



Design and Implementation of a Web-Based Student Information Management System

Kenjiro Mardaniyan

Department of Chemical and Process Engineering, Andaman Polytechnic for Technology and Trade, Thailand

kenjiro.mardaniyan@aptt-th.net

Peer Review Information

Submission: 22 Jan 2019

Revision: 07 Feb 2019

Acceptance: 23 March 2019

Keywords

Student Information System, Web-Based Application, Database Management; Role-Based Access Control, Educational Management

Abstract

Student information is generated continuously through admission, course registration, attendance, assessment, progression, and graduation. When these records are maintained in isolated spreadsheets or paper files, duplication, delayed reporting, inconsistent updates, and limited access become common. This paper presents a methodology for the design and implementation of a web-based Student Information Management System (SIMS) that centralizes academic and administrative records through a role-based browser interface. The proposed system follows a design-science-oriented development process comprising requirement analysis, three-tier architecture design, relational database modeling, module implementation, security controls, and functional/performance verification. Core functions include authentication, student profile management, course and semester records, attendance, marks, search, and report generation. A lightweight prototype was evaluated using 2,000 synthetic student records and local HTTP requests. All CRUD and reporting functions completed successfully; mean response times for the tested operations were approximately 0.70–1.08 ms on a loopback test environment. The approach emphasizes data integrity, controlled access, usability, maintainability, and extensibility. The study shows how a structured web architecture can reduce administrative fragmentation while providing timely and consistent student information to authorized users.

Introduction

Educational institutions must maintain accurate information about students across the complete academic life cycle. Typical data include personal details, enrollment status, course allocation, attendance, internal assessment, examination results, and progression history. Manual or semi-computerized approaches make it difficult to keep these records synchronized, especially when several departments update the same student profile. Early student information management research therefore emphasized integrated database design, modular functions, and user-friendly interfaces as important

requirements for dependable administration (Liu, Wang, & Zan, 2010).

Web technology is particularly suitable for this problem because authorized users can access a centralized application through a standard browser without installing a separate desktop program. Tang and Zhang (2009) demonstrated the value of web-service-based student information management and role-based access, while Lv, Li, and Gao (2014) described campus-network deployment as a means of automating university student administration. Ouyang (2014) similarly reported that database-backed

student information systems can simplify insert, query, update, and delete operations and improve management efficiency.

A successful student information system is not only a database. It is an information-system artifact that must satisfy organizational requirements, produce reliable information, remain usable for nontechnical users, and protect academic records. The DeLone and McLean (2003) information-system success model identifies system quality, information quality, service quality, use, and user satisfaction as central evaluation dimensions. Technology acceptance research also indicates that perceived usefulness and perceived ease of use strongly influence adoption (Davis, 1989), while Venkatesh et al. (2003) extend this view by emphasizing performance expectancy, effort expectancy, social influence, and facilitating conditions.

Accordingly, this paper develops a practical methodology for a web-based SIMS with three objectives: (1) centralize student records in a relational database; (2) provide secure, role-specific functions to administrators, faculty members, and students; and (3) verify the design through functional and performance testing. The scope is limited to core academic information management and does not include online fee payment, learning-management content, or biometric attendance hardware.

Literature Review

Tang and Zhang (2009) designed a college student information management system using web services and highlighted database design, functional modules, and role-based access control. Their work is directly relevant to a modern browser-based architecture because it separates user access from centralized data services. Liu et al. (2010) focused on system functionality, database design, and front-end application development, reinforcing the need for a coherent relationship between the user interface and persistent student records.

Later studies increasingly emphasized networked and modular deployment. Lv et al. (2014) proposed a student management information system for a campus network environment to automate information handling and improve sharing. Ouyang (2014) implemented student information management using Java technology and emphasized the fundamental CRUD operations required for maintaining student records. Widiartha and Aristana (2018) developed a web-based system for managing student grades and computerized

recap reports, demonstrating the practical importance of browser-accessible academic data processing.

The student-affairs perspective broadens the system beyond a simple student table. Abdulazeez, Zeebaree, and Sadeeq (2018) described an electronic student affairs system containing multiple institutional service modules and evaluated it with a usability questionnaire. Their reported work supports the inclusion of service workflows, role separation, and user-centered evaluation. For methodology, Hevner et al. (2004) describe information-systems design science as the creation and evaluation of useful artifacts. This is appropriate for SIMS development because the research output is a working architecture, database, application process, and evaluation procedure rather than only a descriptive analysis.

Usability is also essential because administrative software is often used by staff with different levels of technical experience. Lewis (2018) reviews the System Usability Scale (SUS) as a widely used standardized measure of perceived usability. Although the present study concentrates on functional and technical prototype testing rather than a full human-subject usability experiment, SUS is retained as the recommended instrument for future institutional validation. Taken together, the literature suggests five design requirements: centralized structured data, modular web functions, role-based security, user-oriented interfaces, and systematic evaluation.

Methodology

The study follows an artifact-centered design methodology influenced by design-science principles (Hevner et al., 2004). The procedure is organized into six sequential stages: requirements analysis, architecture design, database design, interface and module design, implementation with security controls, and verification. Iteration is permitted between stages when test results reveal missing requirements or data constraints.

1. Requirement Analysis

Requirements were derived from common academic record workflows described in earlier student information system studies (Tang & Zhang, 2009; Liu et al., 2010; Abdulazeez et al., 2018). The administrator requires account management, student registration, course/semester configuration, record correction, and reporting. Faculty users require controlled access to attendance and marks for

assigned students. Student users require read-only access to their own profile, attendance summary, and results. Nonfunctional requirements include responsive browser access, unique student identifiers, referential integrity, input validation, authorization, auditability, backup capability, and acceptable response time.

2. System Architecture and Database Design

A three-tier architecture is adopted to reduce coupling between presentation, business logic, and data storage. The presentation layer contains

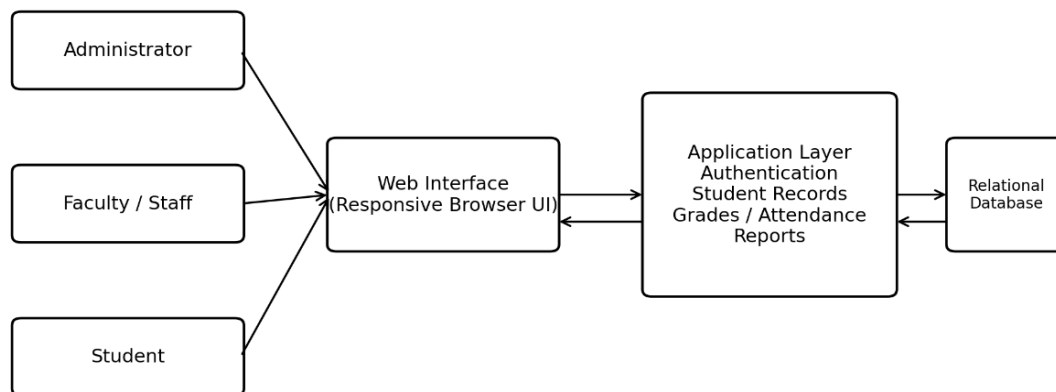


Figure 1. Three-tier architecture and principal user roles of the proposed SIMS.

3. Functional Modules and Security

The application is divided into authentication, student profile, academic structure, attendance, assessment/results, search, and reporting modules. Authentication establishes the user identity; authorization then limits each request according to role. Administrator accounts can create and correct institutional records, faculty accounts can update only permitted academic records, and student accounts can view only records associated with their identity. Passwords should be stored as salted hashes, sessions should expire after inactivity, and all production traffic should use HTTPS. Server-side validation is required even when client-side validation is present. Parameterized queries, least-privilege database accounts, routine backups, and audit logs further reduce risks to confidential educational records.

4. Implementation and Verification Procedure

For methodological verification, a lightweight local prototype was constructed around an HTTP service and a relational SQLite database. The logical design remains portable to production databases such as MySQL or PostgreSQL because the schema and operations use standard relational concepts. A synthetic dataset of 2,000

responsive browser pages. The application layer validates requests, enforces authorization, executes academic rules, and prepares reports. The data layer stores normalized relational entities such as User, Student, Department, Course, Enrollment, Attendance, Assessment, and Result. Primary keys uniquely identify records, while foreign keys connect academic transactions to the correct student and course. Unique constraints on roll number and email reduce duplicate registration; indexes on roll number, department, semester, and course improve retrieval.

student records was generated across four departments and eight semesters; no real student information was used. Black-box test cases covered create, retrieve, search, update, report, and delete operations. After warm-up requests, operation timings were measured through local loopback HTTP requests. This setup measures application/database processing overhead rather than Internet latency and should therefore be interpreted as a controlled prototype benchmark, not a campus-wide production guarantee.

Evaluation criteria were derived from information-system quality concepts (DeLone & McLean, 2003). Functional correctness was checked by comparing requested operations with database state, information quality was checked through unique identifiers and relational constraints, and performance was summarized using mean and 95th-percentile response time. Future acceptance testing should add perceived usefulness/ease-of-use measures based on Davis (1989), UTAUT constructs from Venkatesh et al. (2003), and a standardized SUS questionnaire as discussed by Lewis (2018).

Results

The prototype completed all defined CRUD and reporting test cases successfully on the synthetic dataset. Table 1 summarizes measured local response times. Retrieval, search, insertion, update, and deletion were all below 1 ms mean response time in the test environment, while the grouped department report averaged 1.08 ms.

The 95th-percentile values remained close to the corresponding means, indicating stable execution under the single-machine benchmark. These values should not be generalized to real deployments because network delay, concurrent users, hardware, database engine, encryption, and dataset size will change observed performance.

Table 1. Controlled prototype functional and local response-time results (2,000 synthetic records).

Operation	Test Purpose	Mean (ms)	95th pct. (ms)	Outcome
Retrieve student	Fetch one record by roll number	0.70	0.89	Pass
Search students	Prefix search across student identifiers	0.76	0.94	Pass
Add student	Insert validated student record	0.73	0.86	Pass
Update student	Modify semester and CGPA fields	0.84	1.04	Pass
Department report	Group count and average CGPA	1.08	1.29	Pass
Delete student	Remove designated test record	0.74	0.87	Pass

Beyond speed, the architecture addresses the principal weaknesses of fragmented record keeping. Centralized data prevents separate departmental copies from becoming inconsistent; role restrictions limit unnecessary access; validation reduces malformed or duplicate records; and report queries transform stored data into usable administrative information. These design outcomes are consistent with earlier work that emphasized centralized database operations and web accessibility (Lv et al., 2014; Ouyang, 2014; Widiartha & Aristana, 2018).

The results also identify limitations. The benchmark used one host, synthetic data, and no concurrent workload. It did not evaluate human usability, long-term maintainability, accessibility compliance, failure recovery, or security penetration resistance. Therefore, the next evaluation stage should deploy the system on a realistic server, conduct concurrency and backup/restore tests, and collect structured feedback from administrators, faculty, and students. A SUS-based evaluation would provide a comparable perceived-usability score, following the measurement guidance discussed by Lewis (2018).

Conclusion and Discussion

This paper presented a step-by-step methodology for designing and implementing a web-based Student Information Management System. The proposed approach begins with stakeholder and workflow requirements, converts those requirements into a three-tier architecture and normalized relational schema, implements modular browser-based functions,

applies role-based security and validation, and verifies core operations through controlled tests. The prototype benchmark demonstrated correct CRUD, search, and reporting behavior for 2,000 synthetic records, with low and stable local processing times.

The main contribution is the integration of technical design and information-system evaluation into one methodology. Direct student-management studies show the practicality of web access, modular functions, and centralized databases (Tang & Zhang, 2009; Liu et al., 2010; Abdulazeez et al., 2018), while information-system theory explains why quality and user acceptance must also be evaluated (DeLone & McLean, 2003; Davis, 1989; Venkatesh et al., 2003). Thus, implementation should not be considered complete when database transactions merely work; the system must also deliver accurate information, protect records, support the users' tasks, and remain understandable in day-to-day academic administration.

Future work should replace the local prototype database with a production relational database, add audit dashboards and automated backup policies, test concurrent users, and conduct an institutional pilot. Mobile-responsive views, notification services, document upload, transcript generation, and integration with learning-management or finance systems can be added as separate modules. A formal usability study using SUS and user-acceptance measures would strengthen evidence about adoption and satisfaction. With these extensions, the methodology can support a scalable student information platform while preserving the core

principles of centralized data, controlled access, and reliable academic reporting.

References

- Abdulazeez, A. M., Zeebaree, S. R. M., & Sadeeq, M. A. M. (2018). Design and implementation of electronic student affairs system. *Academic Journal of Nawroz University*, 7(3), 66–73. <https://doi.org/10.25007/ajnu.v7n3a201>
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340. <https://doi.org/10.2307/249008>
- DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems*, 19(4), 9–30. <https://doi.org/10.1080/07421222.2003.11045748>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>
- Lewis, J. R. (2018). The System Usability Scale: Past, present, and future. *International Journal of Human–Computer Interaction*, 34(7), 577–590. <https://doi.org/10.1080/10447318.2018.1455307>
- Liu, Z., Wang, H., & Zan, H. (2010). Design and implementation of student information management system. In 2010 International Symposium on Intelligence Information Processing and Trusted Computing (pp. 607–610). IEEE. <https://doi.org/10.1109/IPTC.2010.160>
- Lv, Y., Li, X., & Gao, L. N. (2014). The design and implementation of student management information system under the campus network environment. *Advanced Materials Research*, 998–999, 1725–1728. <https://doi.org/10.4028/www.scientific.net/AMR.998-999.1725>
- Ouyang, G. X. (2014). Design and implementation of student information management system based on Java technology. *Applied Mechanics and Materials*, 687–691, 2474–2476. <https://doi.org/10.4028/www.scientific.net/AMM.687-691.2474>
- Tang, Y.-F., & Zhang, Y.-S. (2009). Design and implementation of college student information management system based on Web Services. In 2009 IEEE International Symposium on IT in Medicine & Education (Vol. 1, pp. 1044–1048). IEEE. <https://doi.org/10.1109/ITIME.2009.5236230>
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3), 425–478. <https://doi.org/10.2307/30036540>
- Widiartha, K. K., & Aristana, M. D. W. (2018). Sistem informasi pengelolaan nilai mahasiswa berbasis web pada STMIK STIKOM Indonesia. *Joutica*, 3(1), 109–116. <https://doi.org/10.30736/jti.v3i1.197>