Case Study

International Journal of Advanced Computer Research, Vol 8(34)
ISSN (Print): 2249-7277   ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2017.733025

Data exchange architecture for the development of mobile applications that support eHealth systems interoperability: a case of Tanzania

Frederick Chali[1*], Zaipuna O. Yonah[1] and Khamisi Kalegele [2]
Nelson Mandela African Institution of Science and Technology, Arusha, Tanzania[1]
Tanzania Commission for Science and Technology, Tanzania[2]

Received: 25-September-2017; Revised: 28-November-2017; Accepted: 02-December-2017

Abstract

Electronic Health (eHealth), especially in the form of electronic health records (eHRs), has proved to be helpful in health care services both in urban and remote areas. However, many of the systems are fragmented and work independently. Therefore, apart from their benefits, these systems are stand alone and as such they face some operational challenges. One of which is interoperability among the eHRs. Early efforts to interoperate eHealth systems, suggest that a single mobile application that can access multiple integrated eHRs is among the viable solutions on the way to eHRs interoperability. The proposed data exchange approach offers a mechanism to integrate a single mobile application with multiple eHRs while mitigating the interoperability problem. The design work builds on the results of a survey previously done to collect preliminary requirements of the data exchange system. System design techniques were used to design this system. In this paper, an architectural system design of an eHealth data exchange that can mitigate the interoperability challenge is presented.

Keywords

Multiple databases, Data exchange, eHealth records (eHRs), mHealth, Mobile app.

1.Introduction

The global health sector today is enjoying benefits enabled by ICT facilities especially in the form of eHRs. Many countries globally have reported to continue installing eHRs and some of them, creating an environment for their implementation. About 70% of the countries in the world today have national policies for eHealth or having strategies to implement such health information systems (HISs). As a result and collectively, the eHRs technology has proved to improve health care services greatly both in urban and rural or remote areas [1, 2]. However, most of these systems are not interconnected. They are fragmented and work independently. They do not exchange data and information among themselves, that is, they are not interoperable. Interoperability of the HISs is the capability of the systems to work together within and across organization boundaries [3]. Unfortunately, achieving interoperability of healthcare systems remain a daunting challenge [4].

Most of the eHRs, specifically those in developing countries, face the challenge of system fragmentation (i.e systems are not interoperable) due to the fact that individual system deployment is done by multiple system contractors and organizations [5, 6]. As a case study, Tanzania is among the countries facing both the fragmentation of eHRs and the interoperability challenges. Further, the government of Tanzania through Ministry of Health, community development, gender, elderly and children has identified eHRs interoperability as one of the challenges that are facing existing eHRs in Tanzania [7]. Hospitals and healthcare facilities in Tanzania are mostly using open source healthcare software. Examples of these systems/software are OpenMRS, OpenLMIS, Care2x, LIS, TIVA and CTC2. Unfortunately, these software are not interoperable [8]. Several benefits accompany the interoperability of eHRs. In [9], Zayas *et. al.*, list two general benefits of interoperability of eHRs as: (a) reduces the cost of healthcare, and (b) contributes into more effective and efficient patient care. Other benefits include: reduces the incidence of medical errors [10], provides timely access to the healthcare information and enable informed decision and personalized care [4], provides improved communication of healthcare by supporting

*Author for correspondence

continuity care [11]. In expanded form, interoperability is the ability of different ICT systems and software applications to communicate, to exchange data accurately, effectively and consistently and to use the information that has been exchanged [12]. Therefore, interoperability will only be achieved if the two parties exchanging information, understand well the meaning of information being exchanged and they are able to use them properly [13]. Based on this understanding, interoperability of eHRs is confirmed as the fundamental prerequisite for individual health care improvement [14]. But how does this challenge emerge? Some of the causes of eHealth interoperability identified by different researchers in the healthcare sector include the following:

i. Lack of an architectural guide/framework towards development of eHRs [3, 7].

ii. Lack of standard procedures to guide the lifecycle of eHRs [7, 8, 15].

iii. Poor collaboration among private companies or vendors in eHRs development [8], and

iv. Different data structures among the developed eHRs [3].

The general objective of the reported work is to support the on-going efforts to alleviate the interoperability problem of eHRs by designing and implementing a data exchange mechanism that leverages mobile devices deployed in healthcare. The main design principle is that of connecting a mobile application (mobile app) with multiple eHRs. The data exchange resides between multiple eHRs and a single mobile app where connected eHRs can exchange information. In this arrangement, the connected mobile app can access and transfer information to and fro eHRs. This paper presents the proposed design architecture for a data exchange component (DEC), as a way towards realizing interoperability of eHRs. The philosophy around the proposed DEC is that, it is designed to act as a central application/component that can provide a mechanism to make two or more eHRs to interoperate; i.e. to exchange data and information, in order to improve healthcare delivery and provide access to mobile apps that are used to access information to/fro the interoperable systems. The rest of the paper is arranged as follows: Section 2 provides materials and methods used to obtain results presented in this paper. Section 3 reports on results and alternative setups that could be adopted for DEC deployment and the other connected systems. Section 4, gives the proposed architectural design of the DEC while in Section 5 is DEC implementation and testing. A discussion is presented in Section 6 and conclusions are in Section 7.

2.Materials and methods

In [16] a survey method was used to identify challenges facing mobile app developers during the process of interconnecting multiple eHRs as a way toward alleviating the interoperability problem. Data were collected and analyzed through questionnaires, interviews and observations. The survey was conducted in Dar es Salaam, Dodoma and Arusha regions in Tanzania. The results of this survey revealed that, multiple database integration, user experience and mobile device capabilities, device fragmentation and platform considerations are some of the issues and challenges facing mobile app developers when dealing with the interoperability problem. The results of this study were considered as a basis for the design of the DEC. Details of the DEC architectural design are presented in section 4.

3.Results: alternative setups for DEC

This section presents three possible (and considered viable) setups showing alternatives on how the DEC and the participating eHRs could be installed. In general, the DEC is designed to run within a web server, such that both eHRs and DEC reside within servers. The setups are as follows:

A. Setup I: use of three servers

In *Figure 1*, it is proposed to utilize three separate servers where DEC is installed on a separate server. The two participating eHRs will reside separately in other two servers. A typical example in a real life situation is where the two eHRs represent health information systems (HIS) of two different hospitals. When the hospitals agree to exchange data generated and stored in their particular HISs, they will utilize the proposed central DEC to fulfil their goals. HIS from each hospital will remain where they reside i.e. within a particular hospital server, and then an additional server would be added to install DEC.

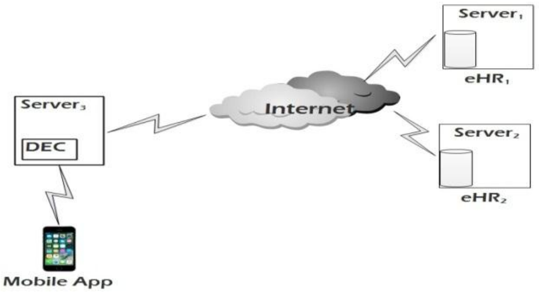


Figure 1 Setup I-three server setup for the DEC and participating eHRs

B. Setup II: use of two servers

The second considered setup is presented in *Figure 2*, which shows utilization of only two different/separate servers. This architecture requires DEC and one of the eHR to be installed in a single server and the second eHR installed in the second server. This second architecture allows use of only two separate servers managed by two hospitals, such that one of the hospitals is a host of DEC.

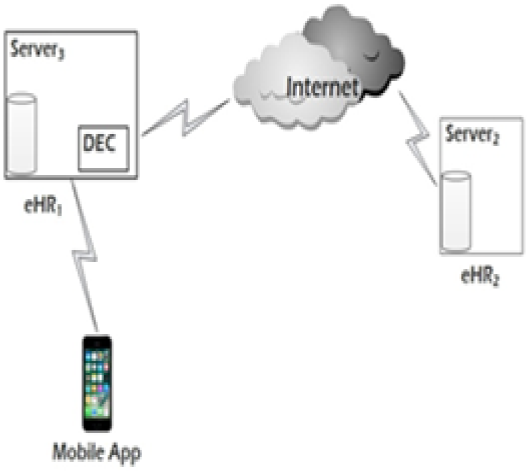


Figure 2 Setup II-two server setup for the DEC and participating eHRs

C. Setup III: use of single server; A local or cloud server

The third proposed setup is such that both eHRs are hosted on the same server with DEC. This architecture can be implemented using two models. One is to host the eHRs and DEC in a physical/local location as illustrated in *Figure 3*; the second model is to host them in the Cloud on a virtual server as shown in *Figure 4*.

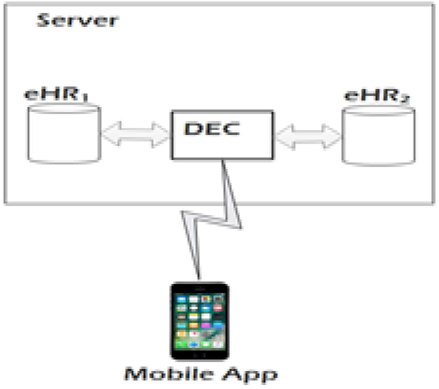


Figure 3 Setup IIIA-single server; physical/local server host setup

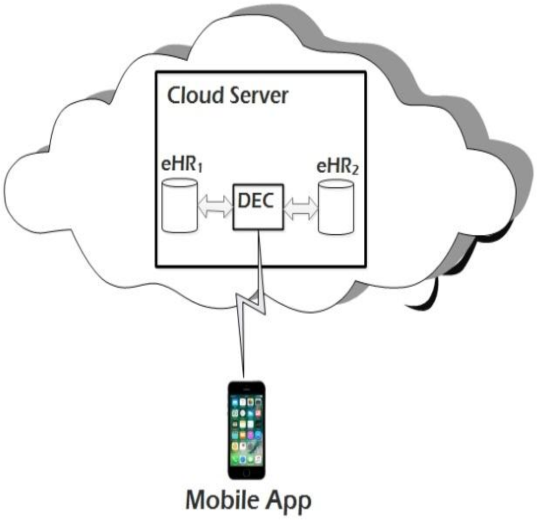


Figure 4 Setup IIIB-single server; cloud server host setups

4.Proposed architectural design of the DEC

Fundamentally, the DEC provides a mechanism to connect a single mobile app with multiple eHRs. The architecture is presented in *Figure 5*. It shows an interface to a mobile app, the processing and exchange mechanism where the authentication and data exchange between the mobile app and the connected multiple eHRs occur. Also, the figure shows an interface to eHRs/databases and the connected eHRs.

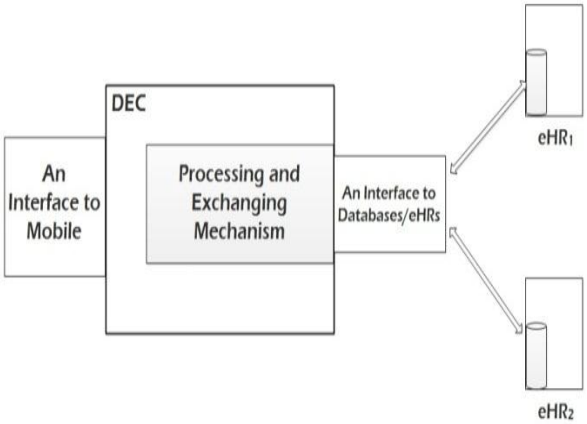

DEC
An Interface to Mobile
Processing and Exchanging Mechanism
An Interface to Databases/eHRs
eHR1
eHR2

Figure 5 DEC to multiple eHRs architecture

A. The DEC building blocks

In [16], the authors propose the use of DEC as a solution component toward eHRs interoperability through a single mobile app. *Figure 6*, presents the DEC architecture's building blocks.

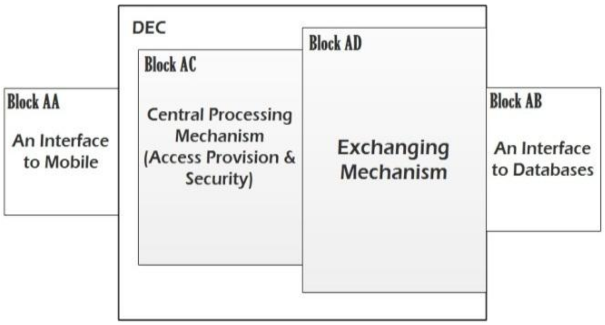

DEC
Block AD
Block AC
Central Processing Mechanism (Access Provision & Security)
Block AA
An Interface to Mobile
Exchanging Mechanism
Block AB
An Interface to Databases

Figure 6 The main building blocks of DEC architecture

The following are the details of each building block: (See *Figure 6*).

i.*An interface to the mobile side (Block AA):* This provides a way for the developed mobile app to be connected to the DEC. By being connected to the data exchange, mobile app users will have the possibility to access data/info located in the connected eHRs through an authentication process, i.e. the user has to be a legally registered user who has been given access to do so. The user has to pass the central processing mechanism, to pull patients info and by the help of DEC application the info-data can be exchanged. To connect the mobile app to the DEC, a mobile app developer will follow the steps on how to connect and post/retrieve data between the two eHRs.

ii.*An interface to database/eHRs side (Block AB):* The Interface to databases resides on the side of DEC where it provides an interface to connect eHRs through which they can interact with the DEC. Through this interface, eHRs can exchange data and information through the exchange mechanism within the DEC.

iii.*Central processing mechanism- access provision and security (Block AC):* The central processing mechanism provides a mechanism that controls access to the connected eHRs. A user has to be authenticated here and thereafter, allowed to access the eHR databases. The authentication should be controlled by the owner of the connected eHRs, where, before accessing /transferring data from the connected eHRs, the mobile app must be approved by both systems.

iv.*Exchange mechanism (Block AD):* This is a step by step SQL procedure that manipulates the two eHRs. These are codes that perform the exchange of information between the two eHRs.

B. How DEC works

The DEC will enable the connected mobile app and web-interface users to access data/information in the connected eHRs. To connect a mobile app to the DEC, a mobile app developer will receive procedures on how to connect the app to the DEC. The developer will configure the app to exchange information with the connected eHRs through DEC. After a successful configuration, the mobile app/web-interface user will continue to use the app/web-interface to add, modify, transfer, search, or retrieve data/information to/fro the connected eHRs through DEC. The DEC will allow authentication process when the app/web-interface users login before accessing or transferring data/information. A display flow chart for this process is given in *Figure 7*.

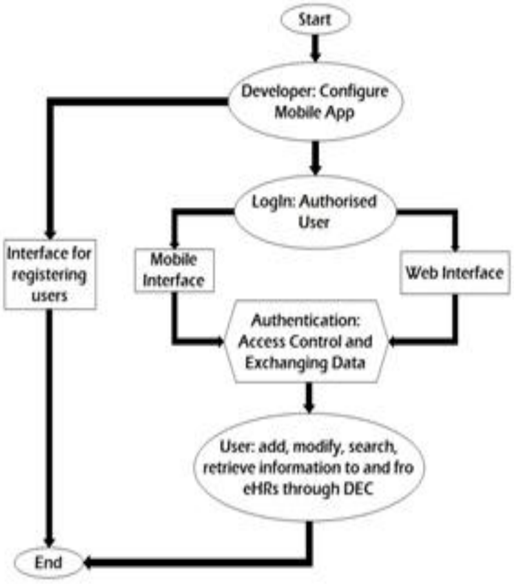

Start
Developer: Configure Mobile App
LogIn: Authorised User
Interface for registering users
Mobile Interface
Web Interface
Authentication: Access Control and Exchanging Data
User: add, modify, search, retrieve information to and fro eHRs through DEC
End

Figure 7 A display flowchart overview

Essentially, the DEC has two blocks, marked as AC and AD, that control the access (authentication) and facilitate data exchange mechanism as explained earlier in this Section. The DEC provides a platform for communication between the two connected eHRs i.e. the connected eHRs will exchange and transfer data/information through DEC.

Users of DEC fall into two main categories;

i. *Category I: Technical Users–Mobile App developers and System Administrators:* - Users in this category are those who are responsible for setting up the connections, configuring DEC to work with the intended systems and the mobile app. This category comprises the hospital system/database administrators and mobile app developers. These users are also responsible for the management of the DEC and the system over its lifetime.

A mobile app developer or hospital system administrator will download the DEC and configure it into the intended server and then connect the targeted eHRs. Consider *Figure 8* for more details.

ii.*Category II: Normal Users:-* The users in this category are mobile app/web-interface users; they are the intended beneficiaries of the DEC. This category includes hospital personnel such as health practitioners, administrators etc. and any other approved users within the participating hospitals.

Figure 9 illustrates how category II users interact with DEC.

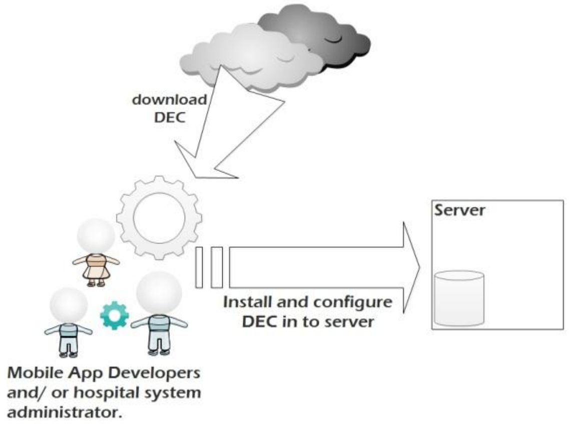

download DEC
Server
Install and configure DEC in to server
Mobile App Developers and/ or hospital system administrator.

Figure 8 The DEC downloaded by mobile app developer or hospital system administrator and configures it into a sever

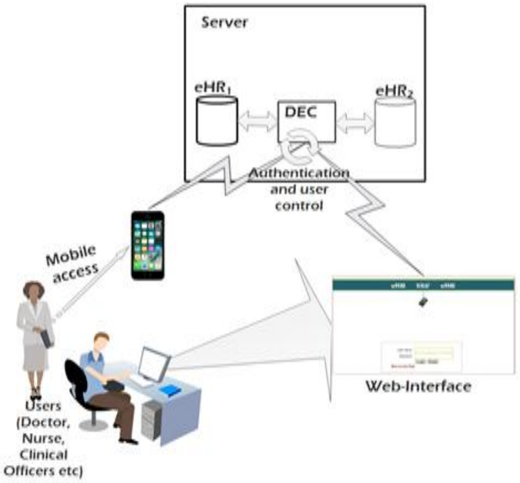

Server
eHR1
DEC
eHR2
Authentication and user control
Mobile access
Web-Interface
Users (Doctor, Nurse, Clinical Officers etc)

Figure 9 Category II users' interaction with the DEC

C. Data repositories for the participating eHRs

Two or more participating eHRs may be used to demonstrate interoperability between eHRs through the developed DEC. These systems may be stored in repositories such as MySQL, Oracle etc. The repositories in relation to DEC are planned to reside in the single, two or three different servers as explained in Section 3.

D. System, DEC building blocks and users' interaction

Users of the eHRs (doctors, clinical officers, health practitioners, etc.) will perform actions such as add, modify etc. to the eHRs. The interface to access data will be provided through mobile app and the web-interface. With the interfaces configured and enabled, a user will be able to login to the DEC through authentication processes provided by the DEC and the participating systems (eHRs). Upon successful login, a user will search patient details by using the known patient ID where the details could be transferred to the other connected eHR (the other hospital in this case). Through the data exchange mechanism described in sub-section A of this section, hospitals could exchange medical documents such as discharge summary, referral details, lab results, etc. The block diagram in *Figure 10* illustrates the described interaction.

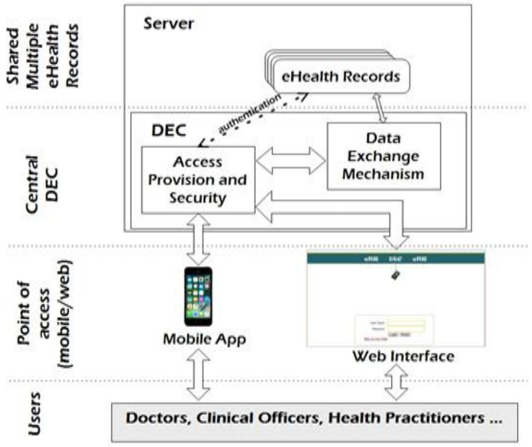

Shared Multiple eHealth Records
Central DEC
Point of access (mobile/web)
Users
Server
eHealth Records
authentication
DEC
Access Provision and Security
Data Exchange Mechanism
Mobile App
Web Interface
Doctors, Clinical Officers, Health Practitioners ...

Figure 10 The block diagram for system, DEC building blocks and user interaction

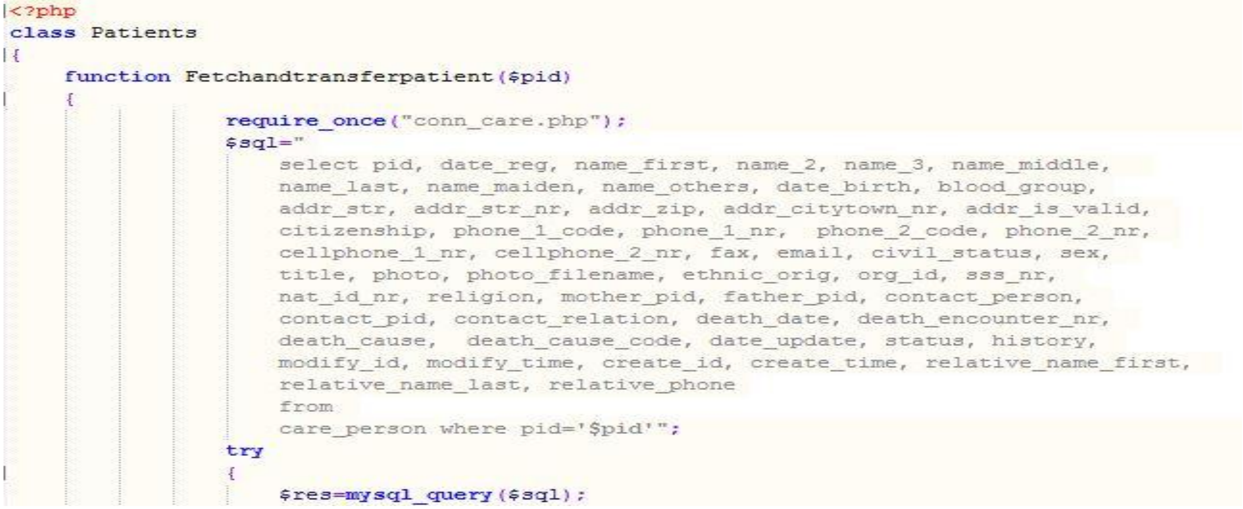
5.DEC implementation and testing

OpenMRS and Care2x Information systems were used as sample eHRs to demonstrate the applicability of the developed DEC architecture. The DEC, as a web application, resides and runs within a web server. The user of the DEC will download it from online repository as a compressed folder. After downloading, the user will un-compress it, install and then configure it in the intended server (see *Figure 7*). The following are pieces of codes showing the implementation of the developed DEC.

A. Data exchange

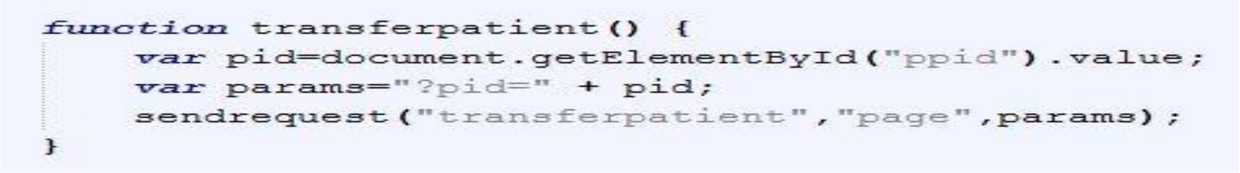
Figure 11 shows the fetch and transfer patient function that illustrates the data exchange process. Also, it shows scripts for patient data transfer from Care2x to OpenMrs. The transfer patient function is presented in *Figure 12*.

Figure13 shows the fetch and transfer patient's data function, these codes display the process of transferring patient data into OpenMrs system.



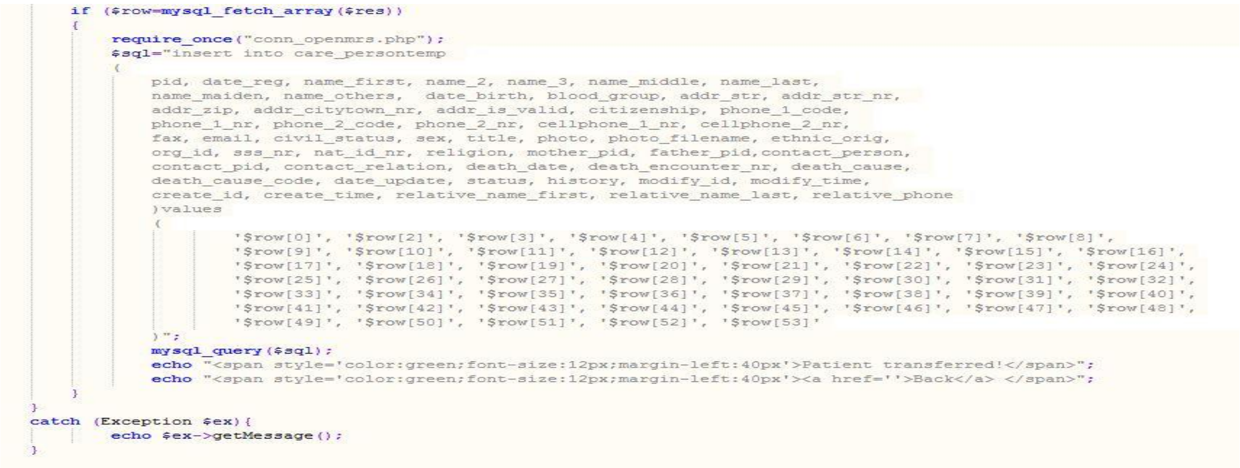
```
<?php
class Patients
{
    function Fetchandtransferpatient($pid)
    {
                require_once("conn_care.php");
                $sql="
                    select pid, date_reg, name_first, name_2, name_3, name_middle,
                    name_last, name_maiden, name_others, date_birth, blood_group,
                    addr_str, addr_str_nr, addr_zip, addr_citytown_nr, addr_is_valid,
                    citizenship, phone_1_code, phone_1_nr,  phone_2_code, phone_2_nr,
                    cellphone_1_nr, cellphone_2_nr, fax, email, civil_status, sex,
                    title, photo, photo_filename, ethnic_orig, org_id, sss_nr,
                    nat_id_nr, religion, mother_pid, father_pid, contact_person,
                    contact_pid, contact_relation, death_date, death_encounter_nr,
                    death_cause,  death_cause_code, date_update, status, history,
                    modify_id, modify_time, create_id, create_time, relative_name_first,
                    relative_name_last, relative_phone
                    from
                    care_person where pid='$pid'";
                try
                {
                    $res=mysql_query($sql);
```

Figure 11 Part of the fetch and transfer function; sample codes showing fetch patient data and transfer it from Care2x to OpenMrs



```
function transferpatient() {
    var pid=document.getElementById("ppid").value;
    var params="?pid=" + pid;
    sendrequest("transferpatient","page",params);
}
```

Figure 12 Transfer patient function

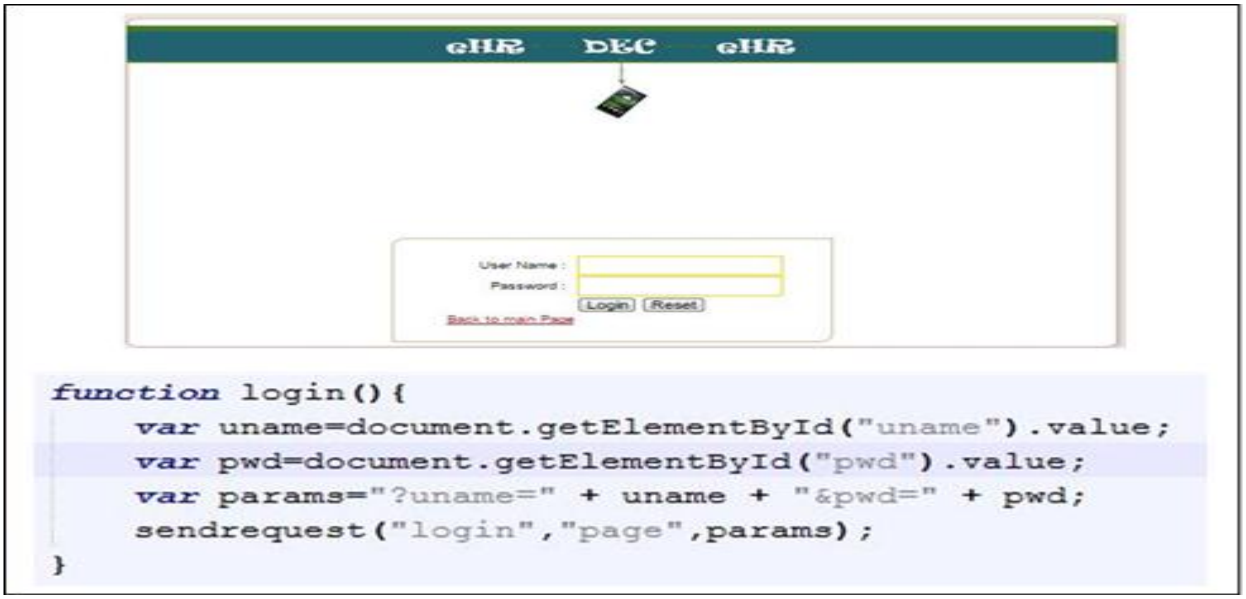


```
            if ($row=mysql_fetch_array($res))
            {
                require_once("conn_openmrs.php");
                $sql="insert into care_persontemp
                (
                    pid, date_reg, name_first, name_2, name_3, name_middle, name_last,
                    name_maiden, name_others,  date_birth, blood_group, addr_str, addr_str_nr,
                    addr_zip, addr_citytown_nr, addr_is_valid, citizenship, phone_1_code,
                    phone_1_nr, phone_2_code, phone_2_nr, cellphone_1_nr, cellphone_2_nr,
                    fax, email, civil_status, sex, title, photo, photo_filename, ethnic_orig,
                    org_id, sss_nr, nat_id_nr, religion, mother_pid, father_pid,contact_person,
                    contact_pid, contact_relation, death_date, death_encounter_nr, death_cause,
                    death_cause_code, date_update, status, history, modify_id, modify_time,
                    create_id, create_time, relative_name_first, relative_name_last, relative_phone
                    )values
                    (
                            '$row[0]', '$row[2]', '$row[3]', '$row[4]', '$row[5]', '$row[6]', '$row[7]', '$row[8]',
                            '$row[9]', '$row[10]', '$row[11]', '$row[12]', '$row[13]', '$row[14]', '$row[15]', '$row[16]',
                            '$row[17]', '$row[18]', '$row[19]', '$row[20]', '$row[21]', '$row[22]', '$row[23]', '$row[24]',
                            '$row[25]', '$row[26]', '$row[27]', '$row[28]', '$row[29]', '$row[30]', '$row[31]', '$row[32]',
                            '$row[33]', '$row[34]', '$row[35]', '$row[36]', '$row[37]', '$row[38]', '$row[39]', '$row[40]',
                            '$row[41]', '$row[42]', '$row[43]', '$row[44]', '$row[45]', '$row[46]', '$row[47]', '$row[48]',
                            '$row[49]', '$row[50]', '$row[51]', '$row[52]', '$row[53]'
                    )";
                    mysql_query($sql);
                    echo "<span style='color:green;font-size:12px;margin-left:40px'>Patient transferred!</span>";
                    echo "<span style='color:green;font-size:12px;margin-left:40px'><a href=''>Back</a> </span>";
            }
        }
        catch (Exception $ex){
                echo $ex->getMessage();
        }
```

Figure 13 Part of the fetch and transfer function; sample codes showing transfer patient data into OpenMrs

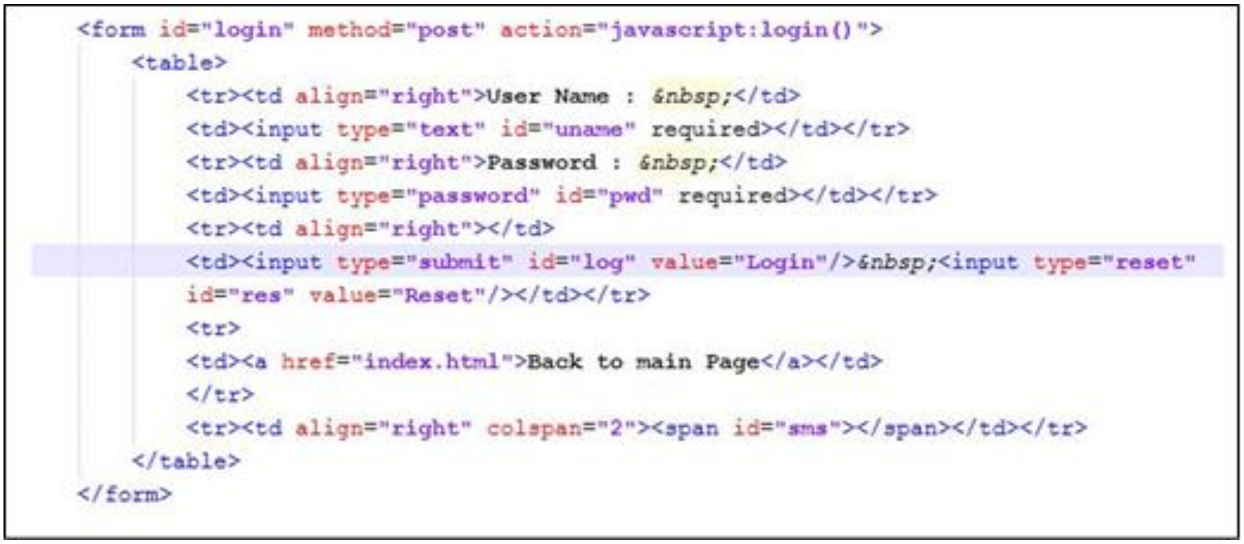
B. Authentication

Figure 14 shows the screenshot of the authentication form and the login function codes where a user is prompted to enter login details. The sample of login form codes is presented in *Figure 15*.



```
function login(){
    var uname=document.getElementById("uname").value;
    var pwd=document.getElementById("pwd").value;
    var params="?uname=" + uname + "&pwd=" + pwd;
    sendrequest("login","page",params);
}
```

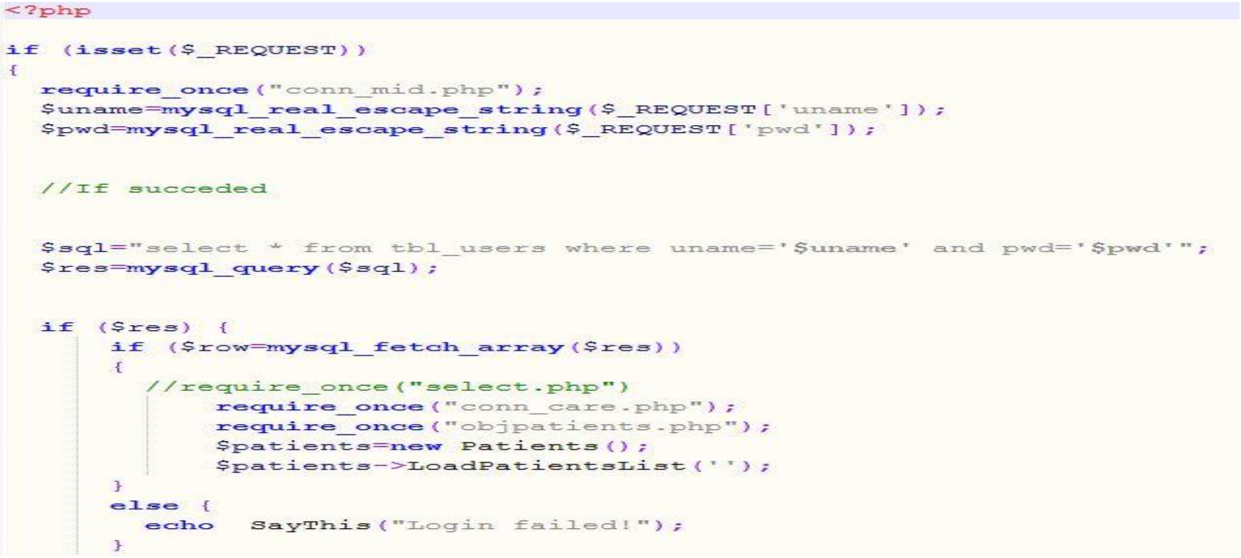
Figure 14 A screenshot for user login web page and a sample login function



```
<form id="login" method="post" action="javascript:login()">
    <table>
        <tr><td align="right">User Name : &nbsp;</td>
        <td><input type="text" id="uname" required></td></tr>
        <tr><td align="right">Password : &nbsp;</td>
        <td><input type="password" id="pwd" required></td></tr>
        <tr><td align="right"></td>
        <td><input type="submit" id="log" value="Login"/>&nbsp;<input type="reset"
        id="res" value="Reset"/></td></tr>
        <tr>
        <td><a href="index.html">Back to main Page</a></td>
        </tr>
        <tr><td align="right" colspan="2"><span id="sms"></span></td></tr>
    </table>
</form>
```

Figure 15 Sample web-based login form codes

Figure 16 shows the scripts that provide access control and security within the DEC. The codes are designed to check the data and then secure them before an insert/retrieve action to/fro the MySQL database. Also, there is a piece of code that selects a user by using the username and password that are inserted by the user through the provided form. Through DEC, a user can transfer/ exchange data from either system, i.e. to and fro OpenMRS and Care2x. *Figure 17* depicts how data are transferred or exchanged from either eHRs through DEC.



```php
<?php

if  (isset($_REQUEST))
{
   require_once("conn_mid.php");
   $uname=mysql_real_escape_string($_REQUEST['uname']);
   $pwd=mysql_real_escape_string($_REQUEST['pwd']);


   //If succeded


   $sql="select * from tbl_users where uname='$uname' and pwd='$pwd'";
   $res=mysql_query($sql);


   if ($res)  {
       if  ($row=mysql_fetch_array($res))
       {
          //require_once("select.php")
                require_once("conn_care.php");
                require_once("objpatients.php");
                $patients=new Patients();
                $patients->LoadPatientsList('');
       }
       else  {
          echo   SayThis("Login failed!");
       }
```

Figure 16 Sample login codes

In the first round of testing process, data access by a mobile app is not included since the goal in this round is to ensure a successfully data exchange process between connected eHRs. Instead, a web-based platform was used where a user can login to the DEC through a web-based login form. Work is in progress to test the process using a mobile app. In the testing process, patient registration details were successfully transferred from Care2X to OpenMrs. The architecture adopted for the purpose of this testing was Architecture IIIA, which is discussed in Section 3 of this paper.

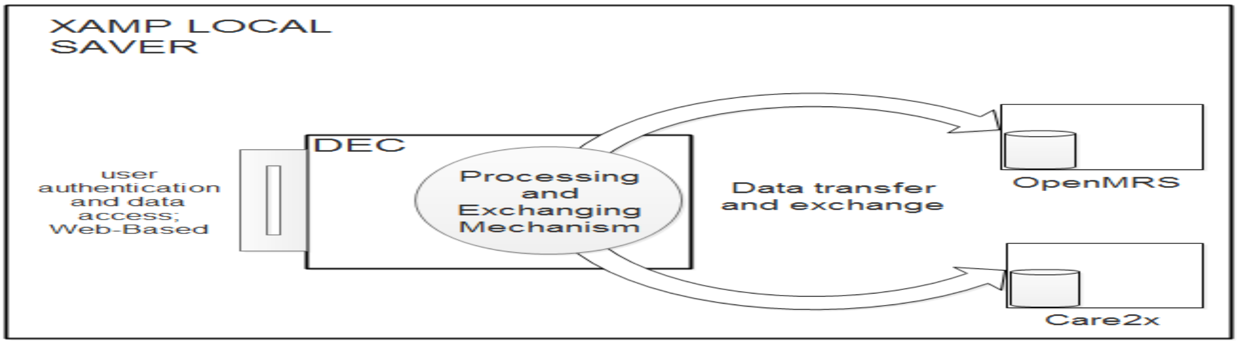

XAMP LOCAL SAVER
user authentication and data access; Web-Based
DEC
Processing and Exchanging Mechanism
Data transfer and exchange
OpenMRS
Care2x

Figure 17 Data transfer/exchange from either system through DEC to XAMP server

6.Discussion

Three alternative setups for deploying DEC were presented in Section 3. These setups provide possible arrangements on how the DEC can be installed. The arrangements, consider options that the hosting institutions could choose based on the cost of implementation within the existing setups, principles and policy. The option of choice by the participating institutions will base on the best viable option that will fit institutional requirements.

In this paper, the authors propose a DEC that provides a platform to connect a mobile app with multiple eHRs. The motivation here is that utilization of mobile devices in healthcare is increasing every day. Deployment of the DEC in this manner is designed to leverage and capitalize on the high penetration of mobile phones (devices) in the healthcare delivery. This is due to the wide range of capabilities and uses of mobile devices especially mobile phones. It is noted that subscriptions of mobile phones globally reached 4.8 billion in 2016, out of which 8.8% are from Sub-Saharan Africa; projected to hit 543m in 2020 [2]. Justifiably then health sectors globally are harnessing the availability of mobile devices to extend its services [17]. As a result, integrating mobile apps with eHRs has become one of the hot research topics in the health sectors.

Implementation and testing of DEC were done and results presented in Section 5. Two open sources eHRs were used as samples. OpenMRS and Care2x health systems were connected to DEC. Point of access used was a web interface where patient registration details were transferred from OpenMRS to Care2x successfully.

7.Conclusion and future directions

Achieving interoperability between two or more eHRs is still a challenge in both developing and developed countries. Yet, interoperability of eHRs continues to be an important aspect towards delivering good health care services. Efforts among health care stakeholders, practitioners, researchers, etc. are still directed toward achieving interoperability among eHRs.

Use of specialized software-based DEC has been suggested as a viable solution to achieve the interoperability of eHRs. The significant contribution of this work is to boost the efforts towards complete interoperability of eHRs by presenting a DEC design architecture, implementation and web-based interface testing. The DEC focuses on achieving interoperability of two different eHRs. It has been shown that, the proposed DEC provides a mechanism where two eHRs can share data/information from each other through it. Therefore, the realization of DEC is a novel effort and could help to add value in the current efforts towards achieving interoperability of eHRs in hospitals, and health institutions.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Khan AA, Krishna S. Using social media in primary healthcare settings. Master’s Thesis, Blekinge Institute of Technology, Sweden . 2013.

[2] WHO. http://www.who.int/trade/glossary/story021/en/. Accessed: 17-October-2017.

[3] Adebesin F, Kotzé P, Van Greunen D, Foster R. Barriers & challenges to the adoption of E-Health standards in Africa. Health Informatics South Africa (HISA), 2013.

[4] Adebesin F, Kotze P, Foster R, Van Greunen D. A review of interoperability standards in e-health and imperatives for their adoption in Africa. South African Computer Journal. 2013; 50(1):55-72.

[5] Mansoor ME, Majeed R. Achieving interoperability among healthcare organizations. Blekinge Institute of Technology. 2010.

[6] MoH U. Health sector strategic plan III 2010/11–2014/15. 2010.

[7] The United Republic of Tanzania. Ministry of Health and Social Welfare. Tanzania National eHealth Strategy 2012– 2018. 2013. http://www.who.int/goe/policies/countries/tza_ehealth.pdf. Accessed: 17-October-2017.

[8] Kajirunga A, Kalegele K. Analysis of activities and operations in the current E-Health landscape in Tanzania: focus on interoperability and collaboration. International Journal of Computer Science and Information Security.2015; 13(6):49-54.

[9] Zayas AD, Gomez PM, Tocado FR. Mobile application profiling for connected mobile devices. IEEE Pervasive Computing. 2010; 9(1):54-61.

[10] Kaushal R, Kern LM, Barrón Y, Quaresimo J, Abramson EL. Electronic prescribing improves medication safety in community-based office practices. Journal of General Internal Medicine. 2010; 25(6):530-6.

[11] Halamka J, Overhage JM, Ricciardi L, Rishel W, Shirky C, Diamond C. Exchanging health information: local distribution, national coordination. Health Affairs. 2005; 24(5):1170-9.

[12] Asuman D, Tuncay NE, Alper O, Gokce L, Yildiray K, Marco E. Key issues of technical interoperability solutions in eHealth and the RIDE project. Software Research and Development Center, Department of Computer Engineering, Middle East Technical University, Ankara, Turkey. 2006.

[13] Iroju O, Soriyan A, Gambo I, Olaleke J. Interoperability in healthcare: benefits, challenges and resolutions. International Journal of Innovation and Applied Studies. 2013; 3(1):262-70.

[14] Guedria W, Lamine E, Pingaud H. Health systems interoperability: analysis and comparison. In international conference on modeling, optimization and simulation of systems 2014. CRAN laboratory, University of Lorraine.

[15] Nehemiah L. Development of a semantically interoperable electronic healthcare record framework to enhance clinical data sharing and decision making. MSc Thesis, Nelson Mandela African Institution of Science and Technology, Tanzania, 2014.

[16] Chali FH, Yonah ZO, Kalegele K. Critical issues and challenges in developing mobile-based health systems: case of Tanzania. International Journal of Computer Science and Information Security. 2017; 15(3):404-12.

[17] Modi S. Mobile health technology in developing countries: the case of Tanzania. Pepperdine Policy Review. 2013; 6:E1-E18.

Frederick Henri Chali received his BSc. In Information and Communication Technology Management at Mzumbe University, Tanzania in 2007 and MSc in Computer Science at The University of Dodoma, Tanzania in 2011. Currently he is a PHD candidate in Information Communication Systems and Engineering (ICSE), specialized in Information Technology Development Systems and Management (ITDSM) at the school of Computational and Communication Systems and Engineering (CoCSE), The Nelson Mandela African Institution for Science and Technology (NM-AIST). Till now he is being an Assistant Lecturer at the Department of Information systems of the School of Informatics of the College of Informatics and Virtual Education (CIVE), The University of Dodoma. His area of research and interests lies in the field of Mobile Computing, Web Technologies, and Data Mining.
Email: kalegs03@gmail.com

Eng. Dr. Zaipuna O. Yonah MIET, MIEEE - holds a B.Sc. Degree (with Honors-1985) in Electrical Engineering from University of Dar es Salaam - Tanzania; and M.Sc. (1986) and PhD (1994) degrees in Computer-Based Instrumentation and Control Engineering from the University of Saskatchewan, Saskatoon, Canada. In Tanzania, he is a Consulting Engineer in ICTs/Telecoms. He is currently associated with The Nelson Mandela Institute of Science and Technology and Applied Engineering & ByteWorks (T) Ltd. He is one of the mentors and pioneers driving the national and regional broadband agenda in Tanzania, EAC, and SADC regions. His current work aims at creating and delivering value through ICT-enabled services in the shortest times possible. His research interests include: ICT4D, Mobile and Web applications, Big Data, Data Mining, High Performance Computing, high-capacity Broadband networks, Intelligent Instrumentation and Control Engineering, Embedded and Mobile systems and ICT enabled 21st Century Education delivery (ICT4E).
Email: zaipuna.yonah@nm-aist.ac.tz

Khamisi Kalegele is an ICT researcher with special interest in transformative applications of technologies for development. He received his Bsc Eng from The University of Dar es Salaam in Tanzania, M.Eng (Computer Science) from Ehime University in Japan and PhD in Information Sciences from Tohoku University in Japan in 2012. Of recent he has mostly dealt with health, agriculture, education and water sectors, conducting research on improving information management, discovering new insights from data, applying remote sensing and GIS techniques to manage fluoride infested lands and water, to mention a few.
Email: kalegs03@gmail.com