# Creating a Modern Electronic Medical Records (EMR) System
# TJHSST Senior Research Project
# Quarter 2
# Computer Systems Lab 2009-2010

Jeremy Chaikind

January 21, 2010

## Abstract

This project will attempt to create a functional, user-friendly medical management and medical records (EMR) system. Web-based programming languages, such as PHP, HTML, and CSS will be used with MySQL databases. Databases will be designed using the Relational Database Model and considering the ACID (*Atomicity*, *Consistency*, *Isolation*, and *Durability*) paradigm. The EMR design will ensure expandability, intuitive interface, and practicality for the end user.

**Keywords:** databases, HIPAA compliance, medical systems, electronic medical records (EMR), web applications

## 1 Introduction and Background

The business of medicine is a topic front and center for many Americans today. Beyond the question of health insurance reform, the United States government is in the process of changing the medical industry itself. Doctors have been given incentives to convert physical, paper charts to electronic ones in the near future. Soon after, physicians will be charged fees for using paper charts. These changes present a difficult situation for doctors. Despite the exorbitant costs of many preexisting Electronic Medical Records (EMR) systems, some popular systems use older programming techniques and languages, and are as a result unintuitive and low-featured. This project plans to remedy the situation by creating an EMR system designed in conjunction with physicians to ensure ease-of-use, using forward-thinking web-based languages, including PHP, HTML, CSS, and MySQL.

## 2 Researcher Experience

Attempting a project of this scale is a difficult undertaking under any circumstances. The researcher's preexisting experience in programming and medical applications makes the task somewhat more reasonable. Prior to beginning this project, the researcher had a strong understanding of the PHP, HTML, and CSS, as well as basic experience with the MySQL and Javascript programming languages that will be used for this project. Within a few weeks, practical MySQL proficiency was cultivated through basic database work.

## 3 Development

### 3.1 Review of Literature

In order to meet medical security standards, the researcher examined HIPAA compliance for physicians

and physicians's offices. Because this project primarily requires technological compliance with HIPAA regulations, an academic article specifically detailing security practices for HIPAA-compliant data transfer of EMR was studied.

After an initial experimentation phase of this project was completed, the exact nature of security measures necessary for HIPAA-compliance was studied. An article about HIPAA-compliant digital storage measures was considered, recommending the implementation of meticulous documentation of actions and backup, as well as suggesting compliance with the National Institute of Standards and Technology (NIST) 800 series of documents, which detail general government guidelines for data storage and security.

Further investigation in this area resulted in discussion with Russell McWey, M.D., a physician at the Virginia Hopspital Center in Arlington, VA who works closely with the hospital IT staff to manage its digital records. Based on a telephone interview with and subsequent typed letter from Dr. McWey, the researcher has determined that no encryption or additional system security is necesary for an intraoffice EMR system, negating the need for NIST compliance. However, an "audit trail" like that mentioned in the HIPAA-compliant storage article is necessary and will therefore be implemented in this project.

The researcher also began the study of modern practices for database management, including the ACID paradigm for database design and the Relational Database model. Initially, the topic was studied by informal work on design with another student of the Computer Systems Lab (Jason Koenig). Exposure to the ACID paradigm continued by studying an article specifically about database design and management. The Relational Database model was further studied in the context of an article about creating a general database system using both the Relational and Object-Oriented models to integrate media in an SQL database.

## 3.2 Theory

To ensure the durability and utility of this EMR system, a server using Linux, Apache, MySQL, and PHP (LAMP) will be used. Unlike many other medical management systems that use older, closed Microsoft database technologies, this EMR will utilize the a more open database model so that the system will be applicable in the future.

The ACID paradigm will also be implemented for this system. Implementation of ACID, an abreviation for *Atomicity*, *Consistency*, *Isolation*, and *Durability*, ensures that information retrieved from a database is always correct.

*Atomicity* specifies that specific database functions must be performed in total or not at all. [3, 289] For example, if a function calls for a database entry to be deleted in one table and added to another, neither database action will occur until both are requested. In this way, should the transaction be interrupted, the entry cannot be deleted in one place without being added to the other. Atomicity prevents database corruption that could provide incorrect information with disasterous results.

*Consistency* states that at all times actions called on the database (assuming Atomicity) leave the database in a correct state. [3, 289-290] While a database that fails to practice Atomicity may crash, allowing the database to fall into an incorrect state, a database that fails to practice Consistency can leave the database in an incorrect state after functioning correctly. As a result, a correctly-functioning Consistent database will never write incorrect data to the database.

*Isolation* demands that all database processes run without knowledge of other functions running concurrently. [3, 290] In a database without Isolation implemented, a user accessing one part of the database could see incorrect data from an intermediate step of an ongoing database process. For example, if one user accessed a patient record in order to call him/her while another user was in the process of changing the patient's phone number, the first user may see the old phone number, the new phone number, or no phone number (if the second user accessed the record while the database transaction was in progress). By implementing the principle of Isolation, no two users could ever access the same record, preventing this problem.

*Durability* ensures the integrity of all data by requiring database data to survive any malfunction. This could be accomplished with relative ease by in-

stituting measures of redundancy and requiring multiple sources to match in order to display information. To prevent data loss on a larger scale, database backup is a necessity. If Durability exists, one can be certain that correctly-entered data will never become corrupted.

The Relational Database model is also important in a modern database design. The Relational model uses a single, global, unique record ID to associate various data to the same individual record. For example, an individual John Doe may have a global ID of 123. Data relating to John Doe will be organized into smaller, more focused tables. In the phone numbers table, there might be three records for global ID 123, representing John's home, cell, and office numbers. Two records for global ID 123 might be present in the addresses table for John's Manhattan penthouse and his country residence. In the medical field, the accuracy of data can literally determine life-and-death situations, so a physician should never be hampered by a strict program design for his/her EMR system. With this in mind, the Relational model will be implemented in this EMR system to ensure the physician has the most flexibility possible in data entry and organization.

# 4 Expected Procedure and Methodology

To program this EMR system, web-based languages, such as HTML, CSS, and Javascript (for the user interface) and PHP and MySQL (for database and other active-web functions) will be used for almost all aspects of the project. Initially, files will be located on a personal remote web server. However, the program will be transferred to a physical server as soon as possible in order to permit security testing to begin.

In order to test the EMR system, false data will initially be used for alpha testing by the researcher. This type of testing will be adequate for evaluating basic functionality of the program. For the program to be effectively tested for intuitive interface design, additional feature requests, and utility for large amounts of data, actual patient data must be used in the context of a physician's office. The researcher plans to test the system in the office of Pediatric Ophthalmologist Melissa Kern, M.D. at the Virginia Hospital Center complex in Arlington, VA. The EMR system functionality and user interface will be designed to best fit the needs of this office. Further testing and application may result in work with Barry Byer, M.D. at the Virginia Hospital Center, a physician who frequently performs mission work in third-world countries. Although his mission would not necessarily benefit from an EMR system, he is in the process of contacting affiliated medical clinics and hospitals in these countries to see if such a system might benefit them.

# 5 Results

## 5.1 First Quarter

First quarter work on the EMR system was largely confined to exploratory work with PHP/MySQL setups. Basic tasks for EMR systems, including adding a patient, searching for a patient, scheduling a patient (Figure 1), and viewing a schedule by month or by week (Figure 2), were implemented. While little code from this experimental phase was used in the final EMR design, implementing EMR *screens* fostered MySQL fluency and understanding while prototypes for the final *screens* were considered. Because these constructions of EMR tasks were not designed to be integrated into the final project itself, unstylized HTML forms were used for the practice *screens.*

## 5.2 Second Quarter

Second quarter work on the EMR system included starting all aspects of program design. Much of this quarter was spent in the design and early implementation phases of the final, "Mander" EMR system. First, a personal web server was created for this project, installing the Ubuntu Server 9.10 Linux distribution on an unused laptop computer. The computer was set up as a LAMP (Linux, Apache, MySQL, and PHP) server, supporting the languages

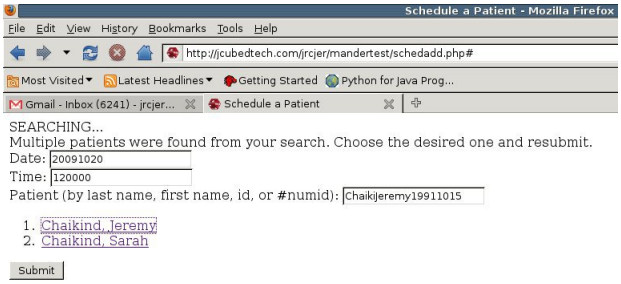


Figure 1: A screenshot of the first quarter Schedule Patient screen.

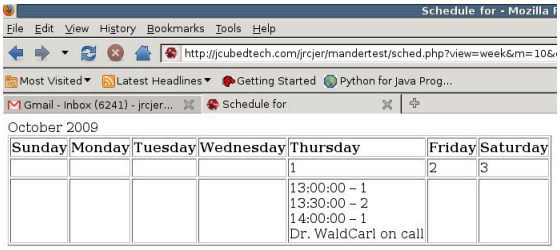


Figure 2: A screenshot of the first quarter Physician Schedule (week view) screen.

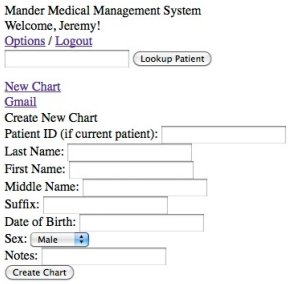


Figure 3: An unstyled screen to add a new patient chart.

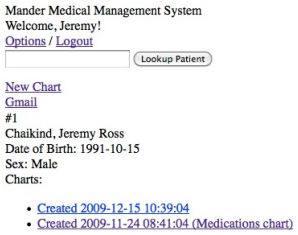


Figure 4: A patient's "Facesheet," displaying all of his/her identifying information and any *Chart*s he/she might have

used in earlier stages of this project.

Next, the final database design for Mander was determined. Based on the relational model, a global user ID would link all patients to a *Names* table, in which crucial information for patient identification, including last, middle, and first names, suffix (if applicable), sex, and any patient notes. If these names were affiliated with system users, their ID would be associated with a username and password in the *SysUsers* table.

More importantly, the IDs for individuals who were patients would be associated with medical records. Patient records would be organized in a hierarchical manner. At the highest level would be the *Chart*, a wrapper designed to organize all related medical information for a patient. Within each chart would be many *Sheet*s, which would represent a specific interaction with a patient. A sheet might, for example, contain a physician's notes from a patient visit or phone call or a scanned document containing a patient's latest blood test results. All the information on a sheet would be organize into *Record*s, which would reflect only a single type of data. A sheet might contain a doctor's templated notes, a plaintext addendum, and a scanned PDF of the patient's last MRI, but each of these three sections, being of different data types, would be stored as a seperate *Record*.

Upon completion of the final Mander design, its implementation began. A login system was first implemented, preventing the accidental discovery and

corruption of this database by others. A variety of basic database functions were coded in an included PHP file, allowing this file to act as a repository for cross-page database access and data retrieval. These functions were tested by a Create New Chart page, which would use the simple functions to create a new row in the *Name* and *Chart* tables if no patient ID was provided or simply in the *Chart* table if the user had specified a patient ID. Later, basic search functionality was added to allow the user to look up a patient's medical data. The patient lookup screen were designed to implement the database retrieval functions at all levels of the EMR hierarchy.

Initial work began on the implementation of a templating system, in which an end user may create a template for which the necessary SQL tables (at the *Record* and/or *Sheet* levels) would be generated automatically. This task is difficult because the end user should not be expected to learn HTML code to stylize his/her template. A What-You- See-Is-What-You-Get (WYSIWYG) approach was first considered. The researcher found an open-source, Lesser General Public License (LGPL), javascript-based WYSIWYG editor, which would enable the end user to use a Microsoft Word-like interface to create a template that would be converted to HTML. However, early tests demonstrated the WYSIWYG to be impractical, as user interface rors made it extremely difficult to use. Further research into parsing Rich Text Files (RTF) resulted in even more disappointing results, as such a process, while relatively simple for basic documents, became unmanageable when advanced features (including document tabs) were used. These difficulties in template management forced the researcher to continue without implementing a template manager.

Finally, basic user interface design began for the Mander EMR. While Figures 3 and 4 do not exhibit any user interface work, navigation bar, page layout, and other tweaks were tested seperately from the rest of the project. However, the actual EMR webpages are coded so that a final user interface could be applied with few changes to the HTML and PHP code.

### 5.3 Testing

For the First Quarter and Second Quarter stages of the project, false data were used. Approximately 20 fake "patients" including the researcher, various friends and family members, and various computer science figures, were added as test data. For the First Quarter, physicians inputed into the system were based on the names of doctors with whom the experimenter is familiar. First Quarter pages were run on a private, remote web server using preinstalled PHP and MySQL support. PHP files were written through a browser-based text editor provided by the company maintaining the web server. MySQL databases were managed through PHP MyAdmin, also preinstalled on the remote server. Second quarter work was written to a personal LAMP server through the Secure Shell (SSH) protocol. MySQL databases were also manages through PHP MyAdmin, installed on the server by the researcher. Further contact was made with Melissa Kern, M.D., so future beta testing within her office may begin in the next stage of this project.

## 6 Discussion

First and Second Quarter work, while relatively unexciting, serves as necessary groundwork for this project. First Quarter work was primarily used as an experiment in PHP/MySQL development and initial prototypes for database organization. Second Quarter work was much more important to the final application itself, creating the LAMP server framework upon which the rest of this project will be based. The creation of a library of database functions will be extremely useful for page-based database interactions, especially those generated by end-user action (i.e. those used to connect user templates to the database). Experimentation in template management served as a first step into finding a method to allow the Mander EMR to become fully expandable, greatly increasing its applicability and lifetime. Finally, basic forays into user interface design will provide a backbone for a final user interface for this program that will make it user-friendly enough for

medical personnel to have little difficulty converting to the new system.

# 7 Conclusion

This project can be expected to yield a system for medical patient management, including the maintenance of electronic charts, as well as any other important patient information. Methods of secure data transfer and integration may be pursued. If this project functions as desired, its value would be immense in a climate where doctors are converting paper charts to digital ones. In a current market where EMR systems with neither major technical prowess nor sufficient medical utility typically cost more than five thousand dollars per physician, a new alternative developed in conjunction with physicians and using modern database technologies could have immense value.

# References


[1] Chaikind, Jeremy. (2009). [Interview with Russell McWey, M.D., physician and technical consultant at the Virginia Hospital Center in Arlington, VA]. This source was used to determine that encryption would not be necessary for an intraoffice EMR to maintain HIPAA compliance.

[2] Davis, J. (2005, February 9). Is your storage management process HIPAA compliant? In TechRepublic. Retrieved from http://articles.techrepublic.com.com/5100-10878_11-5567432.html This source was used as an introductory look into the security measures needed for HIPAA compliance.

[3] Haerder, T., & Reuter, A. (1983, December). Principles of Transaction-Oriented Database Recovery. Computing surveys, 15(4), 287-317. Retrieved from http://portal.acm.org/citation.cfm?id=289.291 This source was used for extensive information about the ACID paradigm and its implementation.

[4] Hristidis, V., Clarke, P. J., Prabakar, N., Deng, Y., and White, J. A., M.D.(2006, November 11). A Flexible Approach for Electronic Medical Records Exchange. Retrieved from School of Computing and Information Sciences, Florida Intenational University website: http://portal.acm.org/citation.cfm?id=1183568.1183576&coll=Portal&dl=ACM&CFID=52786661&CFTOKEN=17622714 This source provided some background into the implications of the HIPAA Privacy Rule on EMR transfer. It also provided some background into methods of data transfer for EMR.

[5] Seyed-Abbassi, B. (1993). Object oriented relational database with SQL interface. In Proceedings of the 1993 ACM Conference on Computer Science (Indianapolis, Indiana, United States, February 16 - 18, 1993). CSC '93. ACM, New York, NY, 497-504. DOI= http://doi.acm.org/10.1145/170791.171128

[6] Summary of the HIPAA Privacy Rule. (n.d.). Retrieved from Office for Civil Rights, US Department of Health and Human Services website: http://hhs.gov/ocr/privacy/hipaa/understanding/summary/index.html This source was used as an introduction to the regulations for physicians' offices created by the HIPAA Privacy Rule.