# Equipment Room Database and Web-Based Inventory Management (EquipRD!)

## Final Report

Sean M. DonCarlos
Ryan Learned

Advisors:
Dr. James H. Irwin
Dr. Aleksander Malinowski

May 12, 2003

**TABLE OF CONTENTS**

**ABSTRACT**

This report describes the design, development and testing of the EquipRD! equipment room database and web-based inventory management system, including full functional descriptions, a table-level outline of the internal database structure, and a system overview presented with emphasis on user interfaces, as well as testing procedures and results. This paper outlines the similarities and differences in design and testing methods between EquipRD! on the Macromedia ColdFusion MX and PHP platforms. A bibliography and a summary of development and implementation costs are also provided.

## SYSTEM OVERVIEW

The inventory management system (hereinafter referred to as "EquipRD!") consists of three main software components: the database, the user interface(s) and the application server between the user interface(s) and the database. The application server permits the web server to communicate with the database; the web server cannot do this directly. (See Appendix 1 for a list of acronyms and abbreviations used in this report.)

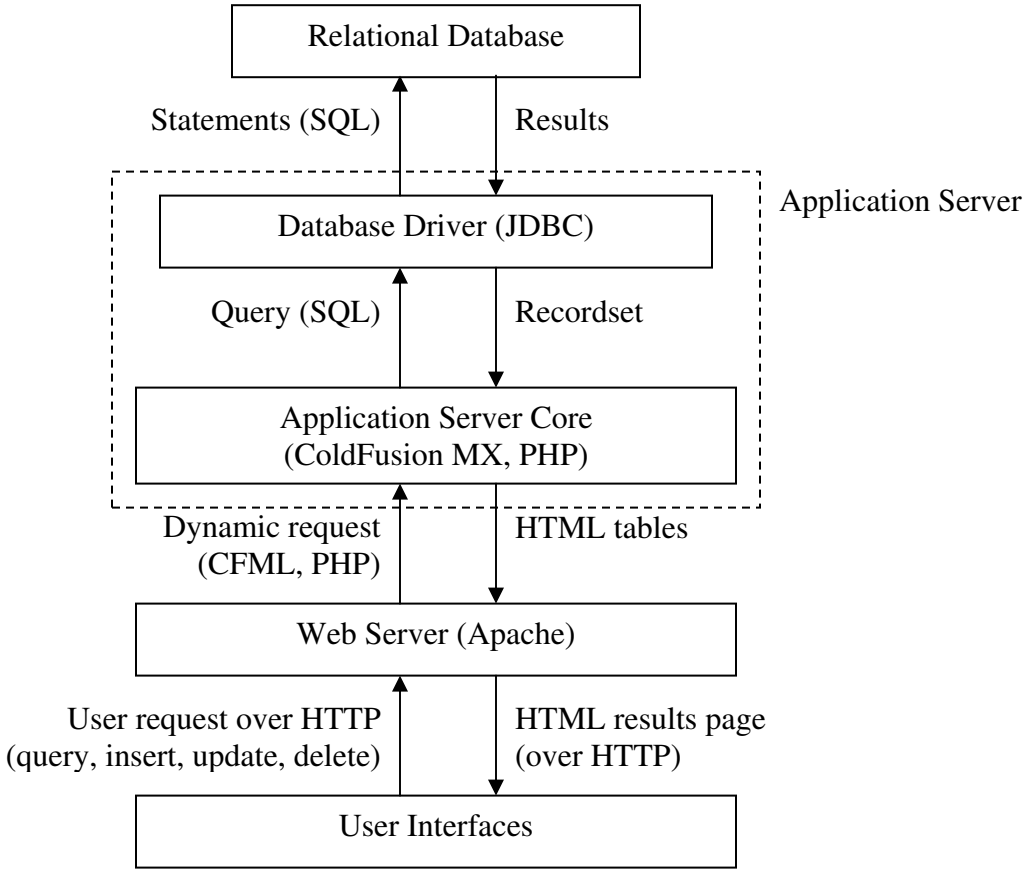


Figure 1 – System block diagram for EquipRD!. The data flow shown represents a user searching for parts.

### Relational Database

The relational database contains all of the information needed to track the equipment room inventory. The database can be searched or changed. Searching the database (called a *query*) causes it to return a subset of the data (called a *recordset*) that matches the search criteria. Changing the database by inserting, updating or deleting information causes the database to return status information regarding the success or failure of the change.

EquipRD! can use either Microsoft SQL Server 2000 or MySQL 4.1 for its database services.

**Database Driver**
The relational database is usually stored in a proprietary format and cannot be interpreted by the user or other software without the assistance of a database driver. The driver takes incoming SQL statements and converts them to a form understood by the database. The driver also passes recordsets and the status of attempted database changes back to the requesting application.

Two of the most prevalent drivers are ODBC (Open DataBase Connectivity) and JDBC (Java DataBase Connectivity). Both SQL Server 2000 and MySQL 4.1 use JDBC drivers.

**Application Server and Web Server**
When a web server receives a request for a web page, it normally retrieves the page and passes it back to the browser. Web pages that retrieve or manipulate data from a database, however, contain sections of non-HTML code (such as JavaScript or PHP) that browsers cannot interpret on their own. Therefore, web servers pass such pages to application servers, which interpret the non-HTML code, retrieve the data and return the results in HTML for display in the browser.

In general, application servers may be implemented as actual server software, such as Macromedia ColdFusion MX Server, or they may be as simple as small Java applets. Versions of EquipRD! have been developed for the Macromedia ColdFusion MX application server and for PHP.

**User Interfaces**
Each class of user will have a separate user interface appropriate for the functions that user class requires. This separation helps enforce security and data integrity.

All I/O shown in the block diagrams below is between the user's browser and the web server, conducted over HTTP (Hypertext Transfer Protocol), as illustrated in Figure 1, unless otherwise noted. In addition, all user interface blocks assume the presence of basic human interface devices (keyboard, mouse and display).

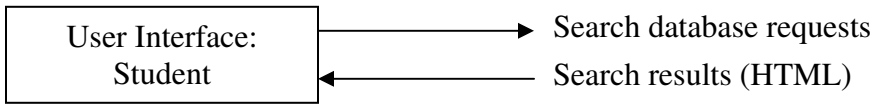


Figure 2 – Block diagram for student interface

**Student Interface**
The student needs to be able to search the database for the availability of a given part or type of part, and also for what parts the student currently has checked out. Preferably, he should be able to do this from any computer with a suitable web browser.

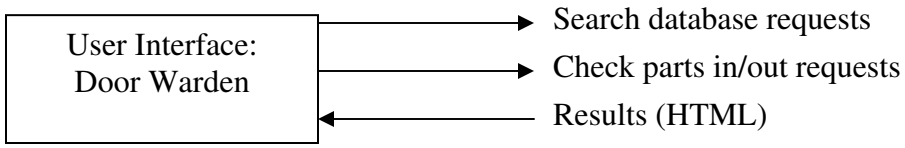


Figure 3 – Block diagram for door warden interface

**Door Warden Interface**
The door warden's primary function is to check parts in or out of the equipment room. The warden can search the database for part availability and can display the list of parts checked out by any student or set of students.

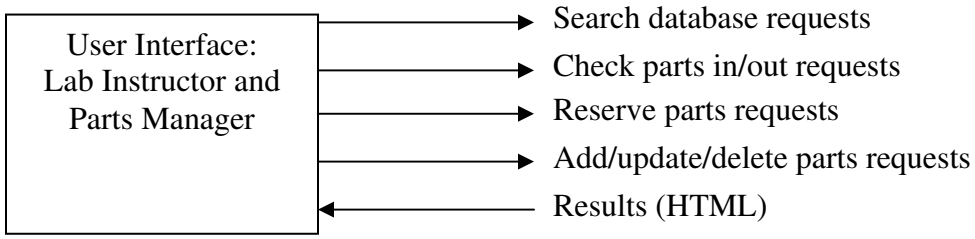


Figure 4 – Block diagram for lab instructor and parts manager interface

**Lab Instructor and Parts Manager Interface**
The lab instructor and parts manager can do everything the door warden can, plus they can also reserve parts, preventing them from being checked out. A lab instructor could do this to ensure availability of parts for a particular experiment. The parts manager may reserve parts to reflect damaged parts that are not suitable for use but are expected to be repaired in the future.

The lab instructor has the limited ability to update the parts count in the database to reflect parts destroyed beyond repair in the laboratory. The parts manager has the more general ability to add parts to reflect received part orders, to delete parts that are no longer used, and to change the quantity or description of parts currently in the database.

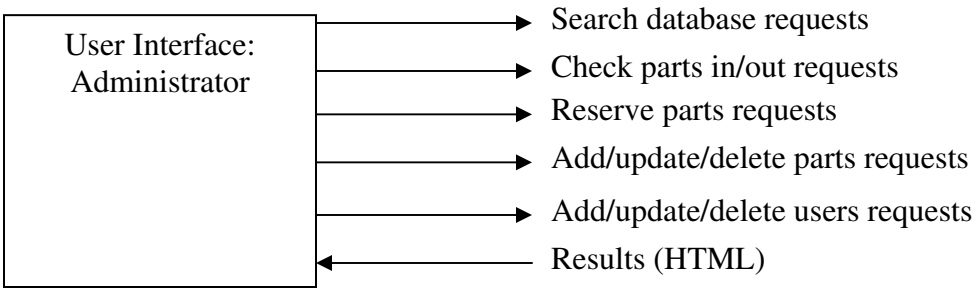


Figure 5 – Block diagram for administrator interface

**Administrator Interface**

The administrator is in complete control of the system and can perform any and all functions associated with the system. The administrator is the only user capable of adding, editing and deleting other users from the system.

## DATABASE OVERVIEW

### Database Table Structure

The database table structure defines the individual pieces of data that may be stored in the tables and any particular formats that apply to the data. The table structure is defined by the number of fields contained in the table and each field's name and data type. Usually, a structure also contains some indication of the purpose of each field.

The inventory management system uses three tables. The Users table maintains the login information and access level for each user of the system. The Parts table contains the part description and quantity information for each part available in the equipment lab. The Checkout table relates Parts checked out to the Users checking them out.

The database table structure is shown below.

### Users Table

- UserID – A six-digit number from 100000 to 999999 that identifies the user to EquipRD! Students use their Bradley ID number; faculty can use any unique six-digit number they like.
- FirstName – The user's first name, up to 20 characters in length.
- LastName – The user's last name, up to 20 characters in length.
- Password – The password the user needs to login to EquipRD! It must be at least 8 characters but no more than 20. The password is stored hashed (using the MD5 algorithm) for greater security.
- Roles – Describes the type of user (*i.e.*, Student, Door Warden, etc.).
- CollegeYear – The year the user will graduate. This is "Faculty" for faculty members.
- LabSection – The current lab section the user is enrolled for. This is "NA" for faculty members.
- LastLoginDate – The date and time this user last logged in to EquipRD!

### Parts Table

- PartNumber – A unique string that identifies the part to EquipRD! Two or more parts can share the same part number if they have different serial numbers.
- PartType – The major classification that the part belongs to, such as "Power Supply" or "74xx Series Logic".
- PartSubtype – The minor classification that the part belongs to, such as "DC Power Supply" or "74ALS Series Logic".
- PartDesc – A full-text description of the part, up to 100 characters in length.
- SerialNumber – A string representing the serial number of the part. Two or more parts can share the same part number if they have different serial numbers. If the part does not have or need a serial number, this field contains "None".
- Quantity – The total number of the part in EquipRD! For serialized parts, this field is always 1. For nonserialized parts, this field can be any positive integer up to and including 32,767.

- Available – The quantity of the part available for checkout. A part can be checked out if it is not already checked out, reserved or marked as damaged.
- InUse – The quantity of the part already checked out.
- Reserved – The quantity of the part put on reserve by a lab instructor.
- Damaged – If a serialized part is damaged and needs to be repaired, this field will contain 1. Nonserialized parts cannot be marked damaged and so this field will always contain 0 for such parts.
- LastCheckoutDate – The date and time when 1 or more of this part was last checked out by any user.

**Checkout Table**

- RowID – a unique number generated by EquipRD! This number ensures each row in the Checkout table is unique.
- UserID – the user that has checked out a part
- PartNumber – the part number of the part checked out
- SerialNumber – the serial number of the part checked out, if applicable
- Quantity – the quantity of the part checked out. For serialized parts, this field is always 1.
- CheckoutDate – the date and time the part was originally checked out.

## EQUIPRD! COMMON FUNCTIONALITY

### Features

EquipRD! provides the following features in addition to the basic EquipRD! functionality described in the System Overview section:

- Easy login/logout procedure – Users login once with their user ID and password. EquipRD! automatically verifies each page request of the user against that user's security role and the required role for the requested pages. Users logout by clicking on the "Logout" link on each page.
- Color-coded user interfaces – EquipRD!'s web interface changes color based on the security role of the current user. Students are red, Door Wardens are green, Lab Instructors are black, Parts Managers are violet and Administrators are blue.
- Full contextual help – All of the functional pages (*i.e.,* any page that searches or modifies the database) in EquipRD! have links to their own reference pages with full descriptions of the functions and the forms used. This reference opens in a separate window so that the user can read the reference and perform the task at the same time. Also, many functional pages have tips on the page that provide hints for more effective use of EquipRD!
- "Search-before-modify" functions – Many functions have search pages built into them. The search results then appear on the same page as the function form, eliminating the need to switch back and forth between pages.
- Defensive coding – EquipRD! protects itself from attempts to change the database in an inconsistent way, such as by checking out 5 parts when only 4 are available.
- Automatic logging – EquipRD! records all login/logout activity and database changes to a human-readable text log file.

### Functions

The following is a list of all the functions available to users of EquipRD!, including the minimum security roles needs for each:

- Student – Display Own Checked Out Parts, Search for Parts
- Door Warden – All functions above plus Check In Parts, Check Out Parts, Display All Users with Given Part, Display Others' Checked Out Parts
- Lab Instructor – All functions above plus Destroy Nonserialized Part, Release Parts, Reserve Parts
- Parts Manager – All functions above plus Add Part to Database, Delete Part from Database, Display Parts by Quantity, Display Unused Parts, Mark Serialized Part Damaged, Unmark Damaged Serialized Part, Update Part Information
- Administrator – All functions above plus Add User to Database, Delete User from Database, Display Unused Users, Search for Users, Update User Information

For more information on functions, see the compdesc.txt file on the enclosed CD or the online function reference in EquipRD!

## EQUIPRD! CF

EquipRD! CF is the version of EquipRD! that runs on the ColdFusion MX application server. It can use either SQL Server 2000 or MySQL 4.1 as its database. (It can also use MySQL 4.0, in which case the "Delete User from Database" function is unavailable.)

### Design Advantages of ColdFusion MX

The Macromedia ColdFusion MX application server provides the following advantages to implementing the features listed in "EquipRD! Common Functionality":

- Simplified login/logout coding – ColdFusion MX and CFML (ColdFusion Markup Language) provide tags such as <cflogin>, <cfloginuser> and <cflogout> that can automate most user authentication and tracking tasks. In addition, the Hash function implements MD5 hashing (for password security) in a single function.
- Simplified security – The GetAuthUser() function returns the current user ID, and the IsUserInRole() function compares the user's security role to a given role. Both functions automatically track each user separately, sparing the programmer from hours of coding and troubleshooting. Additionally, the <cfqueryparam> tag protects the database from malicious users by rejecting invalid query criteria.
- Simplified display of query results – ColdFusion's <cfoutput> tag automatically loops through the results of database queries, eliminating the need to write and maintain separate loops for displaying database output.
- Simplified file operations – Instead of using three separate commands to open a file (such as the EquipRD! log file), write to the file and then close the file, ColdFusion's <cfwrite> tag can perform all these operations in a single step.
- Integrated data validation – ColdFusion can check data entered in forms against common rules with no additional code. ColdFusion can also populate drop-down lists on forms from the database so that only valid values are presented to the user.
- Easy-to-understand language – CFML looks and acts similar to HTML and is much easier to read and understand quickly than Java, C++, etc.

### Testing Procedures and Results

Testing was performed after each page or major function was completed. A number of functions in EquipRD! are similar; therefore, incremental testing prevents errors from propagating throughout EquipRD!

Testing for a given function began with supplying expected values to the function and observing the function's effect on the database (or returned results if the function was a search). After the function gave the expected results for a wide range of expected inputs, testing continued by supplying the function with unexpected, nonsensical, invalid or even malicious inputs and observing the function's response. Revision and testing proceeded until the function operated correctly (or displayed a graceful error message for the user) for a wide range of arbitrary inputs.

A number of problems were introduced when the CFS and CFM versions of EquipRD! (see Version Information below) were merged to produce the current version, primarily

resulting from errors that had been corrected in one version but not the other. Therefore, the first EquipRD! CF version after the merge (1.08) underwent a second round of testing prior to the Student Exposition to remove any lingering errors.

**Version Information**
The current version is EquipRD! CF 1.09.

The following table gives a brief history of the major versions of EquipRD! CF. Note that EquipRD! CF was split into two versions after 1.03, a CFS and a CFM version. This was in response to MySQL 4.0 not supporting subqueries. After the release of MySQL 4.1, this issue was resolved, and the two source codes were merged back into a single CF version.

| **Version** | **Functionality** |
|---|---|
| CF 0.18 | Login, logout, color-coded user interfaces, dynamic forms based on database contents, rejects invalid/malicious page accesses |
| CF 0.31 | Search for user or part, return results, enforces security roles |
| CF 0.33 | Search for checked out parts, all users with given part |
| CF 0.67 | Reserve and release parts, mark and unmark damaged serialized parts, destroy nonserialized parts, enforces consistency in all reserve/damage/destroy operations |
| CF 0.71 | Update part information |
| CF 0.92 | Add/update/delete parts and users, enforces consistency in all add/update/delete operations |
| CF 1.02 | Check in/out parts, automatic logging, first version to meet all original specifications |
| CFS/CFM 1.04 | Custom error-handling pages, minor errors fixed |
| CFS/CFM 1.07 | Online function reference |
| CF 1.08 | Source codes remerged |
| CF 1.09 | Password hashing using MD5, current version |

Figure 6 – Version history of EquipRD! CF

## EQUIPRD! PHP

EquipRD! PHP is the version of EquipRD! that runs on the open source PHP engine and MySQL database. The basic workings of the website behave exactly like the CF version of EquipRD! Many of the "amenities" offered by ColdFusion are not available in PHP. Some features had to be hard coded in order to be implemented. The reason the PHP version was implemented was to provide a free alternative to the CF version.

### Design Choices for EquipRD! PHP

EquipRD! PHP has the same features as the CF version but required minor changes in order to be implemented. CF 1.04 and 1.07 were used as templates for EquipRD! PHP. The following is a summary of design choices from converting to the PHP version from the CF version:

- Login/logout procedure – It is native to ColdFusion to provide login and logout capabilities. It is also simple to check if a user has the access required to view certain pages. With PHP, this capability must be manually coded. It was implemented using a session variable which holds the user's access.
- Security – The session variables contain the user's access levels from page to page. Each page will check this session variable to make sure the user can view the page.
- Display of Query Results – ColdFusion provided a simple way to create the forms on the search pages. This was coded into PHP to show only the necessary items in the select box for search-before-modify functions and to show all items in the main search pages.
- File operations – PHP requires several lines of code to write to a file for logging purposes. The only problem will arise when a user does not logout using the logout link on the pages.
- Integrated Data Validation – The data entered on pages must be checked to make sure it is proper. For example, a user should not be able to enter letters in for a quantity. The PHP version checks the data validity with JavaScript. Server side checking would be preferred, but for now JavaScript will be required for EquipRD! PHP.

### Testing Procedures and Results

Testing was performed after each page or major function was completed. A number of functions in EquipRD! are similar; therefore, incremental testing prevents errors from propagating throughout EquipRD! For example, the search functions on each page are similar to all of the others.

Testing for a given function began with supplying expected values to the function and observing the function's effect on the database (or returned results if the function was a search). After the function gave the expected results for a wide range of expected inputs, testing continued by supplying the function with unexpected and invalid inputs and observing the function's response. This included putting letters where numbers should go. Revision and testing proceeded until the function operated correctly. JavaScript was used for some error checking and has not been implemented on all pages yet.

## ASSESSMENT OF SCHEDULE

### Original Schedule

The proposed schedule is shown in Figure 7. All schedule times were subject to change as the project progressed. There was a built-in window between the time work was to be done on the prototype server and corresponding work was to be done on the production server. This window allowed for unexpected delays and difficulties without disrupting the project timeline.

| Date | Sean's Tasks | Ryan's Tasks |
|---|---|---|
| Jan 23 | Design tables and user interfaces on prototype (ColdFusion) server, begin learning PHP language | Begin learning PHP language, assist in design of user interfaces. |
| Jan 30 | Continue designing and testing of user interfaces. | Continue learning PHP and assisting in design. |
| Feb 6 | Design and test security features on prototype server. | Integrate and test barcode scanner and magnetic striper reader into user interface on prototype server. |
| Feb 13 | Design and test logging features on prototype server. | Begin porting tables and user interfaces to production (PHP) server |
| Feb 20 | Finish design on prototype server. | Continue porting and test. |
| Feb 27 | Begin assisting in port of code to PHP on production server. | Integrate and test barcode scanner and magnetic stripe reader into user interface on production server. |
| Mar 6 | Continue porting and testing. | Port and test security features on production server. |
| Mar 13 | Continue porting and testing, begin writing any needed documentation for end users. | Port and test logging features on production server. |
| Mar 20 | Spring Break | Spring Break |
| Mar 27 | Finish design on production server. | Finish design on production server. |
| Apr 3 | Prepare for project exposition. | Prepare for project exposition. |
| Apr 10 | Project exposition | Project exposition. |
| Apr 17 | Work on presentation and report. | Work on presentation and report. |
| Apr 24 | Work on presentation and report. | Work on presentation and report. |
| May 1 | Work on presentation and report. | Work on presentation and report. |

Figure 7 – Proposed schedule for inventory management system project work

### Actual Progress Compared to Original Schedule

The schedule above reflects the lack of knowledge in December about the length of time needed to implement some features, particularly in ColdFusion. The security and logging features were each given a week of development time, yet they took only 4 hours and 1 hour to implement, respectively. On the other hand, the sheer number of pages to code pushed the schedule back a few weeks.

However, both versions of EquipRD! are completed and ready for use. EquipRD! CF is most likely more stable and robust than EquipRD! PHP, as it has been in existence about a month longer and therefore has been tested more extensively.

Also, the bar code scanner and card reader were dropped from the project in order to allow adequate time to learn PHP and code EquipRD! PHP.

**Labor Invested**

**Sean M. DonCarlos (EquipRD! CF)**

| | |
|---|---|
| Researching available databases and application servers | 6 hours |
| Developing EquipRD! model (table structures, security roles) | 21 hours |
| Installation and setup of ColdFusion and SQL Server software | 4 hours |
| EquipRD! CF coding, debugging and troubleshooting | |
| Login and logout | 5 hours |
| User interface forms and layout | 15 hours |
| User interface graphic elements | 2 hours |
| SQL statements for database | 8 hours |
| Supporting ColdFusion code (color-coding, populating select boxes from database queries, etc.) | 18 hours |
| MySQL incompatibility troubleshooting | 12 hours |
| Documentation and project website on cegt201 | 27 hours |
| Online function reference | 8 hours |
| Presentations, design reviews, papers, Student Exposition | 19 hours |
| Meetings | 17 hours |
| Computer repair of home computer running EquipRD! CF | 5 hours |
| **Total** | **167 hours** |

**Ryan Learned (EquipRD! PHP)**

| | |
|---|---|
| Researching barcode standards and magnetic stripe information | 6 hours |
| Researching barcode and magnetic stripe reader hardware | 4 hours |
| Developing possible implementation of barcodes | 3 hours |
| Learning PHP and MySQL database | 18 hours |
| Exploring keyboard and barcode reader | 4 hours |
| Installation and setup of ColdFusion MX software on laptop | 3 hours |
| Examination of EquipRD! CF code and web interface | 9 hours |
| EquipRD! PHP coding, debugging and troubleshooting | |
| Login and logout | 6 hours |
| Page security | 3 hours |
| Conversion of ColdFusion tags to PHP | 10 hours |
| Conversion of ColdFusion pages to PHP | 35 hours |
| Verification | 6 hours |
| Subquery workaround for MySQL 4.0 | 3 hours |
| Documentation and project website on cegt201 | 4 hours |
| Online function reference conversion | 2 hours |

| | |
|---|---|
| Presentations, design reviews, papers, Student Exposition | 12 hours |
| Meetings | 17 hours |
| **Total** | **145 hours** |
| **Total Invested Labor** | **312 hours** |

## BIBLIOGRAPHY

Much of the following bibliography was used in the comparison and selection of the database and application server software.


**Databases**
IBM DB2 Universal Database 7.2
http://www-3.ibm.com/software/data/db2/udb/features.html

Microsoft Access 2002
http://www.microsoft.com/office/access/default.asp

Microsoft SQL Server 2000
http://www.microsoft.com/sql/evaluation/sysreqs/2000/default.asp

MySQL AB MySQL 4.0
http://www.mysql.com

Oracle 9*i*DB
http://www.oracle.com/ip/deploy/database/oracle9i/

**Application Servers**
Jakarta Tomcat 4.1
http://jakarta.apache.org/

Macromedia ColdFusion MX
http://www.macromedia.com/software/coldfusion/

Macromedia JRun 4
http://www.macromedia.com/software/jrun/

Microsoft Internet Information Services (IIS) 5.1
http://www.microsoft.com/windows2000/server/

PHP 4.2.3
http://www.php.net

Sun ONE Active Server Pages 3.6.2
http://wwws.sun.com/software/chilisoft/sysreq.html

**Books**
PHP and MySQL Web Development, 2nd Edition. Luke Welling and Laura Thomson. Sams Publishing.

Web Database Applications with PHP and MySQL. Hugh E. Williams and David Lane. O'Reilly.

## EQUIPMENT LIST AND COSTS

**Development Costs:**
The following costs were incurred by the EquipRD! project during its development.

| **Software:** | **Cost:** |
|---|---|
| Macromedia ColdFusion MX Server Developer Edition (free download) | $0 |
| Microsoft SQL Server 2000 Developer Edition (already installed) | $0 |
| MySQL (all versions are free downloads for noncommercial use) | $0 |
| PHP (all versions are free downloads) | $0 |
| Software Total | $0 |

| **Hardware:** | **Cost:** |
|---|---|
| Magnetic card reader and keyboard combo | $15 |
| Bar code scanner | $19 |
| Hardware Total | $34 |
| **Grand Total** | **$34** |

Figure 8 – EquipRD! Development Costs

**Implementation Costs:**
The implementation costs are dependent on which versions of EquipRD!, database and application server are selected. The software is therefore listed in valid combinations of these three items; the total cost for the package is given at right. All prices are for full commercial licenses; educational pricing may be available.

| **Software:** | **Cost:** |
|---|---|
| EquipRD! CF, ColdFusion MX Professional, SQL Server 2000 Professional | $2,748 |
| EquipRD! CF, ColdFusion MX Professional, MySQL 4.1 | $1,299 |
| EquipRD! PHP, PHP application server, MySQL 4.1 | $0 |

Figure 9 – EquipRD! Implementation Costs

Note: The card reader and bar code scanner were dropped from the project in early February.

## APPENDIX 1 – LIST OF ABBREVIATIONS

**CF** – The designation given to versions of EquipRD! that run on ColdFusion and either SQL Server or MySQL.

**CFM** – The designation given to earlier versions of EquipRD! that ran only on ColdFusion and MySQL.

**CFML** – ColdFusion Markup Language – the programming language used with Macromedia ColdFusion MX. CFML is very similar to HTML in that it is tag-based.

**CFS** – The designation given to earlier versions of EquipRD! that ran only on ColdFusion and SQL Server.

**HTML** – Hypertext Markup Language – a set of codes inserted into a file meant to be viewed in a web browser.

**HTTP** – Hypertext Transfer Protocol – a set of rules for transferring files, images, and webpages over the Internet.

**JDBC** – Java DataBase Connectivity – A Java-based method for connecting to database systems.

**MySQL** – an open source version of SQL.

**ODBC** – Open DataBase Connectivity – A C++-based method for connecting to database systems, often found in Microsoft products.

**PHP** – PHP Hypertext Preprocessor – a scripting language often embedded in HTML documents.

**SQL** – Structured Query Language – a standard language for retrieving information from and placing information into a database.