

# Equipment Room Database and Web-Based Inventory Management

## Project Proposal

Sean M. DonCarlos  
Ryan Learned

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

December 12, 2002

## TABLE OF CONTENTS

Project Summary .....	3
System Overview .....	4
Relational Database .....	4
Database Driver .....	5
Application Server and Web Server .....	5
User Interfaces .....	5
Student Interface .....	6
Door Warden Interface .....	6
Lab Instructor and Parts Manager Interfaces .....	6
Administrator Interface .....	7
Database Table Structure and Rules Checking .....	8
Database Table Structure .....	8
Rules Checking .....	8
Proposed Schedule .....	10
Bibliography .....	11
Equipment List .....	12

## **PROPOSAL SUMMARY**

The proposal for the EE 452 senior capstone project is to design and implement an inventory management system for the ECE department's equipment room. This system will be web-based. The system will be capable of maintaining the lists of parts and authorized users and tracking parts checked out against users.

To facilitate easy checking in and out of parts, the design of the user interface is to be made as simple as possible. Also, a bar code scanner will be provided in the equipment room to expedite the parts checking process.

## SYSTEM OVERVIEW

The inventory management system consists of three main software components: the database, the user interface(s) and the bridge software between the user interface(s) and the database. The bridge software is required because current web servers and browsers do not have the ability to communicate directly with a database.

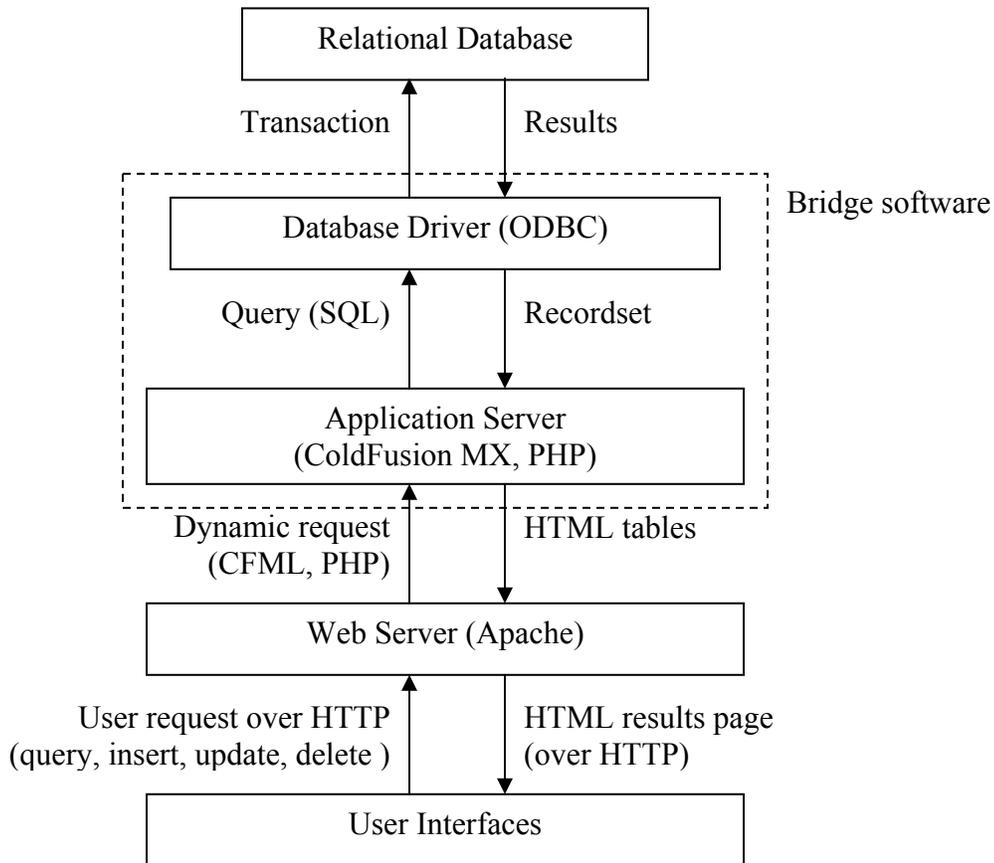


Figure 1 – System block diagram for an inventory management system. The data flow shown represents a user searching for parts.

### 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. Any operation on a database is called a *transaction*.

### 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 or XML transactions 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). Our prototype server will use the Microsoft SQL Server's built-in ODBC driver. Our production server will use MyODBC.

### 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. Our prototype server will use Macromedia ColdFusion MX Server Developer Edition as its application server and web server. Our production server will use PHP as its application server and Apache as its web server.

### 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).

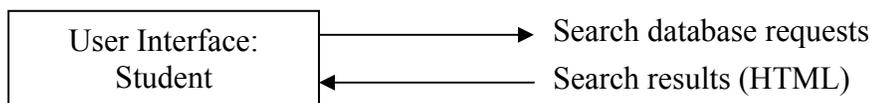


Figure 2 – Block diagram for student interface

### 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.

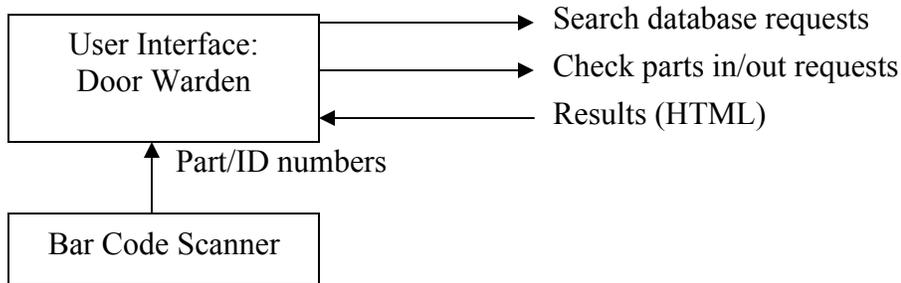


Figure 3 – Block diagram for door warden interface

### Door Warden Interface

The door warden's primary function is to check parts in or out of the equipment room. To aid this process, the door warden will be given a bar code scanner to streamline part number and ID number entry. 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.

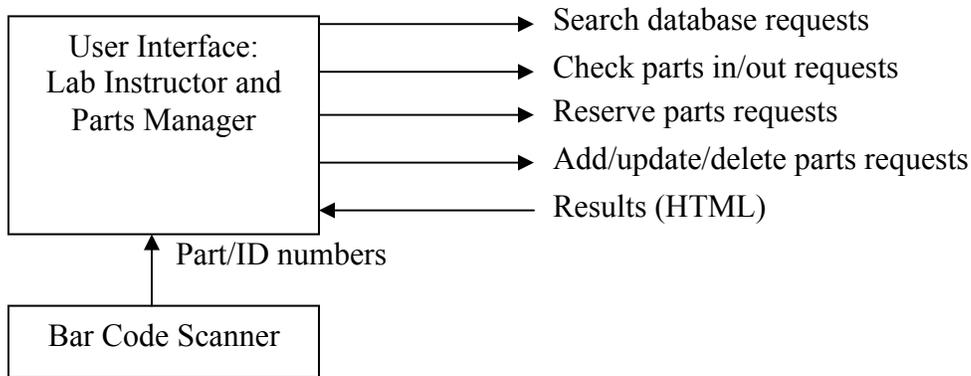


Figure 4 – Block diagram for lab instructor and parts manager interface

### 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.

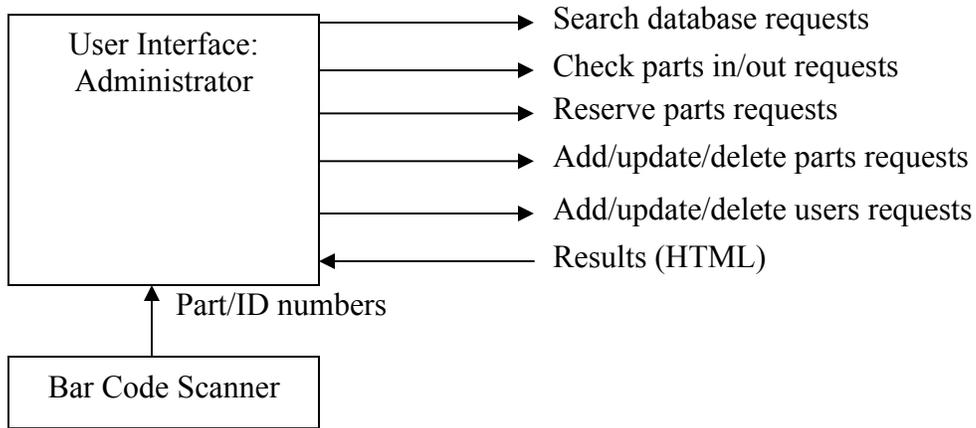


Figure 5 – Block diagram for administrator interface

### **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 TABLE STRUCTURE AND RULES CHECKING

### 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 for the Users table is shown in Figure 6. All structures are subject to change as the project progresses.

Field Name:	Field Data Type:	Purpose:
UserID	32-bit unsigned integer	Unique user ID for each user. Students will use their 6-digit Bradley ID number for their user IDs. Faculty will be assigned unique user IDs differently.
Password	20-character string	A 6 to 20-character password for login and authentication.
FirstName	20-character string	User's first name.
LastName	20-character string	User's last name.
AccessLevel	8-bit unsigned integer	An integer from 1 to 6, representing the level of access the user has to the database.

Figure 6 – Database table structure for Users table

For more information on database table structures, visit the project website at:  
<http://cegt201.bradley.edu/projects/proj2003/equiprd/tablestructure.html>

### Rules Checking

To ensure that the database remains only in consistent states, attempts to interact with the database must be checked against rules that determine whether the attempt is a valid one. Three factors can make an attempt invalid:

**Invalid Input Data Type** - The data entered must match the type expected. For example, it would be inconsistent to enter "Sean" into a User ID field expecting a 32-bit integer.

**Inconsistent Manipulation of Data** - If a value changes, these changes must be reflected throughout the entire database. For example, it would be inconsistent to check out a part against a user and not decrement the value in that part's Available field, as doing so would create an "extra" part in the system that did not physically exist.

**Impossible Manipulation of Data** - A user must not be able to cause a change in the system that is inconsistent with physical reality or the database table structure itself. For example, it would be inconsistent to check out a part if that part's Available field equals 0, as it is physically impossible to check out a part that is not available. It would also be inconsistent to assign a user an Access Level greater than 6, as such access levels do not exist in the system.

Note that security-related rules are not covered by this form of rules checking. Rules checking aims to ensure that a given attempt to interact with the database makes sense; security checking determines whether the system should even allow the user to make the attempt.

Some simple rules for the Users table are shown in Figure 7. All rules are subject to change as the project progresses.

**ID Number**

1. Must be a 32-bit integer
2. Primary key (each record must have a unique value in its ID Number field)

**First Name**

1. Must be 1-20 characters in length
2. Can only contain A-Z, a-z, hyphens and spaces
3. Cannot begin with a space or hyphen

**Last Name**

1. Must be 1-20 characters in length
2. Can only contain A-Z, a-z, hyphens and spaces
3. Cannot begin a space or hyphen

**Password**

1. Must be 6-20 characters in length
2. Can only contain A-Z, a-z, 0-9

**Access Level**

1. Must be an integer between 1 and 6, inclusive.

Figure 7 – Rules for Users table

For more information on database table structures, visit the project website at:  
<http://cegt201.bradley.edu/projects/proj2003/equiprd/rules.html>

## PROPOSED SCHEDULE

The proposed schedule is shown in Figure 8. All schedule times are subject to change as the project progresses. There is a built-in window between the time work is to be done on the prototype server and corresponding work is to be done on the production server. This window allows for unexpected delays and difficulties without disrupting the project timeline.

<b>Date</b>	<b>Sean's Tasks</b>	<b>Ryan's Tasks</b>
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 stripe 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 8 – Proposed schedule for inventory management system project work

## **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 9iDB

<http://www.oracle.com/ip/dep/loy/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://www.sun.com/software/chilisoft/sysreq.html>

### **Barcodes**

Barcode Generator

<http://www.barcodesinc.com/generator/>

Barcode Frequently Asked Questions

<http://www.macromedia.com/software/coldfusion/>

## **EQUIPMENT LIST**

### **Software for Prototype Server:**

Macromedia ColdFusion MX Server Developer Edition (free download)	<b>Cost:</b> \$0
Microsoft SQL Server 2000 Developer Edition (already installed)	\$0

### **Software for Production Server:**

MySQL (all versions are free downloads for noncommercial use)	<b>Cost:</b> \$0
PHP (all versions are free downloads)	\$0

### **Hardware for Door Warden's Computer**

Magnetic Stripe Reader (built into an AT style keyboard)	<b>Cost:</b> \$15
Barcode Scanner Wand (can plug in to the above keyboard)	\$20