

ASHESI UNIVERSITY COLLEGE

**A CLINICAL SUPPORT APPLICATION TO FOSTER COLLABORATION
BETWEEN COMMUNITY HEALTH NURSES AND MEDICAL
SPECIALISTS**

BY

EFUA GAMBRAH – SAMPANEY

Dissertation submitted to the Department of Computer
Science, Ashesi University College

In partial fulfilment of the requirements for the award of
Bachelor of Science degree in Management of Information Systems

2013

Declaration

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 dissertation were supervised in accordance with the guidelines on supervision of dissertation laid down by Ashesi University College.

Supervisor's Signature:.....

Supervisor's Name:.....

Date:.....

Acknowledgements

My sincere gratitude goes to my supervisor, Aelaf Dafla for his immense involvement and guidance in this project. I am also very grateful to Kojo Gambrah- Sampaney for his constructive criticism and the guidance he gave me throughout this project.

Without the morale support of close friends, this project would not have been a success. To my colleagues who contributed in various ways to the success of this project I say thank you.

To my parents and my sister, who have always been there for me and have been most supportive, thank you very much.

Abstract

Community health nurses of the CHPS program find themselves operating in remote environments sometimes, and are cut off from the training sessions and specialist interaction that colleagues in urban areas are exposed to. This means for them that they are unable to provide quality healthcare to the communities they serve in some instances. Because of this, there is little faith in the program by the community members who tend to travel long distances to urban areas, where they believe they will receive good health care.

To improve upon the performance of community health nurses, a mobile application is proposed for their use. This mobile application would foster collaboration between the nurses and specialist and also provide clinical support and educational materials for these nurses. The benefits of this application are access to clinical advice and professional development of community health nurses and better access to healthcare by communities.

Contents

1.	Introduction.....	1
1.1	Introduction.....	1
1.2	Problem.....	2
1.3	Objective.....	4
1.4	Overview of mHealth.....	5
1.4.1	The promise of Clinical Decision Support Systems in mHealth.....	5
1.4.2	Overview of existing remote health training and clinical decision support programs and software.....	6
1.4.3	Conclusion from Literature Review.....	8
1.5	Overview of Report.....	8
2	Design.....	10
2.1	Software Requirements Specification.....	10
2.1.1	User Classes and Characteristics.....	10
2.1.2	Use Case.....	11
2.1.3	Functional requirements.....	11
2.1.4	Non- functional requirements.....	12
2.2	Design.....	13
2.2.1	System Architecture.....	13
2.3	External Interface Requirements.....	15
2.3.1	User Interfaces.....	15
2.3.2	Hardware.....	19
2.3.3	Software Interface.....	20
2.3.4	Communication Interface.....	20
3	Implementation.....	21
3.1	Technology and Tools Used.....	21
3.1.1	Operating Environment.....	21

3.1.2	Technologies.....	21
3.2	Database Architecture.....	23
3.2.1	Database Schema.....	25
3.3	Challenges.....	26
3.3.1	Display of JSON in DOM elements.....	26
3.3.2	Getting values of individual form elements.....	26
3.3.3	Implementation of SQLite Database.....	26
4	Test and Results.....	27
4.1	Unit Testing.....	27
4.2	Component Testing.....	28
4.3	System Testing.....	29
5	Conclusions and Recommendations.....	30
5.1	Application Benefits.....	30
5.2	Future Work.....	31
	Bibliography.....	33
	Appendix.....	35
	Code implementations specific to Android Interface.....	35
	Code implementations specific to Web Interface.....	38

1. Introduction

1.1 Introduction

Health delivery in Ghana has been a concern of many, as it leaves much to be desired. In an effort to improve health delivery in the country, the Ghana Health Services (GHS), a wing under the Ministry of Health (MOH) introduced the Community-based Health Planning and Service (CHPS). The aim of CHPS is to decentralise health delivery and make health services available at the community level. It was adopted in 1999 as a national health policy initiative to reduce geographical access to healthcare (Nyonator, Awoonor-Williams, Phillips, Jones, & Miller, 2005). The Northern Region in Ghana benefited from a pilot of the program, and recorded success, which led to the establishment of CHPS compounds in other regions. Currently, CHPS operates in all the ten regions, with a total number of 389 compounds in Ghana as at 2009 (Ghana Health Service, 2010).

Even though the CHPS program currently has been implemented nationwide, the program is not being effective as it should be. To ensure the effectiveness of community based health delivery in Ghana, attention must be paid to all stakeholders, including community health nurses. Community health nurses play a vital role in the CHPS program. They are the direct, and in some cases only point of contact, for any form of medical consultation for community members. As such, they must be extremely knowledgeable in their medical practice, in order to provide quality healthcare for the communities within which they operate. The community members themselves are also a major stakeholder group that

cannot be taken for granted. Without their co-operation, the nurses will be overwhelmed with their task. This is the reason why the community must also have some insight as to how to take care of themselves and their environments properly to prevent diseases and epidemics.

In light of this background, this project sought to improve the effectiveness of the CHPS program through education and performance support of community health nurses and preventive education for members of the community. To achieve this, the project employs mobile technology as it is presently the most effective tool for reaching field employees like community health workers (Skillsoft, 2012). The rapid increase in the accessibility and use of smartphones which permit a lot of meaningful functionality and user interaction has granted this project the opportunity to develop an interactive mobile health education application for community health nurses and the members of the communities within which they operate.

1.2 Problem

The CHPS program was introduced to improve healthcare delivery in Ghana. Even though the program has recorded successes, for instance in Navrongo in the northern region where it was piloted, the program itself introduces certain challenges and complexities of its own that are yet to be addressed. These challenges include inadequate training of nurses, reluctance of communities to accept proposed medical solutions to health needs and difficulty in dissemination of information to nurses in remote parts of the country (Regional Health Directorate, 2009).

Sometimes, community health nurses find themselves operating alone in a CHPS compound. Not being adequately trained to address certain health conditions, healthcare provision in the communities are sometimes not regarded as effective. The community members begin to lose faith in the community clinic, and start looking for better alternatives, which may be very far from the community (Yakuba, 2012).

Also, heavy reliance on traditional healers by the community members tends to impede on the work of the community health nurses. This is because patients only visit the nurses after several unsuccessful visits to the traditional healers; by the time they avail themselves to the nurses to examine them, their health problems would have already aggravated (Adu, 2012).

To keep community health nurses on top of their medical practice, a mobile application that would update nurses on modern treatments for ailments, would be instrumental to the nurses in their practice. Also, if nurses could direct certain challenging questions regarding their practice to specialists, certain health cases would not have to wait till a specialist comes into the town. Nurses would be able to handle more medical cases and prevent the aggravation of sicknesses and improve the health of community members. Community members would also build confidence in the community health nurses and be encouraged to patronize the services of the nurses for their own convenience.

A mobile application on a smartphone would enable nurses show videos of preventive education of diseases, consequences of waiting too late to see a medical practitioner about a health condition, amongst others to

community members who are still reliant on traditional healers. This would help to reduce the number of serious cases that are brought to the community health nurse, but could have been prevented if the first point of contact for consultancy had been the health nurse.

To address the problems concerning difficulty in dissemination of information to remote parts of the country, a mobile application could be developed to handle the dissemination of information to nurses in remote areas on a large scale and be cost effective at the same time through the use of the internet.

1.3 Objective

The objective of this application is to build a mobile collaboration and clinical support application on the Android operating system for community health nurses in Ghana. The application will deliver educational materials to community health nurses and members alike through the health nurses. It will also be a platform for nurses to ask specialists any challenging clinical questions regarding their practice in the communities. This way, they are able to dispense better and quality health services to the communities. Making available educational materials for community health nurses would also help them in their professional development as nurses. Providing preventive health education with the application through the community health nurses will improve awareness on primary healthcare issues of members of the communities, and also reduce the number of disease breakouts.

1.4 Overview of mHealth

1.4.1 The promise of Clinical Decision Support Systems in mHealth

mHealth, otherwise known as mobile health is a term used to refer to the use of mobile and wireless devices in the practice of medicine and public health to improve health outcomes, health practices and health research. mHealth has currently become very popular amongst the health community as a result of the ubiquitous nature of mobile phones, technology like Bluetooth and NFC and the increase in internet accessibility and reliability. According to the United Nations, by 2011 the number of mobile cellular subscriptions had grown to an estimated six billion, including 1.2 billion active broadband subscriptions. Mobile cellular penetration levels were also 87 per cent worldwide and 79 per cent in the developing regions. Moreover, 45 per cent of the population worldwide launched 3G broadband services and was covered by a high-speed mobile broadband signal. mHealth is currently being applied in areas of education and awareness, remote monitoring, communication and training of health workers, disease and epidemic outbreak, diagnostic and treatment support amongst others (Vital Wave Consulting, 2009).

Clinical decision support is a process for enhancing health-related decisions and actions with pertinent clinical knowledge and patient information to improve health and healthcare delivery (Healthcare IT News, 2012). Clinical decision support systems (CDSS) are very significant in the area of clinical knowledge management technologies and are most effective when paired with Electronic Health Records. CDSS are designed to integrate medical knowledge base, patient data, and an

inference engine to help generate case-specific advice. These systems aid doctors and nurses to verify at the point-of-care that they are making accurate patient diagnosis, prescribing the right medication, amongst others.

CDSS are mostly designed and built as mobile applications due to the fact that they are supposed to assist physicians make decisions at the point-of-care. With the increase in popularity of mobile phones, more specifically smart phones, CDSS has a great potential of becoming a major assistance and learning tool for mHealth.

1.4.2 Overview of existing remote health training and clinical decision support programs and software

1.4.2.1 *Health eVillages*

Health eVillages is a program of the Robert F. Kennedy Center for Justice & Human Rights in liaison with Physicians interactive, to ensure safe and efficient healthcare delivery by providing the latest in mobile health technology to clinicians in the developing world. The goal of the program is to provide medical assistance through mobile health technology to challenging clinical environments around the world. Physicians Interactive is a 'resource for healthcare information, medication samples and mobile decision support tools to medical professionals everywhere' (Physicians Interactive, 2011). Health eVillages gives smart phones with medical reference material and software to healthcare professionals to ensure that they have immediate, reliable medical resources that they need. The healthcare professionals refer to the mobile apps on the devices given to

them when they are faced with challenging clinical decisions. All devices given out have installed on them drug guides, medical alerts, journal summaries and references from over 50 medical publishers' resources powered by Skyscape (Health eVillages, 2012). Countries and communities that have benefited from Health eVillages' pilot projects include Haiti, Lwala and Kijabe in Kenya, Mattuga and Acuma in Uganda, China and rural Louisiana.

1.4.2.2 Skyscape

Skyscape is a clinical decision-support tool that provides users with a vast repository of medical resources. The Skyscape application provides a platform for users to choose from over 600 mobile medical resources, in partnership with medical publishers. Targeted users of the platform are physicians, nurses, students and schools and groups. The platform is able to help physicians and nurses with early diagnosis, as there is a vast repository with fast and intelligent search mechanism to find what the user of the application searches for at any point in time. A component of the applications on the platform is MedAlert™ that allows physicians and nurses to stay on top of the latest medical alerts and breaking news (Skyscape, 2013).

1.4.2.3 Primary Healthcare Nursing Promotion Program

The National School for Nurses of Coban in Guatemala created the Primary Healthcare Nursing Promotion Program with funding from CIDA (Canadian International Development Agency) in an attempt to improving diagnosis and treatment. The program organised virtual training courses for healthcare workers via telephone and other two way communications. A

number of the 2004 graduates of the course became tele-facilitators in rural communities. In this pilot program, the responsibility of these tele-facilitators was to link their rural communities with specialists in urban areas (Youn, 2012). The program was run from 2002 to 2006.

1.4.3 Conclusion from Literature Review

This application software being developed takes advantage of clinical support for mHealth. Like existing CDSS, a medical knowledge base would be integrated into the application. However, unlike existing CDSS's, this application will allow nurses to channel clinical support requests to specialists rather than an inference engine. Also, request for support is not based on integration of an existing patient medical history, but from observations of the nurse. The underlying functions and features of the application also foster collaboration between community health nurses and specialists. Inferring from the above, the application being developed can be described as a collaboration and clinical support system as for community health nurses.

1.5 Overview of Report

The outline for the remainder of this document is as follows. Chapter two (2) of this report is on the design of the application. The chapter expands on the design methods used in designing the product. It states what the application is expected to do and its intended users. Chapter four (3) highlights on the implementation and approaches employed in the development of the application. It includes the technologies and tools

used and the platforms on which the application is developed. Chapter four (4) expands on the tests done on the application before deployment and the results recorded from these tests. Chapter five (5) concludes with recommendations for future work on the application.

2 Design

2.1 Software Requirements Specification

2.1.1 User Classes and Characteristics

The mobile application will have the following user classes:

1. Community Health Nurses – This group consists of trained health nurses under the CHPS program. Users in this group are expected to have basic knowledge in smartphone navigation.
2. Health Specialists – This group consists of trained medical specialist. All members of this group are assumed to be technologically competent.

Even though anticipated users are expected to be technologically competent, for an easy understanding and navigation of both the web and mobile interface, the application is built to be as user friendly as possible. The effectiveness of the application is dependent on the extent to which its users find it easy to operate.

2.1.2 Use Case

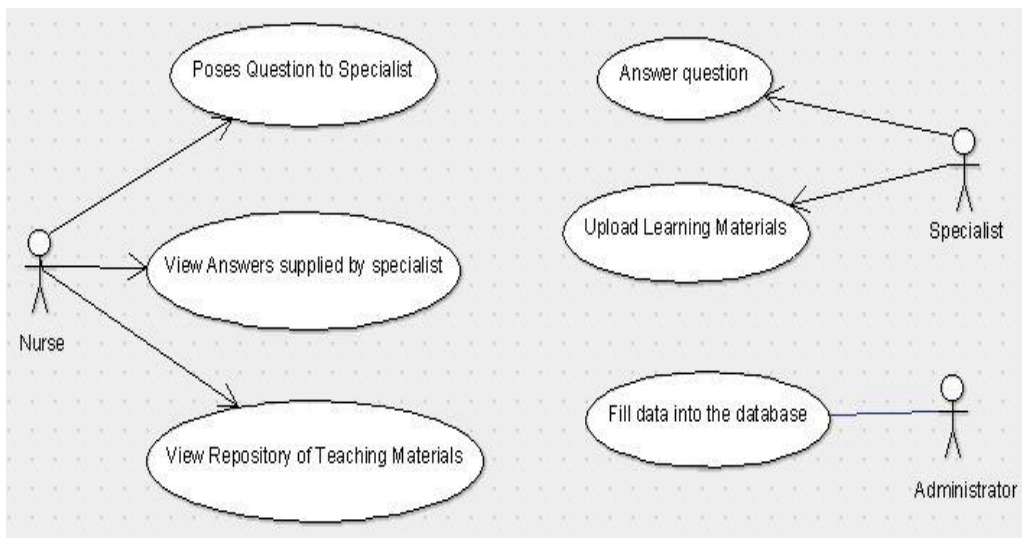


Figure 2:1 Use case diagram of the system

2.1.3 Functional requirements

- **Post Question** - Community health nurses using the mobile application are able to send challenging questions regarding their practice to specialist over the internet.
- **View and answer questions** - Registered specialist are able to login to a web application, where they can view questions asked by nurses, and answer them based on the category the questions fall within.
- **Use same reply for already answered questions** - Specialists are able to match questions with answers that have been given already.
- **Upload teaching materials** - Specialists are able to upload helpful teaching and learning materials in the form of pictures and articles amongst others to the application.

- **View a repository of teaching materials-** Nurses are able to view a large repository of teaching materials submitted by specialists.

2.1.4 Non- functional requirements

2.1.4.1 Performance Requirements

- The application requires an internet connection for most transactions. Both the web and mobile application requires specialists and nurses to login with their credentials over an internet connection to use the application. However, there is offline support for the android application as some nurses operate in rural areas where internet connection is unstable. Responses from requests made to the server are stored in a local database so that a nurse does not have to continually go online for data that has already been downloaded.
- Quick and easy retrieval – Data retrieval on both the web and mobile interfaces of the application is easy and fast. Asynchronous mechanisms are used in fetching data, so that the application can still send and receive data from the server without interfering with the display and in most cases the behaviour of the current page.

2.1.4.2 Security Requirements

The application allows only registered specialists to log in to the specialist page, in order for nurses to receive reliable answers to their questions. The mobile application also allows only registered and logged in nurses to pose questions to specialists.

2.1.4.3 *Secure Database*

User passwords are encrypted in the database using md5 encryption to prevent malicious attempts to steal passwords of users.

2.1.4.4 *Software Quality Attributes*

The mobile application is reliable as it can function almost as well as it would with internet connection. The user interfaces presented are user friendly as it is also easy to understand and navigate, making the application easy to use.

2.2 Design

2.2.1 System Architecture

System architecture encompasses a set of important decisions about the structure and organization of the software. It shows the various elements that come together to form a complete working software, and how these elements communicate with each other.

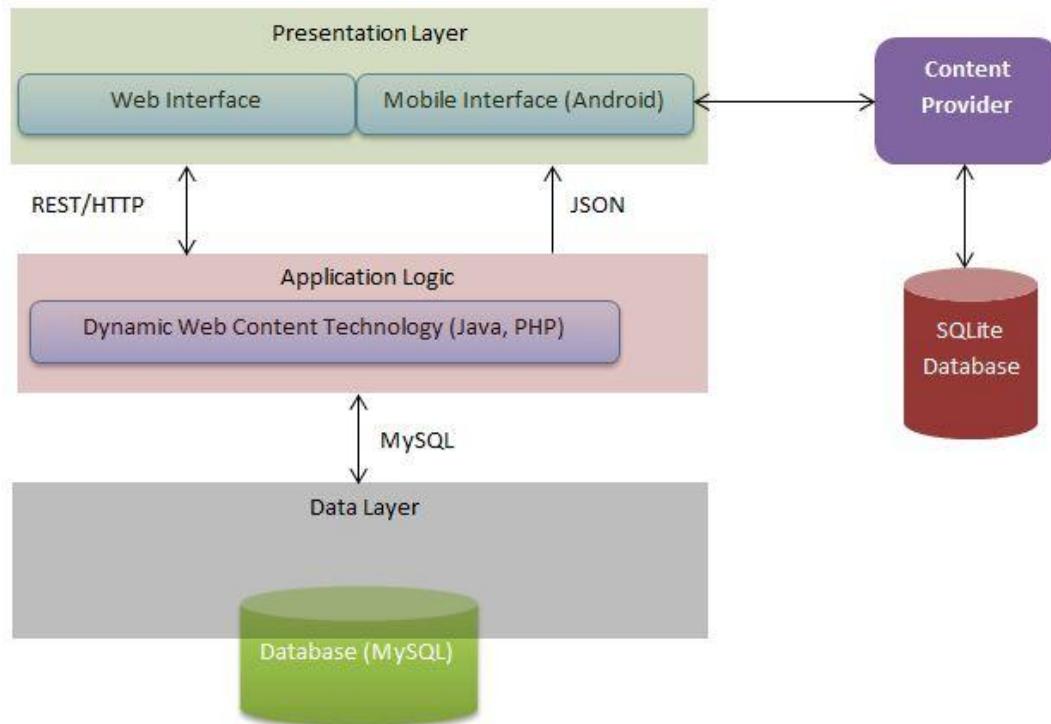


Figure 2:2 Diagram of System Architecture

2.2.1.1 Data Layer

The data layer handles all communication to and from the database. Requests to the database are written in Structured Query Language. Information and responses retrieved are sent back to the application logic layer for processing, after which it is presented to the user. Data to be stored include community details, nurse details, specialist details, questions, answers, educational materials, amongst a few others.

2.2.1.2 Application Logic

The application logic handles communication between the presentation layer and the data layer. Requests to the data layer from the presentation layer are processed by a server-side script (PHP) that can communicate with the database in the database. The application logic is basically the

'brain' of the application. It processes commands and makes logical decisions regarding which data to get from the database for the presentation layer. Results received from the database are JSON encoded.

2.2.1.3 *Presentation Layer*

The presentation is the layer that the user comes into direct contact with. It specifies the views the user is presented with. Both the web application layer and the mobile application receive JSON formatted data. The web application uses jQuery to parse the JSON data into DOM elements and the android application uses a JSON parser class to parse the JSON data into android objects.

2.3 External Interface Requirements

2.3.1 User Interfaces

The following is a description of the user interface design for the mobile and web application. Mock-ups shown in this section are not necessarily the committed screen designs.

2.3.1.1 *Android Interface*

Tabs are created on the page for navigation to three views to access features of the application. The first interface presented to the user when the application is started is the home screen of the application. This screen is called from the first of three tabs on the screen, and displays a list of titles or description of educational materials submitted by specialists and administrators of the application. The user can select an educational

material on this page, and be directed to another page to view the details of the material that was selected.

The second tab displays a page that allows community health nurses to send questions to specialists. A field is created on the screen for the user to enter their question. A spell check is conducted on text entered in the field to ensure that questions sent to the specialists are error-free. Also the text being entered, no matter how long it is, is visible to the user. A button below the text field sends a request to the server with the question entered and inserts the question into the database. When the question is submitted successfully, a success message is displayed to the user.

The third tab, called 'My Library', contains a list of options for viewing items that the user has added to his/her library. Questions asked by the user for which answers have been received are automatically added to the user's library. Elements in the list describe the contents that the user will see when he/she clicks on it. For instance, 'Pharmaceutical Questions' will display the questions that specialists tagged as Pharmaceutical and the answers to them.

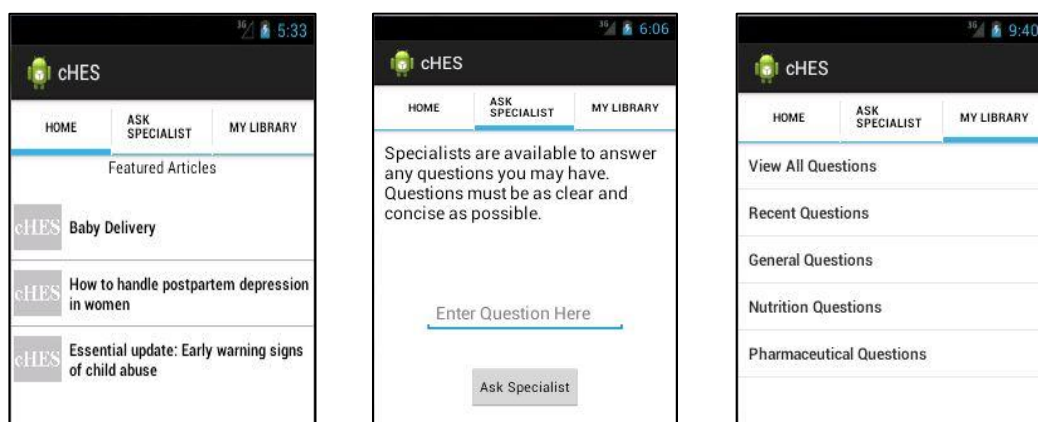


Figure 2:3 Mock- Up of tabs and flow of the android interface

2.3.1.2 Web Application Interface

A web interface was designed for use by specialists. This is because specialists are not expected to be as mobile as community health nurses, as they tend to operate from their consulting rooms. Therefore they would be expected to access the application from laptop/desktop computers. Specialists would also be able to have a wider view of the pages in order to navigate quickly and easily. A specialist would be able to access the application from a mobile device when necessary. To access the web application, the user types in the URL of the application into the address bar of the browser.

The first screen presented to the user is the login screen, where the user can enter his or her credentials, which is authenticated against a data store of registered specialists.



Figure 2:4 Login screen for specialists

When the user is able to login successfully, he or she is presented with a view of:

- **Questions submitted by community health nurses (Answered and unanswered).** The left side of the screen is populated with

questions that have been asked with the name of the nurse that asked the question, and the time the question was submitted.

Below every question displayed on the page is text that reads 'Answer'.

- 'Answer' shows a form that is hidden initially when it is clicked on. The form displayed is an input field for specialists' answers.
 - A group of buttons for selection of category of question. After entering the answer in the text field, the user is required to select the category within which the question that was just answered falls. More than one button can be selected to indicate that the question falls within more than one category.
 - The submit button inserts the answer entered and the category chosen, when clicked on into the database.
- **Questions already answered by specialists and grouped under categories.**

The right side of the screen initially shows a navigation bar displaying the categories of questions which have been answered fall under. When the user clicks on a category, the questions and answers under that category are displayed. The purpose of this section of the page is to allow specialist to answer new questions with existing answers.

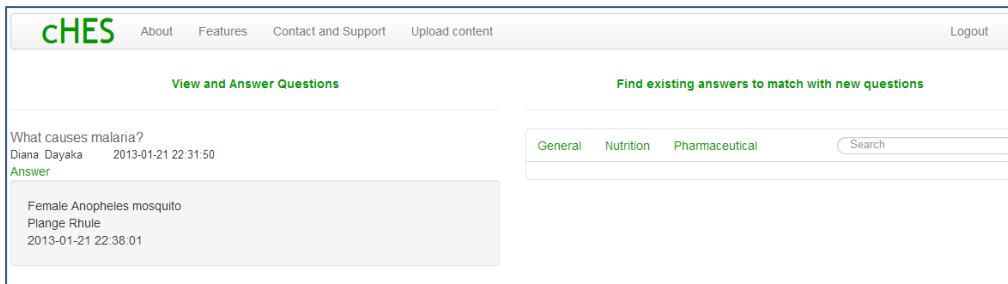


Figure 2:5 Initial display of home screen

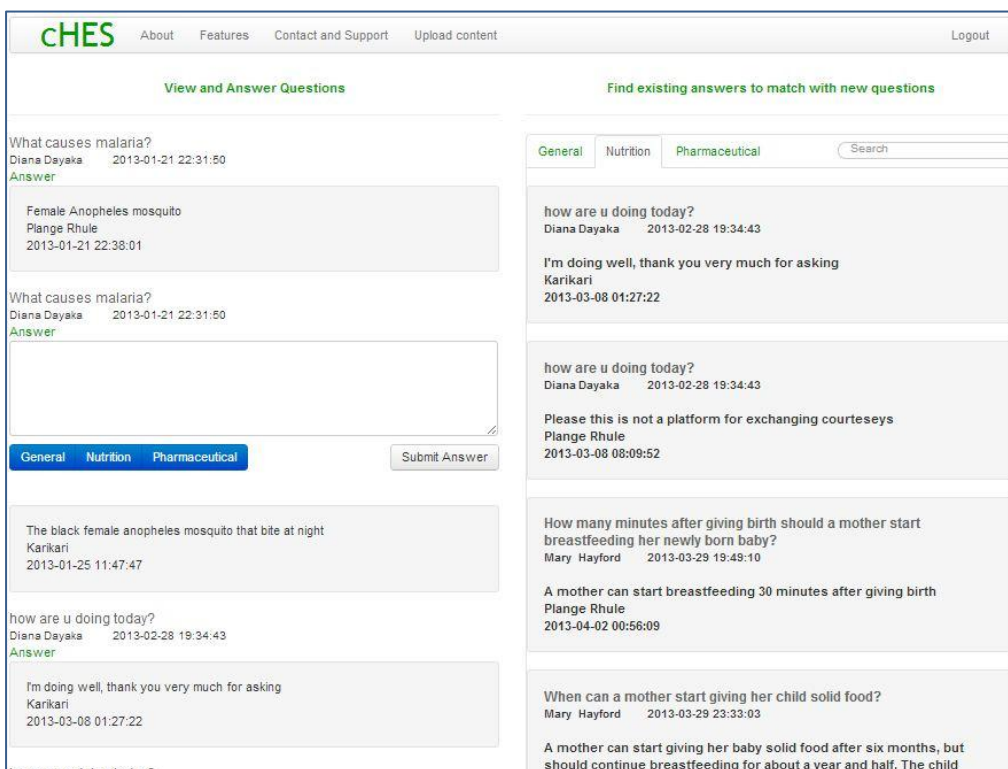


Figure 2:6 Display of home screen during the answering process

2.3.2 Hardware

The mobile application will be built on Android API Level 16 (JELLY_BEAN), and therefore requires an Android device to run the application.

2.3.3 Software Interface

The application, both web and mobile interfaces will communicate with an Apache server which will contain PHP classes and functions as well as with the MySQL database. In addition to this, the mobile interface will communicate with a SQLite Database. Also the mobile application can be run on a minimum API level of 11 even though the target API level for the mobile application is API Level 16 (JELLY_BEAN).

2.3.4 Communication Interface

Both web and mobile interfaces will use HTTP requests to communicate with the SQL database. Question details and answer details will be formatted using JSON. In addition to this, the mobile application will communicate with the SQLite Database through the Android Content Provider, which is made available in Android's API.

3 Implementation

3.1 Technology and Tools Used

3.1.1 Operating Environment

1. **Android Platform:** This is currently the world's most popular mobile platform (International Data Corporation, 2012). The Android platform was used in building the mobile application which community health nurses would be using. The platform was chosen because it supports a number of functionality that can easily be integrated into an application. It is also Java based, and I have experience with Java development, making it easier to implement required features. It also has great online support because a lot of mobile developers are currently developing for android.
2. **Web Application:** The application also employs a web application interface for use by specialists. A web application is suitable for the application because it does not restrict users to a particular machine. This implies that users can access the application from any computer. It also allows users the liberty to use whichever browser that may be available to them to access the application, as the application was designed to be compatible with popular browsers like Google Chrome and Mozilla Firefox.

3.1.2 Technologies

The following outlines the technologies employed in developing the application. The reason for choice of these technologies was that they are popular, have great online support and have cross-platform support as well.

1. MySQL – MySQL (Structured Query Language) is a fast and reliable open source relational database management system (RDBMS). This technology was used to create the database for the application.

2. PHP – PHP is a server side scripting language designed for interactive web development. PHP has built-in API's for accessing the MySQL database. PHP was used in this application to communicate with the database created using MySQL.

3. HTML5, CSS, jQuery, Ajax, JSON– These technologies were the main technologies used to develop the look and feel of the web application. HTML5 is the mark-up language that was used to create the web pages for display in the browser. CSS (Cascading Style Sheet) is a style sheet language used to describe the appearance of HTML elements on the web browser. CSS was used for customisation of all HTML elements in the web application. The Twitter Bootstrap CSS was employed largely in the design and development of the web interface.

jQuery is also a JavaScript library that makes HTML DOM (Document Object Model) manipulation through CSS easier. It also makes Ajax calls much simpler to implement. JQuery was used in the web application to manipulate HTML elements such as forms and divs. The simplicity and ease of its Ajax feature was also used to make asynchronous requests for data through the server, making the application more responsive and faster. An implementation of server request using JQuery/Ajax is shown in Code 3.

JSON (JavaScript Object Notation) is a lightweight data-interchange format. JSON was used in this project to parse query results from MYSQL

into JSON format with `json_encode()`. An example code of encoding result from the database is shown in Code 5 in the Appendix. In the presentation layer, the encoded JSON string is decoded and presented in a human readable format. To present the data retrieved in JSON format in a nice and readable output to the user, JSON Parsers were used. Code showing the JSON parser methods for both the web and android interfaces are shown in the Appendix in Code 4 and Code 2 respectively.

All these technologies come together to make the web application very interactive, fast and beautiful. Requests to the server is also minimized, due to ability of jQuery to load all contents from the database when the application is started, but make them visible to the user only when information is requested.

SQLite Database – A data storage implementation accepted by the Android API is the SQLite Database. This database was used in the application to store responses that had been retrieved from the database, so that repeated connections to the internet would not be established whenever the nurse wants to access a resource that she had already downloaded from the internet.

3.2 Database Architecture

Database architecture is the rules and processes that dictate how data is stored, accessed and manipulated by components of a system (Wingenious, 2005). The database architecture of this application is very critical to the performance of the application. The architecture was built to accommodate integrity, reliability and performance.

Data Integrity – The implementation of the database design enforces integrity of data entered into the database. Data integrity ensures the quality of data by assuring accuracy and consistency.

Integrity constraints – In the implementation of the database, integrity constraints were written to define rules and limitations on rows and tables in the database. Below are some integrity constraints that were written and their functions in the database created for this application.

Integrity Constraint	Function
Foreign Key	Foreign keys were a very important constraint that needed to be enforced to ensure referential integrity. For instance, QUESTION_ID was a foreign key in the 'answers' table because an answer needed to be directed at a particular question.
Primary Key	This constraint was set on all primary identifier columns of tables. Since ID's were the primary identifiers, all ID's in the database were given the Primary key constraint.
Unique Key	Set on community health usernames to ensure that the database does not contain same usernames, referring to different nurses, as most authentication used in the application

	required the combination of username and password.
--	--

Table 1 Integrity constraints in the database and their functions

3.2.1 Database Schema

The final schema of the database is shown below. The schema is a result of several development stages that called for adjustments of the database schema.

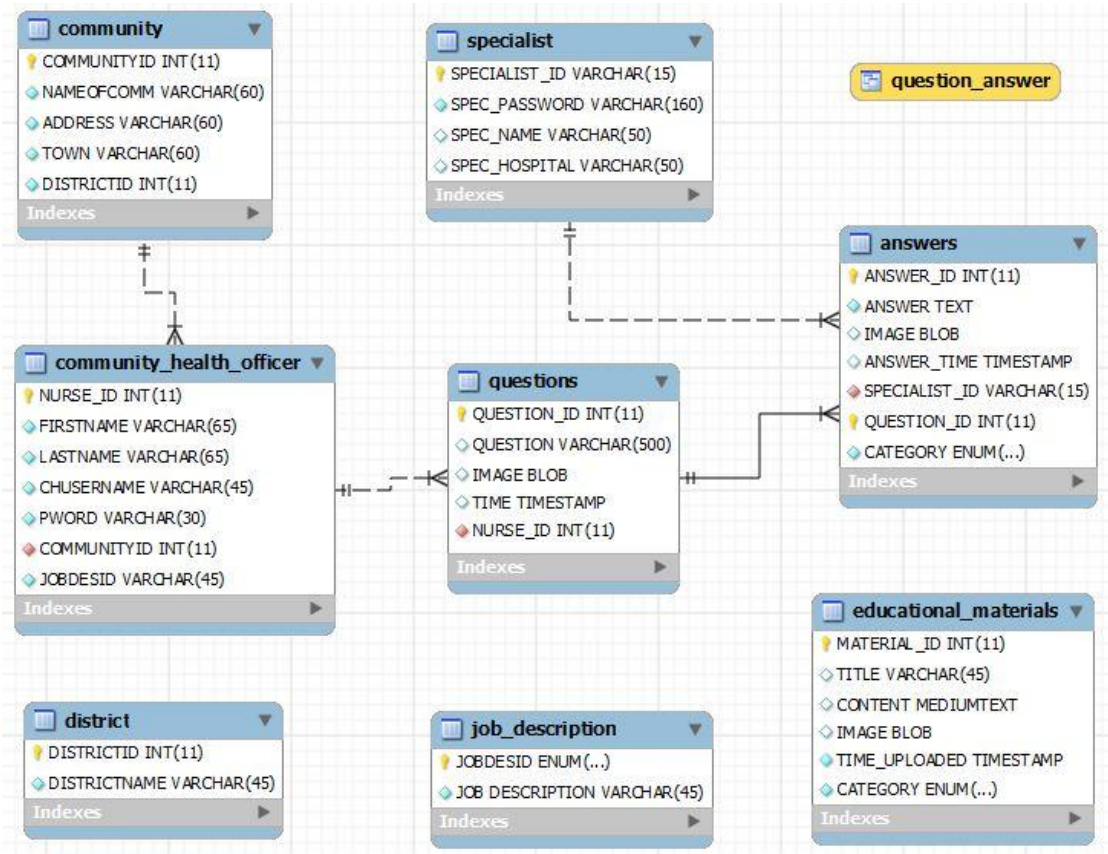


Figure 3:1 Entity Relationship diagram showing the physical schema of the database

3.3 Challenges

3.3.1 Display of JSON in DOM elements

A challenge encountered in the development of this application was the displaying of JSON objects received in the database into DOM elements on the web interface. Achieving a 'one-to-many' display for questions and answers was especially difficult to achieve. For instance, if there was more than one answer to a question, the multiple answers had to be displayed as a group beneath the question that was asked. However, in the development, what was displaying was one question to one answer. In that if a question was answered five times, the question would appear five times with individual answers. This challenge was eventually solved with an algorithm.

3.3.2 Getting values of individual form elements

Retrieving values entered into or associated with individual form elements on the web interface was a challenge. Implementing an iteration to fetch individual form elements and data associated with them took quite a lot of time. For instance, if a specialist answered a question, the answer had to be stored in the database with its associative question, and retrieving these details was a challenge that was eventually overcome.

3.3.3 Implementation of SQLite Database

The Android API makes use of SQLite Database as a storage means. In regards to implementation of the data store, the following were encountered. Data stored into the SQLite implementation of the database could be retrieved alright. However, on restart of the android emulator, the data would no longer display on the device, making the data appear to

be non-persistent. Running a command shell of the Android Device Bridge (ADB), would also show the fields entered into the database, but would no longer display them on restart of the application. A lot of time was spent trying to debug a non-existent bug. After the realization that an emulator restart did not necessarily start the same device that received the fields into its database, the SQLite database functionality was tested with a physical android device which showed that the data inserted was persistent.

4 Tests and Results

The following tests were conducted during the development of the application to verify that the software was meeting its stated functional and non-functional requirements.

4.1 Unit Testing

Unit testing in this application involved testing individual methods and object classes.

testPoseQuestion – This test was conducted to check the functionality of the PoseQuestion class. This test revealed even though the class was functioning as expected, bad data being sent to the database was not handled properly by the system. This defect was fixed by introducing checks that would attempt to force the user to enter text that suited the purpose of the text field provided for asking question.

testAnswerQuestion – This test also revealed a similar behaviour of the answerQuestion class as the poseQuestion class. The defects in this class were also solved using with checks that prompted the user for valid input.

testViewQnAs – This test was conducted for viewing questions and their respective answers in both interfaces. It was realised that even though the right output was being received, the presentation of the retrieved questions and answers was not interactive enough. Questions that had multiple answers would display whenever an answer was given to it. To improve upon the presentation, the JSON data being fetched was put in an iteration to ensure that a question was displayed only once, even if it had multiple answers.

4.2 Component Testing

Component testing is done to test all the components that come together to form a system. In this section, the following components were tested.

JSONParser – A JSON parser was employed in fetching the JSON data retrieved from the database and displaying it on the web and android interfaces. Even though parser methods for the interfaces were different, they both required the URL to the PHP function that retrieved the JSON data from the database. Passing of the URL as a parameter to the JSON parsers and specifying a display format on the user interfaces produced the expected output, proving that the JSON parsers were working as expected.

Android Async Task – Android asynchronous was a component used in running UI threads in order for background threads not to interfere with the main application. This component was also tested by passing the URL of database function that fetched objects from the database to a method in the Async task class as a parameter. The class was tested by calling its

.execute() method inside the application, which output the expected results from the database.

SQLite Database- Data persistence of the data entered into the SQLite Database was tested by installing and running the application on a physical android device. The fields inserted were displayed and the device was restarted several times, to ensure that the data was persistent.

4.3 System Testing

At this point in the development testing, the various components are integrated and tested to check that the entire system is functioning as expected. This section of testing focused on interaction of objects and components in the system. This test was done to check the compliance of the system against specified functional requirements. The following tests were conducted on the system.

1. Retrieval of educational materials – Authentication of nurses is successful. Nurses are able to view educational materials uploaded by specialists on their android devices.
2. Pose question to specialist – Nurses are able to post questions from android devices to specialists for them to view in their web interface.
3. Retrieve library of nurse – A nurse is able to see details, such as questions and answers that are meant for only her.
4. Retrieval and answering of questions (Answered and unanswered) – Specialist authentication is also successful, and grants permissions to a specialist. The specialist is able to view all the questions that

have been asked by community health nurses and answer them or make additional comments.

5. Uploading of educational materials – A specialist is able to upload images successfully to the database for nurses to view on their home screen when a connection is established with the server.

5 Conclusions and Recommendations

The objective of this project was to improve community healthcare delivery in communities in Ghana through a mobile application. To achieve this, an android based application was built to provide community health nurses and effective collaboration medium between them and specialists. The application provides nurses with clinical support that integrates a knowledge base to which they could refer in challenging situations. The solution is quite similar to existing solutions around the world, but peculiar in its real time communication and collaboration with specialists feature.

5.1 Application Benefits

1. Access to clinical advice: The application would help community health nurses in remote areas to have access to clinical advice from medical specialists.
2. Access to better healthcare: Community health nurses would be well equipped with information on diseases and treatments. This would allow members of the community to have better access to healthcare.

3. Convenience: The application would be convenient for the members of the community as an increased confidence in community health nurses would encourage them to seek the medical advice before travelling to hospitals in the urban areas if necessary.
4. Healthier communities: Community health nurses would provide better healthcare to the communities with the support of the application, and would also be able to give community members better health education as a result of the informative educational materials that the application presents them.
5. Professional development: The application would enable community health nurses to develop their professional skills while in the rural areas as they would be receiving educational materials from administration and specialists.

5.2 Future Work

The impact on healthcare delivery as a result of the application could be improved immensely by integration this application with existing system used by existing projects which are provided by third – parties. An example of the existing third parties includes Medical Aid Films, which provides innovative education and training through films to save lives of women and children in developing countries. Community health nurses could refer to video content provided by Medical Aid Films to make clinical decisions and to train and educate themselves. Another software integration or partnership that could be forms is with Physicians Interactive has a large repository of resources for healthcare information

and mobile decision support tools and provides this to medical professionals everywhere.

Bibliography

Adu, T. (2012, March 15). Challenges Community Health Nurses Face. (E. G. Sampaney, Interviewer)

Ghana Health Service. (2010). The Health Sector in Ghana: Facts and Figures. Accra: Ministry of Health.

Health eVillages. (2012). Chronicles of Uganda and Kenya. Retrieved March 26, 2013, from Health eVillages:
http://healthevillages.org/wp-content/themes/foundation/images/HeV_Brochure_R2_6-4-12-1.pdf

Healthcare IT News. (2012, February 15). Clinical Decision Support (CDS). Retrieved March 20, 2013, from Healthc IT Index:
<http://www.healthcareitnews.com/directory/clinical-decision-support-cds>

International Data Corporation. (2012, December 4). Worldwide Mobile Phone Growth Expected to Drop to 1.4% in 2012 Despite Continued Growth Of Smartphones, According to IDC. Retrieved March 25, 2013, from
<http://www.idc.com/getdoc.jsp?containerId=prUS23818212#.UVF3qwxqxy>

Nyonator, F. K., Awoonor-Williams, J. K., Phillips, J. F., Jones, T. C., & Miller, R. A. (2005). The Ghana Community-based Health Planning and Services Initiative for scaling up service delivery innovation. *Oxford Journals*, 25-34.

Physicians Interactive. (2011). About Us. Retrieved March 25, 2013, from Physiciaians Interactive :
<http://www.physiciansinteractive.com/about/>

- Regional Health Directorate. (2009). Report On Upper East Regional Retreat Of Health Managers On Health PoW Implementation Challenges/Issues. Ghana Health Service.
- Skillsoft. (2012). Five Calls to Make When Developing a Mobile Learning Strategy . Retrieved from <http://lib.trinity.edu/research/citing/apaelectronicssources.pdf>
- Skyscape. (2013). Free Apps. Retrieved March 26, 2013, from Skyscape: <http://www.skyscape.com/Apps/Default.aspx>
- United Nations. (2012). The Millennium Development Goals Report. New York: United Nations.
- Vital Wave Consulting. (2009). mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World . Washington, D.C. and Berkshire, UK: UN Foundation-Vodafone Foundation Partnership.
- Wingenious. (2005). Database Architecture. Retrieved March 20, 2013, from wingenious: <http://www.wingenious.com/database.pdf>
- Yakuba, F. (2012, October 12). African voices: community health in northern Ghana. Retrieved from <http://www.one.org/international/>
- Youn, S. (2012, February 3). Primary Healthcare Nursing Promotion Program. Retrieved March 28, 2013, from Center for Health Market Innovations: <http://healthmarketinnovations.org/program/primary-healthcare-nursing-promotion-program>

Appendix

Code implementations specific to Android Interface

```
public class ParseJSON {  
    ArrayAdapter<String> adapter;  
    ArrayList<String> arrayItem;  
    static InputStream is = null;  
    static JSONObject jObj = null;  
    static String json = "";  
  
    public JSONObject getJSONFromUrl(String url) {  
        // Making HTTP request  
        try {  
            // defaultHttpClient  
            DefaultHttpClient httpClient = new DefaultHttpClient();  
            HttpPost httpPost = new HttpPost(url);  
  
            HttpResponse httpResponse = httpClient.execute(httpPost);  
            HttpEntity httpEntity = httpResponse.getEntity();  
            is = httpEntity.getContent();  
  
            } catch (UnsupportedEncodingException e) {  
                e.printStackTrace();  
            } catch (ClientProtocolException e) {  
                e.printStackTrace();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
  
        try {  
            BufferedReader reader = new BufferedReader(new  
InputStreamReader(  
                is, "iso-8859-1"), 8);  
            StringBuilder sb = new StringBuilder();  
            String line = null;  
            while ((line = reader.readLine()) != null) {  
                sb.append(line + "\n");  
            }  
        }  
    }  
}
```

```

    }
    is.close();
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " + e.toString());
}
// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " + e.toString());
}
return jsonObj;
}
}

```

Code 1: JSON Parser class in Android

```

class LoadAllAnswers extends AsyncTask<String, String, String> {
protected String doInBackground(String...args){
try {
//HashMap for ListView
arrayListAnswers = new ArrayList<HashMap<String, String>>();

JSONObject jsonObject = jsonParser.getJSONFromUrl(url);
//Getting array of answerDetails
answerDetails = jsonObject.getJSONArray(TAG_ANSWERDETAILS);

//looping through all answerDetails
for (int i=0;i<answerDetails.length();i++){
    JSONObject a = answerDetails.getJSONObject(i);

//Storing each json item in variable
String question=a.getString(TAG_QUESTION);
String time=a.getString(TAG_TIME);
String answer = a.getString(TAG_ANSWER);
String spec_name = a.getString(TAG_NAME);

```

```
String answer_time = a.getString(TAG_ANSWER_TIME);

    // creating new HashMap
    HashMap<String, String> map = new HashMap<String, String>();

    // adding each child node to HashMap key => value
    map.put(TAG_QUESTION, question);
    map.put(TAG_TIME, time);
    map.put(TAG_ANSWER, answer);
    map.put(TAG_NAME, spec_name);
    map.put(TAG_ANSWER_TIME, answer_time);

    // adding HashList to ArrayList
    arrayListAnswers.add(map);
}
}
```

Code 2: Implementation of JSON Parser Class in an AsyncTask in Android

Code implementations specific to Web Interface

```
$(document).on("click", '.Answer', function()) {  
    //use regex to obtain digits in id of Answer link clicked on  
    var re = /\d{1,2}/;  
    var m = re.exec(event.target.id);  
    var forms = document.getElementsByTagName("form");  
    /* iterate through the form elements to find a match in the form id and the  
    Answer link id  
    to generate the specific form for that context */  
    for (var i = 0; i < forms.length; i++) {  
        if (forms[i].className == 'Answer_'+m) {  
            //hide/show answer form  
            $(forms[i]).toggle();  
            qid =  
$(this).parent().parent().parent().siblings(".objs").children("li#question_id");  
            var submit =  
$(this).parent().parent().parent().children().children().children('input#formBut  
ton');  
            $(submit).click(function()) {  
  
                //get text from text area  
                text =  
$(this).parent().parent().parent().children().children('textarea#formtxtarea').v  
al();  
                var qid1=qid.val();  
                alert(qid1+"yes");  
  
                //send data obtained over the server and insert into the database  
                $.ajax({  
                    url:  
"http://localhost/community/answerQs.php?cmd=4&answer="+text+"&question_  
id="+qid1,  
                    type: "POST",  
                    contentType:'text',  
                    data: {answer:text,question_id:qid1},
```

```
        dataType:'text'  
    });  
    alert("answer submitted");  
});  
}  
}  
});
```

Code 3: Code showing extraction of a field id from an array of elements and passing values to an Ajax method for processing

```

$.getJSON("http://localhost/community/fetchQnAs.php?cmd=1",
  function(data){
    $.each(data.QnAs, function(i,question){
      if (currId == question.QUESTION_ID){
        count++;
      }
      if (count > 1){
        currId = question.QUESTION_ID;
        count--;
        $("#data").append('<ul class="well" style="margin-left:0px;">'+
          '<li class="ANSWER">'+ question.ANSWER +'</li>'+
          '<li class="SPEC_ID">'+ question.SPEC_NAME +'</li>'+
          '<li class="ANSWER_TIME">'+ question.ANSWER_TIME + '</li>'+
          '</ul>');
      }
      else{
        currId = question.QUESTION_ID;
        $("#data").append('<ol id="QnAs">'+
          '<ul class="objs" style="margin-left:0px;">'+
          '<li class="QUESTION" id="question_' + i + '">' + question.QUESTION +
          '</li>'+
          '<li class="QUESTION_ID" id="question_id" value=' +
question.QUESTION_ID + '></li>'+
          '<li class="FIRSTNAME" id="firstname">' + question.FIRSTNAME +
          '</li>'+
          '<li class="LASTNAME" id="lastname">' + question.LASTNAME + '</li>'+
          '<li class="TIME" id="time">' + question.TIME + '</li>'+
          '</ul>'+
          '<ul class="ulToggle" style="display: inline-block;margin-left:0px;" >'+
          '<li><a href="#"><span style="display: inline-block; float: left;"
class="Answer" id="Answer_' + i + '">Answer</span></a></li>'+
          '<form style="display:none;margin-left:0px;" class="Answer_' + i + '"
id="form" method="POST" action="" value="form">'+
          '<textarea id="formtxtarea" style="margin-left: 0px; margin-right: 0px;
min-width: 568px;max-width: 568px;min-height:110px;max-
height:110px;"></textarea>'+
          '<div class="btn-group" data-toggle="buttons-checkbox"
style="display:inline-block;">'+

```

```

'<button class="btn btn-primary" id="genBtn">General</button>' +
'<button class="btn btn-primary" id="nutBtn">Nutrition</button>' +
'<button class="btn btn-primary"
id="phBtn">Pharmaceutical</button></div>' +
'<span><input id="formButton" style="float:right;" class="btn"
type="submit" value="Submit Answer"></input></span>' +
'<br></form>' +
'<br></ul>' +
'<ul class="well" style="margin-left:0px;">' +
'<li class="ANSWER">' + question.ANSWER + '</li>' +
'<li class="SPEC_ID">' + question.SPEC_NAME + '</li>' +
'<li class="ANSWER_TIME">' + question.ANSWER_TIME + '</li>' +
'</ul>' +
'</ol>');
}
});
});

```

Code 3: Implementation of JSON Parser in JQuery

```

function fetchJSONquestions(){
    $str_query=("SELECT QUESTION_ID, QUESTION, FIRSTNAME,
    LASTNAME, questions.TIME
    FROM questions, community_health_officer
    WHERE community_health_officer.NURSE_ID =
questions.NURSE_ID");

    $exec =$this->query($str_query);

    if($this->get_num_rows($exec)){
        while($row[]=$this->fetch($exec)){
            $value['QnAs']=$row;
        }
        $encode = json_encode ($value);
    }
    return $encode;
}

```

Code 5: Implementation of json_encode on a select query