

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

Web Based Warehouse Management System

Yang Yang

Thesis of Information Technology Degree Programme

Bachelor of Engineering

KEMI 2012

PREFACE

Thank you Mr. Antti Niemelä and Mr. Thai Bui for all valuable assistance and support that you have offered to this project.

ABSTRACT

Kemi-Tornio University of Applied Sciences, Technology	
Degree Programme	Information Technology
Name	Yang Yang
Title	Web Based Warehouse Management System
Type of Study	Bachelor's Thesis
Date	13 March 2012
Pages	38 + 5 appendices
Instructor	Thai Bui, Antti Niemelä

The target of this project was to develop a web based warehouse management system for an electronics laboratory in Kemi-Tornio University of Applied Sciences. The web-based WMS supports to manage multiple electronic component types, such as resistors, capacitors, transistors, diodes, coils, MCUs, which includes different attributes, Barcodes and locations.

This system contains two kinds of user groups. The first user group is the normal user who has authority to view the components list, borrow or return components, search for components and check transaction records. The second group is the administrator, who has much more authority in this system. An administrator can register users, change a user status (like disable a user or upgrade a user to an administrator), and he can delete components which normal users cannot do.

The final outcome is satisfactory, which not only meets the high requirements but also gets positive feedback from clients.

Keywords: Barcode, HTML5, CSS3, Bootstrap, jQuery, DataTables, PHP.

TABLE OF CONTENTS

PREFACE.....	I
ABSTRACT.....	II
TABLE OF CONTENTS.....	III
EXPLANATION OF CHARCTERS AND ABBREVIATIONS	V
1. INTRODUCTION.....	1
1.1. Warehouse management system	1
1.2. Web based application	1
1.2.1. Benefits of web based applications	2
1.3. Motivation and Objectives	3
1.4. Target users	3
2. TECHNOLOGY AND SOFTWARE ENVIRONMENT	4
2.1. Technology.....	4
2.1.1. HTML5	4
2.1.2. CSS3.....	6
2.1.3. PHP	7
2.1.4. JavaScript.....	8
2.1.5. jQuery.....	8
2.1.6. AJAX	8
2.1.7. Bootstrap	8
2.1.8. jQuery UI.....	9
2.1.9. DataTables.....	9
2.1.10. Jeditable	10
2.1.11. Barcode	10
2.2. Software Environment	10
3. REQUEMENTS AND ANALYSIS.....	11
3.1. Requirements.....	11
3.2. Analysis.....	11
3.2.1. Use-case diagram	11
3.2.2. Database EER-model	12

3.2.3.	Web structure	13
4.	IMPLEMENTATION	16
4.1.	System Architecture	16
4.2.	User Interface	17
4.2.1.	Page navigation bar	18
4.3.	Page Implementation.....	20
4.3.1.	Barcode page	20
4.3.2.	Component page.....	24
4.3.3.	Transaction page	29
4.3.4.	Insert page	30
4.3.5.	Barcode generator page	33
4.3.6.	User management page	35
5.	CONCLUSION	37
6.	REFERENCES.....	38
7.	LIST OF APPENDICES	1

EXPLANATION OF CHARCTERS AND ABBREVIATIONS

HTML5	the fifth revision of the HTML standard
CSS3	the last revision of Cascading Style Sheets
Bootstrap	front-end toolkits from Twitter
JQuery	a JavaScript library
DataTables	a plug-in for the jQuery JavaScript library
PHP	server-side HTML embedded scripting language
MySQL	a famous relational database management system
Apache	a web server
JavaScript	a prototype-based scripting language
JQuery UI	jQuery user interface library
AJAX	Asynchronous JavaScript XML
WMS	warehouse management system

1. INTRODUCTION

This part describes the background information of web based WMS.

1.1. Warehouse management system

A warehouse management system whose primary goal is to manage the movement and storage of products within a warehouse, and handle the connected transactions is a main part of the supply chain. WMS also manages the stock based on real-time information about the status of products and storage locations. There is no doubt that WMS is extremely significant for most businesses. WMS can be an independent system or modules of an ERP (Enterprise resource planning) system or supply chain execution suite, which provides an efficient, productive and accurate way to manage a warehouse.

WMS is not only the barcode and RF or batch processing, but also a lot more than inventory management, transportation management, order management and integrated with the accounting systems. Nowadays warehouse management systems are evolving into warehouse based enterprise resource planning system./1/

1.2. Web-based application

A web-based application means that any web browser can be a tool to use the application through the internet. Web-based applications have been evolving conspicuously over these years, especially based on some new technologies like AJAX, HTML5 and cloud computing.

Most of the web applications use client-server architecture. Figure 1 shows an example of WMS client-server architecture. There is a huge number of web-based applications today

such as eBay, Google search, Gmail, Facebook and Twitter. People can do most work via a web browser nowadays.



Fig. 1. WMS client-server architecture /2/

1.2.1. Benefits of web-based applications

This part lists benefits of web-based applications.

1. Never installed and easy to update

The client devices do not need to install any applications, because the web browser is the tool. Updating application on server side instead of patching each client device, the user always uses the latest one.

2. Cross platform capability

Any devices with any operating systems which include a web browser can use the applications.

3. Secure stored data

It is easy to manage different user groups by defining different accesses, which means that data can be stored and secured through the internet.

4. Reduced costs

Lower requirements on the end user system and simplified architecture bring economic benefits.

1.3. Motivation and Objectives

There is a good chance to help our school's electronics laboratory to develop a warehouse management system for managing electronic components. The system mainly uses HTML, PHP and JavaScript to provide services. Consequently, clients can use any kinds of devices with a web browser to manage information on the server. It offers a user-friendly, mobile-friendly, web-based user interface which is efficient to check in-store components, borrow or return components and record user actions by using HTML5, jQuery, AJAX and Bootstrap.

1.4. Target users

The target users of this system are the students and staffs of Kemi-Tornio University of Applied Sciences who need to use the electronics laboratory. All enrolled users are able to use the system.

2. TECHNOLOGY AND SOFTWARE ENVIRONMENT

2.1. Technology

The technologies serve the project demand below.

2.1.1. HTML5

Hypertext Markup Language (HTML) is the basic element of a web page. HTML5 is the newest vision of HTML but still in developing. The most modern web browsers support HTML5 well such as IE, Safari, Chrome, Firefox, Opera and so on. HTML5 supports many new features like new HTML elements (table 1), new attributes, full CSS3 support, video and audio, 2D/3D graphics, local storage and local SQL Database that can bring huge benefits for both developers and users./3/

Table 1. New HTML5 semantic/structural elements /4/

Tag	Description
<article>	Defines an article
<aside>	Defines content aside from the page content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<command>	Defines a command button that a user can invoke
<details>	Defines additional details that the user can view or hide
<summary>	Defines a visible heading for a <details> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

<figcaption>	Defines a caption for a <figure> element
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<hgroup>	Groups a set of <h1> to <h6> elements when a heading has multiple levels
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<ruby>	Defines a ruby annotation (for East Asian typography)
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<rp>	Defines what to show in browsers that do not support ruby annotations
<section>	Defines a section in a document
<time>	Defines a date/time
<wbr>	Defines a possible line-break

After Apple Inc. decided to abandon the supporting of Flash player for their mobile browser in iOS, the increasing numbers of websites started to use HTML5 especially for media websites. The reason to use HTML5 is that HTML5 really has more advantages than HTML4. Definitely, HTML5 has a big future. Figure 2 presents HTML5 page architecture.



Fig. 2. HTML5 page architecture /5/

2.1.2. CSS3

CSS (Cascading Style Sheets) is a style sheet language for describing the framework and layout of Web pages, including colors, layout, and fonts. Like HTML5, CSS3 is also not a W3C standard yet, but all the newest browsers are already supporting many of the new features. Figure 3 shows the supported features of each browser.

	MAC						WIN						
													
	5.1	8	9	11.1	15	17	5.1	6	7	8	9	8	15
RGBA	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
HSLA	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Background Size	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Multiple Backgrounds	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Border Image	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
Border Radius	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Box Shadow	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Text Shadow	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
Opacity	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
CSS Animations	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✓	✓
CSS Columns	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
CSS Gradients	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
CSS Reflections	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓
CSS Transforms	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
CSS Transforms 3D	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓
CSS Transitions	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
CSS FontFace	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FlexBox	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✓	✓

Fig. 2. CSS3 for each browser /6/

2.1.3. PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a server-side language that is designed for web development to implement dynamic Web pages and can be embedded into HTML. The difference between PHP and client-side language like JavaScript is that JavaScript code executes on client browser directly but PHP code executes on server before sending the result to browser./7/

2.1.4. JavaScript

JavaScript is one of the most important languages for developing a dynamic website, and it is widely used by most websites or web-based applications. It is a scripting language, which provides client-side interactivity via web-browser directly. It is also an object based functional language. In addition, JavaScript is running from top to bottom.

2.1.5. jQuery

jQuery is a JavaScript Library. It provides a simple way to use JavaScript. jQuery includes many APIs, which are really good and effective. jQuery can do most works that JavaScript did, but with less code. jQuery can really save time for developing a dynamic website. The learning curve is short, and the official website offers very readable documentation. The huge high quality jQuery plugins on the internet can serve many demands of a web application project.

2.1.6. AJAX

AJAX (Asynchronous JavaScript and XML) is not a kind of computer language, but a skill to combine some technologies, which requests server in the background and then display result information on the current page. This technology started appearing in 2000. MS Outlook Web Access and Google Suggest made it popular in 2005.

“document.getElementById("myDiv").innerHTML=xmlhttp.responseText;” this kind of JavaScript code is the typical part of AJAX.

2.1.7. Bootstrap

Bootstrap is a front-end toolkit for developing web applications, which was made by Twitter. Bootstrap is powerful but light and fast, which includes many features like

HTML5 elements, CSS3 framework, useful jQuery plugins, and responsive web design and cross platform./8/

2.1.8. jQuery UI

jQuery UI is one of famous jQuery plugins, which includes many useful UI widgets such as Autocomplete, Datepicker, Dialog, Progressbar... It is suitable for building a highly interactive web application. jQuery UI provides many colorful themes, which is customizable. jQuery UI supports well in IE 6.0+, Firefox 3+, Safari 3.1+, Opera 9.6+, and Google Chrome./9/

2.1.9. DataTables

DataTables is one of wonderful plugins for jQuery JavaScript library, which will add advanced interaction controls to any HTML table. Many big companies such as USA TODAY and Amazon are using DataTables. Moreover, it is free. Figure 3 shows an example of DataTables. DataTables can bring many features for a basic HTML table, such as search function, sortable column, variable length pagination, jQuery UI theme support./10/



The screenshot shows a DataTables table with the following data:

Rendering engine	Browser	Platform(s)	Engine version	CSS grade
Gecko	Firefox 1.0	Win 98+ / OSX.2+	1.7	A
Gecko	Firefox 1.5	Win 98+ / OSX.2+	1.8	A
Gecko	Firefox 2.0	Win 98+ / OSX.2+	1.8	A
Gecko	Firefox 3.0	Win 2k+ / OSX.3+	1.9	A
Gecko	Camino 1.0	OSX.2+	1.8	A
Gecko	Camino 1.5	OSX.3+	1.8	A
Gecko	Netscape 7.2	Win 95+ / Mac OS 8.6-9.2	1.7	A
Gecko	Netscape Browser 8	Win 98SE+	1.7	A
Gecko	Netscape Navigator 9	Win 98+ / OSX.2+	1.8	A
Gecko	Mozilla 1.0	Win 95+ / OSX.1+	1	A

Additional features visible in the screenshot include: 'Show 10 entries', a search box, and pagination controls (First, Previous, 1, 2, 3, 4, 5, Next, Last).

Fig. 3. Datatables /10/

2.1.10. Jeditable

Jeditable brings a simple way to create in place editing fields with jQuery, which means any visible text is editable by using Jeditable. Combining DataTables and Jeditable together makes cells of a HTML table editable. That brings excellent experience.

2.1.11. Barcode

This system uses unique barcode to mark different components. The barcode is a very important part of a WMS, which highly improves work efficiency for management. This system not only can use barcode scanners to read and search for components, but also can generate barcode labels and then print them.

2.2. Software Environment

WAMP is an integrated development environment, which includes Apache, MySQL and one of PHP, Perl or Python on a Windows-based computer. Apache is a free web server. MySQL is an open-source database. PHP is a scripting language, which supports to generate dynamic web pages. Perl and Python have similarly effect as PHP; all of them are server side language. This project is using PHP.

3. REQUIREMENTS AND ANALYSIS

3.1. Requirements

The requirements specifications as in Appendix 1 are from Antti Niemelä who is the project commissioner. The system supports to manage multiple electronic component types, such as resistors, capacitors, transistors, diodes, coils, MCUs, which includes different attributes, Barcodes and locations. It will also record user transactions like borrow or return. Two user groups provide different authorities to manage components and users. The system is easy to insert new component types, component attributes, components and locations. It supports barcode scanner to read barcode. Moreover, a function can generate barcode labels. A search function allow user to search for components.

3.2. Analysis

The system analysis is based on requirements through using some technical skills. This chapter will point out the important designs of the system.

3.2.1. Use-case diagram

Figure 4 illustrates all functions of this system for both user and administrator. As it shows, the normal user has the permission to manage his personal information, view components, read components via barcode reader, search for components, edit components, borrow or return components, check transaction records, create new components or generate barcode labels. The administrator who can register user, change user status like disable a user or upgrade a user to administrator, delete components has much more authority comparing with normal users.

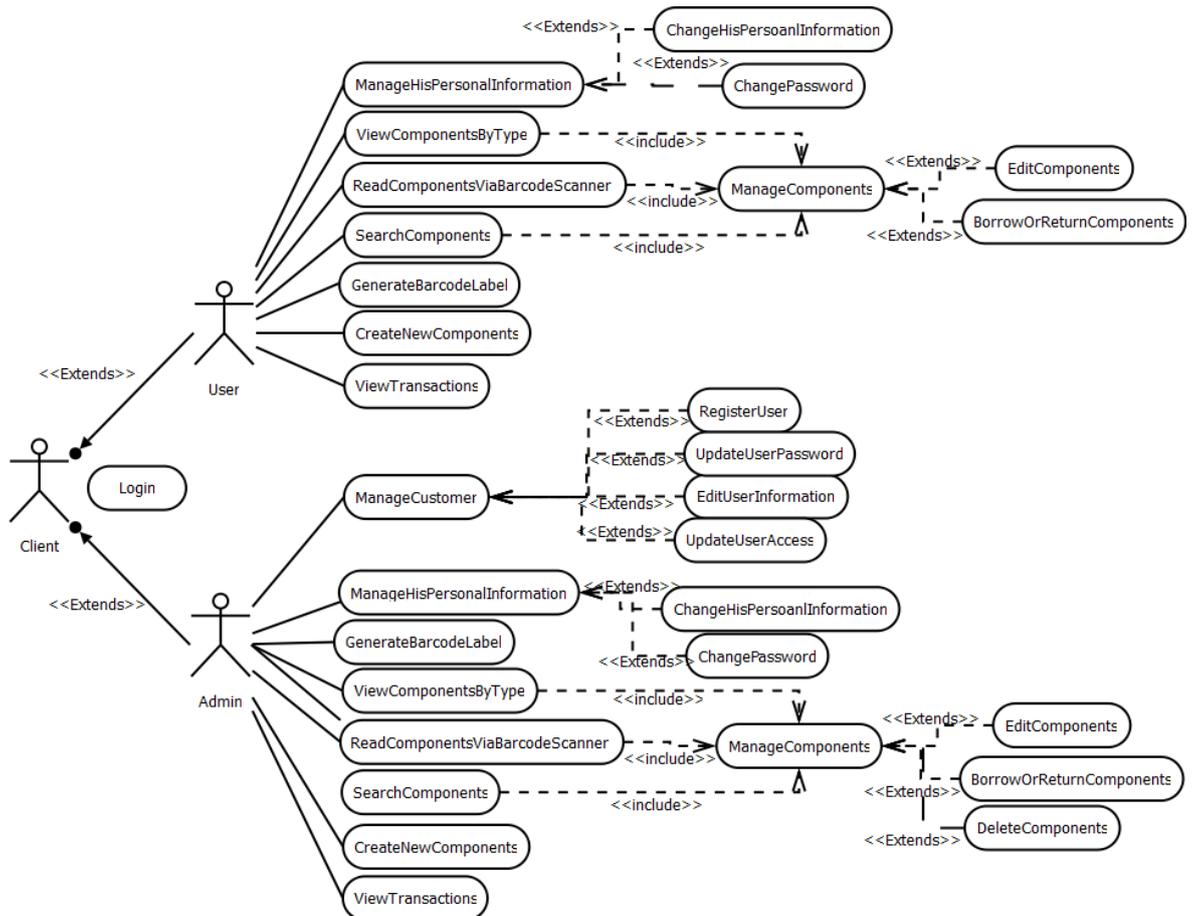


Fig. 4. Use case

3.2.2. Database EER-model

Using enhanced entity-relationship (EER) model is a good way to design a database. The EER model design (Figure 5) is based on requirements and use case diagram. The figure shows the relationship between each table. “Users” table records the information of users. Registered users have three kinds statuses which are 0(unable user), 1(normal user) and 2(admin). The transaction table includes time, user id, status (borrow or return), location id, component and number. The “Component has location” table records the amounts and locations of a component. The system supports multiple locations for a component. The component table records the basic information of components. The “type has attributes” table contains the component type and attributes of a component.

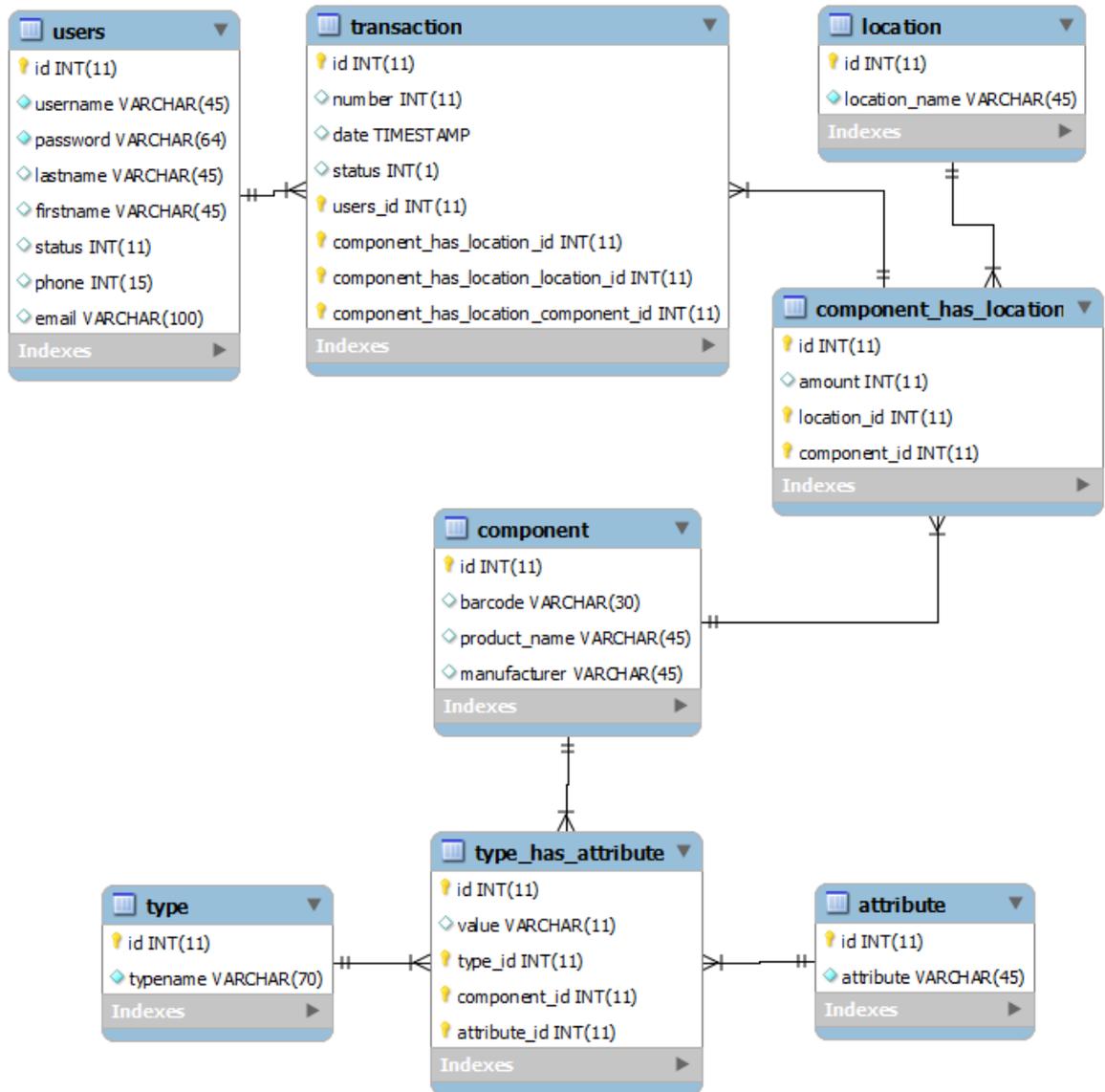


Fig. 5. EER model

3.2.3. Web structure

The web structure (Figure 6) illustrates all the web pages of this WMS:

1. A barcode page can use barcode scanner to read barcode or use a keyboard to input barcode, then the page will display the component based on this barcode and the user can borrow or return it.

2. A component management page can search for components based on the component type, barcode, product name, attributes or locations. Users can modify the whole information of these components and borrow or return it. Administrators have access to delete these components.
3. A transaction page can display records of each transaction.
4. An insert page can insert multiple components, component types, locations or component attributes to the database at once.
5. A user management page of administrators can provides a platform to administrators that create a new user account and modify normal users' information, such as name, phone number, email, password, account status etc...
6. A barcode generator page can create hundreds of barcode images at once and a print button allows printing barcode labels without other elements of this page.
7. A page named "my account" allows the user to update his personal information and password.

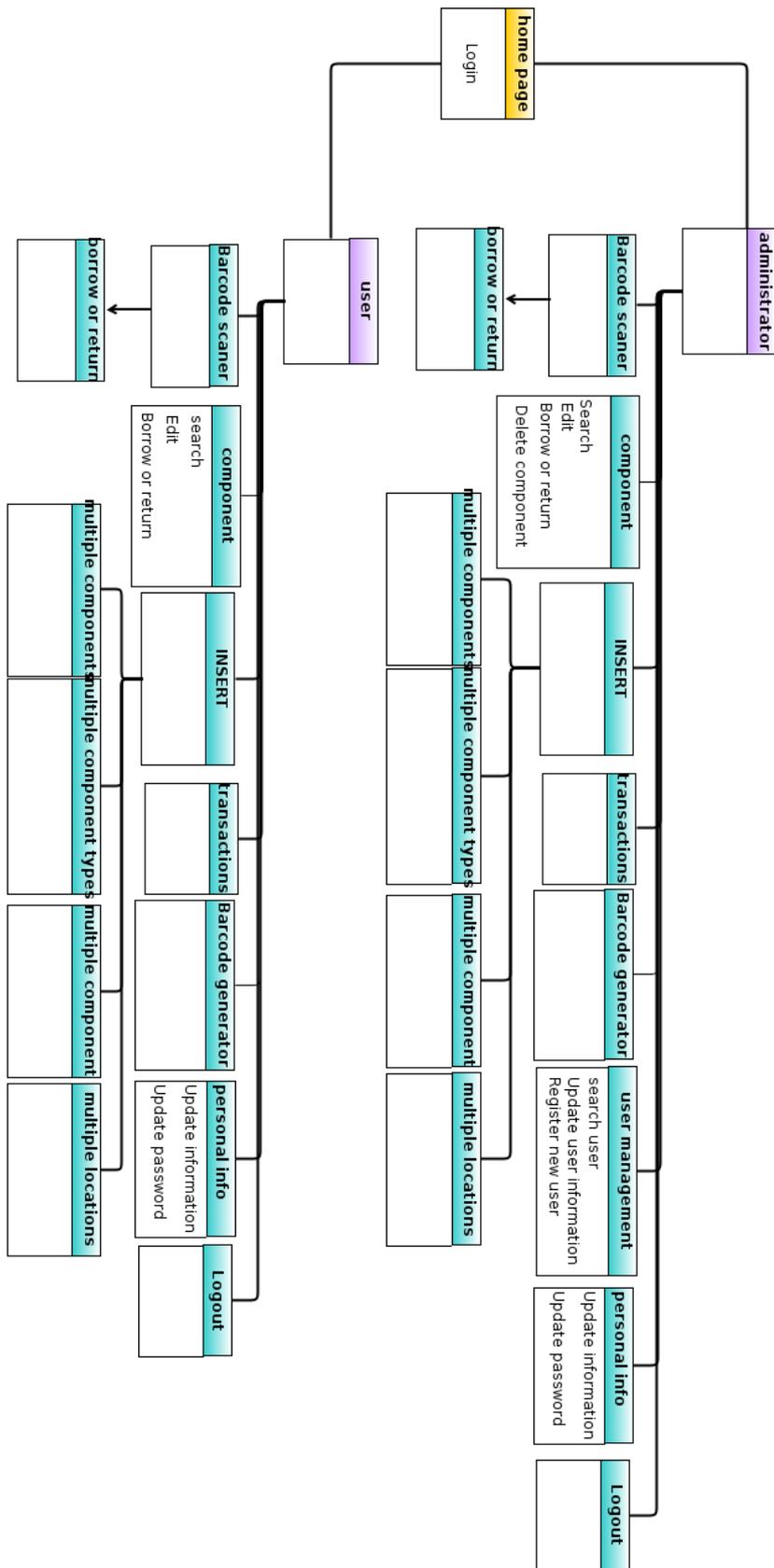


Fig. 6. Sitemap

4. IMPLEMENTATION

4.1. System Architecture

This WMS is based on client-server architecture, similar to most of web applications. Figure 7 shows a simple case of this WMS that includes web servers with Apache and PHP and a database server with MySQL.

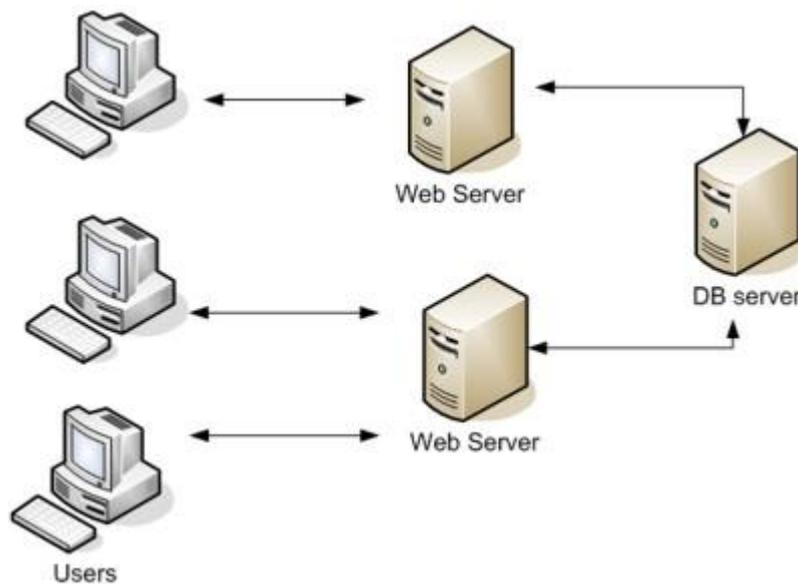


Fig. 7. Client-server architecture /11/

The first step of implementation is to build a database according to the EER model. The way by using MySQL Workbench to design an EER model is easy to export SQL file, then import the file to MySQL server. Thus, the first step is well done.

4.2. User Interface

The second important step is to design a user interface. The user interface allows users to communicate with the server. The user interface is a framework that uses HTML, CSS and JavaScript to display the layout of a page. This system mainly uses Bootstrap to provide UI. Bootstrap is a friendly front-end toolkit for developing web applications. Bootstrap is powerful but light and fast, which includes many features like HTML5 elements, CSS3 framework, useful jQuery plugins, and responsive design and cross platform.

Bootstrap is an open-source toolkit from Twitter, which means that it is free to develop a web application outlook as good as Twitter.com if we use Bootstrap. Figure 8 shows the login page of WMS, it based on Bootstrap. This page is a case of using Bootstrap; it needs include three files to work, here is the code below:

```
<link href="css/bootstrap.css" rel="stylesheet">  
<link href="css/bootstrap-responsive.css" rel="stylesheet">  
<script src="js/bootstrap.min.js"></script>
```

Define HTML body background image:

```
<body style="background-image:url(img/12.jpg);background-  
attachment:fixed;background-position: center;">
```

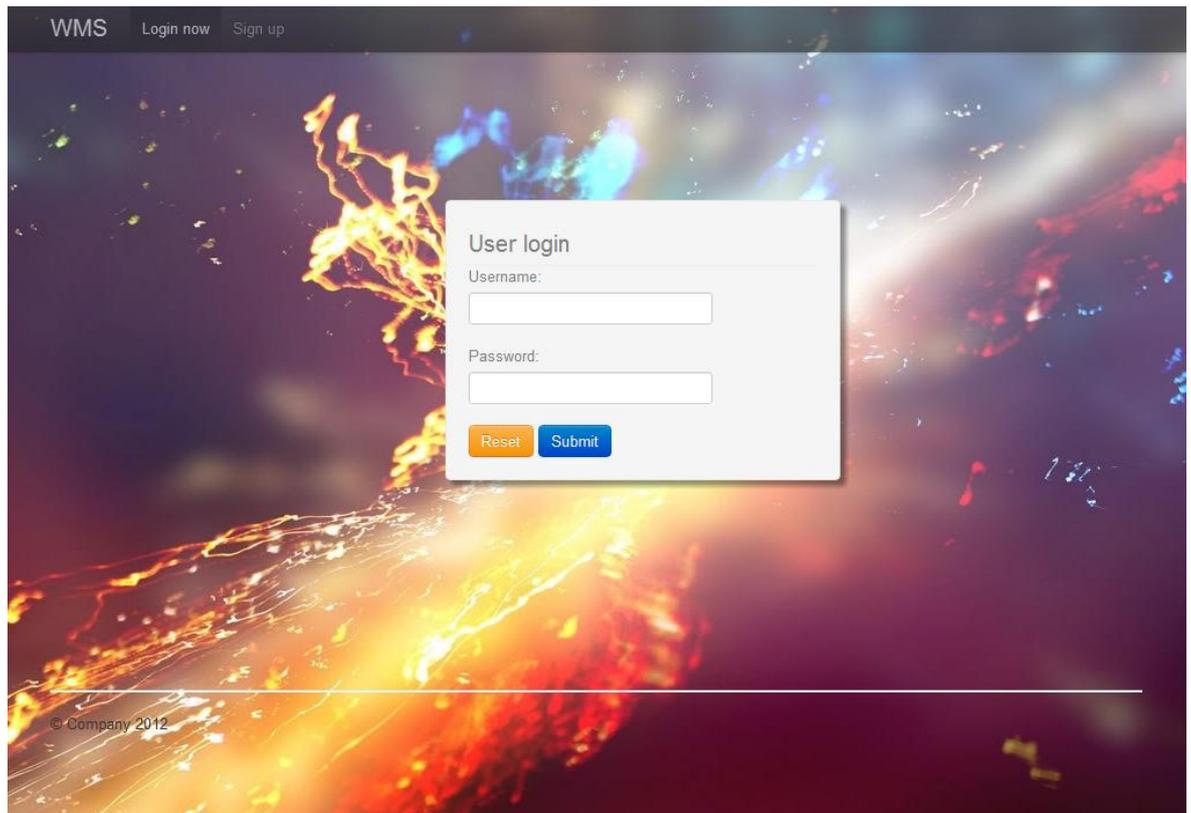


Fig. 8. Login page

4.2.1. Page navigation bar

Figure 9 shows the main navigation bar of this WMS. What the code shows in Appendix 2 is a page navigation bar in Bootstrap. The PHP code defines the user management page only display for administrators. CSS class navbar-fixed-top can keep the navigation bar always at the top of the page. An ul HTML element with class nav and pull-right can make the page links on the right side of the bar. A class named brand can make the special layout of a link. There is a head icon in front of My Account, which uses an i HTML element with the class icon-user and icon-white. It is easy to draw a typical and useful page navigation bar in Bootstrap.

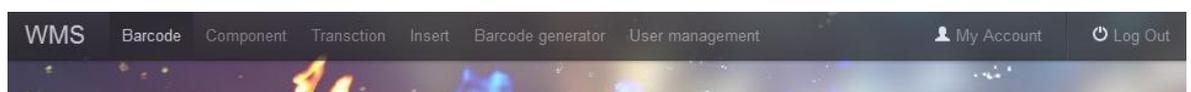


Fig. 9. Navigation bar

Toggable tab is one of JavaScript plugins in Bootstrap, which provides quick and dynamic switch from different tabs inside a single HTML page. Figure 10 shows an insert page of this system with this plugin. "Toggable tabs" uses HTML code to work, class nav and nav-tabs is needed:

```
<ul class="nav nav-tabs navz" style='border-radius: 15px;opacity: 0.9;'>
  <li class="active"><a class='a' style='border-radius: 15px;' href="#profile" data-
toggle="tab">Component insert</a></li>
  <li ><a class='a' style='border-radius: 15px;' href="#settings" data-toggle="tab">Location
insert</a></li>
  <li ><a class='a' style='border-radius: 15px;' href="#home" data-toggle="tab">Component
type insert</a></li>
  <li ><a class='a' style='border-radius: 15px;' href="#messages" data-
toggle="tab">Attribute insert</a></li>
</ul>
```

In addition, here is the code for each tab to display different contents:

```
<div class="tab-content">
  <div class="tab-pane active" id="home">...</div>
  <div class="tab-pane" id="profile">...</div>
  <div class="tab-pane" id="messages">...</div>
  <div class="tab-pane" id="settings">...</div>
</div>
```

Moreover, the CSS code define the custom layout. Appendix 3 includes the code. Figure 10 displays the beautiful result.

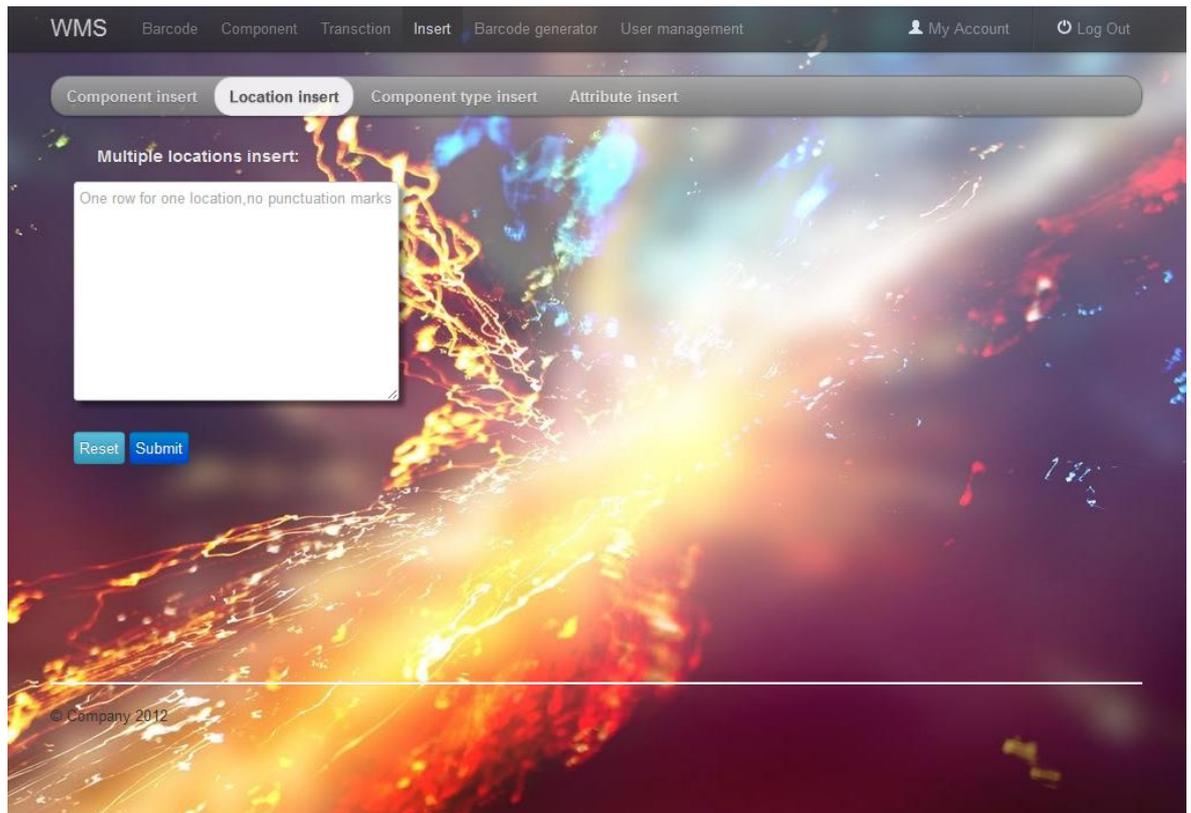


Fig. 20. Toggable tab

4.3. Page Implementation

This part describes the main procedures of implementation, which will cover many functions.

4.3.1. Barcode page

After users log in, barcode page is the home page. This page has some features:

4.3.1.1. Auto focus in jQuery

When the page has displayed, the barcode input field will become focus. It means users do not need use mouse to click the input field, just directly use barcode scanner or use keyboard to input barcode. Here is the key code in jQuery:

```
$("#input#barcode").focus();
```

4.3.1.2. A case of using icon in Bootstrap

Bootstrap has many staple icons and many ways to use them. The HTML code explains an example of using it, the effect as figure 11 shows the perfect combination of an icon and a text input element:

```
<div class="controls">
<div class="input-prepend" >
<span class="add-on" style='box-shadow: 5px 5px 5px rgba(0, 0, 0, .4);'><i class="icon-
barcode"></i></span>
<input name='barcode' id="barcode" class="span2" id="iconInput" type="text" style='box-
shadow: 5px 5px 5px rgba(0, 0, 0, .4);'>
</div>
</div>
```

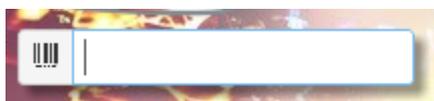


Fig. 11. Bootstrap icon

Figure 12 shows a normal text input element without Bootstrap and shadow effect.



Fig. 12. Text input element

4.3.1.3. Autocomplete

Autocomplete is a component of jQuery UI. It can list suggestions from the database when a user inputs any letters (just like using Google search). In this case, the system provides barcodes from the database based on what a user put in (figure 13). It works from the cooperation of jQuery UI and PHP. Here is the explanation below.

The first step is to include the requisite files:

```
<link rel="stylesheet"
href="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.17/themes/overcast/jquery-ui.css" />
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.16/jquery-ui.min.js"
type="text/javascript" ></script>
```

Here is the jQuery code to use Autocomplete below. The source attribute defines the Autocomplete source, which is JSON array in normal. In this case, file search.php will get source from the database. Autofocus attribute if set to true the first item will be automatically focused. The delay attribute default in 300 milliseconds, for which the system will wait 300ms after a keystroke to activate itself. Setting to 0 can speed up Autocomplete but bring more load for server.

```
$("#barcode").autocomplete({
    source: "search.php",
    autoFocus: true,
    delay: 0,
    minLength: 1
});
```

Moreover, code in search.php file to connect with database server and save the source to array:

```
<?php
$q=$_GET["term"]; // term is default by autocomplete
require_once('conn.php');
$sql="SELECT `id`, `barcode` FROM `component` WHERE `barcode` LIKE '$q%' limit
0,5";
$result = mysql_query($sql, $conn);
while($row = mysql_fetch_array($result)){
    $result1[] = array(
        'id' => $row['id'],
        'label' => $row['barcode']
    );
}
echo json_encode($result1);
?>
```

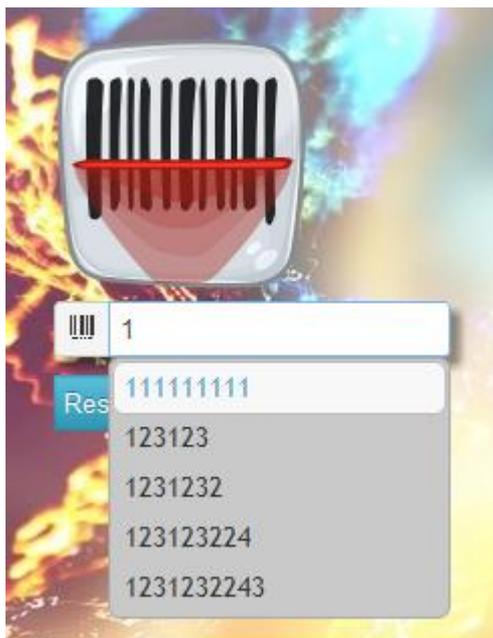
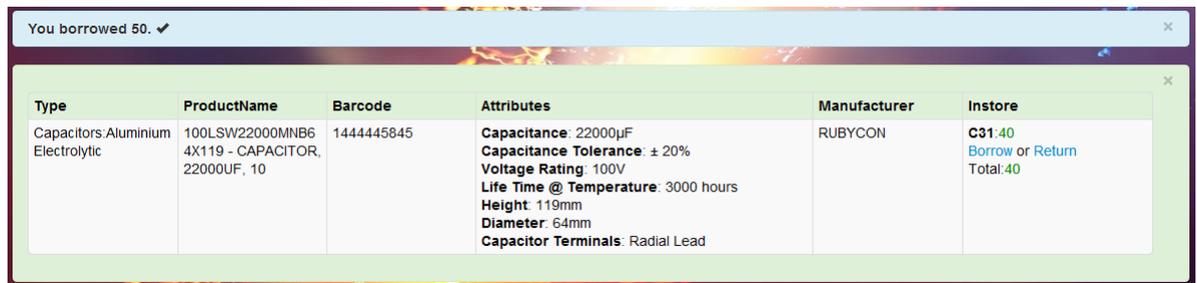


Fig. 13. Autocomplete

After a user submits the barcode, the result will display in a table (figure 14). Then the user can borrow or return it.



Type	ProductName	Barcode	Attributes	Manufacturer	Instore
Capacitors:Aluminium Electrolytic	100LSW22000MNB6 4X119 - CAPACITOR, 22000UF, 10	1444445845	Capacitance: 22000µF Capacitance Tolerance: ± 20% Voltage Rating: 100V Life Time @ Temperature: 3000 hours Height: 119mm Diameter: 64mm Capacitor Terminals: Radial Lead	RUBYCON	C31:40 Borrow or Return Total:40

Fig. 14. Submit result

4.3.2. Component page

The component page (as figure 15) can search for a component based on component type, barcode, product name, attributes or locations. After the system gets the result, the user can modify the whole information of these components and borrow or return them. Administrators have authority to delete these components.

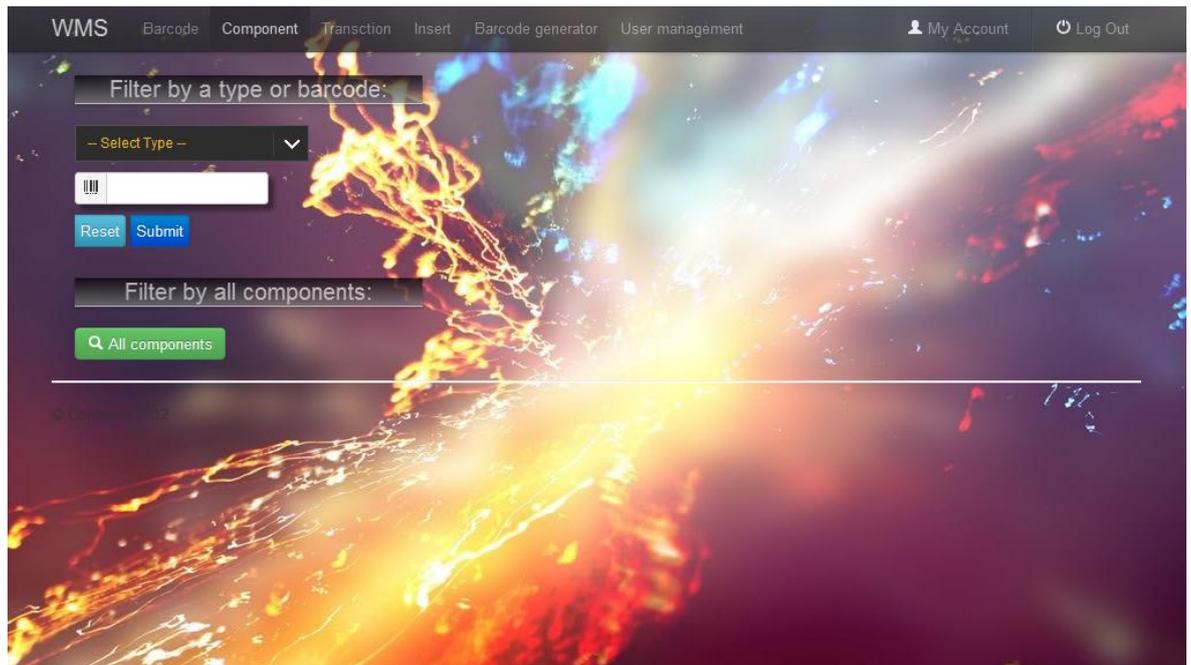


Fig. 15. Component page

Here are the features of the page:

1. AJAX select menu

Every component has a component type in this system. The AJAX select menu (as figure 16 shows) specifies components by a component type. An example for finding components of a type named “Capacitors: Aluminium Electrolytic”: first, select “Capacitors” from the menu; then, the second select menu which results to be part of “Capacitors” will display; now select “Aluminium Electrolytic” from it. After that, a table will show the components. The way by using jQuery and PHP can bring this dynamic effect. jQuery can post the changed value of the select menu to server. PHP will be in charge of the data in the server side.

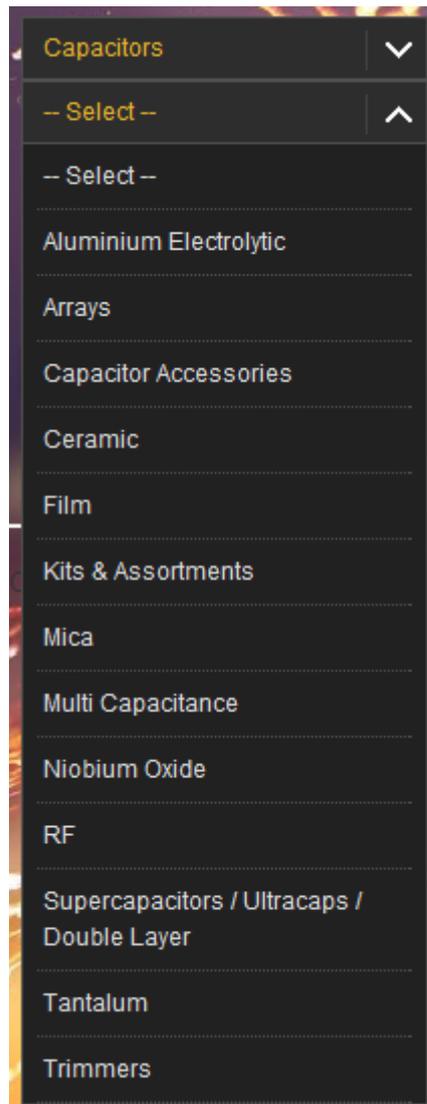


Fig. 16. Select menu

2. DataTables

DataTables makes a normal HTML table powerful. As figure 17 shows, users can sort column, define how many components display in one page or use the search function.

Here is the jQuery code to make normal HTML table to DataTables:

```
$('#datatables').dataTable({
```

```
"jQueryUI": true,
"sPaginationType": "full_numbers",
});
```

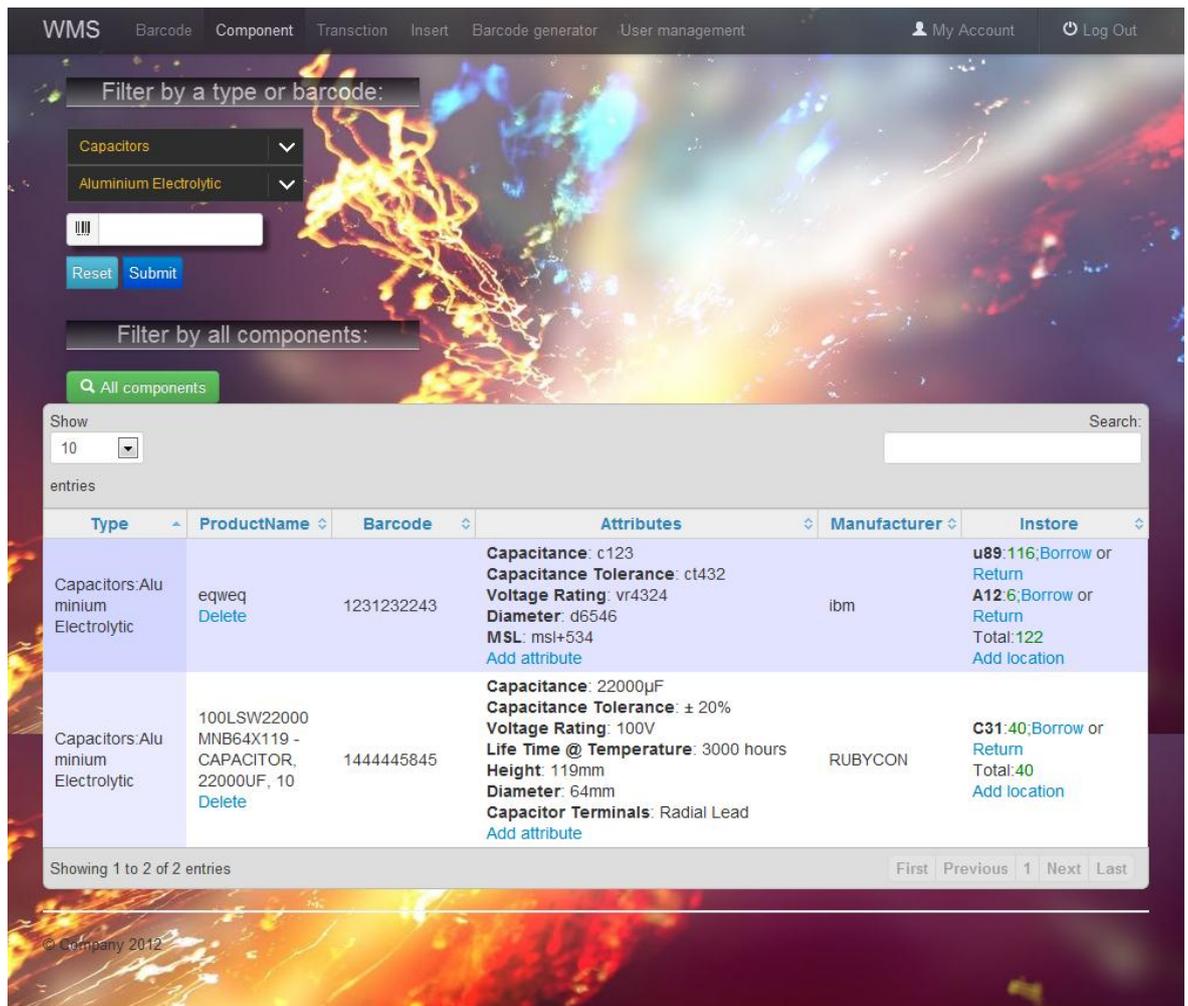


Fig. 17. Datatables

3. Jeditable

Jeditable is an in-place editor for jQuery. Figure 18 shows an example when a user clicks a barcode in the table. Using DataTables and Jeditable can create a table which has individual cells edited. One important thing is to mark every TR element

by a unique id attribute. That can tell the server which row needs updating. In addition, it uses `fnDrawCallback` to re-draw the result.

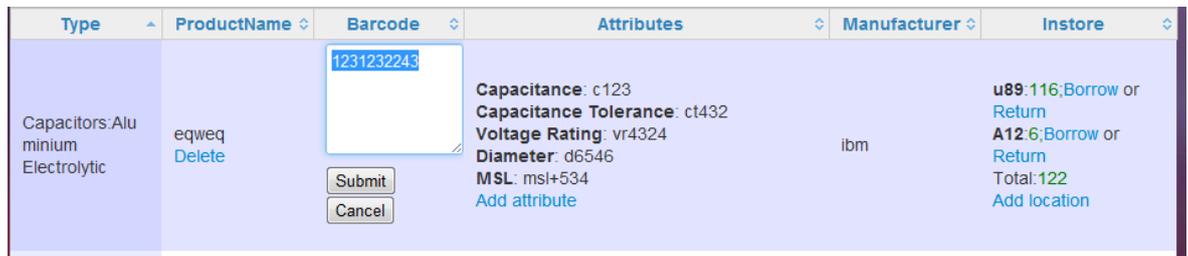


Fig. 18. Jeditable

The jQuery code shows below is an example of Jeditable. If the selected attribute is true (code below), the text inside the input area is selected as figure 18 shows. The type attribute has two options which are `textarea` and `select`. `Textarea` as figure 18 shows; `select` means select menu. Figure 19 shows an example of updating an attribute name of a component from a selection menu, whose content source is from the database. The `tooltip` attribute defines the tip information of the mouse hover event. The `cancel` and `submit` attributes defines the names of their buttons. The `indicator` attribute can define the loading image when a user submits changes at waiting time. `Height` defines the height of the input `textarea`. After a user edits any information, `componentupdate.php` will receive the data; then, this PHP file can update the data to the database.

```

$('td[name="td"]').editable( 'componentupdate.php', {
    select: true,
    type   : 'textarea',
    tooltip : "Click to edit",
    cancel  : 'Cancel',
    submit  : 'Submit',
    indicator : "<img src='js/images/indicator.gif'>",
    "height": "80px"
});

```

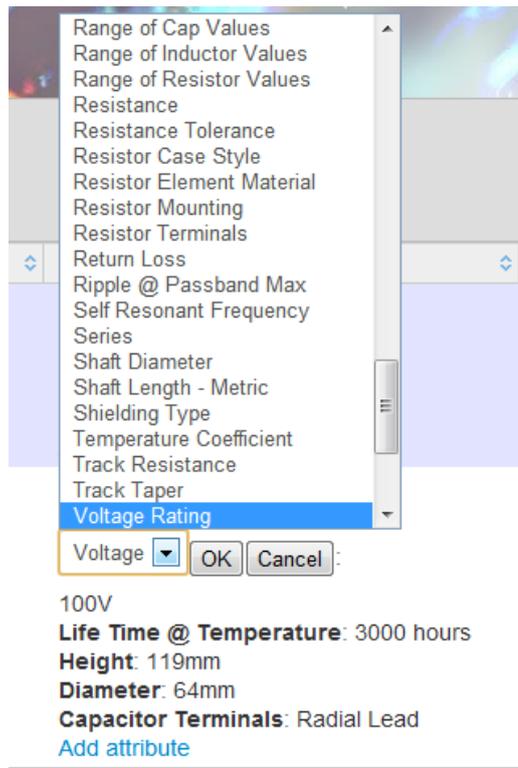
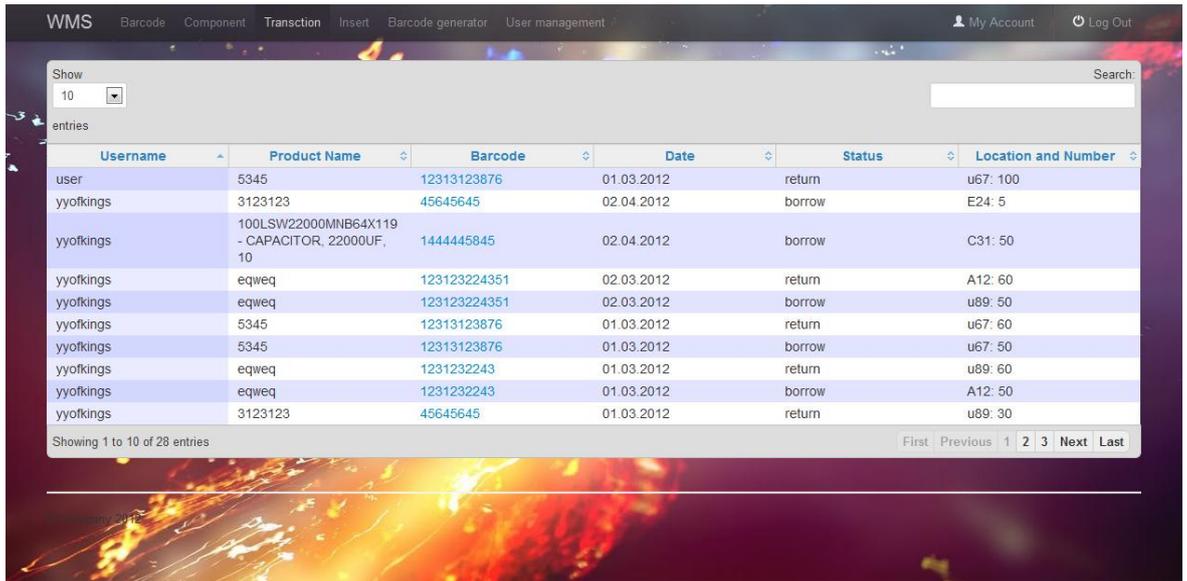


Fig. 19. Jeditable

4.3.3. Transaction page

This page (as figure 20) can display all records of users' actions, which includes username, product name, barcode, date, status, location and number.



The screenshot shows the 'Transaction' page in a WMS system. At the top, there are navigation tabs: Barcode, Component, Transaction (selected), Insert, Barcode generator, and User management. On the right, there are links for 'My Account' and 'Log Out'. Below the navigation, there is a 'Show' dropdown set to '10' and a search box. The main content is a table with the following data:

Username	Product Name	Barcode	Date	Status	Location and Number
user	5345	12313123876	01.03.2012	return	u67: 100
yyofkings	3123123	45645645	02.04.2012	borrow	E24: 5
yyofkings	100LSW22000MNB64X119 - CAPACITOR, 22000UF, 10	1444445845	02.04.2012	borrow	C31: 50
yyofkings	eqweq	123123224351	02.03.2012	return	A12: 60
yyofkings	eqweq	123123224351	02.03.2012	borrow	u89: 50
yyofkings	5345	12313123876	01.03.2012	return	u67: 60
yyofkings	5345	12313123876	01.03.2012	borrow	u67: 50
yyofkings	eqweq	1231232243	01.03.2012	return	u89: 60
yyofkings	eqweq	1231232243	01.03.2012	borrow	A12: 50
yyofkings	3123123	45645645	01.03.2012	return	u89: 30

At the bottom of the table, it says 'Showing 1 to 10 of 28 entries'. On the right side of the table, there are navigation buttons: 'First', 'Previous', '1', '2', '3', 'Next', and 'Last'.

Fig. 20. Transaction page

4.3.4. Insert page

This page allows users to insert multiple components, component types, locations or component attributes to the database at once. Here are the features of the page:

1. HTML5 placeholder for IE

Placeholder is a new element of HTML5, which supported well by most web browsers except IE. The placeholder attribute displays a hint (as figure 21) in an input field before user input values. When inputting anything, it will disappear. IE do not support this HTML5 attribute, but jQuery can bring a same effect for IE:

```

$('textarea').focus(function() {
    var input = $(this);
    if (input.val() == input.attr('placeholder')) {
        input.val("");
        input.removeClass('placeholder');
    }
})

```

```

}).blur(function() {
    var input = $(this);
    if (input.val() == "" || input.val() == input.attr('placeholder')) {
        input.addClass('placeholder');
        input.val(input.attr('placeholder'));
    }
}).blur();

```

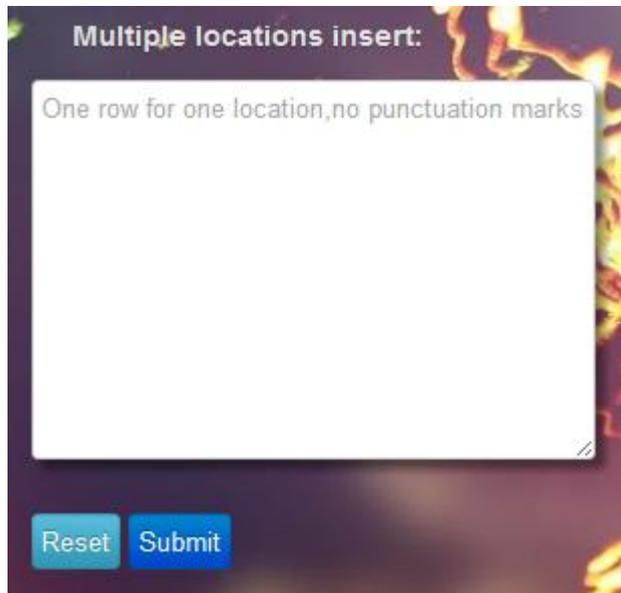


Fig. 21. Placeholder

2. Bootstrap Popovers

Popover is a Bootstrap component, which displays a mouse hover tip (figure 22).

It works with HTML and jQuery code:

```

<textarea style='box-shadow: 5px 5px 5px rgba(0, 0, 0, .4);' class="input-xlarge"
id="textarealo" rows="10" placeholder="One row for one location, no punctuation
marks" rel="popover" title="Tips" data-content="Every location with one row,
multiple input like this:<br>B24<br>D33<br>T15<br>"></textarea>

```

```
$('#textarea').popover();
```

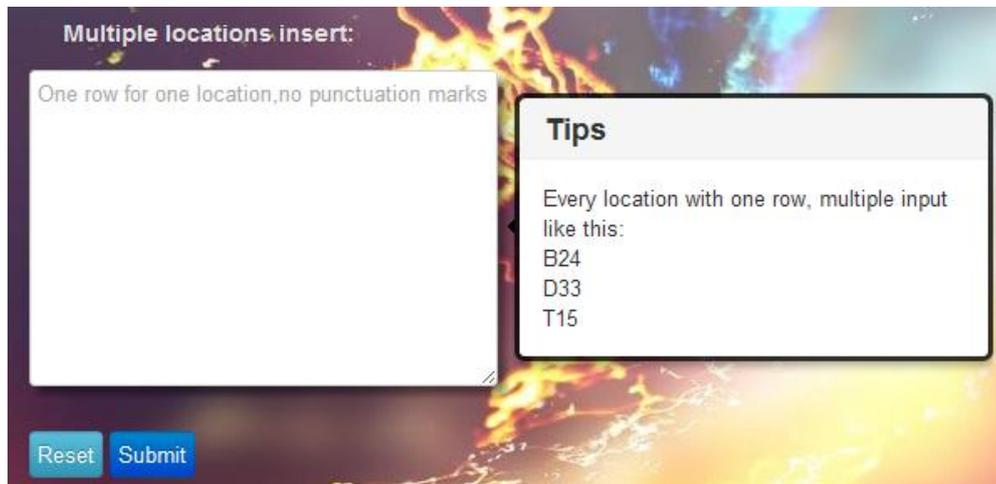


Fig. 22. Popover

3. Multiple insert

Multiple insert allows a user to insert multiple locations, component types or attributes to the database. For example, a textarea can insert multiple locations to the database at once, but each location with a new row. The idea is to post data to PHP file. Then PHP code explode every row, and save the nonempty row to the database.

The main code of multilo.php file:

```
<?php
$attr=$_POST['location'];
$a="One row for one location,no punctuation marks";
$lev = strcmp ($attr,$a );
$attr = str_replace("\r\n", "\n", $attr);
//This line can make Linux and Windows layout the same newline mark.
```

```
$attr = explode("\n", $attr);
//explode data by newline mark

if(!empty($attr) && $lev!=0)
{
require_once('conn.php');
    foreach($attr as $attr2)
    {
        if(!empty($attr2)){
            echo "Insert location: ";
            $sql="INSERT INTO `location` (`id`, `location_name`) VALUES (NULL,
'$attr2')";
            mysql_query($sql, $conn);
        }
    }
}
?>
```

4.3.5. Barcode generator page

A barcode generator page (as figure 23) supports to generate hundreds of barcode labels at a time. A print button allows the user to printing those barcode labels.

This page uses a jQuery plugin to generate barcode. This barcode generator supports many kinds of barcode types, such as ean8, ean13, code11, code39, code128, codabar, msi, datamatrix. This system only needs code128. In addition, the plugin is free to use.

Using this plugin, first to include it:

```
<script type="text/javascript" src="js/jquery-barcode-2.0.2.min.js"></script>
```

Here is the main code to generate multiple barcode labels below. Every label displays in a new div element with span3 CSS class, which can economize papers when printing barcode labels.

```
<?php

$attr=$_POST['location'];
$a="One row for one barcode";
$lev = strcmp ($attr,$a );
$attr = str_replace("\r\n", "\n", $attr);
$attr = explode("\n", $attr);

if(!empty($attr) && $lev!=0)
{
    $a=0;
    foreach($attr as $attr2)
    {
        if(!empty($attr2)){
            $a=$a+1;    ?>
            <div class="span3" id=bar<?php echo $a;?> ></div>
            //create new div for every barcode label
            <script>
            $("div#message").fadeOut('slow');
            $("div#bar<?php echo $a;?>").barcode("<?php echo $attr2;?>", "code128");
            //output barcode to barcode label with code128
            </script>
        }
    }
}
?>
```

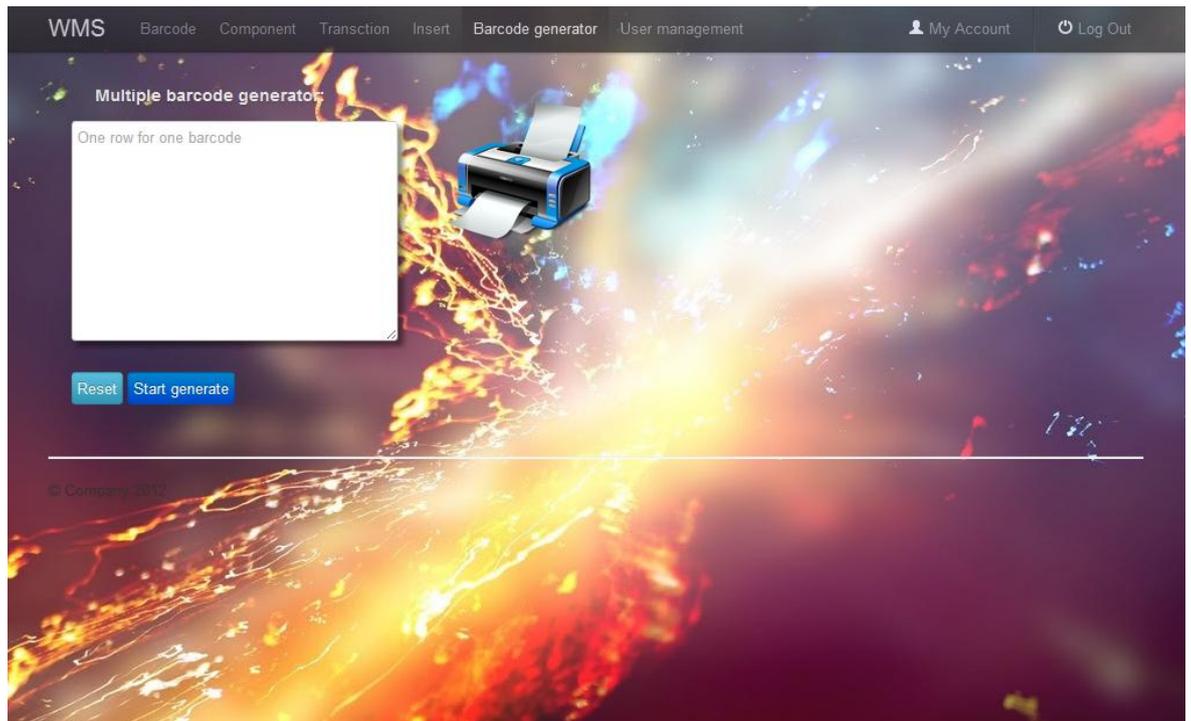


Fig. 23. Barcode generator page

4.3.6. User management page

Administrators have the access to search, update or register users through this page (figure 24). A column shows every user's account status which is closed, user or administrator. These account statuses can be displayed by using radio buttons of Bootstrap, which can update a user's account status by one click. This function uses HTML, PHP and jQuery.

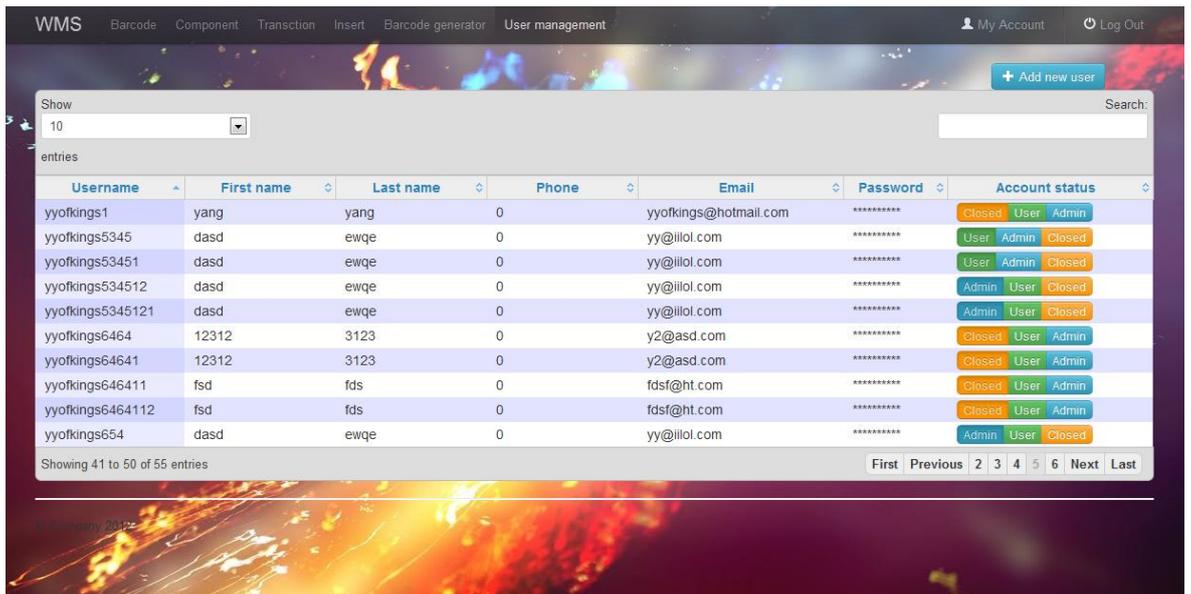


Fig. 24. User management page

5. CONCLUSION

The project focused on creating a web based WMS during three months. It uses many technologies such as HTML5, CSS3, JavaScript, jQuery, Bootstrap, barcode, and WAMP services. This project is complex with limited costs and time. The database design is not easy, because of the requirements for managing complicated electronic components. The final design of the database is suitable because of the information review in those days. jQuery is convenient and fast for web designing, which changes the way of writing JavaScript. Bootstrap 2.0 is a big surprise, thanks to Twitter and authors.

The project was accomplished successfully and many functions are beyond the blueprint. That was unimaginable at the beginning.

6. REFERENCES

- /1/ Dave Piasecki, [WWW document],
[http://www.inventoryops.com/warehouse_management_systems.htm] April 2012.
- /2/ SOLVO Ltd.,[WWW document], [<http://www.solvo.ru/en/products/wms/Solvo.WMS/>]
April 2012.
- /3/ Refsnes Data, [WWW document], [<http://www.w3schools.com/html5/default.asp>] April
2012.
- /4/ Refsnes Data, [WWW document],
[http://www.w3schools.com/html5/html5_new_elements.asp] April 2012.
- /5/ Tim Wright, [WWW document],
[<http://coding.smashingmagazine.com/2009/07/16/html5-and-the-future-of-the-web/>] 16
July 2009.
- /6/ Deep blue sky, [WWW document], [<http://findmebyip.com/litmus>] April 2012.
- /7/ The PHP Group, [WWW document], [<http://fi.php.net/manual/en/intro-what-is.php>] 30
Mar 2012.
- /8/ Twitter, [WWW document], [<http://twitter.github.com/bootstrap/index.html>] February
2012.
- /9/ JQUERY PROJECT, JQUERY UI TEAM, [WWW document], [<http://jqueryui.com/>]
February 2012.
- /10/ Allan Jardine, [WWW document], [<http://datatables.net/>] February 2012.

/11/ Gergely Orosz , [WWW document], [<http://gregdoesit.com/2009/01/monitoring-website-load/>] 20 January 2009.

7. LIST OF APPENDICES

APPENDIX1. WAREHOUSE MANAGEMENT SYSTEM FOR ELECTRONICS COMPONENTS

Description

The warehouse management system is a web-based UI to a database that contains information of all the components in storage for a quick and easy way to check, add and remove components from the storage. It also has support for a barcode scanner which can be used to add new components or to remove components.

Requirements

1. Supports multiple component types such as resistors, capacitors, transistors, diodes, coils, MCUs
 - a. Different component types have different attributes (resistors have resistance (Ohms), capacitors Faradays, coils have Henrys etc.)
2. Possibility to add new component types using a simple graphical user interface
3. An easy to use list view for components in the system
4. Possibility of running a search based on different attributes (such as component type, dimensions, solder type etc.)
5. Components can be added
 - a. For each component the user must be able to fill: dimensions, solder type, tolerances, product name, manufacturer, Farnell –code (can be added by reading barcode)
 - b. Using the barcode scanner to read the barcode on a product that has not been entered into the system open an add a component view
6. Components can be removed
 - a. Components must be able to be removed from the storage quickly and easily so that the storage status of each component stays up to date

- b. Removing the last component will mark the storage status as 0
 - c. Using the barcode to read the code of a product that is in the storage will result in a remove component view where the user can enter how many components have been removed from storage
7. Every removal or adding of a component or component type is saved into the database for usage monitoring (storage transactions)
- a. Username "abcd" takes 10 resistors from the storage -> information of this is added to the usage monitoring database
 - b. User recognition and management
 - c. User needs to log in to the system
 - d. When removing or adding the username of that person is saved to the usage monitoring database
 - e. Support for RFID user recognition (optional)
8. Possibility to see storage transactions in a list

APPENDIX2. PAGE NAVIGATION BAR

```
<?php
session_start();
if (!(isset($_SESSION['status'])) or $_SESSION['status'] ==null) {
    header ("Location: index.php");// Jump to login page, if not login yet.
}
else {
    ?>
<div class="navbar navbar-fixed-top" style='opacity: 0.7;'>
<div class="navbar-inner">
<div class="container">
<a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
<!-- display icons for small device -->
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
</a>
<a class="brand" href="#">WMS</a>
<div class="nav-collapse">
    <ul class="nav">
        <li><a name="barcode" href="barcode.php">Barcode</a></li>
        <li><a name="component" href="component.php">Component</a></li>
        <li><a name="transction" href="Transction.php">Transction</a></li>
        <li><a name="Insert" href="insert.php">Insert</a></li>
        <li><a name="generator" href="barcodeg.php">Barcode generator</a></li>
    </ul>
</div>
<?php
if ($_SESSION['status'] ==2)
//status=2, for admin
{ ?>
```

```
<li ><a name="usermanagement" href="usermanagement.php">User  
management</a></li>
```

```
<?php      } } ?>
```

```
</ul>
```

```
<ul class="nav pull-right">
```

```
<li><a name="account" href="myaccount.php"><i class="icon-user icon-white"></i> My  
Account</a></li>
```

```
<li class="divider-vertical"></li>
```

```
<li><a name="logout" href="logout.php"><i class="icon-off icon-white"></i> Log  
Out</a></li>
```

```
</ul>
```

```
</div><!--/.nav-collapse -->
```

```
</div>
```

```
</div>
```

APPENDIX3. CSS CODE

```
<style type="text/css">
.navz {
    margin: 0;
    line-height: 100%;

    box-shadow: 0 1px 3px rgba(0, 0, 0, .4);

    background: #8b8b8b; /* for non-css3 browsers */
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#a9a9a9',
endColorstr='#7a7a7a'); /* for IE */
    background: -webkit-gradient(linear, left top, left bottom, from(#a9a9a9),
to(#7a7a7a)); /* for webkit browsers */
    background: -moz-linear-gradient(top, #a9a9a9, #7a7a7a);
    /* for firefox 3.6+ */

    border: solid 1px #6d6d6d;
    border-radius: 15px;
}
.a {
    font-weight: bold;
    color: #e7e5e5;
    text-decoration: none;
    display: block;
    padding: 8px 20px;
    margin: 0;
    text-shadow: 0 1px 1px rgba(0, 0, 0, .3);
}
</style>
```