

Asset Management

By:

Brian Johnson

A Design Freeze Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Computer Science Technology

University of Cincinnati
College of Applied Science

March 2006

Asset Management

By:

Brian Johnson

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Computer Science Technology

© Copyright 2006 Brian Johnson

The contents of this document are under copyright of the author. It may not be reproduced and distributed in whole or in part without the written permission of the author.

Brian Johnson

Date

Russell McMahon

Date

Patrick C. Kumpf, Ed.D. Interim Department Head

Date

Acknowledgements

I would like to give special thanks to Erma Fritsche, department head of Library Information Technology for allowing me the chance to create my project, and all of the support she has given me. I would also like to give special thanks to Thomas Scherz, current Network Analyst for Library Information Technology for helping me learn PHP and for believing in me. In addition, I would like to give special thanks to Paul Dozier, current Network Analyst for Library Information Technology for taking me under his wing and showing me the ropes of Senior Design and the Information Technology field. I would also like to thank the rest of the staff and co-op's of the Library Information Technology department for their input on my project. I would like to thank my mom, dad and my whole family for believing in me all of these years and for their support. Finally last but not least, I would like to give a special thanks to my wonderful girlfriend Cindy for all her support and help on this project.

Table of Contents

Section	Page
Acknowledgements	3
Table of Contents	4
List of Figures	5
Abstract	6
1.Statement of the Problem	7
2.Description of the Problem	8
2.1User Profile	9
2.2Technical Elements	9
3. Deliverables	11
4. Design and Development	12
4.1 Timeline	12
4.2 Budget	13
5. Proof of Design	14
5.1 Login Page	15
5.2 Menu Page	17
5.3 Add Equipment Page	19
5.4 Edit Equipment Page	20
5.5 Remove Equipment Page	22
5.6 View Page	24
6. Testing	25
7. Conclusion	27
8. Recommendations	27
Appendix A	29
Appendix B	31
References	33

List of Figures

Figure Numbers	Page
Figure 1. Budget	14
Figure 2. Asset Management Outline	15
Figure 3. Login Page	16
Figure 4. Fill in Required Field	16
Figure 5. Incorrect Password	17
Figure 6. Menu Page	18
Figure 7. Menu Page Open	18
Figure 8. Add Equipment Page	19
Figure 9. Add Equipment List Page	20
Figure 10. Edit Equipment Page	21
Figure 11. Edit Certain Record	21
Figure 12. List Edit Equipment	22
Figure 13. Remove Equipment Page	23
Figure 14. Sure You Want to Remove	23
Figure 15. Equipment Gone	24
Figure 16. View Page	25
Figure 17. Problem Table	26

Abstract

Asset Management is a shipping and receiving Web application and database that keeps track of all the equipment that is in production and all of the equipment that is put away in the stockrooms. This web application also keeps track of all of the equipment that is received and equipment that has gone to surplus. This application has many different users that have their own user names and passwords. There are two main elements to this application: the database and the user interface. The user interface is the link to the database. Users can add, move, view, or remove equipment in the database at any time they wish. The database keeps track of where everything is and when the equipment was added. Creating this Web application database will help the University of Cincinnati libraries become more aware of where all of the equipment is located and who has it. This web application database will be used for many years to come.

Asset Management

1. Statement of the Problem

The Library Information Technology (IT) department has a problem with keeping track of its shipping and receiving. So I created a shipping and receiving database that will help the department keep things in order. Currently when Library IT receives an order, we check the invoice to make sure that the order is complete. The order is then placed in a stock room until we are ready to move it into production. Our department has many different stock rooms, and for the most part, there are no records of what is in each of the stock rooms.

Another problem is moving things from production to surplus. Surplus is a huge stock room on the bottom floor of Langsam Library. It has old, out date, and non-working equipment in it. The equipment in the surplus room will not be used anymore in the university libraries, and it will be removed from the campus. When a piece of equipment is removed from production (equipment that is currently being used), it is taken to surplus, but no record of the equipment is taken and this is a problem. Currently the department has records of all of the equipment that was or still is in production. Each computer, monitor, printer, and data jack is assigned to a specific person working within the libraries. When the person receives a new piece of equipment, the record is updated and the record of the old equipment is frequently lost. Library IT cannot account for each of the items in the stock rooms once the equipment record is updated.

2. Description of the Solution

My senior design project is a shipping and receiving Web application and database that keeps track of all the equipment that is in production in each of the libraries at the University of Cincinnati and, in each stock room. This Web application will also keep track of all of the equipment that is received or equipment that has gone to surplus. This application has a few different users that will have their own user names and passwords.

The proposed shipping and receiving database will be created in Navicat which was developed in 2001. The application allows the configuration of any user to access multiple connections to local and remote databases in Windows Explorer style, cascading open to databases, then tables, and ultimately to data. Navicat's Server Monitor allows the review of server status, process list, and server variables. The first two monitors are available in real time, with auto-refresh for troubleshooting and tracking (8).

Hypertext Preprocessor (PHP version 4.3.2) was used for the Web application portion of the project. PHP is an open source, server-side HTML scripting language used to create dynamic Web pages. PHP is embedded within tags, so the author can move between HTML and PHP instead of using large amounts of code. Because PHP is executed on the server, the viewer cannot see the code. PHP can perform the same tasks as a **Common Gateway Interface (CGI)** program and is compatible with many different kinds of databases (1,5).

This database Keeps track of:

- All equipment that is received by the Libraries
- All equipment that is in production
- All equipment that is in stock

- All equipment that is in surplus
- The location of each piece of equipment

The Web application gives Library IT access to all of the tables inside of the database.

Library IT will be able to gain web access to the application from the main site

<http://ul.uc.edu/uclit/protected/common/login/login.php> or directly to the application

<http://ul.uc.edu/uclit/johnson/common/Login/login.php>.

2.1 User Profile

Users will have their username and password, which will allow them to

- Login to the application
- Add data to the database
- Edit data from the database
- Delete data from the database
- View data from the database

Co-op students and staff of Library IT are the target users of the application; users should have some knowledge of using the Web, and must understand how the database works. No other users will use this Web application.

2.2 Technical Elements

This shipping and receiving database was created on Navicat. The application allows the configuration of any user to access multiple connections to local and remote databases in Windows Explore style, cascading open to databases, then tables and ultimately data. Navicat's Server Monitor allows the review of server status, process list, and server variables. The first two monitors are available in real time, with auto-refresh

for troubleshooting and tracking. Navicat was chosen for this project because we already have a few other databases set up with the use of this project (8, 9).

Other features of Navicat:

- Support multiple connections for local or remote MySQL servers
- Browse databases, modify your data, create and delete databases, tables, indexes and users accounts
- Create or run SQL queries with visual query builder
- Create reports with visual report builder: reports can be saved as an archive file (.raf) for backup and increased portability
- Manage user privileges [security configuration]
- Backup/restore your database, the new Schedule Backup feature is used to create a new schedule that will backup your data
- Transfer Data – Support transfer databases from one MySQL server to another MySQL server (1. Local to Remote, 2. Remote-to-Remote, 3. Remote to Local)
- Import/Export data – The Import and Export Wizard would help you to import/export data. It supports up to 18 of the most popular formats including MS Access, MS Excel, XML, PDF, and TXT (8, 9).

This project was also created with the help of MySQL (version 3.23.58). MySQL is the world's most popular open source database, with more than six million active installations. Many of the world's largest organizations, including Sabre Holdings, Cox Communications, The Associated Press, NASA, and Suzuki, are realizing significant cost saving by using MySQL to power Web sites, business-critical enterprise applications, and packaged software. MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP/ Perl/

Python), a fast growing open source enterprise software stack. MySQL will be used in this project because, right now, no other open source database equals MySQL features and benefits. Plus, we have a Linux server in the Library IT department (3, 6)

Business Benefits:

- Reduces database licensing costs by over 90%
- Cuts system downtime by 60%
- Lowers hardware expenditure by 70%
- Reduces administration, engineering and support costs by up to 50%
- Delivers less complicated solutions that complement existing corporate database such as Oracle, IBM DB2, and Microsoft SQL Server (3, 6, 7)

PHP was also a key factor in creating this project. PHP is an open source, server-side HTML scripting language used to create dynamic Web pages. PHP is embedded within tags, so the author can move between HTML and PHP instead of using large amounts of code. Because PHP is executed on the server, the viewer cannot see the code. PHP can perform the same tasks as a CGI program, and is compatible with many different kinds of databases. PHP will be used in this project because the Library IT department already has PHP based Web sites, plus the department agrees that PHP would be the best choice (1, 5, 10)

3. Deliverables

During the designing phase, certain key items were necessary to make this project come to life. The deliverables are what my project will offer. Following is a list of features that my project offers.

1. This Web application database will be part of the libraries resource page.

2. The user interface will be written in PHP.
3. Users will have their own usernames and passwords to login, to the site.
4. The PHP pages will be directly linked to the database
5. The database will be created on Navicat.
6. Users will be able to:
 - a. Login to the application
 - b. Add data to the database
 - c. Edit data from the database
 - d. Delete data from the database
 - e. View data from the database

4. Designs and Development

This section describes my project's timeline, and overall budget with software cost.

4.1 Timeline

Senior Design I Spring of 2005

- Research (4/10 – 5/1)
- Collecting Data (5/2 – 5/15)
- Finalize Project (5/16)
- Senior Design I Presentation (5/30)

Summer of 2005

Worked On:

- Database Part I (6/1 – 6/15)
- Login Page (6/16 – 7/1)
- Database Part II (7/2 -7/15)

- View Page (7/16 – 7/30)
- Add Equipment Page (8/1 – 8/30)
- Remove Page (9/1 – 9/20)

Senior Design II Fall of 2005

- Style Sheet (9/23 – 10/10)
- Test all Web (10/11 – 10/20)
- Test all Databases (10/21 – 11/10)
- Senior Design II Presentation (11/28)

Senior Design III Winter of 2006

- Finish Work on Project (1/1 – 2/1)
- Debug (2/1 – 2/10)
- Other Test
- Complete Project (2/22)
- Senior Design III Presentation (3/6)

4.2 Budget

The budget is shown in the table below. I had licenses for all of the necessary software for this project, so this table shows the prices of real world cost.

<i>Software</i>	<i>Explanation</i>	<i>Cost</i>	<i>In office Yes / No</i>	<i>Total Cost</i>
MySQL Version (3.23.58)	License	\$295.00	Yes	
Navicat	License	\$95.00	Yes	
PHP Version (4.3.2)	Free	Free	Yes	
Internet /Firefox	Free	Free	Yes	
				\$390.00

Figure 1: Budget

5. Proof of Design

This section shows in detail how the deliverables for my project were met, and what problems I came across. The outline for the Web application can be found below (See Figure 2.).

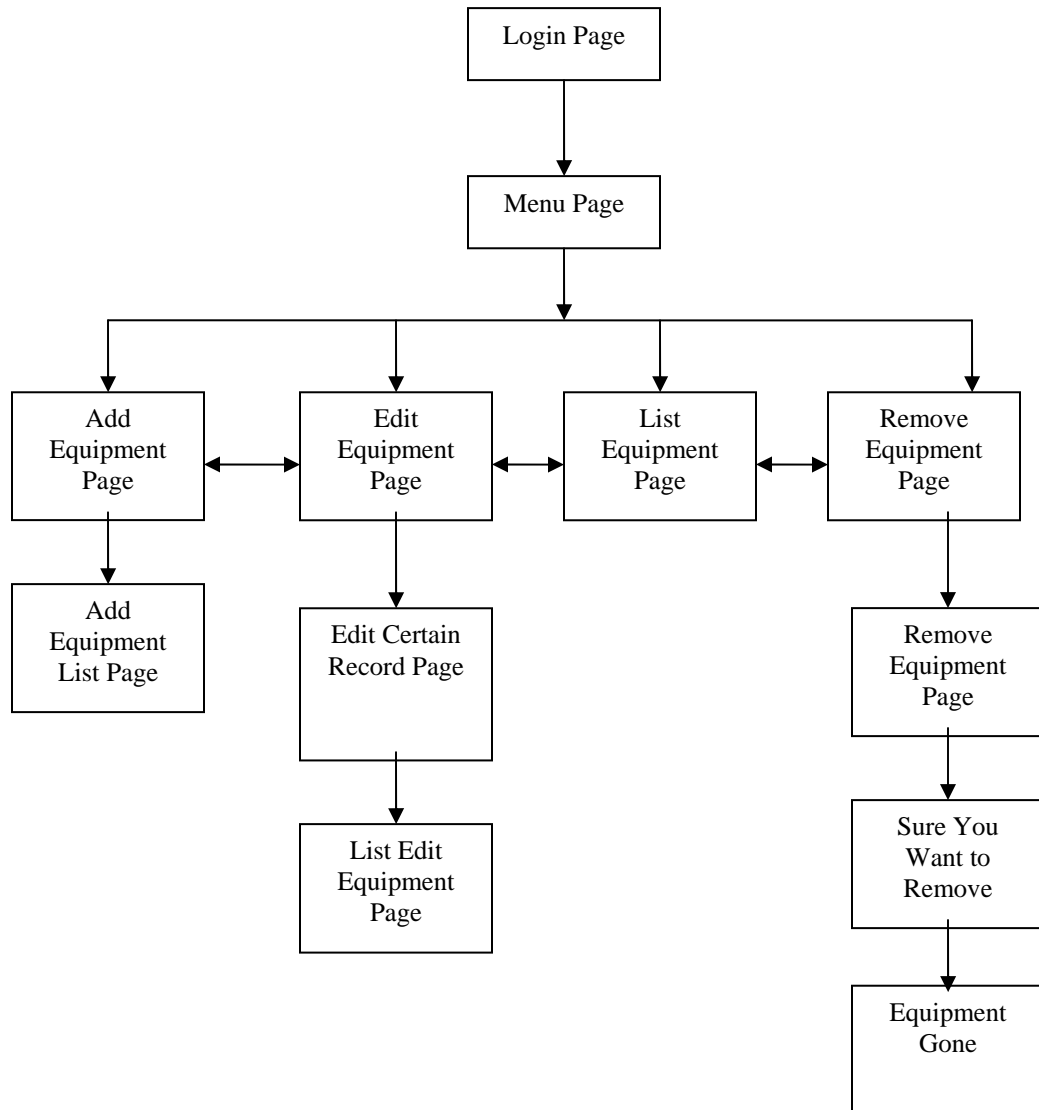


Figure 2: Asset Management Outline

5.1 Login Page

This first page that a user comes across is the login page. The login page shows the banner “Asset Management” in a pleasing font and color (See Figure 3.) with the copyright date at the bottom of the page. If the user inputs nothing, the page will tell the

user to “Fill in a required field” (See Figure 4.). If the user inputs the wrong username or password, the page will tell the user, “Incorrect Username or Password” (See Figure 5.).

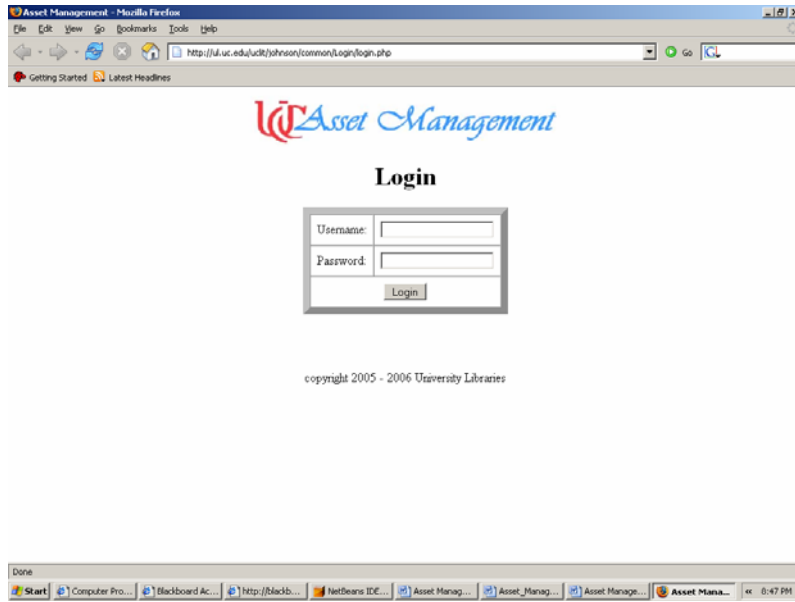


Figure 3: Login

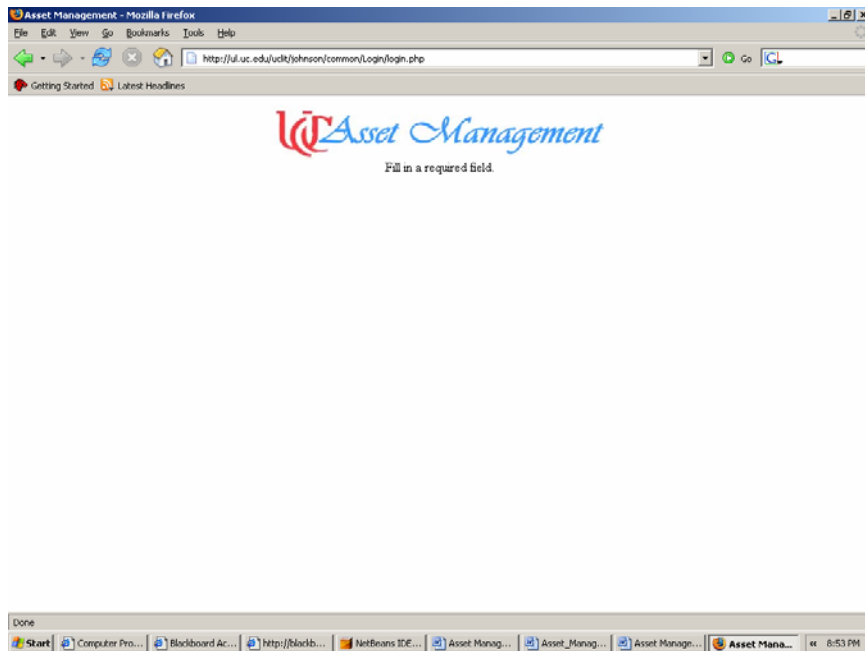


Figure 4: Fill in a required field

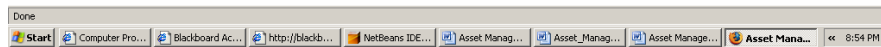
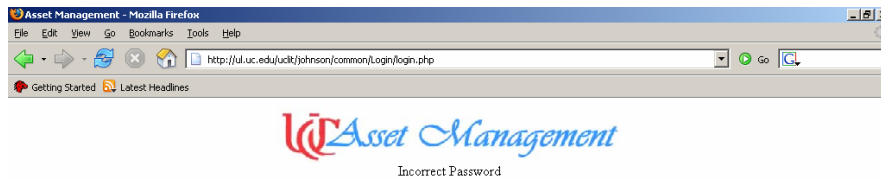


Figure 5: Incorrect Username or Password

5.2 Menu Page

After a user logs into the application they come across the menu page. The menu page lets the user know that they are logged in and welcomes the user back to the application (See Figure 6.). The menu is made up of one equipment folder, once a user clicks on the folder, the folder will open up and the user can now choose one of the following options: Add equipment, Edit equipment, Remove equipment, or View equipment (See Figure 7.).

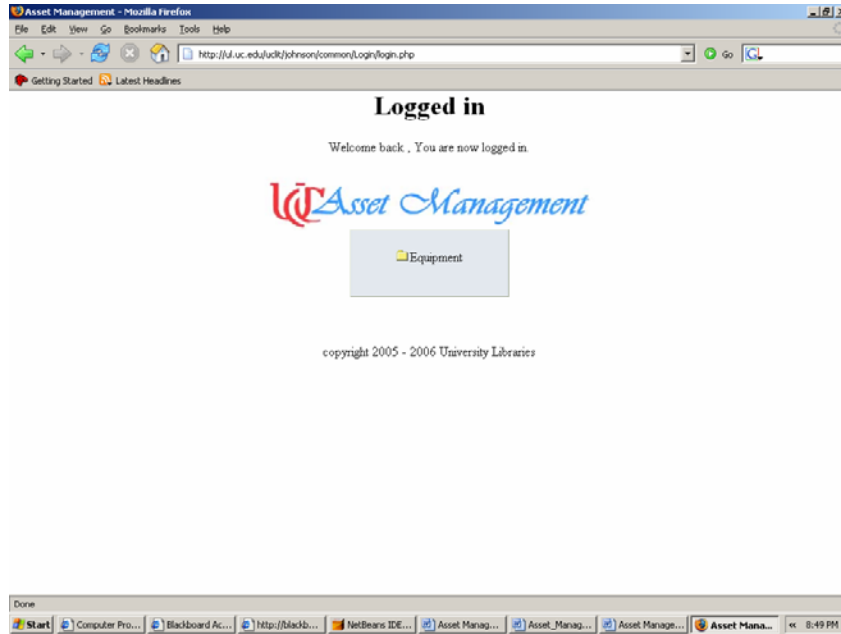


Figure 6: Menu Page

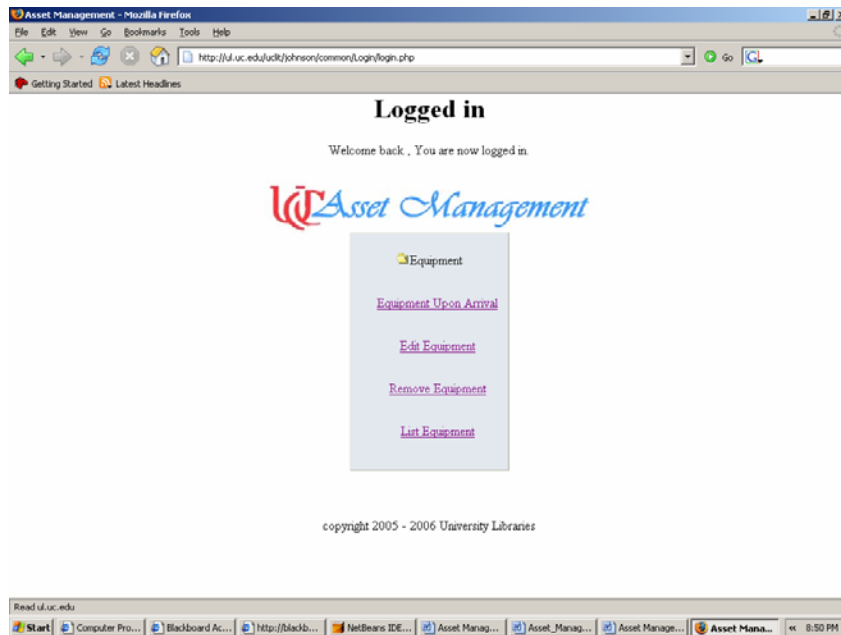


Figure 7: Menu Page Open

5.3 Add Equipment Page

The add equipment page is the first choice on the menu page. The add equipment page allows the user to add new equipment. The add equipment page allows the user to enter the following data: data received, type of equipment, serial number of equipment, maker, location that equipment will be moved to, quantity, and any comments needed (See Figure 8.). After the user adds some new equipment, they will come to the next page, which will show them what they have inputted into the database and also gives them the option of where they would like to go next (See Figure 9).

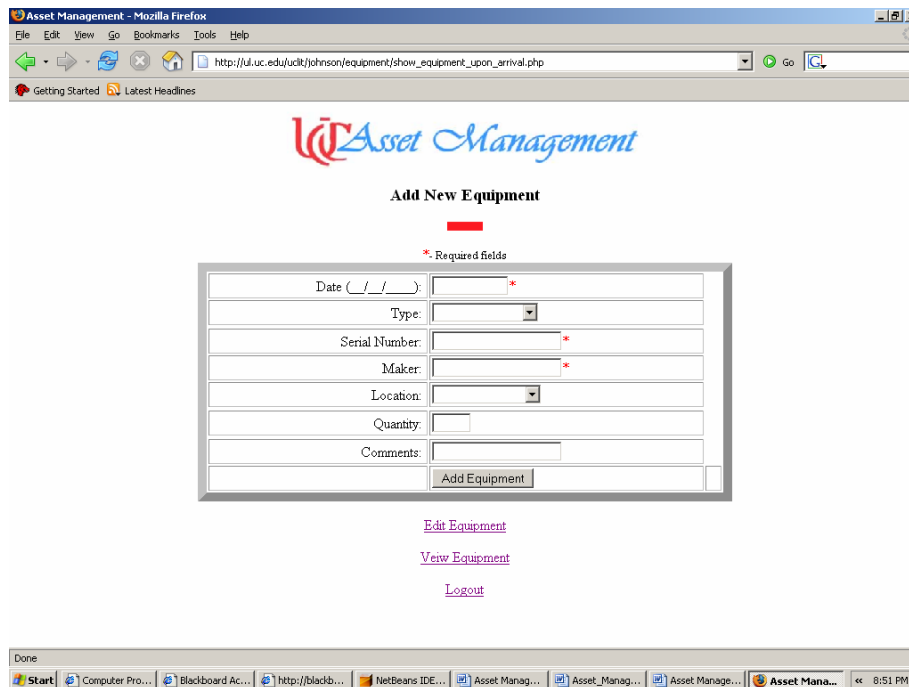


Figure 8: Add Equipment Page

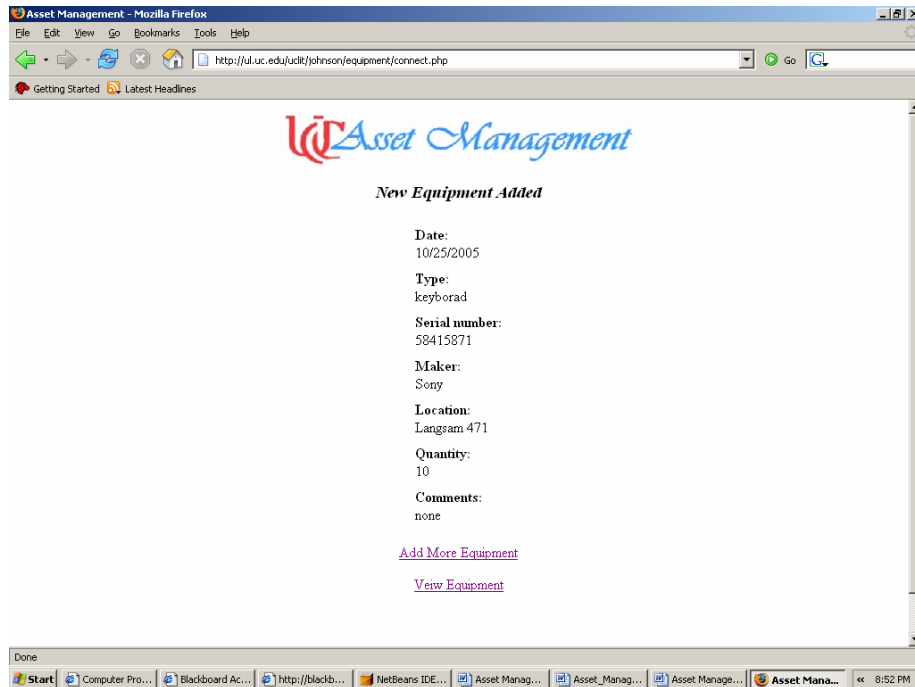


Figure 9: Add Equipment List Page

5.4 Edit Equipment

The second option on the menu page is the edit equipment page. This page shows all of the equipment that is currently inside of the database and allows the users to pick a certain record if something in that record needs to be updated (See Figure 10). After the user selects a record that they would like to update, the edit button takes them to that record and allows the user to edit any field of that record except for the equipment_id number (See Figure 11). If the user edits some equipment they will come to the next page. This page will show what the user has just edited and also gives them the option of where they would like to go next (See Figure 12.).

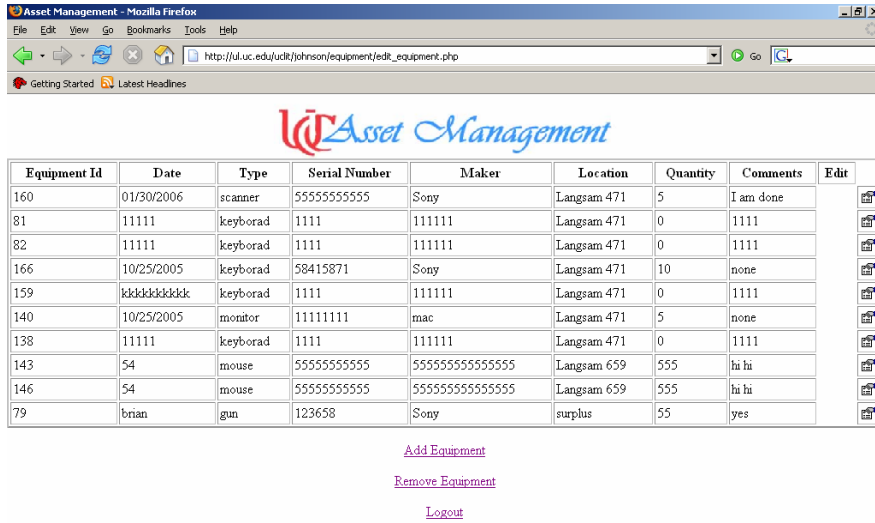


Figure 10: Edit Equipment Page

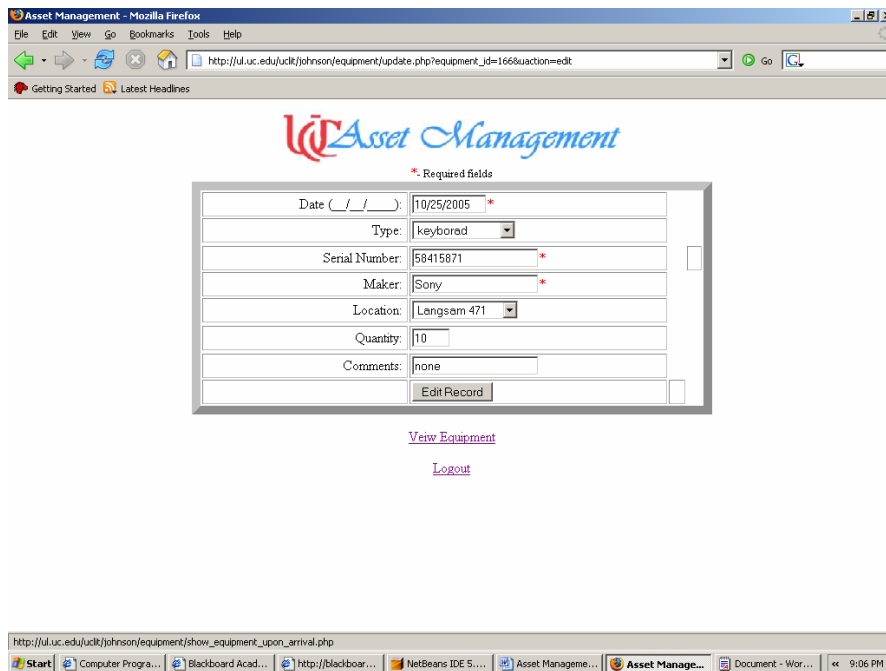


Figure 11: Edit Certain Record

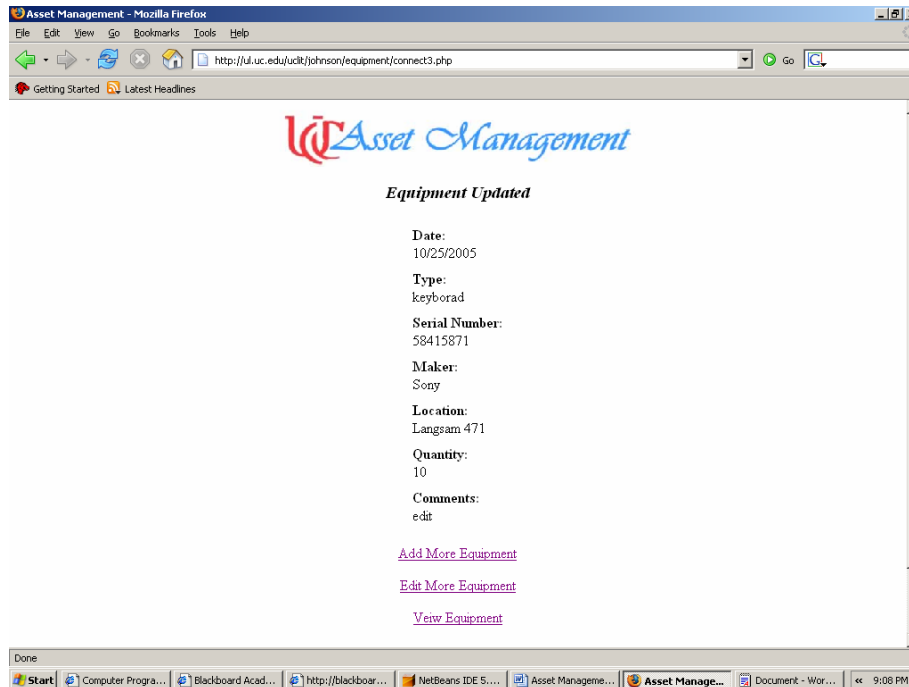


Figure 12: Edit Equipment List Page

5.5 Remove Equipment

The third option on the menu page is the remove function. This page shows all of the equipment that is currently inside of the database and allows the users to pick a certain record if something in that record needs to be removed (See Figure 13). After the user selects a record that they would like to remove, the remove button takes them to that record and allows the user to double check to make sure that they want to remove that record (See Figure 14). Then, if the user is sure that they would like to remove the equipment, they can click on the remove button. This will bring up the next page which will tell the user that the equipment has been removed and also gives them the option of where they would like to go next (See Figure 15.).

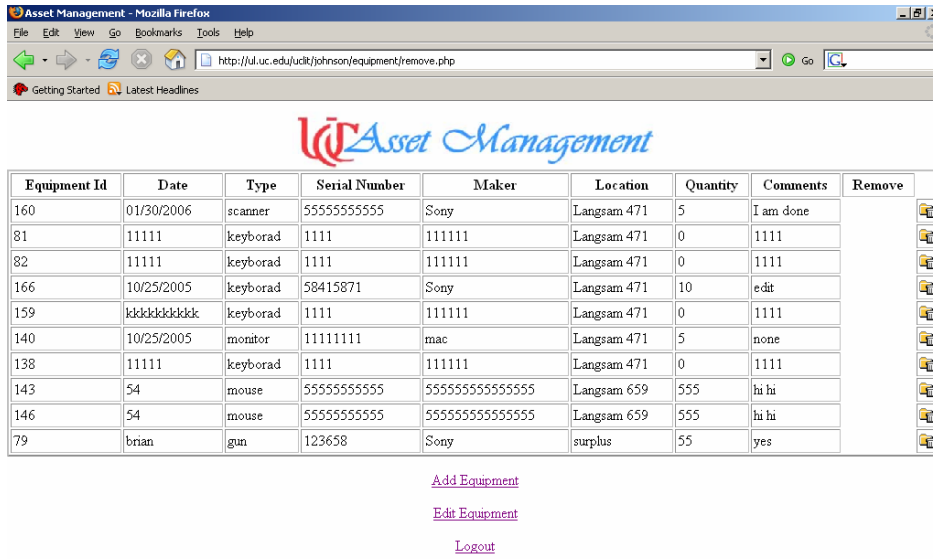
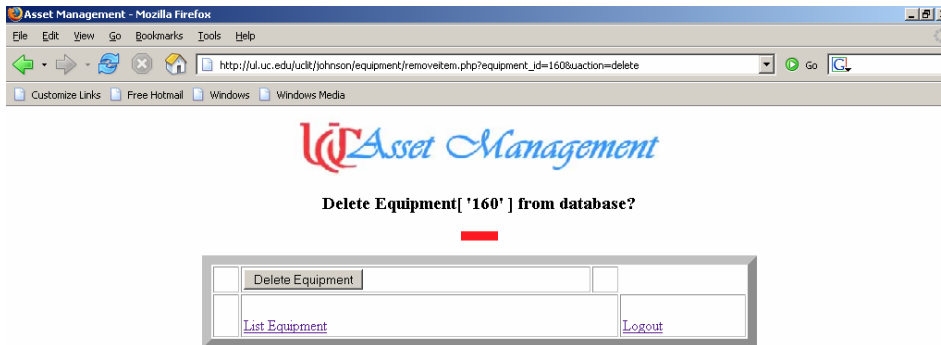


Figure 13: Remove Equipment Page



copyright 2005 - 2006 University Libraries

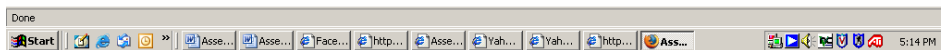


Figure 14: Sure you want to Remove

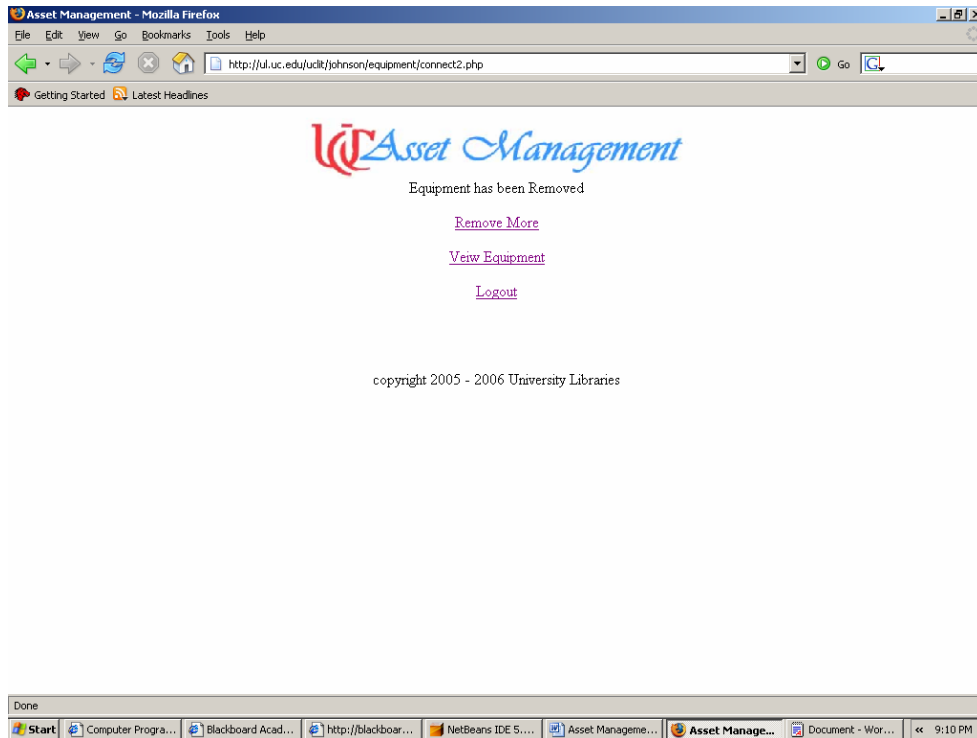


Figure 15: Equipment Gone

5.6 View Page

The final option that the user can select on the menu is the view function. This page shows all of the equipment that is stored inside of the database and also gives them the option of where they would like to go next (See Figure 16.).

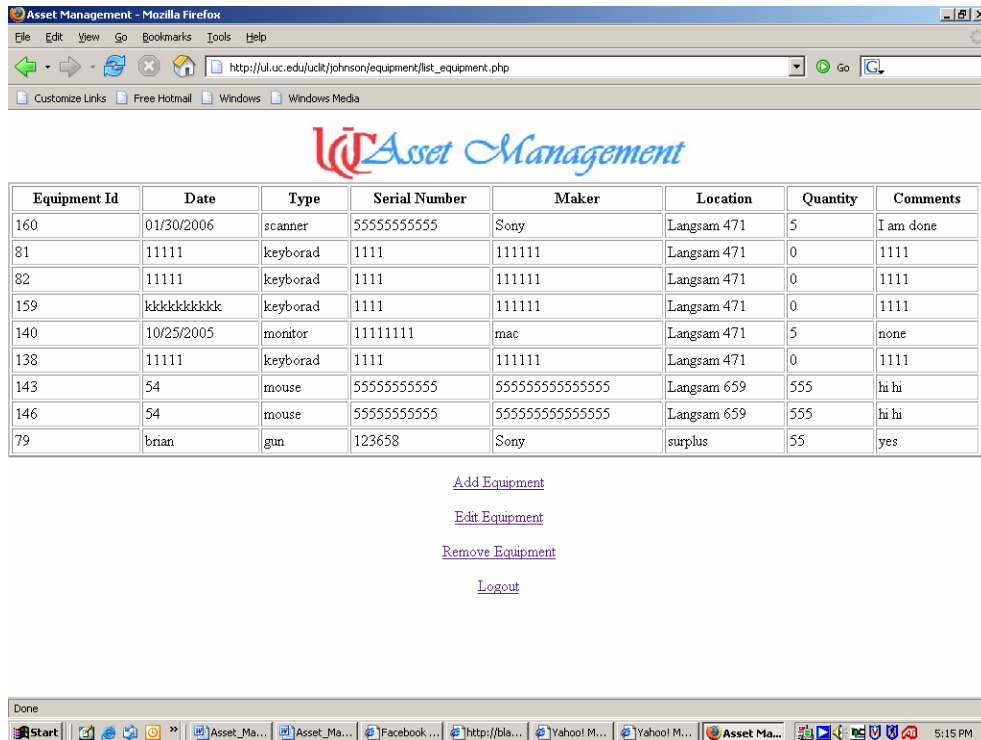


Figure 16 View Page

6. Testing

Code testing was an very important part of the development process. Each piece of code was tested after every update or modification made. Code testing help rid of minor syntax errors, as well as any other errors. Careful Planning before, during, and after the coding process reduced any delays on the project.

System testing was completed on the following systems: Microsoft Windows 2000, and Microsoft Windows XP. The following Web browsers were tested: Microsoft Internet Explorer 5.0 or later, Netscape 7.0 or later, and Mozilla Firefox 1.0.6 or later.

This is the testing plan for the project. This plan was done every time the project was updated. The testing will ensure that:

- All users can log in
- That all of the information entered in the fields is sent to the database

- PHP code is free of errors and that everything is running correctly
- The database keeps the correct data
- All of the buttons work
- The style sheet is working correctly

Testing was done a great deal in this project. Every time I updated some code on one of the pages, I completed the testing plan. I made sure that all of the codes were free of errors and that the connection to the database worked. Further testing was performed by a few of my classmates, friends, and employees of Library IT. Testing was done by Kinh Bui (classmate, friend), Thomas Scherz (friend and Library IT employee) and Paul Dozier (friend, and Library IT employee). Everything that they found wrong was analyzed and corrected. The following table shows some of the problems I came across while working on my project (See Figure 17.).

<i>Problem</i>	<i>Solution</i>
On the Menu Page I came across a problem of getting Equipment folder to open.	In Navicat I had to set the node value to 1 instead 0, setting the node value to 1 makes the options in the menu table inside of the equipment folder.
On the Add New Equipment Page I came across the problem of getting the type and location drop downs to work.	I had to program the drop downs into the code as well as put them into a table (Types Table) inside of Navicat.
Checking to see if the data enter on the Web page was getting to and from the database	I learn to use the echo command (it is a language construct that outputs one or more strings).

<p>In the edit page the record that I click on to be edited, the information would not come up. The equipment_id was not being passed from one page to the next.</p>	<p>I had to pass the equipment_id as a hidden variable in order for this to work.</p>
--	---

Figure 17 Problem Table

7. Conclusion

This project was created in response to Library IT poor inventory tracking. Creating this Web application database will help the University of Cincinnati libraries become more aware of where all of the equipment is located and who has it. This Web application is a very useful tool for Library IT to have and will be used for many years to come.

Library IT receives a lot of equipment over the course of a year, and I am going to give them the chance to keep track of it all. I enjoyed creating this Web application and I have learned a lot in doing so. The project fulfilled all of the Design Freeze deliverables. In addition this project was tested repeatedly and usability.

8. Recommendations

While working on this project, I faced a few challenges. I recommend that a strong understanding of how a database works is needed in order to create this type of a project. I also recommend that a good understanding of computer programming is needed to save a lot of time and energy. Knowing HTML (HyperText Markup Language) can help a great deal. I learned PHP in the process of doing this project and from my co-op. Knowing HTML helped me a great deal with learning PHP.

Connecting to the database gave me a little trouble in the beginning but was worked out after I went back, and looked over the code and made sure that all of my variables were correct. The only page that gave me a lot of trouble was the edit page. In the edit page I have to pass the equipment_id variable through three pages in order for the data in the record to be edited. I had to pass the equipment_id as a hidden variable in order for this to work. The following example shows the hidden variable inside of the table:

```
<tr>
  <td align=right>Date ( __/__/____ ):&nbsp;  </td>
  <td><input name="date" type="text" size="10" maxlength="10"
value="01/30/2006"><font color=red>*</font>
<input name="equipment_id" value="160" type="hidden"></td>
</tr>
```

Those few lines of code gave me the most problems from the whole project. In conclusion, be prepare and prepare yourself for anything.

Appendix A

Code Snippets

This section will just show some of the codes that were used in order to create a few of my Web pages.

I. Login Page Snippets

```
<center><h1>Login</h1>
<form action="/uclit/johnson/common/Login/login.php" method="post">
<table align="center" border="9" cellspacing="0" cellpadding="8">
<tr><td>Username:</td><td>
<input type="text" name="uname" maxlength="40">
</td></tr>
<tr><td>Password:</td><td>
<input type="password" name="passwd" maxlength="50">
</td></tr>
<tr><td colspan="3" align="center">
<input type="submit" name="submit" value="Login">
</td></tr>
</table>
</form>
```

II. View Page Snippets

```
<table border='2' width='100%' cellpadding='2' cellspacing='2' >
<tr>
<td align='center'><b>Equipment Id</b></td>
<td align='center'><b>Date</b></td>
<td align='center'><b>Type</b></td>
<td align='center'><b>Serial Number</b></td>
<td align='center'><b>Maker</b></td>
<td align='center'><b>Location</b></td>
<td align='center'><b>Quantity</b></td>
<td align='center'><b>Comments</b></td>
</tr>
<tr>
```

III. Remove Page Snippets

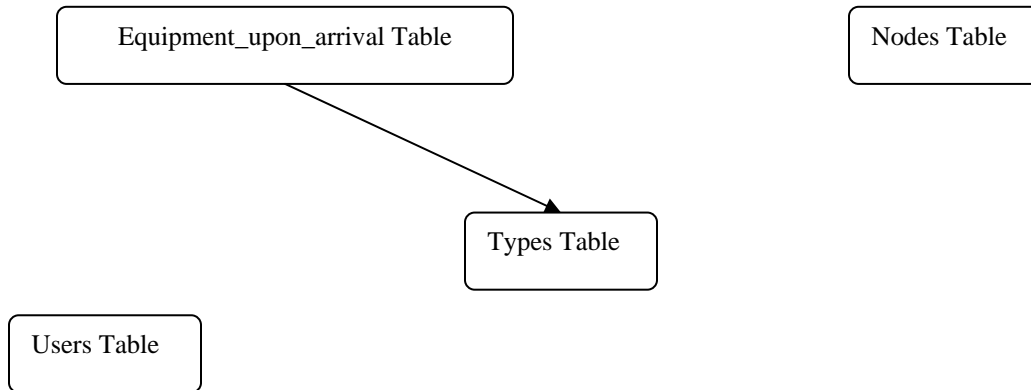
```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="/uclit/johnson/common/style_sheet_one.css">
<link rel='stylesheet' type='text/css' href= />

<center><title>Asset Management</title>
</head>
<body>

<div id="wrapper">
  <div id="headwrap">
    <div id="headerleft">
      <center>
    </div>
    <div id="headright"></div>
  </div>
  <div id="nav">
```

Appendix B

Data Relationship Diagram & Fields



Equipment_upon_arrival Table Fields
Equipment_id (varchar)
Date (char)
Type (char)
Serial_Number (char)
Maker (char)
Location (char)
Quantity (int)
Comments (char)

Types Table Fields
Type_id (varchar)
Type_Name (char)
Location_Name (char)

Nodes Table Fields
Nodes_id (varchar)
Nodes_Name (char)
Value (int)

Users Table Fields
User_id (varchar)
User_Name (char)
Links (char)
Password (char)

References

1. Gilmore, W. J. (2004) Beginning PHP 5 and MySQL: From Novice to Professional, Apress
2. Graham, Steven. (2003) Linux Administration 3rd Edition, McGraw-Hill
3. Jacobson, Reed. (2000) Step by Step Microsoft SQL Server 2000 Analysis Services, Microsoft Press
4. Kathy, Schwalbe. (2004) Project Management 4th Edition, Thomson Course Technology
5. Sklar, David. (2004) Learning PHP 5, O'Reilly
6. Unknown "MySQL Database Server- The Foundation of MySQL Network." <http://www.mysql.com/products/mysql/>
7. Unknown "About MySQL AB." <http://www.mysql.com/company/>
8. Warrene, Blane. "Navicat Review." <http://www.onlamp.com/pub/a/onlamp/2004/12/22/navicat.html>
9. Unknown "Navicat the Gateway to MySQL." <http://www.navicat.com/download.html>
10. The PHP Group "What is PHP." <http://us2.php.net>