

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1. System Implementation

This chapter contains some explanations regarding the testing of the application, individually (by module) or by whole running application (integrating it to the existing whole system).

5.2. Implementation specifications

The application will be tested with these hardware and software. The first one is the hardware, which is the notebook.

Toshiba Qosmio F50

- ✓ Processor: Intel Core 2 Duo T9400, 2.53 GHz
- ✓ RAM: 3 GB RAM
- ✓ VGA: NVIDIA GeForce 9600M GT, 512 VRAM
- ✓ Display: 1280 x 800 pixels, 15"1 wide-screen with 16:10 ratio
- ✓ Hard Disk Capacity: 400 GB, with partitioned 72 GB from the server's partition
- ✓ Operating System: Windows Vista Professional Service Pack 2

The application is tested using these following softwares.

- ✓ **Database**
 - MySQL Server 5.0

- MySQL Query Browser 1.1.20

- ✓ **Web Server**
 - XAMPP Apache Friends Server 1.4.16

- ✓ **Browser**
 - Mozilla Firefox 3.6.3
 - Google Chrome 5.0.375.70

5.3. Software Configuration

There are a few steps to follow before the user may start testing the application. It includes the installation of a few programs and configuring a few applications and files. The following steps need to be completed for the server: (Note that the steps explained is only applied to Windows OS)

[1] Install the XAMPP Apache Friends software, and follow the installation guidelines. The attached file is named “xampp-win32-1.4.16-installer”. Should the tester’s OS version is different (such as windows 64 bits or MAC), please visit the Apache Friends home site (<http://www.apachefriends.org/en/xampp.html>) for an installer compatible to the tester’s OS, and follow the instruction given by the web.

[2] Install MySQL Server 5.0, and follow the installation guidelines. The attached file is named “mysql-essential-5.0.16-win32”. Again, should the OS of the tester is different, please visit the MySQL website

(<http://dev.mysql.com/downloads/mysql/>) for another type of installer.

During the installation, there are a few things to note:

- a. Leave everything by default (do not change any configuration that is recommended by the installer).
- b. Set the port of the database to 3306.
- c. Set the username and password to “root” and “root” respectively without the semi-colon. Both of these points (b and c) are crucial, since the database will not connect if these points are not followed properly.

[3] Install MySQL Query Browser 1.1. This step is optional for the server to run. But just in case if the query browser is somehow useful for the tester in any way (such as looking at the real database tables and its attributes, along with the data types), this application will provide the utilities necessary. Should the OS is different, please visit this website: <http://dev.mysql.com/downloads/workbench/>. Note that the name is completely different if the installer is downloaded from website compared to the application attached. This is because the version of MySQL 5.2 onwards changes the name to workbench instead of query browser, due to its merged functionality with other applications [18].

[4] Copy the project folder named “DailyActivities” to the Apache Friends web server’s folder. If the user does not alter the destination of which the Apache Friends application is installed, then it should be at the path:

C:\Program Files\xampp\htdocs. However, if the user alters the destination of the installer, then it should be at: [UserSpecifiedPath]\xampp\htdocs.

- [5] Check the PHP version of the server's script. To do this, go to the folder where your Apache Friends web server is installed (similar to step #4). Create a folder using any name (for instance, test). Go into the folder then create a file using a notepad, save it by the type of "all files" with .php suffix attached to its name (for instance, version.php). Open the file, write these following codes:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <title></title>
  </head>
  <body>
    <?php phpinfo()?>
  </body>
</html>
```

Fig 5.3a – Code to check PHP version

Open a browser. Write the following URL:

<http://<domain-name>/<your-folder-name>/<your-file-name>.php>

For instance:

<http://localhost/test/version.php>

However, be advised that “localhost” can be altered with the domain name of the server. It is written localhost since client is the server itself.

After entering the URL, the browser will display something like below:



The screenshot shows a purple header bar with the text "PHP Version 5.0.5" on the left and the PHP logo on the right. Below the header is a table with 16 rows of system and configuration information.

System	Windows NT FLARE-PC 6.0 build 6002
Build Date	Sep 22 2005 03:21:42
Configure Command	cscrip/nologo configure.js "--enable-snapshot-build"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Program Files\xampp\apache\bin\php.ini
PHP API	20031224
PHP Extension	20041030
Zend Extension	220040412
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, ssl, sslv3, sslv2, tls

Fig 5.3b – Display of PHP scripts information

Take a look at the topmost text. Like the example figure above, the version of the PHP is 5.0.5. The minimum requirement is 4.0.0 and above, but it is recommended for the client and server to update the version by using the next step.

[6] This step is not necessary for the user whose PHP version is above 4.0.

However, should the PHP version is below the requirement, the server is required to update their PHP version by installing the latest version of the web server. It is not, however to be in general. Some of the web servers

require the PHP scripts itself to be updated. Apache Friends web server installer comes in bundle with the latest support of the PHP technology, so upgrading the server's version will automatically upgrade the PHP version too (See the #1).

And these are the following steps that need to be completed for the client side:

- [1] Install Mozilla Firefox or Google Chrome. This step is also optional should the user want to use the browsers. The other reason of why this step is still mentioned because the application itself has been tested with these two browsers.
- [2] Do the same steps of #5 and #6 in the client side.

5.4.Implementation results

After following the above procedures, the application is ready to be used. These are the results of the testing phase. The further testing result by the user himself is displayed in the appendix section.

No	Functionalities	Expected Output	Status
1	Login	The user is logged in, giving a unique session	OK
2	Logout	The user is logged out, destroying the session created	OK
3	Change password	Change the user's current password	OK

4	Create a plan	Create a plan	OK
5	Edit a plan	Change the planning of the activity	OK
6	Delete a plan	Delete a planned activity	OK
7	View all users	Display all the users along with their divisions	OK
8	Add User	Add a user into database, enabling them to log in	OK
9	Remove user	Delete a user from a database	OK
10	Edit user	Edit the user's profile	OK
11	Reset user's password	Reset the user's password to the password specified by the admin	OK
12	View all divisions	Display all the divisions	OK
13	Add division	Add a division	OK
14	Edit division	Edit the division's name and its position	OK
15	Remove division	Delete a division, but halting it if there is still a user inside.	OK
16	View all classifications	Create a classification for a plan	Not OK
17	Edit classifications	Edit the class name, along with its position	Not OK
18	Remove classifications	Remove the classification of an activity, but restrain it if there is still	Not OK

		an activity inside	
19	Report all activities	Create a table reporting of the summarized activity done for (a) certain day(s).	Not OK

Table 5.4 – Implementation Results

5.5.Implementation issues

In this sub-chapter, all of the issues that are encountered during the testing will all be listed here.

5.5.1. Incompatible PHP version

The PHP scripts are not all updated within each workstation. Due to the shortage of time, it has not yet been determined and tested whether the one that needs the PHP scripts update is the server or the client. Although the PHP scripts are not installed, PHP still has a version within a workstation. There are a few issues that arise out of the codes. After a few researches and tests that is run in the application, it turns out that the problem originates from these type of codes:

```
<input type="hidden" name="id" value="<?=$activityList->get($i)->getId()?>">
```

Fig 5.5.1 – Problematic Part of Application Codes

Please take a look at the PHP tag. This kind of tag will give out error for the PHP version that is outdated. The tag `<?= ?>` is only supported at PHP version 4.0. So the workstations that have not yet been configured to

PHP version 4.0 or above will cause the application to work improperly. The solution has been given at point #5 and #6 at sub-chapter 5.4.

5.5.2. Tree root system problem

Another problem that arise during the testing is that if a root is deleted, its leaves will point to a root that does not actually exist in the database, possibly creating an infinite loop in some processes within the application (such as tracing parent). This is caused by the table's constraints of the foreign keys that are declared in the database. The current constraint is "No Action", making the table of one end unaffected by the changes of the other end of the foreign key. The problem is illustrated in the figure below.

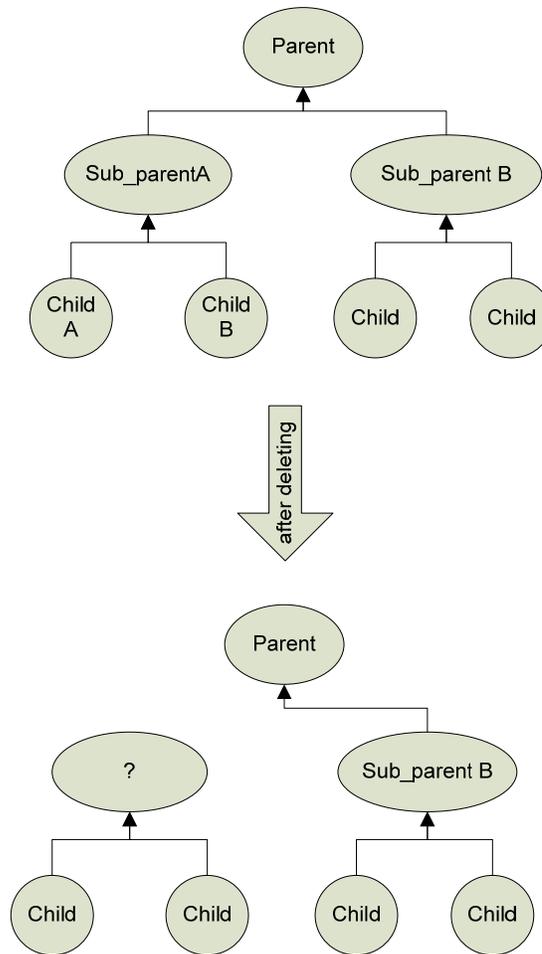


Fig 5.5.2a – Problematic Tree System

The way to fix this problem is simply by changing the constraint to from “No Action” to “Set Null”. This constraint will nullify another end of the foreign key should one end of the entry is deleted. This process of fixation is illustrated in the figure below.

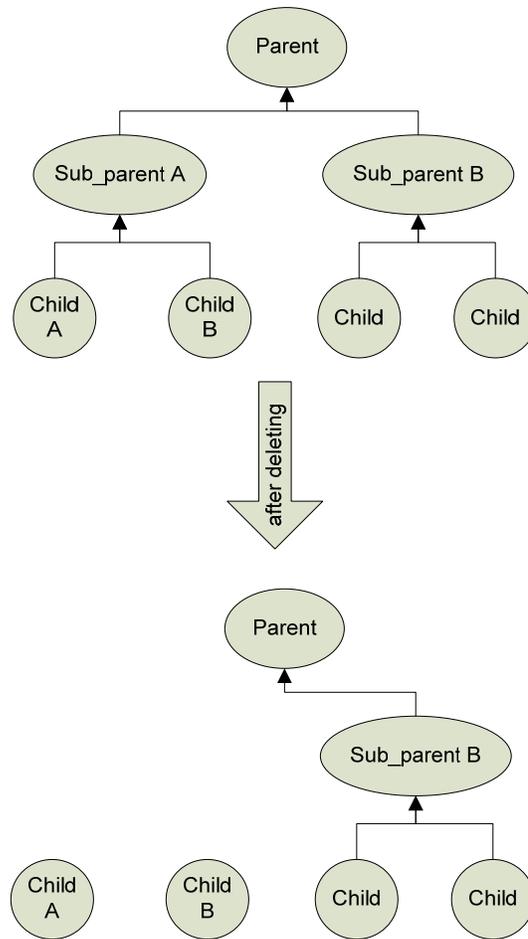


Fig 5.5.2b – Non-problematic tree system

CHAPTER 6

DISCUSSION

6.1. Discussion Summary

The project was requested by Diyurman Gea, the current head of IT of Binus International. This project is meant for only IT department. At the first glance the IT department seems small. However, there may be chances that the head of IT estimated the growth of this department. A growing department will mean a wider span of control. And the wider the span of control goes, the lesser the head of IT will get a grip on. Thus, rather than waiting for the growth later, the head of IT decides to take early measurement of the future growth, which is developing this application. That is why the feature of the infinite tree system is created (refer to chapter 4 for details).

There are some certain factors of why does all the technologies are provided, aside from the ones being explained earlier in the chapters (chapter 2 and 4). Since it is quite closely related to the integrating process, it is further discussed here rather than being discussed on the previous chapters.

The PHP is used as the main language for the web instead of other languages such as JSP (java), or ASP (C++) is because the whole web server that the application is to be integrated on only supports PHP. Should the language differs, the application will be a standalone application, unable to be integrated to the currently existing application that the IT department has.

Aside from the reasons that have been stated on the previous chapters, MySQL has much more advantages. Compared to other database software such as Microsoft Excel, Oracle, or FileMaker, MySQL is the most suitable ones for the development of this application. One, of all reasons, is that it is free. Developers will not develop an application that will take quite amount of time, just for one project, and pay for it. Another advantage is that MySQL supports a lot of OS such as Windows, MAC OSX, Linux and SunSolaris.

As of why the user interface design is left behind, it is because the head of IT himself, said that this aspect should not be prioritized due to the extremely short time frame. The other functionalities should first work properly, then the user interface will be worked upon.

In addition, there is no change of plan, since every functionality and approach has already been calculated and specified from the very beginning of this project.

6.2.Result

Due to the shortage of time, the implementation process only stops at functionality testing, and not proceeding to the integration testing. Although it will come to that, it cannot be stated that the integration process will run smoothly. Hence there is no clear-cut result of the integration testing.

However, there is one problem that is considered to be quite significant that arise out of the solution that has been implemented.

Very large amount of redundant data

With the application has been developed in such a way, the requirement of a database administrator is inevitable. If there already exists a database administrator from the current IT department, then the workload of the administrator will be increased quite significantly. It is due to the application's processes of monitoring that have not the ability to self-archive.

Every activity monitoring process is recorded in the database. Let's do the math. One activity monitor result will produce one row of database. Assume that each staff has three, at the very least, of daily activity for five days (Monday to Friday). This would mean that every week, the staff will have 15 rows of record. Every month will make it 60 rows, and every year will make it 720 rows. Imagine if there are five staffs, at least, within the IT department. That will make the database's weight to be increased by 3600 rows, which is extremely significant reduction time of the query to be done.

If left unattended, the database may not last for another year. Hence, a database administrator will need to archive or transfer these entries somewhere else once at every specified interval, which will lighten up the database's query process.