Creating a Modern Electronic Medical Records (EMR) System
TJHSST Senior Research Project
Quarter 1
Computer Systems Lab 2009-2010

Jeremy Chaikind

October 27, 2009

Abstract

This project will attempt to create a functional user-friendly, HIPAA-compliant medical management and medical records (EMR) system. Web-based programming languages, such as PHP, HTML, and CSS will be used with MySQL databases. A variety of security measures will be explored in order to meet government standards (HIPAA) and protect patient privacy. Databases will be designed using the Relational Database Model and considering the ACID (*Atomicity*, *Consistency*, *Isolation*, and *Durability*) paradigm.

Keywords: databases, encryption, security, HIPAA compliance, medical systems

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 made 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.

The researcher also began the study of modern practices for database management, including and especially the ACID paradigm for database design. 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.

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.

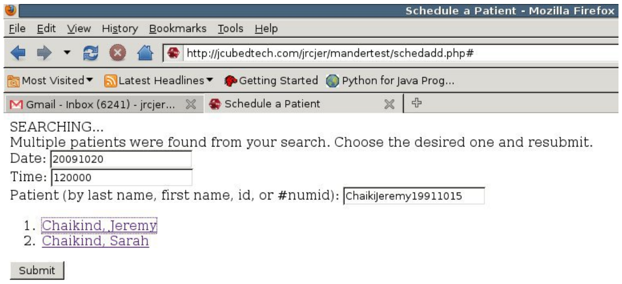
Isolation demands that all database processes run without knowledge of other functions running concurrently. [3, 290] In a database without Isolation implemented, a user accessing one part of the database could see incorrect data from an intermediate step of an ongoing database process. For example, 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 instituting 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.

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.

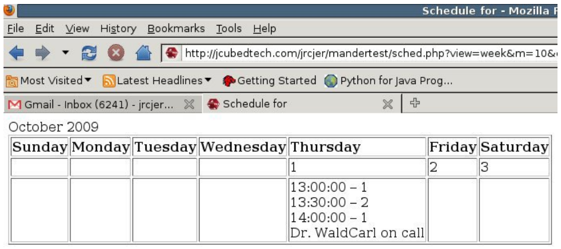
In order to test the EMR system, false data will initially be used for alpha testing by the researcher. This type of testing will be adequate for evaluating basic functionality of the program. For the program to be effectively tested for intuitive interface design, additional feature requests, and utility for large amounts of data, actual patient data must be


Schedule a Patient - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://jcubedtech.com/jrcjer/mandertest/schedadd.php#
Most Visited
Latest Headlines
Getting Started
Python for Java Prog...
Gmail - Inbox (6241) - jrcjer...
Schedule a Patient
Submit

SEARCHING...
Multiple patients were found from your search. Choose the desired one and resubmit.
Date: 20091020
Time: 120000
Patient (by last name, first name, id, or #numid): ChaikiJeremy19911015

1. Chaikind, Jeremy
2. Chaikind, Sarah

Figure 1: A screenshot of the Schedule Patient screen.


Schedule for - Mozilla F
File Edit View History Bookmarks Tools Help
http://jcubedtech.com/jrcjer/mandertest/sched.php?view=week&m=10&
Most Visited
Latest Headlines
Getting Started
Python for Java Prog...
Gmail - Inbox (6241) - jrcjer...
Schedule for

October 2009

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1	2	3
				13:00:00 – 1 13:30:00 – 2 14:00:00 – 1 Dr. WaldCarl on call		

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

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.

5 Results

To date, work on the EMR system has largely been 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 will be present in the final EMR system, implementing EMR *screens* allowed MySQL proficiency and understanding to be cultivated 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.*

5.1 Testing

For this stage in the project, false data were used. Approximately 10 fake "patients" including the researcher, his sister, Steve Jobs, Bill Gates, and Linus Torvalds were added using business addresses and phone numbers. Physicians inputed into the system were based on the names of doctors with whom the experimenter is familiar. The practice *screens* 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.

6 Discussion

Although much of this initial code could not be salvaged for the final EMR system itself, the work done in this first quarter was worthwhile for a number of reasons:

- Programming in a web-based environment reinforced syntax, techniques, and proper coding practices of the internet.
- Programming with PHP and MySQL helped to expand the researcher's knowledge and understanding of both languages and the interfaces between them.
- Despite ignoring style and UI tweaks, creating practice medical *screens* enabled fundamental structures of the user interface and the program itself to be considered.

In addition, research in the areas of HIPAA compliancy, medical data transfer, and general database design will undoubtedly prove invaluable in creating a modern, lasting Electronic Medical Records 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 often cost five thousand to fifteen thousand dollars per physician, a new alternative developed in conjunction with physicians and using modern database technologies could have immense value.

References

[1] 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.

[2] 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.

[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.