The Asset and Maintenance Management System (AMMS) Project

Rosina Surovi Khan
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
Email: surovi.cse@aust.edu

***Abstract*— With the advent of ever evolving technology, it becomes necessary to manage IT information efficiently, and managing it the right way cuts down maintenance costs. A variety of IT-maintenance software products exist in the market which can be used to manage IT equipment efficiently. But often these IT-maintenance software products do not always fulfill all the IT requirements of an organization. For example, most of the IT-maintenance software products available in the market manage hardware and software assets quite well but fail to capture non-pc data such as telephone system and office material and also lack the ability to support heterogeneous networking in MAC environment. In such cases, developing a good interface to a database where IT information are stored often helps to provide good overviews, good updates and low maintenance costs. This paper is just about such a system that the author worked on. The system is called the Asset and Maintenance Management System (AMMS).**

Keywords- IT-maintenance, Interface, reports, Data-entry forms, SQL database

I. Introduction

Technology has been advancing very fast. With it the complexity of IT infrastructure too. Different companies and organizations look forward to gain competitive advantage at the same time. The complexity of IT-infrastructure must be resolved somehow.

InfoCom GmbH, a reputable telecom company in Stuttgart, Germany wishes to manage its IT resources more efficiently, with faster access to IT information and reduced costs. For this reason, the author creates a relational SQL database first where relevant IT information will be stored and then creates an interface using JavaScript, PHP and HTML for extracting, downloading, inputting/modifying information of the IT-database. The system interface comprises of reports and data-entry forms in HTM format. This system has been named as ‘Asset and Maintenance Management System’ (AMMS).

II. System Requirements of AMMS

InfoCom needs to manage information not only on assets like hardware, software but also information on users, their rooms, computer backups, office supplies and office keys.

Starting with users, InfoCom needs to keep track of user ids, user names, their addresses, contact numbers, office email addresses, email code numbers and individual backup codes.

Next coming to the asset hardware, information on the hardware id, hardware name, hardware description, purchase description, purchase date and distributor will have to be maintained.

For the asset software, information on software id, user id, hardware id, software name, software description, purchase description, purchase date and distributor will have to be maintained.

For Backups, InfoCom would like to manage information on back up number, admin user id, hardware id, backup description and time and date of the backup.

For Office keys, information on key id, user id, key description, purchase description, purchase date, distributor and the duration the user will have the key will have to be maintained.

For Office Supplies, InfoCom wishes to maintain information on office supply id, office supply name, supply description, purchase description, purchase date, distributor, the date the supply will be used in the office, where the supply will be used and what is the remaining status of the supplies after usage.

For telephones, information on telephone id, extension number, patch field number, purchase description, purchase date, distributor and telephone description will have to be maintained.

For rooms, InfoCom would like to manage information on room number, user id, hardware id, telephone id, office supply

The author worked on the Asset and Maintenance Management System Project in InfoCom GmbH, Stuttgart, Germany in affiliation with the Stuttgart University of Applied Sciences as a thesis in partial fulfilment for the requirement of Masters of Science in Software Technology from August 2004 to February 2005.

id, the duration the user will have the room and whether the user sits left or right of the room.

A debate arises whether a customized solution or a commercial product should be used in order to maintain the IT information at InfoCom. It, therefore, becomes necessary to judge the pros and cons of some commercial products available in the market that the author researched.

III. Evaluation of the Commercial Products under research

The various IT-Maintenance software available in the market that may be of interest are the Nimbus, GPMate, Imaging Symantec Ghost 7.5 , Novell Zenworks 3.2, IT Asset Management System, Snow Inventory 2.07, Jlocator, ManageSoft, Syslist, Asset Manager and MAXIMO® MainControl.

Nimbus Software:

Pros

Nimbus software is suited to managing a wide variety of IT equipment like servers, network services, switches, routers, emails and graphical display. [1]

Cons

Nimbus software is rather sophisticated. At InfoCom GmbH, the IT-maintenance need is more on hardware like CPU, memory, monitors, drives, printers etc and software like operating system, Microsoft office, antivirus etc.

GP Mate:

Pros

GP Mate software is suited to supporting maintenance by work orders to which a lot of tasks can be assigned. [2]

Cons

GP Mate requires a lot of manpower. Here in InfoCom the author has only been assigned to do the job of IT-maintenance.

Symantec Ghost and Novell Zenworks:

Pros

Symantec Ghost manages big-sized software efficiently while Novell Zenworks manages small-sized software well.

Cons

It becomes a complex system to utilize both Symantec Ghost and Novell Zenworks for managing big and small-sized software separately. At InfoCom, the IT-maintenance requirement is not only for software but also hardware, telephone system and office material.

Asset Management System (AMS):

Pros

Asset Management System (AMS) is a good tool for supporting the management of IT infrastructure like computers, monitors, printers and software installations. [3]

Cons

AMS lacks the ability to support heterogeneous networking in Mac environment and input non-PC data such as, information on telephone system, office material manually into the inventory database.

Snow Inventory 2.07:

Pros

Snow Inventory generates the most detailed reports which list what hardware is there in workstations and what software has been installed. [4]

Cons

Like AMS, Snow Inventory lacks the ability to support heterogeneous networking in Mac environment and input non-PC data such as information on telephone system, office material manually into the inventory database.

Jlocator:

Pros

Jlocator is not only able to locate assets physically but also provide information on what assets are contained in a place. [5]

Cons

Like AMS and Snow Inventory, Jlocator lacks the ability to support heterogeneous networking in Mac environment and input non-PC data such as information on telephone system, office material manually into the inventory database.

ManageSoft:

Pros

ManageSoft tracks and analyzes software and hardware assets with reduced IT costs and improved IT service levels. [6]

Cons

Like AMS, SnowInventory and Jlocator, ManageSoft lacks the ability to support heterogeneous networking in Mac environment and input non-PC data such as information on telephone system, office material manually into the inventory database.

Syslist:

Pros

Syslist can quickly search the organization's inventory for useful information such as serial numbers, IP addresses or a list of PCs with a given software type installed. Unlike AMS, Snow Inventory, Jlocator and ManageSoft, Syslist is uniquely built to accommodate non-PC data such as, information on telephone system, office material manually into the inventory database. [7]

Cons

While Syslist does support Unix and Linux environment, it does not support heterogeneous networking in Mac environment.

Asset Manager

Pros

Asset Manager records complete hardware and software information, including type, manufacturer, serial and bar code numbers, configuration, physical location, assignee etc. [8]

Cons

Like AMS, Snow Inventory, Jlocator and ManageSoft, Asset Manager lacks the ability to support heterogeneous networking in Mac environment and input non-PC data such as, information on telephone system, office material manually into the inventory database.

MAXIMO MainControl

Pros

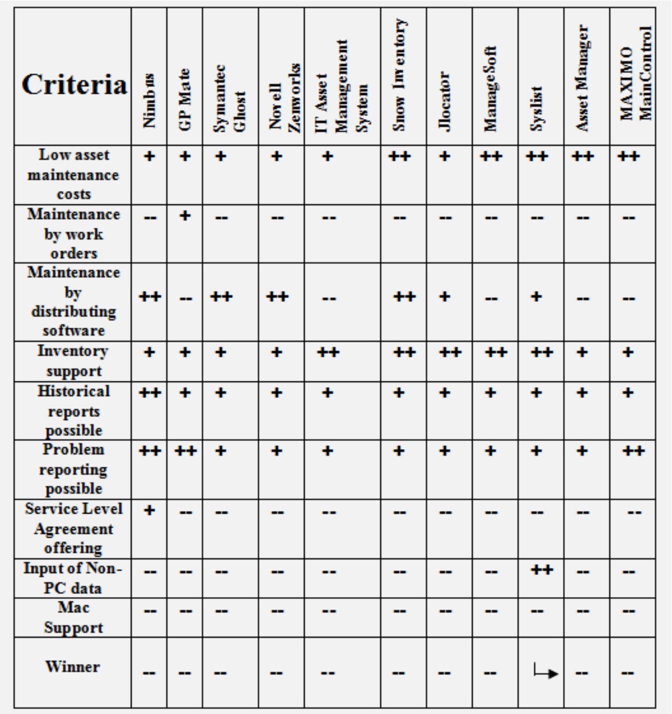
MAXIMO MainControl not only helps one to know what assets are there and in which location but also gives full control of the IT infrastructure so one can add, move or retire equipment and software as needed. [9]

Cons

Like AMS, Snow Inventory, Jlocator, ManageSoft and Asset Manager, MAXIMO Main Control lacks the ability to support heterogeneous networking in Mac environment and input non-PC data such as, information on telephone system, office material manually into the inventory database.

A. *Individual Product Evaluation*

TABLE I. OVERVIEW OF ASSET SOFTWARE



Criteria	Nimbus	GP Mate	Symantec Ghost	Novell Zenworks	IT Asset Management System	Snow Inventory	Jlocator	ManageSoft	Syslist	Asset Manager	MAXIMO MainControl
Low asset maintenance costs	+	+	+	+	+	++	+	++	++	++	++
Maintenance by work orders	--	+	--	--	--	--	--	--	--	--	--
Maintenance by distributing software	++	--	++	++	--	++	+	--	+	--	--
Inventory support	+	+	+	+	++	++	++	++	++	+	+
Historical reports possible	++	+	+	+	+	+	+	+	+	+	+
Problem reporting possible	++	++	+	+	+	+	+	+	+	+	++
Service Level Agreement offering	+	--	--	--	--	--	--	--	--	--	--
Input of Non-PC data	--	--	--	--	--	--	--	--	++	--	--
Mac Support	--	--	--	--	--	--	--	--	--	--	--
Winner	--	--	--	--	--	--	--	--	↳	--	--

+	**:** indicates good support of the given functionality
++	**:** indicates strong support of the given functionality
--	**:** indicates does not support the given functionality
↳	**:** indicates winner product fulfilling most of IT requirements at InfoCom

B. *Comparative Market Products Analysis*

All the asset software provide low asset maintenance costs, especially Snow Inventory, ManageSoft, Syslist, Asset manager and Maximo MainControl . GP Mate is the only software to support maintenance by work orders [2]. Software such as Nimbus, Symantec Ghost, Novell Zenworks, Snow Inventory, Jlocator and Syslist support maintenance by network distribution while software such as IT Asset management, ManageSoft and Maximo MainControl provide maintenance of assets through inventory support only [3,6,9]. All the software have inventory support, historical reporting and problem reporting options. While only Nimbus offers service level agreements [1], input of non-PC data into the inventory database is only possible through Syslist [7]. None of the software provides MAC support in heterogeneous networking environment. Based on the above comparisons, Syslist is the ideal winner to meet most of the IT-Maintenance requirements at InfoCom because not only does it support management of hardware and software assets but also it captures non-PC data.

C. *Make/Buy Discussions*

While Syslist helps to capture data in Unix/Linux environment and also non-PC data, it does not support heterogeneous networking in MAC environment, which is an essential criterion in InfoCom IT management.

Moreover, 16+ licenses for Syslist, covering a 100 systems per license for InfoCom’s around 35 users (at the time of writing the thesis) would cost US $197 x 35= approx. US $6000 yearly which is not very ideal from economic standpoint. [7]

The requirement for Syslist is a Linux/Apache web server running PHP 4.2.1 or greater and MYSQL. [7] InfoCom already has the necessary software tools such as PHP and Microsoft SQL Server installed. So it does not want to cover extra expenditure on the similar software tools that Syslist provides.

Under these grounds, Syslist is rejected although it succeeded in becoming a winner in meeting InfoCom’s IT management requirements. And thus the decision of customized development of a Javascript/html interface, namely the Asset and Maintenance Management System (AMMS) interface using PHP for an SQL IT-database is taken.

IV. DATABASE DESIGN

A. *Entity Relationship Modeling*

According to the system requirements, the IT-information that need to be included in the AMMS system are the users themselves, their rooms, the hardware and software assets, telephone system, office supply and office key. Each of these assets is one family and will represent a database table. How a specific table is linked to other table(s) become clear through the visualization of an entity-relationship diagram as shown in Figure 1. From this diagram, it can be seen that Room is a central relationship linked to UserAll, Office_Supply, Hardware_Asset and Telephone_Asset. (The name ‘UserAll’ has been used and not ‘User’ because ‘User’ is a reserved keyword in MicrosoftSQL Server, the software tool that has been used in the thesis to create the database). Room is a relationship linking to UserAll, Office_Supply, Hardware_Asset and Telephone_Asset so that it can be tracked what assets are there in Room which will become a separate table in the relational schema as shown in Figure 2. For each user, there can be many hardware assets, office supplies and telephones. On the other hand, one Hardware_Asset, one Telephone_Asset and one Office_Supply can belong to one User only. For these reasons there is a 1-n-m-o relationship among UserAll, Hardware_Asset, Telephone_Asset and Office_Supply.

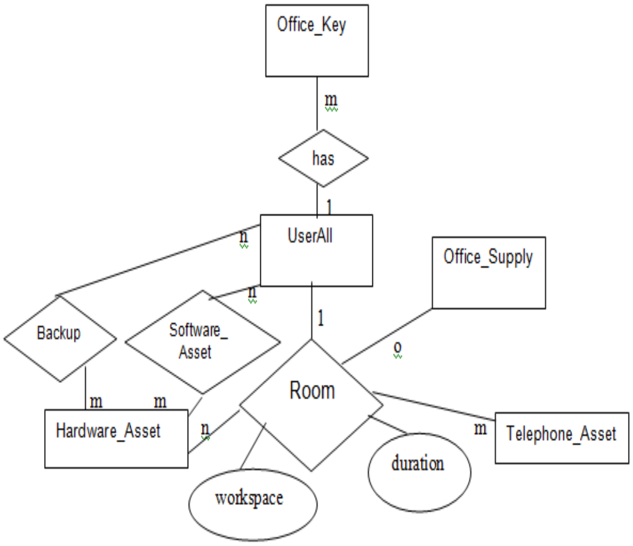


Figure 1. Entity Relationship Diagram

Backup and Software_Asset are also relationships linked to UserAll and Hardware_Asset which have many-many relationships (n-m). This is because every user can access software assets on many hardware assets and every Admin user can also backup many hardware assets. On the other hand, one hardware asset can be backed up by more than one Admin user and one hardware asset’s software may be accessed by many users. A User in UserAll table has a one to many relationship with Office_Key (1-m) because one user may have many office keys while one office key may belong to one user only.

B. Relationships between Different Database Tables in AMMS

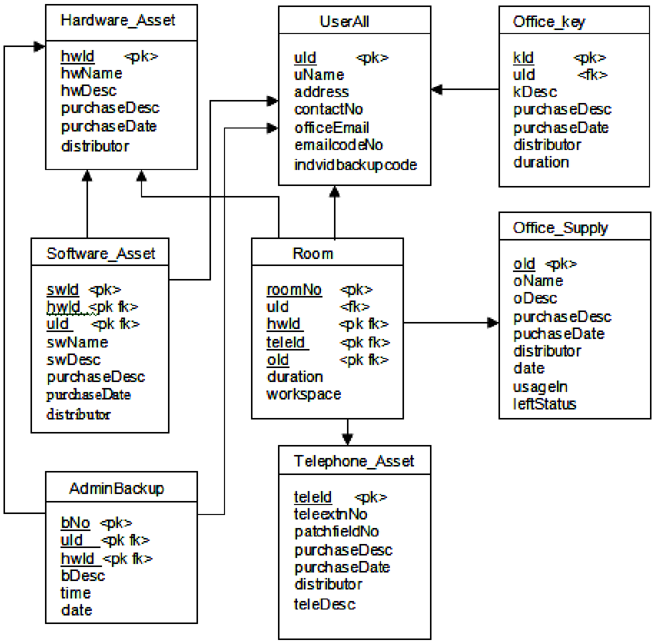


Figure 2. Relationships between the Different Database Tables

Now that the entity relationship model has been designed, the database can now be designed through a relational schema, showing the relationships between different database tables in the system. From the ER model, the entities like Hardware_Asset, UserAll, Telephone_Asset, Office_Supply and Office_key as well as the relationships like Room, Backup, Software_Asset become separate database tables in the relational design as shown in Figure 2.

V. DESIGN OF GRAPHICAL USER INTERFACE

The next step would be to design a graphical user interface (GUI).

A. Report Design

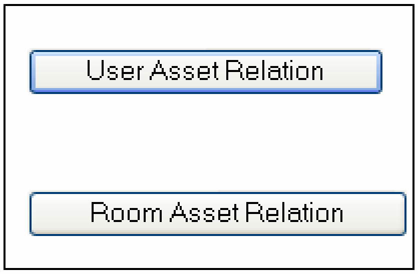


Figure 3. Report's first page

The Report design has two option buttons: **User Asset Relation** and **Room Asset Relation.**

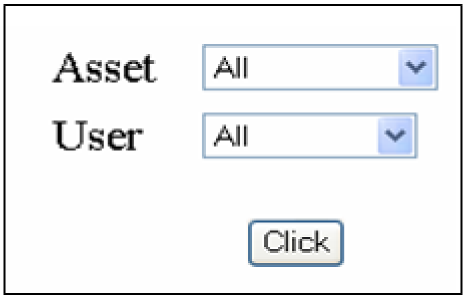


Figure 4. Report's second page corresponding to User Asset Relation

On clicking **User Asset Relation,** this corresponding output comes up. There is a combo box for **Asset** and another combo box for **User**. An option for Asset and an option for User can be selected here.

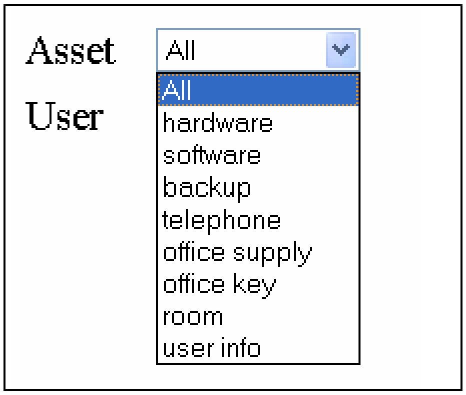


Figure 5. Asset Options for User Asset Relation

The options for the combo box Asset include hardware, software, backup, telephone, office supply, office key, room and user info. On choosing All from Asset combo box and All from User combo box and clicking **Click**, all asset information including hardware, software, backup, telephone, office supply, office key, room and user info come up corresponding to all users. It will simply give a summary of all these information.

Now it is worthwhile to look at what happens if each Asset such as hardware, software, backup, telephone, office supply, office key, room or user info is chosen and if 'All' from 'User' is chosen.

If 'hardware' is chosen from Asset and 'All' from User, the following page comes up.

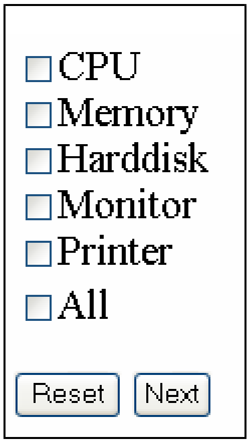
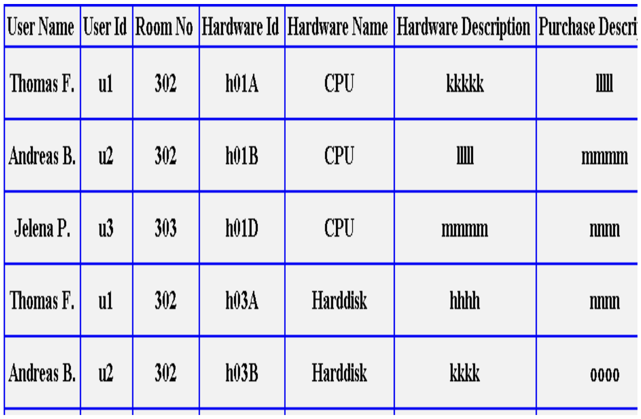


Figure 6. Hardware Asset Options

A check box appears next to each hardware such as, CPU, Memory, Hard disk, Monitor, Printer, and All. Checking each hardware such as, CPU, Memory, Hard disk, Monitor or Printer and clicking Next will show the particular hardware information corresponding to 'All' users. Checking 'All' from the above page will show all the hardware information corresponding to all the users. A part of the latter output can be shown as follows:

TABLE II. HARDWARE INFORMATION



User Name	User Id	Room No	Hardware Id	Hardware Name	Hardware Description	Purchase Descri
Thomas F.	u1	302	h01A	CPU	kkkkk	lllll
Andreas B.	u2	302	h01B	CPU	lllll	mmmm
Jelena P.	u3	303	h01D	CPU	mmmm	nnnn
Thomas F.	u1	302	h03A	Harddisk	hhhh	nnnn
Andreas B.	u2	302	h03B	Harddisk	kkkk	oooo

The above steps for choosing 'hardware' from Asset combo box and 'All' from User combo box are similar for choosing other option user names from User combo box. For example, if hardware is selected from Asset combo box and 'Thomas F.' from User combo box, the page corresponding to Figure 6 will come up as usual.

On checking 'All' on this page and clicking 'Next', the user will be able to see all the hardware information corresponding to Thomas F. On checking any of the other options such as CPU, Memory etc and clicking 'Next', the user will view all the particular hardware information corresponding to Thomas F.

If the user selects 'software' from Asset combo box and 'All' from User combo box, the user will view the following page.

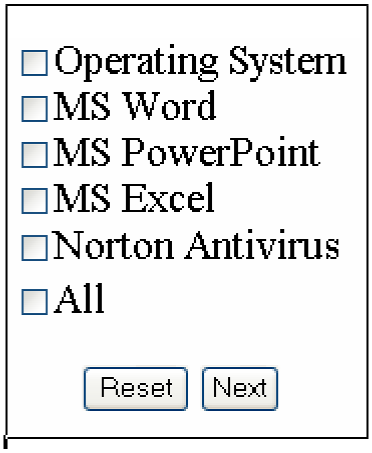
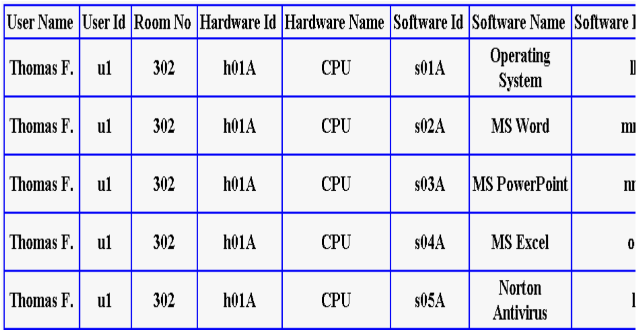


Figure 7. Software Asset Options

As with hardware options, here are some software options which can be checked individually and the particular software information corresponding to 'All' users can be viewed. A maximum of two options can be selected together and the corresponding information viewed. Otherwise check box 'All' can be selected to view all the software information.

Assuming the user selects 'All' from the software intermediate page and 'All' from User combo box, then the following output will be obtained.

TABLE III. SOFTWARE INFORMATION



User Name	User Id	Room No	Hardware Id	Hardware Name	Software Id	Software Name	Software I
Thomas F.	u1	302	h01A	CPU	s01A	Operating System	ll
Thomas F.	u1	302	h01A	CPU	s02A	MS Word	m
Thomas F.	u1	302	h01A	CPU	s03A	MS PowerPoint	n
Thomas F.	u1	302	h01A	CPU	s04A	MS Excel	o
Thomas F.	u1	302	h01A	CPU	s05A	Norton Antivirus	l

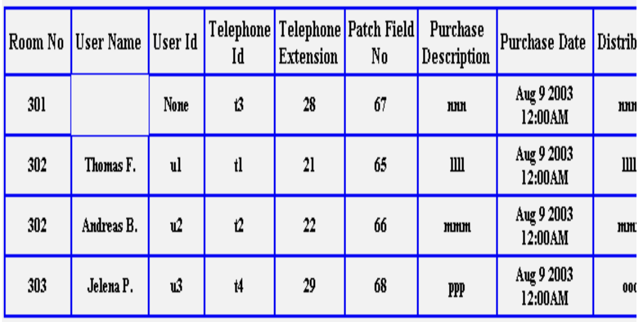
Here a part of the software information corresponding to Thomas F. is seen. The table is actually a long list of all the user names showing their corresponding software information.

Likewise, if the user selects 'software' from Asset combo box and a specific user name such as 'Thomas F.' from User Combo box, all or a maximum of two software options information can be viewed corresponding to the specific user.

However, if the user selects backup, telephone, office supply, office key, room or user info from Asset combo box, there is no intermediate page but the combo boxes file page is directed directly to the corresponding table information.

A part of final tabular report information for the asset telephone corresponding to the users of InfoCom is shown below.

TABLE IV. TELEPHONE INFORMATION



Room No	User Name	User Id	Telephone Id	Telephone Extension	Patch Field No	Purchase Description	Purchase Date	Distrib
301		None	t3	28	67	nnn	Aug 9 2003 12:00AM	nnn
302	Thomas F.	u1	t1	21	65	llll	Aug 9 2003 12:00AM	llll
302	Andreas B.	u2	t2	22	66	mmm	Aug 9 2003 12:00AM	mm
303	Jelena P.	u3	t4	29	68	ppp	Aug 9 2003 12:00AM	oo

So far everything that has been explained corresponds to User Asset Relation.

Similar steps for report design apply to the button Room Asset Relation in Figure 3. The page that comes up has the Asset options All, hardware, telephone, office supply and user info in relation to Room. Selecting 'All' from Asset and 'All' from Room, the user will be able to view all the hardware, telephone, office supply and user info information corresponding to All rooms. The user is able to view information more specifically by selecting only hardware, telephone, office supply or user info from Asset combo box and choosing 'All' from Room combo box in which case the particular asset info will correspond to all the rooms in the office. There is also the possibility to view asset information room-wise by selecting the specific room no. from Room combo box.

B. Data Entry Form Design

This design is typically of the format as shown in Figure 8. On clicking any of the option buttons in the form in Figure 8, an authentication form as shown in Figure 9 comes up. After filling in **User Login** and **Password** and clicking **Next** button on the authentication form, the corresponding data-entry form for the particular database table comes up.

The format of data-entry form for the table **UserAll** is shown in Figure 10. In the UserAll data-entry form, buttons for 'Insert Row', 'Edit Row', 'Delete Row', 'Previous Row' and 'Next Row' do the functions on clicks as their names imply. 'View Table' button can be used to view the changes after an insert, edit or delete is carried out.

A sample output on clicking 'View Table' can be shown as in Table V.

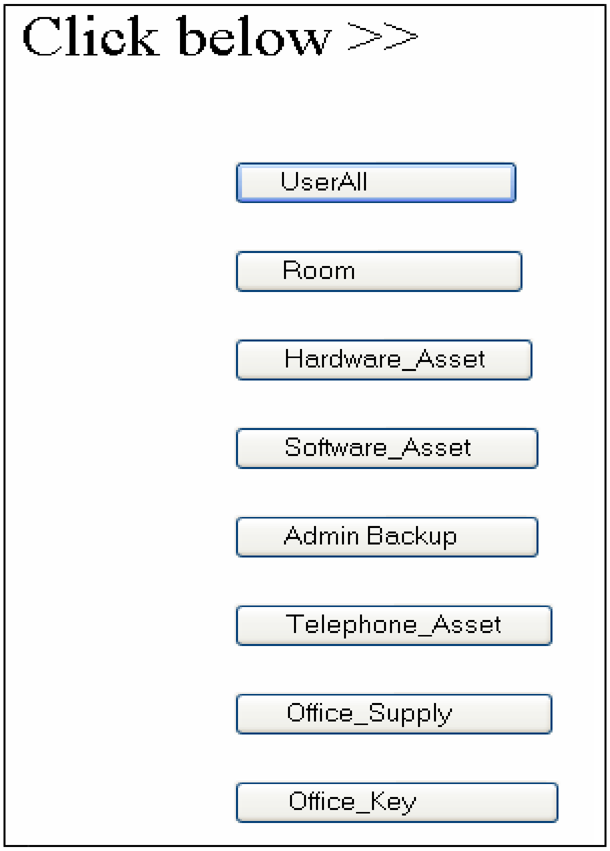


Figure 8. Main Form

The formats for other data-entry forms corresponding to the remaining database tables such as Room, Hardware_Asset, Software_Asset, Backup, Telephone_Asset, Office_Supply and Office_key are similar to that for UserAll data-entry form.

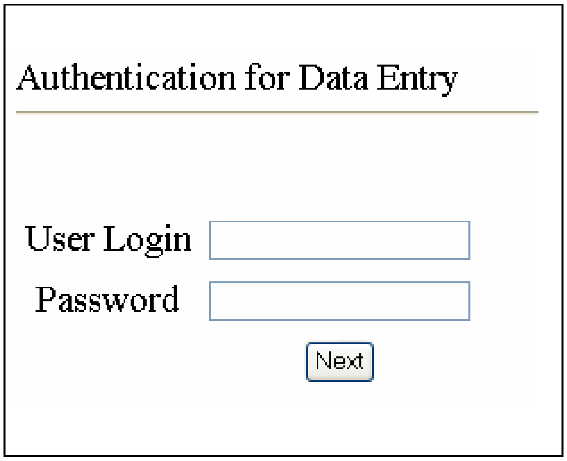
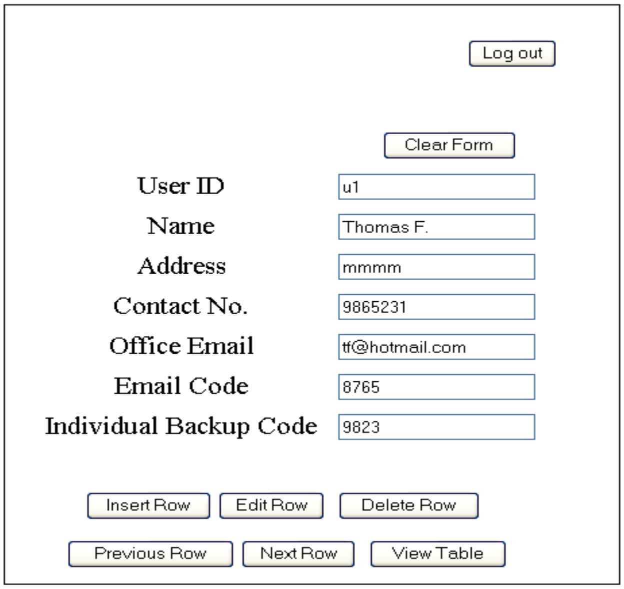


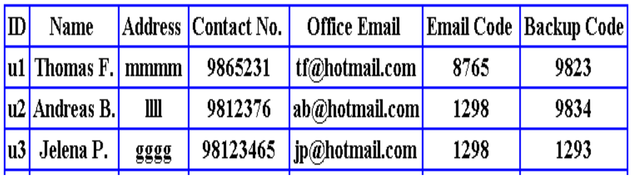
Figure 9. Authentication Form



User ID	u1
Name	Thomas F.
Address	mmmm
Contact No.	9865231
Office Email	tf@hotmail.com
Email Code	8765
Individual Backup Code	9823

Figure 10. Data-Entry Form for UserAll

TABLE V. USER INFORMATION



ID	Name	Address	Contact No.	Office Email	Email Code	Backup Code
u1	Thomas F.	mmmm	9865231	tf@hotmail.com	8765	9823
u2	Andreas B.	llll	9812376	ab@hotmail.com	1298	9834
u3	Jelena P.	gggg	98123465	jp@hotmail.com	1298	1293

VI. REPORT IMPLEMENTATION

Final reports can be viewed (as outlined in Sec V.A) based on which assets each user of InfoCom uses or what assets are there in each room For viewing hardware/software asset information, intermediate page comes up with the hardware/software options such as CPU, Memory, Hard disk etc or Operating System, MS Word, MS PowerPoint etc. In order to pass the value of 'User' (in asset-user relation page, for example) to the intermediate page, cookies were used by which means data can be passed from one page to another in JavaScript [10]. The intermediate page then links to the php page which connects to Microsoft SQL Server and uses the relevant SQL query to view hardware/software information.

For example for "All" user option, and "All" hardware option on the intermediate page , the intermediate page links to the following piece of PHP code.

```
<?php
 $myServer = "192.168.0.163";
 $myUser = "sa";
 $myPass = "sa";
 $myDB = "ITInfo1";
 $s = @mssql_connect($myServer, $myUser, $myPass)
 or die("Couldn't connect to SQL Server on $myServer");
 $d = @mssql_select_db($myDB, $s)
 or die("Couldn't open database $myDB");
 $query="SELECT uName,UserAll.uId,roomNo,
 Hardware_Asset.hwId,hwName,hwDesc, purchaseDesc,
 purchaseDate,distributor FROM UserAll, Hardware_Asset,
 Room WHERE UserAll.uId=Room.uId AND
 Hardware_Asset.hwId=Room.hwId  ORDER BY
 hwName";
 $result =mssql_query($query);
?>
```

Code 1. Code to view Hardware Info

In the above code, first of all a Microsoft SQL Server connection is established with the variables for $myServer, $myUser and $myPass. $myServer is the variable for server name; $myUser is the variable for user name; $myPass is the variable for password. It may be mentioned here that a database 'ITInfo1' was created in Microsoft SQL Server initially with all the relationships as shown in Figure 2.

Next an SQL statement is established which selects the fields from the tables UserAll, Hardware_Asset and Room. The relationships between these tables can be seen clearly in Figure 2. The SQL statement is assigned to a variable $query whose output is passed on to the $result variable by means of the function mssql_query().

The above implementation steps for choosing 'hardware' from Asset combo box and 'All' from User combo box are similar for choosing other option user names from User combo box corresponding to the asset hardware. Likewise the implementation steps for displaying software information are similar to that for hardware information.

For viewing other assets such as backup, telephone system etc, there is no intermediate page and links directly to the php page with relevant SQL queries. The php codes for displaying the layout of these tables are similar to that shown in Code 1.

VII. IMPLEMENTATION OF FORM DESIGN

HTML, JavaScript, PHP and Microsoft SQL Server are the software tools that have been used to implement reports and data-entry forms. Data-entry forms for the different database tables have the buttons Insert Row, Edit Row, Previous Row, Next Row and View Table in addition to textboxes for the attribute values of the tables. All the buttons except Previous Row and Next Row buttons were implemented by directly directing to php pages on button clicks which connect to Microsoft SQL server and use the corresponding SQL queries such as insert, update, delete etc. Implementing Previous Row and Next Row was a bit complex because on these button clicks, they are directed to JavaScript functions as usual but here the JavaScript variable curRow (standing for current row) after being decremented or incremented need to be passed back to a php page on the server as part of a request. An inline frame needs to be created targeting to this php page so that the buttons Insert Row, Edit Row, Delete Row together with textboxes stay on the target php page, and the values of the textboxes change dynamically here, while the Previous Row, Next Row and View Table buttons stay statically on the page where the inline frame is defined. By creating an inline frame with the target php page as source, it is possible to send the JavaScript variable (curRow) back to php as part of a request from within JavaScript functions for Previous Row and Next Row. This method of sending JavaScript variable back to PHP on the server was used because JavaScript is client side and PHP is server side and hence, it is not possible to access JavaScript variable with PHP. In other words, PHP acts on the code before the document is delivered to the client and JavaScript acts on the code after it has been delivered to the client. [11]

The PHP code for Insert button written in the file insert.php for UserAll data-entry form is as follows:

```
<?php
     // Microsoft SQL Server Connection here
       $uId=$_POST['text2'];
       $uName=$_POST['text3'];
       $address=$_POST['text4'];
       $contactNo=$_POST['text5'];
       $officeEmail=$_POST['text6'];
       $emailCodeNo=$_POST['text7'];
       $invidbackupcode=$_POST['text8'];

       $query4="INSERT INTO UserAll VALUES ('$uId',
      '$uName', '$address',  '$contactNo','$officeEmail',
      '$emailCodeNo','$invidbackupcode')";
       $result4 = mssql_query($query4) or die ('Could
       not insert data');
       mssql_close();  ?>
```

Code 2. Code for Insert button

In insert.php file for Insert button, $_POST is used to retrieve the values of textboxes of UserAll data-entry form as seen in Figure 10. Once the textbox values were retrieved, SQL Insert query statement is assigned to the variable $query4 and the output of $query4 is assigned to $result4 by means of mssql_query function as shown in Code 2.

Similarly codes for “Delete Row”, “Edit Row” and “View Table” buttons on UserAll data-entry form are shown below.

```
  <?php

  //Microsoft SQL Server Connection here
  $uName=$_POST['text3'];
  $query6 = "Delete from UserAll where
  uName='$uName'";
  $result6= mssql_query($query6) or die ('Could not
  delete data');
  mssql_close();

  ?>
```

Code 3. Delete Row Button

```
  <?php

     //Microsoft SQL Server Connection here

      $uId=$_POST['text2'];
      $uName=$_POST['text3'];
      $address=$_POST['text4'];
      $contactNo=$_POST['text5'];
      $officeEmail=$_POST['text6'];
      $emailCodeNo=$_POST['text7'];
      $invidbackupcode=$_POST['text8'];

      $query7 = "Update UserAll Set
      uName='$uName',address='$address',
      contactNo='$contactNo',
      officeEmail='$officeEmail', emailCodeNo =
      '$emailCodeNo',  invidbackupcode=
      '$invidbackupcode' where uId='$uId'";

      $result7 = mssql_query($query7) or die
      ('Could  not update data');
      mssql_close();
   ?>
```

Code 4. Edit Row Button

```
 <?php

       // Microsoft SQL Server Connection here

       $query = "SELECT * FROM UserAll ORDER BY
       uId";

        $result =mssql_query($query);

     ?>
```

Code 5. View Table Button

The code for Next Row button in JavaScript in a file index.php can be shown as follows:

```
  var curRow=0;
  function clickNextButton(form)
  {   if(curRow==<?=$rows?>-1)
            alert("End of records");
       else
            curRow++;
      if((curRow>0) && (curRow <= <?=$rows?>)) {

  document.form1.action="dbpage2.php?currentRow
="+curRow;
       }
      else
         curRow--;
}
```

Code 6. Next Row Button

If the value of curRow is greater than zero and less than the number of database records, the JavaScript variable curRow is passed back to dbpage2.php on the server by sending it as part of a request otherwise curRow is decremented by one. Similar code and logic applies to Previous Row button.

VIII. Conclusion and Future Outlook

The AMMS system interface to an SQL IT-database consists of data entry forms and reports. These were created to manage IT equipment available at InfoCom GmbH efficiently with good overviews, good updates and low maintenance costs. While this has been achieved on the overall, there is still scope for future work. A feature that can be included in data-entry forms is having the history of the dates of changes when a particular input user would make changes to the actual database via data-entry forms. This will aid in tracking which input user edited the data in the database via data–entry forms on which dates. Another feature such as retrieving a record of the UserAll table in the textboxes of the data-entry form corresponding to a user name as input on the form can be included. It has been designed that the deletion of one record with a primary key in one table will cause the tuples with foreign keys in other tables to be deleted as well. However for a special case, it has been designed that when a user info is deleted, the room info and office key info corresponding to the user will still exist, making the user Id attribute in Room and Office_key tables null. For future scope, it can be designed that when user info is deleted, the software info on his/her hardware will still exist. This suggested future development will make the task of input user's data entry via data-entry form much easier and prevent her from inserting existing information again.

Although InfoCom had the author develop the AMMS system on a rather cheap level, the system is going to face maintenance costs in future which include the technological maintenance costs for example, upgrades of software tools such as, PHP, upgrades of operating systems on the machines or upgrades of client browsers as well as the internal costs which may be an increase in the number of people from about 35 users (at the time of writing the thesis) to 100 users or more.

Acknowledgment

The author thanks manager and supervisor Mr. Thomas Franz (InfoCom GmbH, Stuttgart, Germany) for allowing her to work on AMMS project in InfoCom GmbH. She is also grateful to her other supervisors Prof. Dorothee Koch (Stuttgart University of Applied Sciences) and Mr. Pawel Wojnar (InfoCom GmbH) for giving valuable guidance and advice during the work.

References

[1] Converse software (2004), Service Level Management....within reach, A Converse Software White paper, www.conversesoftware.com, Accessed August 2004

[2] GP Solutions Inc (2004), GPMate Software , www.gpsonline.com, Accessed August 2004

[3] J.Kreutzkaamp, L. Hagge, E. Deffur, A. Gellrich, B. Schulz (2003), Experience with an IT Asset Management System (pdf), Electronic Report, Desy, Hamburg, Accessed August 2004

[4] Nätverk Communications (2003), Inventory this (pdf), IDG International Data Group, Electronic Report, Accessed August 2004

[5] Luca D.(1999) , Java-Based Mobile Asset Location (pdf) , IEEE Monet, Electronic Report, Accessed August 2004

[6] ManageSoft Corporation (2004), ManageSoft, The Software management Company, www.managesoft.com, Accessed August 2004

[7] Syslist.com Company (2004), Syslist Software, A Hardware and Software Inventory Management, http://www.syslist.com/, Accessed August 2004

[8] Eden Company (2004), Asset Manager, http://www.eden.com/products/assetmanager/assetmanager.php, Accessed, Sep 2004

[9] MRO Software Inc (2004), MAXIMO® MainControl® for IT Asset Management (pdf), Maximo MainControl brochure, Accessed August 2004

[10] Code Punk's Web Tutorial (2004), Cookies in JavaScript, Aaronz WebWorkz, http://codepunk.hardwar.org.uk/ajs26.htm, Accessed October 2004

[11] Webmaster World.com (2004), JavaScript and PHP, http://www.webmasterworld.com/forum91/2591.htm, Accessed Dec 2004

Author's Profile

The author of the paper titled "The Asset and Maintenance Management System (AMMS) Project" is currently working as an Assistant Professor in Ahsanullah University of Science and Technology, Dhaka, Bangladesh. Her medium of instruction has always been in English. She obtained her masters in Software Technology from Stuttgart University of Applied Sciences, Germany. She also completed PhD coursework in Computer Science from the University of Texas at Arlington, USA.