

Implementation

6.1. Introduction

Some of the key implementation issues are as follows.

Other than what is being discussed here, developer has to assist the clients to decide what hardware, and networking options to be considered in implementing the system in a laboratory. Developer's recommendation was to have a Linux based server with LAMP web server configuration and an Ethernet local area network (LAN) with a 10/100 Mbps Ethernet switch. ADSL Router can be added if clients need an internet connection and email facilities as well.

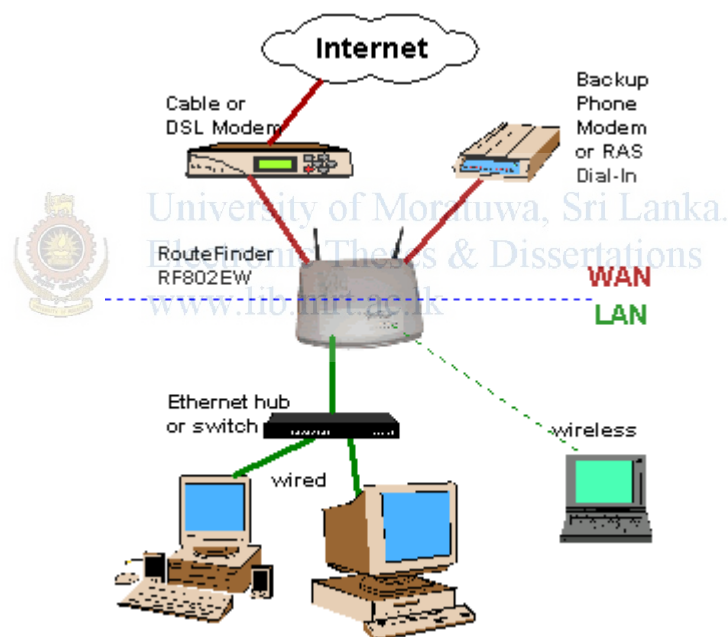


Figure 18: Possible LAN configuration for the system

Since the clients have an internet and LAN facility for the proposed laboratory, developer's suggestion was approved.

6.1.1 System functionality and file system

The functions of the architectural design are represented by different PHP files as follows.

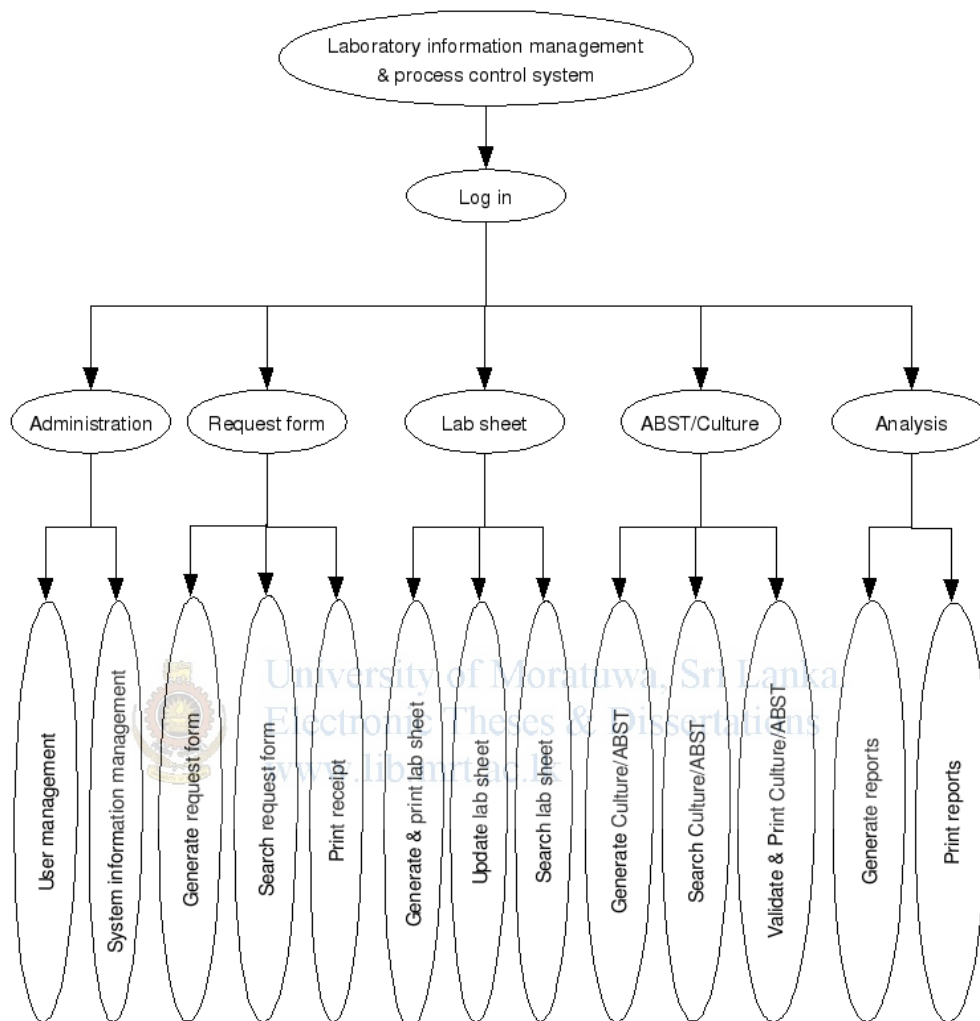


Figure 19: Architectural design

Sub system of Architectural design	PHP files
Login	Index.php
User Management	useradd.php
System Information Management	Sysup.php
Generate request forms	Genreq_form.php
Search request forms	searchRFs.php
Print receipt	printRcpt.php
Generate and print lab sheet	Genlab_sheet.php, printls.php
Update lab sheet	updateLS.php
Search lab sheet	searchLS.php
Generate culture/ABST	Genculture.php
Search culture/ABST	searchABST.php
Validate and print ABST	authorizePrnt.php
Generate reports	genRep.php
Print reports	repPrnt.php

Table 9: Relationship between sub systems and php files

Similarly, the methods designed for the class diagrams were embedded in the individual files. PHP require_once() was used to get the functionality of one class to another class.

Table 10: Relationship between classes and php files

Class	Method	PHP files
System User	Login()	Index.php
IssueClerkHm	searchIssuedReports() issueABST()	searchRF.php issue.php
SpecimenReceivingClerk	searchReqForms() sddReqForm()	searchRFs.php addRF.php
MLT	searchReqForms() generateLabSheet() updateLabSheet() generateABST() searchLabSheet() searchABST()	specimen.php genLS.php updateLS.php genABST.php srchLS.php srchABST.php
Consultant	searchReqForms() searchLS() searchABST() authorizePrnt() audit() analyze()	serchRF.php serchLS.php searchABST.php printAuth.php audit.php analyse.php
Administrator	updateSystem() addUser() addAntibio() addBact()	sysUp.php addUser.php addAbio.php addBact.php

6.1.2 Data exchange between sub systems

When a class needs to use functionality of another class, data exchange occurs between participating php files and methods. PHP super global array of `$_POST[]` was used to pass values from one interface to another.

E.g. `specimen.php` will collect the data which belongs to a request form and uses `printRcpt.php` to get it printed as a receipt. In this situation `specimen.php` will pass the `$_POST[]` array to the `printRcpt.php`. `PrintRcpt.php` will capture the elements of `$_POST[]` array with `isset()` method and assign values to local variable for the printing purpose. Then `printRcpt.php` will call the `printPage()` javascript function to activate the operating systems default print method.

```
<script type="text/javascript">
function printPage()
{
    if (typeof(window.print) != 'undefined') {
        window.print();
    }
}
</script>
```

This will complete the printing of the receipt.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

6.2. Security

Security of the system is implemented by the user authentication and the sessions.

Each time user logs in to the system, Login interface will check the password given by the user with password stored in the database. Passwords are encrypted with 32 bit MD5 encryption algorithm before storing in the database. When user gave the password to the Login interface, interface will do the encryption of the password in passed by the \$_POST method. Then it will compare 2 encrypted strings for any mismatch.

Since row passwords are not in transit with the \$_POST array, session hijacking is difficult with this approach.

Successful user authentication will create a session array. This array will be checked while trying to load each page and if not successful, user will be forwarded to the login page. To improve the security, developer has added the additional parameter (\$_SESSION[User_AGENT]) to the \$_SESSION array. Since this is encrypted in the login page, session is more robust to external attack.

.htaccess file is created in the site root. This will define the file access and prevent file listing if the file requested is not available.

6.3 Database design

Database created using InnoDB database engine of MySQL. With this approach, foreign key can be defined and it is possible to create relationships among the tables of the database.

Following are some of advantages of the InnoDB over MyISAM database engine.

- InnoDB recovers from a crash or other unexpected shutdown by replaying its logs. MyISAM must fully scan and repair or rebuild any indexes or possibly tables which had been updated but not fully flushed to disk.
- MyISAM relies on the operating system for caching reads and writes to the data rows while InnoDB does this within the engine itself, combining the row caches with the index caches.
- InnoDB stores data rows physically in primary key order while MyISAM typically

stores them mostly in the order in which they are added. Storing the rows in primary key order corresponds to the MS SQL Server feature of “Clustered Indexes” and the Oracle feature known as "index organized tables." When the primary key is selected to match the needs of common queries this can give a substantial performance benefit.

InnoDB allows Joint database queries, which make the data extraction much faster than the isolated sequential set of queries.

6.4. Form Processing

Java script used for client end form validation. This approach will free the server of the burden of form validation using PHP and make the form processing faster.

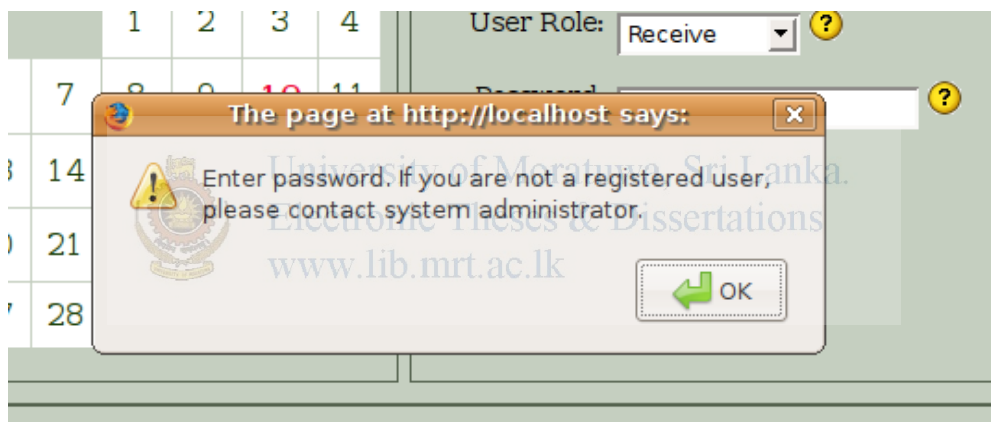


Figure 20: Form Validation with Java Script

6.5 Form Printing

Form printing is done with special print cascading style sheets. This will remove the text boxes and buttons appearing on the printed paper. In addition to regular classes, .print_ignore { display: none; } class will be used to hide unnecessary element in the HTML form.

6.6. Dynamic Help

Instead of having a user manual, system will use dynamic help text to guide users where necessary. Java Script is used for this purpose [14].

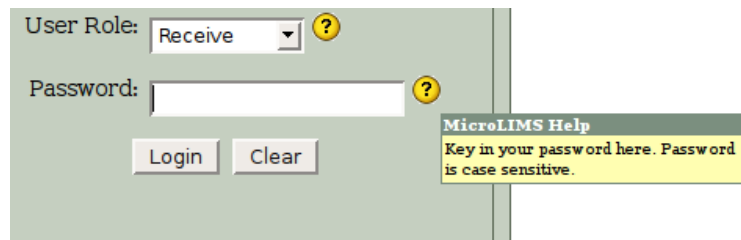


Figure 21 : Dynamic help text activation

6.7 Sound like name search

When searching for patient names, hospital names etc, MySQL Soundex algorithm will be used. Soundex algorithm will extract similarly sounding words from the data base.

A screenshot of a search interface. At the top, there is a search bar with the text 'Search for:' followed by a text input field, a dropdown menu set to 'by Name', and a 'Search' button. Below the search bar is a table with six columns: BHT, Name, Hospital, Specimen, Date Received, and Date Issued. The table contains two rows of data. A watermark for the University of Moratuwa, Sri Lanka, is visible in the background.

BHT	Name	Hospital	Specimen	Date Received	Date Issued
08/2111	Amarasekare	TH Kalubowils	Blood	9-10-2008	Not issued
09/8876	Amarasekera	GH Nagoda	Urine	9-10-2008	Not issued

Figure 22 : Use of Soundex in search options

6.8. Dynamic Charting Library

PHPlot (version 5.0.5) open source charting library was used to generate charts to visualize data. PHPlot [46] creates images using the PHP GD library dynamically. The advantage of using the server to create an image (server side scripting) is that one does not have to worry about browser compatibility or client operating system compatibility issues. PHPlot is a

graphics library which provides a means by which a PHP enabled web server can create and manipulate graphs as objects and display the completed graph as an image. Data sets passed to PHPlot use a very convenient way for database driven sites, in rows with y coordinate data. Bar charts, Pie charts and various other charts can be generated through this library. AJAX functionalities were used where appropriate [6].

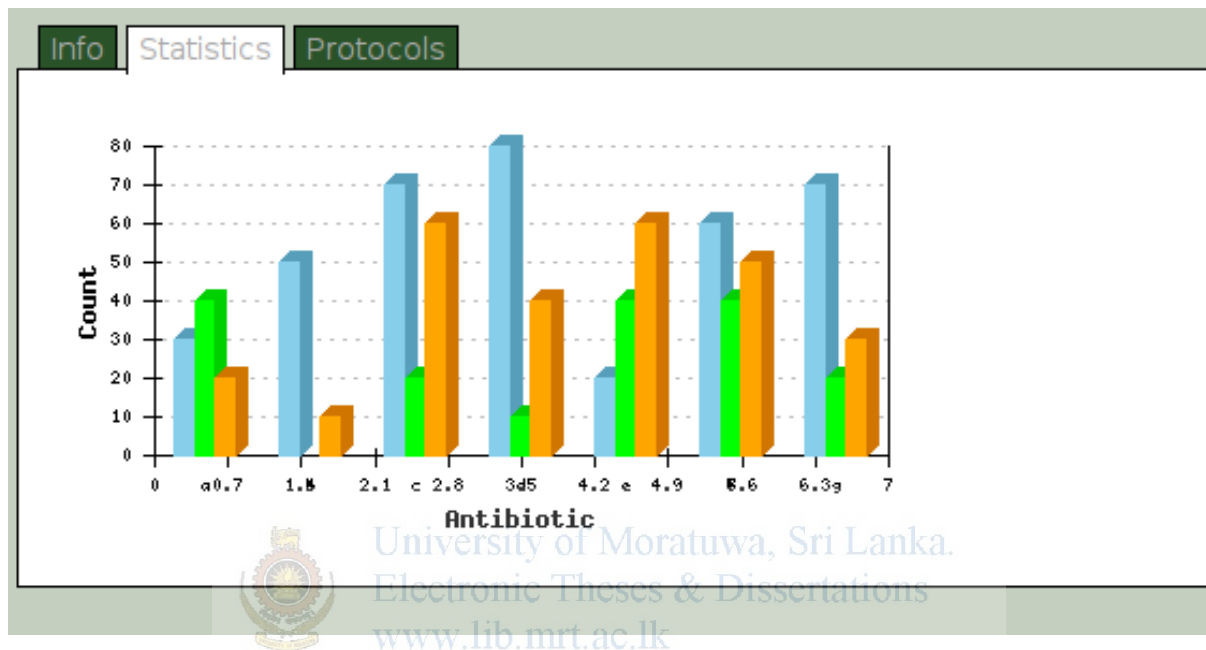


Figure 23 : Sample charts developed using PHPlot charting library

Summary

Salient features of the implementation are described in this chapter with reasons for choosing particular development solutions.

Next chapter will focus on the evaluation of the software.