J Med Syst (2010) 34:25–33
DOI 10.1007/s10916-008-9212-2

ORIGINAL PAPER

Web-based Multi-center Data Management System for Clinical Neuroscience Research

Alexander Pozamantir · Hedok Lee · Joab Chapman · Isak Prohovnik

Received: 29 May 2008 / Accepted: 2 September 2008 / Published online: 17 September 2008

Abstract Modern clinical research often involves multi-center studies, large and heterogeneous data flux, and intensive demands of collaboration, security and quality assurance. In the absence of commercial or academic management systems, we designed an open-source system to meet these requirements. Based on the Apache-PHP-MySQL platform on a Linux server, the system allows multiple users to access the database from any location on the internet using a web browser, and requires no specialized computer skills. Multi-level security system is implemented to safeguard the protected health information and allow partial or full access to the data by individual or class privilege. The system stores and manipulates various types of data including images, scanned documents, laboratory data and clinical ratings. Built-in functionality allows for various search, quality control, analytic data operations, visit scheduling and visit reminders. This approach offers a solution to a growing need for management of large multi-center clinical studies.

Keywords Management system · E-health · E-collaboration · MySQL · PHP · Database · WWW

A. Pozamantir · H. Lee · I. Prohovnik (✉)
Department of Psychiatry, Mount Sinai School of Medicine, New York, NY, USA
e-mail: Isak.Prohovnik@mssm.edu

I. Prohovnik
Department of Radiology, Mount Sinai School of Medicine, New York, NY, USA

J. Chapman
Department of Neurology, Sheba Medical Center, Tel Hashomer, Israel

Introduction

Modern medical research places severe demands on data management systems. While basic research often involves small, focused datasets, clinical studies often require volumes of data acquired from many subjects and sources. These large patient samples, often comprising hundreds of subjects, usually cannot be found in a single hospital, requiring multi-center, and often multi-country, studies. Even when sampling requirements can be satisfied in a single hospital, rigorous studies often involve the participation of experts from other institutions, as well as auditing and quality assurance measures by external groups. Almost inevitably, thus, data are shared across institutional and national boundaries. This, in turn, creates extraordinary demands for control and coordination of procedures and data management. Further, the current legal atmosphere and the arduous struggle to satisfy ever-increasing expectations of data privacy, as well as the requirements of blind experimental designs, demand that access to the data be strictly limited to individuals with proper privilege. In this article, we describe a web-based data management system developed for longitudinal, multi-center clinical study of Creutzfeldt–Jakob Disease (CJD).

CJD is the most notable of human prion diseases, a group of severe and fatal neurodegenerative disease. Prions are a new group of pathogens, poorly understood, that present significant scientific and public health challenges. They are thought to consist of proteins, devoid of the usual DNA/RNA genetic apparatus, which gain their pathogenic potential, as well as infectious properties and resistance to standard sterilization, solely from conformational changes. In other words, prions are chemically identical to normal bodily proteins, but become abnormal, infectious and toxic by misfolding and achieving an aberrant three-dimensional

structure. Etiology can be infectious, hereditary, sporadic, or iatrogenic, and its symptoms primarily include movement disorders and rapid cognitive decline [1]. The most publicized infectious form is vCJD, mostly encountered in the UK due to consumption of beef contaminated by Bovine Spongiform Encephalopathy, or 'Mad-Cow disease'. Sporadic CJD (sCJD) is the most common subtype of CJD (85~90%), with incidence of about 1/1,000,000/year while hereditary CJD (fCJD) accounts for about 10% of cases worldwide, caused by mutations of the gene encoding the normal form of the prion protein. The most common of these mutations occurs in codon 200 (E200K). There is currently no treatment for any of the prion diseases.

This study was designed to investigate a genetic form of the disease, and in particular to examine the transition from preclinical to clinical stage of mutation carriers. All of the participants of the study were recruited and examined in Israel, which has the world's largest cluster of families affected by the E200K mutation known to cause the disease. The data collected in Israel included medical history, biochemical and genetic tests, structured neurological examinations, neuropsychological tests, and numerous other types of data, as well as extensive MRI imaging data. Participants include three groups: healthy mutation carriers, healthy noncarriers, and symptomatic subjects, and all undergo longitudinal follow-up with periodic examinations. Healthy subjects are examined annually, unless a clinical change or our system indicates the need for more frequent exams, and the CJD patients are examined monthly. The data are inspected for quality assurance, organized and analyzed at the Mount Sinai Medical Center in New York, with the participation of several consultants and collaborators in other US hospitals. All of these investigators require access to the data, at varying degrees of authority and blindness. This article describes the structure, function and capabilities of the system developed to manage these data.

Methods and system description

The planning and specifications for the system were primarily performed by a small group consisting of the overall project principal investigator, the Israeli site principal investigator, the coordinating site system administrator and the system engineer. The initial planning and design phase took about 3 months, prototype implementation about three more months, and then (phase 2) another 6 months of testing, debugging and upgrading. During phase 2, the system was in rudimentary operation, unskilled users were entering data, and their feedback, questions and requests were the basis of final implementation.

Specifications

Reflecting the nature of this research project, the following specifications were defined for the system:

Software and hardware requirements

1. Simple, inexpensive and non-specialized hardware and software platforms, since it must be accessible from multiple locations and the budget was severely constrained by NIH grant funding.
2. Flexible and open code and structure, allowing portability, adaptive adjustments and expansion.
3. Due to network security concerns in Israel, conventional computer communication protocols such as telnet, ftp/sftp, or ssh were not permitted in either direction. This left only the World Wide Web (WWW) protocol as a viable solution.
4. Store and manipulate up to about 1 GB of data per record.

System interface requirements

1. Contain, organize and maintain several types of data, including imaging, genetics, clinical tests and ratings, and personal and demographic information.
2. Maintain several levels of confidentiality, security and privileges, due to the complex nature of our collaboration, the need to maintain blindness for several investigators, and the sensitivity of genetic information. For example, some types of MRI data were acquired in Israel, anonymized at Mount Sinai, rated blindly at Yale University, and integrated with other data back at Mount Sinai.
3. Allow export of data for statistical analysis.
4. Simple and convenient data entry and system operation, accommodating users with minimal computing skills and basic equipment.

Data flow needs

1. Allow data entry, examination, and manipulation from several sites world-wide.
2. The need for data migration should be minimized, preferably eliminated altogether. All intermediary data carriers, such as paper documents, computer files, CDs etc. should be eliminated in favor of online real-time information entry and storage.
3. Being a longitudinal study requiring multiple and variable repeat examinations, remind investigators of pending future visits, as well as alarms changing clinical status.
4. Maintain one central repository for all the research data incoming from various sites.

5. Continuously check and verify data integrity, quality, and completeness, and alert investigators about incomplete data.

Hardware and software implementation

The system was implemented on a single server (Dell Power Edge 2800) equipped with 6.0 Gbytes of RAM, 600 GB of storage (non-RAID) and two Intel Pentium 4 Xeon processors running at 3.0 GHz. Overall diagram of the system structure is depicted in Fig. 1. The server is connected to both local area network (LAN) and the internet through the institutional Mount Sinai firewall using a 1 Gbps built-in Ethernet card and a T1 line. The hardware specification on the server side can be minimal as the number of users is limited and strictly controlled. The server runs the Apache 2.0 (http://www.apache.org/) web server under Linux RedHat version 4 WS (http://www.redhat.com/). The MySQL database server 4.1 (http://www.mysql.com/) is accessed through web pages written in HTML and PHP 4.1. These software packages are all free and included in most other Linux distributions, so that a system like ours can easily be implemented using other Linux based operating systems. Daily backups are performed for the management system, excluding the MRI data, onto a locally mounted DVD-RAM. MRI data are also backed up daily onto another server through network file system (NFS), and the two systems are synchronized.

The hardware requirements for the end users, or clients, are minimal as well: any computer with internet access, running a recent version of any browser. Since not all browsers adhere to a single standard, the data management system does not employ unusual or cutting-edge HTML/PHP techniques to maintain maximum compatibility with all common browsers. It has been tested and found robust with various browsers (Firefox, Safari, and Internet Explorer) running on all common operating system (Mac OS X, Windows XP, and Linux). Data input is reduced to filling out on-screen HTML forms and uploading data files from the user's computer to the server using the browser and explicit point and click standard file dialogs. Therefore the user needs not posses any computer skills beyond standard browser operation.

Server data security

Ensuring the security of research data has been high on our priority list. We adopted a many-tier approach to this problem. The first line of defense against potential intruders through the internet is maintained by the Mount Sinai institutional firewall, which monitors and controls incoming and outgoing traffic within the institution. All network ports and incoming traffic are initially closed by the IT department for security purpose, and can only be opened upon authorized request and approval from the institution. For the data management system, we only opened the https port (# 443) to the public. The https protocol was chosen

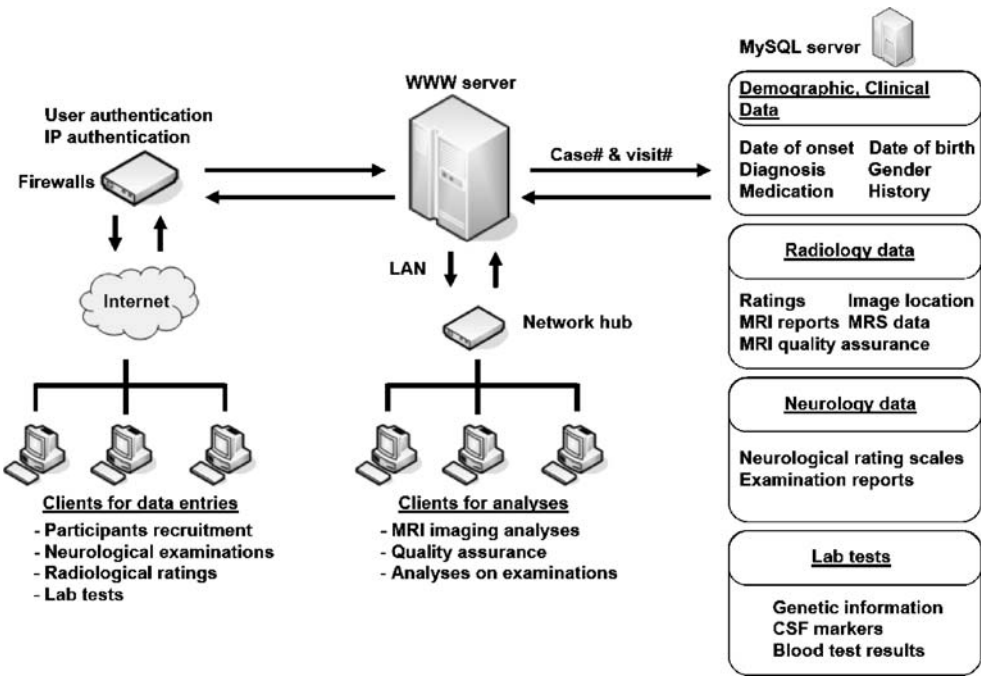

Fig. 1 The client-server configuration of the CJD data management system

over the conventional WWW protocol, http, as it combines additional encryption (128 bit) and an authentication layer of SSL. No other ports are available to standard users, to maximize security: all operations are submitted through https and performed internally by the server. For example, SQL calls are performed by PHP routines invoked by the user through the web browser. To further protect the system, access is only allowed from a defined set of registered computers. This was accomplished by restricting the WWW service to specific computers by IP address, as specified within the Apache configuration file. The last security barrier consists of two levels of password protection before entering into the system.

Although several security measures are taken to prevent any prohibited access, there are always potential intruders seeking to breach server security. As a precaution, we have been monitoring the server logs which store all access data for over 14 months now. So far, no unauthorized access to the server has been granted. To monitor the traffic along with other administrative related tasks, open-source free software for system administration called Webmin (http://www.webmin.com/) is utilized as a way of visualizing all of incoming and outgoing traffics. Physical access to the server is also restricted and only core personnel (four people), who work closely with the project, have access to the room.

User privileges

Only after the user's computer is verified with a proper IP address, the first dialog is displayed where the first-level user name and password are requested. This security level allows access to the web-server, but not yet to the data management system itself. Correct login at this level displays the first page of the system, which requests the second-level login. Successful login then allows access to the system, but only at a specific privilege level mandated by the user identity.

A two-level password system was implemented so that breaches at either the web server or the system level would not be fatal by themselves. The second level also defines the user access privileges. While the single data management system is shared by all investigators, each user holds different privileges to control the ability to view, change or delete data. For example, a radiologist performing blind ratings on the MRI scans has no privilege other than seeing a webpage with a list of anonymized IDs, which leads to the individual radiology rating page. A neurologist is typically allowed to enter, see and modify information related to neurological examinations, but not the imaging data. Thus our PHP code further defines the appearance of information on screen and the user's ability to see or modify it according to specific privileges. A user may be authorized to enter information, but then is not allowed to see or modify it in the future. Other users may be allowed to see some information but not to modify it. Very few users are allowed to view genetic information or delete records. Most users can see and modify information in their own area, but not other areas. At the code level, each browser display is defined by its own HTML code embedded in a PHP file, while all the database access, display functionality and analysis functions are put together into a separate PHP file. This file also includes the implementation of the explicit PHP-based control of user access to various browser displays and functions. Additionally, for certain crucial database records, any change made in the data produces a timestamp and the user ID associated with this change; this details are also stored in the database. This provides improved data maintenance, audit trail and troubleshooting.

Data types and record organization

Currently, the data management system is equipped to handle the following data types:

1. Numerical and alphanumeric data are entered by users into appropriate tables that perform automatic formatting and range testing. Such data can include personal and demographic information (e.g., name, date of birth, date of death, gender, height, weight, address), results of examinations (e.g., blood pressure, neurological findings, neuropsychological test results), and laboratory and genetic findings.
2. Some data exist only as printed documents (e.g., family history, clinical discharge summary, some laboratory reports). These are scanned into PDF files, and the files uploaded into the database and stored internally. Long multi-page documents can be stored this way, and the maximum acceptable file size is assigned by Apache configuration parameters. When requested, the browser uses the default PDF reader to display or print these files.
3. Some data are created online by specialized html tables. For example, MRI scans can be rated online into special forms for both clinical interpretation and quality control.
4. MRI image files are not stored by the db but rather are stored in specialized directories on the server. They are referenced and manipulated using unique pointers generated by the system and associated with each file.
5. Finally, the database calculates and stores values of some data functions. For example, duration of disease to the current visit is computed by subtracting the onset date from the visit date.

The basic records are organized by unique tags of case and visit numbers, and the database structure closely reflects that of the actual examinations performed and data

obtained (Fig. 2). Further, because there are two basic types of subjects, diseased and healthy, the sequential case number is generated automatically in two separate sequences for the two diagnostic groups. In other words, case numbers are assigned sequentially but in two separate series, depending on initial diagnostic group assignment.

The information retrieved by the system is presented to the user for viewing, entering, or editing in a sequence of data screens in the browser. Navigation between the screens is hyperlink-driven in accord with the data structure in the data management system. Upon login, the first screen the user sees allows display of a list of subjects, which can be either all available subjects or a subset corresponding to search criteria. Search criteria provided to the user currently include first or last name (or part of a name), case number, initials, or diagnosis. This listing of subject records contains permanent subject information that does not change throughout the study as shown in Fig. 2. This includes biographical, demographic and social data, as well as the initial diagnosis and disease onset information.

In each subject record, the next level of data is a Visit Record. Each visit is identified by the subject (case number) and visit number. In this study, so far, there have been four distinct groups of medical data associated with each visit: laboratory, neurological, neuropsychological, and radiology test results. Each of these data groups may contain multiple records and tables of data which, in turn, may be divided into smaller groups of data, etc. The general data structure of neurology data is shown in Fig. 3.

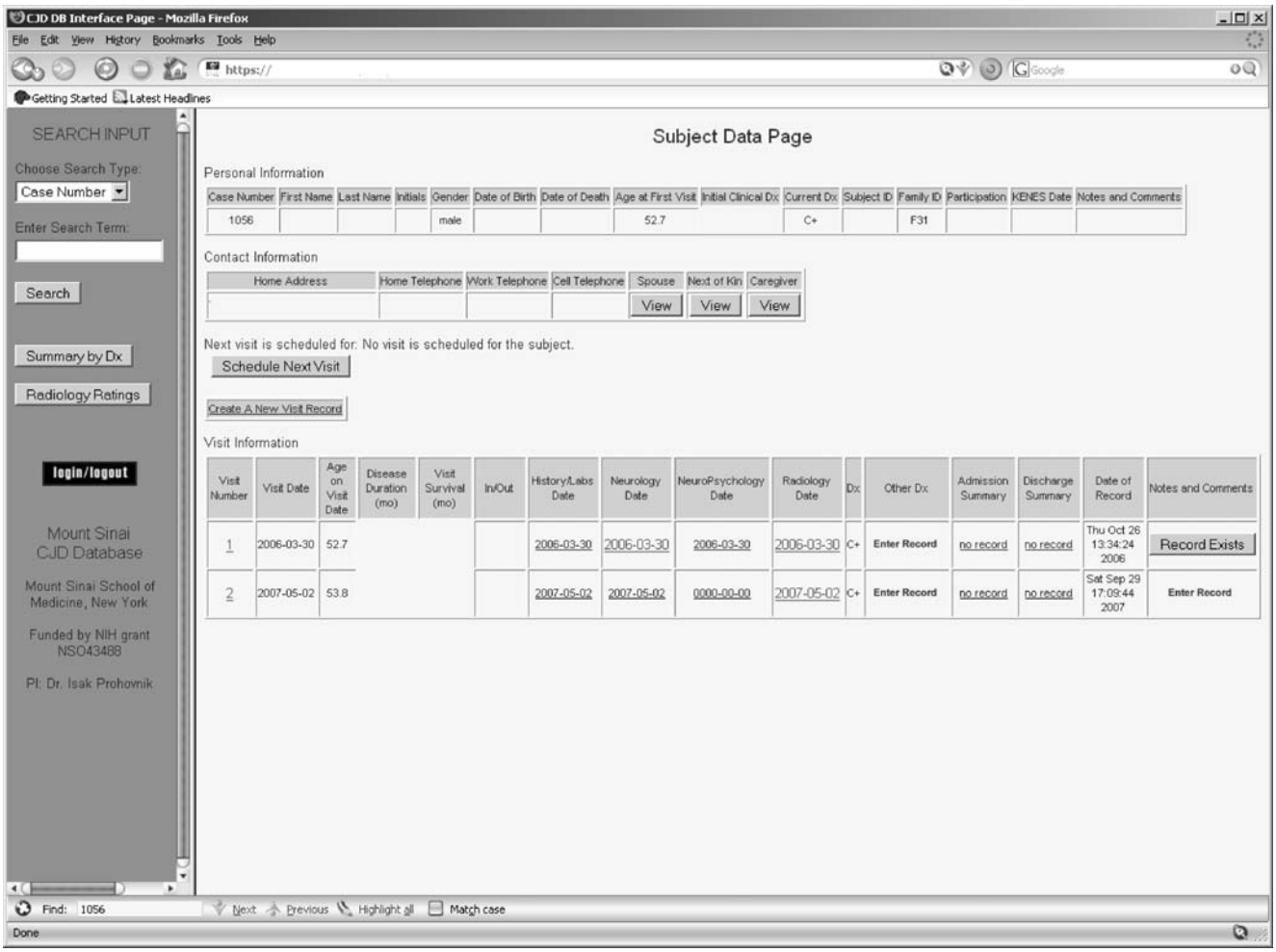

Fig. 2 The 'subject' screen summarizes all data of a single subject, both permanent data (*upper part*) and longitudinal follow-up events (*bottom part*, showing all hospital visits). The *left frame* allows general operations, such as log in and out and search options. The *middle part of the subject page* provides specific administrative operations and alerts related to future visits by this subject. *Underlined text or buttons* are clickable links that lead to more detailed screens and operations. In addition, items may be depicted at different colors or font sizes to indicate status

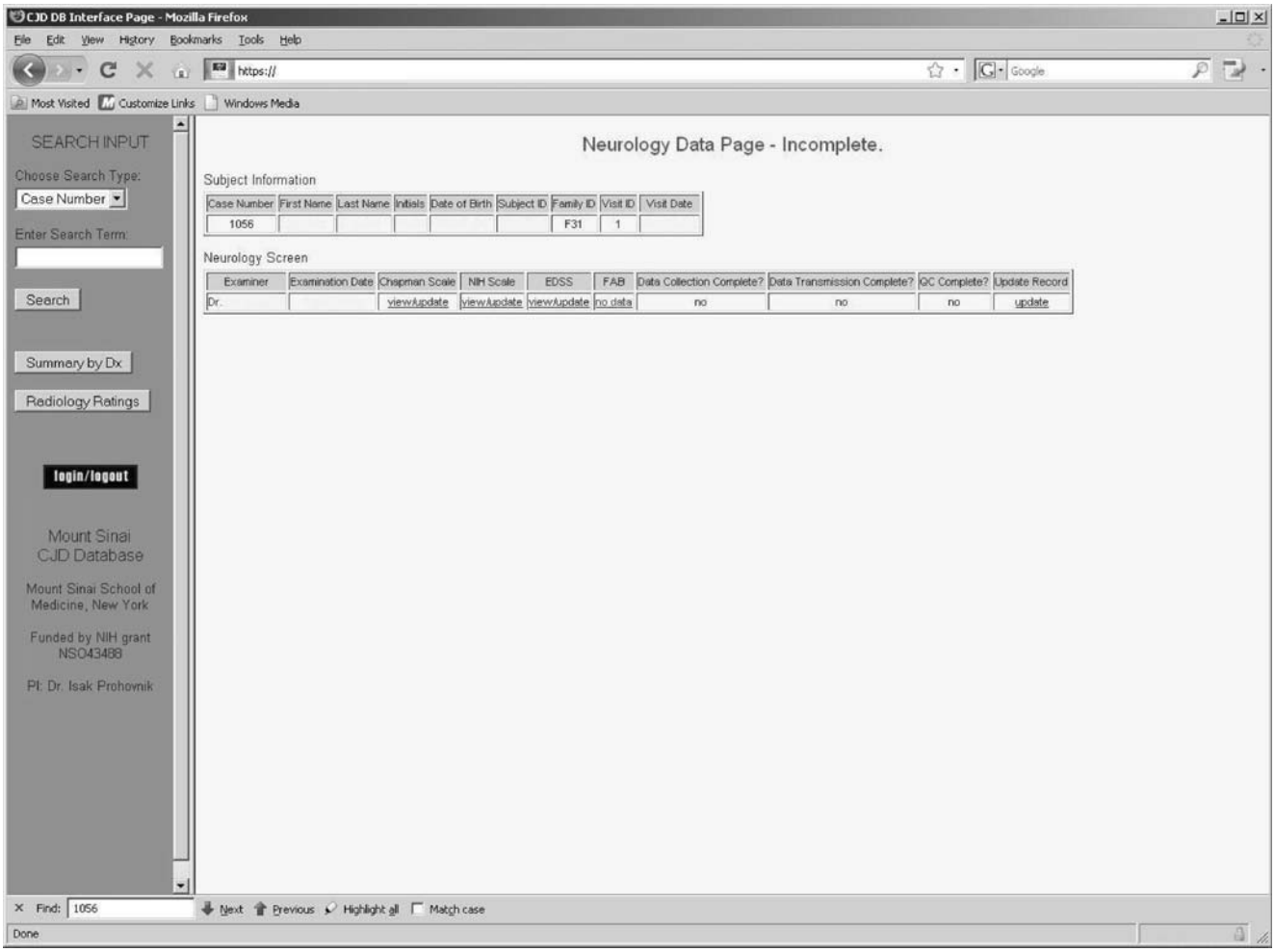

Fig. 3 The 'neurology' screen is the next level of detail, providing a view of one domain of examination of one visit. In this case it details four different neurological rating scales. Clicking any of them leads to the next level of detail that lists the individual symptoms or items of the particular rating scale

Functionality, quality control, and analysis tools

One of the most important issues in any data management system is data quality control (QC). Our system implements QC at three levels to facilitate proper data entry and flexibility to modify data. First, upon data entry in the system, the data type and range are checked and appropriate warnings are issued, if needed, before the data are accepted. Second, a data completeness check is run on the server. Completeness criteria are predefined for various types of records and screens. This may include the existence of valid results for various tests, completeness of personal data records, etc. If the data completeness test fails, the screen appearance will indicate this by a larger red font denoting the item. Finally, the principal investigator has an option to manually override completeness status for the special cases deemed complete even in the absence of all the prerequisites (for example, if a subject dies or leaves the study before all tests are performed). All important data are recorded in the system with the timestamps of their entry which helps ensure data consistency and helps in trouble-shooting and error trapping.

To expedite longitudinal designs, investigators and coordinators need to be reminded to schedule and perform the next visit for each subject, which occurs at varying intervals according to their diagnosis. Automatic visit reminders functionality has been implemented in our system based on three different visit cycles: 14-days, 6-month and 12-month cycle. A scheduling script runs on the system weekly and determines which subjects need to be scheduled for the next appointment according to their visit cycles. The script generates a list of subjects into a text file. A custom Linux shell script delivers this list by email to the appropriate group of researchers who then schedule the visits and record it in the system. For this purpose, we are running the SMTP email server POSTFIX (http://www.

postfix.org/). Similarly, the system can track progression of some parameters across repeated visits of individual subjects, and alert the investigators when certain pre-defined changes occur. This is important to ensure that clinical changes are detected and acted upon immediately. For example, adverse events in a drug trial need to be noted, reported, and possibly initiate action without delay. Notification of clinical changes is sent to appropriate personnel through email, similar to the appointment reminder.

Finally, the system allows both rudimentary and sophisticated statistical analyses. The simple on-line data analysis currently includes summary distribution of subjects according to their initial diagnosis, gender and number of visits. For any desired powerful analyses, all data can be extracted as text files for importation into statistical analysis software.

Discussion

Our data management system was developed by a single software engineer, requiring about 6 months to plan and build a basic functionality. Since the initial development, the system has been debugged and updated continuously to meet evolving needs. In the early phase of development, for example, a challenge was to check and debug the access privileges of different users. As there are hundreds of data values to be entered into the system with various privileges for different users, we invested much of our effort in verifying system integrity. Although it was a time consuming process, close relationship and frequent communication among the end users, design team and the developer helped expedite this process. The system has now been fully operational for about one year, and meets its design goals. To date, about a dozen users have been granted privileges. While all reported minor bugs and requested minor interface and workflow changes, all have been satisfied and productive. These users include individuals with very little prior computer experience, but learning to interact with this system was easy, since it only requires basic browser skills (point and click) at the most basic level of operation.

All users were able to productively interact with the system after 2–3 h of initial training, although further questions always arose later. All have been satisfied overall with the system, although requests for minor adjustments, improved user interface and more features are still being submitted, and usually accommodated. In this regard, an in-house system has the distinct advantage of being able to meet such requests quickly, and to allow immediate dialog between developers and users.

Other than ease of use and accessibility for any hardware and software clients, our high priority was in security of the system. Given the sensitive nature of medical records involved, the system should ensure strong data security and it should comply with all the legal and technological aspects of medical data management at the various research sites. Security here includes three major concerns. First, the usual concerns about any web server on the internet. Second, maintaining the confidentiality of protected health information, mandated by the recent HIPAA legislation in the USA but also part of general Helsinki and Institutional Review Board guidelines. Third, in our specific case, maintaining the experimental structure and blindness controls that ensure trial validity. We also wanted the system layout and functionality to be intuitive and closely reflect the data structure of the actual medical studies performed. Its functionality should readily accommodate such various needs as statistical data analysis, data exporting, ensuring data blindness for appropriate investigators, and automatically perform certain recurring duties such as scheduling visits, checking data completeness, etc.

Before deciding to build our custom system, we searched for solutions available, academically or commercially. A growing number of health care and medical research groups report their results on development and implementation of such systems in their respective fields [2–8]. Much of the recent literature, however, is concerned with sharing data of a uniform nature across multiple studies [2, 9]. In our case, we are engaged in a multi-national, multi-center, longitudinal clinical study which includes collection of sensitive medical information and many data types. The multi-center nature and the large quantity of data impose unusual requirements for data organization, entry, retrieval and manipulation, as well as confidentiality of subject information and maintaining blindness required by the study design. We are a clinical research group with minimal IT support, and had no wish to ‘reinvent the wheel’; our first plan was to find a commercial or academic platform that could be adapted to our needs. Our search for appropriate products was not limited by operating system constraints. Our laboratory is solely based on Unix variants, with Macintosh and Linux computers. For this critical study, however, we were even willing to consider Windows-based solutions. However when cost, performance, and scalability were considered, as well as user-friendliness and web inter-applicability, we could not find any adequate commercial product. Of the commercial systems, some (e.g., Oracle) could satisfy our objectives but were very expensive and, once implemented, not easy to continue to modify and evolve. The database manager closest to meeting our requirements was FileMaker Pro 8 [10]. It is a popular relational model for small business which is capable of sharing data over local area network as well as WWW. It supports both Macintosh and Windows operating system with minimal hardware requirement. The software package also includes all necessary tools to launch the system and requires little

prior knowledge of system structure and computer programming skills. However, when it was tested over WWW, there was a severe lag in speed and response to any data entry in our test, and its responsiveness was considered insufficient for our needs. Similarly, we could not find any appropriate system already developed by the academic or open-source community, and decided to deploy our own data management system. Although a custom-developed system has its own disadvantages, such as long debugging period, we weighted our decision based on our budgetary constraints and flexibility to tailor the system interface to meet our needs [11]. Once we decided to develop our own solution, we chose to do it over Linux due to its wider availability of open-source tools and its lower cost.

Finally, we realized at the outset that as the system undergoes constant improvements and the medical research changes in scope, system modifications are unavoidable. Research questions are answered over time, and new questions are posed; hypotheses are tested, discarded, and new ones formed. These all require changes in methodology and data collection, which necessitate modifications in data storage and analysis. As our research technology and goal evolve to answer new questions, a system has to be modified and new privilege as well as new functionality must be reviewed and developed. For example, when we started the research, the only available MRI technology consisted of magnets with 1.5 T static field strength. We have now upgraded to a 3 T magnet, so a new data field must be defined to indicate the field strength at which each record was obtained. Similarly, when we started the study, subjects were only classified genetically on the presence or absence of the E200K mutation. We have now realized that a genetic polymorphism at another location on the same gene (the codon 129 M/V polymorphism) may influence our data, so this classification must also be considered. New MRI techniques, and new ancillary laboratory procedures as well as clinical rating scales, are constantly added, and old ones become obsolete.

This places high value on adaptive qualities of the system—the users will constantly interact with the development team to reassess the needed functionality. The latter requirement and the requirement of scalability weighed in heavily in our choice of using MySQL—PHP open source implementation for the data management system. While not competing head-on with the latest available SQL-based commercial packages, the open source solution proved to be more than adequate for the size and scope of this study. The PHP-based interface allowed for easy modifications of the system functionality such as ever changing data completeness requirements, changing subject visit scheduling procedures, data access privileges, statistical data analysis, and new data integration. We currently hold data of about 160 subjects and 310 visits. On average, medical records of a single subject occupy 0.8 Mbyte in our system. In addition to the medical records, a MRI (200 Mbytes) scan is obtained at each visit and is analyzed through a chain of image processing steps, which requires additional 800 Mbytes. Thus the system allocates nearly 1 Gbytes of disk space for each visit. On line storage requirements are reasonable for modern servers and will not limit the growth potential of our system.

When we decided to develop our own system for this study we looked at similar solutions implemented by other research groups. The basic architecture of our system is similar to that described in Minati et al. [5]. The two systems share the same basic architecture: an Apache server driven by PHP code and servicing a MySQL with various online and offline applications ported to the PHP suite. Due to the complexity of the imaging data, and the fact that the storage architecture is not yet standardized for cutting-edge imaging sequences, we chose not to store them directly into our system, in contrast to other systems [2, 5]. Instead, the files are stored separately, with only pointers to their location and meta-information about their status contained in the database records themselves. Our study requires an extra emphasis on real-time data completeness and quality control. The many built-in checks and procedures are designed to ensure correctness and timeliness of the data which are critical for a project spanning multiple centers, countries and continents. Another characteristic feature of our study is the existence of large standardized structures, such as an on-line radiology ratings template containing several hundred unique data fields in one record. This requires an additional set of on-screen checks and color-coding schemes to provide proper QC. It seems appropriate here to emphasize the importance of planning out the database data structure to the finest possible detail before the actual programming phase begins, while allowing for the necessary future flexibility in this structure. Our experience shows that optimizing this process is a key in reducing the time it takes to bring the database system from design stage to production stage.

The current system has several limitations. Handling scanned documents (PDF files) is still fairly primitive. While the system can accept long multi-page documents (this was accomplished by adjusting the appropriate parameter in the php configuration file), it cannot modify them once they have been accepted. Thus, if a page has to be added to the document at a later date, this can only be done by deleting the document and re-uploading a modified document. Clinical history is often unpredictable and difficult to categorize before admission. We have originally defined only four such variables: two including text only (patient progress notes and visit progress notes) and two for scanned documents (admission letter and discharge letter). Clinical courses are individual, however, and there are

additional categories of progress notes and clinical information that do not have a predefined field. Examples include sleep study reports, EEG and evoked potentials, and other ancillary tests. Currently, these are lumped together with either admission or discharge documents, but managing them is inefficient. All reports have to be individually coded in php—the system has no built-in reporting or data analysis functions. Text entry is limited by SQL requirements, so that multilingual or special characters may be problematic. Coding each individual screen may require significant manual effort, depending on its complexity, because there is no built-in editor to facilitate GUI development. Some major operator mistakes are not yet recognized by the system and may require significant efforts to correct. For example, if the operator mistakenly enters a later visit before an earlier one, it may require significant work to correct. The user interface and work flow are not yet optimized. In particular, we are still in the process of adding warnings for user actions that may be dangerous and adding intelligence to the work flow to minimize the necessary steps and number of clicks for each action sequence.

Conclusion

We have presented a data management system with a unified data entry and QC mechanisms for multi-center clinical research. The system is based on PHP-MySQL suite of code and is run by Apache server on a Linux machine. The system features round-the-clock multi-user availability over the Internet, and constant data access and modification monitoring. It achieves reduction in direct and indirect costs of data collection and managing by reducing work load on human resources and mailing overhead while ensuring high data security and quality control. The system is scalable (additional users and additional research topics can be readily added to the existing structure) and portable (it should work on any Apache server with PHP and MySQL on any Unix/Linux platform) and adaptable (can be run on Windows, Macintosh and other Unix variants). Transferring data entry duties to each participating center also allows for reduction and more even distribution of data entry work load among the various centers. Most importantly, the system requires no computer skills for basic data entry and retrieval beyond standard browser use.

Acknowledgements This work was supported by NIH grant # NS043488. We thank Janet Ben-Mordechai, RN, Ilana Seror, B.Sc., and Vered Luufman-Malkin, CSW, for their assistance with the development and utilization of the system.

References

1. Johnson, R. T., and Gibbs, C. J. Jr., Creutzfeldt–Jakob disease and related transmissible spongiform encephalopathies. *N. Engl. J. Med.* 339:1994–2004, 1998. doi:10.1056/NEJM199812313392707.
2. Hasson, U., Skipper, J. I., Wilde, M. J., Nusbaum, H. C., and Small, S. L., Improving the analysis, storage and sharing of neuroimaging data using relational databases and distributed computing. *Neuroimage*. 39:693–706, 2008. doi:10.1016/j.neuroimage.2007.09.021.
3. Kelly, K. J., Walsh-Kelly, C. M., Christenson, P., Rogalinski, S., Gorelick, M. H., Barthell, E. N. et al, Emergency Department Allies: A web-based multihospital pediatric asthma tracking system. *Pediatrics*. 117:S63–S70, 2006. doi:10.1542/peds.2005-0794.
4. Laule, O., Hirsch-Hoffmann, M., Hruz, T., Gruissem, W., and Zimmermann, P., Web-based analysis of the mouse transcriptome using Genevestigator. *BMC Bioinformatics*. 7:311, 2006. doi:10.1186/1471-2105-7-311.
5. Minati, L., Ghielmetti, F., Ciobanu, V., D'Incerti, L., Maccagnano, C., Bizzi, A. et al, Bio-image warehouse system: Concept and implementation of a diagnosis-based data warehouse for advanced imaging modalities in neuroradiology. *J. Digit. Imaging*. 20:32–41, 2007. doi:10.1007/s10278-006-0859-2.
6. Thriskos, P., Zintzaras, E., and Germenis, A., DHLAS: A web-based information system for statistical genetic analysis of HLA population data. *Comput. Methods Programs Biomed.* 85:267–272, 2007. doi:10.1016/j.cmpb.2006.11.005.
7. Zuberbuhler, B., Galloway, P., Reddy, A., Saldana, M., and Gale, R., A web-based information system for management and analysis of patient data after refractive eye surgery. *Comput. Methods Programs Biomed.* 88:210–216, 2007. doi:10.1016/j.cmpb.2007.09.003.
8. Knop, C., Reinhold, M., Roeder, C., Staub, L., Schmid, R., Beisse, R. et al, Internet based multicenter study for thoracolumbar injuries: A new concept and preliminary results. *Eur. Spine J.* 15:1687–1694, 2006. doi:10.1007/s00586-006-0135-7.
9. Van Horn, J. D., Grethe, J. S., Kostelec, P., Woodward, J. B., Aslam, J. A., Rus, D. et al, The Functional Magnetic Resonance Imaging Data Center: (fMRIDC): The challenges and rewards of large-scale databasing of neuroimaging studies. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 356:1323–1339, 2001. doi:10.1098/rstb.2001.0916.
10. Gorman, M. J., Jacobs, B., Sloan, M., Roth, Y., and Levine, S. R., A web-based interactive database system for a transcranial Doppler ultrasound laboratory. *J. Neuroimaging*. 16:11–15, 2006.
11. Tsay, B. Y., and Stackhouse, J. R., Developing a management information system for a hospital: A case study on vendor selection. *J. Med. Syst.* 15:345–358, 1991. doi:10.1007/BF00995973.