



DEVELOPMENT OF AN ONLINE FAULT REPORTING/MANAGEMENT SYSTEM FOR POWER HOLDING COMPANY OF NIGERIA (PHCN), IKEJA ZONE

Iyidobi, J. C.¹, Mba, I. V.² & Opata, C. E.³

¹Department of Electrical and Electronics Engineering, Enugu State University of Science and Technology, Enugu, Nigeria

²Research Assistant, Swift Tech Consult Enugu, Nigeria

³Department of Electrical and Electronics Engineering, Enugu State University of Science and Technology, Enugu Nigeria

Abstract

The fault reporting system put in place by in Nigeria power distribution companies have been found to be ineffective in capturing fault reports from customers. This has contributed to the increase in outage duration in distribution networks. This project successfully developed an effective and efficient fault reporting and management system for a Nigeria electric power distribution network. The developed system allows distribution network customers to make online reports of electrical faults in their properties. The system will also allow distribution networks' fault monitoring teams to report faults in their facilities, online. Distribution Companies will also use data captured by this system to manage fault rectification and plan against fault re-occurrence. The developed system will also educate users on safety issues and how best to handle distribution facilities to reduce fault re-occurrence. Though the project was successful, further works could still be done on the project to make it even more effective and efficient in fault reporting and management. The incorporation of an automation system that sends e-mail alerts to users when relevant updates is made by the system administrator and the introduction of security features to protect the system and its users from external and internal attacks are recommended future works that will make this system much more robust and effective.

Keywords Online Fault Reporting; Management System; Power Holding Company of Nigeria (PHCN)

Citation Iyidobi, J. C., Mba, I. V. and Opata, C. E. (2023). Development of an Online Fault Reporting/Management System for Power Holding Company of Nigeria (PHCN), Ikeja Zone. *American Journal of Applied Sciences and Engineering*, 4(2) 32-48. <https://doi.org/10.5281/zenodo.15126253>



Introduction

Certain products and services require intermittent maintenance by the service providers if consumers are to enjoy utmost satisfaction during use. Faults or issues develop along the line in the delivery of these services to consumers. For providers to respond swiftly to these faults or issues, these faults need to be reported to the service providers using an effective reporting system. With the wide use of the internet in our society today, one of the most effective means of reporting faults to service providers is by the use of an online fault reporting/management system. Such a system allows consumers of services to make fault reports and check progress of fixing the faults, online. Effective fault reporting systems reduces the time it takes providers to respond to faults on consumers' systems and therefore reduces down time and improves the systems' reliability.

Electricity is one utility that is widely used all over the world. Both domestically and industrially, electric power has become indispensable in our modern world considering that we need electricity to power electrical and electronic gadgets, most of which we depend on to make life worthwhile. Because virtually every household and business establishment consume of electricity, there is tremendous pressure on most electrical networks and facilities that is employed in the supply and distribution of electricity. Like explained earlier electric power distribution networks and facilities are heavily under pressure due to its high customer base. As a result, it is common for faults to develop in the power networks and facilities not only because of limited supply of these facilities but because of the numerous customers depending on these limited facilities. For these faults to be promptly fixed by power company operators, these faults need to be reported. For the power operators to be able to manage the fixing of these faults promptly and plan against frequent re-occurrence of reported faults, faults have to be reported in a systematic and organized way that will present the fault reports to power supply providers in a well detailed and documented form.

The fault reporting system currently in use in electric power distribution networks in Nigeria is ineffective, inefficient and not fully functional due to some technical issues in the design. The inability of electricity customers to report electrical faults through the system has continued to keep outage duration very high with power system's reliability adversely affected. The current system also lacks some necessary features that will help the fault reporting system to progressively build up a fault database that will help electric power distribution networks in future system planning and fault management. The existing electric power distribution network fault reporting system in Nigeria did not make provision for educating customers on best practices in handling power network facilities, safety measures and how best to prevent frequent fault re-occurrence. This project seeks to develop a new system that will overcome the above shortcomings identified in the existing system. One of the most effective and viable ways of improving the existing system is by developing an online fault reporting system that incorporates a database to capture and store the fault reports.

This project is targeted at developing an online fault reporting/management system for the electric power distribution networks in Nigeria. This is in addition to allowing the power company's customers log in fault reports and check status of its rectification.

The proposed system will enable electricity consumers to make online reports of electrical faults observed in power installed facilities/network in their immediate location. Also, power network fault patrol team will use this system to login faults observed in power equipment located along the power lines and in the power stations. The fault data will immediately be used by power system operators to manage the fixing of these faults promptly and in the long run utilise the fault data built up over a long period to make forecast and effectively plan against frequent re-occurrence of faults.

The Ikeja Zone of Power Holding company of Nigeria (PHCN) as constituted as of 2011 was selected as a case study and as such this system is built to serve users in Ikeja Zone only. Also due to insufficient time, this project will concentrate on the design and implementation of the web (Front and server end) section of this system. The research paper will not go into the details of the design and implementation of the database component of this system. The research will not also delve into features that will secure the system from attacks, but will focus more on the requirements specified.

Methodologies

The methodology proposed for use in this project is the incremental model. This model as observed by Deming (2006) is a flexible approach that sees an artefact developed build by build (or stage by stage) to ensure that the end product satisfies the client (users). As much as possible, effort will be made to gather all the user requirements in the beginning. After each build, testing and evaluation will be carried out and possible errors detected and fixed. After completing the final build that will be delivered to the client general testing of the system will be done to ensure correctness, good performance, reliability and usability. Due to its flexibility of this methodology, it is easier to accommodate new requirements during development (if need be), unlike in the waterfall model. This is a good strategy since sometimes users are not able to describe accurately what they want. However, efforts will be made to ensure that all user requirements are gathered as accurate as possible to limit changes as much as possible. Development of the artefact build by build until a perfect product is delivered in the end helps to carry the user along and factor his inputs in the following build. Carrying out testing at the end of each stage (segment) of development will ensure that the end product will be a working piece and above all meets all the user requirements.

Key merits of the incremental model as observed by Bhattacharjee (2011) includes lower risk effect of additional or modified requirements; initial speedy completion of product, reduced cost of initial project delivery, early incorporation of vital functionalities for consumers, increase in user participation in the development achieved through their feedback on the initial build and a progressive addition of functionalities in subsequent builds that leads to a near perfect implementation. Some of the challenges usually posed by this model as noted by Bhattacharjee (2011) that the model demands that at the early stage of development, developers need to define all basic requirements that will make the system fully functional and be able to accommodate increments.

This methodology is suitable for this project considering that it will allow for the accommodation of new requirements with minimum risk. This is very important considering that the users (electric power customers and staff) do not have an in-depth knowledge about software development and are therefore likely not to accurately specify the requirements. Also considering time constraint in this project, this model allows us to produce a complete functional system with the limited time while creating room for improvements in subsequent builds. These challenges observed will be tackled to ensure that they do not constitute any form of setback to the success of the project.

Interface Design and Blue Print

Just like the back-end (database) of this project, the web interface is also a very important aspect of this project and therefore time is taken here to ensure that a good and effective design and blue print is produced. The essence of the web interface is to capture inputs from the users, transmit it to the database and extract and display information from the database to the user when the need arises. In this design, HTML codes will be used to style the forms that will capture users' inputs while Java server pages (JSP) codes will be used to transmit or retrieve data from the database.

The Home Page

The home page is the first page that will greet the user. It will be designed to be informative and inviting so as to encourage the user to use the system. As can be seen from the home page blue print shown in figure 3.1, we see that the page is static and will bear the company's full names and logo. The page will also bear links to other important pages including login, registration, company's contact, and learning more (a page for displaying information on best practices as regards handling the case study company's facilities, safety, and strategies of reducing frequent fault occurrences). The Company's name, logo, contact address and these links will serve as a template that will be replicated for all other web pages that will be involved in this design. This is to ensure easy and flexible navigation from one page to the other. The home page will also contain information that will educate the users on the need to login fault report using the system. Pictures of some common facilities and rectification activity on going in the case study company will be displayed in the home page to depict what the system is doing. Also, the contact address and telephone number of the company will be displayed at the base of the page.

PHCN LOGO	POWER HOLDING COMPANY OF NIGERIA (PHCN) PLC IKEJA				
ZONE					
HOME	CONTACT US	REGISTER	CUSTOMER-LOGIN	STAFF-LOGIN	Learn More
FAULT REPORTING/MANAGEMENT SYSTEM					
Pictures depicting <u>phcn</u> facilities and <u>phcn</u> men at work		Educative information on the need to use this system as a means of fault reporting and how to use it		pictures depicting <u>phcn</u> facilities and <u>phcn</u> men at work	
PHCN office and postal address, e-mail					

Figure 1: Proposed home page for this design

Registration

To use the system, users need to login, and to login, they need to obtain login details (user ID and password). As pointed out earlier, like every other page, the registration page will retain the re-occurring template (Company name, logo navigation links and contact address). Most importantly, the registration page will have fields that will allow users to fill in their first name, surname, e-mail address and password. There will be an additional field for re-typing password to compel the user to confirm his password by re-typing exactly what he typed initially. The registration page will also bear brief instructions/guide on how to register successfully. All the fields in this page will be marked with (*) sign to make them compulsory fields that must be entered by users. This is to ensure that users enter complete registration information. Adequate error messages will be displayed if user fail to fill any field and the form will not be submitted until all fields are filled. There will also be a submission button and a reset button. HTML will be used to style the registration form will.

Between the HTML registration form and the database, there will be a JSP page that will capture the inputs from the forms and then pass them to the database. The connection between the web pages and the Mysql database is provided by Tomcat server. After a successful login, the JSP page will display a confirmation message to the user. The message will also direct the user to proceed to login.

Login

As we pointed out earlier during analysis of primary research, the essence of login in was to ensure that users' activities in the system are regulated so as to avoid any abuse. There will be two login pages: one for PHCN customers and the other for PHCN staff. This is to make authentication easy as the login details for the two set of users are held in separate tables in the database. The login form will have just two fields, one for e-mail address of user and the other for users' password. The password field will be made to hide its values as the user inputs them, for security reasons. Submission of form will not be successful until the two fields are filled. An error message will be displayed to the user if an attempt is made to submit the form with either or both of the fields are not filled. There will be a submission button for submitting the login details. There will also be a link that will ask new users without login details to register with the system. Two JSP pages (one for each of the two login forms) will be in place to capture the login details, compare them with the ones in the database and then either give the user access to the next step or deny him access, and request him to re-login. For a successful login, the JSP page will display a confirmation message and this message will bear a link to the user details form/page, fault details form and the track fault status form.

User Details

There will be two of these forms styled using HTML. One will serve the PHCN customers while the other will serve the PHCN staff. As earlier pointed out, the essence of this page is to allow users to provide their personal data that will help PHCN to locate them and the fault they are reporting. The forms contain the following fields corresponding to the customer table in the database: Customer ID (PHCN NO), first name, surname, e-mail address, Property address, Phone number and Business unit. To restrict users from filling in un-existing Business unit, users are required to select Business Unit from a pre-defined list provided. All fields will be marked with (*) sign. Therefore, filling all the fields will be made a condition for a successful submission of the details form and adequate error messages will be generated to prompt the user to enter any empty field. There will be brief guiding information contained in the page. A submission button will also be provided. Two JSP pages (one for each form) will be put in place to transmit the user's details to the appropriate table in the database. The JSP pages will also display a message to confirm a successful submission of details. The confirmation message will bear a link that will direct users to the Fault details form/Page.

Fault details

Like the user details form, fault details form will be two in number (one for customers and the other for staff) and will be styled using HTML. It is one of the most important forms in this system. This is because as discussed earlier, it will allow users to enter information about the faults they are reporting. This will give PHCN management an idea of what will be required to fix the fault in terms of man power and materials. The fields that make up this form include: Customer ID (PHCN NO), fault ID, fault date, fault time, fault description and fault status. Filling Customer ID (PHCN NO) and fault description will be made compulsory and as such they will be marked with the sign (*). Adequate error report will be generated to prompt the user to enter a compulsory field left blank on clicking the submit button. A submission button is provided for submitting the form after entering the fault details. Two JSP pages (one for each form) will be put in place to transmit the fault details to the right table in the database. The JSP pages will also display a message to confirm to the user that the values entered have been taken. A logout button will also be provided to allow the user logout his account.

Observe that the link to the user details form and fault details form are not accessible from the home page and the common template mentioned earlier. The user details, the fault details forms and the track fault status form are only accessible after a successful login. This is to ensure that only registered and validated users are able to report faults.

Track Fault

The track fault form will also be coded with HTML and will be two in number, one for customers and the other for staff. It will help users to track the progress of rectification of fault faults reported. The page can only be accessed after logging into the system. It will have only one field, customer ID (PHCN ID) and entering the field will be made compulsory. The forms will have their corresponding JSP pages that will capture and transmit the value entered by users for the purpose of retrieving from the database, the status of faults reported by the user. A submit button will be provided to send the data across.

The Learn more page

This page is a HTML static page and will contain information targeted at educating the user on the best practices as regards handling PHCN facilities in their domain, safety measures when fault occurs and proactive measures to take so as to avoid frequency of fault re-occurrence. The page will be constantly updated by the administrator when new information needs to be displayed. Like every other page in this application, this page will have the usual common template mentioned earlier.

Contact Us Page

This page will simply display in detail the company's contacts such as Office and postal address, contact phone numbers, emergency lines, e-mail addresses, etc. It will also bear the usual template so that users can flexibly navigate to other pages from there.

The web design blue print shown in figure 2 below shows all the development files that will be produced in this project and how they will work together to fulfill the requirements of the system.

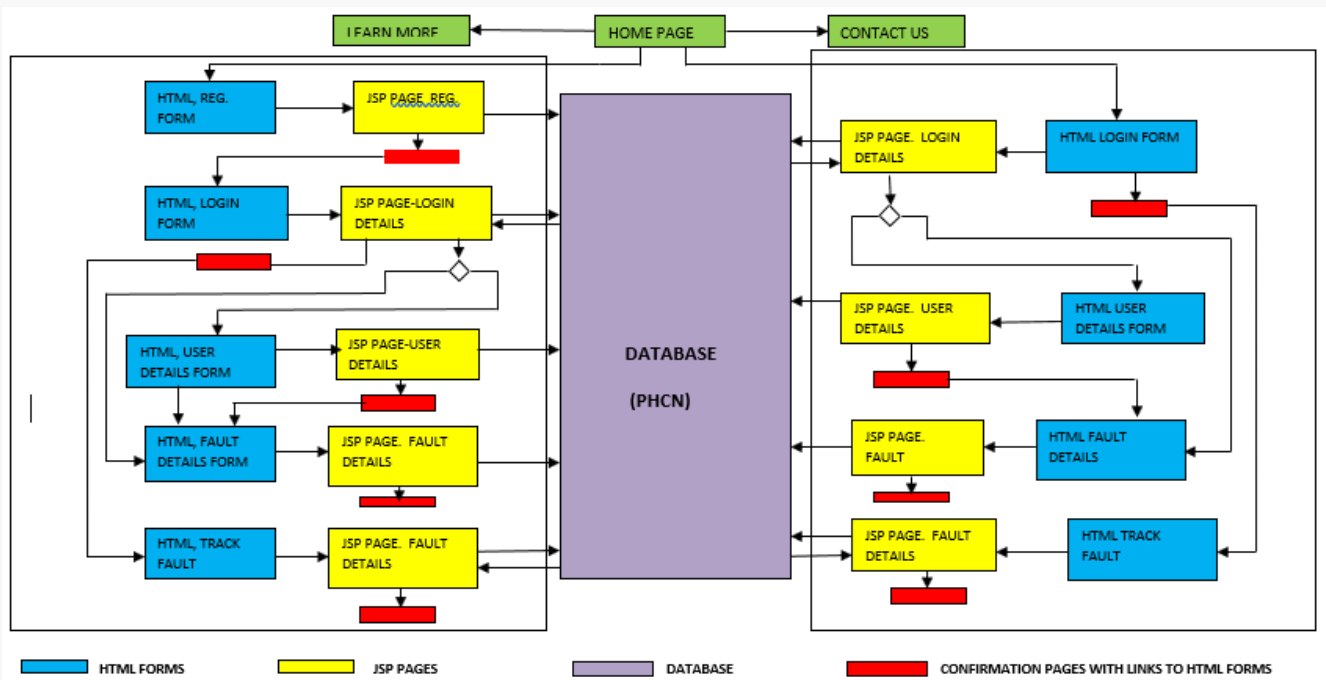


Figure 2: Comprehensive web interface blue print showing development files

Implementations

This section gives an overview of the basic technologies applied in this project and a detailed description of the steps taken to implement this project. Some of the major technologies that were employed in realising this project are discussed below. They include: Mysql, HTML, JSP and Tomcat.

MYSQL

According to Valade (2010) Mysql is a simple relational database management system used widely in web development. Valade noted that Mysql has the advantage of being very fast and cheap considering that it is an open-source package. Mysql apart from being secure has a unique flexibility that makes it compatible with most operating systems, added to the fact that a free technical support for users is available (Valade, 2010). Mysql database was used to implement the database of this project. Mysql commands relevant to this project are already very familiar to me having applied it in studying the module, internet development. Due to lack of space and the large volume of this paper, details of the design and implementation of the database used in this online fault reporting system was not captured in this paper.

JAVA SERVER PAGES (JSP)

According to McGrath (2002), Java Server Pages is a technique that makes it possible for web pages to be developed in a dynamic way from the web server such that depending on the situation, the content of the page can be changed. McGrath noted that JSP makes use of popular mark-up languages like XML and HTML. McGrath (2002) observed that that apart from having the ability to be run in any platform (operating system), JSP technique is very good at retrieving information from a database, applying Java based mechanisms such as JDBC (Java database connectivity) in linking to databases. The ability of JSP to be run on portable PC based servers like Tomcat makes testing easy during web development with JSP (McGrath, 2002). JSP is also very familiar to me having applied it in studying the module, internet development.

HTML

According to McGrath (2009), Hypertext Mark-up Language (HTML) of contemporary standard that employs tags to show a web browser how a web page should be displayed. HTML codes can easily be created in notepads or in edit plus environment. It is easy to learn and very flexible to use. Most of the static elements of this development were implemented using HTML. Ahmed (2010) noted that HTML is mainly concerned with how data is displayed and cares less about whether the displayed data makes sense. Ahmed (2010) added that only predetermined tags and attributes are employed in creating HTML files.

Tomcat

According to Ahmed (2010a), Tomcat is a servlet container that supports programming in the server side by making it possible for JSP to stay on the server and offer programming capability in both dynamic and static manner. Ahmed (2010a) noted also that high speed and sophisticated web site processing capability of dynamic type can be generated with a connection between Apache web servers and tomcat. This arrangement that is easily accessible in our CIS labs will be employed as a PC based server for easy testing while developing this application.

Web Interface Implementation

As pointed out earlier, the major aim of this project is to enable users of this system to login fault reports, online such that these reports are captured and stored in a PHCN database built for this system. As stated earlier, the web service utilised in this development was provided by Tomcat server which was easily accessible in our CIS labs. HTML was used to develop the client side of this application while JSP was used in the server-side programming. The use of this system requires the user to enter certain details at all stages. To capture these details from the user at the client side, HTML forms were used. HTML was also used to develop the home page which is the first page that greets the user. The home page and the HTML forms were first developed, this was followed by the development of the JSP pages and then the remaining two pages (contact us and learn more). In line with the adopted methodology, testing was done after each phase of the development to ensure that the first build is working.

The Home Page

The home page was named "home.html" and titled "HOME". The first block that bears the company's name was set to a light red background while the company's name was shown in white colour. PHCN logo which was copied from PHCN main website (2010) was positioned in the extreme left corner of this first block. The width of this block was set to 1400 while the height was set to 50. The next block bears the navigation links. The background colour of this block was set to blue while the links were shown in white to make them very visible. Note that the colours blue, light red and white was used in this implementation to match the colours in the PHCN logo. The width of this block is also 1400 while the height was set to 35. In the main body of the page, there are four pictures showing PHCN facilities and repair works. The pictures were copied from PHCN website (2010). Two of the pictures were positioned at the right-hand side while the other two were placed at the left-hand side. At centre of the page (between the pictures), information on the need to use the system and how to use the system was posted. The words were left in black colour. The background of the main body of the page was set to light ash. This colour is neither too shouting nor too dull but was just moderate in bringing out features placed on it. Brief contact information of the company was positioned at the base of the page. To produce this page; the <table> tag, <body>, <td>, <tr>, <center>, , <div>, <p>, <a>, etc, were extensively used to implement this page. "HREF" were used to point the navigating links to the corresponding files. Apart from the pictures and the information displayed at and the center of the page, all other features in the home page were retained in subsequent pages. This therefore served as a starting template from where the static elements of all the other pages were developed.

Registration Form

The first html form to be developed was the registration form. It was named registration.html and its title was registration form. The html codes in the home page were copied to a fresh edit plus page since most of the features will be retained except the pictures and the message at the main body. The pictures and the information in the main body were first deleted. To create the form (bearing the field spaces for entering values) itself, a new tag <form> was introduced. It was named 'registrationform.html' and the action of the form in on 'register.jsp', the JSP file that will pass the data to the database. The method used is 'post'. Five field spaces were created for firstname, surname,

e-mail address, password and re-type password respectively. For the fields, text was selected as the type of data expected in the first three fields while password was selected in the last two fields (password and re-type password). This is to ensure that the password being typed is hidden for security reasons. The size and maximum length of text permitted were selected to be proportional and suitable for the expected data. A submit and reset button was also created as part of the form.

The next step taken was to validate the form to ensure that users enter data in all required fields before being able to submit the form. The validation will also prompt users to enter required data which they have not entered. For the registration page, all fields are required fields so validation was such that all fields must be entered for submission to be successful.

The Java script used by Ahmed (2010) in internet development lecture material was adapted and used for all the form validation in this project. In the registration form, the java script has been coded to prevent submission of an empty form and to progressively alert the user to enter the required details starting from the firstname till the all fields are entered. For instance, if the user attempts to submit an empty form, he will be alerted to fill in the first required field (firstname). He will progressively be alerted to fill in the next field in line or any one randomly left unfilled during submission. The java script was also coded to prevent the submission of the form until the fields password and re-type password are the same. This ensures that users confirm their initial password entered. To ensure that user enter predetermined Business units, the <select> tag was used to specify the available list of Business Units from where users can select from. All the required fields were marked with (*) sign to alert the user that these fields must be entered before the form can be submitted.

All other forms (login, user details, fault details and track fault) have the same structure. The only difference between them is in mainly in the field names, number of fields, the JSP the action will be directed to, and the brief guiding information provided for users. The java script also has the same structure in all the form. In view of the similarities in the structure of the forms, the same process described above (for the registration form) was used to develop all the forms. However, For the sake of clarity, things that are peculiar to each of the forms are highlighted below.

Login Form

There are two login forms; one for PHCN Customers and the other for PHCN staff. The login form Customers was named 'cloginform.html' and titled 'customer login form' while the one for Staff was named 'sloginform.html' and titled 'staff login form'. The forms capture the login details of users. For each form, there are just two fields: e-mail address and password. The two fields were marked (*) as they must be entered to submit the form. As stated above, java script similar to the one used in the registration form was adapted for these forms. The action of cloginform.html is on 'clogin.jsp' while the action of sloginform.html is on slogin.jsp. Submit and reset buttons were also provided.

User Details Form

There are also two user details forms; one for Customers and the other for Staff. The user details form for Customers was named 'cdetailsform.html' and was titled 'cdetails form'. The one for Staff was named 'sdetailsform.html' and was titled 'sdetails form'. These forms allow the user to enter his personal details. The fields in these forms include: PHCN NO (customer ID), firstname, surname, e-mail address, property address, phone number and Business Unit. All fields were made required fields. As stated above, java script similar to the one used in the registration form was adapted for these forms. The action of cdetailsform.html is on 'cdetails.jsp' while the action of sdetailsform.html is on sdetails.jsp. Submit and reset buttons were also provided.

Fault Details Form

There are also two fault details forms; one for Customers and the other for Staff. The user details form for Customers was named 'cfaultform.html' and was titled 'report fault'. The one for Staff was named 'sfaultform.html' and was also titled 'report fault'. These forms allow the user to enter details of fault being reported. The fields in these forms include: PHCN NO (customer ID), fault ID, fault description, fault date, fault time and status. The fields for PHCN NO and fault description were made required fields. As stated above, java script similar to the one used in the registration form was adapted for these forms. The action of cfaultform.html is on 'cfault.jsp' while the action of sfaultform.html is on sfault.jsp. Submit and reset buttons were also provided.

Track Fault Form

There are also two track fault forms; one for Customers and the other for Staff. The user details form for Customers was named 'ctrackform.html' and was titled 'track fault'. The one for Staff was named 'strackform.html' and was also titled 'track fault'. These forms allow the users to check the status of the faults they reported. There is only one field in each for, the PHCN NO (customer ID). This field was made a required field. As stated above, java script similar to the one used in the registration form was adapted for these forms. The action of ctrackform.html is on 'ctrackfault.jsp' while the action of strackform.html is on strackfault.jsp. Submit and reset buttons were also provided.

The JSP Pages

With the help of the server, the JSP pages interact with the database either to insert values captured from the forms or to retrieve values from the database and display to the user. Only the JSP pages responsible for retrieving users' fault status will be displaying data from the database to the user. All the other JSP pages will only be inserting into the database, data captured from the form filled by the user. However, all the JSP pages will be displaying messages to confirm successful capturing of details entered by the user in the database. Also to be displayed along with these messages are the links that will take the user to the next html form to be filled. Html codes where was used to make the display of these messages and these links possible. The usual html template (bearing company's name, logo, navigation links and brief contact) copied from the home page formed the starting code for all JSP pages. This was to see that the information displayed by the JSP pages in the template described above. Apart from these html codes, each JSP page also contains some jsp codes that do the inserting or retrieving of data from the database depending on the function of that particular page.

Register.jsp

This page captures input from registration.html. It has the html template copied from the home page and the html codes for displaying confirmation messages and links to other forms. The jsp codes used by Ahmed (2010b) in internet development lecture material was adapted and used for all the JSP pages of this development. Before writing the jsp codes, all the required packages were first imported. Having imported all required packages, the request.getParameter() method was used to capture the values coming from the registration form and then storing them as string variables. In doing this, it was ensured that the string values contained in the argument of the method where exactly the same as the corresponding field names in the registration form. The captured values were then saved as string names corresponding to the column names in the database. After this, the JDBC driver was loaded and a connection to the database was established using the getConnection() method. The user's name and password for the machines in CIS 106 lab where this development was mainly done, is root and password respectively. These details together with 'phcn' (the name of the folder of this development) were used to connect to mysql database of the machine. Appropriate mysql query was (INSERT statement) was used alongside a prepared Statement to insert the values into the specified columns in the table in question. Having done this, all connections were then closed. Some html codes were then used to display a congratulatory message to the user after a successful registration.

Login.jsp

This page is similar to register.jsp in some respect. However, there is a major difference. While register.jsp inserts data into the database, login.jsp retrieves data from the database and compares them with the values captured from the login form. Apart from this key difference, most of the other elements are the same. To develop this page, register.jsp was copied to a new edit plus page and then modified. All the html codes were retained except the congratulatory message. Like in register.jsp, the values coming from the login form (e-mail address and password) were first captured and stored as string values. JDBC driver was loaded and connection to the database established. In order to send the Mysql statement (SELECT statement), the create Statement method was used to create a statement object. Inside a while loop, the pair of values (e-mail address and password already in the database) are retrieved and compared with the one captured from the login form to see if there is a match. This comparison is done for all the pairs of values in the database until a match is obtained. If there is a match, the user is allowed access else, the user is denied access and directed to re-login. A major issue was encountered at this stage of the development. On attempting to login with a valid e-mail address and password already inserted into the database,

the system two messages: one was a confirmation for a successful login and the other was denial of access and a request to re-login. See figure 3 below

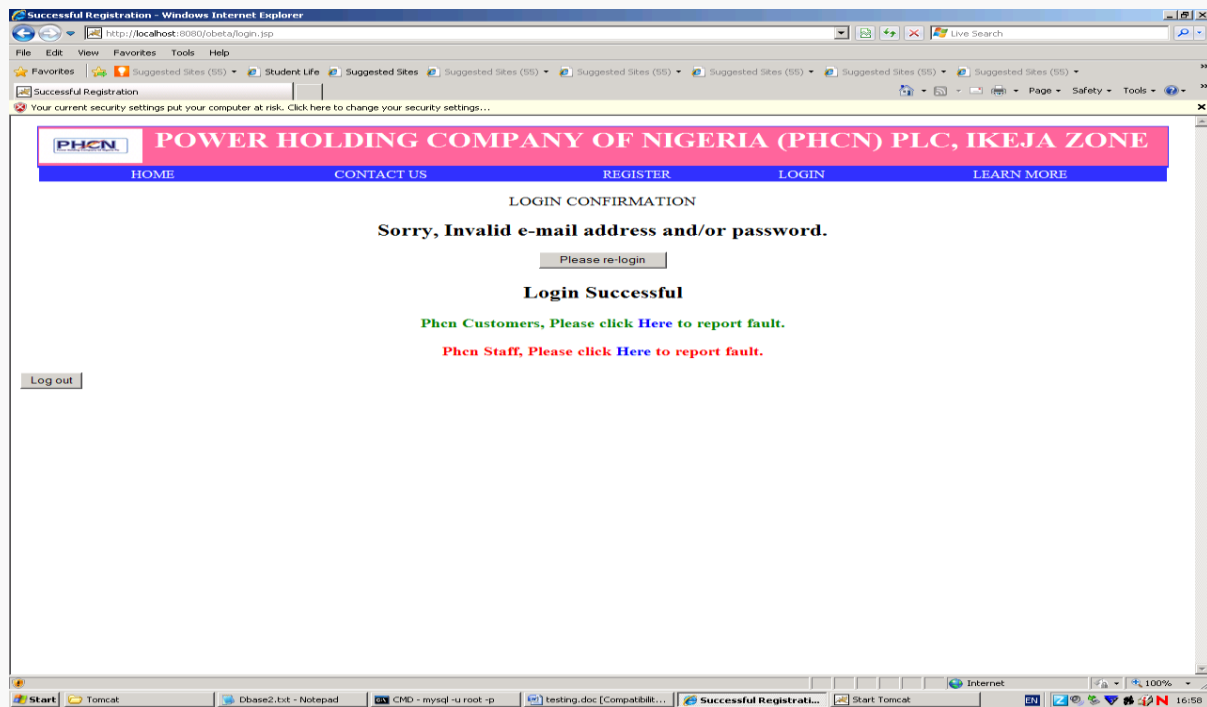
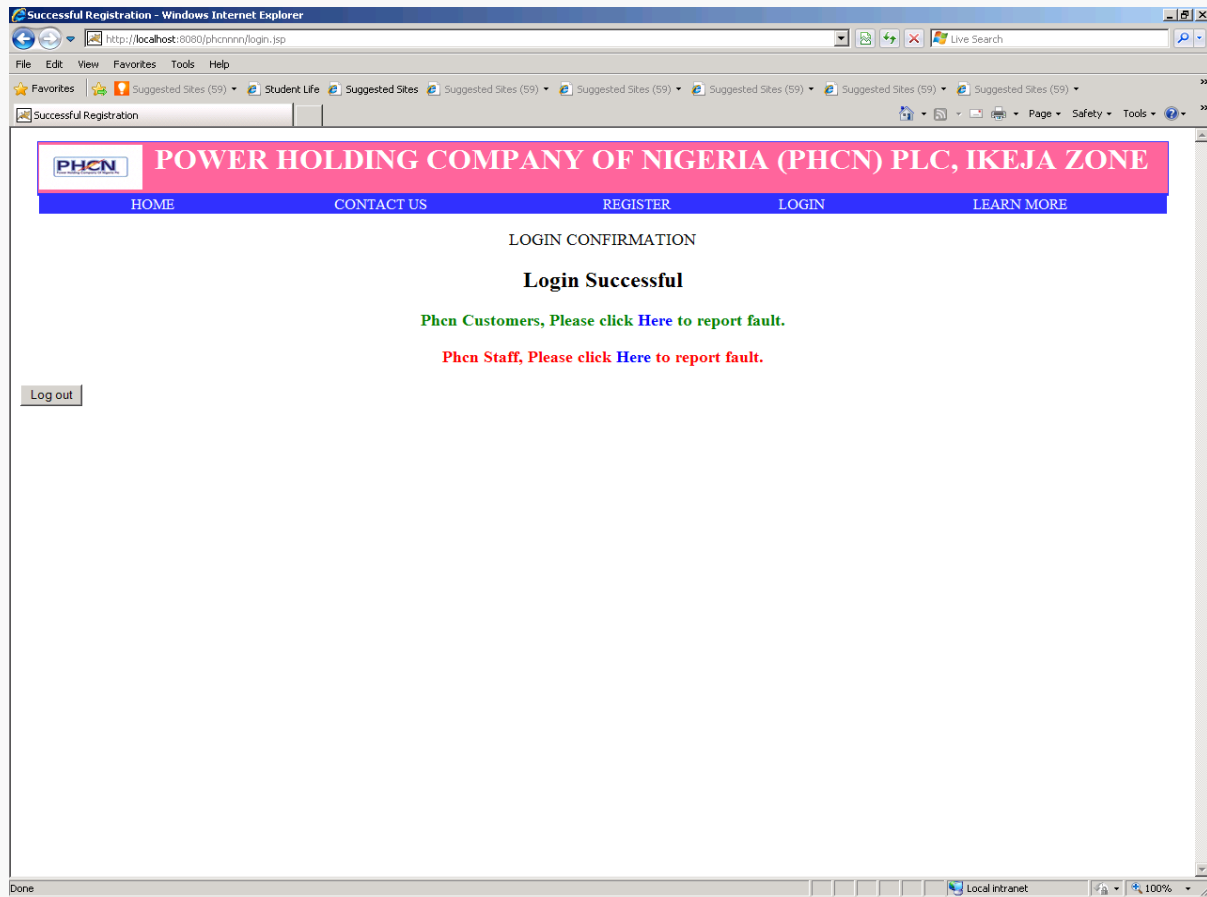


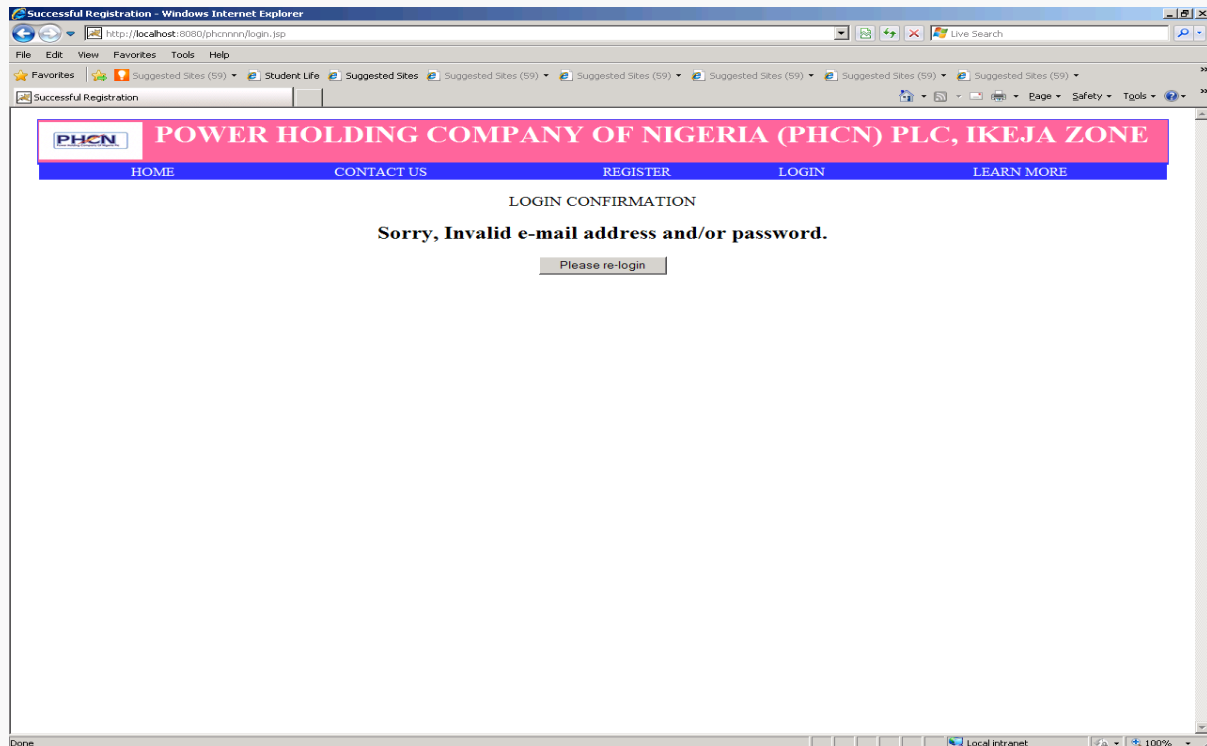
Figure 3: Screen short showing inconsistent login confirmation message.

At this stage, there was just two pair of values inserted in the database. When one more pair of values were inserted and another login trial was made, three sets of information was displayed. Two were denial of access corresponding to the two non-matching pair of values in the database and the other message was a confirmation for a successful login corresponding to the pair of matching values. It became clear that the system displays a message for every set of values compared in the database.

To resolve this, a break statement was introduced and the code restructured to avoid break the loop as soon as a matching value is found and to display only one confirmation message or an access denial message as the case may be. Figures 4 and 5 shows screen shorts of the system display after resolving the problem.



Figures 4: Screen shot of a confirmation page for successful login after problem has been resolved.



Figures 5: Screen shot of an access denial page for a non successful login after problem has been resolved.

It will be important to point out that these screen shots were taken when it was being considered to use only one user details form for both PHCN customers and PHCN staff. This idea was not adopted because data from the two set of users will be going into separate tables and writing the code to implement this will be difficult. It was therefore decided that a separate login form and user details form be created for each set of users. Also, two major amendments were done in the flow of event after this screen shot was take. One of the changes was that after logging in, the confirmation message should have a link that can take the user to trackform.html to enable him track faults already reported. The other change was that users will only be required to enter their personal details once (when reporting a fault for the first time) and as such should have access to the link, faultform.html subsequently without having to enter their details again. Requiring users to enter their details only once was done to prevent multiple submissions of personal details that will introduce undesirable redundancy in the system. More so, the database will reject such multiple submissions since the tables were duly normalised. To achieve this, the link to faultform.html was also provided in the login confirmation message. Html codes were used to display the confirmation and access denial information as required.

Cdetails.jsp and Sdetails.jsp

These two pages are similar in every respect. As noted earlier, the only difference is that cdetails.jsp captures data from cdetailsform.html and inserts it into the database while sdetails.jsp captures data from sdetailsform.html and inserts it into the database. This two pages functions exactly the same way register.jsp function. Their role is to capture data from the corresponding html form and then insert them into the database. Because of the similarity in functions, cdetails.jsp and sdetails.jsp were developed just the same way register.jsp was developed. The major difference was just in the data being captured and inserted into the database. The html code that displays the confirmation message for successful entering of details was also modified to ensure that the links in the message directs users to the right forms.

Ctrackfault.jsp and Strackfault.jsp

These two pages are also the same except that one tracks fault for Customers while the other tracks fault for Staff. Ctrackfault.jsp captures Customer's PHCN NO from ctrackform.html and then uses it to display status of faults corresponding to that PHCN NO in the database. Strackfault.jsp does exactly the same for staff. The development of this page was similar to the development of login.jsp. However, in this case, the Mysql statement is different. Here the Mysql query asked the database to display the fault status and some other columns from the fault table for all the rows corresponding to the customer ID (PHCN NO) captured from the trackform.html. Html codes were then used to style how the data retrieved will be displayed in form of a table. It was not very easy to get the jsp codes in this page running. Some errors arising from inability to compile the jsp was encountered. However, since the error messages usually suggest which part of the code is having issues, it was possible to track the problem which most time were either omissions or wrongly typed commands.

Learn more and Contact us Pages

These two pages are not executable files. They are static pages just like the home page. To create them, the html codes from the home page were copied. The four pictures and the message in the main body of the page were removed. For the Learn more page, information for educating the user was posted in the main body using additional html codes. In the case of Contact us page, detailed contact information for the company was posted in the main body using appropriate html codes.

Testing and Evaluation

Software Testing

DeMillo et al (1987) views software testing as the validation of software by comparing its functionalities with the functions it was intended to perform. Pezze and Young (2008) added that the aim of testing in a software development is basically to detect faults so as to eliminate them before releasing the software. In line with the methodology of this project, test was carried out in the course of implementation of this project at various stages. Some of these tests have already been highlighted in chapter four. A general testing is also carried out before delivering the product to ensure that the developed product meet minimum requirements specified at the start of this project and to detect and correct as much as possible faults in the system developed.

DeMillo et al (1987) identified various bases for software testing. Due to time constraint, only functional testing will be carried out on this project. DeMillo et al (1987) noted that functional testing seeks to find out the difference between what the built systems can do and what it was intended to do. The intended functionalities of this project will be taken and tested one after the other to see if it is able to perform its intended function.

Registration

The system was tested to see if the data inputted by users in the registration form was inserted and stored in the database. In doing this, the validations performed in the form to ensure that users can only submit a form after filling up all required spaces was also tested.

After testing the validation done on the form, the validation was found to be working well. Users are not able to submit the form until all required fields are completed. Also, all the expected alerts are displayed to prompt the user to enter any unfilled required field. An additional validation was carried out in the password fields to ensure that the password re-typed by a user is exactly the same with the initial one typed. This was tested and the result was positive.

On submitting the registration form, a confirmation message was displayed to the user and when the database was queried, it was discovered that the registration details of the user was duly inserted in the right table.

Login

Like in the registration form, the validations implemented on the two fields in the form were tested to ensure that users enter both before being able to submit their login details. The e-mail address and password of a registered user was correctly entered to see if the system will log the user in. The test was positive as the system allowed the user access to the system and displayed a confirmation message to the user along with links to the next steps depending on what the user wants to do. On the other hand, a wrong login details were entered to see if the user will deny the user access from the system. The result showed that the system is able to deny non registered users access from the system.

Entering User Details

Like in the earlier tests conducted, test was conducted on the form for entering users' details to ensure that users enter all required field before being able to submit their details. On submitting a user's details, the system was able to display a confirmation message and showed the link to the next step. When the database was queried, the user's details were found to be duly inserted in the right table in the database.

Entering Fault Details

As usual, the form was also tested to ensure that the validations implemented on them are active. This was positive. The PHCN number (customer ID) of a user whose details is already in the database was used to enter fault details to see if they will be inserted in the right table. On submitting the form, the system displayed a confirmation message to the user. Also, when the database was queried, the data was seen to have been duly inserted in the database. On the other hand, the PHCN number of a user whose details is not in the database was used to enter fault details to see if the details will be captured by the database. It was observed that the system generated an error report indicating that the details were not entered into the database.

Tracking Fault Status

The form was tested for field validation and the result was positive. The PHCN number of a user that has already been used to enter fault details was used to track fault status. On submitting the form, all the fault status of all the faults reported by the user was displayed in a table. See figure 6. On the other hand, when a PHCN number of user that has no fault details was used to track fault, no fault status was displayed

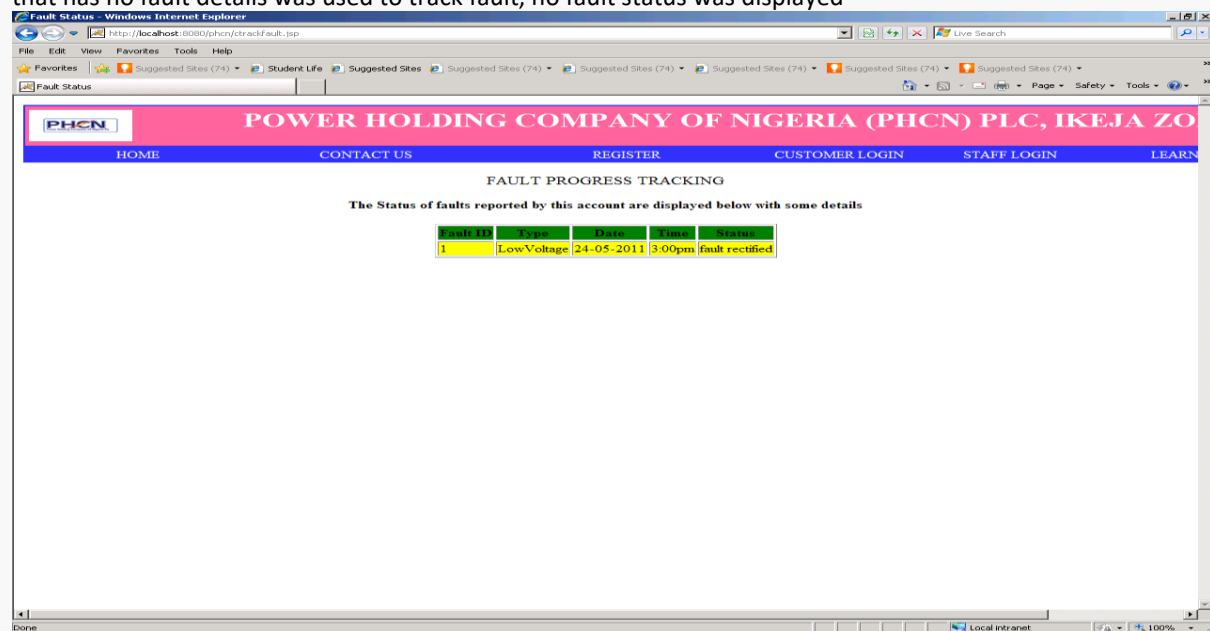


Figure 6: Screen shot of displaying fault status tracked by user.

Evaluation

Evaluation is a very important aspect of software development as it helps the developer to get the users' views on the product released. Time constraint in this project did not give room for elaborate evaluation. To evaluate this product, a potential user was allowed to use the system to report an imaginary fault. After using the system, the user was given a questionnaire to grade the system based on its ability to fulfil the requirements set out at the onset of this project. Due to lack of time, only one user evaluated the system. His responses are tabulated below in table 1

The feedback obtained from the user as tabulated below shows that the system was able to fulfil most of the set requirements with impressive rating. Apart from the system's ability to prevent multiple fault reporting, that was rated 'fair' by the evaluator, all other specified requirements was rated either very good or good. The implication of this result is that the system is good enough to fulfil its intended purpose but will have to go through some upgrade in future so as to achieve a 100% 'very good' score for all requirements in the next evaluation.

Table 1: Tabulation of responses from PHCN customer on evaluation of the system

	<i>Questions</i>	<i>Very good</i>	<i>good</i>	<i>fair</i>	<i>poor</i>	<i>Very poor</i>
1	Ability of the system to display vital information that will educate users on fault issues and how best to handle PHCN facilities so as to minimise fault occurrences.		X			
2	Ability of the system to capture and store in the database, vital information relating to the reported faults (such as fault type/category, fault time, description, location, etc.)	X				
3	Ability of the system to capture and store in the database some details (name, address, phone number, e-mail address) about the fault reporter to help in reaching him and the fault reported.	X				
4	Provision made by the system to allow PHCN management to update the system to indicate the status of the faults reported by users.	X				
5	Ability of PHCN management to use the information provided by this system to allocate fault rectification jobs to their field staff	X				
6	Ability to prevent multiple reporting of a particular fault			X		
7	Ability of this system to assist PHCN in fault management, and planning against high outage duration and frequency		X			
8	Effectiveness of the system's fault status tracking system	X				
9	Ability to improve PHCN services delivery to their customers.	X				
10	User registration and login system put in place to regulate activities of users.		X			

Conclusion

The main aim of this project was to develop an online fault reporting and management system that will allow PHCN customers to send online reports of electrical faults in their property to PHCN. The system was also designed to allow PHCN fault monitoring team to submit online reports of faults developing in PHCN facilities being patrolled by them.

It can be concluded that the key requirements of the system include: provision of a web interface for capturing these reports from users, need for the system to provide features for user registration and login, need for the system to compel users to enter their personal details before being allowed to enter fault reports, ability of the system to enlighten users on how to use the system and on maintain best safety practices while dealing with PHCN facilities in their properties, ability to allow users to track progress of repair of faults reported earlier, provision for entering details of fault being reported and the need to allow PHCN management to update the system based on faults reported and rectified was fully supported by the potential users based on the outcome of the primary research..

From what has been done, it can be concluded that using MYSQL technology, a comprehensive database was successfully built for to capture and store fault reports from PHCN customers and monitoring staff. It can also be concluded that with some HTML codes and some java scripts, login forms were constructed to help users submit their registration details, login details, personal details and details faults being reported. It can also be concluded that with the help of the technologies, JSP and Tomcat, information entered by users in the forms where successfully inserted into the database and can be retrieved on demand. It can also be concluded that the system has provided means of educating users on fault and safety issues by creating a new page called 'Learn more'.

Summarily, it can be concluded that the main aim of the project which is to develop an online fault reporting/management system for PHCN was realised. However, there features that will make this system even more effective and efficient that were not incorporated in this system due to time constraint. These features are discussed in the next section as recommendations for further works.

Recommendation for Further Works

For want of time, this system was built for user in Ikeja zone only. It is possible to build a harmonised system that can allow users all over Nigeria to report fault irrespective of the user's location.

Some automation can be introduced in this system to make it even for efficient in fault management. A good example is the incorporation of an automation system that automatically sends an e-mail to a repair team member informing him of a repair job allocated to his team and bearing all the required details, as soon as the manager updates the database with the information. Another example is also the incorporation of an automation system that automatically sends an e-mail or a text message to a customer informing him of the status of the recent fault reported once it is updated by the system administrator.

Due to want of time too, this paper did not look into the incorporation of special security features that will protect the system from online attacks. Further works could be done on this application to ensure adequate online protection of both the users and the owners of this developed system.

References

- Ahmed, M. (2010a). *Week 3: Tomcat*. Retrieved from <http://breo.beds.ac.uk>
- Ahmed, M. (2010b). *Week 3: Java Server Pages*. Retrieved from <http://breo.beds.ac.uk>
- Bhattacharjee, S. (2011). *Software development life cycle (SDLC)*. College of Agricultural Banking, RBI, PUNE. Retrieved from <http://cab.org.in/Lists/Knowledge%20Bank/Attachments/83/SDLC.pdf>
- Deming, W. E. (2006). *Software development methodology today*. Retrieved from http://ptgmedia.pearsoncmg.com/images/0131872508/samplechapter/0131872508_ch01.pdf
- DeMillo, R. A. (1987). *Software testing and evaluation*. Menlo Park, CA: Benjamin/Cummings Pub. Co.
- McGrath, M. (2002). *Java Server Pages in easy steps*. Southam, UK: Computer Step.
- McGrath, M. (2009). *HTML in easy steps*. Southam, UK: Computer Step.
- Pezze, M., & Young, M. (2008). *Software testing and analysis: Process, principles, and techniques*. Hoboken, NJ: Wiley.
- PHCN Ikeja Distribution Zone. (2010). *PHCN Ikeja zone website*. Retrieved from <http://www.phcnikejazone.org/home.html>
- Power Holding Company of Nigeria (PHCN). (2010). *Online complaint reporting system*. Retrieved from [http://www.phcnonline.com/TIMSCClient/\(Iscnyq45d2df3155zd2o4r3u\)/GeneralComplain.aspx](http://www.phcnonline.com/TIMSCClient/(Iscnyq45d2df3155zd2o4r3u)/GeneralComplain.aspx)
- Power Holding Company of Nigeria (PHCN). (2011). *PHCN website*. Retrieved from <http://www.phcnonline.com/TIMSCClient.aspx>
- Valade, J. (2010). *PHP and MySQL for dummies*. Hoboken, NJ: Wiley.