Medical Student Database Development:  
A Model for Record Management in a Multi-Departmental Setting

Darin M. Vercillo¹, Kevin C. Holmes¹, Matthew J. Pingree³,  
Bruce E. Bray M.D.¹,², and Michael J. Lincoln, M.D.¹,².

University of Utah Medical Center, Salt Lake City, Utah 84132  
¹School of Medicine  
²Department of Medical Informatics  
³Medical College of Wisconsin, Milwaukee, Wisconsin 53226

Student records flow through medical school offices at a rapid rate. Much of this data is often tracked on paper, spread across multiple departments. The Medical Student Informatics Group at the University of Utah School of Medicine identified offices and organizations documenting student information. We assessed departmental needs, identified records, and researched database software available within the private sector and academic community. Although a host of database applications exist, few publications discuss database models for storage and retrieval of student records. We developed and deployed an Internet based application to meet current requirements, and allow for future expandability. During a test period, users were polled regarding utility, security, stability, ease of use, data accuracy, and potential project expansion. Feedback demonstrated widespread approval, and considerable interest in additional feature development. This experience suggests that many medical schools would benefit from centralized database management of student records.

INTRODUCTION

Electronic record keeping continues to change and improve the practice of medicine and medical education. The electronic student record, like the electronic medical record, can quickly provide up-to-date, accurate records to those requiring their use. Implemented as a centralized repository, a computerized database can serve numerous offices and organizations, reducing the incidence of inaccessible and disparate information.

The Medical Student Informatics Group (MSIG) at the University of Utah School of Medicine consists of students pursuing advancement in medical information systems, curriculum reform, and technology awareness. MSIG acts under the supervision of faculty within the Department of Medical Informatics and the Office of the Dean. A core group of students identified student record maintenance limitations within administration offices. The core group approached staff, faculty, and students to assess the need for improvement in this area.

The applicability of a relational database solution was reviewed. The MSIG team gathered and processed records, and programmed a web-based record management application. The software was deployed for use in several administrative, hospital, and student areas. The application employs multiple security levels necessary in this environment. Peer and staff feedback has led to revisions, improvements, and increased scope and scale of the project. This paper presents the analysis preceding the design phase, development methods used, and a discussion of user satisfaction, project findings, and future goals.

BACKGROUND

Ten years ago the American Association of Medical Colleges (AAMC) acknowledged that, "...each school should develop a database that complements the information that the SAIMS (Student and Applicant Information Management System) database can provide." Many schools are only beginning to undertake this task. Certainly, the task of creating an integrated, multi-user information system can appear daunting. Furthermore, a paucity of literature exists that clearly explains the development process.

In 1995 came a call for AAMC standards. After researching student records at a number of
institutions, Michael Stocks noted that the majority of student records were "...on paper, were poorly organized, and were missing critical information." A standardized student record could "...be used to explore and even shape the future of medical education in America."

The institution constructing a centralized records database has three basic development options:

- Purchase a multipurpose software package and tailor it to specific needs.
- Contract a database consultant to design a proprietary system.
- Develop the system with in-house personnel.

Each alternative has benefits and disadvantages. The literature is unclear as to the best solution. In 1998, the AAMC Group on Student Affairs reviewed variations on commercial personnel database systems. Such packages can tackle a variety of institutional requirements, and generally offer both predefined utilities and the flexibility to create custom designs.

Database consulting groups abound, ranging tremendously in experience and reliability. Ideally, the institution and consultant work together from the ground up. Addressing departmental needs on an individualized basis can minimize data conversion and user apprehension.

Finally, for an institution with qualified personnel, the project can be developed in-house. The experience serves to educate those involved, and often acts as a bridge to future development. We determined to approach our student information database in this manner.

Each of these options can vary greatly in the development process, and in overall expense. They are not mutually exclusive, however, and an institution may elect to employ a combination of the three.

**DEVELOPMENT METHODS**

**Needs Assessment**

Students, student organizations, staff, faculty, and associated health care personnel were all identified as potential users. Each group was reviewed for the type of student information they already stored, and the manner which they utilized these records. The design team also inquired about successful and unsuccessful elements of previously used software. The information gathered from the potential users formed the groundwork for the application development.

**Source Identification**

We approached each organization that regularly utilized student records, and those that anticipated following some aspect of student data. Varied copies of similar information, defined by the particular needs of each group, were being maintained with only occasional cross-referencing. Earlier complaints of discrepancies between a published student directory, the school web-site, and the official Dean’s office records were already strong indications for centralizing this information.

We initially assessed the various methods used to store student records. The medical school office staff tracked personal and contact information in word processor files and on printed media. The student body maintained directory information on a secured, static web page, updated on a yearly basis. Other health sciences departments generally distributed new lists of contact information at the beginning of classes and rotations. Some database tables, spreadsheets, and delimited text files had been derived from office data, and were in use by the various student organizations.

![Figure 1: Data combined from multiple sources form a single standardized record set.](image)

Each organization had requested updated records from another office during the previous quarter. The consensus between staff, faculty, and students interviewed was that greater accuracy would certainly result and efficiency likely improve, as a result of a centralized database.

**Gathering and Processing Data**

The utility of any database depends on the integrity and currency of its contents. We gathered the most current student information from various sources (Figure 1) and combined
these archives to initiate our database. Entry errors, name contractions, and stylistic differences often confused the records. We utilized OCR software to speed data extraction from printed media. Word processing, spreadsheet, and database software was utilized to trim unwanted characters and group common records.

**Normalization**

Valuable data is not useful unless stored in a manner that is easy to manipulate. *Normalized* data is information prepared so that it can be uniquely identified and retrieved. The records from which we drew were in essence large, unmanageable files. We parsed these records into fields such as first, middle, and last names, street address, city, state, spouse, offices held, etc. This information was then combined into an ordered table structure. Approximately four thousand incomplete files were ultimately combined to create approximately six hundred detailed records.

To fully normalize a database, a unique identifier must be associated with each record. Our database tracks students’ personal, academic, and extracurricular records. The early choice for primary key was the student identification number. Most students use their social security number as the student ID, virtually guaranteeing uniqueness in our setting. Given the security and legal pitfalls associated with using social security numbers, we later opted to generate distinct auto-incrementing identification numbers. Most databases programs will generate an incrementing integer to serve as the primary key.

**Relational Architecture**

A single table could include all the data desired for each student. Over time, as the number of records and quantity of data increase, a single table grows unwieldy. Relational databases link tables sharing information about common subjects. We placed essential information (name, SSN, class) in a principal table. Subsequent tables now store contact information, enrollment records, grades, club affiliation, etc. When performing a query, a common field in each table, usually a primary key, links the active tables (Figure 2).

![Figure 2: Queries between tables are linked using a common field, usually a primary key.](image)

Each organization can create a new table to track their unique information. It is imperative to store data elements in one place only, as entry duplication opens the door for error. By employing a clean relational architecture, disparate information is eliminated.

**Client Software Development**

Many expressed concerns about possible limitations accessing records, tailoring output, and securing data. To address these and future design needs, we asked students, office staff, and faculty to submit specifications for a computer based student record management system. Top issues raised included accessibility, adaptability, maintenance, and security.

**Front End Interface**

The majority of end-users concern themselves only with functionality and easy of use. Each office specified various tools, queries, and reports they desired. While designing the user interface, we reviewed successes and failures from previous applications, and kept simplicity as our motto.

Our first database utilized Microsoft® Access for the simple design mechanism and a straightforward user interface. Despite the ease of use, we soon encountered several limitations. Each workstation required a fully installed copy, dramatically inflating cost. Network mapping increased setup and maintenance time. Finally, frame server access was not adequately established to make the application universal under this environment. We reconsidered our initial approach and decided that network restrictions and platform incompatibilities critically limited wide spread deployment. A web-based model emerged as the solution to many of these problems.
Internet Database Development

The current client software is a web-based application administrated on the MSIG Internet server. The server operates under a Windows NT 4.0 environment and runs Allaire's Cold Fusion application server. Web-based design offers universal availability, both on and off campus. We combined HTML, CFML (Cold Fusion Markup Language), and JAVA to create an interface that operates on common browsers, regardless of platform.

![Figure 3: A record search form with security login. Security levels defined in the database protect sensitive data.](image)

The application provides search tools (Figure 3) for students, faculty, and other campus personnel. Users can retrieve simple contact information, or sensitive personal records and grades depending on security access. Several offices were interested in generating printable lists based on current data. Many are now available, and serve across various departments (Figure 4).

![Figure 4: This menu yields dozens of lists based on simple radio button driven queries.](image)

Since the site became active, the project has expanded beyond providing contact information and lists. Two exams, four student forums, attendance sheets, class catalogues, and residency applications have gone online. Most users are already comfortable with the browser environment, so training time is minimal and generally focuses on content issues.

Security Issues

Data security generally parallels availability through standard university channels. Pager numbers are freely available. More detailed contact information requires a basic username and password known to faculty and students. A designated staff member who issues usernames, passwords, and security levels maintains specific user accounts. Record editing requires high level access, where list-generating functions are accessible to general office staff (Figure 5).

![Figure 5: A student record editing form. Modification of such data is restricted to a few designated personnel.](image)

Some faculty and students expressed concern about security pitfalls associated with Web transmission. We utilized both CFML and NT security to prevent unauthorized access. Secure Sockets Layer (SSL) transmission guards against data interception. These measures protect data from both unauthorized viewing, and potentially damaging attack.

Internet Database Connectivity

To utilize a database via the web, an application processes calls to "open" or "registered" databases on the server. We investigated JAVA, CFML, Visual Basic, and various CGI scripts to handle database calls. CFML programming proved versatile and intuitive. Figures 6 and 7 illustrate simple CFML query and output code.

```cfml
<CFQUERY name="Detail" datasource="Students">
SELECT Last_Name, First_Name, Year, Pager, Email FROM SOM_Students WHERE Year = #PreserveSingleQuotes(MSCLASS)#</CFQUERY>
```

![Figure 6: Cold Fusion Markup Language utilizes SQL text to form server side databases queries.](image)
**Figure 7:** HTML pages are dynamically generated based on output from database queries.

**User Reviews**

With the web application now functioning for five months, we analyzed use and user feedback. 5216 general and 1931 detailed searches were performed. 74% of searches originated from campus computers or dial-up connections. Dean’s office functions (e.g., for editing student information, creating mailing lists) were accessed 843 times. Eighteen technical support calls were logged with respect to the system. Two students objected to their information being available to authorized campus users, so their records were further restricted. The Office of Student Affairs has fielded numerous information update requests, and amended the files online. Recently, staff and students approved the addition of new search tools and an online residency letter utility. This positive and progressive feedback attests to the strength of the web design.

**FUTURE DEVELOPMENT**

Each year MSIG recruits additional students to maintain current systems and continue development. New student participation adds innovation, but requires continuing education. The three-year student overlap and ongoing faculty involvement provides continuity. MSIG holds regular meetings with Department of Informatics faculty and the Office of the Dean. These meetings serve to plan MSIG objectives, provide development updates, and discuss the feasibility of new projects. Communication between students, faculty, and administration ensures the stability of the project.

The success of the student record database has served to expand the scope of our development efforts. Related projects continue to surface. This summer (1999), MSIG students sponsored by the Department of Medical Informatics and supported by National Library of Medicine training funds will locally mirror the AAMC CURRMIT database to facilitate a curriculum analysis and medical school calendar system. Additionally, the admissions department will construct a database to track candidates directly from AMCAS data, through medical school, and ultimately to alumni status. Time will undoubtedly produce a host of valuable utilities.

**CONCLUSIONS**

Electronic record keeping improves information handling, accessibility, and overall utility. A centralized database best serves to handle these records. In this project, we found that a student record database is ideally suited to start this process. Properly processed departmental archives produce a single coherent data source. Relational table design enhances expandability. A web-based user interface streamlines development and provides excellent user access. Finally, feedback driven design ensures a straightforward, user friendly application. The future of medical school information management lies in database integration across multiple office and departments. Such progress will enhance every aspect of medical education.

**References**


**Acknowledgements**

The authors would like to thank the University of Utah Department of Medical Informatics and the School of Medicine Office of the Dean for their encouragement in undertaking this project.