



California State University, San Bernardino
CSUSB ScholarWorks

Electronic Theses, Projects, and Dissertations

Office of Graduate Studies

3-2017

ANDROID MOBILE APPLICATION FOR HOSPITAL EXECUTIVES

Vihitha Nalagatla

California State University - San Bernardino, 005086688@coyote.csusb.edu

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd>



Part of the [Data Storage Systems Commons](#), and the [Other Computer Engineering Commons](#)

Recommended Citation

Nalagatla, Vihitha, "ANDROID MOBILE APPLICATION FOR HOSPITAL EXECUTIVES" (2017). *Electronic Theses, Projects, and Dissertations*. 435.

<https://scholarworks.lib.csusb.edu/etd/435>

This Project is brought to you for free and open access by the Office of Graduate Studies at CSUSB ScholarWorks. It has been accepted for inclusion in Electronic Theses, Projects, and Dissertations by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

ANDROID MOBILE APPLICATION FOR
HOSPITAL EXECUTIVES

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Vihitha Nalagatla
March 2017

ANDROID MOBILE APPLICATION FOR
HOSPITAL EXECUTIVES

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Vihitha Nalagatla
March 2017

Approved by:

Owen J. Murphy, Advisor, School of Computer Science and Engineering

Ernesto Gomez, Committee Member

Qingquan Sun, Committee Member

© 2017 Vihitha Nalagatla

ABSTRACT

Hospitals are the largest and most complex organizations where health care is provided. Safe and effective patient care services in hospitals depend on the efficient decisions made by hospital executives. The main task of hospital executives is to ensure the hospital can provide high quality patient care and services.

“Android Mobile Application For Hospital Executives” is an Android application used for displaying hospital performance metrics on a daily basis. This application allows hospital executives to review and monitor hospital operational data with ease of access and in a portable manner. Thus, reducing the effort of the hospital executives to perform their tasks.

.

ACKNOWLEDGEMENTS

I take this opportunity to thank everyone who has supported me during my Masters. I am grateful to my advisor Dr. Owen J. Murphy for believing and encouraging me and for his strong involvement in this project. I would like to thank Dr. Ernesto Gomez and Dr. Qingquan Sun for their suggestions and support.

I am indebted to my family for their unconditional support and belief which gave me leverage to accomplish my goals. I wish to express my special gratitude to Prime Healthcare Services for making this project happen by providing essential requirements.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
LIST OF FIGURES	ix
CHAPTER ONE: INTRODUCTION	
Background	1
Purpose	2
Existing System	3
Problems in the Existing System	3
CHAPTER TWO: SYSTEM ANALYSIS	
Proposed System	4
Hospital Performance Metrics in the Mobile Application	5
Admit	5
Discharge	5
Census	5
Insurance	5
Demographics	5
Length of Stay	5
Diagnosis	5
System Requirement Specification	6
Software Requirements	6
Hardware Requirements	6

CHAPTER THREE: NOTES ON HL7 ADT MESSAGE

What is the ADT Message?	8
Types of ADT Messages	8
ADT^A01 Admit Message	8
ADT^A03 Discharge Message	8
ADT^A04 Register Message	9
Structure of HL7 ADT Message	9

CHAPTER FOUR: SELECTED SOFTWARE

What is Android?	10
RESTful Web Service	11
JSON	12
PHP	13
Package Definitions Used in Project	13
import android.os.AsyncTask	13
import org.json.JSONArray	13
import org.json.JSONObject	13
import java.net.HttpURLConnection	14
import android.widget.DatePicker	14
import android.content.Intent	14
import java.io.InputStreamReader.....	14
import android.widget.TextView	14
import android.widget.ImageView	15

import android.widget.Toast	15
CHAPTER FIVE: SYSTEM IMPLEMENTATION	
Implementation Overview	16
Steps Involved in Backend Implementation	18
Decoding HL7 ADT Test Messages	18
PHP Files	20
Deploying PHP Files	21
CHAPTER SIX: APPLICATION CODES	
Mobile App	23
.idea	23
Gradle	23
External Libraries	24
Project Module (app)	24
MainActivity.java	25
Login.java	25
Metrics.java	25
Admit.java	25
Discharge.java	25
Census.java	25
Insurance.java	26
Length.java	26
Demographics.java	26

Diagnosis.java	26
CHAPTER SEVEN: SYSTEM TESTING	
Testing	30
Unit Testing	30
Module Testing	31
Integration Testing	31
System Testing	31
User Acceptance Testing	31
Testing Performed For My Application	31
CHAPTER EIGHT: OUTPUT SCREENSHOTS	33
CHAPTER NINE: FUTURE ENHANCEMENT.....	39
CHAPTER TEN: CONCLUSION	40
REFERENCES.....	41

LIST OF FIGURES

Figure 1. RESTful Web Service	12
Figure 2. Implementation Overview	16
Figure 3. Welcome Screen	33
Figure 4. Login Screen	33
Figure 5. Login Error Screen	34
Figure 6. Metrics Screen	35
Figure 7. Admit Screen	35
Figure 8. Discharge Screen	36
Figure 9. Census Screen	36
Figure 10. Insurance Screen	37
Figure 11. Demographics Screen	37
Figure 12. Length of Stay Screen	38
Figure 13. Diagnosis Screen	38

CHAPTER ONE

INTRODUCTION

Background

Hospital executives need to monitor hospital performance metrics on a daily basis. Such metrics include hospital occupancy i.e. daily census, surgery schedule for the day, staffing requirements and patient care metrics like the length of stay of each patient, diagnosis details, patient demographic distribution, etc. The hospital executives review such metrics through reports generated on a daily basis either on paper or on their desktops through proprietary software.

As a consequence of The Health Information Technology for Economic and Clinical Health (HITECH) Act [6] and The Patient Protection and Affordable Care Act (PPACA) [1], all the health care providers including hospitals have implemented Electronic Health Records (EHR) [10]. Furthermore, these regulations require under “meaningful use” standards interoperability and electronic exchange of health data through standard interfaces.

The interface for health data exchanges utilizes Health Level 7 (HL7) data set. HL7 is a US-based, ANSI-accredited health information development standard [7]. HL7 Standard helps to improve better hospital administration, record transfer and maintenance and enables the exchange of information between medical software applications. Therefore HL7 interfaces containing most of the above mentioned hospital metrics is ubiquitously available in all

hospital EHRs. HL7 interfaces contain different types of messages such as Order Messages (ORMs), Admission Discharge Transfer (ADT), Detail Financial Transactions (DFTs), etc.

This mobile application utilizes ADT test feed as input because using live patient data affects patient privacy under The Health Insurance Portability and Accountability Act (HIPAA) [9] and displays hospital operational data in an android mobile platform using Representational State Transfer (REST) web services. The main significance of the project is the mobile application developed in association with Prime Healthcare services [3] using HL7 ADT test feed data as the input for this application.

Purpose

The main purpose of this project is to develop an Android mobile application that helps hospital executives monitor and review hospital performance metrics on a mobile platform rather than using reports generated on a daily basis either on paper or on their desktops through proprietary software. The mobile app is mainly geared to the hospital executives and can be used by all hospital management personnel including hospital staffing coordinators and directors. Hospital executives can quickly and easily access the information and time can be saved.

Since all the healthcare providers, including hospitals are using EHRs due to the HITECH Act, eligible healthcare providers can get incentive payments

when they adopt certified EHR technology [6]. Hence, this mobile application will help in maintaining health records electronically in a portable manner using an android mobile platform.

Existing System

Problems in the Existing System

In the existing system, hospital executives will use either desktop or paper generated reports for reviewing the hospital performance metrics. It is extremely difficult to monitor the hospital operational data in a speedy and portable manner using the existing system.

CHAPTER TWO

SYSTEM ANALYSIS

Proposed System

The mobile application allows hospital executives to review the hospital performance metrics with ease of access. The mobile app will contain hospital operations data, including daily census, average length of hospital stay, patients admitted and discharged for a given day, patient insurance and diagnosis details and patient demographic distribution.

The mobile app will in part free the hospital personnel from the need for daily paper reports and the need for the hospital personnel to login to the proprietary software only available on their individual office desktops.

To use this mobile application the user needs to login with valid credentials. Improper or invalid authentication results in an error message. Successful login leads to the dashboard comprised of a series of icons for each of the above mentioned metrics. The user can easily view the required information just by tapping the metric icon. In order to review another metric, the user can go back to the metrics screen by using a back option. This mobile app contains refresh option in order to see updated information. Once the reviewing process has finished, the hospital executive can logout from the application using the logout option and will need to login whenever he/she wants to use the application again.

Hospital Performance Metrics in the Mobile Application

Admit

It gives information about number of patients admitted for a given day.

Discharge

Discharge metric provides discharged patient count details for a given day.

Census

The daily census of a hospital is the combination of number of patients admitted and registered for a given day and this information can be obtained using this metric.

Insurance

It consists of insurance details like name of the insurance company and number of patients using the corresponding insurance.

Demographics

This metric shows patient demographic distribution in the form of a bar graph which represents the percentage of patients from each county.

Length of Stay

The average length of patient stay in days i.e. how long the patient stayed in the hospital before he/she gets discharged from the hospital.

Diagnosis

It consists of information about patient diagnosed details such as the least and the most common diagnosed disease.

Analyzing information about the above mentioned metrics helps hospital executives making effective decisions. For instance, consider an organization having a chain of hospitals in each county. By analyzing the demographics of patient population for any given geographical area, the hospital administrators have the information to properly forecast events, and environmental changes to adjust the facilities' services. The changes permit supporting additional patients, patient beds, health care specialties, and hiring skilled clinicians targeted to support the patient's needs.

System Requirement Specification

Software Requirements

- Android Mobile Operating System: Version 4.4 or higher.
- IDE tools: Android Studio and Eclipse.
- SQL Server 2014 Management Studio.
- FileZilla FTP client.
- User Interface: XML.
- Code Behind: JAVA, XML, PHP, SQL and JSON.
- HL7 ADT Test feed.
- Microsoft Azure portal.
- Internet: Yes.

Hardware Requirements

- Android mobile: Version 4.4 or higher.

- RAM: 512 MB or higher.
- Device must be enabled for USB debugging.
- Processor: 1GHZ or higher.

CHAPTER THREE

NOTES ON HL7 ADT MESSAGES

What is the ADT Message?

Admission Discharge Transfer (ADT) messages are the most common messaging in the HL7 standard. Hospital Information System (HIS) initiates these messages to communicate with other hospital systems about trigger events such as information related to admit, discharge and transfer of a patient. These messages are also used to store diagnosis details, insurance details and patient demographics like patient name, address. It also helps to find outpatient visit information such as patient location, primary facility, attending doctor, etc.

Types of ADT Messages

51 types of ADT messages are used to represent various trigger events.

This mobile application makes use of 3 types of ADT messages. Those are

ADT^A01 Admit Message

This trigger event has generated when a patient is admitted to the hospital.

ADT^A03 Discharge Message

This trigger event has initiated when a patient is discharged from the hospital.

ADT^A04 Register Message

The hospital HIS system initiates A04 messages when a patient has registered as Outpatient within the system or checked into a specific visit.

Structure of HL7 ADT Message

ADT test message in HL7 standard format is represented as below:

```
MSH|^~\&|EPIC|WAMC|IGUANA||20160314092821|EDRN|ADT^A01|423||2.4
EVN|A01|20160314092821||ED_AFTER_ARRIVAL|EDRN^EMERGENCY^NUR
SE^^^^PHSOC^^^^WAMC
PID|1||800000812^^EPI^MR||YOUNG^ASAP||19701015|M||^L|||||770
0000037504|555-12-4652|||||||N
DG1|1||^brain injury | brain injury ||^10151;EPT
IN1|1||MEDICARE||||||20061115|20170316|| YOUNG^ASAP |Self||555 MAIN
ST^ONTARIO^CA^91761^US^^SAN
BERNAR|||1*1|||||||7|4522155||||||^L||BOTH
```

The HL7 ADT test message shown above contains information about the hospital performance metrics. It is very difficult to read these messages and obtain the required metric information. In this application, these messages are decoded into a meaningful format like tables in the database using Java programming language.

CHAPTER FOUR

SELECTED SOFTWARE

What is Android?

Android was developed by Rich Miner, Nick Sears, Andy Rubin and Chris White in Palo Alto, California in October 2003. On August 17, 2005, Google acquired Android and manufactured a mobile device which was powered by the Linux kernel, by a group of individuals called team Rubin. Google guaranteed to give an adaptable, upgradable system platform for users. It is around that period Android was upgraded, and Android is now an operating system developed for smart phones, regular PCs, tablets, and other electronics [8].

Android is now the most popular operating system on the market. According to 2013 sales, Android software has the highest number of sales when compared to iOS, Microsoft Windows, and Mac OS. Over one million Android applications have been released by Google play store and around fifty billion applications have been downloaded according to July 2013 census reports. Based on May 2013 survey, up to 71 percent of mobile developers are using the Android platform for building their applications [8].

The software tool like Android studio required for developing Android applications is an open source utility and freely available for download to the public. This played a key role in projecting Android popularity and dominance on its fellow competitors.

RESTful Web Service

Web Service is defined as the service provided by an electronic device in order to communicate with another electronic device through the World Wide Web. It includes Web technology such as the Hypertext Transfer Protocol (HTTP) and is mainly intended to provide communication between machines, such as transferring specific file formats like JSON and XML. In general, the Web service usually provides an object-oriented Web-based interface to a database server, used for example by another Web server, or by a mobile application, that provides a user interface to the end user [5].

Web services that use Representational State Transfer (REST) architecture are called RESTful web services. In practice, everything is a resource in REST architecture and is lightweight, highly scalable and maintainable. Uniform Resource Identifiers (URIs) are used to access the resources such as records of database in REST web services. It uses HTTP for communicating with clients like Android phones, etc. In order to represent the resources in different formats like XML, plain text, JSON, etc. resources are generally isolated from their original representation. JSON format was used in this application.

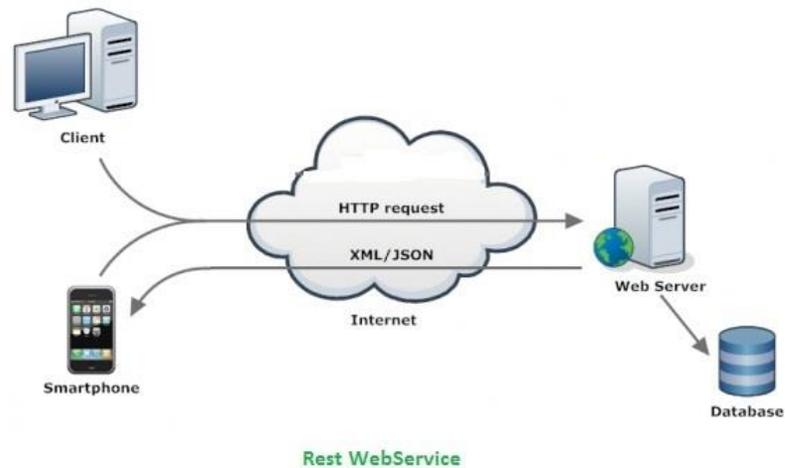


Figure 1. RESTful Web Service

JSON

JavaScript Object Notation (JSON) is a data exchange format [4]. It is lightweight and easy for Android applications to parse JSON values and display the results on the user interface. The JSON format is represented here.

```

{"dia1":[{"diagnosis_description":"Allergies","count_des":1},
{"diagnosis_description":"Amputation of leg below knee left traumatic subsequent encounter [S88 112D] ","count_des":3},
{"diagnosis_description":"appendicitis ","count_des":5}]}

```

The above JSON format consists of object dia1 which is an array that contains three objects. Each object has two records such as diagnosis_description and count_des.

PHP

Hypertext Preprocessor (PHP) is a scripting language used for developing dynamic web pages and Internet applications [12]. Files that have the .php extension are called PHP files and can contain text and scripts. These files are processed by a web server and then returned required values in the web page. In this application, PHP files are used to generate a JSON representation of values that are fetched from the database using Sequential Query Language (SQL).

Package Definitions Used in Project

Below are some of the important packages that are mainly used to run this application. These packages also contain class files that are used to implement this application.

import android.os.AsyncTask

It helps to perform operations in the background and allows simple use of the user interface without having to manipulate threads.

import org.json.JSONArray

It contains information about the JSON array which consists of a dense indexed sequence of values.

import org.json.JSONObject

It is responsible for holding information about a JSON object which is a modifiable set of name or value mappings and needs to be parsed in order to

display on the user interface.

import java.net.HttpURLConnection

URL Connection class allows communicating with the URL over the network. It helps the client such as an android application to read JSON data from a URL using HTTP connection.

import android.widget.DatePicker

It provides a widget for selecting a date called as date picker. The date can be selected using year, month and day present in the calendar view of date picker.

import android.content.Intent

It holds the responsibility of connecting different activities present in the application such as binding between login and metrics screen and so on.

import java.io.InputStreamReader

It is used to read data present in bytes and decode the data into the platform's default charset or specified charset. In this application, JSON data is read and converted into string format in order to update the user interface.

import android.widget.TextView

It handles the displaying of text to the user using text view and also allows the user to copy the text view values. It does not allow editing text. However, subclass edit text can be used for editing the text such as entering a password, etc.

import android.widget.ImageView

It helps to display arbitrary image, such as an image icon. This class can also load images from resources or content providers, etc.

import android.widget.Toast

It contains the toast class that provides a view containing a quick little message for the users. The duration of the message can be adjusted using short and long term functions. In this application, the toast class is used to display login error messages.

CHAPTER FIVE

SYSTEM IMPLEMENTATION

Implementation Overview

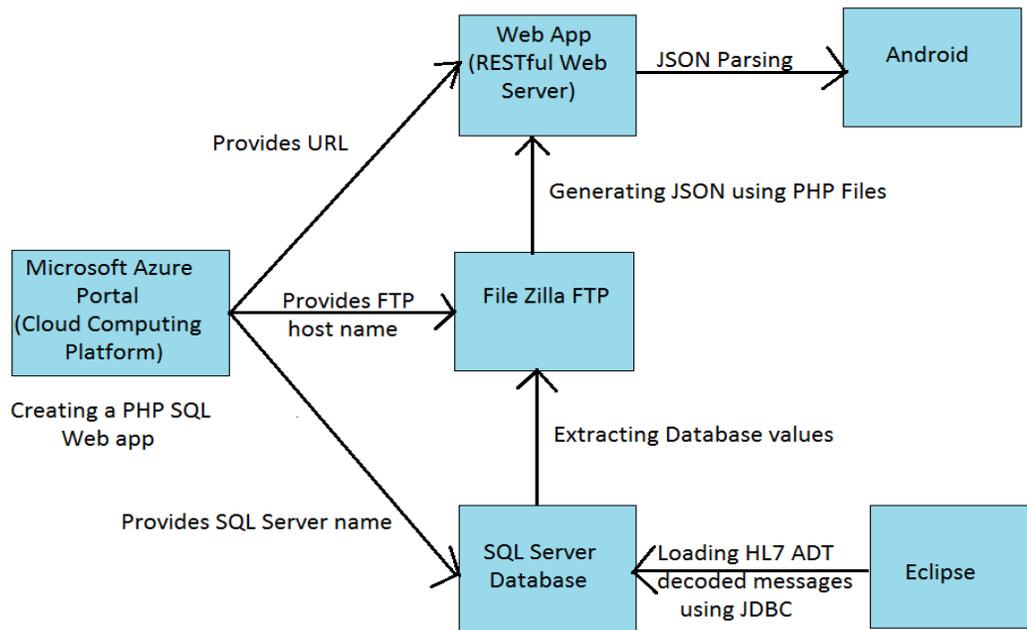


Figure 2. Implementation Overview

In this application, HL7 ADT test messages need to be decoded into a meaningful format and stored into the SQL database using Java programming language and Java Database Connectivity (JDBC) connection through the Eclipse IDE tool.

Android applications cannot fetch data directly from the database. So in order to get the data, an application is needed to connect to the outside REST web services. This application makes use of the Microsoft Azure Portal, which is a cloud computing platform developed by Microsoft for creating, publishing, and managing various applications and services using a global network of Microsoft-managed data centers [11]. The primary advantage of using a cloud computing platform allows users to access the data remotely rather than through a local host. This application requires creating a PHP SQL Web app which acts as the REST web service. In Practice, the developer needs to pay for building the web app using Microsoft Azure. As I developed this application in association with Prime Healthcare Services, they have provided me the credentials for the PHP SQL Web App, which can be deployed using the File Transfer Protocol (FTP).

Those credentials include server name for the SQL server Management studio, which stores decoded HL7 ADT test messages in tables, Uniform Resource Locator (URL) for the web app which holds records from the database in JSON format. The host name for FileZilla FTP is used for deploying PHP files to the web app in order to generate a JSON representation of values that are fetched from the database.

The next step is to connect the Android application to the web app for accessing the JSON data and updating the user interface. This can be achieved by using asynchronous task (Async task) in Android.

Steps Involved in Backend Implementation

Decoding HL7 ADT Test Messages

Below is the Java code for decoding HL7 ADT test messages and loading into the SQL database using the Eclipse IDE tool.

```
public class HL7_decode {  
  
    // Adding database credentials for jdbc connection  
  
    static String db = "database name created in sql server";  
  
    static String url = "jdbc:sqlserver://servername;databaseName="+db+"";  
  
    static String userid = "username";  
  
    static String password = "password";  
  
    static Connection con = null ;  
  
    public static void main(String args[]) throws SQLException,  
        ClassNotFoundException,IOException{  
  
        // Connecting to database using jdbc  
  
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");  
  
        con = DriverManager.getConnection(url, userid, password);  
  
        // Declaring required fields of ADT test message, here declared only one field  
  
        String diagnosis_description = "DG1-4";  
  
        // Reading ADT test messages  
  
        BufferedReader br = new BufferedReader(new FileReader(" path of ADT test  
messages.txt"));
```

```

String line = ""; int i = 0, j = 0, new_line = 0, msg_count = 0;

// Logic for inserting values into database tables after reading ADT test feed
while ((line = br.readLine()) != null)
{
j++; String[] fields = line.split("\\");
if (fields[0].equals("MSH"))
if (msg_count > 0 || (fields[0].equals("MSH") && fields[8].substring(4,
7).equals("A01/A03/A04")) {
switch (fields[0]) {
case "MSH":  i++;
msg_count++;
if (msg_count != 1){
String inst = "INSERT INTO [dbo].[table name]([DIAGNOSIS_DESCRIPTION]) "
+"VALUES('"+diagnosis_description+"')";
Statement s = con.createStatement();
s.executeUpdate(inst);}
new_line = 1;
break;
case "DG1":  i++;
diagnosis_description = fields[4];
break;
default:

```

```
}}
```

```
}}
```

PHP Files

PHP files consist of SQL queries for obtaining required data from the database. It uses the `json_encode` function for returning JSON representation of values that are fetched from the database using an SQL query. Generally, execution of PHP files requires a web server like Tomcat or Apache. A web app was created using cloud computing platform acts as the default web server. Hence this application only requires FTP FileZilla, which uses the File Transfer Protocol, in order to deploy the PHP files. Each metric of this application requires a separate PHP file and after deployment, it can be accessed by using a URL. For instance, consider the diagnosis metric. The PHP code for this metric is shown below:

```
<?php
// Returns an array named dia1 into web page
$return_arr = array();
$return_arr['dia1'] = array();
try {
// Adding database credentials
$serverName = "tcp:server name";
```

```

$connectionOptions = array("Database"=>"database name",
"Uid"=>"username", "PWD"=>"Password");

// Connecting to sql server database

$conn = sqlsrv_connect($serverName, $connectionOptions);

// sql query for fetching values from database here diagnosis metric details

$sql = "select count(DIAGNOSIS_DES) as 'COUNT_DES', DIAGNOSIS_DES
from dbo.PATIENT1 GROUP BY DIAGNOSIS_DES"

$getProducts = sqlsrv_query($conn, $sql);

while($row = sqlsrv_fetch_array($getProducts, SQLSRV_FETCH_ASSOC)){
$row_array['COUNT_DES'] = $row['COUNT_DES'];

$row_array['DIAGNOSIS_DES'] = $row['DIAGNOSIS_DES'];

array_push($return_arr['dia'],$row_array);}

// Returns json representation of database values in an array named dia

echo json_encode($return_arr);

sqlsrv_free_stmt($getProducts);

sqlsrv_close($conn); }

?>

```

Deploying PHP Files

PHP files need to be deployed to the web app for generating a JSON array using FTP FileZilla. Then the Android applications are able to access the

JSON data by parsing and displaying the values in the user interface. Once the deployment has completed, the JSON array will be generated.

Similarly, JSON values need to be generated for other metrics also using PHP files. Therefore, PHP files of each metric are deployed separately using FTP FileZilla. The developer can view the generated JSON on the web app just by entering the “URL/filename.php”. For instance, entering URL/admit.php in the web browser allows the developer to view the required web page which consists of admit information in JSON format.

CHAPTER SIX

APPLICATION CODES

Mobile App

The main folder of this application is Mobile App, which consists of the below sub-folders:

- .idea
- Project Module (app)
- Gradle
- External Libraries

.idea

The .idea folder contains project settings in the form of configuration files (.xml). Each xml file is comprised of only a portion of configuration data related to the particular functional area which is defined by the name of the file such as compiler.xml, modules.xml, etc. These files contain information about the names and locations of the project component modules, compiler settings, etc. Almost all files under .idea contain information about the project itself.

Gradle

The Gradle folder consists of a gradle build system's jar wrapper and this jar is used to communicate Android studio with gradle installed in the operating system. The windows operating system is used for this application.

External Libraries

External Libraries is not actually a folder but it contains information about Referenced Libraries and the targeted platform of SDK in Android studio. In my application Android API 23 is used as the targeted platform.

Project Module (app)

The app folder is the actual project folder which contains project application code in the form of java files. The application folder contains build, libs and src sub-folders.

build is the directory which stores the complete output of the make process such as classes.dex, compiled classes and resources, etc. The libs folder consists of libraries and .jar files. The src folder contains android application code in the main subfolder and the unit test script in the android Test subfolder. The main folder again contains two subfolders java and res.

The res folder contains sub folders like drawable, layout, mipmap, etc. The layout folder includes xml files which are mainly used to create the user interface. Each layout will have a corresponding java class. The mipmap folder is used for placing the app icons which can be seen on the application dashboard.

The java folder takes care of all the java codes including methods developed by the application developer, and these java classes act as backend code for connecting the user interface layouts present in res, and also obtaining values from the RESTful web services. The java files in my application are as follows:

MainActivity.java

This class is used to develop the welcome screen of this application.

Clicking on the welcome screen directs the user to the login screen.

Login.java

This class is used to develop functions i.e. clicking on login button results in the metrics screen if valid credentials are entered and invalid authentication pops up an error message.

Metrics.java

Clicking on a particular metric icon present in the metrics screen directs the user to the corresponding screen. This function is implemented using this java class.

Admit.java

The Admit metric user interface is comprised of a date picker, and this java class helps the developer to get the patient admit count after selecting a particular date.

Discharge.java

This java class is used for displaying the number of patients discharged for a given day after selecting a date in the date picker calendar.

Census.java

The daily census count is displayed using this java class.

Insurance.java

A display of insurance company names and the number of patients in a list view can be implemented using this java class.

Length.java

The average length of patient stay can be shown on the text view using this java class.

Demographics.java

This class is used to draw a bar graph on the android canvas.

Diagnosis.java

This class is used to display diagnosed disease names on the text view.

The common interfaces and methods present in the above mentioned java classes are described below:

OnClickListener. Onclicklistener is an interface that defines an onclick method in which clicking on the certain text, button or image implements the defined onclick method.

Intent. Intent in Android is an abstract description of an operation to be performed. It is used to connect different activities.

In this application intent is used as an onclick method, clicking a button, image or text implements the intent method of entering into another activity. Java classes such as Admit, Discharge, Insurance, Census, Metrics, Length, Demographics and Diagnosis contain an onclicklistener in which clicking on the image view like back, refresh and logout directs to the corresponding screen.

The Login screen uses the button and the Welcome screen uses the text view.

The onclick listener along with the intent method is described below:

```
Image View iv1= (Image View) findViewById (R.id.image2);
iv1.setOnClickListener (new View.OnClickListener () {
@Override
public void on Click (View a) {
Intent b = get Intent ();
finish ();
start Activity (b);
}});
```

The above code implements an image view and then assigns the onclicklistener to the created view. When the user clicks, the onclicklistener will call the onclick function (intent method).

Asynchronous Task. It is used to perform operations in the background and publish results in the user interface. In order to get the values of a JSON array, Android applications need to connect to the RESTful web services by using HTTP URL connections. Async task helps in fetching the values from RESTful web services and updating the user interface. Below are the steps involved in Async task [2].

HTTP URL Connection. Connecting to RESTful web services using HTTP URL connection in Async task can be achieved by using the code below.

```
public class Http extends AsyncTask<Void, String, Integer> {  
  
    @Override  
  
    protected Integer doInBackground(String... params) {  
  
        InputStream input = null;  
  
        HttpURLConnection urlCon = null;  
  
        Integer result = 0;  
  
        try {  
  
            URL url = new URL(params[1]);  
  
            urlCon = (HttpURLConnection) url.openConnection();  
  
            urlCon.setRequestMethod("GET");  
  
            int statusCode = urlCon.getResponseCode();  
  
            if (statusCode == 200) {  
  
                Json parsing (output);  
  
                return result;  
  
            }  
  
        }  
  
    }  
  
}
```

JSON Parsing. This method is used to parse JSON data for obtaining required record values [4]. The code below is used to parse the JSON array.

```
private void json parsing (String Result) {  
    try {  
        JSONObject output = new JSONObject(Result);  
        JSONArray A = output.optJSONArray("name of the Json array");  
        for (int j = 0; j < A.length(); j++) {  
            JSONObject O = A.optJSONObject (j);  
            String rec = O.optString(" record name");  
        }  
    }  
}
```

Post Execute Method. The Post Execute method is used when the downloading of values from the web page has been completed and the user interface needs to be updated. Below is the code used for displaying the JSON value on the text view.

```
public void onPostExecute (Integer Result) {  
    if (Result == 1) {  
        Textview.setText(rec);  
    } else {  
        system.out.println("failed to fetch ");  
    }  
}
```

CHAPTER SEVEN

SYSTEM TESTING

Testing

Testing is an essential part of the project used for software quality assurance. Testing is the process of executing the application in order to find the software bugs. Testing includes designing different test case methods which are executed using a sequence of steps for the successful development of software.

Testing is a process of verifying and validating all units of software. The main purpose of the testing process is to make sure that the software functions correctly for a given set of possible data inputs. Testing allows the developer to discover errors present in the application and rectify those errors in advance so that users will not have a problem when using the application.

Different types of testing is needed to perform at different levels in order to ensure the quality of software.

Unit Testing

Unit testing is used for testing the design and logic of the software or application. This level of testing is used for ensuring the smooth flow of code, such as application flow in the main class between different methods, without any problem of identifying user input and for ensuring the proper function of the software/application.

Module Testing

Module testing allows the tester to test each component present in the application/software. For example, Module testing of a computer involves examination of the operating system, arithmetic and logic unit, keyboard, mouse, etc. in order to ensure each unit functions properly.

Integration Testing

Integration testing allows the developer to test the entire system after integrating each and every module of the application as a single system. This level of testing allows the developer to find a solution for all feasible bugs.

System Testing

System testing ensures that the application or software satisfies the system requirements after testing with all the anticipated user requirements.

User Acceptance Testing

User Acceptance testing is a critical level of testing in the testing process since it involves significant participation of the developer with end users in order to make sure that the software/application fulfills the functional requirements of the users, and also ensures whether the final developed software provides ease of access and control over the application to the users.

Testing Performed For My Application

I have tested my application at different levels by performing different tests that are mentioned above in order to verify whether my mobile application is functioning properly or not. Initially, I conducted unit testing to determine the

program flow of my application. Later I performed module testing which involves verifying each function individually to know its performance i.e. in my application, checking the functionality of each metric means whether it is properly fetching JSON data or not. After completion of writing the whole code, I tested whether all metrics in the application are displaying the same data as present in the database. Moreover, I have developed this project in association with Prime Healthcare Services where hospital executives have tested this application which meets their requirements.

I have tested the application several times after updating the database and it displayed only updated values. I have also tried to give invalid credentials but it does not accept and shows an error message. For better performance, I have performed all the above mentioned tests on my application at different levels.

CHAPTER EIGHT

OUTPUT SCREENSHOTS

Screen shots captured while testing this application at different levels which executed successfully without errors are shown below:

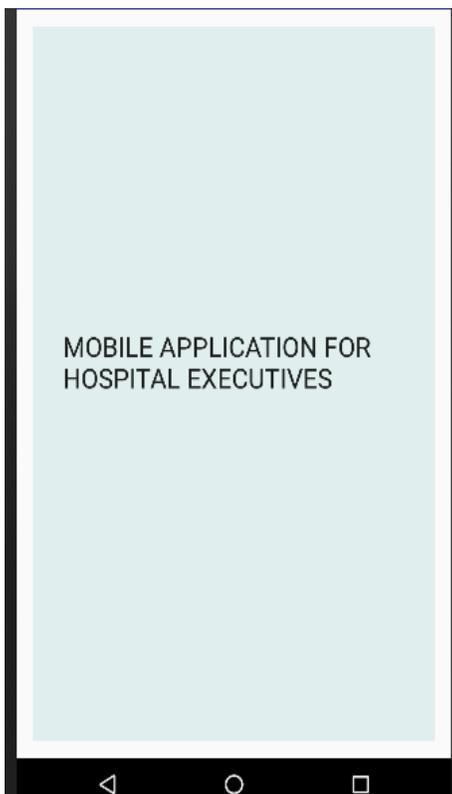


Figure 3. Welcome Screen

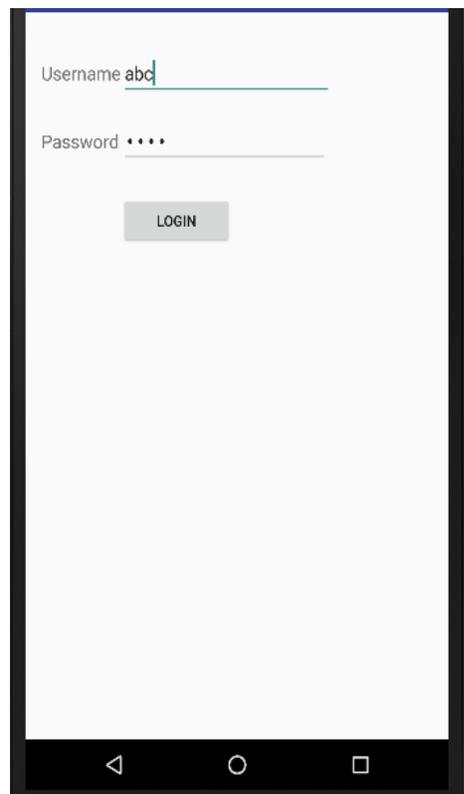


Figure 4. Login Screen

Figure 3: The above screenshot is a welcome page of this application, touching the screen results in the login page.

Figure 4: The above screenshot is a login page where the user needs to enter credentials for accessing the application. Successful authentication results in the metrics screen. An error message would pop up when the user enters invalid credentials as shown in Figure 5.

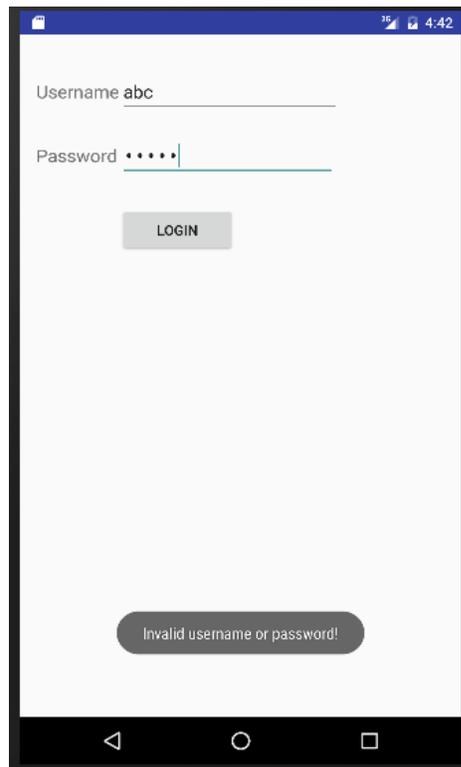


Figure 5. Login Error Screen

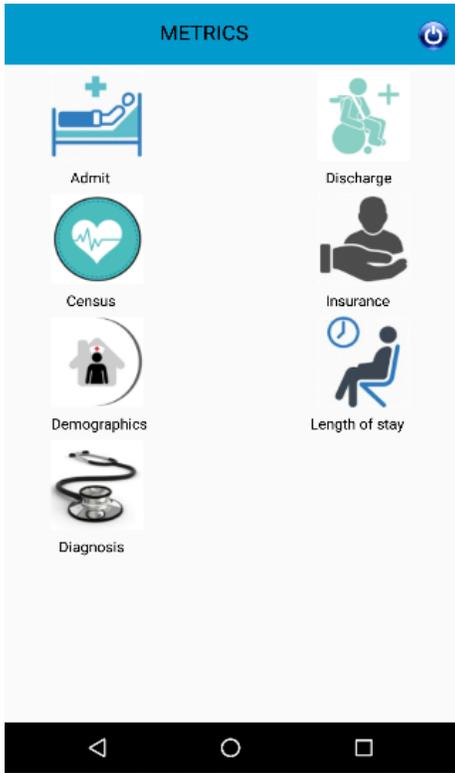


Figure 6. Metrics Screen

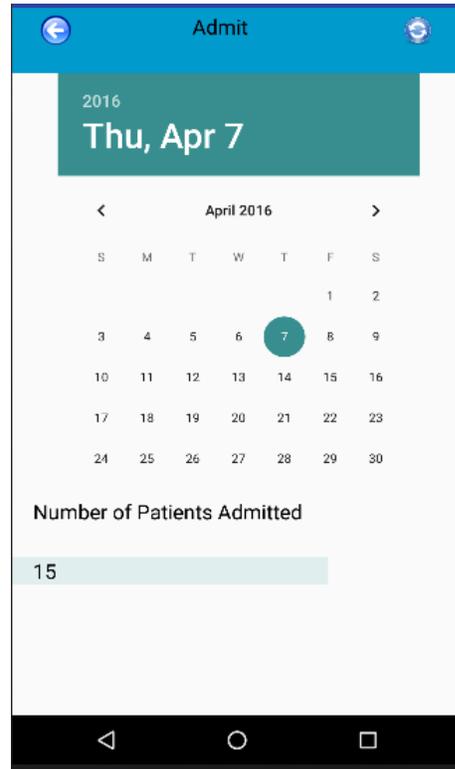


Figure 7. Admit Screen

Figure 6: This screenshot shows the metrics page. Here we can see hospital performance metrics, clicking on the particular metric results in the corresponding page.

Figure 7: Here we can see the admit screen where selecting a particular date displays the count of the patients admitted on that date.

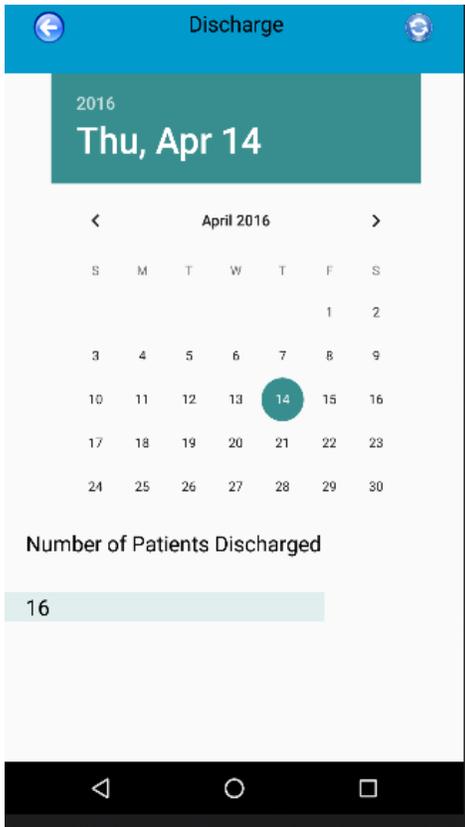


Figure 8. Discharge Screen

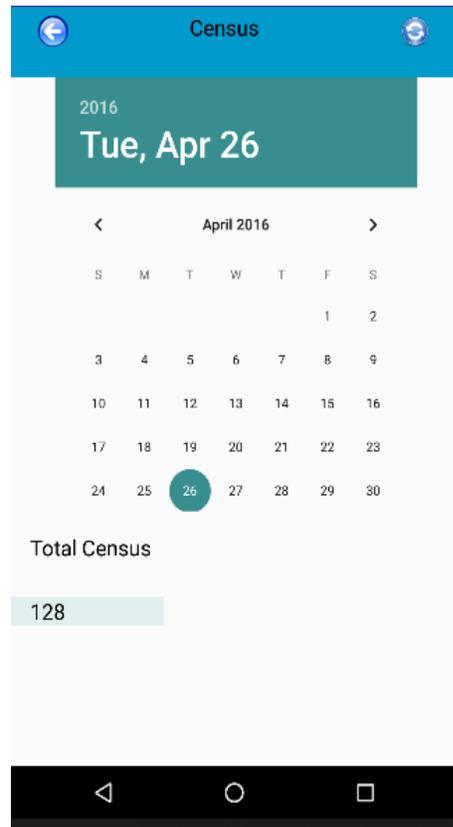


Figure 9. Census Screen

Figure 8: Shows the discharge screen in which selecting a date displays the number of patients discharged on that date.

Figure 9: This screen describes how the user can view the total census after picking a date from the date picker.



Figure 10. Insurance Screen

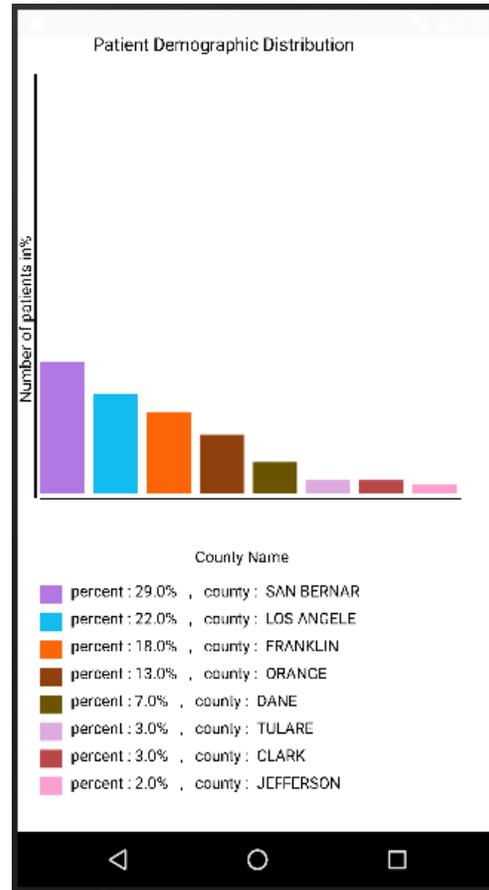


Figure 11. Demographics Screen

Figure 10: Shows insurance details consisting of insurance company names and number of patients.

Figure 11: Patient demographic distribution in the form of a bar graph indicates the County name and Percentage of patients as shown in the screenshot.

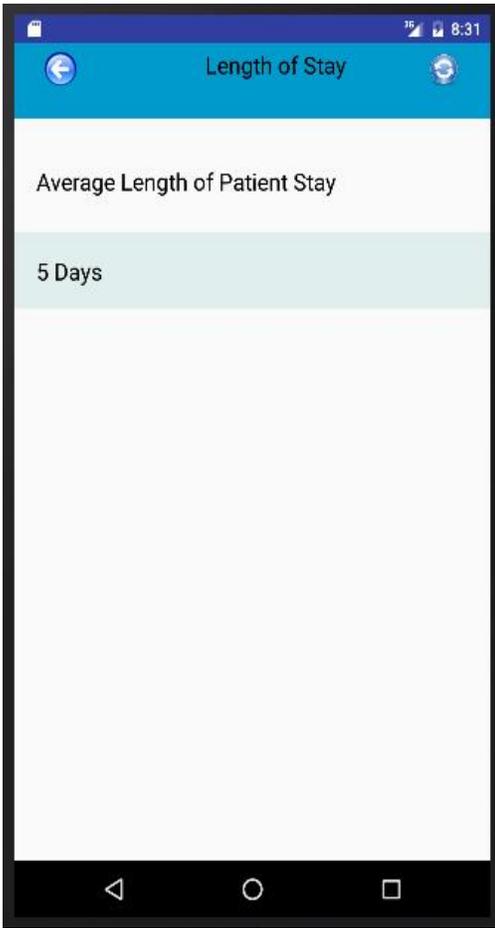


Figure 12. Length of Stay Screen

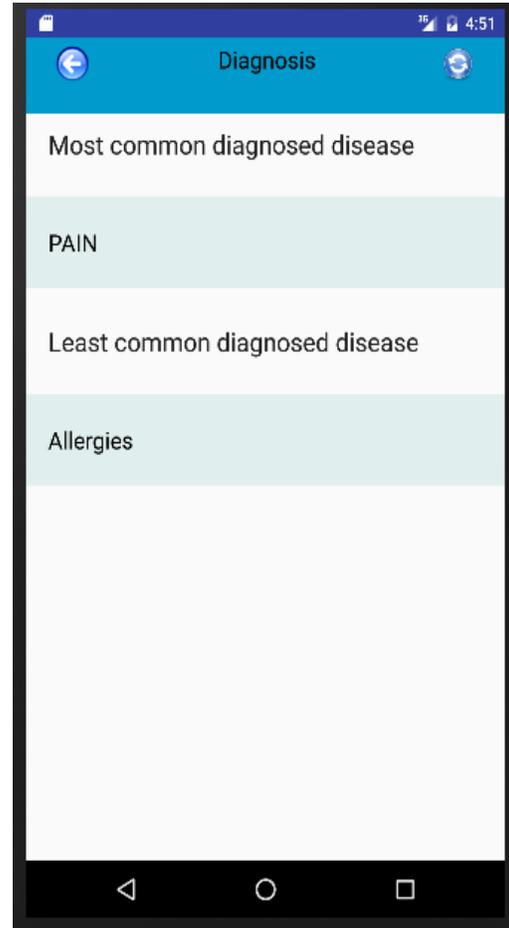


Figure 13. Diagnosis Screen

Figure 12: Here we can see the average length of patient stay in days.

Figure 13: The diagnosis details screen displays the least and most common diagnosed illness.

CHAPTER NINE

FUTURE ENHANCEMENT

This application can be enhanced using Interface Engines which store decoded HL7 messages. The Interface Engines can be mapped and synchronized with the database. The application can be further extended to display new metrics like surgery schedule, staffing requirement, etc. on the Android User Interface.

CHAPTER TEN

CONCLUSION

This application is a utility for hospital executives and helps them to review, monitor and analyze the hospital performance metrics on a daily basis such as daily census, average length of patient stay in the hospital, number of patients admitted and discharged for a given day, patient insurance details like insurance company name and number of patients using the corresponding insurance, diagnosis details like the least and the most common diagnosed diseases and patient demographic distribution like percent of patients coming from each county, without using paper-based generated reports and desktop reports.

This system can improve the efforts of hospital executives by providing hospital operational data in a portable manner with ease of access and helps them in making effective decisions required for improving the quality of patient care services.

REFERENCES

- [1] Patient Protection and Affordable Care Act - Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/Patient_Protection_and_Affordable_Care_Act.
(Visited on 01/10/2016).
- [2] AsyncTask.
<https://developer.android.com/reference/android/os/AsyncTask.html>.
(Visited on 03/20/2016).
- [3] Prime Healthcare
<http://www.primehealthcare.com/>. (Visited on 03/20/2016).
- [4] Android – JSON Parser Tutorial.
http://www.tutorialspoint.com/android/android_json_parser.html.
(Visited on 03/20/2016).
- [5] Web Service- Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/Web_service. (Visited on 03/20/2016).
- [6] J. Carlson, "Healthcare goes HITECH," *Baylor Business Review*, Sep. 2013
- [7] P. Chinniah and S. Muttan, "HL7 Standard based hospital information System," *International Journal of Biosciences & Technology (IJBST)*, Vol. 5, pp. 32-39, Sep. 2012.
- [8] Android (Operating System) - Wikipedia, the free encyclopedia.
[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)).

(Visited on 03/20/2016).

[9] Health Insurance Portability and Accountability Act (HIPPA) –

Wikipedia, the free encyclopedia.

[https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_](https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act)

[Act](https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act). (Visited on 01/10/2016).

[10] D. Blumenthal and M. Tavenner, “The Meaningful Use Regulation

for Electronic Health Records,” *The New England Journal of Medicine*,

Vol. 363, no. 6, pp. 501–504, Aug. 2010.

[11] Microsoft Azure - Wikipedia, the free encyclopedia.

https://en.wikipedia.org/wiki/Microsoft_Azure. (Visited on 03/20/2016).

[12] PHP - Wikipedia, the free encyclopedia.

<https://en.wikipedia.org/wiki/PHP>. (Visited on 03/20/2016).