# CHAPTER 5: IMPLEMENTATION PHASE

## *5.0 Introduction*

The implementation stage will involve the installation of software, testing, maintenance and training of users by the developer. The system is then tested to discover any errors so that corrections can be made before it can be installed. Testing process is repeated several times until all identified errors have been corrected.

The system will run parallel to the existing system for two weeks so as not to loose any information should the system fail for any reason. One disadvantage of this method of implementation is that there is duplication of effort imposed upon the user department as it is likely that the same persons have to run both the system during the periods of parallel running.

## *5.1* PROGRAM CODE SEGMENTS

Code for login, making changes, adding birth certificate

<?php

include "header.php";

?>

<div id="main\_content">

<div id="left\_content" style="width: 400px">

<h2>Login</h2>

<p>

</p>

<form action="login.php" method="post" name="sub" id="sub" enctype="multipart/form-data">

<table border="0" align="center">

<tr><td> Email :-</td><td>:&nbsp;</td><td><input required type="text" name="user" /></td></tr><br>

<tr><td> Password :-</td><td>:&nbsp;</td><td><input required type="password" name="pass" /></td></tr>

<tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" name="sub" value="Login" /></td><td></td><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td></tr>

</table>

</form>

</div><!--end of left content-->

<div id="right\_content" style="width: 400px">

<h2>Hints To Register</h2>

<?php

include "how.php";

?>

<div class="products\_box">

</div>

</div>

<?php

include "infor.php";

?>

</div><!--end of main content-->

<?php

include "footer.php";

?>

<?php

// Inialize session

session\_start();

// Include database connection settings

include('rb.php');

$db=R::setup('mysql:host=localhost;dbname=portal', 'root', '');

$\_SESSION['last\_login\_timestamp'] = time();

$username = $\_POST['user'];

$password =md5($\_POST['pass']);

$activity = "Log in";

$Time = time();

/\*

---------------------------- ---------------------------------------------\*/

$init = R::findOne('users', 'email = ? AND password = ?', [$username, $password]);

if ($init == null) {

// $message = "invalid details";

print ("<script>window.alert('invalid details')</script>");

print ("<script>window.location.assign('index.php')</script>");

} else {

@session\_start();

$\_SESSION['email'] = $\_POST['user'];

$\_SESSION['name'] = $init->name;

$\_SESSION['user'] = $init->username;

$\_SESSION['grade'] = $init->grade;

$\_SESSION['place'] = $init->place;

$\_SESSION['role'] = $init->grade;

$\_SESSION['id'] = $init->id;

$act = $activity;

$Time = time();

echo '<div class="alert alert-success" role="alert" style="background-color:transparent">...<h2 style="color:white">

<img src="images/success.png"/>

!login successfull </h2></div>';

if($\_SESSION['role']=="admin"){

echo ' <h2 align="center">

<meta content="2;admin/dashboard.php" http-equiv="refresh" />

</h2> ';

}else{

echo ' <h2 align="center">

<meta content="2;bentry.php" http-equiv="refresh" />

</h2> ';

}

// print ("<script>window.location.assign('index.php')</script");

}

?>

<?php

include'conf.php';

// Inialize session

session\_start();

// Check, if username session is NOT set then this page will jump to login page

if (!isset($\_SESSION['user'])) {

header('Location: timeout.php');

}

$user\_id=$\_SESSION['id'];

$invp=$\_SESSION['user'];

$inv="select \* from users where username='$invp'";

$ansinv=mysqli\_query($con,$inv);

while($row=mysqli\_fetch\_array($ansinv))

{

$invoicepre = $row['ccode'];

}

$ino="select id from birthreg";

$ansin=mysqli\_query($con,$ino);

while($row=mysqli\_fetch\_array($ansin))

{

$no = $row['id'];

}

@$sql2 ="SELECT id FROM birthreg WHERE id > '$id' ORDER by id DESC LIMIT 1";

$resultn = mysqli\_query($con,$sql2);

$nextrows = mysqli\_num\_rows($resultn);

while ($nextrow = mysqli\_fetch\_array($resultn)) {

$next=$nextrow['id']+1;

}

$d=date('Y');

$m=date('M');

$no=$ino;

$a=$invoicepre .'/'.$d.'/'.$m.'/'.$next;

$status=false;

$filetmp= $\_FILES['image']['tmp\_name'];

$filename = $\_FILES['image']['name'];

$filetype = $\_FILES['image']['tmp\_name'];

$filepath= "documents/bcr/".$filename;

move\_uploaded\_file( $filetmp,$filepath);

$date= date('d-M-Y');

$sql="INSERT INTO birthreg (Registerno, districit, zone, name, sex, dob, pob, father, mother, p\_address, a\_t\_birth, remark,bcr,users\_id,status,mid,fid, dor)

VALUES('$a','$\_POST[dist]','$\_POST[zone]','$\_POST[chname]','$\_POST[sex]','$\_POST[bday]','$\_POST[placebirth]','$\_POST[father]','$\_POST[mother]','$\_POST[paddress]','$\_POST[aaddress]','$\_POST[remarks]','$filename','$user\_id','$status','$\_POST[mid]','$\_POST[fid]','$date')";

echo @$id;

if (!mysqli\_query($con,$sql))

{

die('Error: ' . mysqli\_error());

}

echo"<script language='javascript'>

alert(' Birth Registration Sucessfully submitted.......! ' )

window.location = 'bentry.php';

</script>";

//echo $a;

//

//echo "<b>Invoice Created Sucessfully........! </b><br><br>";?>

<?php //echo"<a href='makebill.php'>Back</a>";

mysqli\_close($con)

?>

</center>

**<?php  
include** "header.php";  
**?>** <div id="main\_content">  
 <div id="left\_content">  
 <h2>Welcome to Online Birth Registration</h2>  
  
 <p class="clear">  
 <h3>Mission</h3>  
 <img src="images/pic1.jpg" alt="" title="" class="left\_img">  
  
 To deliver efficient service in the processing of births and deaths, registration of brands,  
 marriages, citizenship, national identity documents, designing, manufacturing,  
 processing and issuance of travel documents.  
 </p>   
<!-- <div class="read\_more\_link"><a href="documents\Birth certificate.docx">read more</a>-->  
 </div>  
   
   
   
 </div><!--end of left content-->  
  
  
  
 <div id="right\_content" style="background-color: white">  
 <h2>Department of the Registrar General. </h2>  
 <div class="products\_box">  
<!-- <img src="images/bg.jpg" style="height: 200px">-->  
 <p>  
 Welcome to the official website of the Department of the Registrar General.  
 </p>  
 <p>  
 This website provides you with timely and accurate information about the Department.  
 <br>  
 <br>  
 Our main objective is to effectively serve the people of Zimbabwe: Security is paramount.  
 </p>  
 <p><b>Core Values</b></p>  
 <p>In carrying out its mandate the department is guided by the following core values:  
  
 Integrity, Transparency, Professionalism, Impartiality and Timeless.</p>  
<!-- <img src="images/box\_icon.gif" alt="" title="" class="box\_img">-->  
<!-- <h3>Login here</h3>-->  
<!-- <p> -->  
<!-- <form action="login.php" method="post" name="myForm" onSubmit="return validateForm()">-->  
<!--<center>-->  
<!--Email :- <input type="text" name="user" /><br /><br />-->  
<!--Password :- <input type="password" name="pass" /><br /><br />-->  
<!--<input type="submit" name="sub" value="Login" />-->  
<!--</center>-->  
<!--</form>-->  
<!-- </p>-->  
 <br />  
 <br />  
 <br />  
 </div>  
  
 </div>  
  
</div><!--end of right content-->  
  
  
   
 <div style=" clear:both;"></div>  
 </div><!--end of main content-->

## *5.2 Coding*

This applies to the actual coding of the modules and the database. The code is likely to be more that 50 pages as this code will be for each text box, label or command button. Code will also be put to activate certain queries. The code being used is the Visual Basic Version 6.0 with Microsoft Access Database. See Appendix for Sample Code

A pseudo code was first developed to represent the general functionality of the system. It briefly describes how a user can navigate in the system starting from log in into the different modules found in the system.

**Pseudocode**

* **Login**

Enter username, password and access level

If correct then

Go to system main menu

Else

Try again

End if

* **Creating a new record**

Validate all the information

If invalid input is entered

Report error

Else

Using the established connection, save details

* **Updating data in the database**

Get the key field

Check if record does exist

If not then

Report error

Else

Validate all the information

If some of the input is invalid

Report error

Else

Using the established connection, save record

* **Searching for a record**

Get the record ID

Using the established connection, retrieve data

If data has not been found then

Report error

Else

Display data

Allow the user to edit

End

* **Deleting records**

Get the record ID

Using the established connection, display record

If record not found then

Report error

Else

Display data

Allow user to delete data

End

* **Viewing records in the database**

Get the records from database

Using the established connection, display all records

End

* Preparing reports from database

Get data from database

Use the specified query to prepare the report

End

## *5.3 Testing*

Intensive testing was done in order to make sure that all components of the database were functioning as required. This involves numerous methods of testing so that the system could meet the user specifications. The diagram below shows the procedure in testing.

Unit Testing.

Module Testing.

Sub-System Testing.

System Testing.

Acceptance Testing.

### Figure 5.1 System Testing Procedures

### 5.3.1 Unit Testing

This test will focus on one unit of the program, a module or function, and the objective is to check whether each unit performs its task as specified. Unit tests are done in two scenarios:

#### Black Box Tests

This is concerned with the inputs and outputs of a unit only. In our case we tested the functionality of querying the database by selecting records that we wanted to view and we filtered by name such that only records of a certain child will be displayed. If all is selected as an option the all invoices in the database will be displayed.

#### 5.3.1.2 White Box Tests

This focuses on the inner working detail of a unit and it will help find errors not immediately identifiable by treating a unit as a black box.

### 5.3.2 Module Testing

This method of testing combines dependent components and testes them together. A collection of procedures and functions were tested together. A single component can be tested without other system modules. Usually module testing is done using the set objectives; we used objectives to measure the reliability and functionality of our modules. In this type of testing modules such as the childs’ records were tested for their dependency on billing modules. The idea behind was to check if the system was updating the accounts before the child left. Also to find out if the program would make subtractions in database once the child pay his or her bill. The system was capable of producing the systems objectives and therefore concluded to be working properly.

### 5.1.3 Subsystem Testing

This is also called link testing. It tests a collection of modules, which have been integrated into subsystems. It ensures job streams are correct. There is detection of interface mismatches and rigorous exercises of the interface between programs. The modules, which are linked together were tested, in this case the childs’ module was tested. When a child visits they are to be entered into database for updates. This was done and the system was a success in performing the function.

### 5.3.4 System Testing

This involved testing of the entire information system and it includes all processing situations. The sub modules tested above are integrated to make a system. These were tested to find if there were any errors, which would hinder the performance of the system as a whole. Live data was used and there was verification that the system worked correctly.

### 5.3.5 Acceptance Testing

This involves testing using end users of the system and some other programmers rather than the project programmer so as to reduce bias. A few friends validated the system to check the following characteristics:

* System security
* Reliability
* Accuracy
* Processing speed

Also the user acceptance testing is done to determine if the system is complete, whether it meets business needs and acceptable to the user. It is composed of alpha and beta testing.

#### 5.3.5.1 Alpha testing

The system will be delivered to the organization that will use it and report any problems posed by the system and detect errors. If errors are present they are corrected and further release of system is carried out until it meets al functional requirements.

In our case the Acceptance testing was done in the presence of the Provincial Registra and end users of Marondera Registra. The system was thoroughly reviewed and function areas that were not working properly were corrected.

#### 5.3.5.2 Beta testing

The system is tested using real data supplied by the Administrative users and the administrator who would be using the system will also input their data. Errors and omissions in system requirements can be discovered and this process continues until the organisation accepts that the delivered system is ready for delivery.

### 5.3.2 Test strategies

We used a number of test strategies in trying to ensure the correct functionality of programs and the system as a whole. We managed to identify syntax errors and logic errors through the use of code reviews and structured walkthroughs. Some of the test strategies used is below.

#### 5.3.3.1 Validation

This is whereby we evaluated the system to check whether we had built the right system. Various objectives of the system were looked upon to see if they had been achieved. For example if we enter the numeric text in the field for alpha text then an error will appear for example upon pressing a key which is not a letter in the surname text box an error message will appear as shown by the example below, also if you enter alpha text in numeric field such as phone number an error will appear. We concluded that the system is a true representation of the system development project. This process was to ensure that the system would be delivered without errors.

**Figure 5.2 validation screen shot**

#### 5.3.3.2 Verification

This was also intensively done. This is whereby we looked at the system to see whether we had the developed the correct system. This means that the system was checked to see if it was meeting the customer specifications and requirements. The system might be excellent (running and fully functional) but not meeting the user requirements. Verification was also done intensively. We managed to check and analyze system representation using static techniques to check on requirements documents, design diagrams, program source code and inspections. We tested the system with some data and compared the results with already known results. We used the white box method of verification where the tests were conducted to ensure that the internal operation of the system performed according to specifications and all the internal components had been adequately exercised.

### 5.3.3 System Security Testing

Security is implemented through the use of physical and software measures. These include passwords, security user levels, burglar bars, tags and security guards. System security aims at protecting the system from vandalism, intentional negative incidences like fraud, theft and accidents. The objective is to avoid the occurrence of damage to the system or minimize the effects of an attack. These accidents can occur to hardware, software, data and the network.

We looked at how the risks may occur and then decided on ways which we can control and protect these risks. A threat to the system is any potential adverse occurrence that can do harm to the application or its data such as a computer virus or unexpected natural disaster or disruption. Disruptions occur when there is power failure or user mistakes causing the network to cut or cease functioning. Some disruptions may also be caused by destruction of data for example a virus may destroy files and other destruction may be catastrophic such as natural.

An example of one of the security measures in the system is the use of password, username and access level. If the user or any unauthorized user tries to login and if any of the text fields is wrong. The system will display an error message and prompt the user to log on again as illustrated by the example below.

**Figure 5.3 system security screen shot**

**5.3.3.6 Control of destruction, disruption and disaster.**

* Viruses can be prevented through the use of anti-virus packages such as esset node anti-virus and Microsoft security essentials Software which should be updated regularly. This checks disks and files to see if there are any viruses, which should always be updated.
* The system shall be in the server room, which shall be locked to be only accessed by the IT personnel (database administrator) with tags and their secret numbers that can swipe in after punching the number.
* Disasters must be controlled through having the application and its data stored in a separate location so that it can be retrieved on emergencies.
* Hardware can be insured.
* Having backup for data. Backup shall be done through the use of the Transmission tape (20-40 GB) in case the system crushes. Daily backup is done and disks can be run for particular days on which a system would have crushed. There are also weekly and monthly backups in order to safeguard un eventualities following a system’s crush. In addition backup shall also be done through the use of CDs on a weekly basis. The backup CDs together with the floppy disks are going to be kept in a strong room that is locked all the time and can only be accessed by the IT personnel. This strong room is located some distance away from the IT section and server room in case some disaster occurs in the IT section.
* Update method for Network interruptions.
* Uninterrupted Power Supply devices for any loss of power supply.
* There must also be a good restart function, which makes use of program status indicators so as to try and control errors caused by interruptions human made destruction.

#### 5.3.3.7 Software Security

The system uses passwords, security levels and contact numbers for tracking of transactions. Each user is assigned a password by the database administrator who has super rights. The modules and granules to view depend on the level of permissions assigned.

## *5.4 Installation*

Installation is when the developed system is being installed. Users are changing from using the old system to using the new system. In this phase the required software is installed on the appropriate hardware converting from the as-is system to the to-be system. Users are moved from using the old system to using the new system though in most cases the two systems (new and old system) may run parallel to each other.

A number of activities such as training, file conversion, system changeover methodologies such as direct changeover, parallel running and pilot operation are carried out during this stage.

### 5.4.1 User Training

We developed a training plan for users, which included all the entities in the running of the Integrated Systems. The training was mainly conducted to familiarize end users with the new system. We conducted a training that encompassed the management as well. Management should be also familiar with the system so that they are able to evaluate reports that are given to them for management evaluation purposes.

Training

g is done so that users are able to use the system. Training is not only for users, management also needs to be trained to be able to know the operations of the system in right time and to the right people. The system developers conduct the training until users are acquainted with this new system they would use.

We conducted training in groups with separate programs for distinct groups. The training was well structured as each module was demonstrated before proceeding to the next module. Questions were entertained from the trainees and test questions were also given l so that users become free to ask any queries they would not understand. We concluded our training by conducting a full-scale simulation for the staff for them to gain experience and confidence. At the end of the training session users were given forms with questions to fill so that we could evaluate the impact of the training.

### 5.4.2 Operation Environment

This is tested before the file conversion. Access was limited to users and we used live actual data and managed to verify all changes and we then obtained user approval. We examined all the system components that affect system performance such as:

* Hardware
* Software configuration
* Operating system programs
* Utilities
* Network resources

We found these to be functioning efficiently before we loaded and after we loading programs

### 5.4.3. File conversion

After the users have verified the results of the testing process the files are the transferred to the new system. Programs need to be designed which would do this conversion process. As the users would be using the system during working hours, the files will be in use, thus the conversion will be done during the weekends or outside working hours.

File conversion follows soon after operational environment has been established and training has been conducted. Under the normal schedule we had set, it took us three weeks to transfer manual data into the new system.

### 5.4.4 System Changeover

It is the technical process by which the new system replaces the old system. There are a number of approaches to system changeover namely:

#### 5.4.4.1 Direct Changeover

The old system ceases to function and is replaced immediately by the new system.

##### *Advantages of direct change over*

* Efficient method in so far as it minimizes duplication of work
* Less costs as only one system would be in operation

##### *Disadvantages of direct change over*

* New system may not be entirely correct
* It is difficult to make the system operational when some errors are identified after changeover and need correction
* Requires careful planning, testing and attention to operational detail

#### 5.4.4.2 Parallel Running

The new and old systems run in parallel for a short period then the old ceases operation after sometime.

##### *Advantages of parallel running*

* Low risk as a results can be verified and a back up option exists

##### *Disadvantages of parallel running*

* Relatively high costs as both systems are in operation at the same time for some time
* Method cannot be used for systems which are not similar

#### 5.4.4.3 Pilot Operation

The old and new systems operate at the same time but at a pilot site. The rest part of the organization continues using the old system.

##### *Advantages of pilot operation*

* Moderate costs as only a chosen site would be running two systems at once
* A moderate risk of failure as the new system is only installed at the pilot site.

The developed system is placed in actual site and tested before actual use to check if the system operates according to the set aims and objectives. System processes are checked to see if they perform according to the user requirements.

#### 5.4.4.4 Phased changeover

With this, the system is implemented in stages or modules across the organization. Phased changeover gives part of the system to the organization and cost is relatively moderate as the system is implemented in stages rather than all at once. Risk is also very moderate because the risk associated is limited to the module being implemented.

### 5.4.5 Post Implementation Evaluation or recommendations

We discovered that the new system was accurate and timeliness in the producing of output. It also yielded also a high level of user satisfaction. Generally, the new system was reliable and maintainable. It had also better system control and security. The users responded well to the training and they deduced that the new system was user friendly and had an effective database that added business value scheme.

### 5.4.6 Justification

For installation we chose the Parallel Running installation method because we found that though it may be relatively strenuous it was the most ideal method because of the following reasons:

* Risk is relatively low since back up (existing system) exists.
* Data is input into both systems and results obtained from both systems can be compared and verified.
* The users are subjected to gradual change that they can easily follow thus they are motivated and demoralized by abrupt change.

## *5.5 Maintenance*

Following implementation of the system, there comes a time when there is need for its

Review and this is normally done on monthly basis. It is carried out in this phase.

The system should be maintained to make sure that it still conforms to the specifications. Reviews should be done periodically and if the specifications or environment changes then the system should be upgraded. Maintenance of the system is an ongoing process.

### 5.5.1 Types of maintenance

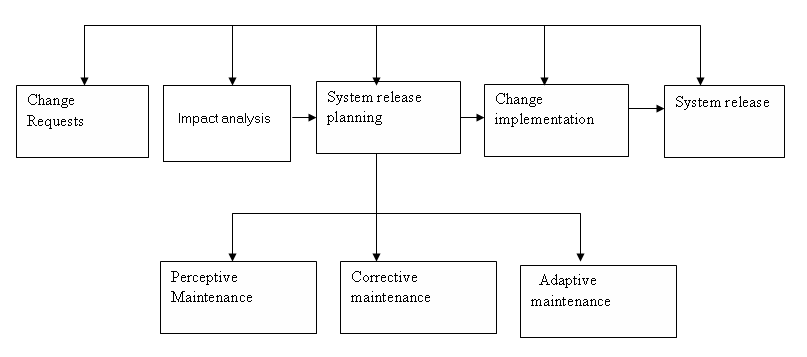
Maintenance is the process of monitoring, evaluating and modifying the operational information system to make necessary improvements. There are a number of approaches to the maintenance process. The need for maintenance will arise as a result of many different situations occurring throughout the life of the organization and these can include the following:

* If the organization grows, it means a change in the way it operates. The system will therefore require modifications to mirror the changes.
* The users of the system may require additional information that will help perform the system tasks better thus the need to make a few additions to meet such requirements.
* It is possible that initially, the requirements may have been incorrectly specified and when the system is now in operation, corrections will have to be made.

We implemented the system and examined it to see if it is meeting the objectives set out in the original specifications. The system analysts and the project team concluded that the system was meeting the specified objectives but since the environment changes so does the customer requirements.

### 5.5.2 Structured Maintenance

The maintenance process will be carried out as follows:



### Figure 5.4 Structured Maintenance

A change request form is submitted and analyzed. If the request is accepted a new release is planned. Release planning involves elements of adaptive, corrective and perfective maintenance.

There are three types of system maintenance are explained:

**5.5.2.1 Corrective maintenance**

This is the maintenance to correct the software faults. Coding errors ere usually relatively cheap to correct, design errors are more expensive as they may involve the rewriting of several program components. Requirements errors are the most expensive to report because of extensive system redesign which may be necessary.

#### 5.5.2.2 Adaptive maintenance

This is the maintenance to adopt the software to a different operating environment. This type of maintenance is required when some aspect of the system’s environment such as hardware, the platform operating system or other support software changes. The application system must be modified to adapt it to cope with these environment changes.

#### 5.5.2.4 Perfective maintenance

This is maintenance to add or modify the system’s functionality. This type of maintenance is necessary when the system requirements change in response to organizational or business change. The scale of the changes required to the software is often much greater than for other types of maintenance.

### 5.5.3 Unstructured Maintenance

**Change**

**request**

**Deliver**

**Repaired**

**System**

**Modify**

**Source**

**code**

**Analyse**

**Source**

**code**

**Figure 5.5 Unstructured maintenance diagram**

At times a system would need a quick repair for the business to run thus the need for this type of maintenance. This is where a request is given by users on an area where a change is requested and the developer analyses the source cone, make changes as per user request and deliver the modified version.

### 5.5.4 Recommendations for maintenance

The management was taught that maintenance is an important process to be given good attention if the system is to live longer while delivering goods.

The system is important because:

1. A system continues to changes and evolves as it is used.
2. The changes were arising from the change and evolve as it is used.
3. The changes were arising from change request due to the problems. Reports from operating groups who identifies the bugs in the system that must be fixed.
4. Changes were arising from users.

## *5.5.6 System Review*

This process ensures the newly implemented system meets the development objectives stated. Errors in the use of the system must be corrected. Periodic reviews are held with the users to audit the system. The purposes of review are as follows:

* To determine if specified objectives have been met and are still being met and if not, why?
* To ensure the users are using the system correctly.
* To check if the system is maintainable and flexible.
* To highlight shortfalls in the development process so that any mistakes can be rectified in future systems.

## *5.5.7 System Back-up Services*

The system should have a back up service in case it is corrupted or destroyed. There should be at least two CDs an external hard drive; one should be kept at the office to cater for corruption of system. The other should be kept away from the business premises in the case of fire destroying resource in the office; a backup would be available. Both CDs should be updated at regular intervals. In our case back up CDs are kept at the Head office and a backup is done on the server everyday and on weekly basis they are brought over on Friday and taken back on a Monday.

**5.5.8 User Manual**

The User Manual is the documentation that will give guidance to the System User on how to use the system. Most of the terminology and interface setup has been adopted from the basic structure and look of Microsoft Windows products. This was done so as to maintain the feel that is associated with using a product that you would be well versed with. Special training to the System Users will be provided to ensure that they become familiar with the new system and its operation as a system.

**See the appendix for the user manual**

## *5.5.10 Recommendations*

From what I have gathered throughout the Life cycle of the project, I recommend that the system be installed and begin to be used with immediate effect. I recommend that before using the system, users should consult the User Manual to familiarize themselves with the system. The users can simply use the system with minimal training because of the temptingly friendly user interface but if they do not know how to exactly go around the system, then the User Manual will be productive in information on how to fully utilize the system.

I would also recommend that the users maintain system security. That is, they should not disclose their password to any unauthorized personnel or let anyone temper around with the system.

I also recommend that the Marondera Registra be consistent with all levels of maintenance because it is an integral process as far as the performance of the system is concerned. Maintenance should be included in the core activities that the work-related-learning department has to cover in a stipulated period of time.

After the completion of all the phases, there was need to appraise the performance of the newly developed system and see whether it managed to meet all Objectives and Functional Requirements that were specified in the Proposal and Analysis phases respectively.

## *5.5.12 Limitations and Constraints*

Whilst all of the objectives were met, there were also some constraints and limitations in achieving these. There were some shortfalls that were encountered in the development of the system and also in the implementation of the objectives. Time was very scarce. I needed to have carried out enough research before developing the system whilst at the same time the user was on my back for results. Therefore, the system was built too quickly to meet the organizational needs disregarding what the market really needs.

Also the head office chose the programming language that was used and this meant that it was to be used. I also needed time to scale and evaluate programming languages because Visual Basic is not a flexible tool in report developing.

## *5.5.13 Performance in relation to Objectives*

The newly developed system works satisfactory so to say. It has managed to meet the following objectives as originally specified:

* Check stock before issuing out an invoice or receipt
* Distinguish between cash payment and credit payment and post to respective accounts or books
* Reduction in the flow of data
* Faster processing and retrieval of data.
* Enhanced security features and logon facilities.
* Run a number of reports as per management such as reorder level reports, interest reports, payment reports and cash book reports.
* Reduction in calculation errors.

## *5.5.14 Suggested Improvements*

After the completion of all the phases, there was need to appraise the performance of the newly development system and see whether the requirements and functional requirements that were specified in chapter one and chapter 3(proposal and analysis) were met.

5.5.15 Suggestions for further Improvements

Given that the system was successfully implemented and that parallel conversion is being used, there is room for further improvements and partial automation to the System. Below is what can be done to enhance the system.

* Since the database will be growing there will be need to move to an advanced Database Management system like Microsoft SQL Server from Microsoft Access 2000.

Although the project took some time to come up due to under budget, it was a successful one. Now it is being implemented and due to the changes in this harsh environment in Zimbabwe changes are continuously made make it adapt to the given environment.

**5.6 Conclusion**

The project was carried out smoothly and it was a success. The implementation was done according to what was really required by the user and the documentation will provide the necessary information if there be need to maintain the system.