iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

A SYSTEM DESIGN FOR THE DIGITIZATION OF STUDENT RECORDS IN THE REGISTRAR'S OFFICE

Leslyn Bonachita Reazol
ORCID ID – 0009-0006-0230-1837
https://orcid.org/0009-0006-0230-1837
University of Northwestern Mindanao, Tangub City, Misamis Occidental, Philippines

Eljun Karl C. Atay
University of Northwestern Mindanao, Tangub City, Misamis Occidental Philippines

ABSTRACT
The risks to paper records of student information include: misused files, physical deterioration, lack of organized backups, lost retrievals, and unauthorized access. This study was developed to help create a digital record-keeping system for Tangub City Global College Registrar's Office through the definition of a technical system design. The documented record-keeping problems, relevant literature, established technology standards, and privacy requirements were integrated together into a logical technical plan using both scientific design and documentation research methods. Required technologies for the record-keeping system were evaluated on factors such as institutional compatibility; cost; scalability; interoperability; system security; and maintenance. The proposed technical design deliverables include: data-flow diagrams; entity-relationship diagrams; relational database designs; three-layer cloud architectures; low-fidelity wireframes of the user interface; role-based and permission-based security models; processes for creating secure backups and recovering from loss; and pseudocode for digitizing records. The risks to paper records of student information include: misused files, physical deterioration, lack of organized backups, lost retrievals, and unauthorized access. This study was developed to help create a digital record-keeping system for Tangub City Global College Registrar's Office through the definition of a technical system design. The documented record-keeping problems, relevant literature, established technology standards, and privacy requirements were integrated together into a logical technical plan using both scientific design and documentation research methods. Required technologies for the record-keeping system were evaluated on factors such as institutional compatibility; cost; scalability; interoperability; system security; and maintenance. The proposed technical design deliverables include: data-flow diagrams; entity-relationship diagrams; relational database designs; three-layer cloud architectures; low-fidelity wireframes of the user interface; role-based and permission-based security models; processes for creating secure backups and recovering from loss; and pseudocode for digitizing records.

Keywords:
Cloud-Based System, Student Records, Registrar's Office, Document Management, Record Digitization

INTRODUCTION

The Registrar's Office is responsible for maintaining the records that students need in order to enroll, advance academically, get certified, graduate, and report to the institution with accurate compliance to regulations. The majority of records that are kept in paper folders are often difficult to retrieve, improperly classified, duplicated, physically deteriorating, not able to trace back, and challenge to recover once they have been lost/damaged. Studies based on registration processes and the services offered by records management throughout the Philippines have also concluded that organized digitization as well as credible document management processes are needed (Duruin & Siddayao, 2024; Falolo et al., 2022).

Organizations can enhance their record organization and access through electronic document management. Digitization alone does not define what is meant by "digitizing" paper documents; it encompasses the digitization of other types of documents, too. A viable institutional design must have defined metadata, database relationships, user roles and access restrictions, audit events, storage locations, integrity and backup procedures, and user/system interaction. According to (Aliazas et al., 2024), electronic document management in higher education supports

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

administrative efficiency and document security, while user readiness and having appropriate organizational processes are important factors that should be considered in any implementation.

Due to the personal and academic nature of student records, they're part of a privacy and security regime. Accordingly, Republic Act No. 10173 (also called "The Data Privacy Act of 2012") establishes that all personal information found on ICT (Information and Communications Technologies) Systems must be managed in a secure manner (National Privacy Commission, 2020). The Privacy Advisory for the Education Sector elaborates on the need for appropriate and secure handling of student personal data transferred electronically in support of educational activities (National Privacy Commission, 2023). Therefore, security planning must be included in both the architecture and data models of information systems during the initial design stages and not just added after system development has been concluded. To assist with creating identity management, access control, data protection, secure logging, and recovery planning for application development, the NIST Cybersecurity Framework 2.0 and the OWASP Application Security Verification Standard should serve as reference tools during software development (National Institute of Standards and Technology, 2024; OWASP Foundation, 2025).

A large number of system studies move straight from systems analysis through software design and testing to the use of the new system (i.e., the implementation of the new system). By contrast, this study is primarily concerned with the pre-implementation phase of the system life cycle; that is, the study aims to identify candidate technologies, describe the information framework to support the operation of the system, design the database to contain the information needed, provide mapping for system processes and architectures, develop interface wireframes, and define security and continuity protocols for the system. The results of this study intend to assist the future development team, as well as reduce ambiguities prior to starting software coding.

Therefore, an example of a cloud-based design and system to digitize and manage Student Records in the Registrar's Office of the Tangub City Global College will be developed by this project, but operational software has neither been developed nor will be developed & deployed. The contribution of this project will be a collection of design artifacts linked together that illustrate how a future system for organizing records, controlling access, preserving integrity, documenting user activity, and supporting backup and recovery could be designed.

OBJECTIVE OF THE STUDY

General Objective

The general objective of this study is to develop a proposed system design for the digitization and management of student records in the Registrar's Office of Tangub City Global College.

Specific Objectives

Specifically, the study aims to:

1) analyze the limitations and risks associated with the existing paper-based handling of student records.
2) identify the functional, non-functional, data, compatibility, and security requirements of the proposed system.
3) assess and recommend suitable technologies for the proposed cloud-based student-record digitization environment.
4) design the data flow, entity relationships, database structure, and three-layer system architecture.
5) create low-fidelity user-interface wireframes for the principal registrar workflows.
6) specify protocols for authentication, role-based access, audit logging, encryption, integrity checking, backup, and recovery.
7) develop an integrated record-digitization model and pseudocode that can guide future implementation.

LITERATURE REVIEW

Records digitization is an organizational and technical activity. (Falolo et al., 2022) examined registration and records-management services as a basis for academic records digitization and showed that current service conditions should inform the design of future digital processes. (Duruin & Siddayao, 2024) additionally, the relevance of structured record management systems in education has been displayed. These studies support the use of current record workflows and document classifications as input into system requirements.

Electronic document management systems provide a centralized method for organizing and locating institutional files. In a higher-education setting, (Aliazas et al., 2024) the benefits of using an electronic document management system include improved access to documents and reduced administration time. A digital document repository must also provide solutions for the classification, accountability, privacy, and recovery of files. Using a separate, but connected, set of information for a digital file and its associated metadata will simplify these processes. In this

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

architecture, a database retains the structured description and control of the digital file, while a separate object-storage service holds the actual digital document.

The approach of the study is design science based. Design science research has recently been described as a problem-solving method that produces artifacts and design knowledge (vom Brocke et al., 2020). From a design science point of view, issues such as how to define the problem, develop a solution, instantiate and validate a solution, the relevance of the solution, the rigor of the solution, and the innovative quality of the solution can be clarified through the study of software engineering (Engström et al., 2020). For this study, all of the artifacts created through this study are based on a process model, database model, architecture, wireframes, protocols, and pseudocode, and do not consist of a working application.

The utility of data flow diagrams lies in their ability to represent both external sources & processes and how data flows within an organization (Aleryani, 2024). On the other hand, entity-relationship diagrams assist in providing definitions for the entities, their attributes, the identifiers that uniquely identify them and how they relate to one another in order to develop databases (Bagui & Earp, 2022). Combined these two tools will help to define a clear picture of what the system is supposed to process and how the data will be organized.

The design constraints included software quality and security references. ISO/IEC 25010:2023 contains a product quality model for determining the requirements for functional suitability, reliability, security, maintainability, compatibility, and usability (International Organization for Standardization., 2023). ISO/IEC 27001:2022 specifies the requirements for information security management systems and NIST SP 800-53 Rev. 5 provides a catalogue of security and privacy controls covering access control, audit and accountability, identification and authentication, contingency planning, system integrity, and other security safeguards (International Organization for Standardization, 2022; Joint Task Force, 2020). NIST CSF 2.0 further defines cybersecurity outcomes along the lines of governance, identification, protection, detection, response, and recovery (National Institute of Standards and Technology, 2024). OWASP ASVS Version 5.0 provides up-to-date web application security requirements applicable to authentication, access control, validation, cryptography, log files, secure communication, and file handling (OWASP Foundation, 2025).

For the tech layer, the design looks to have a responsive web interface, supported application framework, relational database and object storage. The candidate interface framework that Bootstrap was included in is due to its ability to support responsive (browser-based) layouts with its latest version having responsive interface components (Bootstrap Team, 2024). PostgreSQL is included because it's able to continue providing a platform for reliable relational data management, indexing, authentication integration, and extensibility (PostgreSQL Global Development Group, 2025). S3-compatible object storage can be optionally used in conjunction with certain relational databases to separate large document files from transactional database records and to provide identity-based access control, encryption, versioning, monitoring, and security practices for cloud architectures (Amazon Web Services, 2024). The recommendations of these technologies as candidate components stem from their respective studies; however, there is no assertion in either study that (1) these technologies were the subject of actual deployments or (2) these technologies were actually benchmarked through any sort of testing process.

METHODOLOGY

The current research will concentrate on the design process of a proposed cloud-based solution to the digitization and administration of student records in the Registrar's Office of Tangub City Global College. The research is not aimed at developing, deploying, or implementing the proposed system. Therefore, the research does not guarantee the success of uploading, searching, backing up, restoring, or recovering student records using a software application.

The primary output of this study includes the design of a cloud-based solution for the digitization of student records; Data Flow Diagrams; Entity-Relationship Diagram; database design; user interface layout; security measures; record digitization process; algorithm design; and pseudocode. The evaluation of design outputs will be done through an expert review process and scenario-based validation, but not through software testing.

Research Design

The research methodology used in this study is an applied System Design utilizing design science methodology. The reason for using this method of research is to create and document technical artifacts that resolve a specific record management issue for this study. The selection and documentation of requirements, architecture, design, quality and security were also supported by the software engineering domains from SWEBOK Guide Version 4.0 (IEEE Computer Society, 2024). The study does NOT include software coding, cloud deployment, operational performance testing, penetration testing, or user acceptance testing.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

The design phases are broken down into six sequential steps: (1) problem definition and requirements; (2) technology evaluation; (3) process and data modeling; (4) architecture and interface design; (5) defining security, integrity, and continuity requirements; (6) verifying the design by tracing requirements back to test cases and walking through scenarios. This process is similar to the most current design science literature about how to create artifacts, understand problems, develop solutions, and validate solutions (Engström et al., 2020; vom Brocke et al., 2020).

Design Context and Sources

In this design process, Tangub City Global College has chosen to use the Registrar's Office as the basis for the research study. The Registrar's Office handles all the activities involved with academic records, including receiving, sorting, maintaining, retrieving, updating, and releasing records. To aid in the design, the research team used the workflow processes of how paper-based methods are used to produce the various types of records (registrar), forms completed by students and their record classifications (form type), other studies related to how the records are used, technical documentation used by the organization's registrar department, quality standards for software, and guidelines for privacy and cybersecurity.

During the study, a non-production cloud environment and/or live database of students' records was not used, so no confidential student records were posted or available to the public. Illustrations of interfaces and fields are provided with completely fictitious and/or representative data. The limitation of using solely fictitious values in the study supports the design of the study by eliminating the potential for confidentiality breaches while preparing artifacts.

Table 1:Design Inputs and Corresponding Outputs

Design input	Information examined	Resulting artifact
Paper-based workflow	Receiving, classification, filing, retrieval, updating, and release activities	Process requirements and Data Flow Diagram
Representative registrar forms	Student identifiers, document categories, dates, status fields, and metadata	Entity-Relationship Diagram, schema, and data dictionary
Technical literature and official documentation	Architecture, database, interface, storage, and security capabilities	Recommended technology stack and architecture
Privacy and security standards	Access control, data protection, logging, incident response, backup, and recovery	Security-protocol and continuity framework
Dummy records and scenarios	Illustrative values and workflow conditions	Wireframes, pseudocode, and scenario walkthroughs

Technology Assessment Criteria

The candidate technologies that were evaluated based on six criteria; features for security, maintenance, interoperability, scalability, and cost as well as hosting facilities as a reasonable solution for management of established records at an institution. The candidate assessment was qualitative due to the lack of any software platforms having been deployed or evaluated (at least in terms of performance). Therefore, when referring to a candidate as "recommended", it refers to a decision for a design solution meeting all of the identified requirements. It does not imply an overall superior technology.

- Security: support for authentication, authorization, encryption, secure configuration, auditability, and patching.
- Maintainability: availability of documentation, modular architecture, active support, and developer familiarity.
- Interoperability: ability to expose standard web interfaces and integrate with storage, database, and identity services.
- Scalability: ability to accommodate additional records, users, document types, and storage capacity.
- Cost and practicality: compatibility with commonly available institutional hosting and open-source components.
- Data suitability: support for transactions, constraints, indexing, structured relationships, and controlled document storage.

Design Development Procedure

1) **Requirements definition.** The limitations of paper-based record handling were translated into functional and non-functional requirements. Each requirement was assigned to one or more design components.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

2) **Data Flow Diagram development.** External users, principal processes, data stores, and intended data flows were mapped to show the logical movement of credentials, metadata, documents, requests, audit events, and backup instructions.
3) **Database design.** Core entities, primary keys, foreign keys, attributes, and cardinalities were defined. Structured metadata was separated from binary document storage so that the database could index and retrieve file references without storing large documents directly in transactional tables.
4) **Architecture design.** The system was divided into presentation, application, and data layers. Security and access-control concerns were modeled as cross-cutting controls that apply to all layers.
5) **Interface wireframing.** Low-fidelity screens were created for login, dashboard monitoring, student and document metadata entry, record search, requests, and audit review. The wireframes specify content and navigation but do not represent a coded interface.
6) **Security and continuity design.** Authentication, role-based authorization, least-privilege permissions, metadata validation, checksum generation, encryption, audit logging, backup scheduling, versioning, retention, and recovery responsibilities were documented.
7) **Algorithm and pseudocode development.** The intended sequence for authorization, validation, duplicate checking, checksum generation, storage assignment, audit logging, and backup scheduling was expressed as implementation-neutral pseudocode.

Design Verification

To verify if the artifacts met the requirements, many of them underwent 2 key activities; requirements-to-design traceability and scenario walkthroughs. The traceability assessed whether an identified concern has an associated process, database element, interface component, and/or control to mitigate a risk. The scenario walkthroughs evaluated each of the identified concerns against typical/scenario-based conditions; e.g., a new document with complete metadata, a new document with incomplete metadata, a potential duplicate document, an unauthorized access attempt, an approved search request, a checksum mismatch, and a failed attempt to backup. The outcome from the 2 activities only provided evidence of logical coverage and relative consistency; they did not assess operational performance.

RESULTS AND DISCUSSION

Recommend Candidate Technology Stack

The Technology Assessment delivered a potential stack of components for separating the interface, application logic, structure data and document files. Separating these components supports maintainability and allows for the replacement of storage or application component parts without redesigning the total Information Model. Because the study did not perform comparative performance testing, the recommendations provided were based only on fit to requirements and on the capabilities of each product as documented, (not based upon measured superiority).

Table 2: Recommended Candidate Technologies for Future Development

Component	Recommended candidate	Design rationale
User interface	HTML5, CSS3, JavaScript, and Bootstrap or an equivalent responsive framework	Supports browser-based access, responsive layouts, reusable form controls, and consistent navigation.
Application layer	Laravel or an equivalent supported server-side framework	Provides structured routing, validation, authorization policies, database access, logging, and storage abstraction.
Relational database	PostgreSQL	Supports transactions, constraints, indexing, roles, privileges, and row-level security for structured record metadata.
Document storage	Private S3-compatible object storage	Separates files from the database and can support encryption, versioning, retention rules, and controlled download links.
Authentication	Individual institutional accounts; multi-factor authentication for supervisors and administrators; session timeout; account lock or rate limiting after repeated failures (Temoshok et al., 2025).	Reduces dependence on passwords alone and supports role-based access.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

Component	Recommended candidate	Design rationale
Integrity control	SHA-256 or a stronger approved cryptographic hash	Provides a reproducible checksum for identifying unintended file changes.
Transport security	HTTPS using a currently supported TLS configuration	Protects credentials, metadata, and files while transmitted between the browser and server.
Deployment and backup	Supported Linux hosting, automated database backups, versioned object storage, and an off-site backup copy	Supports separation of services, scheduled backup, and recovery planning.

The favored relational database, PostgreSQL, has been selected because the design implementation is heavily reliant upon strict relationships, uniqueness, transactional updates and granular security throughout student record metadata via the ability to set constraints; and thus both "constraints" and "row-level security" mechanisms are substantially warranted related to the lack of required identifying information necessary for a record to be considered valid by any entity, or because the user does not have authorization regarding the visibility of a given piece of data; public file directory access has not been used to provide access to documents to authorized users via public web directories (i.e. documents should only be retrieved via authorized requests to applications) rather than via predictable web directories. The application framework used for this architecture is expected to be replaceable. Laravel is an example of an acceptable application framework since its released documentation offers various features that will be useful to the implementation of the proposed architecture (e.g. user validation/authentication/authorization, document encryption/hashing, error logging, the ability to store files, rate limit and monitor system health) (Laravel, 2024). An implementation team may also identify an alternative application framework that offers similar functionality based on the complexities imposed by the institutional knowledge base, hosting policy or procurement guideline/frameworks, or requirements for integration with existing systems. In addition, the implementation of the authentication/authenticator facility must align with the NIST SP 800-63B v 5 (as well as to the password hashing portions of the RFC 9106, Argon2id user passwords should be hashed (Biryukov et al., 2021; Temoshok et al., 2025)).

Requirement-to-Design Traceability

The main outcome from performing a requirements analysis is mapping both paper-based risks and proposed design responses. This mapping shows that the design covers more than just document scanning; it also covers classification, integrity, accountability, confidentiality and continuity.

Table 3 Requirements-to-Design Traceability Matrix

Identified concern	Derived requirement	Proposed design response
Slow or inconsistent retrieval	Searchable metadata and indexed identifiers	Student, record, and document entities; search interface; query-processing component
Misfiled or incomplete records	Required fields and controlled classification	Metadata validation, document-type table, status fields, and form validation
Duplicate records or files	Duplicate detection before acceptance	Unique identifiers and a duplicate-record review step
Unauthorized access	Authentication and least-privilege authorization	Role entity, permission matrix, application policies, and restricted object access
Undetected alteration	File-integrity verification	Checksum field and verification procedure
No activity history	Traceable user actions	Audit-log entity and event-logging protocol
Physical loss or deterioration	Digital storage with controlled copies	Private object storage and structured metadata references
Lack of recovery procedure	Scheduled backup, status monitoring, and restoration plan	Backup record, versioning, off-site copy, and recovery workflow

Proposed Data Flow

The flow of information logically is shown in Figure 1. The Registrar staff provide credentials and record information to process the authentication and record-management stages. The created digital documents and their associated metadata enter the digitization and classification stages before being assigned a file reference to store on private cloud storage. Criteria for searching are submitted to the proposed search process, while the administrator manages users, roles, and permissions. Audit-logs for user activity are stored in the audit-log database. Scheduled backup instructions will connect the file-storage component to a separate backup repository

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

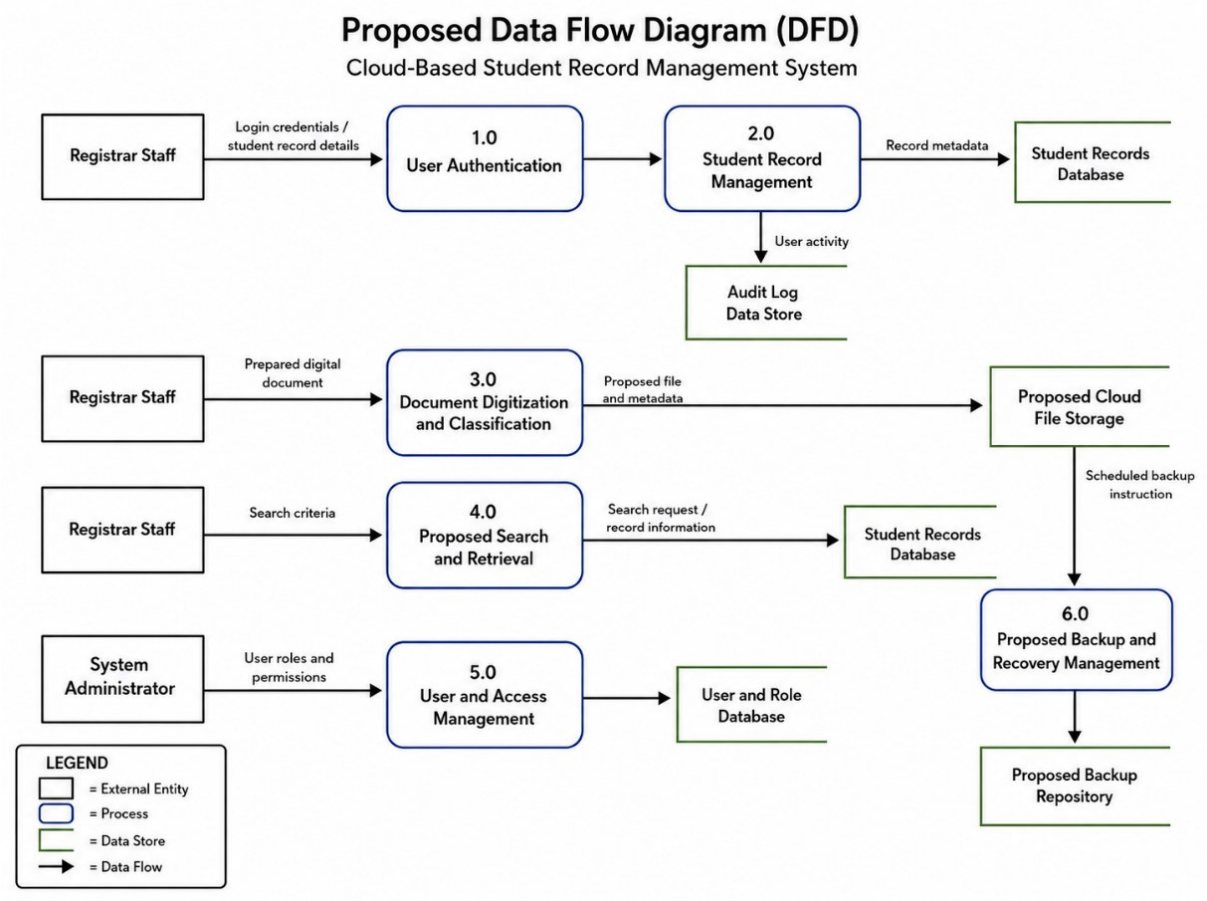

Figure 1. Proposed Data Flow Diagram for the Cloud-Based Student-Record Management Design.

The DFD will differentiate the student record database from document file storage. This separation prevents the database from being treated as a general file repository, while maintaining the association of each document with its metadata. The DFD also makes it possible for accessing decisions to be made on the basis of database records before the application issues a temporary authorized file-access request.

Proposed Database Design

The proposed Entity Relationship Diagram is shown in Figure 2. The Role and User entities are elements of a role-based authorization system. The student entity will maintain the institutional identity and status for all learners, while the Student_Record entity stores information relating to the academic period of all learners. The Document_Type entity has an established and controlled list of documents for classification purposes. The Document_File entity creates a link between the prepared digital file and the student, uploader, document type, storage path, checksum, and date recorded. The Document_Request entity maintains a record of completed formal requests submitted by users; the Audit_Log entity maintains a record of user activity, and the Backup_Record entity documents the planned state of each file at the last planned backup.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

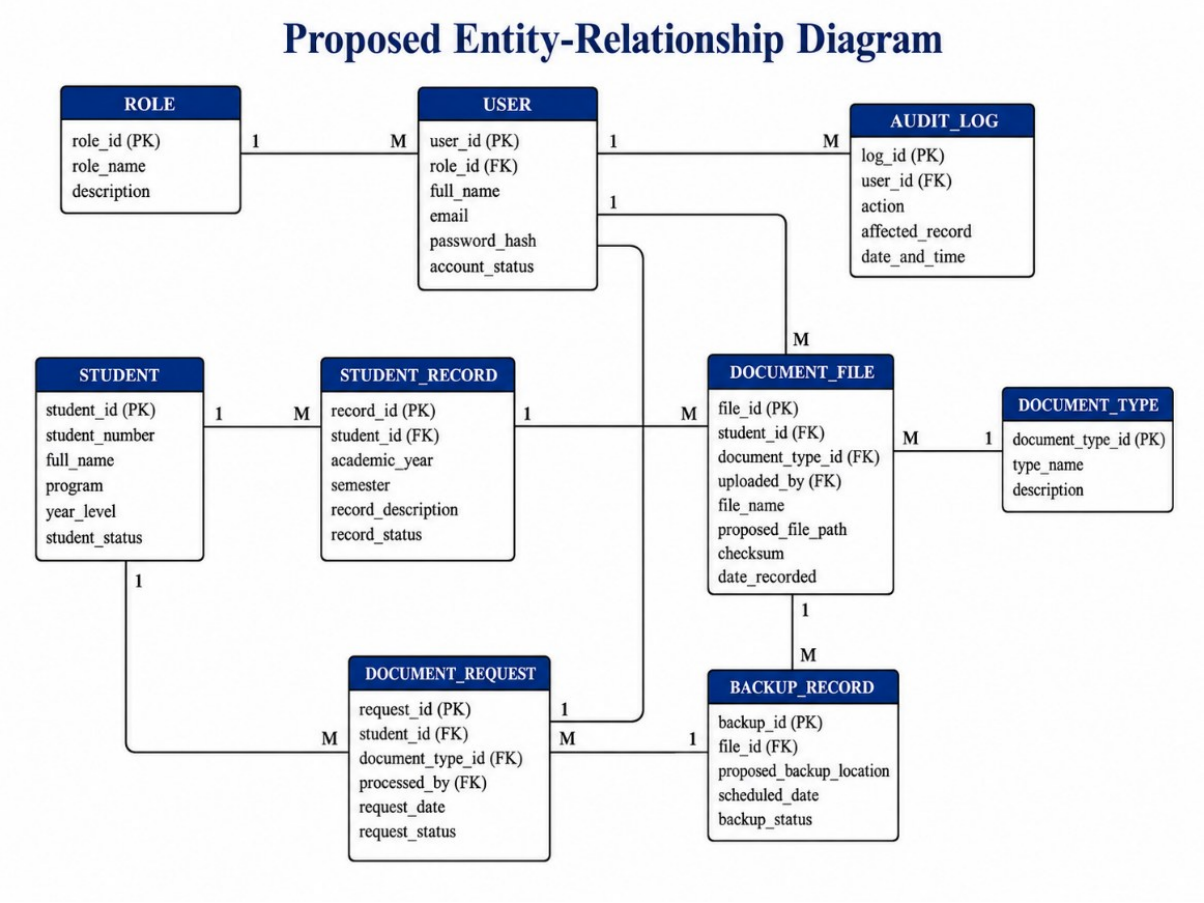

Figure 2. Proposed Entity-Relationship Diagram

Table 4 Purpose of the Core Database Entities

Entity	Primary purpose	Important controls
ROLE	Defines organizational access categories.	Unique role name and documented permissions.
USER	Stores authorized account information.	Unique email, password hash, account status, and role foreign key.
STUDENT	Stores the principal student identifier and profile fields required for record organization.	Unique student number and controlled student status.
STUDENT_RECORD	Groups record information by student and academic period.	Foreign key to student and controlled record status.
DOCUMENT_TYPE	Maintains standardized document classifications.	Unique type name and description.
DOCUMENT_FILE	Stores document metadata and the private object-storage reference.	Checksum, uploader, document type, file path, and date recorded.
DOCUMENT_REQUEST	Tracks requests for document access or release.	Request status, processor, student, document type, and date.
AUDIT_LOG	Preserves accountability for important actions.	User, action, affected record, timestamp, outcome, and source information.
BACKUP_RECORD	Documents backup instructions and status.	File reference, schedule, location, status, and verification date.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

To maintain referential integrity, the design includes the use of primary and foreign keys. Other implementation constraints should include: uniqueness of student numbers, user email addresses, document type names, and approved file identifiers. Indexes should be created on frequently searched fields such as student number, student name, academic year, document type, request status, and date recorded. The checksum should not be the sole identifier of a duplicate file because it is possible for two files to have exactly the same content, so it should be used in conjunction with the student identifier, document type, academic period, and authorized review.

Proposed Three-Layer Architecture

The proposed system is separated into three distinct layers: presentation, application, and data. The presentation layer consists of all the screens that allow authorized users to enter, find, and see information. The application layer is responsible for user authentication, as well as authorizing and validating access to resources; checking for duplicate data; performing search queries; generating checksums; creating audit logs; and scheduling backups. The data layer contains the relational database, object storage, audit data, and the backup repositories. A security/access-control layer is across the three layers.

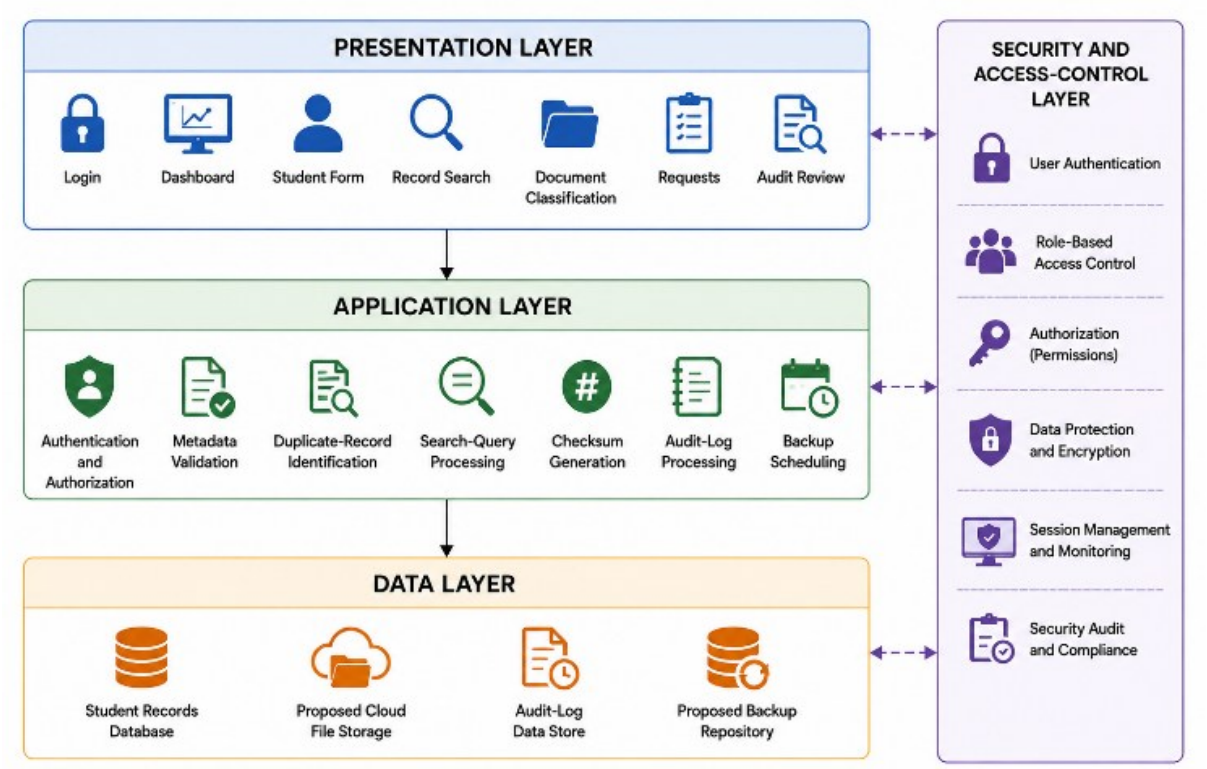

Figure 3. Proposed Three-Layer Cloud Architecture with Cross-Cutting Security and Access Control

A layered architecture limits direct resource access and simplifies data management. User interactions all occur through the presentation layer; business rules are enforced through the application layer; and the data layer only receives requests from approved applications. As a result, future changes to the system can occur easily because any change to an interface will not alter the underlying structure of the database; storage-only changes will be handled separately from the application's storage abstraction.

Proposed User Interface Wireframes

In Figure 4 there are low-fidelity wireframes for the proposed login, dashboard, record entry, search, request and audit review functions. The login function allows limited access to only those who are authorized. The main dashboard displays a summary of records, requests and reviews with no disclosure of any non-essential sensitive information. The record entry screen separates student identifiers from any document metadata as well as ensuring that any item may not be stored until it has passed through a validation process. The search function provides access to an Audit Trail and Document Request Work-Steps, using controlled filters, to provide ability to locate specific items.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

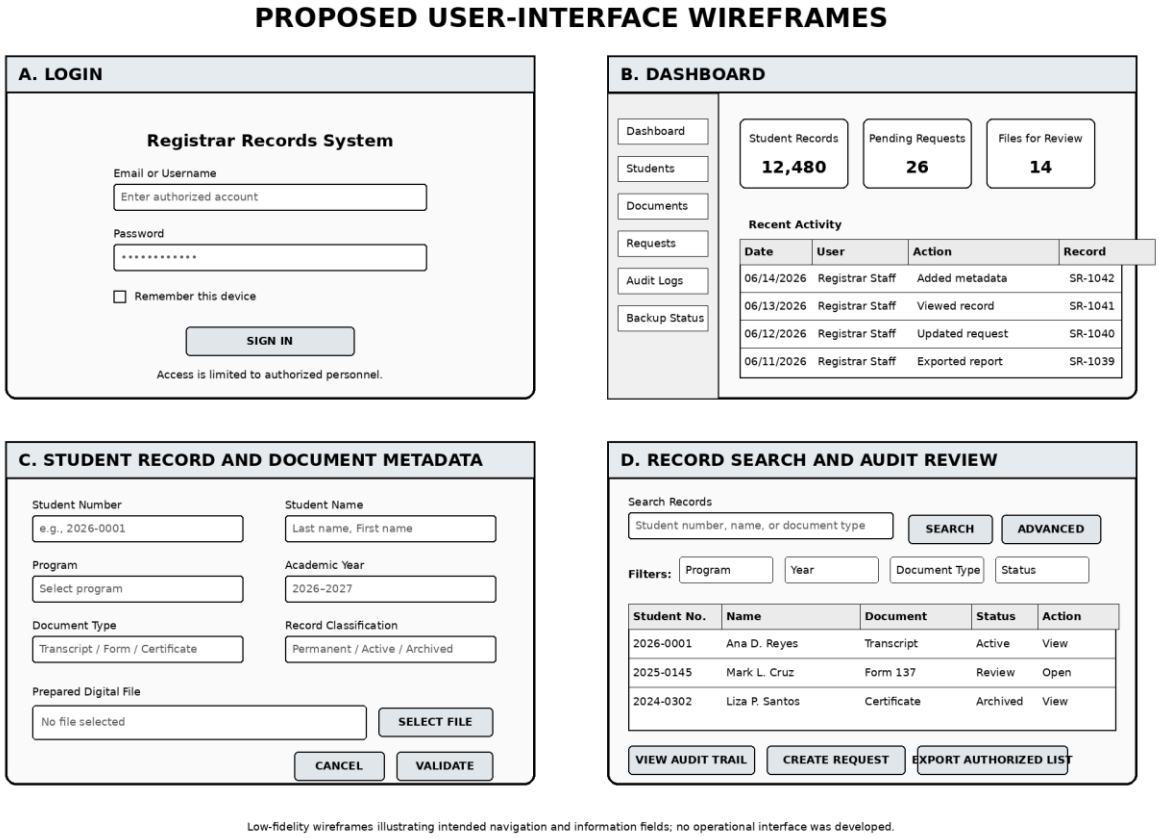

Figure 4. Proposed low-fidelity user-interface wireframes.

Wireframes utilize minimum-required information. The required ID means staff can validate an individual record by the search results, but the contents will only be displayed once the web application checks authorization and purpose. Errors must provide the necessary corrective action but not expose any information about existence of an account, its physical storage location or any internal security information.

Role and Permission Design

The role model being proposed employs a principle called "least privilege access". Responsibilities are defined by the role, but the assignments of permission must occur at the action level (not via a single unrestricted admin account). Below is a minimum intended separation of responsibilities using a matrix.

Table 5 Proposed Role and Permission Matrix

System action	Registrar staff	Registrar supervisor	System administrator	Auditor/authorized reviewer
Create and update student metadata	Allowed	Allowed	Not normally required	Read only
Add document metadata and prepared file	Allowed	Allowed	Not normally required	Read only
Approve duplicate-review decision	Submit for review	Allowed	Not allowed	Read only
Search and view authorized records	Allowed within assigned scope	Allowed	Only for support with approved purpose	Read only within review scope
Process document requests	Allowed	Allowed	Not allowed	Read only
Manage users and roles	Not allowed	Request only	Allowed	Not allowed
Review audit logs	Own/relevant activity	Allowed	Technical events	Allowed
Configure backup jobs	Not allowed	Monitor status	Allowed	Review status

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

System action	Registrar staff	Registrar supervisor	System administrator	Auditor/authorized reviewer
Authorize restoration	Request	Business approval	Technical execution	Observe and review

Security, Integrity, and Continuity Protocols

The design around security is an implementation of preventative, detective, and recovery or contingency controls. The preventative controls include; authentication, multi-factor authentication for privileged roles, role-based access controls, input validation, encryption, and private object storage. The detective controls include; audit logging, monitoring of failed access, checksum verification, and backup status monitoring. The recovery controls include; versioned copies, different backup locations, authority to restore, and recovering from backup during subsequent implementations. All of these controls are aligned to the NIST SP 800-53 Rev. 5 control families and the ISO/IEC 27001:2022 (risk based) Information Security Approach (International Organization for Standardization, 2022; Joint Task Force, 2020). Current ransomware-resilience guidance also recommends maintaining protected backups and regularly testing recovery procedures (Cybersecurity and Infrastructure Security Agency, 2023).

Table 6 Proposed Security and Continuity Protocols

Control area	Proposed protocol	Design purpose
Authentication	Individual institutional accounts; multi-factor authentication for supervisors and administrators; session timeout; account lock or rate limiting after repeated failures (Temoshok et al., 2025).	Reduces account sharing and automated credential attacks.
Authorization	Deny by default; role and action policies; least privilege; separate business approval from technical administration.	Limits access to records and actions required by each responsibility.
Password storage	Use an approved adaptive password-hashing algorithm, preferably Argon2id with framework-managed parameters and unique salts (Biryukov et al., 2021).	Prevents storage of plaintext or reversible passwords.
Transport protection	Use HTTPS with a currently supported TLS configuration and verified secure-communication requirements (OWASP Foundation, 2025).	Protects credentials and records in transit.
Data at rest	Use database and object-storage encryption; protect encryption keys separately from application data.	Reduces exposure if storage media or backups are accessed without authorization.
Metadata validation	Require student identifier, document type, classification, academic period, source, and authorized uploader before acceptance.	Improves completeness and consistency.
Duplicate control	Compare identifiers and metadata; use checksum as supporting evidence; route uncertain matches to authorized review.	Reduces duplicate or misclassified records without automatic destructive merging.
Audit logging	Log authentication outcomes, record creation and change, search and view events where appropriate, downloads, request decisions, role changes, backup events, and restoration actions.	Supports accountability and incident investigation.
File integrity	Generate and store a checksum after file preparation; recheck during authorized retrieval, migration, or restoration.	Identifies unintended modification or corruption.
Backup	Automated encrypted database backup; versioned object storage; separate protected copy; retention schedule; monitored backup status; and regularly tested recovery procedures (Cybersecurity and Infrastructure Security Agency, 2023).	Supports continuity after deletion, corruption, or service failure.
Recovery	Document request, approval, restoration, validation, and audit steps; verify record count, metadata links, and checksums after restoration.	Ensures controlled and traceable recovery.
Privacy	Use data minimization, purpose limitation, retention rules, authorized disclosure procedures, and secure disposal consistent with the Data Privacy Act.	Protects student personal and academic information.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

Integrated Record-Digitization and Integrity Model

In Figure 5, we combine all the separate artifacts into one workflow as Proposed. When we send a paper record for digitization, it is first created and then described with the necessary metadata, checked for possible duplicate entries, assigned a checksum, and assigned access rules based on each user's role within the organization. The database will hold both the structured metadata and an object storage reference to the actual paper record, while audit logs will be maintained for any activity that has occurred in the audit logs associated with that paper record. Backups and recovery planning will occur after the storage assignment step, and authorized users will use the metadata index versus actually browsing the storage folders directly for searching or retrieving documents.

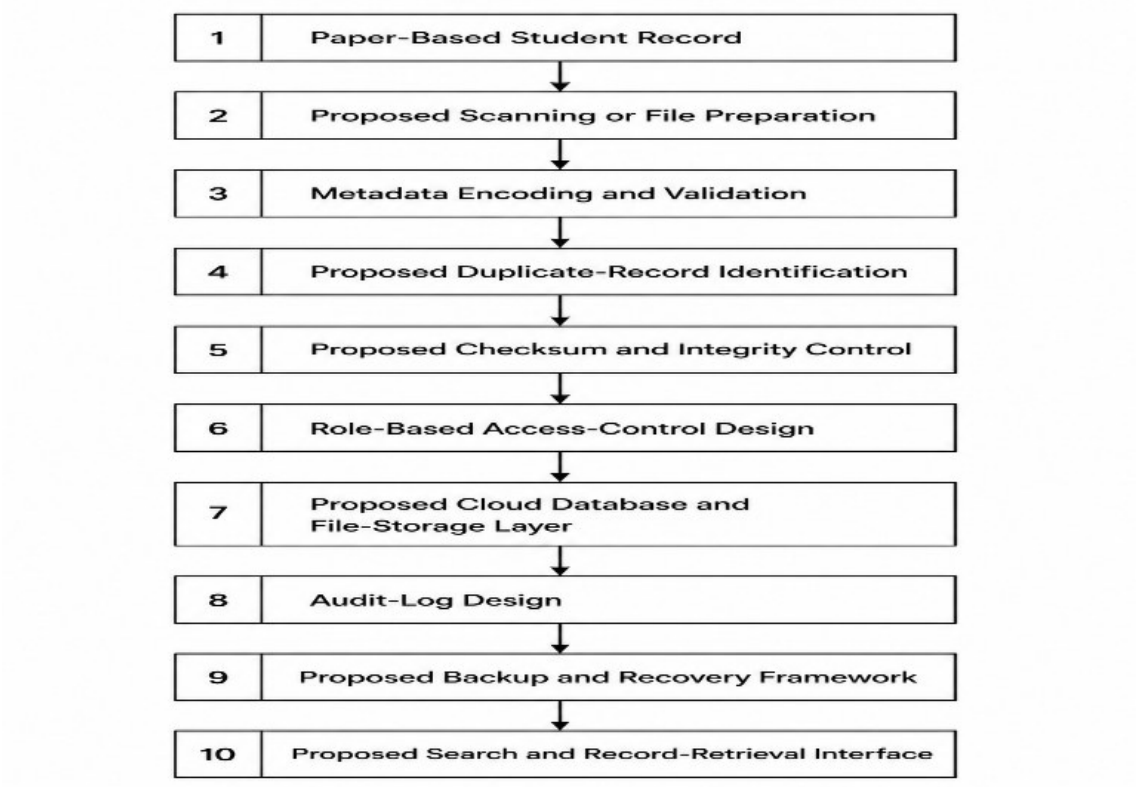

Figure 5. Proposed Secure Cloud-Based Record Digitization and Integrity Design Model

The model contributes to the integration of functions that are commonly discussed separately. Scanning does not by itself ensure completeness, confidentiality, integrity, or recoverability. The model therefore positions validation, duplicate review, integrity checks, access controls, auditability, and continuity as integral components of the same record life cycle.

Proposed Pseudocode

The following pseudocode expresses the intended logical sequence without prescribing a programming language or claiming implementation:

```
PROCEDURE PROPOSED_RECORD_DIGITIZATION
INPUT: authorized_user, student_metadata, document_metadata, prepared_file
OUTPUT: proposed_record_reference, audit_event, backup_instruction

1. Authenticate authorized_user.
2. If authentication fails, record the failed attempt and stop.
3. Determine the user role and requested action.
4. If the role is not permitted, record the denied action and stop.
5. Validate all required student and document metadata.
6. If required metadata is missing or invalid, return the validation errors and stop.
7. Search for possible duplicates using student identifiers, document type, period, and file checksum.
8. If a possible duplicate is found, mark the submission for authorized review.
9. Generate the approved cryptographic checksum for the prepared file.
10. Assign the relational record identifier and private object-storage key.
11. Define access rules according to the approved role and record classification.
12. Prepare the audit event containing user, action, record, timestamp, and outcome.
13. Create the backup instruction according to the retention and versioning policy.
14. Return the proposed references and status for future implementation.
END PROCEDURE
```


iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

Scope of the Results

Results show that this design is complete but not operationally effective. The results of this project provide a logical set of specifications and a description of how all different parts will work together. However, because there was no production-ready system or working application available to conduct these tests, there are no measures for response time, storage performance, usability rating, resistance to cyberattack, backup success rate, or recovery time. These items will be covered in the next phase of the project to implement and evaluate the system.

CONCLUSION

An analytical overview was produced in conjunction with the proposed system's design. An overall system design specification began with digitization of the Registrar's office student record system. The specific designs that were developed correspond to standard paper-based recordkeeping concerns and convert to related technical definitions that can be used for the future development of a student record system (e.g., requirements, data flow diagrams, entity relationship diagram, relational database schema, three-tier cloud architecture, user interface wireframe, role and permission model, security architecture, backup and restore methodology, integrated digitization model and pseudocode). In the Technology Assessment documentation there were four (4) primary candidates for technology architecture proposed for future student record system development, to include: server-side web framework, PostgreSQL and S3 compatible object storage or equivalent. The proposed system design divides all documents/records into metadata records and corresponding document records and implements assurance measures including: validation, duplicate checking, checksum-based integrity checking, least privilege access, logging of all audit trail events, encrypted storage/secure transmission, auditing/reviewing of systems backup, auditing/reviewing of versions of records, and controlling all records restorations. The primary contribution of the study is to provide a context-specific technical design specification which is a blueprint and not an operational system. No software was developed or tested, nor were any actual student records processed. The technical design must be utilized for implementation plan purposes, as an institutional review document, privacy policy review, security tests, usability evaluations, and controlled pilot deployments.

RECOMMENDATIONS

1) A future development study should convert the approved design into a prototype using dummy or anonymized records before any production data are introduced.
2) The institution should conduct a privacy impact assessment, define record-retention rules, and obtain formal approval for roles, permissions, disclosure procedures, and backup locations.
3) The implemented system should undergo functional, integration, usability, accessibility, performance, vulnerability, backup, and restoration testing.
4) A controlled pilot should be conducted with trained Registrar's Office personnel, with documented acceptance criteria and rollback procedures.
5) The database schema and security protocols should be reviewed whenever institutional policies, privacy requirements, document classifications, or technology platforms change.

REFERENCES

1) Aleryani, A. Y. (2024). Analyzing data flow: A comparison between Data Flow Diagrams (DFD) and User Case Diagrams (UCD) in information systems development. *European Modern Studies Journal, 8*(1), 313–320. https://doi.org/10.59573/EMSJ.8(1).2024.28
2) Aliazas, J. V., Cruz, R. Dela, & Ilagan, N. (2024). Enhancing university operations: A study of the Electronic Document Management Systems (EDMS) of one higher education institution. *TWIST, 19*(3), 229–237. https://doi.org/10.5281/zenodo.10049652
3) Amazon Web Services. (2024). *Security pillar—AWS Well-Architected Framework*. https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html
4) Bagui, S. S., & Earp, R. W. (2022). *Database design using entity-relationship diagrams*. https://doi.org/10.1201/9781003314455
5) Biryukov, A., Dinu, D., Khovratovich, D., & Josefsson, S. (2021). *Argon2 memory-hard function for password hashing and proof-of-work applications* (RFC 9106). RFC Editor. https://doi.org/10.17487/RFC9106
6) Bootstrap Team. (2024, February 20). Bootstrap 5.3.3. *The Bootstrap Blog*. https://blog.getbootstrap.com/2024/02/20/bootstrap-5-3-3/

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

7) Cybersecurity and Infrastructure Security Agency. (2023). *#StopRansomware guide*. CISA. https://www.cisa.gov/resources-tools/resources/stopransomware-guide
8) Duruin, R. A., & Siddayao, G. P. (2024). Development of student records management system of Magalalag National High School. *AIDE Interdisciplinary Research Journal, 8*, 84–94. https://doi.org/10.56648/AIDE-IRJ.V8I1.112
9) Engström, E., Storey, M. A., Runeson, P., Höst, M., & Baldassarre, M. T. (2020). How software engineering research aligns with design science: A review. *Empirical Software Engineering, 25*(4), 2630–2660. https://doi.org/10.1007/S10664-020-09818-7
10) Falolo, V. M., Capillas, K. T., Vergarra, N. A., & Cerbito, A. F. (2022). Student registration and records management services towards digitization. *International Journal of Educational Management and Development Studies, 3*(1). https://doi.org/10.53378/352867
11) IEEE Computer Society. (2024). *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), version 4.0*. IEEE Computer Society. https://www.computer.org/education/bodies-of-knowledge/software-engineering
12) International Organization for Standardization. (2022). *ISO/IEC 27001:2022: Information security, cybersecurity and privacy protection—Information security management systems—Requirements*. https://www.iso.org/standard/27001
13) International Organization for Standardization. (2023). *ISO/IEC 25010:2023—Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model*. https://www.iso.org/standard/78176.html
14) Joint Task Force. (2020). *Security and privacy controls for information systems and organizations* (NIST Special Publication 800-53, Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5
15) Laravel. (2024). *Release notes—Laravel 11.x: The clean stack for Artisans and agents*. https://laravel.com/docs/11.x/releases
16) National Institute of Standards and Technology. (2024). *The NIST Cybersecurity Framework (CSF) 2.0* (NIST CSWP 29). U.S. Department of Commerce. https://doi.org/10.6028/NIST.CSWP.29
17) National Privacy Commission. (2020). *Data Privacy Council Education Sector Advisory No. 2020-1: Data privacy and online learning*. https://privacy.gov.ph/wp-content/uploads/2023/05/DP-Council-Education-Sector-Advisory-No.-2020-1.pdf
18) National Privacy Commission. (2023). *Security of personal data in the government and the private sector*. https://privacy.gov.ph/wp-content/uploads/2024/03/NPC-Circular-Repeal-16-01-Signed.pdf
19) OWASP Foundation. (2025). *OWASP Application Security Verification Standard (ASVS)*. https://owasp.org/www-project-application-security-verification-standard/
20) PostgreSQL Global Development Group. (2025, September 25). *PostgreSQL 18 released!*. https://www.postgresql.org/about/news/postgresql-18-released-3142/
21) Temoshok, D., Fenton, J. L., Choong, Y.-Y., Lefkovitz, N., Regenscheid, A., Galluzzo, R., & Richer, J. P. (2025). *Digital identity guidelines: Authentication and authenticator management* (NIST Special Publication 800-63B-4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63B-4
22) vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to design science research. https://doi.org/10.1007/978-3-030-46781-4_1

