Cloud-Based Medical Laboratory Information System

Nehal Abdullah
Department of Biomedical Engineering
University of Cairo
Cairo, Egypt
nehalnasr23@gmail.com

Heba-Allah Hussam
Department of Biomedical Engineering
University of Cairo
Cairo, Egypt
egyptiangirl.94@gmail.com

Yusra Abd-Rabou
Department of Biomedical Engineering
University of Cairo
Cairo, Egypt
yusraabdrabou16@gmail.com

Mohamed Magdy
Department of Biomedical Engineering
University of Cairo
Cairo, Egypt
mohamedmagdy1994@yahoo.com

***Abstract* - The purpose of this project is to develop and implement a cloud-based solution. This solution integrates all lab functions in one user-friendly system. This system has the potential of interfacing with lab devices and communicating with different health care systems.We chose the N-tier architecture design to model the system. In this project, we used Java 2 Platform Enterprise Edition and MySQL database management system to design all pages and implement all system functions. The staff will use a website to handle their tasks through it. When the patient comes, all the required information about him/her will be collected.Once he/she is on the system database, he can easily log into his/her account. Lab manager could easily limit the access of all the staff according to their tasks. We used health level 7 standard to send tests results from analyzers to LIS automatically and finally we deployed the system and the database to google cloud server so the information can be accessed remotely.**

***Index Terms*—Lab information system, N-tier application,Health Level 7.**

I. Introduction

A laboratory information system (LIS) plays an important role in the quality of Interfacing with lab devices and other health care systems. It can deal with lab data such as blood samples, patients information, tests information and results, and staff information. Such procedures could be effectively utilized to decrease human inaccuracy and mistakes. It also enables establishment of a paperless environment and save time and efforts. LIS is a demand-able component in healthcare process , While the complexity of the laboratories has been increasing over time with the use of new technologies. Our LIS will have many desirable features. It can be implemented in single lab and poly labs to ease data interchange between laboratories. It can interface with lab devices to deliver samples results automatically using Health level 7 standard.The front disk will submit test orders using a user-friendly website through a registration form. This form contains patient and provider information. Then he can choose the required tests through a predefined lab tests list contains information about the test.

The system will provide a definable reports to report patient results which can be printer easily or notified by e-mail. User Management process will be included as a tool for the system admin to add or delete users. He also can limit their access according to their tasks. The significance of this project is to build a cloud based LIS has the ability to interface with lab devices and other healthcare systems. There are different cloud based LIS solutions in market like Biotracer by Cloudlims[1], Cloudpath by Optra scan[2], XIFIN LIS Anywhere by XIFIN[3], and MediEaz LIS by Exleaz[4].

II. Materials and Methods

Preparing the architecture was the first step to implement our LIS..Simple idea of our architecture is called N-tier application shown in Fig. 1, It provides separate layers distributed between the client and server. N-Tier application include presentation layer that represent the user interface, data layer to access database and business layer used to data validation and business rules to connect between other tiers.

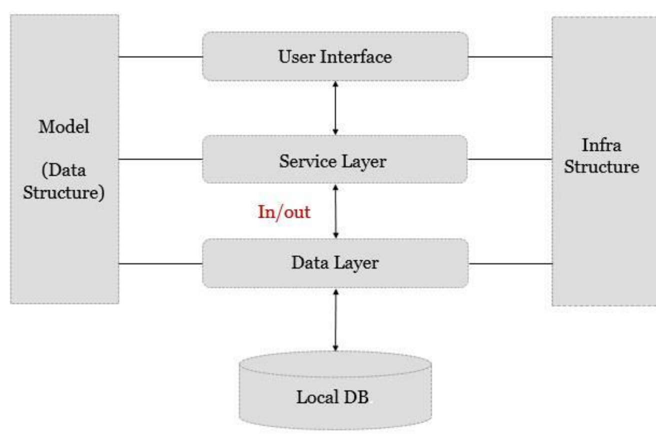


Fig. 1 N-tier application architecture.

A. Web Application

We started our design by implementing GUI (graphical user interface) using JSP (java server page) to develop interactive web page (Figure 2). JSP represent as normal HTML pages include Java code. Using JSP allows us to access database to save all information about patient and its tests.

Fig. 2: JSP interactive page.

B. Database

Database was created by MySQL that is one content of Java package. JDBC (Java database connectivity) is an application interface specification used to connect programs written in java to database. We developed our LIS by using eclipse which is an open source platform. Implement JSP specification is done by Apache Tomcat server (6),it is an open web server developed by Apache software foundation (ASF).

C. Other technologies

We also used JavaScript, jQuery, jQuery Validation Plugin version 1.14.0, Ajax, JSP, MySQL Workbench version 6.2 CE, iText library version 5.5.9, Java Mail version 1.4.7 and Java Development Kit (JDK) version 8. We used Apache Tomcat Server version 7 to test our web app locally. All these technologies are integrated in our source codes and implemented via Eclipse IDE. We used google app engine to deploy our web app to be accessed remotely.

D. LIS-Device Interface

Clinical analyzers produce test results, which must be sent to the LIS .Once these results are in the LIS it can be used in different ways.

We used Health level 7 standard to send requests and receive results.To deliver the results from an analyzer for example to the LIS system host,we took some basics in consideration:1) Serial communications between the analyzer and its data management system.2)The Analyzer and the LIS are on the same network or VPN. This part walked through two steps:- using Hl7 to create the strategy of request and response message. Then building the connection between the analyzer and the LIS to send results files and design mappers for data transformation. We worked on two types of HL7 messages,laboratory order message OML and its response laboratory response message ORL.We used two laptops one acts as the analyzer PC and the other is the LIS. We used mirth connect integration engine to connect the analyzer and the LIS through lower layer protocol. We designed and implemented the mappers using mirth channels.we designed a jsp page that takes some information about the patient and the test. Then generate an HL7 message and sends it to the analyzer. We implemented three mirth channels. One is the TCP sender/listener between LIS and analyzer,one is for sending results from analyzer to LIS. Then mapping the HL7 message fields to the results table in the database. And the last one is for sending request message from the web page to the analyzer PC.

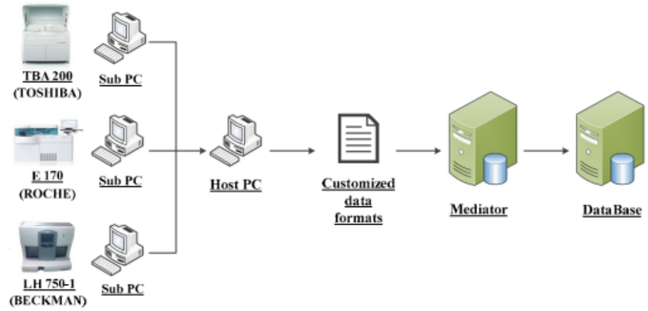


Fig.3.interfacing between analyzers and lab host

E. Cloud Computing

Cloud computing refers to the many different types of services and applications being delivered in the internet cloud[5].

We choose **Google Cloud Server** to publish the system as we registered for a free trial with 300 $ products for 60 days. The free trial gave us 250 GB storage on the server. The deployment went through Four stages as follows

- **Instance creation**

We created an instance to deploy the system on it with instance ID **boreal-coyote-13702345**

- **Database publishing**

We uploaded the previously constructed local database to the **google cloud storage** then we imported it to **google cloud SQL** with IP address **173.194.252.218.**

- **Web application deployment**

We used **google app engine** to deploy the web application by creating an application in it and then deploy the web application to that application using its ID.

- **Connection between the web application and the database**

Due to the lack of time, we could not finish this final stage and this will be our future work to accomplish our target.

III. Results

A. Model View Controller Pattern (MVC)

We achieved MVC concept where a software application has business logic (model) separated from presentation logic (view)[6]. We achieved this separation into a model layer (responsible for implementing the business logic) and a view layer (responsible for rendering the user interface to a specific client type). In addition, the controller layer sits between those two layers, intercepting requests from the view (or presentation) layer and mapping them to calls to the business model, then returning the response based on a response page selected by the controller layer. **Figure 3** shows MVC cycle[7].

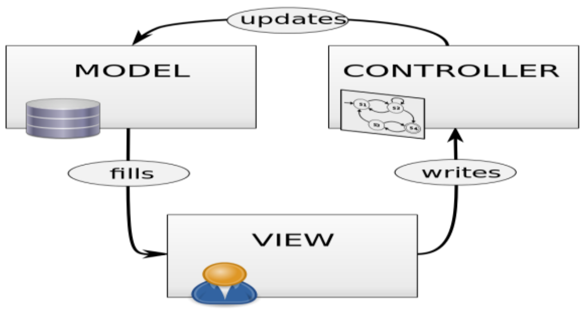


Fig.4.MVC Cycle

B. *Web Application Workflow*

Figure 4 shows structure of our web application and workflow of WebPages.

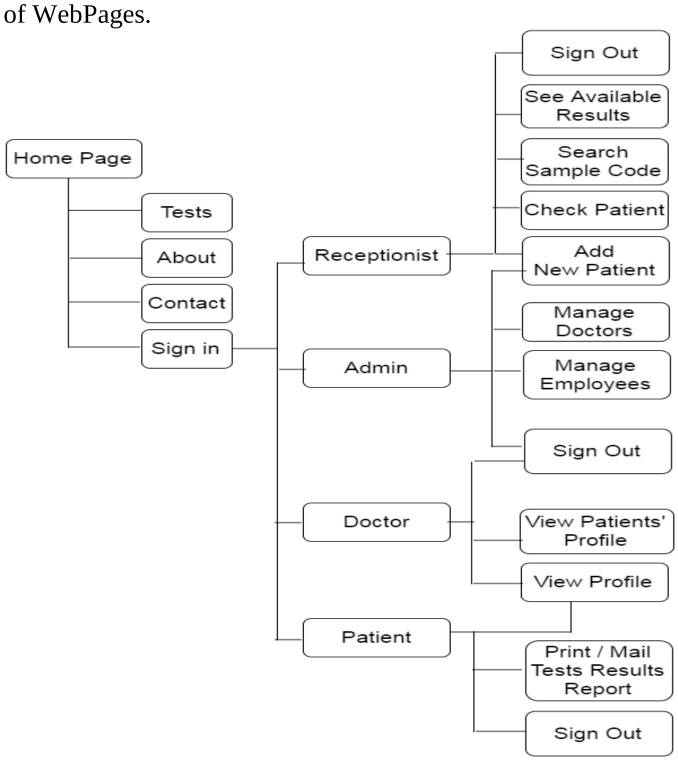


Fig.5.Structure of Web application

C. *Access Control*

Following MVC concept, controller checks username and password values coming from sign in form with all usernames and passwords in database. If these data are valid and user's activity is "*active*" in database then controller allows the user to access one of four pages (patient.jsp or receptionist.jsp or doctor.jsp or admin.jsp) according to their role assigned before in database. **Figure 5** shows sign in form.

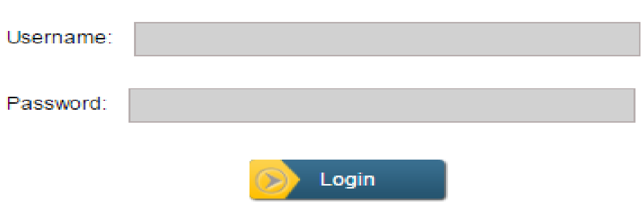


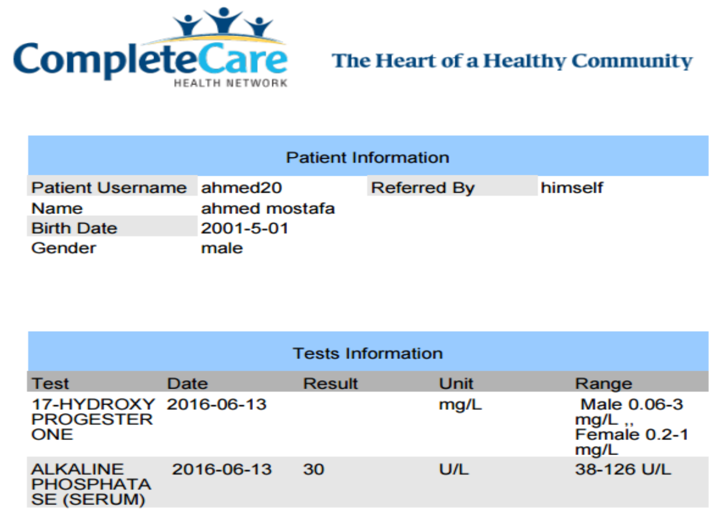
Fig.6.Sign in form

D. Barcode Generator

The system allows receptionist to generate barcode and assign it for each patient sample. When the sample runs on the barcode reader, the barcode is scanned and then receptionist can search in the LIS with this code to identify the patient and their done tests.

E. *PDF Generation and Mail Sending*

The system enables patient to view, download, print and get their test results in a generated **pdf** file (shown in **figure 6)** by mail. Receptionists and doctors –from their profile– can also view patients' test results pdf file.



Patient Information			
Patient Username	ahmed20	Referred By	himself
Name	ahmed mostafa		
Birth Date	2001-5-01		
Gender	male		

Tests Information				
Test	Date	Result	Unit	Range
17-HYDROXY PROGESTER ONE	2016-06-13		mg/L	Male 0.06-3 mg/L ,, Female 0.2-1 mg/L
ALKALINE PHOSPHATA SE (SERUM)	2016-06-13	30	U/L	38-126 U/L

Fig.7. Pdf file

F. Billing System

We can calculate the total price of selected tests then calculate total price after discount (Figure 7).

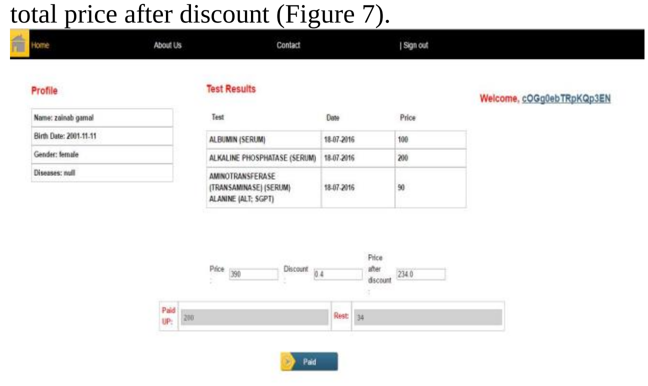


Fig.8..Billing System

Both price paid up and rest store in database.

We can say if the registration process takes about one or two minutes, so LIS will succeed in registering at least 30 patients per hour according to registration period.

G.LIS-Device Interface

When the result HL7 message received at the analyzer PC, it is automatically sent to the LIS,decoded and saved in database.**Fig. 9** shows the HL7 message sent from the analyzer system.**Fig.10** shows the same data saved in the LIS database.

```
MSH|^~\&|XXXX|C|HEALTH|HEALTH|20080511103530||OR1^O22|Q335939501T337311002|P|2.3|||
PID|1|1024189|012993817||nader^sara||19520101|F||
OBX|1|NM|1751-7^ALBUMIN^LN||3|g/dL|3.4-5.4|H|||F|
SPM|1|13059||^Blood||||||||||||||||||||||^Collection Tube||
```

Fig.9.Hl7 result message

Measurment unit	comment	cofidance intervel	patientID	specimenID	testName	result	specimenty
g/dL		3.4-5.4	1024189	13059	ALBUMIN	3	Blood
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig.10.saved data in database

We implemented a mapper using mirth channels to transmit map data from the form page to a HL7 message and sends it to the analyzer PC.**Fig.11**shows the fields with values which are the same as the generated message that received on the analyzer pc **(Fig. 12)**

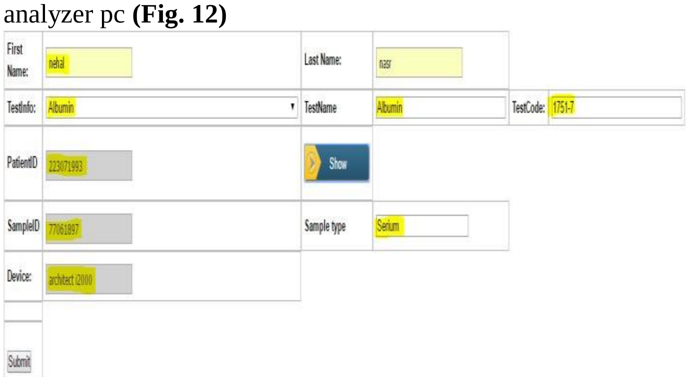


Fig.11.Entering information in web page

```
MSH|^~\&|LIS|Complete Care|Device interface|Device interface|||ORU^R01||P|2.5
PID|||223071993||nasr^nehal|
PV1|1|O
ORC|NM|
OBR|1|||1751-7^Albumin^LN
SPM|1|77061897||^Serium||||||||||||||||||||||^Collection Tube||
SAC||||||
```

Fig.12.recieved Hl7 message

IV.Discussion

We can find that Sign in form and access control are significant ways to secure laboratories data and users information. The system can manage operations happened dailyin clinical laboratories and make them much easier.

A. Difficulties faced

- Difficulties in visualising the use cases and all the expected scenarios that could happen in the system because of the unhelpful medical laboratories.
- Difficulties in using medical devices for the system-device interface testing.
- Difficulties in understanding the H7 standards and applying it .

B. Weak points

- Not connecting the web application with the deployed database on cloud because of the lack of time.
- MySQL does not support unlimited length of data type.
- Some of the system components are not logical in the real life usage because we could not test the system in real medical laboratories.

C. Suggested solutions

- If we had more time we would publish the system to google cloud server and connect it to the deployed database as software as a service SAAS.
- With increasing lab data volume, we think if we used **Mongo DB,** it would be a perfect solution than MYSQL.

Our future work will focus on using cloud computing to allow all users access the system using all variant devices from anywhere and host the database online on remote servers.

V. Conclusion

Cloud-based laboratory information systems have much potential to improve patient care by optimizing the operation of clinical laboratories and most importantly, the interface between health care providers and the clinical laboratories. We envision the LIS as replacing humans in most activities that allow the option of human error. Humans would interact with the LIS through user-friendly interfaces designed with a lean approach to optimize efficiency and maximize productivity.
The purpose of this paper is to pass on our experience of critical design issues and required capabilities to make similar systems work on-site. Though other projects will need to design and roll-out laboratory information systems, we hope to make the process less onerous next time around.

References

[1] https://cloudlims.com/sample-management-software/BioTracer-Standard.html.
[2] http://optrascan.com/cloudpath.html.
[3] http://www.xifin.com/solutions/xifin-lis-anywhere.
[4] http://exleaz.com/medieaz-lis/
[5] https://cloud.google.com/java/
[6] https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller
[7] https://commons.wikimedia.org/wiki/File:MVC_Diagram_(Model-View-Controller).svg