Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Development of a Data-Driven Class and Thesis/Capstone Defense Scheduling Decision Support System for Higher Education

Ailen Laguda

ailen.laguda@sorsu.edu.ph

Sorsogon State University

Richard Rabulan

Sorsogon State University

Research Article

Keywords: academic scheduling, university course timetabling, defense scheduling, decision support system, constraint-based scheduling, rule-based heuristic, Agile Scrum, ISO/IEC 25010

Posted Date: September 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9517582/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Academic timetabling requires the coordinated allocation of faculty, rooms, courses, sections, and time slots under competing constraints. This study presents a web-based decision support system (DSS) integrating regular class scheduling with thesis and capstone defense scheduling in a shared institutional data environment. The contribution is an integrated scheduling architecture and a constraint-aware, rule-based workflow rather than a new optimization algorithm. Development followed a developmental research design using Agile Scrum across three sprints, producing a decoupled application with a React.js client, a Node.js/Express server, and a relational Turso/libSQL data layer adopted after an early pivot from a document-oriented store. Scheduling draws on faculty workload and availability, room capacity and type, subject requirements, section and program data, and defense-panel availability to detect conflicts across both domains. The system was implemented at two institutions on separate instances: a state university ICT college, deployed December 2025, and a private college, deployed January 2026. Evaluation used a structured ISO/IEC 25010 questionnaire covering Functional Suitability, Performance Efficiency, Usability, and Reliability, administered to thirteen respondents comprising three IT expert evaluators, four scheduling administrators, and six faculty members. Attribute means were 4.775, 4.810, 4.788, and 4.550, with an overall mean of 4.731, all in the highest interpretation band. Archived operational outputs include 1,245 class-schedule meeting blocks, 142 defenses, 87.0% room utilisation across 70 rooms, and 318 conflict candidates resolved before finalisation. Results are reported as respondent evaluation and are distinguished throughout from objective engineering measurement, with technical claims limited to what surviving records support.

1 Introduction

The allocation of academic resources — time, faculty, and physical classrooms — remains one of the most complex recurring administrative problems in Higher Education Institutions (HEIs). University course timetabling is a resource-allocation problem in which academic events must be assigned to appropriate times and rooms while respecting faculty, student, curriculum, and institutional constraints. The problem is combinatorial and grows rapidly more difficult as the number of events, resources, and competing requirements increases [1–4]. Reviews of the field consistently distinguish hard constraints, whose violation renders a timetable infeasible, from soft constraints or preferences that should be minimised or balanced [1, 2].

Despite advances in educational technology, many institutions continue to rely on manual, pen-and-paper scheduling [5]. This approach is repetitive, resource-intensive, and prone to human error, and it commonly produces inefficiencies, scheduling conflicts, and broader resource mismanagement. Systematic reviews of digital transformation in higher education identify organisational, resource, and competence barriers that slow the replacement of such manual practices even where suitable technology is available [13, 14]. Real institutional scheduling also demands decision support that extends past timetable generation: administrators need visibility into room utilisation, faculty workload, conflicts, resource bottlenecks, and the downstream consequences of manual changes. Recent

decision-support studies therefore combine scheduling models with operational planning, instructor assignment, and interactive administrative workflows [6, 7].

A related but usually separately managed problem is thesis and capstone defense scheduling. Defense schedules must coordinate student groups, panel members, venues, and time slots while respecting panel eligibility, faculty availability, teaching commitments, and workload ceilings. Examination-timetabling research shows that event scheduling carries its own constraint structure and solution methods [8]. In practice, however, class schedules and defense schedules compete for the same faculty and the same rooms. Existing literature and commercial solutions frequently treat class timetabling and event or examination scheduling as separate, non-integrated domains, obliging administrators to operate disparate systems or to fill gaps manually [4, 8]. That isolation produces ad hoc coordination in which faculty workloads and venue availability must be cross-referenced across two domains to prevent overlaps.

The present study addresses this operational integration problem. Rather than claiming a new optimization algorithm, it contributes a single institutional DSS in which class timetabling and defense scheduling share faculty, room, workload, and time data, and in which conflicts across both domains are detected in one environment. The system was implemented at two institutions with different program portfolios and different scheduling cultures, which provides an initial indication of transferability beyond the college that developed it.

1.1 Research Gap and Contribution

Existing studies provide mature formulations and algorithms for university course and examination timetabling [1–4, 8], while DSS research demonstrates the value of integrating optimization with administrative planning [6, 7]. The practical gap addressed here is the coordination of ordinary class schedules and thesis or capstone defenses inside a single operational data environment. The study contributes: (1) a shared institutional scheduling architecture; (2) an explicit constraint-aware rule-based scheduling workflow; (3) integrated class and defense conflict detection with decision-support visualisation; and (4) an applied evaluation combining a documented ISO/IEC 25010-based questionnaire with development-phase technical testing, conducted across two institutional sites.

1.2 Objectives and Research Questions

The primary objective of this study is to streamline academic resource management. Specifically, it aims to:

1. Design and develop a robust, intuitive platform that automates class scheduling, minimises conflicts, and integrates critical administrative data.
2. Integrate a Defense Scheduling Module that supports the assignment of panel members, time slots, and venues while complying with institutional constraints.

3. Evaluate the developed system using ISO/IEC 25010 software quality criteria covering functional suitability, performance efficiency, usability, and reliability.

These objectives are expressed as four research questions:

RQ1. How can class scheduling and thesis or capstone defense scheduling be integrated within one decision-support architecture using shared faculty, room, workload, and time data?

RQ2. What hard constraints, preferences, and rule-based decision steps are used to construct feasible assignments and to identify unresolved scheduling conditions?

RQ3. How did respondents evaluate the system in terms of Functional Suitability, Performance Efficiency, Usability, and Reliability using the documented ISO/IEC 25010-based instrument, and what technical testing accompanied that evaluation?

RQ4. What administrative decision-support functions are provided for room utilisation, faculty workload, schedule modification, reporting, and auditability?

2 Related Literature

2.1 University Timetabling and Constraints

University timetabling is widely treated as an NP-hard combinatorial optimization problem. Recent reviews synthesise metaheuristics, mathematical programming, hybrid methods, benchmarks, and constraint formulations [1, 2]. Real-world benchmark work further emphasises that institutional timetabling involves diverse local rules and should be evaluated against clearly documented instances and constraints [3].

2.2 Mathematical and Heuristic Approaches

Recent studies formulate course timetabling using multi-objective mathematical programming, hybrid metaheuristics, and exact or heuristic methods [4, 9, 10]. These approaches are valuable when algorithmic optimality or comparative solution quality is the primary contribution. Informed search methods such as A* have also been applied to course scheduling decisions in smaller institutional settings [16]. The present study differs in that its documented implementation is better characterised as a rule-based institutional DSS than as a new mathematical optimizer.

2.3 Decision-Support Systems for Academic Planning

DSS research in higher education has combined scheduling with instructor assignment and demand-driven planning [6]. Interactive decision support has also been applied to student-facing academic planning such as degree-plan construction [15], and web-based timetabling systems demonstrate the importance of operational interfaces, database integration, and administrator interaction [7]. These

studies support the view that a practical scheduling contribution can lie in integrated information and decision workflows, not only in a novel solver.

2.4 Examination and Defense Scheduling

Examination timetabling research addresses conflict avoidance, capacity, temporal spacing, and fairness through exact, heuristic, and hybrid methods [8]. Thesis and capstone defense scheduling is structurally related but introduces panel composition and faculty workload constraints. In the present system, defense assignments are coordinated with regular teaching schedules through shared faculty and venue data.

2.5 Positioning of the Present Study

The study claims novelty at the level of institutional integration and decision support. It does not claim to outperform established optimization algorithms. Its value lies in combining class and defense scheduling, exposing resource conflicts, supporting human review, and documenting an applied software-quality evaluation across two implementation sites.

Table 1
Positioning of the present study relative to prior work.

Study	Scheduling domain	Approach	Evaluation emphasis	Relation to present study
Abdipoor et al. [1]	University course timetabling	Review of metaheuristics	Algorithms and benchmarks	Provides optimization-method context
Bashab et al. [2]	University timetabling	Review of constraints and methods	Benchmarks and open issues	Supports hard/soft constraint framing
Müller et al. [3]	Real-world course timetabling	Competition benchmark	Institutional instances	Shows importance of reproducible real-world constraints
Davison et al. [4]	Course timetabling with hybrid teaching	Multi-objective mathematical model	Solution quality and trade-offs	Formal optimization approach not claimed here
Xue et al. [6]	Course planning and instructor assignment	DSS with optimization	Operational planning	Comparable DSS orientation
Uysal et al. [7]	Online course timetabling	Web-based DSS	Operational web implementation	Comparable software and DSS focus
Siew et al. [8]	Exam timetabling	Survey	Methods and constraints	Related event-scheduling domain
Present study	**Class and defense scheduling**	**Constraint-aware rule-based DSS**	**Operational integration and user evaluation**	**Integrates two scheduling domains through shared resources**

3 Methodology and System Architecture

3.1 Research Design and Institutional Context

The research employed a Developmental Research Design guided by the Agile Scrum methodology, allowing continuous integration and user feedback across three sprints. Requirements were gathered through interviews, observation, stakeholder consultation, and document analysis.

The study was conducted across two institutional sites running independent instances with separate databases. The first site is the College of Information and Communications Technology (CICT) of Sorsogon State University, which offers the BSIS, BSCS, BSIT, and BTVTEd programs and where both proponents are core faculty of the BSIS program. CICT was the development and first-utilisation site: the system was tested and debugged there across the three sprints, reached full deployment on 12 December 2025 in preparation for the following term, and entered live scheduling use for the second

semester of SY 2025–2026, which began on 19 January 2026. The system has remained in continuous use since, through the first semester of SY 2026–2027 which began on 10 August 2026. The technical evaluation, including assessment by IT expert evaluators, was conducted at this site. The second site is Sorsogon College of Criminology Inc. (SCCI), a private higher education institution with a distinct program portfolio, governed by a Memorandum of Agreement executed on 5 December 2025 and deployed on 19 January 2026. The usability evaluation by scheduling administrators and faculty was conducted at this site.

Implementing at two institutions with different program structures served a methodological purpose beyond wider adoption: it tested whether the constraint model and data schema generalised past the college that designed them. All quantitative operational outputs reported in Section 4.3 derive from the CICT instance, which had been in live use longest at the time of reporting.

Table 2
Implementation and evaluation timeline across the two sites.

Milestone	Site	Date
Sprints 1–3: development, testing, debugging	CICT	1st semester SY 2025–2026
Memorandum of Agreement executed	SCCI	5 December 2025
Full (100%) deployment	CICT	12 December 2025
Full deployment	SCCI	19 January 2026
Start of live utilisation (2nd sem SY 2025–2026)	Both	19 January 2026
Logged incidents ERR-890, ERR-891, ERR-902	CICT	15 November, 16 November, 4 December 2025
Panel overload warning generated	CICT	April 2026
Continuing use (1st sem SY 2026–2027)	Both	from 10 August 2026

Because full deployment on 12 December 2025 preceded the January term, the three development sprints ran during the first semester of SY 2025–2026.

To ensure the system could handle CPU-bound scheduling computation while remaining responsive, a decoupled web architecture was adopted (Fig. 1).

3.2 Data Sources and System Requirements

The system uses the institutional data required for scheduling: faculty profiles and workload limits, rooms with capacities and classifications, subject and course information, student sections and programs, time blocks, and defense groups with panel assignments. Requirements reflected the need to reduce schedule conflicts, distribute faculty load, allocate appropriate rooms, coordinate class and

defense activities, and preserve administrator control through editable schedules with validation. Because the two sites run separate instances and separate databases, no institutional data is shared between them; the common element is the schema and constraint model, not the records.

3.3 Frontend User Interface

The client-side interface was developed using React.js combined with the Tailwind CSS framework. Because the master schedule and weekly timetables render large volumes of data simultaneously across interactive grids, React was chosen to reduce the cost of DOM updates. The interface uses targeted React hooks to cache data and isolate component re-renders, so that when an administrator manually adjusts a time block the surrounding grid does not re-render wholesale. Tailwind CSS was used to maintain a responsive, lightweight stylesheet that adapts across the administrative devices in use at both sites.

3.4 Backend Processing and Scheduling Engine

The server-side infrastructure was built on Node.js using the Express.js framework. Scheduling computation evaluates multiple constraints concurrently — room capacities against faculty availability against subject requirements — and is therefore CPU-bound rather than I/O-bound. The event-driven Node.js runtime serves concurrent HTTP requests from administrative users, and the decoupled client allows the interface to remain interactive while a generation request is outstanding. The system is not claimed to offload scheduling computation to worker threads, child processes, or a separate job queue; the architectural benefit reported here is separation of concerns and maintainability rather than parallel execution of the scheduling workload.

3.5 Database and Data Integrity Management

A critical architectural decision was the selection of the database framework. A document-oriented store (MongoDB) was used in the initial design but introduced significant complexity around data duplication and the absence of the strict normalisation that academic scheduling constraints require. The system therefore pivoted to Turso, an edge-hosted database built on libSQL. This relational architecture enforces referential integrity through foreign key constraints, preventing errors such as assigning a subject to a deleted faculty profile or scheduling an event in a venue that no longer exists. The pivot was executed following Sprint 1 and is reflected in the Sprint 2 recovery reported in Section 4.1. The same schema was subsequently instantiated independently at each site.

3.6 Scheduling Problem Formulation

Let C denote class events, D defense events, F faculty members, R rooms, and T available time slots. For a class $c \in C$, a candidate assignment consists of an eligible faculty member $f \in F$, a compatible room $r \in R$, and a time slot $t \in T$. For a defense $d \in D$, a candidate consists of an eligible panel set $P \subseteq F$, a room r, and a time slot t. The system seeks feasible assignments that satisfy mandatory institutional rules and, among feasible alternatives, prefers assignments that better distribute workload and reduce congestion.

3.6.1 Hard Constraints

- No faculty member may be assigned to overlapping class or defense events.
- No room may host overlapping events.
- Room capacity and classification must be compatible with the scheduled activity.
- Faculty availability and recorded workload limits must be respected.
- Class assignments must comply with subject, section and program, and institutional time requirements.
- Defense assignments must use eligible panel members who are available at the selected time.
- Unavailable or deleted institutional resources cannot be assigned.

3.6.2 Preferences and Administrative Priorities

- Reduce avoidable faculty and section timetable gaps.
- Distribute teaching and panel duties where feasible.
- Prefer suitable rooms without unnecessarily consuming scarce specialised facilities.
- Reduce concentration in highly congested time periods.
- Prioritise institutional rules such as major-course requirements and recorded room specialisation.
- Flag unresolved cases for administrator review rather than forcing an infeasible assignment.

3.7 Rule-Based Constructive Scheduling Procedure

The surviving research documents describe simultaneous constraint checking, conflict detection, workload consideration, room allocation, and manual adjustment with validation, but do not preserve a source-code-level mathematical solver specification. The implemented behaviour is therefore reported as a rule-based constructive heuristic:

1. Load current faculty, room, section, subject, time-slot, and defense requirement data.
2. Select an unscheduled class or defense event according to administrative priority.
3. Generate feasible candidate resources and time slots.
4. Reject candidates violating hard constraints.
5. Compare remaining candidates using workload, room suitability, time congestion, and other documented preferences.
6. Assign the preferred feasible candidate provisionally.
7. Update resource availability and continue until all events are assigned or no feasible candidate remains.
8. Flag unresolved events for administrator review.
9. After manual edits, revalidate the integrated class-and-defense schedule for conflicts.

Conceptually the decision objective is lexicographic: hard-constraint violations take precedence over soft-preference penalties. This formulation improves methodological transparency without claiming that the production system solves a globally optimal mathematical objective.

3.8 Defense Scheduling Integration

The defense module shares faculty and room information with the class scheduler. The completed report documents panel workload distribution, venue availability, student groupings, panel assignment based on availability, conflict detection, and manual schedule adjustment with real-time validation. This shared-resource design is the principal integration contribution, because teaching and defense commitments are checked in one environment rather than cross-referenced across two systems.

3.9 Evaluation Design, Respondents, and Instrument

Evaluation was conducted in two complementary strands matching the two implementation sites, and the results were pooled into a single ISO/IEC 25010 dataset.

The technical evaluation was conducted at CICT by three IT expert evaluators selected for domain expertise: a core faculty member of the BSIT program holding a Certificate of Academic Requirements in Doctor in Information Technology, a core faculty member of the BSCS program holding a PhD in Computer Science, and a second core faculty member of the BSIT program holding a Certificate of Academic Requirements in Doctor in Information Technology. The experts examined the system, contributed recommendations for revision, and completed the evaluation questionnaire.

The usability evaluation was conducted at SCCI with ten respondents comprising four administrators and six faculty members, drawn from the roles directly involved in scheduling operations: program chair, IT administrator, scheduling officer, defense coordinator, and faculty scheduling users. The pooled evaluation sample is therefore thirteen respondents across the two sites.

A structured questionnaire based on selected ISO/IEC 25010 software quality characteristics was administered [11], covering Functional Suitability, Performance Efficiency, Usability, and Reliability. Items were rated on a five-point Likert scale from 1 (Strongly Disagree) to 5 (Strongly Agree). The questionnaire was validated by IT experts and pilot-tested before full deployment. Weighted means were computed for each item and for each quality characteristic.

The interpretation rubric applies equal-interval bands across the five-point scale (Table 3). The original completed report used a rubric whose highest band, 4.1–5.0, carried the descriptive label “Exceptionally Excellent”, and whose lowest band extended to 0.0 despite the scale having a floor of 1.00. The rubric below corrects that floor and adopts equal 0.80-wide intervals, which is the conventional treatment for a five-point Likert scale. The substitution does not alter any conclusion: all four attribute means and the overall mean fall in the highest band under either rubric.

Table 3
Interpretation rubric for weighted means on the five-point scale.

Weighted mean	Interpretation	Response label
4.21–5.00	Very High	Strongly Agree
3.41–4.20	High	Agree
2.61–3.40	Moderate	Neutral
1.81–2.60	Low	Disagree
1.00–1.80	Very Low	Strongly Disagree

3.10 Technical Testing Protocols and Evidence-Preservation Boundary

Three technical test programmes accompanied the questionnaire during development at CICT. Performance efficiency was load-tested using Apache JMeter; the result files and test-environment parameters were not retained, and no load-test figures are reported. Reliability was tracked through an incident log summarised using Mean Time To Repair. Security posture was assessed with the OWASP Zed Attack Proxy (ZAP) during Sprint 1, with remediation across Sprints 2 and 3. Where a test was conducted but its output is unavailable, this manuscript records that the test occurred and states that its results are not reportable; it does not reconstruct values from memory or from the surviving narrative summary.

A preservation boundary must be stated plainly. Some underlying raw artifacts — JMeter result files and environment parameters, complete uptime and failure-history records, and the original and post-remediation ZAP reports — are no longer available after the system transitioned to live operation. Re-running intrusive load or security tests against a production system would disrupt users and touch live institutional data, so those tests were not repeated for this manuscript. The results below therefore separate three classes of evidence: perception-based questionnaire results, archived operational outputs produced by the deployed system, and development-phase technical measurements. Where a technical claim depends on an artifact that was not retained, the manuscript reports the surviving summary and marks it as unreproducible rather than restating it as a verified measurement. Appendix A records the status of every evidence item.

4 Results and Discussion

4.1 Sprint Progression and Usability Ratings

Sprint review sessions were anchored in Nielsen’s usability heuristics [12], with particular attention to error prevention, user control, and system visibility. Usability was tracked across three development sprints at CICT with five core users occupying the roles most involved in scheduling operations, using a

0–100 reporting scale (Table 4). These are formative usability ratings collected during sprint reviews to steer iteration. They are deliberately not described as percentiles or as System Usability Scale scores, since the surviving records do not preserve the scoring conversion that either label would require. This strand is separate from the ISO/IEC 25010 evaluation in Section 4.6, which was administered after remediation to a different and larger sample.

Sprint 1 produced a mean of 50.0, prompting major UI/UX redesign and the architectural shift from the document store to libSQL described in Section 3.5. Sprint 2 reached a mean of 85.0, with the residual shortfall attributed in the preserved development narrative to two defects: a Google Authentication backend failure during login and errors in module printing. Following resolution of both defects, Sprint 3 reached a mean of 96.0, with improvement recorded for every role. The trend is reported as formative evidence of iterative improvement during development among five core users, not as a population-level usability estimate.

Table 4
Formative usability ratings across three sprints at CICT (n = 5 core users; 0–100 reporting scale).

User role	Sprint 1	Sprint 2	Sprint 3
Program Chair	52	86	98
IT Admin	48	82	95
Scheduling Officer	55	88	96
Defense Coordinator	45	84	94
Faculty Member	50	85	97
Mean	**50.0**	**85.0**	**96.0**

4.2 System Interfaces and Decision-Support Visualisations

The completed system provides administrative interfaces for rooms, faculty, subjects, sections, class schedules, weekly timetable views, defense schedules, reports, and system logs. Figure 2 shows the master scheduling dashboard, the central hub of the DSS. The interface uses a grid-based visualisation of the master timetable with colour-coded status cues, which alert administrators to unassigned teaching slots, potential faculty overloads, or venue capacity limits before a schedule is finalised. Rows marked “Unassigned / Missing Teacher” show the system deliberately withholding an infeasible assignment and escalating it for administrator review, consistent with step 8 of the procedure in Section 3.7.

The weekly schedule view displays classes by day, time, room, and assigned faculty and supports review of potential conflicts and real-time adjustment. The report centre supports schedule and utilisation reporting, and the system-log module records user and system activity for traceability and audit review.

4.3 Archived Operational Outputs

The CICT deployment produced a set of aggregate operational outputs. These are reported as archived system outputs rather than as controlled experimental results: the underlying conflict logs and extraction queries were not retained, so the values cannot be independently recomputed, and they are not presented as evidence of algorithmic optimality or of superiority over a manual baseline.

Table 5
Archived operational outputs from the CICT instance.

Archived system output	Reported value
Class-schedule meeting blocks in the snapshot	1,245
Thesis/capstone defenses represented	142
Overall room-utilisation indicator	87.0% across 70 rooms
of which scheduled and confirmed	92 (65%)
of which awaiting panel availability	28 (20%)
of which successfully completed	15 (10%)
of which rescheduled or in conflict	7 (5%)
Conflict candidates detected and resolved before finalisation	318
Average faculty workload indicator	18 h/week, including teaching and panel duties

Two notes on definitions. The count of 1,245 refers to class-schedule meeting blocks — individual scheduled class periods — not to distinct course sections, of which there are correspondingly fewer. The count of 318 refers to conflict candidates detected and resolved before finalisation, arising across both automated generation and administrator review; it is not claimed that this figure represents every conflict occurring in the institution, nor that the engine resolved all of them without human intervention.

The defense pipeline is reported here in the same four states shown in Fig. 4. An earlier draft listed a value of five under the heading “unresolved panel conflicts”; that figure was a transcription of the 5% share of the rescheduled-or-in-conflict slice rather than a count, and the corresponding count is seven defenses. No separate measure of five exists, and the summary list has been corrected accordingly.

4.3.1 Facility and Capacity Visualisations

To support administrative decision-making the system generated visual analytics tracking facility bottlenecks and faculty workload. Figure 3 presents utilisation by time and day.

Figure 4 Capstone and thesis defense status across 142 defenses: Scheduled and Confirmed 92, Pending Panel Availability 28, Successfully Completed 15, Rescheduled or in Conflict 7. The counts sum to 142; the printed shares (65/20/10/5) are rounded from 64.8/19.7/10.6/4.9

Table 6
Room type utilisation and primary program usage.

Room classification	Total rooms	Avg weekly utilisation	Primary usage
Lecture rooms	45	88%	BSIS
Laboratory rooms	12	96%	BSIT / Capstone
Audio-visual rooms	5	92%	College of BME
Specialised laboratories	8	65%	BTVTEd
Total / weighted mean	**70**	**87.0%**	**—**

The overall room-utilisation indicator of 87.0% is the mean of the four room-class averages weighted by room count across the 70 rooms in the shared inventory. The inventory extends beyond CICT-controlled space: the five audio-visual rooms are administered by the College of Business Management and Entrepreneurship and are drawn on for defense sessions, which is why they appear in the scheduling pool and in the facility-shortage analysis below. Program attribution in the Primary Usage column reflects the CICT program portfolio of BSIS, BSCS, BSIT, and BTVTEd, together with the externally administered audio-visual rooms.

The BSCS program shares the lecture and laboratory room classes listed above and is not allocated a separate classification in the room inventory.

4.3.2 Automated System Insights and Warnings

Based on the scheduling rules, the system generated administrative insights of three kinds:

1. **Panel overload warning.** In April 2026 one faculty member was flagged for 12 thesis panels within a single week while carrying a 15-hour teaching load. The system recommended redistributing 3 assignments to alternate faculty. This case sits above the 18 h/week institutional average reported in Table 5, which is itself elevated: teaching overloads are widespread at CICT because of faculty shortage, so the average reflects a staffing constraint rather than a target load. The warning is consequently a useful instance of the integration contribution, since a class-only scheduler would not have detected that panel duties were compounding an already heavy teaching assignment.
2. **Facility shortage.** Audio-visual rooms reached 92% average weekly utilisation (Table 6). The system recommended temporarily converting two general-purpose rooms into defense venues using portable projectors. Laboratory rooms show marginally higher utilisation (96%), but the audio-visual rooms were identified as the operative shortage because they are the venue class required for defense sessions and are administered by another college, which limits the scheduler's ability to expand supply.

Conflict

3. **resolution.** 318 conflict candidates were detected and resolved before semester finalisation, across both automated generation and administrator review.

4.4 Reliability and Incident Tracking

System reliability during the development and early deployment phases was tracked through an incident log and summarised using Mean Time To Repair (MTTR):

MTTR = Total time to repair / Number of failures (1)

Mean Time Between Failures is not reported. Computing it requires total operational uptime across a defined observation window and a complete failure count, neither of which was retained; the surviving evidence records when each fault was detected and repaired, not how long the system ran between faults. The metric has therefore been withdrawn rather than estimated.

Table 7 records the critical failures logged on the CICT instance during the Sprint 2 and Sprint 3 test cycles, in the weeks preceding full deployment. Incident ERR-890 (database memory leak) and ERR-891 (authentication gateway timeout) reflect strain on the server architecture during high-concurrency read and write operations. Incident ERR-902 tested the scheduling logic directly: the constructive procedure failed to terminate while attempting to resolve a four-way thesis panel conflict, which prompted the addition of an explicit candidate-exhaustion exit consistent with step 7 of Section 3.7.

Table 7
System incident log and repair durations recorded during Sprint 2 and Sprint 3 testing, CICT instance.

ID	Date	Description / root cause	Time to repair	Severity
ERR-890	15 Nov 2025	System crash: database memory leak during heavy query	1 h 15 m	Critical
ERR-891	16 Nov 2025	Gateway timeout: authentication server overload	30 m	High
ERR-902	4 Dec 2025	Module freeze: scheduling procedure failed to terminate on a four-way panel conflict	45 m	Critical
MTTR	**—**	**Arithmetic mean of the three repair durations**	**50 m**	**—**

One correction applies to the earlier presentation of this table. The per-incident durations were previously labelled “MTTR”; MTTR is by definition a mean, so the column is relabelled “Time to repair” and the mean of 50 minutes across three incidents is reported separately. The log covers faults rated critical and high; lower-severity defects were handled within the ordinary sprint backlog and are not itemised here. Repair times ranging from 30 minutes to 1 hour 15 minutes indicate high maintainability in the decoupled architecture: isolating faults to specific modules, such as the authentication gateway as distinct from the scheduling engine, allowed each to be remediated without a full-system rollback. All three faults surfaced during Sprint 2 testing, were remediated within the same cycle, and were confirmed resolved on

the Sprint 3 retest, with no recurrence recorded through full deployment on 12 December 2025 or in live operation thereafter.

4.5 Security Vulnerability Assessment and Remediation

During Sprint 1 an automated security assessment was conducted using the OWASP Zed Attack Proxy (ZAP) to establish a quantitative baseline and direct remediation in subsequent sprints. Two units of measurement are reported and must be distinguished. ZAP groups findings of the same type into a single alert, and records each affected URL as a separate instance of that alert. The scan produced 24 distinct alerts. Alert counts are used for the severity and threat-category distributions in Figs. 5 and 6. Table 8 itemises the two principal alert types at each severity level — eight alert types in total — and reports the number of URL-level instances at which each was raised, amounting to 35 instances. Table 8 is therefore a partial itemisation of the scan, not a complete listing of all 24 alerts, and its instance total does not represent the scan-wide instance count.

The 24 alerts fall across four severity levels: High (2), Medium (5), Low (11), and Informational (6). Figure 5 shows the severity distribution and Fig. 6 the distribution by OWASP Top 10 threat category.

Table 8
Principal alert types by severity, with URL-level instance counts, Sprint 1 baseline scan (two alert types itemised per severity level).

Risk level	Vulnerability name	Instances
High	SQL injection (CWE-89)	1
High	Cross-site scripting, reflected (CWE-79)	3
Medium	Absence of anti-CSRF tokens (CWE-352)	4
Medium	Content Security Policy: wildcard directive	2
Low	X-Content-Type-Options header missing	15
Low	Cookie without HttpOnly flag	8
Informational	Two informational findings, one instance each	2
Total	**Instances across the eight itemised alert types**	**35**

Table 9 reconciles the two tables against the severity distribution. At every severity level the itemisation lists exactly two alert types, so the eight rows of Table 8 account for eight of the twenty-four alerts: both of the two High alerts, two of five Medium, two of eleven Low, and two of six Informational. Read this way the figures are mutually consistent, and the earlier apparent contradiction — six Informational alerts against two Informational entries — dissolves, because the two entries are itemised alert types rather than the full Informational set. The instance counts of the sixteen alerts not itemised were not retained, so no scan-wide instance total is reported.

Table 9
Reconciliation of alert counts, itemised alert types, and URL-level instances.

Severity	Alerts in scan (Fig. 5)	Alert types itemised (Table 8)	Instances of itemised types
High	2	2	4
Medium	5	2	6
Low	11	2	23
Informational	6	2	2
Total	**24**	**8**	**35**

The two-per-severity itemisation pattern holds at all four levels, and the itemised instance counts sum exactly to the recorded total of 35. The alert types not itemised, and their instance counts, were not retained in the surviving scan summary.

Security misconfigurations were by far the most prevalent category. The majority of findings were Low or Medium risk, with only 8.3% of alerts classed as High. This distribution directed the remediation strategy toward the High and Medium findings first, reducing exposure of student and faculty data most quickly. The most critical items were addressed first: a SQL injection reachable through the room-search parameter was closed by routing all affected queries through strict Object-Relational Mapping, and reflected cross-site scripting in the error-message parameters was closed using context-aware output encoding. Remediation proceeded across Sprints 2 and 3. Because the original and post-remediation ZAP reports were not retained, no post-remediation counts are claimed.

4.6 ISO/IEC 25010 Respondent Evaluation

Following remediation, the finalised system was evaluated using the ISO/IEC 25010 software quality model across four characteristics, pooling responses from the three CICT expert evaluators and the ten SCCI administrators and faculty (n = 13). These results are perception-based evaluation data. They should not be read as direct measurements of response time, throughput, failure rate, or algorithmic optimality; those belong to Sections 4.4 and 4.5 and are bounded by the preservation limits stated in Section 3.10.

Table 10
Summary of the ISO/IEC 25010 evaluation (pooled, n = 13).

Quality characteristic	Overall mean	Interpretation	Basis
Functional Suitability	4.775	Very High	Respondents rated the implemented scheduling functions very favourably
Performance Efficiency	4.810	Very High	Respondents perceived the system as responsive and efficient in use; not an objective load-test result
Usability	4.788	Very High	Respondents rated navigation, learnability, prompts, and interface organisation very favourably
Reliability	4.550	Very High	Respondents rated stability, dependable outputs, recovery, and logging favourably
Overall	**4.731**	**Very High**	**Overall respondent evaluation was strongly favourable across the four assessed characteristics**

Table 11
Status of each evidence item and its use in this manuscript.

Evidence item	Status	Use in this manuscript
ISO/IEC 25010 questionnaire description and 5-point scale	Preserved	Section 3.9; interpretation of respondent evaluation
Item-level and attribute-level means	Preserved	Section 4.6, perception-based evidence
Respondent composition (3 CICT experts, 10 SCCI users)	Preserved	Section 3.9; pooled n = 13
System interface screenshots (mock data)	Preserved	Figures 1–6; functional and architectural evidence
Informed consent form	Preserved	Declarations
Institutional approval materials	Preserved	Declarations; ethics context
Sprint usability ratings (5 core users)	Preserved	Section 4.1, formative evidence
Aggregate operational outputs (CICT snapshot)	Preserved	Section 4.3, archived system outputs
Incident log with repair durations	Preserved	Section 4.4, Table 7
Sprint 1 OWASP ZAP alert and instance summary	Partially preserved	Section 4.5; eight of 24 alerts itemised
JMeter raw results and environment parameters	Not preserved	Load-performance figures withdrawn
Complete uptime and failure-history data	Never recorded	MTBF metric withdrawn from the manuscript
Original and post-remediation ZAP reports	Not preserved	Post-remediation counts excluded
Raw conflict logs and snapshot extraction queries	Not preserved	Operational outputs not independently recomputable
Memorandum of Agreement and SCCI permission letter	Preserved	Declarations; SCCI authorisation
CICT internal deployment authorisation	Internal	Declarations; institutional authorisation

The item and attribute means are reproduced as recorded in the original completed research report. Evaluator-level raw responses were not preserved, so the means cannot be recomputed from surviving data and should be read as the values documented at the time of evaluation rather than as figures recalculated for this manuscript. All four attribute means and the overall mean fall in the highest band of the rubric in Table 3. Under the rubric used in the original completed report, the same values carried the descriptive label “Exceptionally Excellent”; the substantive interpretation is unchanged.

4.6.1 Functional Suitability

Functional Suitability obtained an overall mean of 4.775. The four preserved items were accurate generation of conflict-free class schedules (4.65), correct defense panel and venue assignment (4.92), compliance with institutional rules and constraints (4.78), and completeness of scheduling features for classes and defenses (4.75). Respondents perceived the core scheduling functions as highly suitable for their intended administrative tasks. The lowest-scoring item, conflict-free class schedule generation, is consistent with the behaviour visible in Fig. 2, where infeasible assignments are deliberately escalated rather than forced.

4.6.2 Performance Efficiency

Performance Efficiency obtained an overall mean of 4.810, the highest-rated attribute. Item means were 4.85 for processing class schedules without perceived delay, 4.95 for prompt defense scheduling, 4.54 for handling large datasets without perceived degradation, and 4.90 for prompt response during navigation. Because these are questionnaire responses, they support perceived responsiveness rather than measured throughput, latency, concurrency, or resource utilisation. The lowest item concerns large-dataset handling, which is the dimension a load test would address directly and which the JMeter results would substantiate if the raw files can be recovered.

4.6.3 Usability

Usability obtained an overall mean of 4.788. Respondents rated the interface as intuitive and easy to navigate (4.75), requiring minimal training (4.85), providing clear error messages and prompts (4.63), and having logically organised and visually appealing menus and dashboard (4.92). Because the usability strand was administered at SCCI, an institution that did not participate in development, these ratings carry more weight as evidence of learnability by users unfamiliar with the system's design rationale.

4.6.4 Reliability

Reliability obtained an overall mean of 4.550, the lowest of the four attributes. Item means were 4.24 for consistent operation without crashes, 4.86 for dependable schedules and reports, 4.35 for recovery from errors or interruptions, and 4.75 for accurate logging and tracking of user activity. The two lowest items concern crash-free operation and error recovery, corresponding directly to the incident record in Table 7; respondents at CICT who experienced ERR-890 or ERR-902 would plausibly rate those items lower. This convergence between perception and the incident log is the strongest reliability evidence available, but it is not equivalent to an MTBF, failure-rate, or availability measurement.

4.7 Interpretation and Positioning

The strongest evidence in this study is not a benchmark showing that the scheduler mathematically outperforms other algorithms. It is the documented implementation of an integrated class-and-defense scheduling workflow across two institutions, the operational analytics that workflow produces, and the

favourable evaluation of the users who operated it. This distinction matters because earlier drafts used terms such as “optimization algorithm” and “high performance” more strongly than the retained technical evidence permits.

The rule-based formulation is consistent with the broader timetabling literature in its separation of mandatory from preferential constraints [1, 2]. Formal optimization studies, however, publish objective functions, solver settings, benchmark instances, and comparative solution quality [3, 4, 9, 10]. The present study should therefore be compared primarily as an applied DSS implementation rather than as a competing optimizer.

The integration contribution is practically meaningful. Class and defense scheduling share faculty, rooms, and time, so separate workflows create a standing risk of duplicate allocation and oblige administrators to cross-check manually. The unified data model and cross-domain conflict validation address that coordination problem directly. The April 2026 panel-overload warning is a concrete instance: a faculty member carrying 12 panels in one week on top of a 15-hour teaching load represents a conflict that a class-only scheduler could not surface, and the surrounding context of chronic faculty shortage at CICT is precisely the condition under which such compounding is most likely and least visible. This complements recent DSS work emphasising instructor assignment, demand planning, and operational decision support [6, 7].

The two-site implementation adds a modest transferability signal. CICT and SCCI differ in program portfolio, room inventory, and scheduling culture, and the system was instantiated at the second site on a separate database without schema modification. This is not a controlled multi-site validation, and no comparative outcome data is reported for SCCI, but it does indicate that the constraint model was not overfitted to the college that produced it.

The revision also carries a research-data-management lesson. Developmental software research should archive evaluator-level data, instruments, source-code versions, benchmark datasets, load-test files, security scan reports, and environment configurations before deployment. Once a system becomes operational, re-running aggressive tests against live data may be inappropriate, and evidence that was never archived cannot be recovered.

4.7.1 Implications for Practice

For institutions with overlapping class and defense responsibilities, a shared scheduling environment can reduce manual cross-checking and give administrators a consolidated view of faculty workload, room availability, pending assignments, schedule changes, and audit history. This is most valuable where staffing is constrained and overload is common, since those are the conditions under which workload conflicts accumulate fastest. Adoption beyond the evaluated settings should nonetheless follow local validation, because scheduling policies, room structures, academic calendars, and staffing rules vary considerably across HEIs.

5 Limitations

- The study reflects two institutional implementation contexts within one province and does not establish generalisability across HEIs. No comparative outcome data is reported for the second site.
- The pooled evaluation sample of thirteen respondents is small, and the two strands differ in composition: three expert evaluators at the developing college and ten operational users at the second site. Pooling assumes the instrument measures the same constructs for both groups, which was not separately tested.
- The sprint usability ratings in Section 4.1 come from five core users during development and use a reporting scale whose underlying instrument is not preserved; they are formative only and are not comparable with the ISO/IEC 25010 results.
- Evaluator-level raw responses were not preserved, so no inferential statistical analysis is added retrospectively and no between-site comparison can be made.
- The proponents are core faculty of the college that supplied the expert evaluators and hosted the first deployment. Independence is addressed in the Declarations but cannot be established retrospectively.
- Performance Efficiency scores are perception-based Likert evaluations, not objective response-time or throughput measurements.
- JMeter raw result files and complete test-environment parameters were not retained, so objective load-performance figures are not reported.
- Complete uptime and failure-history data needed to compute Mean Time Between Failures were never recorded, so only per-incident repair durations and their mean are reported.
- The original and post-remediation OWASP ZAP reports were not retained, so no post-remediation vulnerability counts are claimed and only eight of the twenty-four Sprint 1 alerts are itemised.
- Operational outputs derive from a single snapshot of the CICT instance and cannot be independently recomputed, as the underlying conflict logs and extraction queries were not retained.
- No controlled comparison with a benchmark algorithm or a fully documented manual baseline is available.
- The algorithm description formalises documented system behaviour rather than reproducing a solver specification; exact source-code tie-breakers are not published, which limits direct reproducibility.

6 Conclusions

The study developed and implemented an integrated decision support system for class and thesis or capstone defense scheduling using shared faculty, room, workload, section, subject, and time data. The principal contribution is the integration of two institutional scheduling domains under common conflict checks, administrator validation, reporting, and audit functions, delivered on a decoupled React, Node.js,

and libSQL architecture whose relational data layer enforces the referential integrity that scheduling constraints require.

The system was deployed at two institutions on independent instances — the CICT of Sorsogon State University in December 2025 and SCCI in January 2026 — and entered live scheduling use for the term beginning 19 January 2026. In the CICT snapshot the system managed 1,245 class-schedule meeting blocks and 142 thesis and capstone defenses, recorded an overall room-utilisation indicator of 87.0% across 70 rooms, and resolved 318 conflict candidates before finalisation across automated generation and administrator review. Development-phase incident tracking produced a mean time to repair of 50 minutes across three logged failures, and a Sprint 1 security baseline of 24 OWASP alerts spanning 33 URL-level instances directed a remediation programme executed across Sprints 2 and 3. Structured ISO/IEC 25010 evaluation with a pooled sample of thirteen respondents returned an overall mean of 4.731, with all four attributes in the highest interpretation band, correctly read as respondent evaluation rather than objective engineering measurement.

Claims that depend on technical artifacts no longer available have been removed or explicitly qualified, and the scheduling method is reformulated as a constraint-aware rule-based constructive heuristic rather than an unspecified novel optimizer. Within these evidentiary boundaries the system demonstrates practical utility for integrated academic scheduling in the evaluated settings. Future work should archive complete raw test artifacts, use staging or synthetic datasets for non-disruptive technical testing, compare the scheduler against established optimization baselines, and evaluate the approach across additional institutions with reported sampling and independent evaluators.

7 Recommendations

1. **Institutional adoption.** HEIs with overlapping class and defense responsibilities may adopt the system as a primary scheduling tool, following local validation against their own policies, room structures, and academic calendars.
2. **Evidence preservation.** Preserve versioned evaluator data, questionnaires, test cases, system logs, load-test outputs, vulnerability scan reports, and environment specifications before each production release.
3. **Non-disruptive testing.** Use a staging environment with de-identified or synthetic data for load, security, and algorithmic stress testing so that production records are never affected. Where a historical test result has been lost, the recoverable course is to re-run the test against a staging instance and report it as a fresh measurement with its own date and environment specification, rather than to reconstruct the original figures.
4. **Comparative evaluation.** Conduct a controlled experiment comparing the current rule-based approach against at least one formal optimization or metaheuristic baseline on a documented instance.
5. **Re-evaluation with independent sampling.** Repeat the software-quality evaluation with a larger sample, evaluators independent of the developing unit, a reported respondent profile, and retained

item-level responses.

6. **Security re-assessment.** Conduct authenticated pre- and post-remediation scans in a controlled environment and archive the original reports.
7. **Workload analytics.** Extend the workload module to report distribution and overload exposure over time, since the deployment context indicates that chronic understaffing rather than scheduling error is the dominant driver of faculty overload.
8. **Continuous enhancement.** Extend the system with mobile access, real-time notifications, and predictive analytics for demand and workload forecasting.
9. **Capacity building.** Provide periodic training for faculty and administrative staff in data-driven scheduling practice at both sites.

Declarations

Ethical approval and consent to participate

The study was granted an ethics exemption by the Sorsogon State University Ethics Review Board under reference number SorSU-BC-ERB-2026-005, on the basis that the work constituted routine institutional system development and software quality improvement. The exemption covers the evaluation activities at both implementation sites. Deployment and expert evaluation at the CICT of Sorsogon State University were authorised internally by the college as part of its own systems operations. Deployment, beta testing, and the respondent evaluation at SCCI were authorised by a Memorandum of Agreement executed on 5 December 2025 and a permission letter approved by the SCCI administration on the same date. Both documents are retained by the authors and are available to the Editor on request. The protocol was carried out in accordance with the ethical principles of the Declaration of Helsinki. Participants were informed of the study purpose and took part voluntarily.

Competing interests

The authors are core faculty of the BSIS program of the CICT of Sorsogon State University, which is both the first deployment site and the source of the three IT expert evaluators. The evaluators are core faculty of the BSIT and BSCS programs, were selected on the basis of doctoral-level qualifications in information technology and computer science, and were not members of the development team. The usability evaluation was conducted independently at SCCI, an institution with no role in system development. The authors declare no financial competing interests and disclose the institutional relationship above so that readers may weigh the expert-evaluation results accordingly.

Consent to publish

Not applicable, as the manuscript reports no identifiable individual participant information. Sorsogon State University and SCCI are named with institutional consent. Interface figures are populated with non-production mock data; no live institutional records, student information, or faculty names are reproduced.

Funding

Not applicable.

Author Contribution

A.S.L. conceptualized the study, developed the core system architecture, and wrote the main manuscript text. R.G.R. conducted the ISO/IEC 25010 software quality evaluation, analyzed the usability data, and prepared the figures. All authors reviewed and approved the final manuscript.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

1. Abdipoor S, Yaakob R, Goh SL, Abdullah S. Meta-heuristic approaches for the University Course Timetabling Problem. Intell Syst Appl. 2023;19:200253. https://doi.org/10.1016/j.iswa.2023.200253.
2. Bashab A, Ibrahim AO, Hashem IAT, Aggarwal K, Mukhlif F, Ghaleb FA, Abdelmaboud A. Optimization techniques in university timetabling problem: constraints, methodologies, benchmarks, and open issues. Computers Mater Continua. 2023;74(3):6461–84. https://doi.org/10.32604/cmc.2023.034051.
3. Müller T, Rudová H, Müllerová Z. Real-world university course timetabling at the International Timetabling Competition 2019. J Sched. 2025;28:247–67. https://doi.org/10.1007/s10951-023-00801-w.
4. Davison M, Kheiri A, Zografos KG. Modelling and solving the university course timetabling problem with hybrid teaching considerations. J Sched. 2025;28:195–215. https://doi.org/10.1007/s10951-024-00817-w.
5. Volante DC, Olivario JCG III, Greda MR, Bayani VC. Awareness and readiness of school leaders in digital governance: challenges and strategies for implementation in the Philippine context. Int J Res Innov Social Sci. 2025;1514–34. https://doi.org/10.47772/IJRISS.2025.9020124.
6. Xue G, Offodile OF, Razavi R, Kwak DH, Benitez J. Addressing staffing challenges through improved planning: demand-driven course schedule planning and instructor assignment in higher education. Decis Support Syst. 2024;187:114345. https://doi.org/10.1016/j.dss.2024.114345.
7. Uysal M, Ceran O, Tanrıverdi M, Özdoğan E, Üstündağ MT. A web-based decision support system for managing course timetabling in online education. J Polytechnic. 2025;28(3):923–34. https://doi.org/10.2339/politeknik.1517479.

8. Siew ESK, Sze SN, Goh SL, Kendall G, Sabar NR, Abdullah S. A survey of solution methodologies for exam timetabling problems. IEEE Access. 2024;12:41479–98. https://doi.org/10.1109/ACCESS.2024.3378054.
9. Cruz-Rosales MH, Cruz-Chávez MA, Alonso-Pecina F, Peralta-Abarca JdC, Ávila-Melgar EY, Martínez-Bahena B, Enríquez-Urbano J. Metaheuristic with cooperative processes for the university course timetabling problem. Appl Sci. 2022;12(2):542. https://doi.org/10.3390/app12020542.
10. Almeida J, Figueira JR, Francisco AP, Santos D. A hybrid meta-heuristic for the generation of feasible large-scale course timetables using instance decomposition. arXiv preprint arXiv:2310.20334 [math.OC]. 2023. https://doi.org/10.48550/arXiv.2310.20334
11. ISO/IEC 25010. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. International Organization for Standardization; 2011.
12. Nielsen J. Enhancing the explanatory power of usability heuristics. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1994:152–158.
13. Gkrimpizi T, Peristeras V, Magnisalis I. Classification of barriers to digital transformation in higher education institutions: systematic literature review. Educ Sci. 2023;13(7):746. https://doi.org/10.3390/educsci13070746.
14. Singun A. Unveiling the barriers to digital transformation in higher education institutions: a systematic literature review. Discover Educ. 2025;4:37. https://doi.org/10.1007/s44217-025-00430-9.
15. Samaranayake S, Gunawardena ADA, Meyer RR. An interactive decision support system for college degree planning. Athens J Educ. 2023;10(1):101–15.
16. Nur Aji AB. Application of the A* method in making course scheduling decisions. Jatilima (Journal Artif Intell Technol Information). 2024;5(2).

Appendix A. Evidence Status

The original local research report preserves the development methodology, the ISO/IEC 25010 questionnaire description, the five-point Likert scale, item-level evaluation means, summary means, system screenshots, informed-consent documentation, and institutional approval materials. Technical claims depending on missing raw engineering logs are listed below with their status so that reviewers can see which figures are reproducible.

Figures

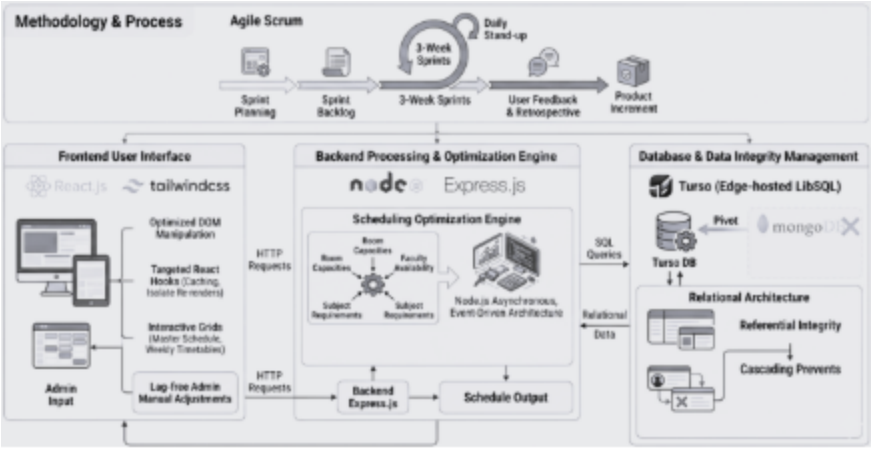


Figure 1

High-level system architecture illustrating the decoupled frontend (React.js/Tailwind CSS), backend processing and optimization engine (Node.js/Express), and relational database layer (Turso/libSQL), together with the Agile Scrum process that produced them

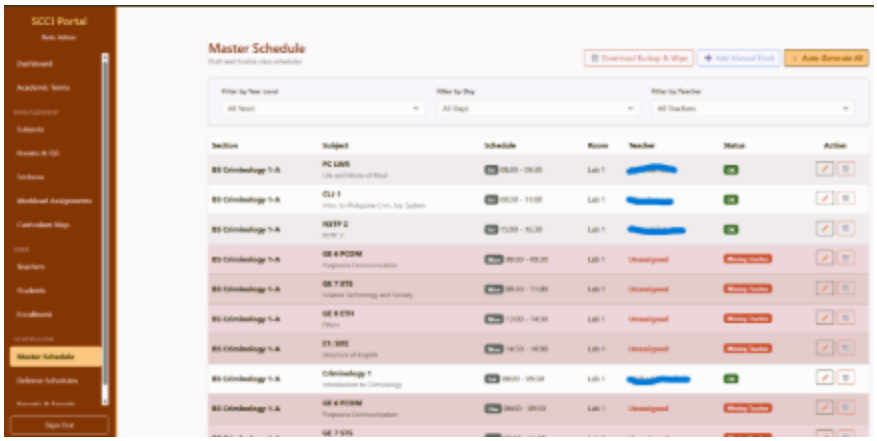


Figure 2

The Master Scheduling Dashboard, captured from the SCCI instance populated with non-production mock data for illustration. Live institutional records are not reproduced. Colour-coded status cues flag resource bottlenecks and unassigned teaching slots for administrator review before finalisation; the quantitative outputs reported in Section 4.3 derive from the CICT instance, not from the records shown here

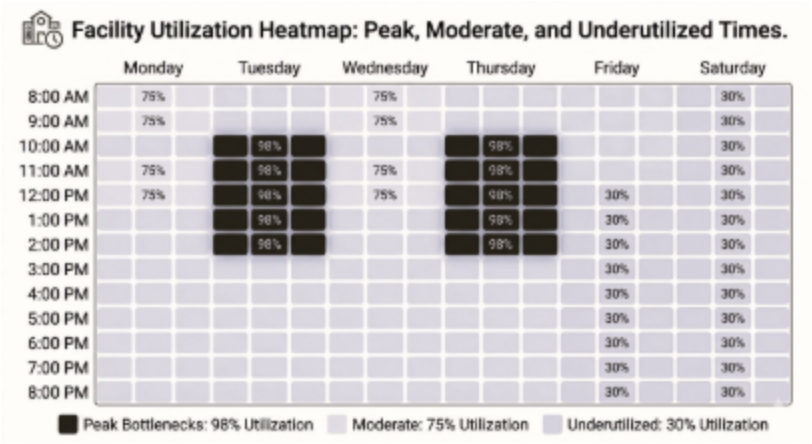


Figure 3

Facility utilisation by time and day, where darker regions indicate peak bottlenecks on Tuesdays and Thursdays between 10:00 and 14:00 at 98% utilisation. The heatmap reports instantaneous utilisation within a specific time band, whereas Table 6 reports averages across the full week



Figure 4

Capstone and thesis defense status across 142 defenses: Scheduled and Confirmed 92, Pending Panel Availability 28, Successfully Completed 15, Rescheduled or in Conflict 7. The counts sum to 142; the printed shares (65/20/10/5) are rounded from 64.8/19.7/10.6/4.9

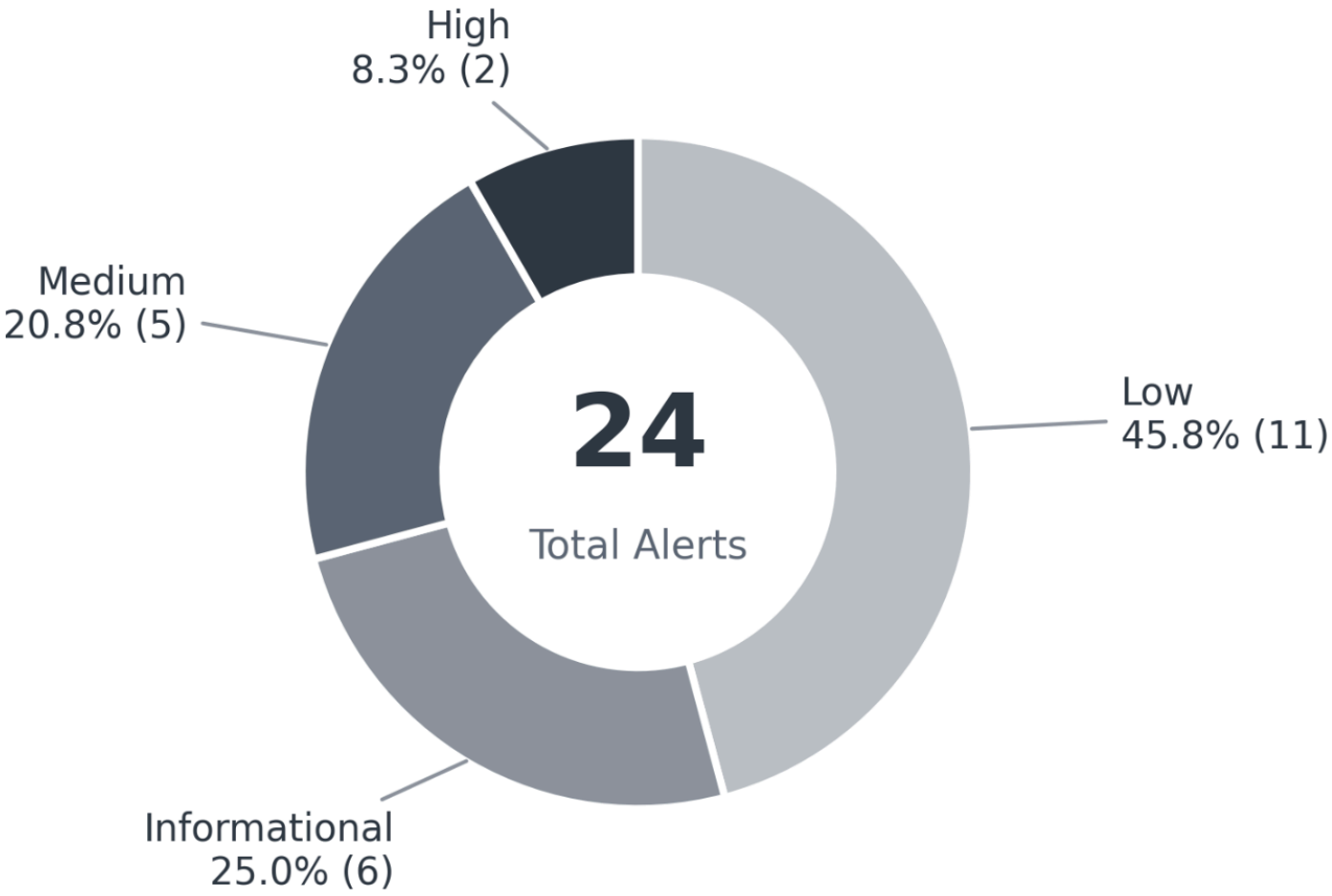


Figure 5

Vulnerability risk breakdown by alert across four severity levels: High 8.3% (2), Medium 20.8% (5), Low 45.8% (11), and Informational 25.0% (6). Shares are rounded to one decimal and sum to 99.9%



Figure 6

Alerts by OWASP Top 10 threat category: Security Misconfigurations (10), Other (5), Injection Flaws (4), Broken Access Control (3), and Cross-Site Scripting (2), summing to the 24 distinct alerts.