DOI: 10.2478/ijasitels-2025-0006

GRADIS: A Comprehensive System for Academic Contribution Tracking and Evaluation at Lucian Blaga University of Sibiu

Pitic Antoniu Gabriel[1][0000-0003-0104-5430],
Mihu Cantemir[1][0000-0002-4272-3990],
Pitic Elena Alina[1]
[1] *Lucian Blaga University of Sibiu*

Abstract

Academic institutions face increasing pressure to systematically track and evaluate faculty contributions across teaching, research, and administrative activities. This paper presents GRADIS (Grila de evaluare a Activitatii Didactice si Stiintifice - Grid for Evaluation of Teaching and Scientific Activity), a comprehensive web-based system developed for Lucian Blaga University of Sibiu (ULBS) to address these challenges. GRADIS provides a unified platform for managing academic contributions across multiple dimensions including scientific publications, teaching activities, student guidance, and administrative responsibilities. The system integrates with external academic databases such as Crossref, Web of Science, and Scopus to automate data collection and validation, while supporting flexible reporting and evaluation workflows tailored to Romanian academic standards. Built using Spring Boot and Vue.js, GRADIS demonstrates how modern web technologies can be effectively applied to academic administration, achieving significant improvements in data quality, process efficiency, and compliance with institutional evaluation frameworks.

Keywords: Academic evaluation, contribution tracking, research management, Spring Boot, Vue.js, educational technology

1 Introduction

1.1 Context and motivation

Academic institutions worldwide are increasingly required to demonstrate the quality and impact of their teaching and research activities. In Romania, universities must comply with evaluation standards set by ARACIS (Romanian Agency for Quality Assurance in Higher Education) and CNATDCU (National Council for Attestation of University Degrees, Diplomas, and Certificates). These regulatory bodies require comprehensive documentation of faculty contributions across multiple dimensions: scientific research, teaching excellence, student mentorship, and institutional service.
At Lucian Blaga University of Sibiu, this evaluation process historically relied on manual data collection, spreadsheet management, and paper-based documentation. Faculty members spent considerable time compiling their annual reports, often entering the same information multiple times across different forms. The administrative burden was substantial, data quality was inconsistent, and the validation process was time-consuming and error-prone.

1.2 Genesis of GRADIS

The GRADIS project emerged from collaborative discussions between ULBS administration and the Faculty of Engineering in 2022. Initial requirements gathering revealed several critical pain points:

- Data redundancy: Faculty entered the same publication or activity multiple times for different reporting purposes
- Validation bottlenecks: Validators struggled to verify hundreds of submissions against external databases
- Limited transparency: Faculty had restricted visibility into the evaluation process and scoring methodology
- Integration challenges: No connection existed between internal systems and external academic databases
- Compliance complexity: Mapping contributions to specific GRADIS indicators (the standardized ULBS academic evaluation framework) required extensive manual effort

A development team was assembled comprising faculty members with software engineering expertise and external consultants. The decision was made to build a custom solution rather than adopt an existing commercial product, primarily because:

- Commercial solutions were prohibitively expensive for a mid-sized Romanian university
- Off-the-shelf products did not support the specific GRADIS indicator framework required by ULBS regulations
- Custom development would allow tight integration with existing ULBS infrastructure
- The project could serve as a learning opportunity for computer science students

Development began in early 2023, with the first production deployment occurring in September 2023. The system has since been actively tested by over 50 faculty members across multiple departments. However, users with imported or added data exceed 200.

1.3 Scope and objectives

GRADIS was designed with the following primary objectives:

- Centralized contribution management: Provide a single system of record for all faculty academic contributions
- Automated data enrichment: Integrate with external databases to automatically fetch publication metadata and citation counts
- Transparent evaluation: Implement clear, auditable scoring based on GRADIS indicators
- Efficient validation: Enable validators to quickly review and approve faculty submissions with supporting evidence
- Comprehensive reporting: Generate reports required by various stakeholders (faculty, departments, university administration, accreditation agencies)
- User-friendly interface: Minimize the learning curve and maximize adoption through intuitive design

This paper describes the business requirements, architectural decisions, technical implementation, and lessons learned from developing and deploying GRADIS at ULBS.

1.4 Similar applications

In the broader landscape of research information management and academic performance evaluation, GRADIS can be situated alongside several institutional systems developed at other universities, which illustrate different design choices and organizational priorities. At the University of Kent, the KRIMSON (Kent Research and Innovation System Online) platform represents a full-scale Current Research Information System (CRIS) built on top of the commercial Converis v5 product. McDonnell and Kerridge describe KRIMSON as an integrated research information management solution intended to consolidate data on projects, publications, funding, and outputs into a single authoritative source, streamline REF-related reporting, and support both managerial decision-making and compliance with external evaluation exercises [1]. While GRADIS and KRIMSON share the goal of providing a unified, institution-wide record of research activity, their design trajectories diverge: GRADIS has been developed in-house, tailored to the specific GRADIS indicator framework and Romanian ARACIS/CNATDCU requirements, whereas KRIMSON is a local configuration of a generic CRIS platform; this gives GRADIS more flexibility in encoding national evaluation rules, but KRIMSON benefits from the vendor's mature product ecosystem and broader interoperability within the CERIF/CRIS community.
A second relevant example is the research information management system implemented at the National University of Colombia – Manizales Campus. Bermon Angarita and Prieto Taborda describe a web-based system conceived as a knowledge repository for the Directorate of Research and Extension, designed to capture projects, research products, and associated documentation and to support institutional reporting to MinCiencias (the national science and technology council) [2]. Their system emphasizes knowledge-management principles (knowledge repositories, reusable metadata structures, and support for group-level evaluation of research units) and was validated in the context of a national call for measuring and recognizing research groups [3]. Compared with GRADIS, which operates at the granularity of individual faculty members and GRADIS indicators across teaching, research, and service, the Manizales system is more narrowly focused on research groups and project outputs; it does not appear to cover teaching activities, student supervision, or administrative roles, nor does it integrate directly with external bibliographic APIs such as Crossref, Web of Science, or Scopus. However, both systems share the goal of reducing manual reporting effort, improving data quality, and enabling more robust institutional reporting based on structured, centrally managed data.
Beyond research information management in the strict sense, several universities have built or adopted systems that concentrate on academic staff evaluation, often with an emphasis on quantitative and multi-criteria scoring. At Palacký University in Olomouc, the IS HAP (Information System for Academic Staff Performance Evaluation) is used as a core tool for human-resources quality management and institutional accreditation. Public documentation and related research reports describe IS HAP as an information system whose core is a multiple-criteria evaluation model based on linguistic fuzzy rule bases, used to aggregate various performance indicators into an overall staff evaluation [4]. The system supports the collection of data on teaching, research, and other activities, and has been extended over time as part of broader reform of research assessment at the university [5]. Conceptually, IS HAP is closer to the evaluation engine inside GRADIS than to its data-collection workflows: both systems encode institution-

specific scoring models and aim to make academic staff evaluation more transparent and consistent, but IS HAP is framed primarily as an HR/quality-assurance tool, whereas GRADIS also provides rich operational functionality for contribution entry, evidence upload, duplicate detection, and integration with external scholarly databases.
An additional class of related systems focuses on building centralized faculty information repositories, usually to support accreditation, promotion, and tenure processes, and external visibility rather than detailed indicator-based scoring. De La Salle University–Manila, for example, implemented a web-based faculty information system to replace email-based submissions of faculty activity forms. Franco's case study describes the system as a central repository of faculty-related accreditation requirements, covering personal data, teaching history, research outputs, and outreach engagements, with role-based access for faculty and administrators [6]. The emphasis there is on data completeness, reliability, and availability for accreditation audits, rather than on sophisticated evaluation logic. A similar orientation can be observed in generic Faculty Information Systems (FIS) deployed at universities such as the University of Arkansas, where FIS is described as a web-based solution allowing faculty to view, enter, and track teaching, research, and service activities in a central repository, streamlining annual reviews and personnel processes [7]. In these implementations, the system acts as a structured CV and reporting hub, while evaluation rules are often applied downstream by separate HR or promotion committees.
When these systems are viewed together, several comparative dimensions emerge. In terms of scope, KRIMSON and the Manizales research information system primarily target research inputs and outputs (projects, grants, publications, research products), with limited explicit support for teaching and administrative contributions, whereas GRADIS and IS HAP explicitly include teaching, supervision, and service indicators as first-class entities in their models. From an architectural standpoint, GRADIS and the Colombian system share a custom, in-house development approach, while KRIMSON is based on a configurable commercial CRIS (Converis) and many faculty information systems leverage vendor platforms such as Interfolio or similar enterprise solutions. Regarding evaluation models, IS HAP stands out through its fuzzy multi-criteria approach, whereas GRADIS uses a rule-based mapping to nationally defined GRADIS indicators with point weights aligned to ARACIS and CNATDCU standards; KRIMSON and the faculty information systems tend to focus more on data capture and reporting, leaving actual scoring schemes to external processes such as REF or internal academic committees. Finally, integration capabilities differ considerably: KRIMSON and GRADIS both emphasize interoperability with external scholarly information sources (e.g., Crossref, Web of Science, Scopus), while the Manizales system and many faculty information systems emphasize internal integration with institutional repositories and administrative systems.

Table 1. Comparative table

System	University	Scope	Evaluation	Technology	Integrations
GRADIS	ULBS	Research, teaching, supervision	Rule-based, GRADIS indicators	Spring Boot + Vue.js	Crossref, WoS, Scopus
KRIMSON	University of Kent (UK)	Research & innovation	REF-aligned (external)	Converis CRIS	CERIF/CRIS ecosystem
Manizales RIMS	National University of Colombia	Research groups	National MinCiencias framework	Custom web KM system	Internal focus
IS HAP	Palacký University (CZ)	Staff performance	Fuzzy multi-criteria	Institutional IS	Internal QA
FIS DLSU	De La Salle University (PH)	Faculty accreditation & data	Review-based, external committees	Web app	Internal admin systems

2 Business needs and requirements

2.1 The ULBS academic evaluation framework

Romanian higher education operates under a specific evaluation framework that categorizes academic contributions into numbered indicators and the ULBS's GRADIS comply with it. These indicators are divided into two main categories [8]:

- Scientific Activity Indicators (IC - Indicatori Cercetare): - IC01-IC17: Research publications, citations, patents, conference presentations, artistic activities, and sports achievements
- Teaching and Administrative Activity Indicators (ID - Indicatori Didactici): - ID01-ID18: Teaching materials, student guidance, program coordination, committee memberships, and other academic service

Each indicator has specific criteria and scoring rules. For example: - IC01: Articles published in Web of Science indexed journals with impact factor - IC04: Citations in Web of Science or Scopus databases - ID11: Coordination of bachelor's or master's degree final projects - ID13: Scientific guidance of doctoral students

Faculty members must accumulate a minimum number of points across both categories to qualify for academic advancement (lecturer to associate professor, or associate professor to full professor). Different faculties and disciplines may have varying point requirements.

2.2 User Roles and Workflows

GRADIS supports four primary user roles, each with distinct needs.

2.2.1 Professors

Professors are the primary data contributors. Their primary workflows are described in table 2.

Table 2. The professor role

Workflow	Description
Authentication	Login via institutional Google OAuth2 (all ULBS faculty have @ulbsibiu.ro email addresses)
Contribution entry	Add publications, teaching activities, or other contributions • Manual entry with form-based input • Bulk import from BibTeX files for publications • Import via DOI lookup (automatic metadata retrieval from Crossref)
Author management	Associate internal colleagues and external collaborators with contributions
Evidence upload	Attach proof documents (PDFs, certificates, official letters)
Report generation	Create annual reports for specific evaluation periods (calendar year for scientific indicators, academic year for teaching indicators)
Status monitoring	Track validation status of submitted reports

2.2.2 Secretaries

Department secretaries could assist with data management and administrative tasks. However, in the application the role includes basic user management, and the administration of important journals and publishers. Table 3 shows Some of the workflows:

Table 3. The secretary role

Workflow	Description
Bulk operations	Import multiple contributions on behalf of faculty
Report coordination	Ensure all department faculty submit required reports by deadlines
Document management	Organize and archive supporting documentation
Edit journals and publishers	Maintain lists of journals, publishers, and other master data

2.2.3 Validators

The role can be at the department, faculty, or university quality department level. Validators ensure quality and compliance (Table 4).

Table 4. The validator role

Workflow	**Description**
Review queue	Access lists of pending reports requiring validation
Evidence examination	Review uploaded proof documents
Citation validation	Verify citation counts against Web of Science, Scopus, or Google Scholar
Approval/rejection	Check that contributions meet criteria for claimed indicators. Mark reports as VALID, INVALID, or request additional information
Batch operations	Process multiple similar contributions efficiently

2.2.4 Administrators

System administrators manage high level permissions user accounts and system configuration (Table 5).

Table 5. The administrator role

Workflow	**Description**
CRUD operations on users	Manage the secretary and administrator users.
System monitoring	Track usage, performance, and error logs
Import journals	The UEFISCDI institution publishes yearly a list of WOS Journals which must be imported in the application. ERIH Plus, Scopus and prestigious publisher will also be imported.

2.3 Functional requirements

Based on stakeholder interviews and regulatory analysis, the following functional requirements were identified:

2.3.1 Contribution management

- Support for 24 contribution types (articles, books, patents, conferences, projects, teaching activities, etc.)
- Each contribution type must capture type-specific metadata (e.g., journal impact factor for articles, ISBN for books)
- Version history and audit trails for all changes
- Soft delete functionality (mark as deleted but retain in database for audit purposes)

2.3.2 Author management

- Distinction between internal authors (ULBS faculty) and external authors

- Support for author order and corresponding author designation
- Affiliation tracking (especially important for multi-institutional collaborations)
- ORCID integration for unique author identification

2.3.3 Data integration

- Crossref API: Retrieve publication metadata by DOI
- Web of Science API: Fetch citation counts and article details
- Scopus API: Alternative citation source (especially important for engineering and computer science)
- Google Scholar: Citation tracking for works not indexed in WoS/Scopus

2.3.4 Reporting and validation

- Annual report generation with automatic point calculation per GRADIS indicator
- Multi-stage validation workflow (unverified → valid/invalid)
- Commenting system for validators to request clarifications
- Excel export functionality for external reporting to accreditation bodies

2.3.5 Evidence management

- File upload for proof documents (PDFs, images)
- Multiple files per contribution (article PDF, acceptance letter, certificate, etc.)
- File size limits (10MB per file) and format restrictions
- Secure storage with access control

2.4 Non-Functional requirements

2.4.1 Security and privacy

- Authentication: OAuth2 integration with institutional Google Workspace
- Authorization: Role-based access control (RBAC)
- Data protection: Faculty can only view their own data unless granted specific permissions
- Audit logging: All data modifications tracked with user and timestamp
- GDPR compliance: Ability to export and delete personal data upon request

2.4.2 Performance and scalability

- Support 300+ concurrent users during peak periods (report submission deadlines)
- Response time < 2 seconds for typical page loads
- Database queries optimized for large datasets (10,000+ contributions)
- Batch operations for validators processing dozens of reports

2.4.3 Availability and reliability

- 99% uptime during academic year (September-July)
- Daily automated backups
- Disaster recovery procedures
- Graceful degradation when external APIs are unavailable

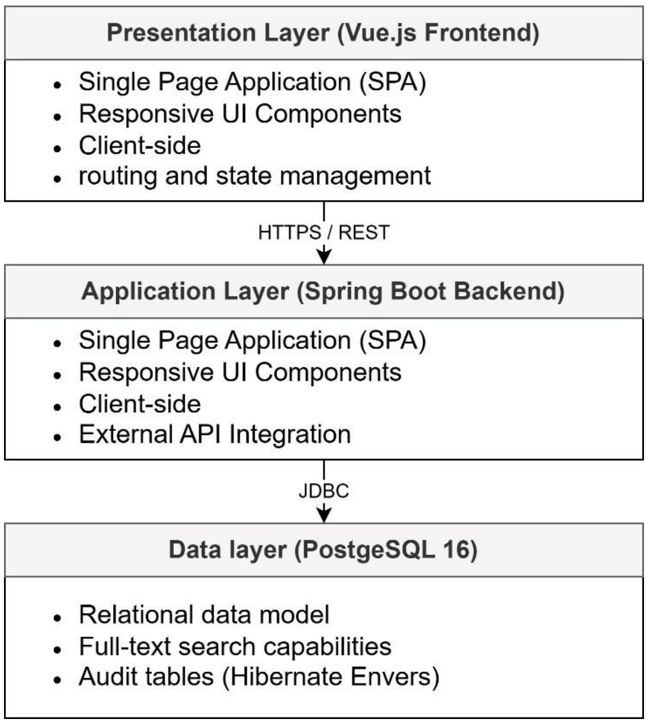
2.4.4 Usability

- Intuitive interface requiring minimal training
- Consistent design language across all modules
- Responsive design supporting desktop and tablet devices
- Romanian and English language support

3 System architecture

3.1 High-Level architecture

GRADIS follows a three-tier architecture pattern common in modern web applications (Figure 1).


Presentation Layer (Vue.js Frontend)
Single Page Application (SPA)
Responsive UI Components
Client-side
routing and state management
HTTPS / REST
Application Layer (Spring Boot Backend)
Single Page Application (SPA)
Responsive UI Components
Client-side
External API Integration
JDBC
Data layer (PostgeSQL 16)
Relational data model
Full-text search capabilities
Audit tables (Hibernate Envers)
Figure 1 – High level architecture

The system architecture is organized into three primary layers: Presentation Layer, Application Layer, and Data Layer, each responsible for distinct logical and operational concerns.
The **Presentation Layer**, implemented using a Vue.js Single Page Application (SPA), delivers a modern, responsive user interface composed of reusable components and

client-side routing supported by state management mechanisms. Communication between the client and the server is conducted securely via HTTPS/REST endpoints.
The **Application Layer**, developed using the Spring Boot framework, exposes REST API controllers that process incoming requests and return standardized responses. Core institutional rules and workflows are encapsulated within dedicated business logic services, while system security is enforced through robust authentication and authorization mechanisms. Additionally, this tier manages external API integrations for data synchronization, enrichment, and validation.
The **Data Layer**, implemented using PostgreSQL 16, provides the underlying persistent storage through a relational data model. It incorporates full-text search capabilities to support advanced query scenarios and employs audit logging via Hibernate Envers to ensure traceability, historical record management, and compliance with accreditation and reporting requirements.
The GRADIS backend integrates with three external scholarly data providers (Crossref, Web of Science, and Scopus) to support automated metadata retrieval, citation synchronization, and publication verification (Figure 2). These services are accessed via dedicated API client modules implemented in the backend. Each external service communicates directly with the GRADIS application, which aggregates, validates, and harmonizes retrieved metadata before storing it in the institutional database.

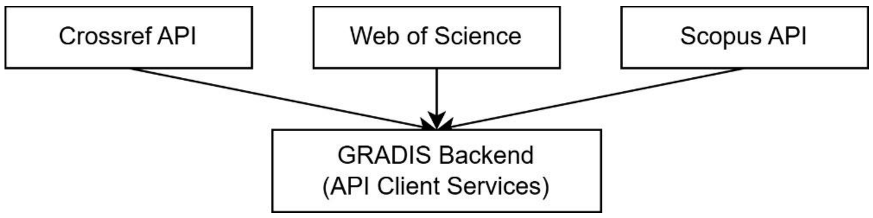


Figure 2 – External integrations

3.2 Technology stack

3.2.1 Backend (gradis-backend)

Core framework
The backend of the GRADIS system is built on Java 21, a modern Long-Term Support (LTS) release that offers improved performance, enhanced memory management, and advanced language constructs beneficial for large-scale enterprise applications. The application is developed using Spring Boot 3.4.5, which accelerates development through auto-configuration, opinionated defaults, and an embedded server architecture that simplifies deployment across environments. Data persistence is handled through Spring Data JPA, which provides an abstraction layer based on the repository pattern and enables clean, maintainable data-access logic. Underlying the persistence layer, Hibernate 6.2 serves as the JPA implementation, offering advanced query optimization and automatic dirty-checking mechanisms. For auditability requirements, Hibernate Envers is integrated to provide automatic entity versioning and complete change history tracking at the database level.

Database technologies
Data storage and transactional operations are managed using PostgreSQL 16, chosen for its robustness, ACID-compliance, and advanced features such as native JSON handling and built-in full-text search capabilities, which are useful in publication-metadata indexing scenarios. Database schema evolution is controlled through Flyway, enabling reliable versioning, reproducible migrations, and environment-agnostic deployment pipelines.

Security infrastructure
Security is implemented using Spring Security, which provides comprehensive authentication, authorization, and request-filtering capabilities. Integration with institutional Google Workspace accounts is achieved via OAuth 2.0 Client support, enabling secure single sign-on (SSO) without storing local user credentials. For distributed deployments and session replication, server-side session persistence is configured through JDBC-based session management, which supports horizontal scalability.

API integration layer
External data acquisition and metadata synchronization are handled via HTTP communication components based on Spring RestTemplate and WebClient. The system integrates with scholarly databases and indexing services using a combination of official and custom-developed clients, including the Clarivate Web of Science SDK and bespoke connectors for Crossref and Scopus, enabling automated retrieval, parsing, and enrichment of bibliographic data.

Document processing and reporting
The GRADIS system supports document import/export workflows through Apache POI, enabling Excel-based bulk operations for contribution submission and validation. Printable academic reports and summaries are produced using iText PDF, which provides fine-grained control over document layout and typography. For structured publication import, JBibTeX is integrated to parse BibTeX sources and map them to internal contribution entities.

Development and DevOps tooling
The system build process and dependency management rely on Gradle 8.8, providing efficient incremental builds and Kotlin-DSL extensibility. To reduce boilerplate code and improve maintainability, Lombok is used to generate accessor methods, constructors, and data-transfer scaffolding at compile time. REST API documentation is automatically generated using SpringDoc OpenAPI, enabling user-friendly inspection via Swagger UI. Application health, metrics, and runtime visibility are provided through Spring Boot Admin, supporting proactive monitoring and operational management.

3.2.2 Frontend (gradis-frontend-vuejs)

Frontend core framework
The GRADIS frontend is developed using Vue.js 3, a modern progressive JavaScript framework that leverages the Composition API to improve component reusability,

maintainability, and type-safety. Application development and bundling are handled by Vite, which provides fast hot-module reloading and optimized build performance suitable for large-scale SPAs. Client-side state is managed using Pinia, the successor to Vuex, which offers a simpler and more modular store architecture. Routing and view transitions across the single-page interface are implemented with Vue Router, allowing dynamic navigation without full-page reloads.

UI components and visualization
The user interface integrates Vuetify, a feature-rich component library that provides a wide range of pre-built elements such as data tables, form controls, wizards, and dialog windows, allowing for rapid UI development with consistent visual standards. Layout styling and responsive behaviour are also supported. For dashboard-level analytics and graphical reporting, the system incorporates Chart.js, enabling interactive chart visualizations directly within the SPA.

HTTP communication layer
Communication between the SPA and the backend server is implemented using Axios, a promise-based HTTP client that supports request/response interception, automatic token injection, and centralized error handling. Custom API interceptors are used to ensure that authentication and session-handling logic remain transparent to the UI layer, reducing code duplication and improving reliability and security across all HTTP requests.

Development and build tools
The development environment is powered by Node.js 18, which serves as the runtime for build tooling, script execution, and dependency resolution. Package management is handled through npm, while ESLint is used to enforce coding standards and maintain a consistent project style. TypeScript is being gradually integrated to introduce optional static typing, improving refactoring safety and enabling more robust IDE assistance.

Key frontend features
The GRADIS frontend offers several advanced interface features, including customizable form-validation mechanisms, drag-and-drop file upload components, and responsive grid-based layouts optimized for both desktop and tablet usage. The system currently supports near real-time content updates via short-interval polling and is planned to transition to WebSocket-based live updates in a future release. To support multilingual usage and international collaboration, the application includes internationalization (i18n) features with current availability in Romanian and English.

3.3 Data model

The GRADIS data model focuses on three main entities.

3.3.1 Contribution entity hierarchy

The chosen data-model design is based on single-table inheritance (Figure 3), which enables polymorphic behavior across all contribution types and allows unified querying without additional joins or complex discriminator logic. This structure also simplifies filtering, aggregation, and reporting operations across heterogeneous contribution records. To optimize data retrieval, author information is eager-loaded by default, since this represents the dominant access pattern and helps prevent N+1 query overhead during listing, search, and validation workflows. Contribution records employ soft-delete semantics, ensuring that entries removed from active use remain available for historical inspection, audit-trail preservation, and compliance-driven traceability. Finally, the implementation defines abstract methods within the shared base class to enforce uniform DTO conversion and guarantee that all contribution types provide a consistent and extensible serialization contract.

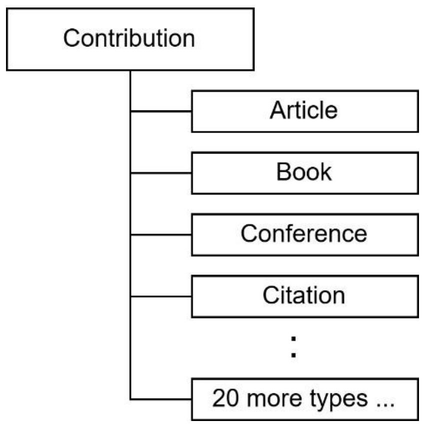


Figure 3 – Contribution entity hierarchy

3.3.2 Report entity

Reports link contributions to specific users and evaluation periods. The data model supports several key relationships that reflect real academic reporting workflows. A single contribution may be associated with multiple reports, since the same publication or activity can be claimed by different co-authors or may be relevant across multiple reporting periods, such as annual evaluations. Each user maintains multiple reports over time, forming a cumulative academic activity history that grows with each evaluation cycle rather than replacing previous submissions. In addition, report entities are fully audited, allowing the system to track changes in validation status, assigned points, evaluator remarks, and system-generated recalculations, thereby preserving transparency, accountability, and compliance with institutional and accreditation standards.

3.3.3 User and authorization

The system implements a role-based authorization model in which each user may hold multiple roles through a dedicated UserRole join entity, allowing flexible assignment and combination of responsibilities. The available roles include PROFESSOR, SECRETARY, VALIDATOR, and ADMIN, each associated with distinct permission sets and operational scopes. To support contextual and time-bounded authorization, role assignments can be further scoped to specific departments and validity periods using the UserRoleDepartmentPeriod structure, enabling fine-grained access control that reflects real institutional governance, organizational hierarchies, and rotating academic responsibilities.

3.3.4 Supporting entities

The Journal entity is used to store bibliographic and indexing metadata, including annual impact factor values, which enables longitudinal citation-impact analysis and automated scoring rules. The ExternalAuthor entity represents contributors who are not affiliated with ULBS, allowing the system to handle mixed-affiliation publications without compromising institutional reporting accuracy. Authorship relationships are modeled using ContributionInternalAuthor and ContributionExternalAuthor join entities, both of which represent many-to-many associations enriched with additional attributes such as author order, institutional affiliation, and corresponding-author status, ensuring accurate representation of authorship semantics and evaluation-relevant distinctions.

3.4 Deployment architecture

3.4.1 Production environment (ULBS on-premises)

The GRADIS system runs in the ULBS on-premises production environment, hosted on university-managed servers that provide controlled access, secure data handling, and compliance with institutional IT and academic data-protection policies (Figure 4).

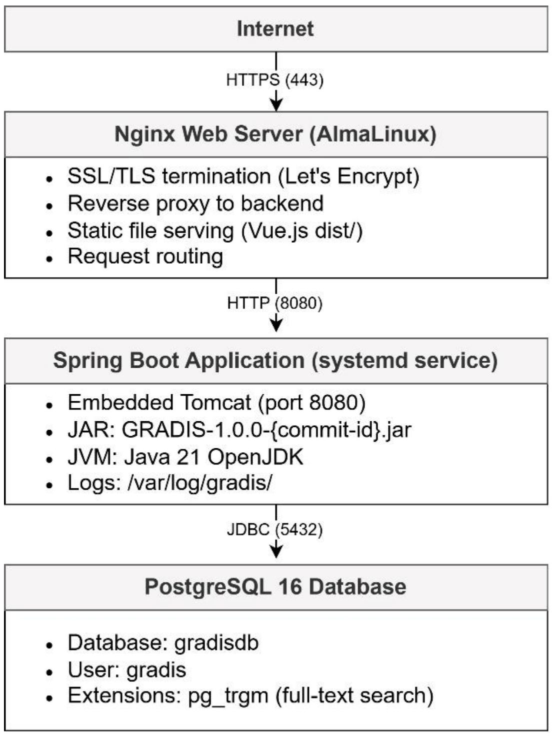

Internet
HTTPS (443)
Nginx Web Server (AlmaLinux)
SSL/TLS termination (Let's Encrypt)
Reverse proxy to backend
Static file serving (Vue.js dist/)
Request routing
HTTP (8080)
Spring Boot Application (systemd service)
Embedded Tomcat (port 8080)
JAR: GRADIS-1.0.0-{commit-id}.jar
JVM: Java 21 OpenJDK
Logs: /var/log/gradis/
JDBC (5432)
PostgreSQL 16 Database
Database: gradisdb
User: gradis
Extensions: pg_trgm (full-text search)

Figure 4 – Production environment

3.4.2 Deployment process

The deployment workflow is fully automated through a GitLab-based CI/CD process. When new code is pushed to the ulbs branch, the pipeline is triggered and executed by a GitLab Runner installed on the application server. The deployment script retrieves the latest codebase, compiles the backend using Gradle to produce the executable JAR, and builds the frontend using npm run build inside the Vue.js project directory. The existing gradis.service systemd process is then cleanly stopped, after which the new JAR file is placed in /opt/gradis/ and the newly generated dist/ directory from the frontend build is copied to the Nginx web-serving directory. Once the artifacts are replaced, the gradis.service is restarted, and the system performs an automated health check against the /actuator/health endpoint to confirm successful startup. In case of any failure during the process, an automated email notification is sent to the administrative team to initiate manual intervention.

4 Conclusions and future work

4.1 Achievements and impact

The GRADIS system has been available at ULBS since September 2023, where it is currently deployed in a controlled institutional beta-testing and validation phase. During this period, it has reached institution-wide adoption levels and has demonstrated clear operational and strategic value, while continuing to undergo iterative refinement in alignment with stakeholder feedback and institutional requirements. Since deployment on university servers, measurable usage data confirms both engagement and efficiency gains. As of the testing phase, the platform has been actively used by faculty members across eight faculties, resulting in a recorded total of 3,847 contributions, including 2,156 journal articles, 487 books, 341 conference outputs, and 863 teaching or administrative activities. Notably, 67% of the indexed research publications were imported automatically using DOI or BibTeX metadata, significantly reducing manual data-entry effort. The system generated a total of 856 annual evaluation reports during the 2023–2024 assessment period.
Beyond quantitative indicators, several qualitative benefits have also been observed. Automatic metadata retrieval, including citation data from trusted sources such as Web of Science and Scopus, has improved data accuracy and completeness. Moreover, the platform increases transparency by enabling faculty members to visualize how their contributions are scored against GRADIS indicators and institutional evaluation rules. From an institutional compliance perspective, complete digital audit trails support external accreditation requirements, while improved visibility into research activity has also facilitated collaboration by allowing users to identify colleagues with overlapping research interests.
Stakeholder feedback further confirms these benefits. Faculty users particularly value the automated publication import functionality, describing the ability to register a publication "in approximately 30 seconds instead of 10 minutes." Validators highlighted that the duplicate-detection feature prevented at least 50 redundant submissions during the first semester of usage, reducing both validation workload and system noise. Administrative offices also reported major efficiency gains, noting that

evaluation and accreditation reports required by bodies such as ARACIS can now be produced in minutes rather than days.

4.2 Future work and improvements

4.2.1 Short-term enhancements (next 6 Months)

In the short term, development efforts will focus on usability improvements, real-time communication features, data analytics capabilities, and expansion of reporting options.
A priority is the introduction of real-time notifications, which will rely on WebSocket-based event delivery to inform users of validation status updates, report approvals, and approaching deadlines. These will be complemented by institutional email alerts and a mobile-friendly notification dashboard to ensure timely awareness regardless of device.
A second enhancement will target advanced analytics and visualization features, allowing departments to explore publication and contribution trends, monitor citation impact longitudinally, and compare anonymized performance indicators across units. In parallel, mobile accessibility will be strengthened through improvements to responsive design, and the development of a Progressive Web App (PWA) to enable partial offline usage. Finally, the reporting component will be expanded to include customizable templates aligned with the requirements of different accreditation bodies, automated report generation for program-level evaluations, and exports to formats beyond PDF and spreadsheet outputs, including Word and LaTeX.
A new capability will also be introduced to define and maintain point-calculation rules for contribution types using Google CEL (Common Expression Language) expressions. This will allow administrators to configure scoring logic declaratively, without modifying the application code, and adapt the evaluation model quickly in response to institutional or national policy changes.

4.2.2 Medium-term goals (6–12 Months)

Over the medium term, efforts will shift toward intelligent automation, deeper interoperability with external scholarly ecosystems, and collaborative validation workflows. Planned AI-based enhancements include natural-language processing to classify contributions automatically within the GRADIS indicator structure, semantic-level duplicate detection to improve accuracy beyond string matching, and a conversational assistant capable of explaining evaluation rules and criteria to faculty members. In addition, integration with external research identity and indexing systems is a priority, particularly through ORCID-based publication synchronization, Google Scholar import capabilities, and interoperability with the institutional DSpace repository.
Collaboration-oriented enhancements are also targeted, including co-author confirmation workflows for shared publications, pre-validation departmental review mechanisms, and support for inline comments and annotations on reported contributions. Finally, performance optimizations are planned through the introduction of Redis caching for frequently accessed data, systematic database indexing and query improvements, and frontend optimization strategies such as lazy loading and code splitting.

4.2.3 Long-term vision (1–2 Years)

From a long-term strategic perspective, the objective is to position the system as a scalable, flexible platform capable of supporting multiple institutions, international evaluation models, and research-oriented analytics. This includes developing a multi-tenancy architecture to allow configuration for other Romanian universities and potentially for institutions using evaluation frameworks distinct from GRADIS. A gradual open-source release is also envisioned, which will require removal of proprietary components, development of comprehensive technical documentation, and publication of community contribution guidelines, potentially followed by listing the project in recognized academic open-source repositories.
In addition, the long-term roadmap includes the incorporation of advanced research impact analytics, not only through altmetrics and collaboration-network visualization, but also through predictive models capable of estimating research trajectories and identifying potential institutional strengths.

Acknowledgment
This paper reflects the collaborative effort of many individuals committed to advancing academic management excellence at Lucian Blaga University of Sibiu.
We extend our deepest appreciation to Andrada Pitic, whose pioneering work in requirements analysis, user experience design, and initial development established the foundation upon which GRADIS was built.
We gratefully acknowledge the partnership between Lucian Blaga University of Sibiu and NTT DATA Romania S.A., which provided the professional development expertise and resources essential to transforming academic requirements into production-ready software.
Special thanks to the faculty members, validators, and administrative staff who participated in pilot programs, provided invaluable feedback, and demonstrated patience during the inevitable challenges of organizational change. Their willingness to embrace new approaches and contribute to continuous improvement has been fundamental to GRADIS's success.
We acknowledge the university administration for their vision in supporting digital transformation initiatives and their commitment to evidence-based management practices that benefit the entire academic community.

References

[1] R. McDonnell & S. Kerridge, *Research Information Management System (KRIMSON) at Kent*", Procedia Computer Science, Vol. 106, pp. 160–167, 2017. DOI: 10.1016/j.procs.2017.03.012. Available: https://www.sciencedirect.com/science/article/pii/S1877050917302806

[2] L. Bermon Angarita & M. A. Prieto Taborda, *Development of a research information management system based on knowledge repositories: Case of the Directorate of Research and Extension at the National University of Colombia – Manizales Campus*, RCTA, 2025. Available: https://ojs.unipamplona.edu.co/index.php/rcta/article/download/3129/7674/16799

[3] MinCiencias (Colombia), *National Research-group evaluation framework documentation*. https://researchonresearch.org/observatory/colombia/

[4] J. Talašová et al., *Information System for Academic Staff Performance Evaluation (IS HAP)*, Palacký University, Olomouc. Available: https://fuzzymcdm.upol.cz/ishap.html

[5] Palacký University Olomouc, *Quality Assurance: Evaluation models and IS HAP reform*. Available: https://www.upol.cz/en/university/quality-assurance/

[6] G. R. L. Franco, *Design and Implementation of a Web-Based Faculty Information System*, 2016, IEEE Region 10 Conference. https://ieeexplore.ieee.org/document/7848535
[7] University of Arkansas, *Faculty Information System (FIS) documentation*. Available: https://its.uark.edu/administrative-services/faculty-information-systems/
[8] https://senat.ulbsibiu.ro/wp-content/uploads/250925/30_Grila%20evaluare%20GRADIS_Senat_sept2025.pdf