



ASHESI UNIVERSITY COLLEGE

**MANAGEMENT SYSTEM FOR NATIONAL HEALTH INSURANCE
HOSPITALS**

NATHANAEL ATO ASAAM

**2010
APPLIED PROJECT**

ASHESI UNIVERSITY COLLEGE

**MANAGEMENT SYSTEM FOR NATONAL HEALTH INSURANCE
HOSPITALS**

By

NATHANAEL ATO ASAAM

Dissertation submitted to the Department of Computer Science,
Ashesi University College
In partial fulfilment of Bachelor of Science degree in Computer Science

APRIL 2010

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:.....

Candidate's Name:.....

Date:.....

I hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision of thesis laid down by Ashesi University College.

Supervisor's Signature:.....

Supervisor's Name:.....

Date:.....

Acknowledgement

There was no way I could have finished this project without help. For this reason, I will like to acknowledge all those whose effort, contribution, counsel and assistance have made this dissertation a reality.

First of all, I am grateful to my supervisor in the person of Dr. Nathan Amanquah, who is the head of Computer Science Department at Ashesi University College. He was all that I needed for this project. He helped me find resources and software to use. He also assisted me during all my difficult times in school. Without him I would not have come this far in my educational endeavour.

Secondly, I want to thank all the wonderful people who proof read my dissertation. My dad, Rev. Stephen Asaam, my mom, Mrs. Juliana Asaam, my new Ashesi friend, Narkie Oforiwa Agbettor, and my cousin Patricia Araba Asaam. God bless you all for sparing time off your busy schedule to proof read my report.

Finally, I want to thank all and sundry whose name I have not been able to mention but have assisted in divers ways to make this project a success. I appreciate every ones effort so much and I know that I can count on all of you in such a time. Thank you once more.

Abstract

It has been observed that it is stressful for patients on the Ghana National Health Insurance Scheme to get quality health care. This stress is due to congestion at the hospitals that operate under the scheme. The congestion does not only waste the time of the patients but can also cause death. There is also the need of sharing medical and patient information among hospitals that operate under the scheme. Inability of sharing data renders the work of the scheme inefficient.

This dissertation seeks to address the problem of congestions at the hospitals that operate under the scheme. The project also seeks to address the problem of how to access and manage patients and dispensary records in every hospital that operates under the scheme. The approach of this project is to develop a mobile web application to keep and manage patients and dispensary records. The main assertion is that, this system will save time and will provide a means to access and share medical information in every hospital under the scheme.

The system proposed composes of a fault tolerant database system, a web interface and a mobile interface. The function of the fault tolerant database is keeping and managing the data. That of the web interface is assisting the hospital staff to administer their services. The mobile interface is a scheduling application that will enable patients to book appointments with a doctor at a specific hospital.

The web interfaces deals with the need of sharing data among hospitals and the scheduling application is the means of controlling the perceived congestion at the hospitals that operate under the scheme.

Contents

Acknowledgement.....	iv
Abstract.....	v
1.0 Introduction	1
1.1 Background	1
1.2 Universalization.....	1
1.3 Competitors	2
1.4 Objectives of Project.....	2
1.5 Problem Definition.....	3
1.5.1 Congestion	3
1.5.2 Old Flat File System.....	3
1.5.3 Data Sharing.....	3
2.0 Context and Proposed Solution	5
2.1 Increase Workforce.....	5
2.2 Introduce Competition.....	5
2.3 Computer Systems	6
2.4 The case of Taiwan.....	6
2.5 Approach.....	8
2.5.1 Fault Tolerant Database.....	8
2.5.2 Hosting	9
2.5.3 Types of Hosting.....	9
2.5.4 Authentication	10
2.6 Literature Review	11
2.6.1 Database	11
2.6.2 Web Application.....	11

2.6.3 Mobile Application.....	13
2.6.4 Database Replication	14
2.6.5 Authentication	14
2.6.6 HTTP Authentication	14
2.6.7 Form – Based Authentication	15
2.6.8 Hosting	15
2.6.9 Tools.....	16
3.0 Methodology.....	17
3.1 Design.....	17
3.1.1Database Layer Design	18
3.1.2Application Layer Design.....	18
3.1.3Middle Layer Design.....	19
3.2 Implementation	19
3.2.1 Database layer	19
3.2.1 Middle layer	20
3.2.2 Application layer	20
4.0 Testing & Evaluation	21
4.1 Administrator	24
4.2 Doctor	24
4.3 OPD	25
4.4 Dispensary.....	25
4.4 Mobile.....	26
5. Discussion and Conclusion	27
5.1Discussion	27
5.2Recommendations	27

5.3 Conclusion.....	28
References	30
Appendix 1	31
Appendix 2	32
Database Layer.....	32
Middle Layer	33
Web Application Layer.....	34
Mobile Application Layer	35

1.0 Introduction

The introduction chapter of this dissertation gives a brief background of the National Health Insurance Scheme. It also discusses the major progress the scheme has made, some of the rivalry that could occur if the scheme is not efficient, the objectives of the entire project, and problems that the project seeks to address.

1.1 Background

The National Health Insurance Scheme (NHIS) is one of the significant programs introduced in Ghana. The motive behind it is to eliminate the cash and carry system which prevents the poor in Ghana from obtaining affordable quality health care. The scheme has a fund which every registered Ghanaian contributes to. The scheme provides the following; child and maternal care, nutritional needs, and treatment of sicknesses like malaria, diarrhoea, upper respiratory tract infection, skin diseases, hypertension, diabetics, asthma etc[1]. However, there are some other health treatments which are not covered by the scheme. These include treatment of chronic renal failure, heart and brain surgery just to mention a few[1]. Citizens who have registered under the scheme get an ID card that gives them the right to benefit from the scheme.

1.2 Universalization

Currently, the NHIS is universal. This means that registered patients can attend any hospital that operates under the scheme. Initially, it was not so. Instead, every registered member was assigned a number of hospitals in the district where he or she was registered. But now, a patient can get medical care in Accra even if he or she registered in Takoradi. Making the NHIS universal is a great improvement to

the scheme but this brings the need of sharing and maintaining data among various health centers that operate under the scheme.

1.3 Competitors

There are other private health insurance programs like *Medex* that are doing well. This should not suggest that NHIS is not doing well. The problem is just how to deal with the congestion and how to manage and share patient, medical and dispensary records among hospitals. The private companies are competitors to the National Health Insurance Scheme Program in that, if the NHIS does not operate efficiently, these private firms can take up the entire health insurance business. This is might occur if their number increases and the cost of their services become relatively cheaper compared to what they are charging now. Currently, these private health insurance companies offer programs and services that are expensive. Usually, the average Ghanaian cannot afford these other health insurance programs. Right now, their services are only reserved for the elite in the society but the cost would become cheap in the future and this brings competition.

1.4 Objectives of Project

1. To control the perceived congestion at the hospitals.
2. To provide a means of sharing medical information among the hospitals.
3. To implement a prototype mobile web application solution.
4. To bring convenience to the patients.

1.5 Problem Definition

This section of the introduction section, discusses the major problems that the project is trying to address. The problems are congestion in hospitals, use of old flat file system, and inability to share data among hospitals.

1.5.1 Congestion

The NHIS though very useful has some shortcomings. One is the fact that it has brought congestion at the hospitals that operate under the scheme [2]. This congestion is due to increase in patient attendance at the hospitals that operate under the scheme [2]. Meanwhile, there has not been corresponding increase in staff for such hospitals [2]. This congestion in my opinion does not only waste time but can also take lives. That is why it is imperative to consider and address this issue urgently. Because of this problem, it looks like one is better off paying cash to the hospitals than depending on the scheme for health care. But this congestion problem can be addressed to make the program more reliable and efficient.

1.5.2 Old Flat File System

Another reason is due to the fact that some hospitals still use manual flat files or folders in record rooms. Retrieving patient records takes a long time. Consequently, patients who are under the scheme must wait long hours in long queues before they get health care. This renders the services of the scheme inefficient and must be addressed to bring convenience to the patients.

1.5.3 Data Sharing

Patients on the scheme can now access health care anywhere in the country where there is a hospital that operates under the scheme. This calls for finding ways to help these hospitals share and manage patient and dispensary records of a particular

registered patient very well. Additionally, there should be a way of tracking the medical history of a particular patient where ever he or she goes so that this information will be intact, accurate, current and secured. This will improve the overall service of the National Health Insurance Scheme.

2.0 Context and Proposed Solution

There are several ways of handling this congestion problem in the National Health Insurance Scheme hospitals. The first three sections under this chapter explain some of the available solutions that can be used to address the problem of congestion and sharing of medical information. The last section is a case study of Taiwan's National Health Insurance and how they have progressed in dealing with data sharing, congestion and security.

2.1 Increase Workforce

One of the ways that can be used to address the congestion problem is to employ more people to work at the records department of the hospitals. This will make retrieving of patient's and medical records faster and this will reduce the perceived congestion to some extent but sharing medical records among hospitals that are on the scheme still remains a problem. The problem with this solution is that remuneration for all these people has to be considered. The budget of the National Health Insurance Scheme has to be increased to account for these expenses. Getting more doctors in the hospitals is an extension to the solution of getting more people to work at the records department since this reduces the average waiting time before consultation but this also increases the cost of running the program. Besides that, it is not even easy to get qualified doctors.

2.2 Introduce Competition

Another solution is to bring competition into the health insurance business. The Government can accomplish this by encouraging and supporting new companies to come into the business. This way, the average cost of the private firms that also

provide health insurance services will reduce. When the cost of their program becomes affordable, more citizens will prefer the services of private firms to the National Insurance Scheme and this might help in reducing congestion at the hospitals that operate under the National Health Insurance Scheme. But this approach may not solve the problem of sharing of data.

2.3 Computer Systems

The next alternative is to rely on Computer Systems. This will normally come with some initial setup and training cost and also maintenance cost. The advantage of this approach is that, retrieving data is very fast and reliable and sharing of data among hospitals becomes feasible. Also performing computerized statistical analysis about the data which results in informed summaries of the data is possible. An extension of this approach is creating a network for the hospitals that operate the scheme and changing the ID card used into a smart card. This way, patient's personal, dispensary and medical information will be on a server. The chip of the smart card will only contain an ID number that relates to the information on the server. As such, the data can be accessed and manipulated by any hospital that is part of the network created and the card identifies the patient. This ensures security of information because only hospitals that are on the network get to access patient and medical information.

2.4 The case of Taiwan

Improving the services of a National Health Insurance Scheme by digitization of medical care information is not a new thing. Taiwan has a National Health Insurance Scheme just like Ghana[3]. It was launched in 1995 by their Bureau of National Health Insurance (BNHI) [3]. The country has now introduced something

called NHI IC intended to bring comfort to patients and ensure security of medical care information[3]. There is a medical network and the NHI IC allows flow of information through the network [3].

At first, they used a paper card which was just like the ID card Ghana uses for the scheme right now and additional cards that make operations tedious[3]. This creates the problem of congestion in the hospitals because before one gets medical care, all these many papers have to be inspected. Forgetting to bring one of these papers may mean that you have to go home and come back another time. Inspection of all these papers also takes a lot of time and keeps other people in queues and this is what normally brings long queues.

The NHI IC has guaranteed information security and removes all this tedious operations thus addressing the issue of long queues that bring congestion. The NHI IC is a smart card that keeps patient's personal information and medical information [3]. Because of the smart cards, all those tedious operations are no more due to the fact that the new smart card replaces all the old ones. Identification of patients and other operations has become faster and patients don't have to wait for long in queues. An additional benefit that this has brought is that, information has become highly manageable and fraud has been reduced to the barest minimum because only hospitals on the network get access to the patient and medical information.

2.5 Approach

This section explains the details of the approach used for addressing the problems explained before in the problem definition chapter. The first section gives a general overview and the subsequent ones dwell on specific essential components. The approach taken is to develop a mobile web application to address this problem. The reason why a mobile approach is the best is because mobile phones are cheaper compared to computers and many Ghanaians have phone but not computers. Secondly the phones come with internet connectivity GPRS SMS etc. some even have a lot of memory. The application consists of a fault tolerant database system which will store and manage patient's records, dispensary records and other information. The system will also include a mobile application and a web application. The mobile application will allow patients to book appointments with a particular doctor at a specific hospital before going to the hospitals using their phones. Additionally, it will also send notifications to patients about the status of their appointments. The web application will be used by the hospitals to store and retrieve patient's records, dispensary records and other details. Doctors and staff in records department and dispensary will use the web application while patients use the mobile application.

2.5.1 Fault Tolerant Database

An integral part of this project is the fault tolerant database. A fault tolerant database is needed so that in the case of any system crash like power cut the original data can be retrieved. One of the ways to attain fault tolerance is by database replication and that is what this project uses. Database replication is hosting the database on more than one database servers. So, there are multiple

copies of the same database. This enhances performance and ensures data security. It is also a load sharing strategy. Replication also supports long distance distribution of the data. That is, the servers can be some kilometers apart. If the organization that needs the data has branches at different locations, the servers near them will provide these branches access to the data. Additionally, the system can continue operating even if one of the servers is down. As such, it improves the availability of the entire system. More light will be thrown on database replication using MYSQL database in the literature review chapter.

2.5.2 Hosting

The web component and database component will be hosted on a computer that is connected to the internet and the mobile part will be deployed on phones of prospective patients. This makes the entire system ready for use. With this proposed solution, a computer network for all hospitals is not required since the application will be on the internet and will be accessible by the valid users. The only thing required here is security and this will be handled by authentication of users. The system will take care of security issues so that only valid users get access to the system. Because of that, an administrator is therefore needed to manage all users of the system and issues pertaining to security breach and authentication.

2.5.3 Types of Hosting

Throughout the previous section, I have been explaining hosting. Hosting is a vital part of every web or database application. First of all, the database must be hosted and secondly, the web application must be hosted before users can get access to the system. Currently, there are four types of hosting in the Information Technology Industry [5]. These are;

1. Free hosting
2. Shared hosting
3. Dedicated servers – managed
4. Dedicated servers – unmanaged.

Hosting will also be discussed in details in the literature review chapter.

2.5.4 Authentication

The other thing to explain in this chapter is authentication of users. Authentication of users is one of the significant features of this application system. Authentication means knowing who the user is and making sure that he or she is a valid user before he or she gains authorized access to the system. It is an issue of security and must be considered seriously in every system. There is also a sub section for authentication under the literature review chapter.

2.6 Literature Review

This section is a review of various components, programming languages, techniques and tools that are required to successfully complete this project. The chapter starts with the components and then ends with the tools.

2.6.1 Database

The database can be implemented on different database management systems (DBMS). Some of which are PostgreSQL, Oracle, MySQL and Microsoft Access. For this particular project MySQL is desirable and suitable because first of all it is free and open source. Meaning there is no need to pay for the DBMS and you can add to the original source code of the DBMS. One may have to pay for other DBMS like Oracle and PostgreSQL. Microsoft Access will not be able to handle the bulk of data expected to be accumulated by this system. Implementing this system with Microsoft Access will be okay for a while but as the data volumes increase, the system will become very slow and this will affect the efficiency and reliability of the entire system. Since retrieving information quickly from the database is very crucial to this project, Microsoft Access is not the best DBMS to use for this project.

2.6.2 Web Application

The web application can be divided into four main modules which are the administrator module, doctor module, patient record module, and the dispensary record module. All four will require at about three skills. These are a client side scripting language, a server side scripting language and HTML (Hyper Text Markup Language). Some of the client side scripting languages are JavaScript and VBScript. Server side scripting languages include PHP, ASP, and JSP.

Client side scripting languages will be used for processes that will run on the client browser when using the application and server side scripting languages will be used for processes that will run on the server. JavaScript and PHP were chosen for this project. The reason is that JavaScript is already available and enabled in most internet browsers like Internet Explorer and Mozilla Firefox. Using a language that is not already enabled in the common browsers may pose problems to the users of the application. VBScript was not used because applications written with VBScript may run properly on only Internet Explorer. It may not run well on other browsers like Firefox, and Opera Mini.

PHP is also free and open source and that makes it desirable because as a developer it reduces cost. Besides that, one general reason why PHP and MYSQL was chosen is the fact that there is a tool called Xampp that integrates the PHP and MYSQL and some other tools like apache web server and mercury mail server. Xampp makes it easy to combine PHP and MYSQL in a project since installing the two separately will involve other configuration that may be rigorous. It also has an engine that interprets and runs PHP scripts. This is very important since PHP does not run on client browser. As such Xampp can be used and that solves the problem.

Apart from the main three skills that has been mentioned earlier, this project also requires skills in another kind of scripting language called Structured Query Language (SQL). SQL can be divided into two main languages. These are Data Definition Language (DDL) and Data Manipulation Language (DML). SQL helps a developer to interface between the application and the database. Meaning it serves as a glue between the web or mobile application and the database. To insert into the database or retrieve from it or edit some record or delete some data one must

use SQL. This category of the SQL language is the DML and is used for accessing and manipulating the database.

There are PHP libraries that support database connection, database disconnection and executing SQL queries on the database. These libraries assist the developer throughout the entire development. DML is what was used extensively throughout this project. DDL is what is required for creating the database but that part is handled by tools like SQLyou and Phpmyadmin that is part of Xampp package. The programmer can just click, drag and drop and the work is done. No need to write DDL all by oneself because the database implementation tools do all the work for the developers.

2.6.3 Mobile Application

There are a variety of programming languages for mobile phones. These include java for mobile (J2ME), python for mobile, flash for mobile etc. Mobile application can be written in one of these languages and made to connect to a server or the application could be strictly on the server. In the second case, the phone will connect to the server and get all information from the server. The difference is that with the later, all data processing is done on the server and the client phone must run software similar to a browser. The first approach leaves some of the data processing burden to the client phone. Only special processes run on the server. The phone will just send the information to the server to do the server side processing. For this project I choose J2ME because java runs on most phones as compared with python, flash and other mobile programming languages.

2.6.4 Database Replication

MYSQL replication is asynchronous[6]. This means that, one of the servers acts as the master and the others act as slaves [5]. All updates and insertion of data is done by the master server but every reading or viewing of data is done on the slave servers. When an update or insertion is performed by the user, it is later on reflected on the slaves. The slaves must not always be running but the master must always be running. Reading of data is divided among the slaves and this improves performance. Analysis of the data can be done on the slave so that it does not affect the actual data on the master.

2.6.5 Authentication

Two techniques for dealing with authentication in a web database application are explained in the sections below. The first one is http authentication and the second one is form – based authentication. I want to mention that there are other ways beside the ones I explain in this report. Also, the one used in this project is form-based.

2.6.6 HTTP Authentication

This mode of authentication is based on the HTTP standard. When a browser requests for a resource such as html document, a script or image, the server can respond with a status code 401 which implies unauthorized[7]. The browser processes an unauthorized response by displaying a dialog box which requires entering of username and password. The browser then resends the original request with extra information including the credentials collected from the user [7]. The user will only get access to the resource after the server has processed the response and his credentials are deemed valid.

2.6.7 Form – Based Authentication

For session based applications where authenticated users need to be tracked, HTTP authentication does not help much [7]. Here, credentials are collected using forms designed and created by the developer. The developer has complete control over the authentication process. The authentication process will be to query a database and check whether the credentials supplied are valid or to read from a file that has the credentials of users and see whether the credential supplied is valid. Often, the credentials are stored in session variables after authentication to track the user.

2.6.8 Hosting

Free hosting as the name suggests is free[5]. But it comes with few features[5]. It is normally used by people who are trying something or prototyping a system. After about a while, usually a month or two, the account may be terminated by the organization that is providing the hosting service. There can also be security breach since a lot of developers who are novice or trying out something use it. Junk mails and hacking may be rampant.

Shared hosting is not free but it is less expensive than free hosting [5]. It has more features and resources for hosting web and database systems than free hosting [5]. But there may be a security breach. That is, hacking and junk mails can be rampant.

Dedicated servers – managed is quite expensive but comes with more features and resources like bandwidth, storage, database, emails services and the like[5]. The service provider manages and administrates the service. You don't have to be an expert. This is more secured.

Dedicated service – unmanaged is also expensive just like the one above. The only difference is that the service provider does not manage the service for you[5]. You need a competent administrator who has been well train to manage the service. Training also requires a lot of money and a lot of hard work on the part of the trainee.

2.6.9 Tools

The list below is a list of the tools used throughout the project.

1. Xampp
2. SQLyoug
3. Microsoft Visio
4. Notepad
5. Adobe Dreamweaver
6. Java Wireless Toolkit

3.0 Methodology

In all of the methods employed in this project, one thing was common, a component is design, implemented, tested a couple of times and then another component follow the same cycle. This is a software engineering process model known as component based software engineering. All components went through these phases of the entire system development process.

3.1 Design

The design phase for such a project is very crucial. Table 1, shows a breakdown of the entire system. The application layer is the part of the system that the users will interact with. This is where processed data has to be displayed in a comprehensive and nice looking way to users. All issues of interaction design like navigation, layout, menus, forms, tables and so on are considered in this phase. In this project, there were two components of the application layer; the web application and the mobile application.

The database layer is the part of this system that stores and manages the information needed to run the application smoothly. Design issues here include identifying entities, data modeling, normalization, definition of attributes of the entities and their data types.

The middle layer is a bridge between the database layer and the application layer in that, it is made up of abstractions, and functions that enhance communication between the other two layers. Design issues include concepts like classes, functions, procedures, and inheritance.

Application Layer
Middle Layer
Database layer

Table 1

3.1.1 Database Layer Design

It is normally good to design the database first because that gives one an idea of the data that the system will use and its structure. First of all, it is essential to identify entities and attributes of those entities and relationships between entities. The next thing is to design a high level representation of the data. Such representation is an entity relationship diagram. This is a high level representation of the data. After that, normalization is the next activity and that results in a database diagram. The entities that were identified in this project were patient, doctor, pharmacist, opd_officer, medicine, consultation, medical records, NHIS card, health facility, appointment. Attributes and the relationship amongst attributes were also identified. After that, the database was modeled using Entity Relationship (ER) diagrams first, and then the tables in the ER diagram were normalized. This resulted in a database diagram. Normalization is a technique for improving the structure of a database. All these diagrams were drawn using Microsoft Visio. Refer to appendix 1 for the ER diagram.

3.1.2 Application Layer Design

With the web and mobile part, rough sketches of how the applications must appear have to be made first. This can be done using paper and pencil. All the major menus, tables and forms have to be drawn. The color for the application need to be decided at this phase. For this application, blue was the color used. Light blue for

background and deep blue for the actual appearance of the application. Various versions of the menus and navigation were made before final choice was made. This project uses a menu on the right side of the web page. This is a slight deviation from the common way of placing main menu at the top of the page but it improves visibility and ease of use because there are other features of the application at the top.

3.1.3 Middle Layer Design

During design of the middle layer, a list of functions that help communication between the database and the applications has to be made. Given the functional requirement of a project, such a list can easily be made. Typically, there will be functions for every relation designed at the database level. Some of these functions will insert into the relations, others edit records in the relations and some others retrieve data from the relations in the database and others will delete data from the relations in database.

In this project, all these functions were written and put into a class called *nhis* during implementation phase. This class inherits from another class called *db*. The *db* class has abstractions and functions that enable connection to the database, and execution of SQL queries. The inheritance issue is possible because PHP is an object oriented programming language that supports classes and inheritance.

3.2 Implementation

3.2.1 Database layer

The database implemented could store the following kind of information. Patient's personal information, patient's medical history, dispensary records, consultation

appointment, basic information about hospitals staff, login details of users, hospitals, clinics, and polyclinic.

3.2.1 Middle layer

The middle layer consists of two files db.php and nhis.php. The db.php file contains functions for connecting to the database and for executing SQL query on the on the database. The nhis.php file contains a class called nhis that inherits all the functions and variables in the nhis.php file. The main functionality of the procedures in this file is inserting data into the database, retrieving data from the database, updating data in the database and removing data in the database.

3.2.2 Application layer

In this project, the application layer is made up of mobile and web. The web application was implemented by first getting a template. Then action handling scripts written in php was implemented. The template can be created using HTML and CSS technologies. The mobile application was implemented using j2me technology. Using java wireless toolkit one can create midlets and develop the mobile application. The wireless toolkit comes with documentation that is handy in developing the application.

4.0 Testing & Evaluation

Generally during implementation, some kind of testing takes place but this section describes the testing that was done after a component was deployed. The criterion for testing and evaluating the system was based on these parameters; functionality, appearance, responsiveness, visibility, feedback and ease of use. Below is a brief description of what the parameter means.

The web and mobile interfaces were both tested based on these parameters. Typically, one of my participants will be allowed to perform a simple task and later give scores based on the parameters. Participants performed one or more of the following tasks;

1. Adding new user
2. Viewing users
3. Adding new patient
4. Viewing patient
5. Adding new medicine
6. Viewing medicine
7. Adding consultation
8. Viewing consultation

Figs 1 to 4 are screen shots of some of the views that were tested and some description.

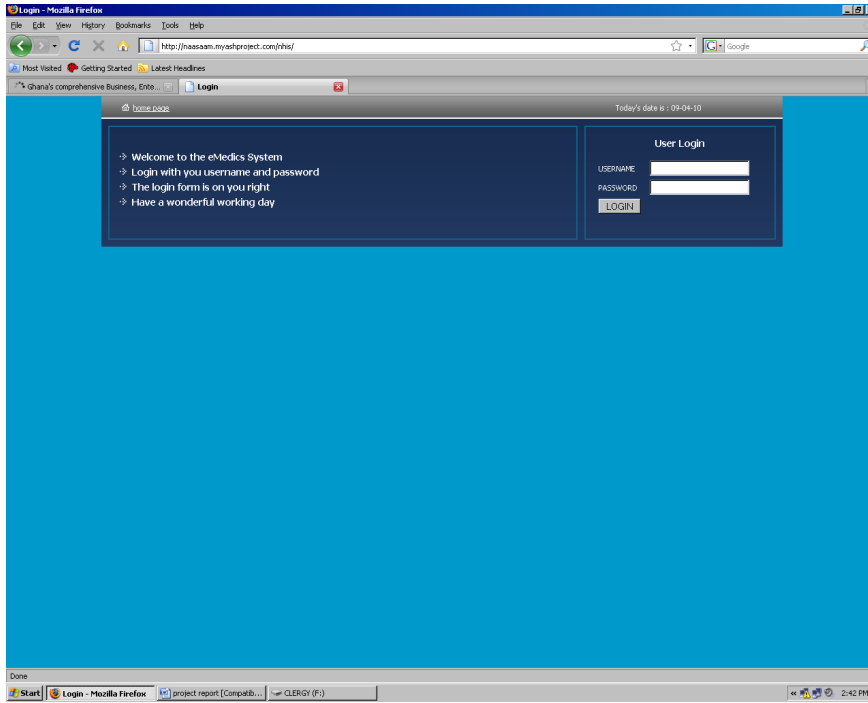


Figure 1 Screen shot of Login

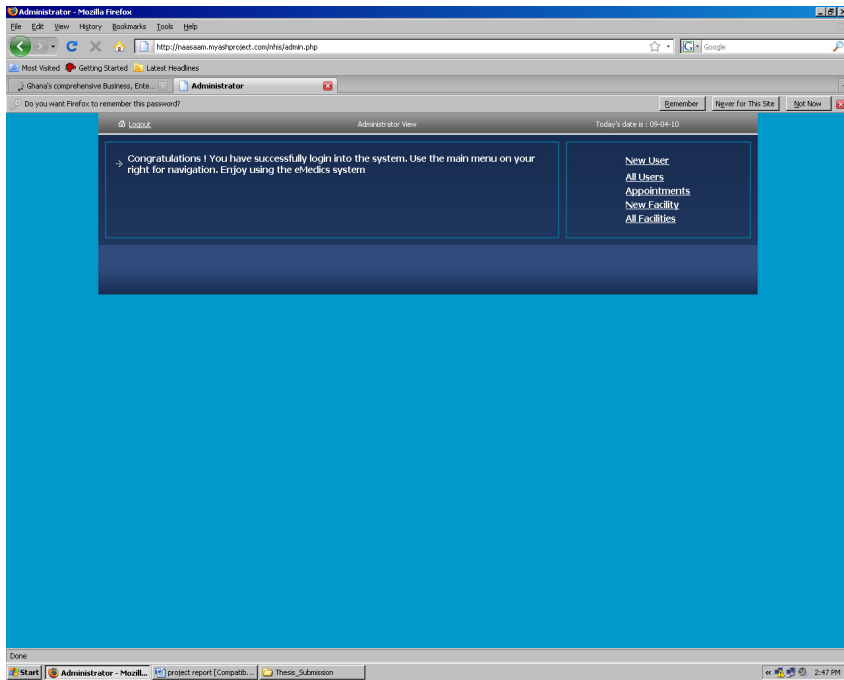


Figure 2 Screen shot of the Administrator view

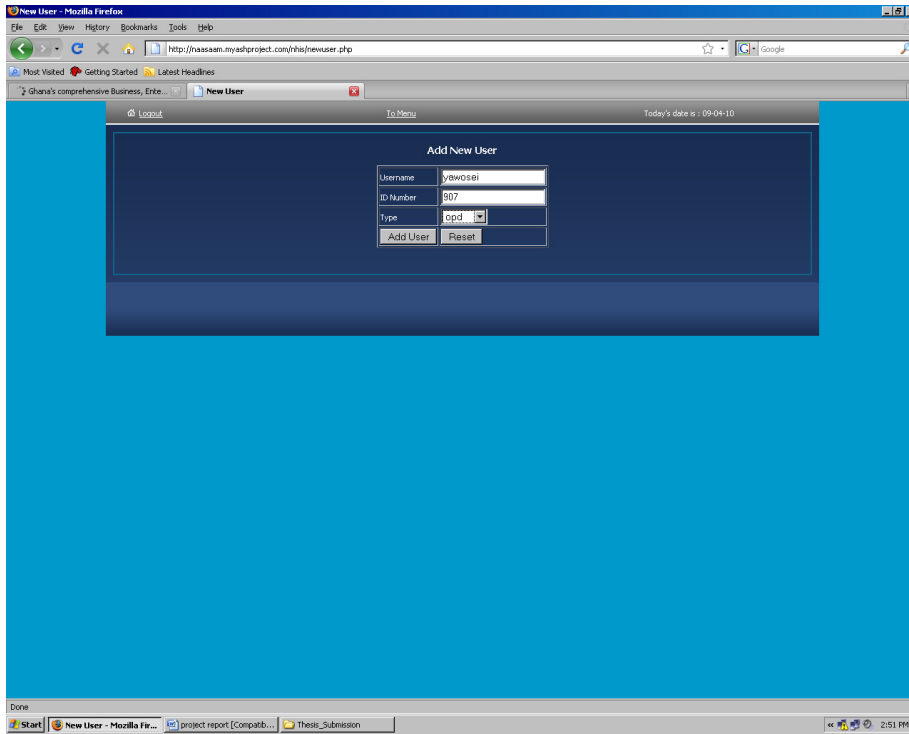


Figure 3 Screen shot of page for adding new user

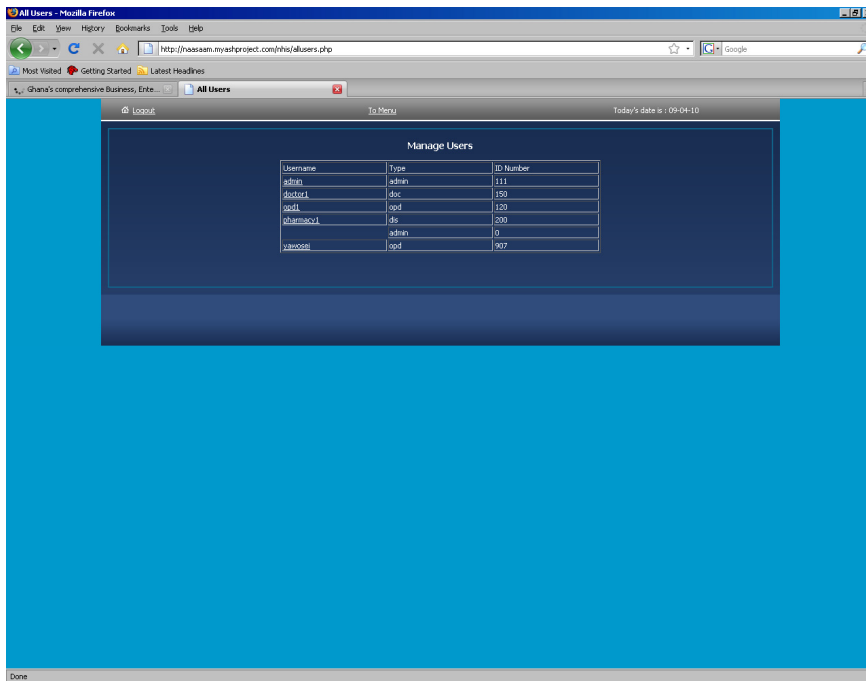


Figure 4 Screen shot of page for viewing all users.

At the end of the project these are the components that were completed successfully. For the web part, I finished all the four modules. Below are the four web modules, their menus and associated functionality.

4.1 Administrator

New user: For adding a new user who will use the system. A user can be a doctor pharmacist or opd officer. Each user was identified by a unique identity number and the type of user; dis for pharmacist, doc for doctor, and opd for opd officer.

All Users: Shows a list of users. For viewing and managing all users. User details can be edited using this functionality.

New Facility: For adding a new medical facility like hospital or clinic to the system.

All Facilities: Shows a list of all facilities whose details have been entered into the system. For managing facility. Details of facilities can be edited using this functionality.

4.2 Doctor

Patient Records: For viewing patient's medical records for that particular day. The information displayed includes temperature, blood pressure, heart beat, height, and weight.

Patient History: Displays a list of patient's history. This is made up of the patient's records above and also information about consultation for every visit to the health facility.

Search Patient: Displays patient's personal and other information.

Consultation: For saving details about a consultation doctor had with a patient. Information saved includes symptoms, diagnosis, prescription, doctor's identity, patient's identity, date and general comments of the doctor.

Appointments: For managing appointments patients have booked with doctor using their phone.

My Account: For managing user's personal and other information.

4.3 OPD

New Patient: For saving information about patient. This includes personal and other information.

Search Patient: For searching for and displaying patient personal and other information.

Patient Records: For saving patients medical records whenever he comes to the hospitals. The information include temperature, height, weight, blood pressure, heart beat, and date.

My Account: For managing user's personal and other information.

4.4 Dispensary

New Dispensary: For saving information about patient dispensary. Information saved includes the medicine, instruction, date and quantity of medicine.

All Dispensaries: Displays a list of all dispensaries. For managing dispensary records.

New Medicine: For saving details about a medicine. The details include name of medicine, quantity in stock and expiry date.

All Medicine: Displays a list of all medicine. Used for managing all medicine.

Search Patient: For Searching for and display patient personal and other information.

My Account: For managing user's personal and other information.

It must be emphasized that the *My Account* page has some errors and does not work for all views.

4.4 Mobile

Part of the mobile application was also completed. The part completed has two main functions. For booking appointments and for view appointments. The booking and the viewing can be performed by pressing the schedule and view command respectively.

5. Discussion and Conclusion

5.1 Discussion

It is important to note that a mobile web approach was chosen for dealing with this issue of congestion and sharing of data because, almost every Ghanaian uses a mobile phone. Comparing this to the fact that not every Ghanaian can offer to own a computer makes it reasonable to have the appointment schedule on phones. This approach is good because it is cost effective yet addresses all the problems discussed under the problem definition section. Booking appointment will help control the perceived congestion at the hospitals. The mobile interface of this solution provides a means of achieving this. The fault tolerant database system also will ensure the availability and reliability of the system. The web interface will help the hospitals in their day to day operations. All these will improve the services of the National Health Insurance Scheme. Patients will be more convenient.

5.2 Recommendations

I recommend that research be done about the kind of phones NHIS patients use. This will help find out about the feasibility of this approach. The results of this research can help in making decision as to whether this approach is the best to dealing with the congestion problem in the hospitals that operate under the NHIS.

Additionally, I believe that dedicated service – managed or unmanaged is the best for the NHIS. Using free hosting may pose a lot of security issues which may be undesirable. These may include hacking. If anyone hacks into the system, it implies that he can get access to undue information. If the information of the system is unsecured, the system becomes liable to all sorts of attack. This attack

may lead to corrupting of very essential data or reading of some confidential information.

Finally, I recommend that this project be extended by establishing an autonomous computer network for the health centers that operate under the scheme.

5.3 Conclusion

So far, as it has been outlined in this report, this system is meant to address congestion in the hospitals that operate under the National Health Insurance Scheme and to share data among the hospitals. Other possible solutions may be desirable but they come with some disadvantages that outweigh that of this proposed solution. That is the main reason this approach was chosen.

But this is still just a prototype of the actual thing to be done. Further research and feasibility studies can be done in order to ascertain the viability of this approach. More extensions and modifications can be done to improve this work in order to get the desired result.

Getting an autonomous computer network that all the hospitals, clinics and polyclinics that operate under the scheme will use will be a very good extension to this approach but that is much more expensive. It requires finding ways to link these health facilities nationwide. But it is much more secured. Hacking becomes very difficult and ensuring information security is easier.

Finally, this problem of congestion and sharing of medical data among hospitals that operate under the scheme is one that can be solved. This project and its report respectively demonstrate and explain one of the ways for handling

the situation. Nevertheless, there may be other possible solutions that can be used and these can be considered as well.

References

[1]Naa Okailey Tagoe "What do we know about NHIS"

<http://www.modernghana.com/print/123423/1/what-do-we-know-about-nhis.html>

[20th October 2009]

[2]Dr. Kobina Arthur Kennedy Improving the National Health Insurance Scheme

(NHIS) <http://news.myjoyonline.com/features/200909/34935.asp> [12th November

2009]

[3]"Bureau of National Health Insurance About NHI"

http://www.nhi.gov.tw/english/webdata.asp?menu=11&menu_id=290&webdata_id

=2637 [12th November 2009]

[4]"Data Replication" <http://www.topbits.com/database-replication> [20th January

2010]

[5]Tizag "web host types" http://www.tizag.com/webhost/host_types.php

[February 2010]

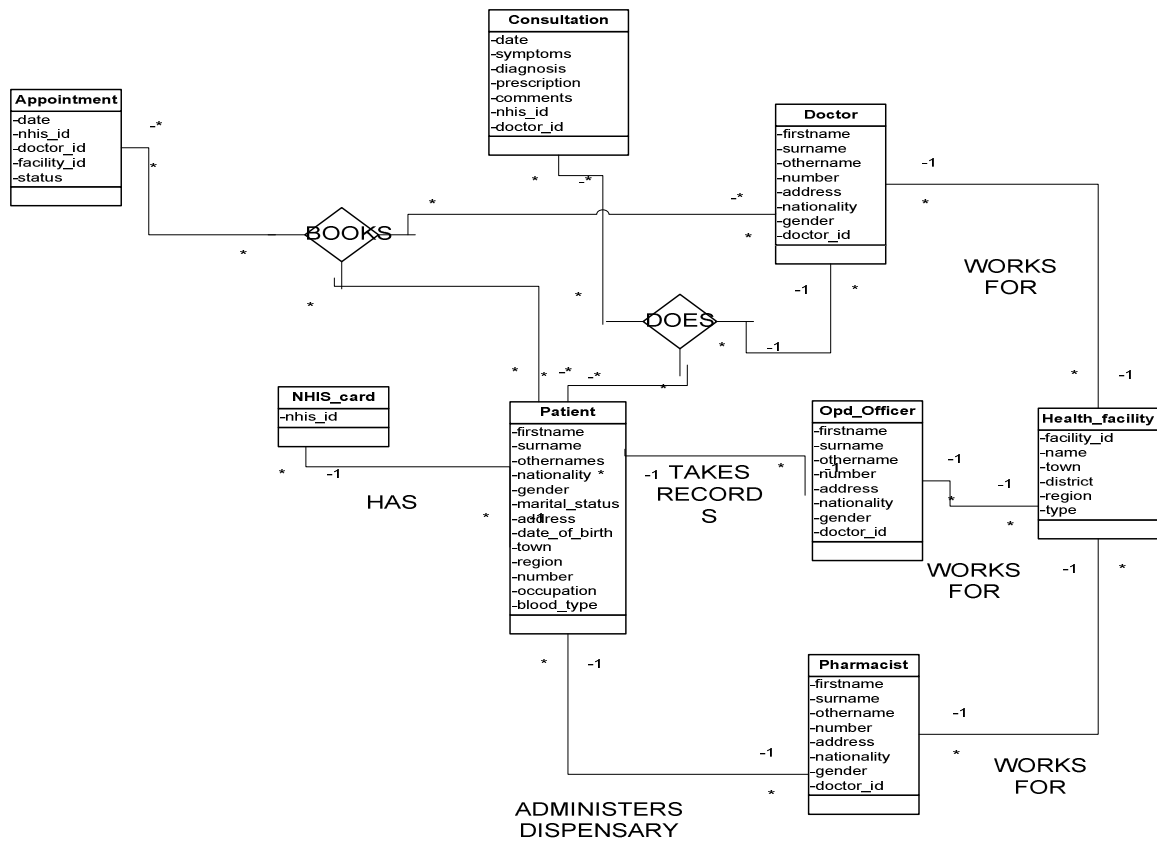
[6] <http://dev.mysql.com/doc/refmen/5.0/en/replication.html> [22nd January 2010]

[7]Hugh E. Williams & David Lane,"Web Database Application with PHP and

MYSQL" O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol GA

95472 , 2nd edition 2004

Appendix 1



Appendix 2

Database Layer

As already mentioned, the database was implemented on a MYSQL DBMS. The database diagram that resulted from the design was what was used while doing the implementation. You can use a software tool called SQLyog. When SQLyog is started, a dialog box will appear. Click the connect button and the main application with all its menus will appear. Then go to DB on the menu bar and select create database. A dialog box will appear again as a result and then enter the name of the database and click create.

Afterwards, select Table on the menu bar and click create table. This will result in the appearance of a window. Enter the attributes of the relation or table that you want to create in the fields provide by the window. All other design restrictions like primary keys, foreign keys, and data types are supported by this window. Just need to select them to make them operational. Repeat this for all other relations on the database diagram that you design.

It has to be mentioned, that the xampp package has a program called Phpmyadmin which could also be used for implementing the database in MYSQL. Once xampp is installed, Start the Xampp control panel and run Apache and MYSQL as services. Then Start a browser and type localhost as the address of a web page. From this click Phpmyadmin from the left menu on the page that will appear. After this, you follow through and complete the implementation. From this stage onwards, it is not much different from using SQLyog.

Middle Layer

Implementing the middle layer requires knowledge and skill in object oriented programming. What has to be done typically is to write the functions identified at the design stage in a programming language that is appropriate for the project; in this case php. Also, as the system is being develop, more functionality may pop up and functions for all those new operations as well need to be written.

For every relation in the database, a function was written for inserting new data, a for retrieving old data, for editing the old data and deleting old data in the database. Separating some of these functions into new classes and files makes debugging, modification and extension easier. Some classes may inherit from other ones that have already written. Just like this project, the class `nhis` inherits from an already existing class called `db`. The `db` class and `nhis` class where all written in php and stored in files called `db.php` and `nhis.php` respectively. These two files make up the middle layer.

The `db.php` file needs to be discussed further to clear any anomaly. Two major functions in the `db.php` file are of interest.

1. `query ($query)`
2. `connect ()`

Function `query ($query)` takes a parameter. The purpose of this function is to accept an SQL query and send it to the database for execution. It first makes sure that a connection has been established before it sends the query. If not, it returns false indicating that connection was not successful. Contrarily, it returns the result of this query if it everything was successful.

The second function is the one that connects the application to the database. It uses internal variables of the db class to do the connection. During implementation, value of some of the variables had to be changed so that it fits this project. So it is apparent that the db class is generic. The function returns a link if connection was successful or false if connection was not successful. The nhis.php file inherits all variables and functions of the db class and so have direct access to those variables and functions. The nhis class itself contains a lot of functions which cannot be discussed in this report. The db.php file is a file is that has been written long ago before this project.

Web Application Layer

Implementation of the Application can be a bit challenging. The web application was first developed by designing a web template that to some extent matches the design features laid down. Color and page layout were the most significant criterion for choosing the template. Throughout my experience as a computer science student, I have realized that this method is always faster than creating my own template or design. However, after settling on a particular template that kind of fits the design, one has to go through more hard work by modifying the template. This activity can take a lot of time, because you need to understand all the HTML codes that were already part of the template before you can modify the template.

After ensuring that the template is fit for the entire web system, you now need to focus on individual views. The major aim of using a template is to ensure consistency on all views or pages. But each page normally has its own appearance in the form of tables, forms and list. Additionally, each page had its own interaction.

For each individual page, you have to develop them separately in another file and test to make sure that they work.

Issues like get and post request and the PHP code to handle them may all be developed into a single page which does not look one bit like what the design requires. The difficult task now is to convert that developed page into that template. All that has to be done normally is to make multiple copies of the template. For every page, try to integrate the html, c.s.s and php into the template. Sometimes you just copy and paste into the template and it is fine and works well. But at other times you copy parts of that developed page into some table in the template and copy and paste the other parts of the page to a different place like the header of the template and still do some rigorous modification before it works.

The aim of this activity is good functionality and appearance. If the final page is not working but looks great you don't settle for that. Also if the functionality for that page is perfect but the appearance is not consistent with the designed template, you still have to do more rigorous work to make everything fit perfectly.

Mobile Application Layer

The mobile application was developed using an IDE called Java Wireless Toolkit. When the IDE is started you will see a toolbar. Click New Project. A dialog box will appear. Enter the following fields; Project Name and Midlet Class Name. Then, select the targeted platform. For this kind of project, JTWI is a good choice. Then click OK. You will see a description of the project message on the console. This includes instruction of where to place specific files that make up the project. After this, you will implement the actual java midlet using a text editor. For this

project notepad was used. When you have placed the midlet class file in the appropriate folder and done likewise for all other files, you can build the project and run it to test how it will behave on a phone using an emulator provided by the IDE. You will keep modifying and adding to the midlet class file and testing till you have fully developed all the functionality.