A Coherent Healthcare System with RDBMS, NoSQL and GIS Databases

Huanmei Wu[1], Ashish ambavane[1], Sunanda Mukherjee[1], Songan Mao[2]

[1]School of Informatics and Computing
Indiana Univ Purdue Univ Indianapolis
Indianapolis, IN 46202, USA
{hw9, aambavan, sumukher}@iupui.edu

[2]School of Electrical & Computer Engineering
Purdue University
West Lafayette, IN 47907, USA
songan@purdue.edu

Abstract

With new database system development and new data types emerging, many applications are no longer using a monolithic, simple client/server structure, but using more than one types of database systems to store heterogeneous data. In this project, we exploit the benefits of combing Relational Database Management System (RDBMS) and NoSQL systems in the development of better Electronic Health Records (EHRs) and Clinical Decision Support Systems (CDSS). Specifically, MySQL, MongoDB, and GIS databases are integrated to improve EHR systems and to provide better clinical decision supports. The ACID (atomicity, consistency, isolation, durability) properties of the RDBMS ensure data integrity, database security, efficient SQL queries, easy data access, and effective transaction processing. MongoDB provides the system with clear internal data structure, easy scaling-out, fine-tuning, and convenient mapping of application objects to the database objects. The GIS database allows vivid visualization of the geographic locations of patients, physician offices, and medical facilities. The integrations of these database systems in healthcare help application systems to comply with the EHR HIPAA requirements without compromising on scalability and performance.

Keywords: EHRs; CDSS; NoSQL; MongoDB; GIS.

1 Introduction

Electronic health records (EHRs) are patient centered digital versions of paper charts that makes information available instantly and securely to the authorized users [1]. EHRs contain patient demographics, progress notes, vital signs, medical histories, diagnoses, medications, immunization dates, allergies, radiology images, lab and test results, administrative and billing data [3, 4]. Two important aspects of the EHRs are (i) allowing patients to create and manage their health information, and (ii) sharing their health information with healthcare providers and organizations such as laboratories, specialists, medical imaging facilities, pharmacies, emergency facilities, schools and workplace clinics. This capacity allows all healthcare providers with access to patient health information to be involved in effective patient care [1, 2].

In recent years, the massive amount of data generated by medical facilities needs to be efficiently gathered, processed and analyzed for improved healthcare and better utilization of data. The data in EHRs contains various information that may be mined for new medical knowledge discovery. For effective data mining and quality data analysis, comprehensive EHR data from various medical facilities should be collected and integrated in one overall system. The integrated EHR system design should consider the key issues for system performance, maintenance and scalability so that it can work efficiently [4]. In addition, as the system will work on patient health information, the design must have HIPAA (Health Insurance Portability and Accountability Act) compliance, regarding the privacy and security provisions for safeguarding medical information [17].

There are several medical database solutions, such as hierarchical, relational, object oriented and entity-attribute value models. It is natural to consider Relational Database Management Systems (RDBMS) because of the ACID properties, especially with regard to overall data consistency and availability [6]. It is currently popular among developers for NoSQL databases, which are intrinsically object-oriented and relatively simple. NoSQL systems also commonly have well-designed APIs (Application Programming Interfaces) accessible via most popular programming languages. NoSQL databases provide a great base for big data storage and have unlimited scalability in nature by using distributed storage. Moreover, they can provide an excellent architecture for multimedia databases [6, 7], which are in great for managing medical images.

With both solutions providing different strengths and weaknesses in the design of the integrated EHR system, neither RDBMS nor NoSQL alone can satisfy the special requirements for managing the large-scale medical information. Additionally, for the patients' convenience in locating the nearest providers (and/or medical facilities) according to their search criteria, and for the providers' effectiveness in referencing a patient to a specialist (or lab, or hospital, or other medical facilities), the geographical visualization of the information is desired. With all these requirements and demands, integration of the RDBMS, NoSQL databases, and GIS databases are necessary to improve the EHR system and to overcome the current EHR limitations [6, 7].

This is the author's manuscript of the article published in final edited form as:
Wu, H., Ambavane, A., Mukherjee, S., & Mao, S. (2017). A coherent healthcare system with RDBMS, NoSQL & GIS databases. In Proceedings of the 32nd International Conference on Computers and Their Applications, CATA 2017 (pp. 313-318). The International Society for Computers and Their Applications (ISCA).

This integrated database is not a replacement of EHR systems, such as the Epic systems, which will run as usual. The proposed system is an addition to the EHR systems, which will work with existing EHR systems to provide more functionalities to the different parties in the healthcare arena.

2 Characteristics of EHR Data

The EHR data is heterogeneous, including structured, semi-structured, and un-structured data. The design of the database systems is guided by these special data characteristics. First, for *structured data*, each record will have the similar information and the structures are relatively stable. These data can be efficiently modeled in a relational database. Sample structured EHR information includes but is not limited to [6]:

- *Patient demographics*: which contain the basic patient information, such as age, sex, race, address, contact information, and others.
- *Provider demographics*: which record the provide basic personnel information, such as education and training records, board certifications, contact methods, and associated medical facilities, and others.
- *Medical facility demographics*: which store the basic facility information, such as address, contact methods, the size of the facility, number of employees, number of different medical professionals, the number of beds, and any associated medical groups.
- *Clinic appointments*: which include the information for patient clinic visits, such as the date when the appointment was made, the exact time and duration for the appointment, and the status of the patient visit.
- *Billing information*: which stores information regarding patient account balance, the expenses, payments, credit card and insurance information.
- *Immunization records*: which keep the immunization records, such as the immunization name, dose, and the means of delivery.
- *Insurance information*, *referral information* and other line-of-business data.

Second, there are many *unstructured data*, such as doctor notes for the description, diagnosis, and other important visit-related information. The notes usually are usually in a plain text format. It is not easy to query and obtain a specific information from such unstructured data without additional information processing. For example, natural language processing (NLP) is frequently used to retrieve valuable information from doctor notes. Other sample unstructured data in the EHRs include medical images, voice recordings, videos, and other un-organized information [6, 8].

Lastly, *semi-structured data* fall between structured and unstructured data. There are internal structures, similar among different records. However, each record can have some distinctive information which may appear in another record. Potentially, it can be organized with a relational database. The drawbacks in doing so will potentially result in un-used space, complicated data schema, or loss of natural hierarchical information. Sample semi-structured data in EHRs include:

- *Allergies*: Patients have different allergic reactions to various conditions and require diverse medications. A simple relational schema cannot manifest the complex the information and make it accessible with efficient searches.
- *Medical histories:* Similar to allergies, the chronical diseases vary with patients and with medications [6, 8], making it hard to be modeled by RDB.
- *Lab results:* which differ with test regarding to samples and measurement units. Even with the same sample, different lab instruments can produce different outcomes.
- *GIS information:* The simple geographic information (such as longitude and latitude) can be modeled in a RDB, however, the visualization results, such as map with directions and distances, cannot.

Our hybrid database systems will address these challenges of the various EHR data resulting in well-organized storage, providing efficient access, and promising better system performance.

3 System Architecture

The overall infrastructure of our hybrid system is shown in Figure 1. It includes five major components: the MySQL database (which is a RDBMS), the MongoDB (which is a NoSQL database), the GIS database (for visualization), the web interface for users to query and retrieve information, and the middleware API which will communicate with the front end user interface and backend databases. There are some previous investigations on hybrid database systems [5, 6, 7]. For example, the Opensky had a hybrid system called *Doctrine* in which the active data was kept in MySQL and archive data stored in MongoDB [5].

The backbone of the entire system is three different database systems, which provide better performance, maintenance and scalability of the EHR data. The relational MySQL database stores the structured data and complies with ACID transactional standards. The MongoDB manages the semi-structured data and de-identified documents for large and dynamic patient clinical visit information. The GIS database is developed based on Quantum GIS (QGIS) system. The QGIS transforms the address information to longitude and latitude, provides real-time location information about the patient and medical facilities, and displays the relative positions.

A web interface connects the users, MySQL, MongoDB, and QGIS databases, implemented using HTML JavaScript, and CSS interfaces. It provides database security with user

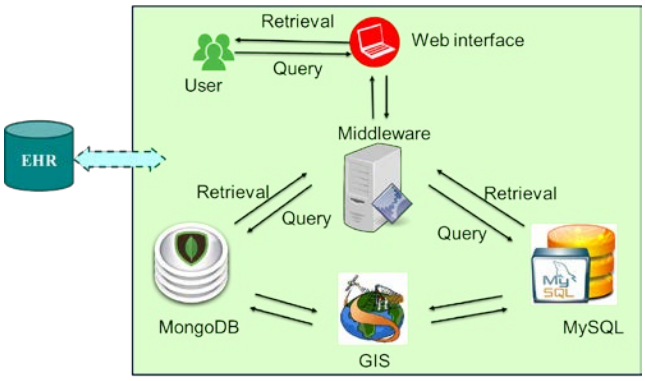


Figure 1: Designed System Architecture.

authentication, such that only authorized end users can fetch the proper information by communicating with the application server (the middleware) to extract the data and generate reports according to the given request. The major functions of the web interface are parsing data access queries with a user-friendly interface and presenting the results to users with a well-formatted web interface [3].

The middleware API ensures the three databases are connected and synchronized through appropriate drivers, developed using JAVA, OpenJS, JQuery and PHP. It authenticates the user based on a username and password and communicates with the user through an interactive web interface. Sample detailed tasks include but not limited to:

- receiving the data access requests from the web interface;
- determining the database (MySQL or MongoDB or both) from which the data should be retrieved;
- generating the appropriate SQL statements for the queries;
- querying the databases for the appropriate data;
- filtering the data, limiting the results only to the requested data;
- sending the data back to the web interface

QGIS is a cross-platform Open Source Geographic Information system for geospatial analysis on the dataset. The advantage of using QGIS is that it considers the environmental factors for the diagnoses and gives a better view of patterns of the disease and epidemiological regions for large population of the dataset. When visualization is requested, the QGIS system will update the searching criteria and display factors for the GIS information to the users.

4 Data Models

For the three database systems, each has different data model and special features. It is worth to mention that not all the EHR data are imported into the integrated systems, but only data, which will be used for the additional functionalities.

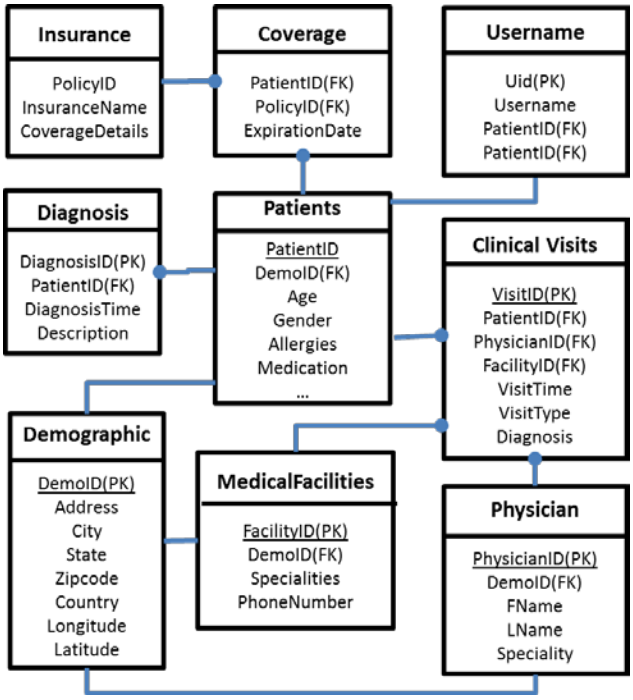


Figure 2: Data model in RDBMS.

Data Model for RDBMS: To ensure the data integrity and to protect the patient privacy, the RDBMS stores the private patient information and critical health records. These data are well-structured and require strict ACID transactions. Figure 2 illustrates a portion of the entity relationship diagram for the RDBMS. The main entities are the patients, the providers, medical facilities and their interactions, including the demographic information, required by the QGIS system. The clinical visits store only the fundamental structured information about a visit for clinical operation tracking. The detailed information of the clinical visits resides in MongoDB.

Data Model for MongoDB: The dynamic clinical information and vital signals includes both unstructured and semi-structured data. These data are de-identified and managed by NoSQL database (MongoDB specifically in this development). The MongoDB document database keeps track of the data changes and builds a knowledge base for effective data analysis and sharing among authorized entities without sacrificing the patient privacy.

The clinical visit data is both patient-specific and visit-specific. It varies according to patient's changing health conditions. This is especially true for lab orders and measurement results. Sometimes, clinical visits result in imaging data, which are totally unstructured. It will be difficult to store the clinical visit data in the RDBMS due to the semi-structured information and dynamic changing properties. MongoDB is a better alternative to store clinical visit data in a document-based model with different sub-collections.

The sample MongoDB collections data model for the clinical visit is illustrated in Figure 3. Each patient clinical visit will generate exactly one *ClinicalVisit* document. Multiple clinical visits for the same patient will generate multiple *ClinicalVisit* documents, tied to one-another by the *patientID* field in the documents, but differentiated by the VisitDate. With this design, users can search and analyze the history of patient clinical visits. There are several potential sub-collections for each clinical visit. Among them, the *PhysicalExams and HealthConditions* will be available for each visit, as they are routinely checked. However, other sub-collections will be optional.

Based on the clinical visit procedure, each ClinicalVisit collection may have different information stored in the sub-collections, such as prescribed medications, lab orders and test results, and risk factors. The structures of the sub-collections vary from collection to collection and even the information of the same collection can potentially differ from time-to-time. For example, the *LabOrder* sub-collection will have all the lab test results. Each test result is a sub-collection inside *LabOrder*, with measurements changing over time link the externally stored large image file to the specific patient clinical visit.

Data Model for QGIS: The information in the RDB and MongoDB will communicate with Quantum GIS (QGIS) for visualization and further analysis using the GIS system connector [10]. QGIS supports connection for MySQL and relational databases so that data can be transferred to it to create the map layers. QGIS also has the *mongodb* connector for MongoDB using the *pymongo* library. Location data are geocoded in the GIS system with extra fields of longitude and latitude for map visualization and further geospatial analysis for patients, physicians, and medical facilities.

For example, when referring a specific patient to a medical specialist by a physician, the demographic information from MySQL for the patient and surrounding specialist offices will be displayed to the physician by the QGIS, centered with the patient location, as shown in Figure 4. The physician can decide the referrals based on the patient requirements, such as the distance from his or her location or the special features of the facilities. Thus, the QGIS will be an integral tool for the clinical decision support systems.

Our developed QGIS System is customized for specific applications using different layers. The following explains the basic sample layers:

- *Map layer:* The geographic map information of the GIS system is the first layer for all the information.
- *Provider layer:* The medical provider data was geocoded using the tool Maps data which allows batch processing of addresses with *Longitude* and *Latitude* information. The results will then be input into QGIS and converted into a *shapefile*. This allows for the visualization of the provider information on top of the map layer.

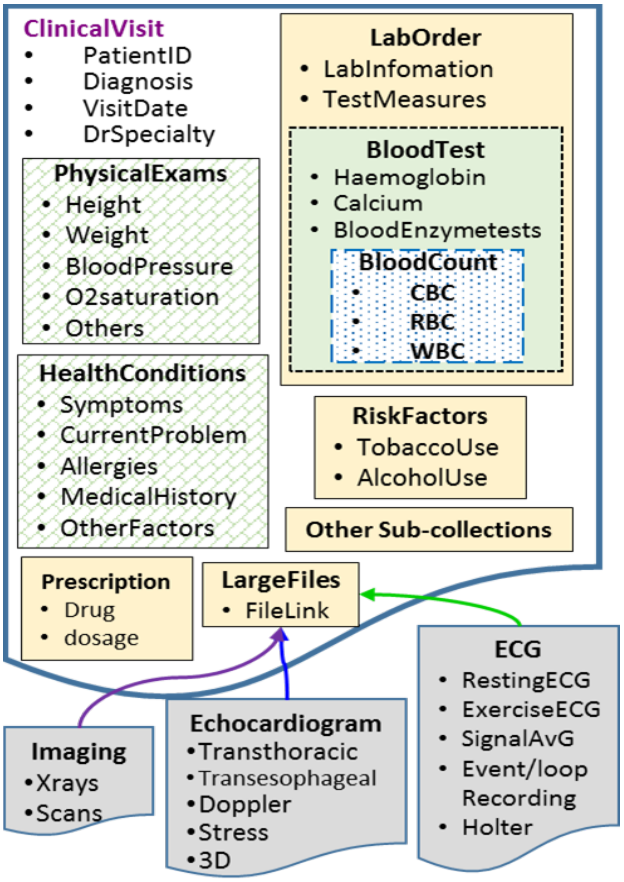


Figure 3: Data model in MongoDB.

- *Patient layer:* The same as the provider layer, the patient GIS information is obtained from the patient data and the resulting *shapefile* will be used to visualize the patient information on top of the map layer.
- *Medical Facility layer:* The same as the patient and provider layer, the facility will also be geocoded for visualization on the map.
- *Additional customized search layers:* The QGIS system can be coded to have additional layers to access and retrieve information. For example, the provider can have additional search layers with their specialties, their sex, and any associated medical groups. The medical facilities can have additional search layers based on hospital size or capabilities.

5 Performances

A prototype has been developed and tested to provide additional functionalities for EHR. Sample data has been collected and stored in the RDBMS and NoSQL systems, combining some demo datasets released by the open source EMR system, ***OpenMRS*** [18]. More than 5000 patient records have been imported to the prototype. OpenMRS dataset lacks the demographic data so sample points are

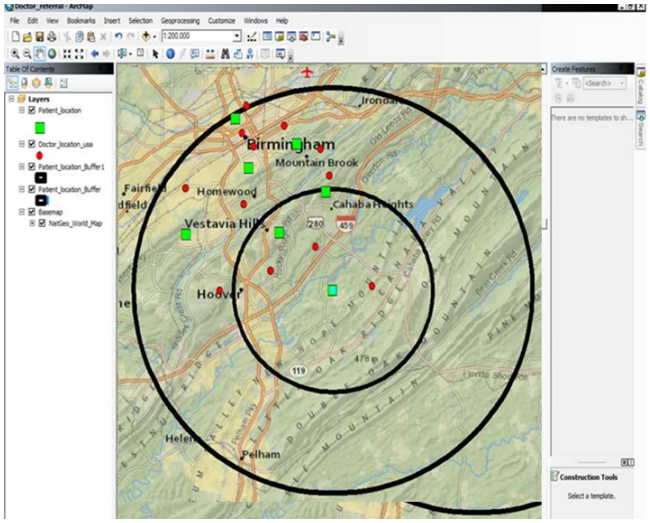


Figure 4: Sample data representation in QGIS system, showing Patient (green) and Doctor Location (Red).The two thick circular lines illustrates the doctors within (2-4 miles) and (4-6miles) from the selected center patient location.

created as patients on the GIS map. It is also difficult to get provider demographic data, which are not freely available on the web. A sample demographic data about patients, physicians, medical facilities are generated by retrieving the physician and facilities information over the web. Additional processing is performed on the addresses to generate the GIS location information.

Enforcing database Security: The regular users of the system do not have direct access to the system. They only have the application level user account and can access the information from the web. The website validates the users for any activities prior to providing access the database. Different user types will allow different access rights. The combination of the role-level access model and the fine-grain data access model are implemented in the system. Users have different privileges to access the database. For example, administrative users can access all the data. However, administrative accounts are only issued to a few database administrators and key system developers. Physician accounts need special authorization and validations are required. Patient accounts can only access the data related to his or her health and any associated minors. The database security measures from RDBMS have been leveraged to make sure to protect the patient privacy and personal health information.

Figure 5 shows an interface for an authenticated physician which lists only the patients for that specific physician. If a specific patient is authenticated, the patient has access to only the data related to his or her health information (or any minors cared for by the patient). As a rule, patients can only view information while physicians are able to add and update information in the system.

Figure 5: The dashboard for the list of patients, viewable by a physician when searched patient names with 'John'.

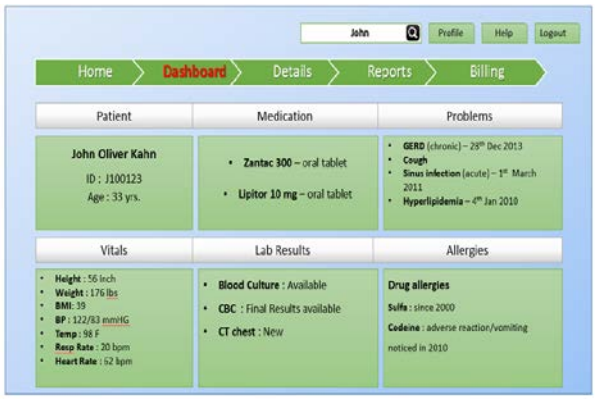
Detail-oriented and User-friendly Dashboards: Once a particular patient is selected, detailed dashboards can be navigated by the authorized users. In our prototype, there are a few dashboard to show the customized information.

- The overall *dashboard* will provide the basic patient health information, such as the medication, previous medical conditions, available lab results, allergies, and vital signs, as shown in Figure 6(a).
- The *Details* will go deeper into the health information, as in Figure 6(b). By clicking on the tabs on the left, the details will be shown in the main window, such as medication history shown as an example.
- The *Reports* section, will keep the current and past medical report belonging to the patient, and can be viewed in reverse chronological order.
- The *Billing* tab will help to check the financial information with the adjustments from the corresponding insurance company. Only administrators or fiscal officers can update it at any given time.

Additional data analytics functions: Additional functions on data analysis and knowledge discovery have been developed based on the system. Customized reports, such as specific disease investigation and epidemiological study, can be generated using the customized report and analytical tools. This is beyond the scope of the current manuscript and will not be further discussed.

6 Conclusions

We implemented a prototype of a hybrid application system that integrates RDBMS, MongoDB, and GIS databases to effectively store and analyze personalized medical data. Each system has its own set of advantages and, together, the integrated application delivers a better clinical decision support system. The system interacts with clinical EHR system and provides additional functionalities on reports, healthcare study, data mining, and statistical analysis.



(a) Dashboard Tab

(b) Details Tab

Figure 6: The personalizaed patient interface.

This currently developed prototype has been evaluated with a relatively small data set. Additional modification and functionalities will be developed to improve the system. Future patient specific studies will also be carried out. For example, on-demand geospatial analysis can help decision making, aid diagnosis, and help epidemiological study [9, 11]. More functionalities with various machine learning and data mining tools will be implemented.

Acknowledgements

This research was partially funded through a Patient-Centered Out-comes Research Institute (PCORI) Award # IH-12-11-5488; Science and Technology Development Project of Henan Province (No.132102410033), National NSF of China (No.U1304606) and Research Foundation for Key Program of Henan Province (No.14B413008).

References

[1] A. Charania, D. Kibler "Electronic Health Records (EHR)" Unpublished.

[2] A. K. Jha, et al. "Use of electronic health records in US hospitals," New England Journal of Medicine 360.16 (2009): 1628-1638.

[3] K. Häyrinen, K. Saranto, and P. Nykänen, "Definition, structure, content, use and impacts of electronic health records: a review of the research literature," International journal of medical informatics 77.5 (2008): 291-304.

[4] A. Hoerbst and E. Ammenwerth, "Electronic health records," Methods Inf Med 49.4 (2010): 320-336.

[5] J. Carter, "Investigating NoSQL for EHR Systems: MongoDB," EHR Science, APRIL 15, 2013.

[6] R. Lawrence, "Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB," Computational Science and Computational Intelligence (CSCI), 2014 International Conference on. Vol. 1. IEEE, 2014.

[7] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," Health Information Science and Systems, pp: 2-3, 2014.

[8] R. A. Iqbal, "Hybrid Clinical Decision Support System: An Automated Diagnostic System for Rural Bangladesh," IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision ICIEV 2012.

[9] M. Bageshwari, P. Adurkar and A. Chandrakar, "Clinical Database: Rdbms V/S Newer Technologies (Nosql and Xml Database); Why Look Beyond Rdbms And Consider The Newer," International Journal Of Computer Engineering & Technology (ICJET) Volume 5, Issue 3, pp: 73-83, March, 2014.

[10] L. Byczkowska-Lipinska, A. Wosiak, "Multimedia NoSQL database solutions in the medical imaging data analysis," Przeglad Elektrotechniczny, December, 2013.

[11] G. Adrián, G. E. Francisco, M. Marcela, A. Baum, L. Daniel, G. B. Fernán, "MongoDB: An open source alternative for HL7-CDA clinical documents management," in Proceedings of the Open Source International Conference, Buenos Aires, Argentina, 2013.

[12] W. Luo, "Using a GIS-based floating catchment method to assess areas with shortage of physicians," Journal of Health & Place, (10):1,pp: 1-11, 2004.

[13] E. M. Geraghty, T. Balsbaugh, J. Nuovo, S. Tandon, "Using Geographic Information Systems (GIS) to Assess Outcome Disparities in Patients with Type 2Diabetes and Hyperlipidemia," J. Am. Board Fam. Med, Vol. 23, pp: 88-96, 2010.

[14] M. A. Horst and A. S. Coco, "Observing the Spread of Common IllnessesThrough a Community: Using Geographic Information Systems (GIS) for Surveillance," J. Am. Board Fam. Med, Vol. 23, pp: 32-41, 2010.

[15] Z. Parker, S. Poe and S. V. Vrbsky, "Comparing nosql mongodb to an sql db," Proceedings of the 51st ACM Southeast Conference. ACM, 2013.

[16] D. Gans, J. Kralewski, T. Hammons and B. Dowd, "Medical groups' adoption of electronic health records and information systems," Health affairs 24.5 (2005): 1323-1333.

[17] K. E. Artnak and M. Benson, "Evaluating HIPAA compliance: A guide for researchers, privacy boards, and IRBs," Nursing outlook 53.2 (2005): 79-87.

[18] B. A. Wolfe et al. "The OpenMRS system: collaborating toward an open source EMR for developing countries," AMIA Annual Symposium Proceedings. Vol. 2006.