

Design and Development of a Web-Based Research Information Management System for Student Research in Artificial Intelligence and Computational Science

Chalerm Sak Saenglee

Master of Science Program in Software Engineering,
College of Arts, Media and Technology, Chiang Mai
University,
Chiang Mai, Thailand
kaidohjung@gmail.com

Pathathai Na Lumpoon

Department of Software Engineering, College of Arts,
Media and Technology, Chiang Mai University,
Chiang Mai, Thailand
pathathai.n@cmu.ac.th

Abstract— This work presents the design, development, and evaluation of the Research Information Management System (RIMS), a web-based platform for managing student research in the Artificial Intelligence and Computational Science program at Chiang Rai Rajabhat University (CRRU), covering bachelor's to doctoral levels. Previously, research was tracked with spreadsheets, e-mail, and paper-based feedback, which scattered submission versions and gave executives no program-wide view of progress. Key features include role-based access for five user groups, step-wise document submission, two-way advisor feedback, a seven-state research status workflow with full history, publication approval, e-mail notifications, an executive dashboard, and a public research catalogue. A review of five university research-tracking systems and three general-purpose tools was conducted to identify gaps and guide the design. Development followed the ISO/IEC 29110 Basic Profile with three iterations, producing 33 use cases and 318 traceable software requirements. The system passed all 792 test cases and 927 automated tests, served up to 200 simulated concurrent users in a local test environment without failed requests, and achieved an overall user satisfaction of 4.51 out of 5 ($SD = 0.68$) from 32 evaluators. The results show that the proposed system centralizes research data, keeps every submission version in one place, and gives executives and administrators a transparent, auditable view of research progress.

Keywords—Research Information Management System, Student Research Tracking, ISO/IEC 29110, Role-Based Access Control, Requirements Traceability, Web Application

I. INTRODUCTION

In the Artificial Intelligence and Computational Science (AI&CS) program at Chiang Rai Rajabhat University (CRRU), student research from bachelor's to doctoral level has been supervised without a dedicated system. Advisors track progress in Google Sheets, Microsoft Excel, or Notion; students submit documents by e-mail and shared files; feedback is given on paper or by e-mail; administrators keep offline spreadsheets; and executives find it difficult to see a program-wide picture of research progress. As a result, submission versions and supervision history are scattered, progress is difficult to audit, and completed student research has no systematic channel for public dissemination. Existing research-tracking systems address only parts of this process and mostly target graduate students (Section II). Moreover, because the system is built by a single developer, a lightweight yet systematic and traceable development process is needed.

A management information system (MIS) collects, processes, and stores operational data to support managerial decision-making [1]. Following this concept, this study develops a web application that consolidates the whole research workflow, from proposal to publication, in one place, and summarizes it for decision-makers. To keep the development process systematic and auditable for a one-developer project, the study adopts ISO/IEC 29110, the software lifecycle profile for very small entities (VSEs) [2], [3]. The objectives are to: (1) design and develop RIMS (Research Information Management System), a web application for managing and tracking student research progress at all degree levels; (2) apply ISO/IEC 29110 so that planning, requirements, design, and testing are systematic and traceable; and (3) evaluate system performance and user satisfaction for each user role. Corresponding to these three objectives, the main contributions of this work are: (1) a web-based system spanning the full supervision loop, from step-wise submission and advisor feedback to approval and publication, deployed across all degree levels in a live academic program; (2) an experience report on applying ISO/IEC 29110 in a one-developer project, including requirement-level test coverage review; and (3) empirical evidence of the system's performance and acceptance, obtained from load testing and role-based user acceptance testing with all users of the program across five roles.

The paper is structured as follows: [Section II](#) reviews related work and existing systems; [Section III](#) presents the system design (Objective 1); [Section IV](#) describes the implementation and testing under ISO/IEC 29110 (Objective 2); [Section V](#) evaluates the system (Objective 3); and [Section VI](#) concludes the paper and outlines future work.

II. RELATED WORK

This section reviews research supervision and information systems in the literature, research-tracking systems used at Thai universities, and the tools currently used in the Artificial Intelligence and Computational Science (AI&CS) program.

A. Research Supervision and Information Systems Putra et al. [4] developed SIMITA, a web-based thesis monitoring information system that replaces manual monitoring forms and uses QR codes to deliver thesis progress information quickly, with the aim of supporting on-time graduation. Wong et al. [5] developed PostConnect, a web-based system for tracking and managing postgraduate research progress, with modules for

proposal defense, supervision management, progress monitoring, candidature monitoring, and a student data dashboard for students, supervisors, and faculty. Azeem et al. [6] developed a faculty dashboard that consolidates teaching, research, and budget data to support decision-making by department heads. Yan et al. [7] observed that existing web-based systems mainly monitor the administrative milestones of thesis management rather than supporting the research itself, and designed a system that supports the master's thesis research process and knowledge sharing. At the university level, current research information systems (CRIS), including Elsevier Pure [8], Clarivate Converis [9], and the open-source DSpace-CRIS [10], support institution-wide management of researcher profiles, projects, outputs, and reporting.

Each of these works addresses only part of the supervision process: [4] and [5] focus on progress monitoring, [6] on executive reporting, and [7] on research support, while [4], [5], and [7] each target a single degree level. CRIS platforms target faculty research administration rather than the day-to-day supervision of student theses, and they require commercial licensing or substantial deployment effort, which is beyond the means of a single academic program.

B. Research-Tracking Systems in Thai Universities To examine these needs in the local context, five research-tracking systems used by Thai universities were reviewed. The SNRU Thesis Advisor Database (TAS) at Sakon Nakhon Rajabhat University is simple to use: advisors list their students and view an overview of each thesis, with status updates approved by an administrator [11]. Mahidol Scholarship and Awards Online tracks the research of scholarship students, focusing on publication milestones that are conditions of funding, and displays all steps on a single screen; status is updated only by advisors and program chairs [12]. In Walailak Graduate Progress Tracking, advisors manage thesis submission status across programs, and students see only a “pass” or “fail” result for each step [13]. The PCRU Information System for Graduate Advisors at Phetchabun Rajabhat University shows students' research activities as an overview across all graduate programs, with progress represented only by the dates of status changes [14]. VRU e-Graduate at Valaya Alongkorn Rajabhat University is an integrated graduate-school portal covering everything from registration to graduation, including thesis examination status, but its many menus make it complex to use [15]. Table I compares these systems with the proposed system.

TABLE I.
RESEARCH-TRACKING SYSTEM COMPARISON

Feature	T	M	W	P	V	R
Undergraduate support	X	X	X	X	X	✓
Step-level progress	X	✓	✓	✓	✓	✓
Advisor–student feedback	–	X	X	X	X	✓
Public research dissemination	X	X	X	–	–	✓
Executive overview	–	–	✓	✓	✓	✓

a. T = SNRU TAS, M = Mahidol Scholarship Online, W = Walailak Graduate Tracking, P = PCRU Advisor IS, V = VRU e-Graduate, R = the proposed system (RIMS); “–” = not reported in the review. “Step-level progress” means that the status of each research step is shown, even if only by date or examination status.

C. Tools Currently Used Interviews with AI&CS advisors showed that they currently track student research with Google Sheets, Microsoft Excel, or Notion. Spreadsheets are general-purpose, and although Notion is more flexible, configuring it

for a supervision workflow is complex. Table II compares these tools with the proposed system.

TABLE II.
COMPARISON WITH TOOLS CURRENTLY USED

Feature	GS	XL	N	R
Role-based permissions	X	X	✓ ^b	✓
Research publication	X	X	X	✓
Interactive submission	X	X	✓ ^c	✓
Categorization and grouping	X	X	X	✓
Progress timeline	✓	✓	✓	✓
User notifications	X	X	✓	✓
User management	X	X	X	✓

a. GS = Google Sheets, XL = Microsoft Excel, N = Notion, R = Research Information Management System. b. Via separate page access. c. Possible but complex. A feature is marked ✓ only if it is available as a built-in, research-specific function without add-ons or custom scripts.

D. Research Gap As Tables I and II show, existing systems mainly serve graduate or scholarship students, keep advisor feedback outside the system, and provide no systematic channel to disseminate student research, while general-purpose tools lack research-specific roles and workflows. No reviewed system covers the full supervision loop of step-wise submission, feedback, approval, and publication from bachelor's through doctoral levels. RIMS addresses these gaps with a single workflow that records every submission and comment in one place and publishes approved research to the public. It builds on the reviewed works by combining step-wise progress monitoring as in [4], role-specific modules for students, supervisors, and faculty as in [5], an executive dashboard as in [6], and advisor feedback that extends monitoring toward the research support called for in [7].

III. SYSTEM DESIGN

To meet the first objective, this section presents the software architecture, the functional design by user role, and the database design of the proposed system, named RIMS (Research Information Management System).

A. Software Architecture

RIMS follows a three-tier client–server architecture (Fig. 1). The presentation tier is a responsive single-page application built with React 19, TypeScript, Redux, and Tailwind CSS, and it communicates with the back end through the native Fetch API. The user interface supports five roles—Administrator, Executive, Advisor, Student, and Guest—and displays data and functions according to each role's permissions.

The application tier runs Node.js with the Hono framework and exposes a RESTful API. It is organized into Routes (endpoints and HTTP methods), Services (business logic), and Repository (data access) layers. Authentication uses JSON Web Tokens (JWT) and bcrypt-hashed passwords, while role-based authorization controls access to protected data and operations. Administrators manage users and system settings and approve publication; executives monitor overall research progress; advisors review submissions and provide feedback; and students manage their research projects and submit work at each stage. Guests can access published research without logging in.

The data tier uses MySQL accessed through Drizzle ORM to store users, roles, research projects, submissions, feedback, and activity histories. The back end and database are

containerized with Docker and deployed on a cloud virtual private server managed with Plesk. E-mail notifications keep students and advisors informed of project updates without requiring them to log in.

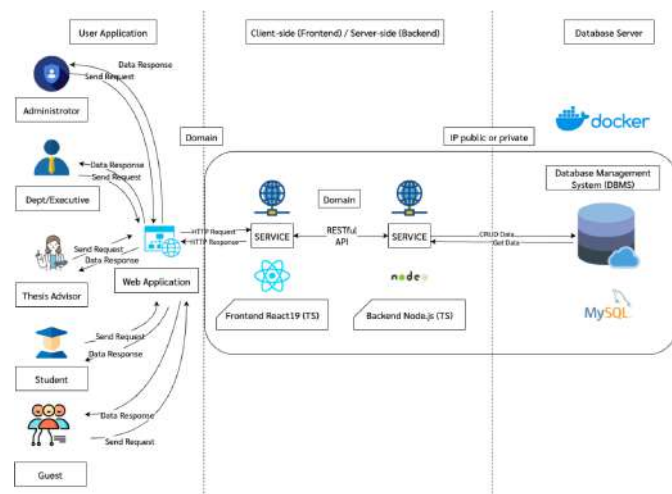


Fig. 1. System architecture of RIMS.

B. Functional Design

Requirements were elicited through semi-structured interviews with students, advisors, and executives. They were specified as 33 use cases, 33 user requirements (URS), 318 software requirements (SRS), and 12 non-functional requirements covering security, performance, reliability, and usability. Five roles are separated under role-based access control (RBAC). Each research project moves through seven statuses: draft, in progress, awaiting review, reviewed, awaiting publication approval, published, and cancelled. Every transition is recorded in a status history. The main user interfaces are shown in Figs. 2–4, and the main features per role are as follows:

1) *Administrator*: creates user accounts for executives, advisors, and students, assigns their roles, and activates or deactivates each account; a new account can sign in only after the administrator approves it as active. The administrator also maintains master data (degree levels, research types, academic years, semesters, and research steps), sets the single current academic year, and approves the publication of research.

2) *Student*: registers a research topic, uploads proposal and thesis files (PDF or Word) for each step, and tracks status and overall progress. Students read and reply to advisor feedback and receive notifications (Fig. 2). Every uploaded file is renamed, so earlier versions are never overwritten.

3) *Advisor*: views supervised students, reviews each step's submissions, and writes feedback that students can reply to as a thread. The advisor then approves a step or returns it for revision, and uploads the publication file of completed research for publication approval (Fig. 3).

4) *Executive*: monitors the program through a dashboard showing the number of projects by degree level, progress by step, publication ratio, and advisor workload. The executive can export filtered reports to Excel or PDF (Fig. 4).

5) *Guest*: searches and browses published research in the public catalogue without logging in. Only research that has already been approved for public release by the university is published in RIMS, after a further approval by the administrator. The catalogue therefore acts as a program-level

showcase that filters the university's research down to that of the AI&CS program, which helps disseminate student work and increase its value.

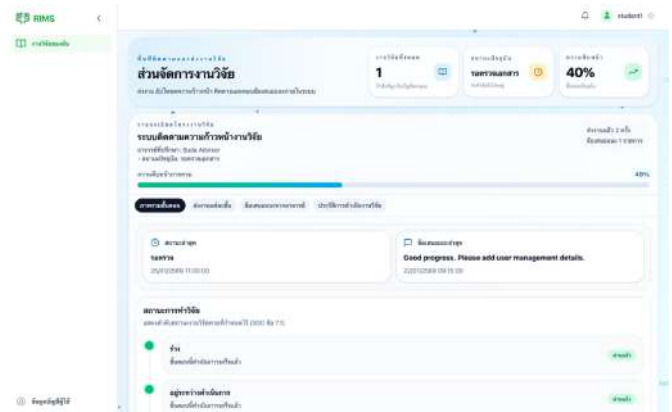


Fig. 2. Student research workspace.

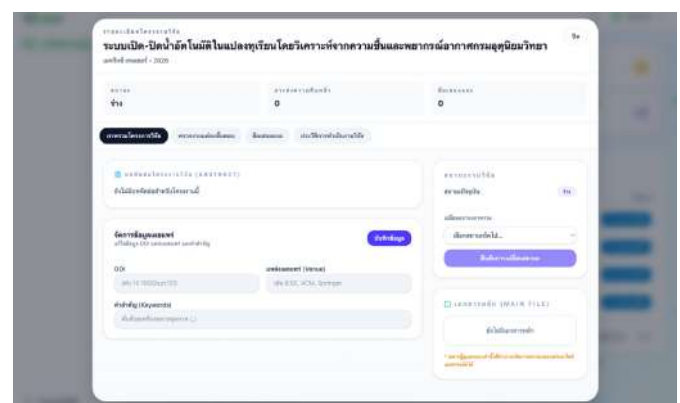


Fig. 3. Research project detail and status control.

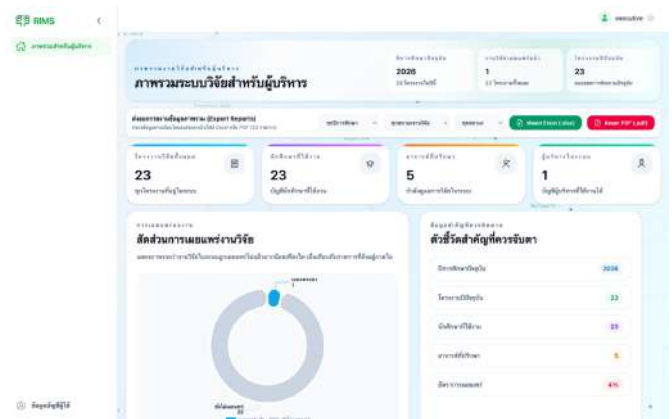


Fig. 4. Executive overview dashboard.

C. Database Design

The database is a normalized relational model of 17 tables. The core tables are users, student profiles, staff profiles, research projects, research project students, project step submissions, project feedbacks, feedback replies, research status history, and publication approvals. Supporting tables include academic years, system master data, notifications, notification event settings, notification delivery settings, email digest items, and audit logs. A user may own several research projects over time, and each project records every step submission as a separate row with its sequence and review result. Together with the status history and audit logs, this gives

a complete audit trail. The high-level entity–relationship diagram is shown in Fig. 5.



Fig. 5. High-level entity–relationship diagram.

IV. IMPLEMENTATION

To meet the second objective, development followed the ISO/IEC 29110 Basic Profile, integrating the Project Management and Software Implementation processes [2], [16] within an iterative development model comprising three iterations. Seven pilot case studies reported by Laporte et al. [17] illustrate the practical application of the standard and its guides in very small entities, providing context for its adoption in RIMS development. The third iteration focused on research tracking. The development process produced eight work products: a project plan, a conflict and configuration management plan, a software requirements specification (SRS), a software design description covering the architecture, class diagram, data dictionary, sequence diagrams, and UI design, a traceability record linking UC→URS→SRS→test cases, a test plan, a test record, and a user manual. No formal change requests were raised during the three development iterations. The SRS comprises 318 software requirements derived by decomposing the user requirements into fine-grained, testable specifications. The codebase was organized into two repositories, frontend-rims and backend-rims, developed using Visual Studio Code and maintained under Git version control.

Testing was performed at the unit, system, and acceptance levels, supplemented by coverage-gap, display and response-time, and security tests (Table III). Unit tests (UTC-01 to UTC-33) and system tests (STC-01 to STC-33) were derived from the SRS. A requirement-level coverage review showed that the original test set covered only 73.0% of the 318 software requirements (232 of 318). Therefore, 101 gap-closing test cases (GTC-001 to GTC-101) were added, which revealed three previously undetected defects. All test cases were converted into automated suites; the final automated run contains 927 tests because some planned cases were split into several automated checks. The final run comprised 640 unit and integration tests (Jest), 15 database tests against a real MySQL instance, and 272 end-to-end tests (Playwright); the suites also contain regression tests added when observations were closed. The 39 security tests are part of these suites and cover three OWASP Top 10:2021 categories [18]: broken access control (26 route-level RBAC tests), cryptographic failures and sensitive-data exposure (7 tests that confirm password hashes

and account fields are not returned beyond each role), and identification and authentication failures (3 middleware and 3 token tests). Fifteen defects were logged: 14 were fixed and retested, including one security defect, and one was closed as not a defect. All 20 test observations were resolved.

TABLE III.
SUMMARY OF TEST RESULTS

Test set	Cases	Passed
Unit test (UTC-01 to UTC-33)	384	384
System test (STC-01 to STC-33)	295	295
Coverage gap tests (GTC)	101	101
Display and response-time tests	12	12
Total planned test cases	792	792
Automated: Jest unit/integration	640	640
Automated: Jest + MySQL database	15	15
Automated: Playwright end-to-end	272	272
Total automated tests, final run	927	927
Security tests (within automated)	39	39

V. EVALUATION

The system was evaluated for performance under load and for acceptance by real users. The evaluation addresses the third objective and covers all five user roles. Section V-A examines whether the system can serve the program's users under load; Sections V-B and V-C assess whether each role can complete its supervision tasks; and Section V-D compares RIMS with the previous fragmented workflow described in Section I. Section V-E reflects on the second objective by reporting lessons from the ISO/IEC 29110 testing process. Because the guest role requires no login, its tasks were performed by all participants on the public website.

A. Performance Under Load

A load-testing script simulated 1, 10, 25, 50, 100, and 200 concurrent users. Each user repeated a role-specific scenario five times after a warm-up, and end-to-end time was measured for each scenario. The production VPS applies rate limiting and flood protection that throttle high volumes of requests from a single source, so the target load could not be generated against it. The test was therefore run in a local environment with the same application build and a MySQL 8 database seeded with 23 research projects and 32 users. The results provide a baseline for the application and database tiers, excluding network latency and the production protection layer. Performance evaluation using technical metrics has also been applied to n-tier cloud-based software services [19]. No request failed at any load level. At 50 concurrent users, the 95th-percentile (p95) response time was 444 ms for the heaviest scenario and at most 286 ms for all others (Table IV). This load is well above the program's actual population of 23 students, 5 advisors, 2 administrators, and 2 executives. The single bottleneck is the administrator's research workspace, which calls nine endpoints per page load. Its p95 latency rises sharply from 40 ms to 444 ms between 10 and 50 users and then roughly doubles with each doubling of load, which suggests queueing at the database connection layer. For loads above 100 concurrent users, this study recommends two remedies: merge the nine workspace requests into one aggregated endpoint, and size the database connection pool to the designed peak load. It also recommends repeating the measurement on the production server once numeric performance requirements are defined.

TABLE IV.
P95 RESPONSE TIME (MS) BY CONCURRENT USERS

Scenario (requests)	10	50	100	200
Public catalogue search (1)	24	60	164	306
Admin research workspace (9)	40	444	1,056	2,412
Student research workspace (6)	20	286	702	1,466
Advisor research list (2)	6	27	57	111
Executive overview (2)	17	69	150	267

a. Failed requests were 0 in every cell. Results for 1 and 25 concurrent users are omitted for space. Measured on Sep. 6, 2026, in a local environment without network latency.

B. User Acceptance and Satisfaction

User acceptance testing involved 32 participants: 23 students, 5 advisors, 2 administrators, and 2 executives, which covers all current students, advisors, and executives of the program. Each participant also acted as a guest by searching the public catalogue without logging in. They used the prototype in a near-production environment with anonymized data. All 14 role-based tasks produced the expected results (100% pass). Participants then rated the system on a 10-item, five-point Likert scale. The overall mean was 4.51 (SD = 0.68), which falls in the “highest” band (4.50–5.00). Item SDs ranged from 0.53 to 0.77, which indicates consistent opinions. The highest-rated item was fit to the program’s context (4.69), followed by project registration and progress tracking (both 4.56). The lowest was research search and display (4.34), which is still rated “high” (Fig. 6). Ratings by role were consistent: students 4.52, advisors 4.52, executives 4.45, and administrators 4.40. Because the overall mean lies only slightly above the threshold for the “highest” band, it should not be interpreted as substantially different from the adjacent “high” band. Furthermore, because all participants were from the program in which RIMS was developed, these results reflect user acceptance within the AI&CS program and should not be generalized to other programs or institutions without further evaluation.

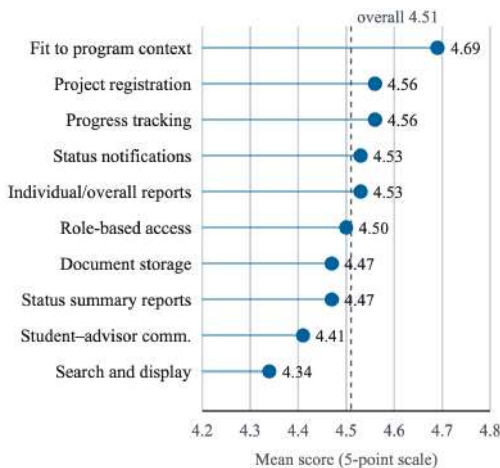


Fig. 6. Mean satisfaction by questionnaire item (n = 32).

C. Task Completion Time

The 14 UAT tasks were divided by role: 6 for students, 4 for advisors, and 4 for administrators, 2 of which were also performed by executives (Table V). Each participant performed each task three times. Mean completion time decreased from the first to the third trial for every task, suggesting improved user efficiency with repeated use and providing evidence of the

system’s learnability. This pattern is consistent with the notion of extended learnability described by Grossman et al. [20], in which learnability is reflected in performance change, such as task completion time, over repeated use. Creating a new research project took the longest (74.7 s on average; 90.9 s in the first trial and 60.7 s in the third), as the task required completing several sections. Common tasks, such as logging in (9.4 s) and searching (9.8–12.2 s), had mean completion times below 15 s.

TABLE V.
UAT TASKS AND MEAN COMPLETION TIME

ID	Task	Role	Time (s)
1001	Log in	S	9.4
1002	Create a research project	S	74.7
1003	Upload a proposal file	S	35.3
1004	Track status and progress	S	14.9
1005	View advisor feedback	S	12.0
1006	Receive a status notification	S	17.6
1007	Give feedback on a project	A	47.5
1008	Reply in a feedback thread	A	27.3
1009	Upload a publication file	A	36.3
1010	Monitor supervised students	A	21.9
1011	Manage master data	M	52.7
1012	Activate a new user account	M	19.1
1013	Search research records	M / E	9.8 / 12.2
1014	Produce an overall status report	M / E	24.0 / 25.0

a. IDs are UAT1001–UAT1014. S = student (n = 23), A = advisor (n = 5), M = administrator (n = 2), E = executive (n = 2). Time is the mean of three trials per participant; executives were timed on tasks 1013 and 1014 only. All 14 tasks passed.

D. Comparison With the Previous Workflow

In semi-structured interviews, each group compared RIMS with its previous practice. Students had submitted research documents by e-mail and shared files, and found RIMS more convenient because everything is in one place and status can be tracked. Advisors had given feedback on paper and by e-mail, and noted that RIMS keeps a systematic history of communication and reduces the need for face-to-face meetings. Administrators had kept offline spreadsheets, and valued online access and retrospective search. Executives had found it difficult to see a program-wide picture, and noted that RIMS summarizes it immediately. The previous workflow was not timed, so this comparison is qualitative only. Measuring the same tasks under both workflows is left for the post-deployment study. The role-specific manuals were found sufficient. Suggestions included more notification channels (students), more detailed research statistics (advisors), multi-level administrator permissions (administrators), and an analytical dashboard for planning (executives).

E. Lessons Learned

Three lessons from testing support the second contribution. First, a 100% pass rate does not prove completeness: requirements without test cases never appear in test results, so coverage must be reviewed per requirement whenever the SRS changes. Second, test cases must be checked against use-case scope before execution. One reported defect expected a profile-editing feature that the SRS explicitly excluded. Third, automated tests that write to a real database need complete, rule-aware teardown. Leftover test accounts caused cascading failures and increased run time from 23 minutes to 1.7 hours.

VI. CONCLUSION

This work presents RIMS, a web-based research information management system that brings the full supervision loop for student research in the AI&CS program at CRRU into one auditable workflow, from proposal and step-wise submission to feedback, approval, and publication, replacing the spreadsheets, e-mail, and paper-based feedback previously used (Objective 1). For the second objective, applying ISO/IEC 29110 to this one-developer project produced end-to-end traceability from use cases to recorded test results. The requirement-level coverage review showed that a test suite with a 100% pass rate still left 27% of requirements untested; closing that gap revealed three defects. The resulting automated regression suite of 927 tests rechecks the system in about 23 minutes.

The current version has five known limitations. Users cannot self-register, edit their profiles, or change their passwords. Data from suspended accounts is purged only by a manual script. The requirements define no numeric performance criteria, so load results serve as a baseline only. Scheduled backup and recovery are not yet implemented. Official graduate-school forms, which use fixed layouts, cannot yet be generated from system data. Future work will integrate with the university's student information API, add chat-application notifications through webhooks, and recommend research topics using AI over the accumulated research corpus. Future work will also generate graduate-school forms and student portfolio pages, define and re-measure performance criteria on the production server, and repeat the satisfaction survey after sustained real-world use.

REFERENCES

- [1] K. C. Laudon and J. P. Laudon, *Management Information Systems: Managing the Digital Firm*, 16th ed. New York, NY, USA: Pearson, 2020.
- [2] L. Castillo-Salinas, S. Sanchez-Gordon, J. Villarroel-Ramos, and M. Sánchez-Gordón, "Evaluation of the implementation of a subset of ISO/IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador. An empirical software engineering experiment," *Computer Standards & Interfaces*, vol. 70, Art. no. 103430, 2020, doi: 10.1016/j.csi.2020.103430.
- [3] M. Muñoz, J. Mejía, A. Peña, C. Y. Laporte, and G. P. Gasca-Hurtado, "Beyond factors that motivate the adoption of the ISO/IEC 29110 in Mexico: An exploratory study of the implementation pace of this standard and the benefits observed," *IET Software*, vol. 15, no. 6, pp. 412–427, 2021, doi: 10.1049/sfw2.12041.
- [4] I G. N. A. C. Putra, C. Pramarttha, M. A. RaHarja, I W. Supriana, I K. G. Suhartana, and N. K. E. Dianasari, "Web-Based Thesis Guidance Monitoring Information System Based on Quick Response Code Technology," in *Proceedings of the First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*, 2024, pp. 255–267, doi: 10.2991/978-94-6463-413-6_26.
- [5] E. C.-M. Wong, F.-F. Chua, and A. H.-L. Lim, "PostConnect: Tracking and managing postgraduate students' research study progress," *AIP Conference Proceedings*, vol. 3153, no. 1, Art. no. 030004, 2024, doi: 10.1063/5.0216501.
- [6] M. Azeem, M. Tukhtasunov, S. Noblitt, M. Jonker, and K. Protzman, "Elevating academic administration: A comprehensive faculty dashboard for tracking student evaluations and research," 2024, *arXiv:2407.07057*, doi: 10.48550/arXiv.2407.07057.
- [7] Y. Yan, X. Han, J. Yang, and Q. Zhou, "On the design of an advanced web-based system for supporting thesis research process and knowledge sharing," *J. Educ. Technol. Develop. Exchange*, vol. 5, no. 2, pp. 111–124, 2012, doi: 10.18785/jetde.0502.08.
- [8] Elsevier. "Pure: Research information management system." Elsevier. <https://www.elsevier.com/products/pure> (accessed Sep. 25, 2026).
- [9] Clarivate. "Converis." Clarivate. <https://clarivate.com/products/scientific-and-academic-research/research-funding-and-analytics/converis> (accessed Sep. 25, 2026).
- [10] 4Science. "DSpace-CRIS." 4Science. <https://4science.com/dspace-cris/> (accessed Sep. 25, 2026).
- [11] Sakon Nakhon Rajabhat University. "Thesis advisor and program lecturer database system (TAS)" (in Thai). SNRU. <https://gsmis.snru.ac.th/advisor/> (accessed Sep. 25, 2026).
- [12] Mahidol University. "Scholarship and Awards Online." Mahidol University. <https://graduate.mahidol.ac.th/scholarship-online/MainPage.php?lang=en> (accessed Sep. 25, 2026).
- [13] Walailak University. "Graduate progress tracking system, Graduate School, Walailak University" (in Thai). User manual, 2021. <https://grad.wu.ac.th/wp-content/uploads/2021/08/Manual-Student-Full-TH.pdf> (accessed Sep. 25, 2026).
- [14] Phetchabun Rajabhat University. "Information system for graduate advisors" (in Thai). PCRU. <https://graduate.pcru.ac.th/advisor> (accessed Sep. 25, 2026).
- [15] Valaya Alongkorn Rajabhat University. "e-Graduate VRU." VRU. https://grad.vru.ac.th/grad_academic-/production/index.php (accessed Sep. 25, 2026).
- [16] A. Methawachananont, M. Buranarach, P. Amsuriya, S. Chaimongkhon, K. Krairaksa, and T. Supnithi, "Software process capability self-assessment support system based on task and work product characteristics: A case study of ISO/IEC 29110 standard," *IEICE Trans. Inf. Syst.*, vol. E103-D, no. 2, pp. 339–347, Feb. 2020, doi: 10.1587/transinf.2018EDP7303.
- [17] C. Y. Laporte, R. V. O'Connor, and L. H. García Paucar, "The implementation of ISO/IEC 29110 software engineering standards and guides in very small entities," in *Evaluation of Novel Approaches to Software Engineering* (Communications in Computer and Information Science, vol. 599), L. A. Maciaszek and J. Filipe, Eds. Cham, Switzerland: Springer, 2016, pp. 162–179, doi: 10.1007/978-3-319-30243-0_9.
- [18] OWASP Foundation. "OWASP Top 10:2021." OWASP. <https://owasp.org/Top10/> (accessed Sep. 25, 2026).
- [19] G. Alzboon and A. Al-Said Ahmad, "A performance evaluation approach for n-tier cloud-based software services," in *Proc. 2022 6th Int. Conf. Cloud and Big Data Computing (ICCBDC)*, Birmingham, United Kingdom, 2022, pp. 31–36, doi: 10.1145/3555962.3555968.
- [20] T. Grossman, G. Fitzmaurice, and R. Attar, "A survey of software learnability: Metrics, methodologies and guidelines," in *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI '09)*, Boston, MA, USA, 2009, pp. 649–658, doi: 10.1145/1518701.1518803.