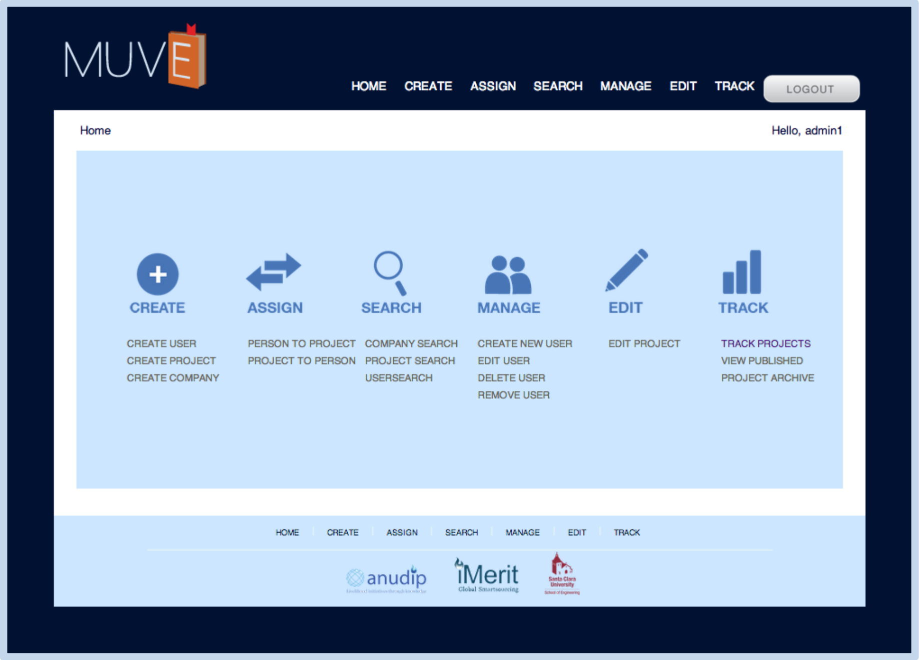
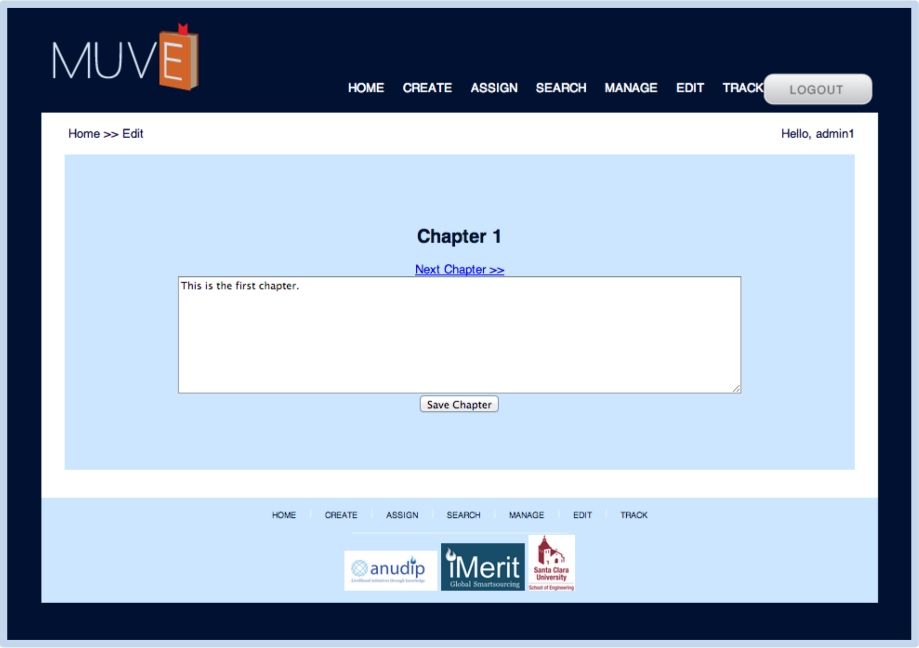


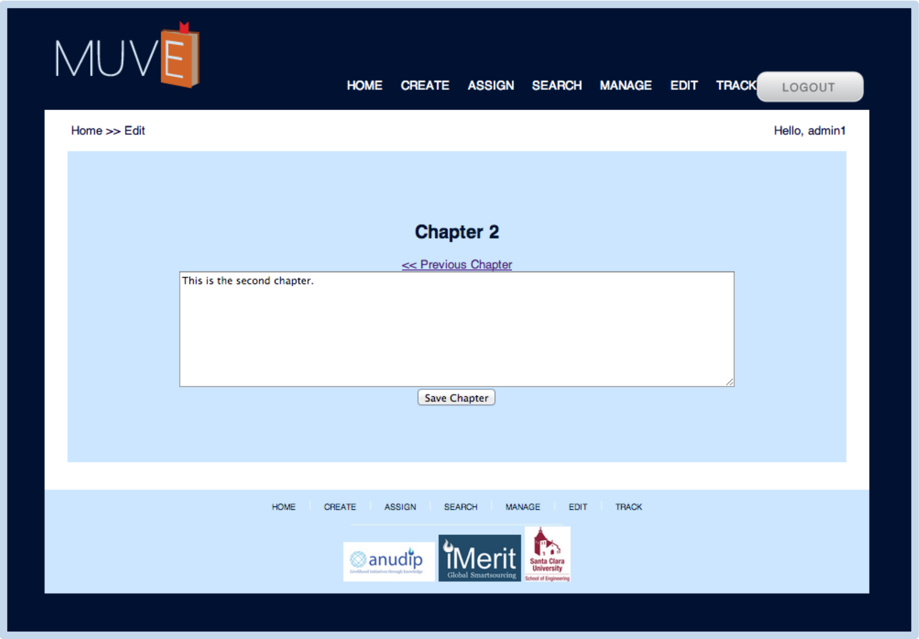
Figure 9.18 Home (Splash) Page



Chapter 1

This is the first chapter.

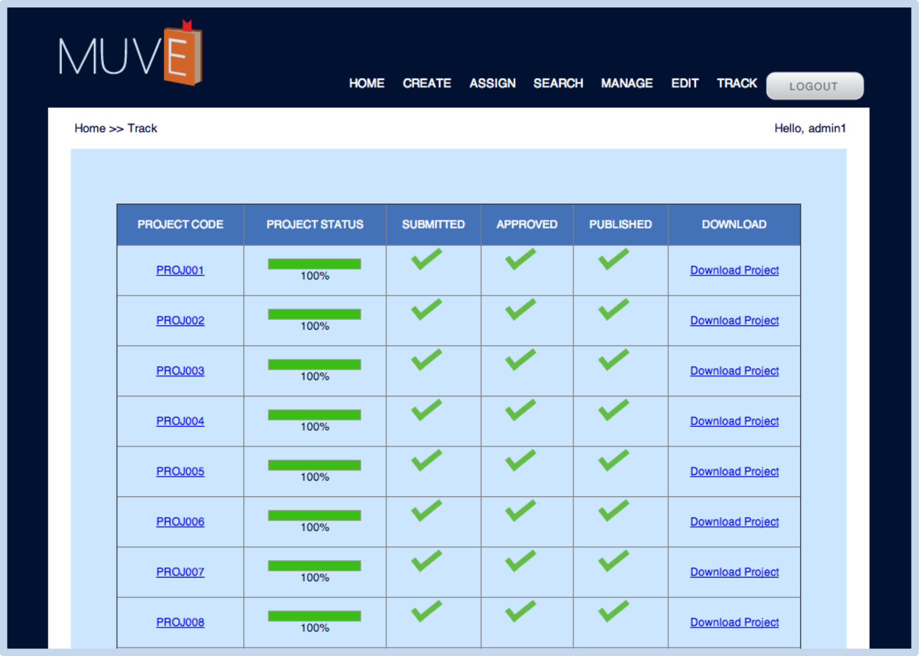
Figure 9.19 Text Editor (First Chapter)



Chapter 2

This is the second chapter.

Figure 9.20 Text Editor (Next Chapter)



Home >> Track

Hello, admin1

PROJECT CODE	PROJECT STATUS	SUBMITTED	APPROVED	PUBLISHED	DOWNLOAD
PROJ001	100%	✓	✓	✓	Download Project
PROJ002	100%	✓	✓	✓	Download Project
PROJ003	100%	✓	✓	✓	Download Project
PROJ004	100%	✓	✓	✓	Download Project
PROJ005	100%	✓	✓	✓	Download Project
PROJ006	100%	✓	✓	✓	Download Project
PROJ007	100%	✓	✓	✓	Download Project
PROJ008	100%	✓	✓	✓	Download Project

Figure 9.21 Track Page

User Manual

The user manual facilitates understanding of our system for each of the four user roles. Through our manual, every user will gain a comprehensive understanding of how to navigate through our site and accomplish each user's goals. In addition, the manual will contain sections specific to each role. The Admin's section will explain how to setup the system in simple terms and use the management tools. The Project Lead's section will describe how to associate Quality Leads and Editors with projects. The Quality Lead's section will elaborate how to monitor the progress of Editors and review projects. The Editor's section will explain how to use the text editor.

Login

Open the homepage of MUVE in a major Internet browser (Firefox, Safari, Internet Explorer). In the text box next to 'Username,' type your Anudip or iMerit e-mail username. Then in the text box next to 'Password,' enter your password associated with your Anudip or iMerit e-mail username. Then click the 'Login' button near the bottom of the screen.

If you have forgotten your password, please contact your manager. If you are not registered on MUVE, please contact a MUVE Administrator.

Logout

To log out of the system, click the "logout" button on the top right-hand corner of the screen. Below, are *steps begin from the landing page, where the six modules, Create, Assign, Search, Manage, Edit, and Track, are listed.*

Create

To create a company, select the Create Company link below the Create module on the landing page. Enter the Company's name in the text field displayed. Click on the "Submit" button to create the project.

To create a project within MUVE, select the Create Project link below the Create module on the landing page. Enter the necessary attributes of the project such as the title and author in the text fields displayed. Click on the "Submit" button to create the project.

To create a user, select the Create User link below the Create module on the landing page. Enter the user's name in the text field displayed. Select a user role from the drop-down menu. Click on the "Submit" button to create the user.

Only users whose role is Admin have access to this module.

Assign

To assign people to a project, select the "Assign People to Project" link below the Assign module on the landing page. Select a user from the menu, and select a project with which to associate that user. Click on the "Assign" button on the bottom right-hand corner to enable that user to begin working on the project.

To assign project to a person, select the "Assign Project to People" link below the Assign module on the landing page. Select a project from the menu, and select users to associate with that project. Be sure that the user roles are correct, and click on the "Assign" button on the bottom right-hand corner to enable those users to begin working on the project.

Users whose role is either Editor or Quality Lead do not have access to this module.

Search

To search for a user, select the "People Search" link below the Search module on the landing page. Select a user from the menu, and click on the arrow next to the search box on the right side.

To search for a project, select the "Project Search" link below the Search module on the landing page. Select a project from the menu, and click on the arrow next to the search box on the right side.

To search for a company, select the “Company Search” link below the Search module on the landing page. Select a company from the menu, and click on the arrow next to the search box on the right side.

Users whose role is Editor do not have access to this module.

Manage

To manage the users within MUVE, Administrators and Project Leads can select from the following links below the Manage module: “Create New User,” “Edit User,” “Delete User,” and “Remove User.”

“Create New User” takes you to the Create module’s “Create User” link.

To edit a user’s role, select the “Edit User” link. Select a new role from the list of options, and click the “Submit” button.

To delete a user from a project, select the “Delete User” link, and type in the user’s name in the text field. Select the project from which you would like to remove the user, and click the “Submit” button.

To remove a user from the system, select the “Remove User” link, and type in the user’s name in the text field. Click the “Submit” button to permanently remove this user from the system. They will no longer be associated with any projects and therefore can no longer login to the system.

Users whose role is Editor do not have access to this module.

Edit

To edit a project within the system, select the “Edit” module on the landing page or from the navigation menus located on the upper right corner or at the bottom of the page. Select the project and the chapter to be edited. Upon making this selection, the user will be directed to the text editor. Type or paste in the text to be added to the chapter. When finished, select “Save Chapter.” To edit the other chapters within the current project, select “Next Chapter” or “Previous Chapter.” Remember to periodically select “Save Chapter” to avoid losing work.

Track

To track projects within the system, select the “Track” on the landing page or from the navigation menus located on the upper right corner or at the bottom of the page. Upon selection, the user will encounter a table of projects they are associated with. Depending on the user, they have the option to submit, download, or publish a project.

To download projects from the server, select the “View Published” link below the Track module. Here, select from the list of published projects, and download the converted XML file. In order to download these completed projects, the project must first be submitted by any user. Then, it must be approved by the Quality Lead, Project Lead, or an Administrator. To enable download of the converted file, the project must be published by the Project Lead and/or an Administrator.

If your user role is an admin, you can track/download all projects within the system; other roles can only track/download the projects with which they are associated.

Set-up Guide & Installation Guide

General Steps

- Upload files onto server
- Set-up MySQL Database
- Connect MUVE to MySQL Database
- Connect MUVE to Anudip Login db
- Deleting temporary/ZIP project folders/files

Upload Files onto Server

1. Un-compress MUVE files
2. Upload MUVE files onto server
3. Check permissions of files

Set-up MySQL Database

1. The folder “sql_script” in the MUVE folder contains various SQL scripts for a MySQL database.

2. Login to your database and run “create.sql” to create the MUVE database

3. Run “initialize.sql” to initialize the MUVE database with preset user roles and test users

3. Run any of the test files to test the system with some premade projects.

4. Running “delete.sql” on your system will delete the entire MUVE database. This will NOT delete projects already created on the server.

Connect MUVE to MySQL Database

1. Open the “db_info.php” file located in the “db_php” folder in a text editor.

2. Define the Host, Account, Password, and Database for your MySQL database in this file.
3. Upload the new “db_info.php” file to the server and override the old file.

Connect MUVE to Anudip Login Database

Currently, the MUVE database only recognizes users and password associations found in the “UserList” Table in the MUVE database. To give Anudip employees access to MUVE you may either 1) update the “UserList” table with Anudip employee information or 2) edit the “validate_login()” function located in “module_php/login.inc.php” to check another Anudip database to verify a user’s password.

Deleting temporary/ZIP project folders/files

Currently, there is no script to delete temporary folders created when downloading a project or the ZIP files created when downloading a project. All temporary folders are stored in “conversion/temp”. The contents of this “temp” folder may be deleted periodically to make more room on the server. All ZIP files are stored in the “conversion” folder. These ZIP files may also be deleted periodically to make more room on the server. However, if “conversion.php” or “conversion.inc.php” in the “conversion” folder are deleted, MUVE will no longer be able to download projects.

Installation Guide

Before MUVE can be set-up and fully functioning with its intended functionality, additional steps must be taken at the discretion of the one installing the system. In order for MUVE to function with the fully intended functionality, the additional steps must be taken:

1. Acquiring an HTTPS security certificate for the URL on which MUVE is to be hosted.

2. Completing the HTML forms of the Create, Assign, and Manage modules to work with pre-existing PHP functions.
3. Styling the Create, Assign, Search, and Manage modules at the discretion of the installer.
4. Additional stress testing of the system at the discretion of the installer.

Code

The following pages contain system code. We included examples from every end of our site and every module. The included function names are titled:

- bindbook
- canassignqualitylead
- canviewproject
- dbconnect
- dbcreate
- getfirstchapter
- iseditor
- listchapters
- submitproject
- test
- userprojects
- writeshelf
- zipme.

```php
<?php

/*********************************************************************************************** BIND_BOOK*/
/* in file: conversion/conversion.inc.php */
/* calls write_shelf() and write_chapter() to create all the necessary files to download a project in a temporary folder before zipping the file and sending it to the browser
    project_id - the unique id of the project that is being downloaded
    connection - connection to the MySQL database */


function bind_book($project_id, $connection)    // project code --- should we use this?
{

    if ((!isset($project_id)) OR (!isset($connection)) ) {
        $error = "Invalid parameters";
        return (array(0, $error));
    }

    // create variable for path to folder store files in temp_projects
    $path = "temp/";

    // create an array $files to store filenames for each file to zip
    $files = array();


    // create a name for the project folder
    $time = time();
    $proj_folder = "muve_" . $project_id . "_" . $time;

    $full_path = $path . $proj_folder;
    // make the folder if it does not exist
    if (!file_exists($full_path)) {
        mkdir($full_path);
    }

    // Write chapter files by calling write_chapter
    // loop to make a file for each chapter

    list($check, $chapter_info) = list_chapters($project_id, $connection);

    $i = 0;
    while(!empty($chapter_info['chapter_id'][$i]))
    {
        // get chapter id and project id
        $chapter_number = $chapter_info['chapter_number'][$i];
        $chapter_id = $chapter_info['chapter_id'][$i];

        // create a filename, pass to write_chapter
        $ch_file =  "c_" . $project_id . "_" . $chapter_number . ".xml";
        $chapter_filename = $full_path . "/" . $ch_file;

        list($check, $err) = write_chapter($chapter_id, $chapter_filename, $project_id, $connection);

        // store filename of chapter file in array
        $files[$index] = $ch_file;

        $i++;

    }

    // Write shelf file: call write_shelf
    $shelf = $full_path . "/shelf.xml";
    list($check, $errs) = write_shelf($shelf, $project_id, $connection);

    // store filename of shelf file in array
    $files[$index] = "shelf.xml";


    // Zip files on server with the download link
    zip_me($proj_folder);

    return ($proj_folder);
}

?>
```

```php
<?php

/**************************************************************************************************************************** CAN ASSIGN QUALITY LEAD */
/* in file: db_php/permission.inc.php */
/* 1 if user has the ability to assign a quality lead, 0 if the user does not have permission
    user_name - unique name of the user
    connection - connection to the MySQL database */


function can_assign_quality_lead($user_name, $connection) { // requires username and link to db

        if( (!isset($user_name)) OR (!isset($connection)) )
            return NULL;

        $query = sprintf("SELECT Action.assign_quality_lead FROM Action INNER JOIN User ON Action.role_id=User.role_id WHERE User.user_name='%s'", mysql_real_escape_string($user_name));
        $result = mysql_query($query, $connection);

        if(!$result) // query failed
            return (NULL);

        $row = mysql_fetch_assoc($result); // expecting only one row from query
        foreach($row as $name => $value) // loop, but only one element in array
            $permission = $value;

        if($permission == 1 || $permission == 0) // make sure the value you get is a 1 or 0
            return ($permission);
        else
            return (NULL); // if something went wrong, return NULL
    }


?>
```

```php
<?php

/**************************************************************************************************************************** CAN VIEW PROJECT */
/* in file: db_php/permission.inc.php */
/* 1 if user has the ability to view the specified project, 0 if the user does not have permission, NULL if something went wrong
    proj_id - unique id of project
    user_id - unique id of the user
    connection - connection to the MySQL database */


function can_view_project($proj_id, $user_id, $connection) { // requires project id, username, and link to db

        if( (!isset($proj_id)) OR (!isset($user_id)) OR (!isset($connection)) )
            return NULL;

        // to have permission a user must
        //  a) be an admin
        //      or
        //  b) be associated with the project in the Work table

        // a) if you can create a project, then you should have sufficient permissions to create any chapter for any project
        $query = sprintf("SELECT Action.create_project FROM Action INNER JOIN User ON Action.role_id=User.role_id WHERE User.user_id=%s", mysql_real_escape_string($user_id));
        $result = mysql_query($query, $connection);

        if(!$result) // query failed
            return(NULL);

        $row = mysql_fetch_assoc($result);
        foreach($row as $name => $value)
        $permission = $value;

        if(strcmp($permission, "1") == 0)
            return (1);

        // b) see if user is associated with the project in the Work table

        $query2 = sprintf("SELECT Project.project_id FROM Project INNER JOIN Work ON Work.project_id=Project.project_id WHERE Work.user_id='%s' AND Work.project_id='%s'", mysql_real_escape_string($user_id), mysql_real_escape_string($proj_id));
        $result2 = mysql_query($query2, $connection);

        if(!$result2) // query failed
            return (NULL);

        $row2 = mysql_fetch_assoc($result2); // expecting only one row from query
        foreach($row2 as $name2 => $value2) // loop, but only one element in array
            $proj = $value2;

        if($proj > 0) // make sure the value you get is a 1 or 0
            return (1);
        else
            return (0); // if something went wrong, return NULL

    }

?>
```

```php
<?php

/**************************************************************************************************** DB_CONNECT*/
/* in file: db_php/connect.php */
/* returns a connection to the MySQL database
whenever this function is used,
'mysql_close' must be called after
the connection is done being used */


function db_connect() {

    // assuming info is in the same folder
    require_once("db_info.php"); // defines info needed to access the db

    // mysql_connect makes a connection to the database given
    //      HOST, ACCOUNT, PASSWORD, and DATABASE
    //      - this connection is needed to make queries to the database
    $connection = mysql_connect(HOST, ACCOUNT, PASS);

    // check that 'mysql_connect' worked:
    //      if 0 then connect failed
    if (!$connection)
        return 0;
    else {
        // now that we are connected to the server, we select a database
        $worked = mysql_select_db (DB, $connection);
        if ($worked)
            return $connection;
        else
            return 0;
    }
}


?>
```

```php
<?php

/****************************************************************************************************************************** CREATE CHAPTER */
/* in file: db_php/create.inc.php */
/* makes a new chapter for a project that already exists */


function create_chapter($ch_title, $ch_number, $ch_sections, $proj_id, $user_id, $connection) {

        // can leave chapter title blank

        if (!isset($ch_number))
            $errors[] = "Chapter Number missing.";

        if (!isset($ch_sections))
            $errors[] = "Must decide whether the chapter has sections.";

        if (!isset($proj_id))
            $errors[] = "Project not specified.";

        if (!isset($user_id))
            $errors[] = "User not logged on.";

        if (!isset($connection)) {
            $errors[] = "Could not connect to the database.";
            return (array(0, $errors));
        }



        // check chapter title
        //  - can be empty
        //  - if not empty, then alphanumeric

        if(strlen($ch_title) > 0) {
            $pattern = '/^[\w\s]*$/'; // only alphanumeric characters with spaces
            if (!preg_match($pattern, $ch_title))
                $errors[] = "Chapter Title contains invalid characters.";
        }

        // check chapter number
        //  - int > 0

        if(!is_int($ch_number) OR $ch_number < 1) {
            $error[] = "Invalid Chapter Number.";
        }

        // check chapter sections
        //  - 0 OR 1

        if($ch_sections != 1 AND $ch_sections != 0)
            $errors[] = "Must decide whether the chapter has sections.";



        // check project id
        //  - project must already exist

        if(project_exists($proj_id, $connection) != 1) { // requires "edit.inc.php"
            $errors[] = "Project does not exist.";
        }


        // check user id
        //  - user must have permission to edit project

        if(can_view_project($proj_id, $user_id, $connection) != 1) // requires "permission.inc.php"
            $errors[] = "User does not have permission to edit project.";



        // if no errors, proceed otherwise return errors
        if(strlen($errors[0]) == 0) {

            $query = sprintf("INSERT INTO Chapter (chapter_title, chapter_number, chapter_sections, project_id) VALUES ('%s', %s, %s, %s)", mysql_real_escape_string($ch_title), mysql_real_escape_string($ch_number), mysql_real_escape_string($ch_sections), mysql_real_escape_string($proj_id));
            $result = mysql_query($query, $connection);

            if(!$result) { // query failed
                $errors[] = "Error creating new chapter.";
                return (array(0, $errors));
            }
            else
                return (array(1, $errors));

        }
        else {
            return (array(0, $errors));
        }

        return (array(0, $errors));
    }


?>


```

```php
<?php

/******************************************************************************************************************************** GET FIRST CHAPTER */
/* in file: db_php/edit.inc.php */
/* retrieves the first chapter associated with a project
    proj_id - unique project id for the project
    connection - connection to the MySQL database */


    function get_first_chapter($proj_id, $connection) {
        if( (!isset($proj_id)) OR (!isset($connection))  )
        {
            return(0);
        }

        $ch_numb = 1;

        $query = sprintf("SELECT chapter_id FROM Chapter WHERE chapter_number=%s AND project_id=%s", mysql_real_escape_string($ch_numb), mysql_real_escape_string($proj_id));

        $result = mysql_query($query, $connection);

        if(!$result)    //Checking if query failed
        {
            return (0);
        }
        else
        {
            $row = mysql_fetch_assoc($result);
            $ch = $row['chapter_id'];

        }
        return($ch);

    }

?>
```

```php
<?php

/**************************************************************************************************************************************** IS EDITOR */
/* in file: db_php/permission.inc.php */
/* returns true if the user is an editor, 0 if not, NULL if something went wrong
    user_id - unique id of the user
    con - connection to the MySQL database */

function is_editor($user_id, $con) {
        if( (!isset($user_id)) OR (!isset($connection)) )
            return NULL;

        $query = sprintf("SELECT role_id FROM User WHERE user_id=%s", mysql_real_escape_string($user_id));
        $result = mysql_query($query, $connection);

        if(!$result) // query failed
            return (NULL);

        $row = mysql_fetch_assoc($result); // expecting only one row from query
        foreach($row as $name => $value) // loop, but only one element in array
            $role_id = $value;

        if($role_id == 4) { // is editor
            return (1);
        }
        else {
            if($role_id > 0)
                return  (0);
            else
                return (NULL);

        }


    }

?>
```

```php
<?php

/******************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************************* LIST CHAPTERS */
/* in file: db_php/edit.inc.php */
/* gets an array of all chapters associated with a project. this is done using the project id. the info we want about each chapter is the id and title
    project_id - unique id of project
    connection - connection to MySQL database */


function list_chapters($project_id, $connection) { // requires project id and link to db

        if( (!isset($project_id)) OR (!isset($connection)) ) { // make sure some sort of parameters were passed in
            $error = "Invalid parameters";
            return (array(0, $error));
        }

        $query = sprintf("SELECT chapter_id, chapter_title, chapter_number FROM Chapter WHERE project_id=%s", mysql_real_escape_string($project_id));
        $result = mysql_query($query, $connection);

        if(!$result) { // query failed
            $error = "Query failed";
            return (array(0, $error));
        }


        $i = 0;
        while( $row = mysql_fetch_assoc($result) ) {


            foreach($row as $name => $value) {
                $chapter[$name][$i] = $value;
            }

            $i++;
        }

        return (array(1, $chapter));
    }
?>
```

```php
<?php

/*****************************************************************************************************************************************************SUBMIT_PROJECT */
/* in file: db_php/edit.inc.php */
/* changes a value in the database to alert the system that a particular project has been submitted for approval
    p_id - unique project id of the project that is being submitted for approval
    con - connection to the MySQL database */


    function submit_project($p_id, $con) {
        if( (!isset($p_id)) OR (!isset($con))  )
        {
            return (0);
        }

        $query = sprintf("UPDATE Project SET project_submitted=1 WHERE project_id=%s", mysql_real_escape_string($p_id));
        $result = mysql_query($query, $con);

        if(!$result)    //Checking if query failed
        {
            return (0);
        }


        return(1);
    }

?>
```

```php
<?php

/**************************************************************************************************************************************** USER PROJECTS */
/* in file: db_php/edit.inc.php */
/* gets array of projects a user is associated with given the user's id
    user_id - unique id of user
connection - connection to MySQL database */


    function user_projects($user_id, $connection) { // requires username and link to db

        if( (!isset($user_id)) OR (!isset($connection)) ) {
            $error = "Invalid parameters";
            return (array(0, $error));
        }

        $user_name = get_user_name($user_id, $connection);
        $admin = can_create_project($user_name, $connection);

        if($admin == 1)
        {

            $query = sprintf("SELECT project_id, project_title, project_code, project_submitted, project_approved, project_published FROM Project");
        }

        else
        {

            $query = sprintf("SELECT Project.project_id, Project.project_code, Project.project_submitted, Project.project_approved, Project.project_published FROM Project INNER JOIN Work ON Work.project_id=Project.project_id WHERE Work.user_id=%s", mysql_real_escape_string($user_id));

        }


        $result = mysql_query($query, $connection);

        if(!$result) { // query failed
            $error = "Query failed";
            return (array(0, $error));
        }


        $i = 0;
        while( $row = mysql_fetch_assoc($result) ) { // expecting multiple rows from query



            foreach($row as $name => $value) {
                $project[$name][$i] = $value;
            }

            $i++;
        }

        return (array(1, $project));
    }

?>
```

```php
<? php

/*********************************************************************************************** WRITE SHELF*/
/* in file: conversion/conversion.inc.php */
/* creates the “shelf.xml” file that is downloaded to the browser
    filename – the path of where the temporary “shelf.xml” file should go
project_id – the unique id of the project that the “shelf.xml” file is being written for
    connection – connection to the MySQL database*/


function write_shelf($filename, $project_id, $connection)
{

    if((!isset($filename)) OR (!isset($project_id)) OR (!isset($connection)) ) {
        $error = "Invalid parameters";
        return (array(0, $error));
    }

    // calling get_project in order to retrieve project_title, project_author, company_id/name, project_chapters
    list($check, $project_info) = get_project_by_id($project_id, $connection);
    if(strcmp($check,"0") == 0)
    {
        $error = "Problem retrieving project title.";
        return(array(0, $error));
    }
    // get useful info from project_info
    $project_title = $project_info["project_title"];
    $project_author = $project_info["project_author"];
    $project_company = $project_info["company_name"];

    // call list_chapters to retrieve project chapter numbers
    list($check, $chapter_info) = list_chapters($project_id, $connection);
    // see below -- get useful info from chapter_info in for loop


    //xml tags
    $shelf_xml = "<shelf>";
    $shelf_xml .= "<book>";
    $shelf_xml .= "<ID>" . $project_id . "</ID>" ;
    $shelf_xml .= "<title>" . $project_title . "</title>" ;
    $shelf_xml .= "<author>" . $project_author . "</author>";
    $shelf_xml .= "<company>" . $project_company . "</company>";
    $shelf_xml .= "<chapters>" ;

    // for loop to list the chapters in the book
    foreach($chapter_info['chapter_number'] as $index => $chapter_number) {
        $shelf_xml .= "<n>" . $chapter_number . "</n>";
    }

    $shelf_xml .= "</chapters>";
    $shelf_xml .= "</book>" ;
    $shelf_xml .= "</shelf>" ;


    //writing to chapter_text to file
    $file = fopen($filename, "w+");
    $check = fwrite($file, $shelf_xml);
    fclose($file);

}

?>
```

```php
<?php

/*********************************************************************************************** ZIP_ME*/
/* in file: conversion/conversion.inc.php */
/* zips a temporary folder of shelf and chapter files
    folder_name - the name of the folder to be zipped */


function zip_me($folder_name) {

    // Adapted from:
    // http://www.ens.ro/2012/04/06/create-zip-archive-from-directory-in-php/

    $source_dir = 'temp/' . $folder_name . '/';
    $zip_file = './' . $folder_name . '.zip';
    $file_list = Utils::listDirectory($source_dir);


    $zip = new ZipArchive();
    if ($zip->open($zip_file, ZIPARCHIVE::CREATE) === true) {
      foreach ($file_list as $file) {
        if ($file !== $zip_file) {
          $zip->addFile($file, substr($file, strlen($source_dir)));
        }
      }
      $zip->close();
    }
}

?>
```