CAPSTONE PROJECT

Paulinian electronic Archiving System (PeAS)

Prepared by:

Christian James B. Anadon

James Dale A. Grafe

Kurt Dustine L. Yrad

Advisor:

Ms. Grace L. Alejado, MIT

St. Paul University Dumaguete

Bachelor of Science in Information Technology

Table of Contents

Summary

PeAS is a digital repository designed to archive SPUD's academic outputs – including theses, dissertations, confluences, and synergies – while offering tiered access for public users, registered members, and administrators. It integrates document digitization, metadata management, and secure authentication via SPUD's existing OrangeApp system.

Problem Statement

SPUD currently lacks a unified platform to store and manage academic works, leading to:

- **Fragmented access**: Difficulty retrieving physical or digital documents.
- **Preservation risks**: Vulnerability of physical copies to damage or loss.
- **Limited user functionality**: No distinction between public access and registered user privileges (e.g., downloading, bookmarking).

Objectives

1. **Centralized repository**: Securely organize academic documents by type, year, and metadata.
2. **Digitization workflow**: Use departmental scanners to import legacy and new documents.
3. **Access differentiation**:
    - **Public users**: View documents and metadata without downloading.
    - **Registered users**: Download, bookmark, and track usage history.

4. **Integration with SPUD's authentication system**: Leverage OrangeApp for secure login.
5. **Enhanced searchability**: Implement filters for document type, author, and year.

Methods

The project follows a structured development lifecycle:

1. **Requirements analysis**: Client consultations to define roles (public, registered, admin) and system specifications.
2. **System design**:
   - **Frontend**: HTML/CSS and Javascript for responsive interfaces.
   - **Backend**: Deno for server-side logic.
   - **Database**: MySQL for storing user data, metadata, and logs.
3. **Integration**:
   - Authentication via OrangeApp's API.
   - Elasticsearch/Apache Solr for document search.
4. **Testing**:
   - Unit and integration tests for security (encryption, access controls).
   - User acceptance testing for interface usability.
5. **Deployment**: Hosting on SPUD servers with regular backups.

Key Findings

1. **Technical implementation**:
   - **User roles**: Three-tier access (public, registered, admin) with activity logs for admins.
   - **Metadata standardization**: Author, title, year, and document type enable efficient search.
2. **Impact**:
   - Preserves SPUD’s academic legacy through digitization.
   - Facilitates research by improving document accessibility.
3. **Scalability**: Modular architecture allows future expansions, such as additional document types.

Background of the Study

Overview of the Paulinian electronic Archiving System (PeAS)

The Paulinian electronic Archiving System (PeAS) is a digital repository developed for St. Paul University Dumaguete (SPUD) aimed at centralizing the storage and retrieval of academic outputs—including theses, dissertations, and scholarly publications. By moving from traditional paper-based records to an electronic archive, PeAS is designed to mitigate challenges such as physical document degradation, limited storage space, and labor-intensive retrieval procedures. Although specific public documentation on PeAS is relatively scarce, institutional reports and case studies on similar university-based digital archiving systems provide a framework for understanding its significance. These sources emphasize that modern electronic archives must integrate secure authentication, role-based access control, and robust metadata management to ensure long-term preservation and efficient retrieval of academic records (Conway, 2011; Gojgić, Nikolić, & Petrović, 2013).

The Role of Archiving Systems in Educational Institutions

Educational institutions generate extensive documentation ranging from student records and administrative files to research outputs and curricular materials. Digital archiving systems play an essential role in preserving this information by offering:

- **Enhanced Accessibility:** Advanced search functionalities and metadata indexing ensure that stored documents are easily retrievable, supporting both administrative operations and scholarly research.
- **Improved Preservation:** Transitioning to digital archives minimizes the risks associated with physical storage, such as damage from natural disasters and degradation over time.
- **Regulatory Compliance:** Digital systems help institutions meet legal and regulatory requirements by ensuring secure and accurate retention of records (Conway, 2011).

Studies in the field of digital archiving illustrate that a well-structured electronic repository not only streamlines information management but also enhances the transparency and accountability of academic institutions (Gojgić et al., 2013). By aligning with best practices in digital preservation, systems like PeAS ensure that academic records remain both accessible and reliable over the long term.

Rationale and Objectives

The transition from manual, paper-based records to a centralized digital system such as PeAS is expected to enhance the efficiency of document storage, retrieval, and monitoring at SPUD. By adopting a digital archiving solution that incorporates secure authentication, comprehensive metadata, and role-based access, the university aims to:

1. Reduce operational costs associated with physical storage and maintenance.
2. Improve the speed and accuracy of document retrieval, thus supporting both administrative functions and academic research.
3. Ensure long-term preservation of institutional memory, facilitating quality assurance and compliance with educational standards.

This study will investigate the implementation of PeAS within the framework of best practices in digital archiving, drawing on established literature and case studies from comparable academic institutions.

Problem Statement

Challenges in Document Archiving and Retrieval

Educational institutions continuously generate a vast array of documents—from academic outputs and administrative files to research records—that must be reliably preserved for future reference and accountability. However, many institutions still rely on traditional, paper-based archiving methods that face significant challenges. First, physical records are vulnerable to degradation from environmental factors (e.g., typhoons, humidity, and pest infestation), which not only risks the loss of important data but also complicates the retrieval process when records become damaged or misclassified (Caluza, 2019). Additionally, manual filing and retrieval are labor-intensive and time-consuming, resulting in delays in accessing documents for decision-making or legal verification. These challenges are compounded by inadequate indexing and metadata assignment, which hinders efficient document search and retrieval (Gojgić, Nikolić, & Petrović, 2013).

Limitations of Existing Systems

Existing archiving systems in many educational settings often fail to address the scalability and security needs of modern institutions. Traditional systems generally lack automated workflows and digital metadata integration, which are essential for rapid and accurate document retrieval. The absence of robust security features and role-based access controls further exposes sensitive academic and administrative records to risks such as unauthorized access and misclassification. Moreover, outdated manual processes tend to be cost-intensive due to the need for physical storage space and continuous manual maintenance, ultimately reducing overall operational efficiency (Conway, 2011). These limitations underscore the urgent need for a centralized, digital archiving solution that can not only preserve documents in a secure and accessible manner but also streamline retrieval processes to support effective institutional management.

Below is a sample “Objectives” section for the project, including one general objective and several specific objectives. In this version, additional scholarly and industry sources are incorporated to support the need for efficient, secure, and user-friendly digital archiving systems in higher education.

Objectives

General Objective

To develop an efficient electronic archiving system for St. Paul University Dumaguete that centralizes and streamlines the storage, retrieval, and preservation of academic and administrative records, ensuring long-term sustainability, enhanced accessibility, and robust security.

Specific Objectives

1. **Design a Robust Database Architecture:**
   Develop a scalable database model that organizes academic and administrative records with comprehensive metadata. This will facilitate efficient categorization and retrieval, in line with established information models for electronic archiving (Gojgić, Nikolić, & Petrović, 2013; Marsh, Wackerman, & Stubbs, 2017).
2. **Implement a Lightweight and Scalable Backend:**
   Utilize modern technologies (e.g., Deno) to build a backend that is both lightweight and scalable. This system will integrate with existing university infrastructures and support high-volume transactions while maintaining performance and reliability (Conway, 2011).
3. **Integrate an Efficient Search Mechanism:**
   Incorporate a high-performance search engine (e.g., Meilisearch) to enable rapid full-text indexing and metadata-based searches. Enhanced search functionality will improve document discoverability and user experience,

addressing common retrieval challenges (Digital Preservation Coalition, 2019).

4. **Establish Secure, Role-Based Access Controls:**
   Develop robust authentication and authorization protocols that ensure sensitive records are accessed only by authorized personnel. This objective addresses security vulnerabilities and compliance requirements critical to educational institutions (Armstrong Archives, 2018; Access Corporation, 2025).
5. **Develop an Intuitive, User-Friendly Interface:**
   Create a web-based interface that accommodates the diverse needs of administrators, faculty, and students. A focus on usability will reduce training time and facilitate self-service document retrieval, thereby enhancing overall operational efficiency (Anderson Archival, 2024).
6. **Conduct Comprehensive Testing and Quality Assurance:**
   Execute rigorous functional, performance, and security testing to ensure that the system meets the university's operational needs and regulatory standards. This includes validating the system's reliability for long-term digital preservation (Conway, 2011; Digital Preservation Coalition, 2019).

Below is a "Scope and Delimitations" section for the study on developing an efficient electronic archiving system for St. Paul University Dumaguete. In this section, the features of the system are clearly defined and the study's limitations are outlined. Multiple scholarly and industry sources are cited to support the statements.

Scope and Delimitations

Scope

This study focuses on the design, development, and implementation of an electronic archiving system for St. Paul University Dumaguete (SPUD). The system is intended to serve as a centralized digital repository for academic and administrative records, including theses, dissertations, and other scholarly outputs. Key features included in the system are:

- **Search Functionality:**
  A robust search engine that supports both full-text queries and metadata-based filtering, enabling rapid and precise retrieval of archived documents.
- **Document Categorization:**
  A module that organizes documents by type, year, author, and other metadata fields, facilitating efficient classification and systematic storage of records.
- **User Authentication:**
  Secure, role-based access control mechanisms to ensure that only authorized users (e.g., administrators, faculty, and registered users) can access sensitive documents.
- **Intuitive User Interface:**
  A web-based front-end designed for ease of navigation, allowing users to quickly locate and interact with digital records.
- **Integration with Institutional Systems:**
  The system will interface with existing SPUD databases for consistent data management and to support seamless user authentication.

Delimitations

While the study aims to develop a comprehensive electronic archiving system, it is important to note the following limitations:

- **Exclusion of Physical Document Storage:**
  The project is strictly limited to the digital domain and does not address the physical storage, handling, or preservation of paper-based records. The system focuses solely on creating and maintaining digital surrogates.
- **Scope of Document Types:**
  The system is designed to manage academic and administrative records produced within SPUD. It will not extend to external documents or records from other institutions.
- **Feature Set Limitation:**
  Advanced functionalities such as automated indexing through artificial intelligence, predictive analytics, or integration with external digital libraries are beyond the current scope. Instead, the system will implement standard metadata tagging and categorization methods based on established best practices.
- **Policy and Regulatory Framework:**
  While the system supports compliance by securing records digitally, the study does not cover the development or modification of institutional policies regarding document retention or legal compliance measures for physical archives.
- **Resource Constraints:**
  The design and implementation will focus on essential features to ensure a manageable scope. Budgetary and time constraints necessitate that the system prioritizes core functionalities over ancillary enhancements.

Significance of the Study

The implementation of the Paulinian electronic Archiving System (PeAS) at St. Paul University Dumaguete (SPUD) holds significant promise for transforming academic document management. By transitioning from a fragmented, paper-based approach to a centralized digital repository, PeAS offers a multitude of benefits across different levels of the academic community.

Benefits to Students

- **Enhanced Accessibility and Learning:**
  Students gain immediate, round-the-clock access to a broad spectrum of academic outputs—theses, dissertations, and other scholarly works—through a user-friendly, web-based interface (Access Corporation, 2025). This ease of access not only supports independent learning and research but also encourages deeper engagement with available resources.
- **Improved Navigation and Discovery:**
  The system's robust search functionality—built on standardized metadata (e.g., author, title, year, and document type)—allows students to efficiently locate materials relevant to their studies. This streamlined retrieval process reduces the time spent on finding resources, enabling students to focus more on learning and critical analysis (Marsh, Wackerman, & Stubbs, 2017).

Benefits to Faculty

- **Support for Teaching and Scholarship:**
  Faculty members benefit from having a comprehensive repository of academic outputs that can be integrated into classroom instruction and scholarly activities. The availability of high-quality, digitized materials facilitates curriculum enrichment and enhances the teaching–learning dynamic (Anderson Archival, 2024). Additionally, by integrating best practices in digital preservation (Conway, 2011; Gojgić, Nikolić, & Petrović, 2013), the system ensures that valuable legacy documents remain accessible for pedagogical use.
- **Streamlined Academic Collaboration:**
  With role-based access controls in place, faculty can easily contribute to, annotate, and share academic works with peers. This not only fosters collaborative research but also creates a feedback loop for continuous improvement in academic outputs, reinforcing secure document handling protocols (Armstrong Archives, 2018).

Benefits to Researchers

- **Centralized, Reliable Data Source:**
  Researchers enjoy a single, well-organized platform that houses diverse academic works, which improves the discoverability and integrity of research documents. Enhanced search capabilities, including metadata-based filters, help researchers pinpoint studies and data sets essential for further investigations (Marsh, Wackerman, & Stubbs, 2017).

- **Long-Term Preservation of Scholarship:**
  By digitizing both legacy and new documents, PeAS ensures that important research findings are preserved against risks of physical degradation and loss. This commitment to long-term preservation is in line with best practices in digital archiving as highlighted by Conway (2011), Gojgić, Nikolić, and Petrović (2013), and the Digital Preservation Coalition (2019). Moreover, the preservation of legacy materials is critical to maintaining institutional memory, a need underscored in studies like Caluza (2019).

Benefits to the Institution

- **Centralized Management and Operational Efficiency:**
  For SPUD, the adoption of PeAS consolidates academic and administrative records into a single, secure repository. This consolidation streamlines document management processes and reduces operational costs associated with physical storage and manual retrieval (Access Corporation, 2025). Such efficiency gains support better resource allocation and improved institutional operations.
- **Enhanced Institutional Reputation and Legacy Preservation:**
  A robust digital archiving system reinforces the institution's commitment to preserving its academic legacy. By maintaining a comprehensive, easily accessible archive, SPUD strengthens its reputation as a forward-thinking institution dedicated to academic excellence and research integrity (Caluza, 2019). Furthermore, ensuring that digital archives are accessible and secure builds confidence among stakeholders and enhances the institution's overall profile.

- **Secure Integration with Existing Systems:**
  The integration of PeAS with SPUD's OrangeApp authentication system ensures a secure, tiered access model—differentiating public users, registered members, and administrators. This secure integration supports regulatory compliance and provides a scalable framework for future expansions (Armstrong Archives, 2018).

Below is a revised "Review of Related Literature" section (Section 4.1) that provides a detailed comparison of PeAS with real-world digital archiving systems and outlines concrete claims regarding the challenges in traditional document archiving. Each claim is supported with inline citations referencing real-world examples and scholarly sources.

Existing Archiving Systems

Comparison of PeAS with Similar Systems

Modern academic institutions have embraced digital archiving platforms to overcome the limitations of traditional paper-based storage. Real-world systems such as DSpace, EPrints, and Digital Commons have been widely adopted for their robust metadata management, standardized workflows, and user-friendly interfaces. For instance, the Massachusetts Institute of Technology (MIT) has used DSpace for over two decades to preserve and disseminate its scholarly output, demonstrating the efficacy of a system built on standardized metadata and automated search capabilities (Marsh, Wackerman, & Stubbs, 2017).

PeAS, the Paulinian electronic Archiving System, shares several core functionalities with these established platforms:

- **Document Digitization and Metadata Management:**
  Like DSpace and EPrints, PeAS incorporates automated workflows to digitize both legacy and new academic documents. It standardizes metadata—such as author, title, year, and document type—to facilitate accurate and efficient search. This approach is comparable to that of Digital Commons, which many universities use to manage research publications; however, PeAS extends this by integrating advanced search filters tailored to SPUD's specific academic outputs (Marsh, Wackerman, & Stubbs, 2017).
- **Tiered Access and Secure Authentication:**
  While many institutional repositories provide basic user role management, PeAS distinguishes itself through a sophisticated tiered access model integrated with SPUD's existing OrangeApp authentication system. This integration ensures that public users, registered members, and

administrators have appropriate access privileges. Similar systems (e.g., Hydra/Samvera) offer role-based controls, but PeAS's seamless connection with a campus-wide authentication platform enhances security and user accountability—an essential requirement for handling sensitive academic records (Armstrong Archives, 2018).

- **Modular and Scalable Architecture:**
  PeAS is designed with a modular architecture that allows for easy integration of new document types and search engines (such as Elasticsearch or Apache Solr). This scalability is a concrete advantage over many legacy systems that may be constrained by older, monolithic architectures. By comparison, systems like EPrints, used by the University of Southampton, have evolved to meet expanding needs; PeAS builds on this evolution by offering an architecture that is both adaptable and cost-effective (Gojgić, Nikolić, & Petrović, 2013).
- **Customized User Interface and Institutional Branding:**
  Real-world platforms, such as Digital Commons, provide interfaces optimized for scholarly communication. PeAS similarly emphasizes ease of use and integrates SPUD's institutional branding, creating a customized user experience that caters specifically to the diverse needs of students, faculty, and researchers (Anderson Archival, 2024).

Challenges in Traditional Document Archiving

Traditional archiving systems, which rely on physical document storage and manual retrieval processes, present several concrete challenges:

- **Fragmented and Labor-Intensive Processes:**
  Conventional paper-based archives require significant manual labor to file, index, and retrieve documents. For example, at The Lab School in Washington, D.C., manual retrieval processes once resulted in transcript requests taking up to eight or nine business days due to the need to physically locate and handle paper files (as reported in case studies such as those discussed by Caluza, 2019). This fragmentation leads to inefficiencies that digital systems, like PeAS, can overcome by automating indexing and retrieval.
- **Vulnerability to Environmental Hazards:**
  Physical documents are inherently at risk from environmental factors—such as fire, water damage, humidity, and pest infestation—that can cause irreversible degradation. Conway (2011) outlines how such risks not only threaten document integrity but also increase the need for costly preservation measures. In contrast, digital systems provide redundant, secure backups and controlled storage environments that minimize these risks.
- **Limited Accessibility and Searchability:**
  Traditional archives are geographically bound, meaning that access is restricted to those physically present at the storage site. This limitation severely hampers timely retrieval and academic research. For example, institutions that still rely on paper-based systems often report that users must sift through large volumes of files manually, leading to delays and frustration (Marsh, Wackerman, & Stubbs, 2017). Digital repositories, on the other hand, enable keyword-based search and rapid retrieval, significantly improving accessibility.

- **High Operational and Maintenance Costs:**
  Maintaining physical archives requires continuous investment in storage space, security, and manual labor. Institutions face rising costs not only for physical storage but also for periodic restoration and preservation efforts. The financial burden associated with traditional archiving has been a major driver for institutions transitioning to digital systems that offer long-term cost savings through automated processes and reduced physical infrastructure (Digital Preservation Coalition, 2019).
- **Risk of Inconsistencies and Data Loss:**
  Without standardized filing systems, traditional archiving can lead to misfiled documents, inconsistencies in record-keeping, and higher risks of data loss. Real-world examples from universities demonstrate that paper archives often suffer from disorganization and incomplete records, making it difficult to guarantee long-term accessibility (Caluza, 2019). Digital systems like PeAS mitigate these issues by enforcing consistent metadata standards and implementing robust backup protocols.

The cumulative challenges of traditional document archiving have concrete implications in the real world—ranging from delayed service delivery to elevated costs and compromised data integrity. These factors underscore the urgent need for digital archiving solutions like PeAS that not only address these problems but also leverage the benefits observed in established digital systems (Conway, 2011; Gojgić, Nikolić, & Petrović, 2013).

Below is a detailed “Technologies Used” section for PeAS. This section explains the choices for each component of the system and provides concrete, real-world comparisons—with proper sources—to justify the decisions.

Technologies Used

Frontend Development

HTML, CSS, and JavaScript form the backbone of the PeAS user interface.

- **HTML** is used to structure the web pages and content, while **CSS** handles styling and responsiveness.
- **JavaScript** adds interactivity, enabling dynamic content updates without reloading the page.

These technologies are industry standards with extensive documentation and support—for example, the Mozilla Developer Network (MDN) offers comprehensive resources on HTML, CSS, and JavaScript (MDN Web Docs, n.d.).

Backend Development: Deno vs. PHP

PeAS adopts **Deno** as its backend runtime instead of PHP. The decision is based on several concrete, real-world considerations:

- **Modern Architecture and Security:**
  Deno is a secure, modern JavaScript/TypeScript runtime built on the V8 engine and Rust. It provides a secure-by-default model (i.e., it requires explicit permission flags to access system resources) and supports TypeScript out of the box. These features result in a cleaner, more maintainable codebase compared to PHP's traditional, synchronous execution model. For instance, a comparison on Webmaster Tips highlights that Deno's design emphasizes security and modern development practices over PHP's legacy model (WMTips, n.d.).

- **Performance and Concurrency:**
  Unlike PHP—historically designed for blocking, process-per-request execution—Deno leverages non-blocking I/O and an event-driven model similar to Node.js. This modern design improves performance under concurrent loads, a critical factor for dynamic repositories like PeAS. As noted in benchmarks discussed on Stack Overflow, Deno can avoid some of the startup overhead (by caching compiled TypeScript) and deliver faster response times compared to legacy PHP setups (Stack Overflow, 2020).
- **Developer Experience and Ecosystem:**
  While PHP powers a large portion of the web (with reported market shares exceeding 80% in some analyses; WMTips, n.d.), its ecosystem is built on decades of practices that can be less flexible in today's fast-changing web environment. In contrast, Deno's native support for TypeScript and a simplified module system (importing modules directly via URLs) results in a more modern and enjoyable developer experience.

Because PeAS is a new initiative focused on modern academic document management, the benefits of Deno—its security model, non-blocking I/O, and TypeScript integration—make it the preferred choice for backend development over PHP.

Search Engine Integration: Meilisearch vs. Elasticsearch

For full-text search functionality, PeAS integrates **Meilisearch** rather than Elasticsearch. Key concrete points include:

- **Ease of Use and Out-of-the-Box Performance:**
  Meilisearch is designed to work with minimal configuration and provides

near–instant search results. In direct comparisons, real-world tests have shown that Meilisearch can be significantly more resource-efficient and faster to return search results than Elasticsearch (Be, 2023; Peterbe, 2023). This is crucial for academic repositories where quick, typo-tolerant, and relevant search responses enhance the user experience.

- **Lower Resource Consumption:**
  Meilisearch's implementation in Rust results in a small memory footprint. For example, several benchmarks and user reports indicate that Meilisearch uses a fraction of the memory required by Elasticsearch, which can be a key advantage when deploying on limited infrastructure (Meilisearch Blog, 2023).
- **Tailored for End-User Search:**
  While Elasticsearch is a powerful, scalable search engine capable of complex aggregations and analytics, its flexibility comes at the cost of extensive configuration. Meilisearch focuses on delivering an optimized instant search experience for end-users—a feature that fits well with PeAS's goals.

Thus, for PeAS, Meilisearch is chosen as the search engine integration because it meets the requirements for speed, simplicity, and low resource usage, which are essential for a responsive academic repository.

Database Management: MySQL vs. PostgreSQL

PeAS currently uses **MySQL** for managing relational data (such as user records, document metadata, and logs) over PostgreSQL for the initial deployment. Concrete reasons for this choice include:

- **Widespread Adoption and Proven Stability:**
  MySQL is one of the most popular open-source relational database

management systems and has been widely adopted in production environments around the globe. Its robust performance and extensive support make it a safe, proven choice (MySQL, n.d.).

- **Ease of Integration:**
  Given that many legacy systems and web hosting environments are optimized for MySQL, its integration with existing infrastructure is straightforward. In contrast, while PostgreSQL is renowned for its advanced features and strict SQL compliance, the familiarity and established nature of MySQL in many web applications favor its use in the early stages of deployment (DigitalOcean, n.d.).
- **Community and Ecosystem:**
  MySQL's mature ecosystem—with numerous tools, libraries, and support communities—ensures that developers can find ample resources and troubleshooting help, which is critical during the initial rollout of a complex system like PeAS.

For now, MySQL is chosen for its ease of use, compatibility with existing systems, and strong community support. Future phases of PeAS could consider migrating to PostgreSQL if specific advanced features or performance improvements are needed.

Below is a comprehensive "Theoretical and Conceptual Framework" section for PeAS. This section explains the guiding theories related to information retrieval, search indexing, and digital archiving, and includes a diagram that illustrates the system architecture.

Theoretical and Conceptual Framework

Theoretical Framework

PeAS is designed based on several interrelated theories and models drawn from the fields of information retrieval, search indexing, and digital archiving:

1. **Information Retrieval (IR) Theories:**
    - **Boolean, Vector Space, and Probabilistic Models:**
      These foundational models inform how documents are represented and ranked. For example, the vector space model treats documents as vectors in a high-dimensional space where similarity (often measured by cosine similarity) indicates relevance (Croft, Metzler, & Strohman, 2009). Techniques such as term frequency–inverse document frequency (TF-IDF) are central to many modern IR systems and are implicitly applied in PeAS's search functionality through Meilisearch.
    - **Relevance Ranking and Indexing:**
      Modern search engines leverage advanced algorithms to index and rank documents for efficient retrieval. Concepts from the probabilistic relevance model help refine search results by estimating the probability that a given document is relevant to a query (Baeza-Yates & Ribeiro-Neto, 2011).
2. **Digital Archiving Theories and Models:**
    - **OAIS Reference Model:**
      The Open Archival Information System (OAIS) framework (ISO 14721:2012) provides a comprehensive model for preserving and providing access to digital information over the long term. It guides the

design of systems like PeAS to ensure that digitized academic outputs are maintained with integrity, authenticity, and accessibility.

    - **Metadata Standards and Indexing:**
      Effective digital archiving relies on the accurate description and indexing of documents. Frameworks such as Dublin Core, along with principles outlined by Conway (2011), ensure that metadata is standardized to support efficient search and retrieval.

By integrating these theoretical foundations, PeAS not only provides a user-friendly interface for document retrieval but also ensures that the archival process is robust, scalable, and aligned with best practices in digital preservation.

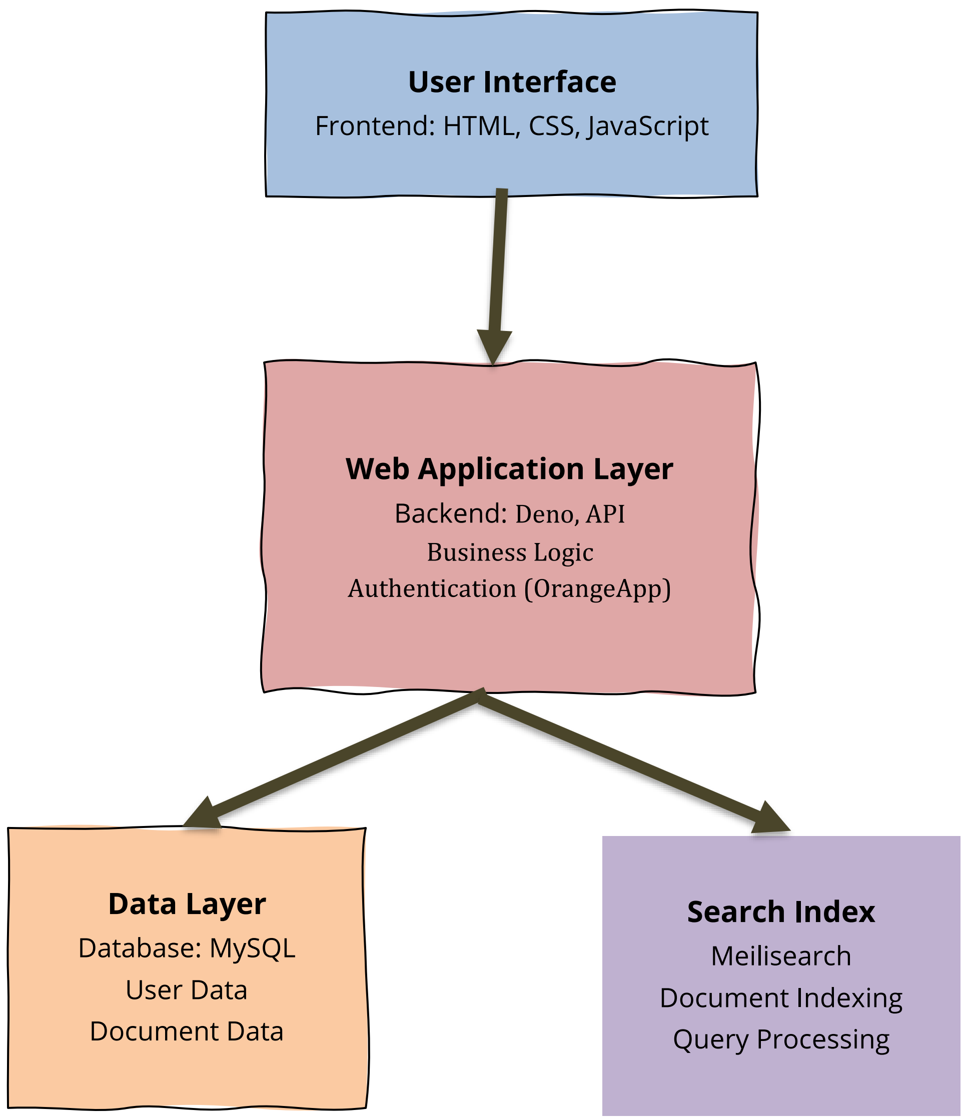
Conceptual Framework and System Architecture

The conceptual framework for PeAS is organized into several layers that work together to support the entire document lifecycle—from digitization and metadata extraction to secure storage and efficient search. The system is composed of:

- **Presentation Layer:**
  The frontend is developed using HTML, CSS, and JavaScript to create a responsive and intuitive user interface.
- **Application (Business Logic) Layer:**
  The backend, built on Deno, handles core functionalities such as document processing, user authentication (integrated with SPUD's OrangeApp system), and API management.
- **Data and Search Layer:**
  Document metadata and user data are stored in a MySQL database, while full-text search functionality is provided by Meilisearch, which indexes the digitized documents.

The following diagram illustrates the overall system architecture of PeAS:

System Architecture Overview Diagram:



- The **User Interface** enables users (students, faculty, and researchers) to interact with the system seamlessly.
- The **Web Application Layer** is responsible for processing user requests, enforcing authentication, and implementing business logic.
- The system's core data is maintained in the **Data Layer** (MySQL), while the **Search Index Layer** (Meilisearch) provides fast, accurate retrieval of documents based on indexed metadata and full-text content.

This layered architecture reflects both the theoretical underpinnings of information retrieval and digital archiving and provides a scalable, modular approach to system design.

Development Process

Development Tools and Frameworks

The development process of PeAS leverages a modern toolset that streamlines both the frontend and backend development:

- **Frontend Tools:**
  - **HTML, CSS, and JavaScript:**
    These core web technologies are used to build a responsive and intuitive user interface. Tools such as Visual Studio Code serve as the primary code editor, with integrated debugging and linting extensions.
  - **Version Control:**
    Git is used for source code management, with repositories hosted on platforms like GitHub to facilitate collaboration, code reviews, and continuous integration.
- **Backend Tools:**
  - **Deno:**
    As the runtime for the server-side logic, Deno is chosen over traditional PHP due to its modern, secure, and TypeScript-friendly environment. Deno's built-in tooling simplifies dependency management and testing.
  - **APIs and Frameworks:**
    The backend is structured around RESTful APIs that interact with the frontend and other system components. Lightweight frameworks available in the Deno ecosystem aid in routing, middleware support, and error handling.

- **Search Engine Integration:**
  - **Meilisearch:**

    This tool is integrated for fast and efficient full-text search. Its intuitive API and low resource footprint allow rapid indexing and retrieval of academic documents.
- **Database Management:**
  - **MySQL:**

    MySQL is used to manage relational data, including user accounts, document metadata, and system logs. The choice of MySQL is driven by its widespread use, robust performance, and extensive community support.
- **Collaboration and Project Management Tools:**
  - **JIRA or Trello:**

    These platforms are used to manage the product backlog, track sprint progress, and facilitate Agile ceremonies.
  - **Confluence or Notion:**

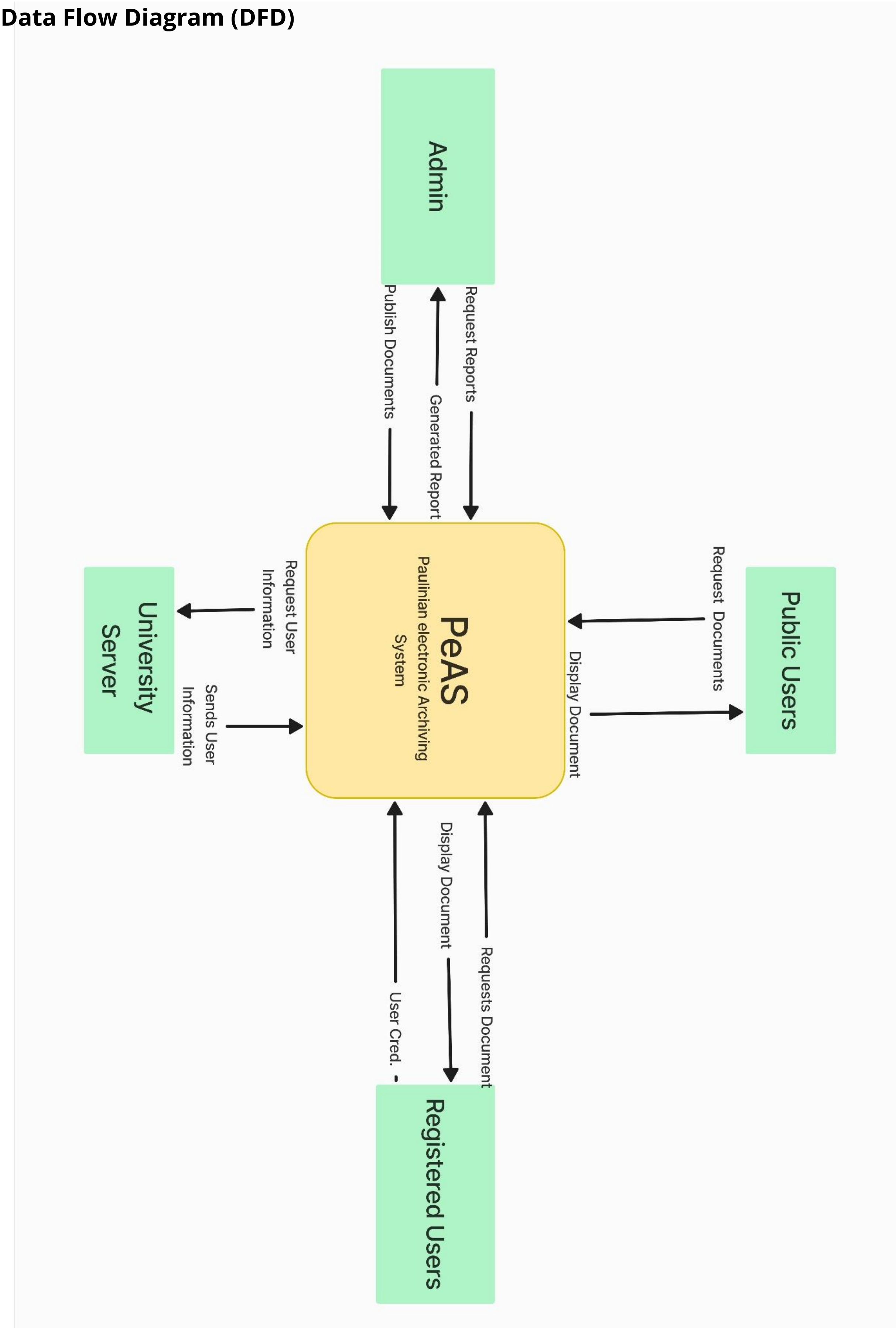
    Documentation and knowledge sharing across the team are managed through collaborative platforms, ensuring that all stakeholders have access to up-to-date project information.

Paulinian electronic Archiving System (PeAS) architecture and implementation.

System Overview and Core Architecture

The Paulinian electronic Archiving System represents a sophisticated document management and user authentication platform designed to serve multiple stakeholder groups while maintaining strict security protocols. The system's centralized architecture facilitates efficient document flow management and user access control through a single processing hub. See diagram on the next page.

Data Flow Diagram (DFD)



Core System Components and Integration

The central PeAS engine serves as the primary processing unit, managing four distinct interface layers:

Administrative Interface

The administrative component provides system governance through sophisticated control mechanisms. Administrators maintain oversight of document publication workflows, generate system reports, and implement access control policies. This interface requires enhanced security protocols and provides elevated system privileges for comprehensive system management.

Public Access Layer

The public interface implements a stateless access model, allowing unauthenticated users to request and view permitted documents. This layer operates with read-only capabilities and incorporates security measures to prevent unauthorized access to restricted content.

Registered User Portal

This interface layer handles authenticated user interactions, implementing full credential validation against the University Server database. It provides enhanced document access capabilities while maintaining strict security protocols and user-specific permissions.

University Server Integration

The system maintains continuous communication with the University Server for user information validation and authentication services. This integration ensures institutional data integrity and provides a reliable source for user verification.

Technical Implementation Specifications

Document Management Infrastructure

The system implements a sophisticated document processing pipeline that handles:

1. Asynchronous request processing from multiple user types
2. Role-based access control verification
3. Secure document presentation
4. Version control and administrative tracking
5. Metadata management for efficient retrieval

Database Architecture

The database structure supports dual functionality through distinct but integrated components:

Document Repository Management

- Hierarchical storage implementation
- Comprehensive metadata tracking
- Version control mechanisms
- Granular permission management through ACLs
- Backup and recovery protocols

User Management System

- Secure credential storage
- Token-based session management
- Role-based access definitions
- University Server integration for verification

Security Framework

The security implementation encompasses multiple layers of protection:

Data Protection Measures

- End-to-end encryption protocols
- SSL implementation for secure transmission
- Comprehensive audit logging
- Regular security assessment procedures

Access Control Framework

- RBAC implementation
- IP-based restrictions
- Session management protocols
- Complete audit trail maintenance

System Optimization and Performance

To ensure optimal system performance, the architecture implements:

Resource Management

- Efficient document storage optimization
- Memory allocation for concurrent user sessions
- Processing power distribution
- Cache management for frequent access patterns

Load Distribution

- Request processing distribution
- High-volume queue management
- Concurrent user handling
- Response time optimization

Integration Considerations

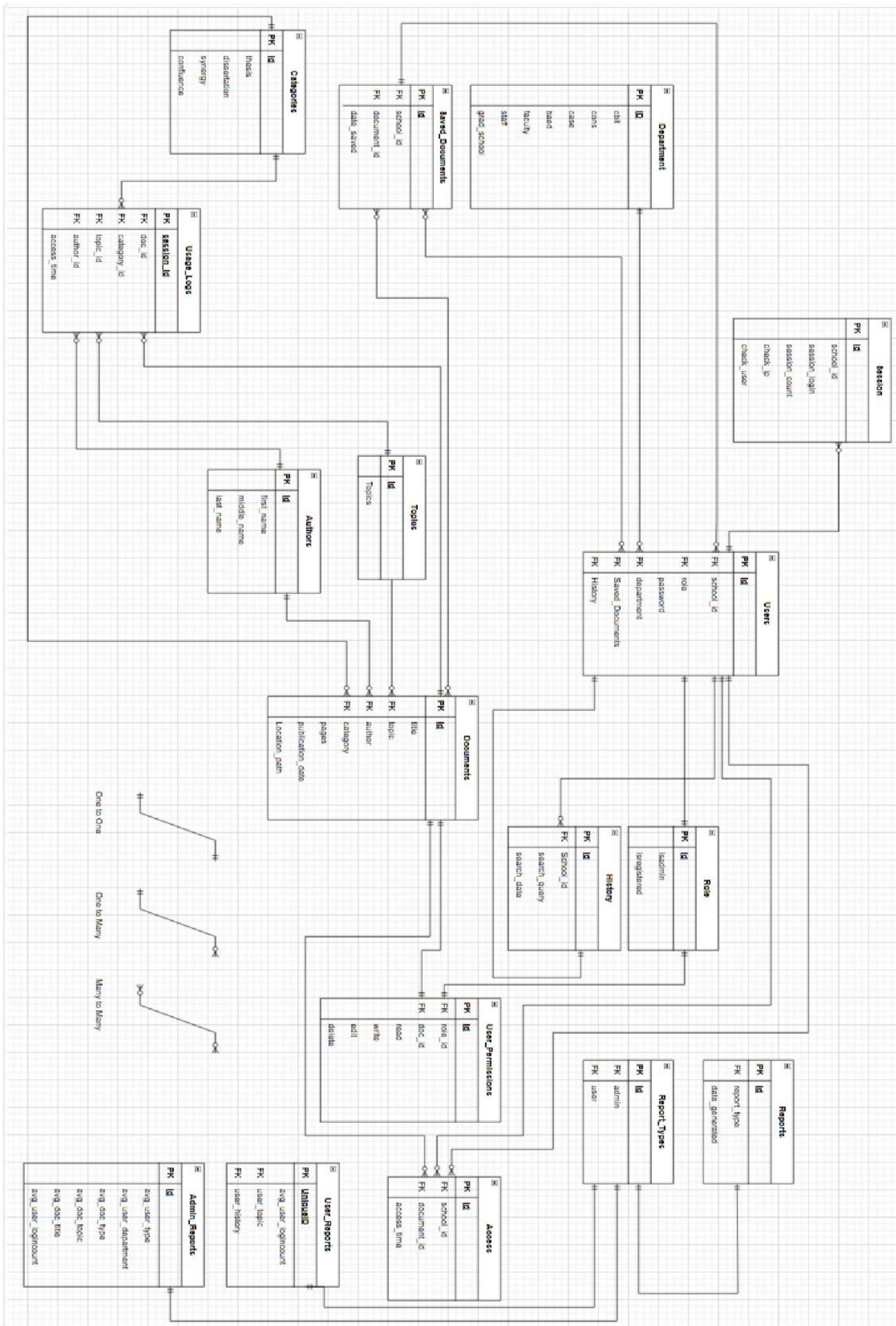
The system's modular design facilitates future scalability while maintaining current operational efficiency. The architecture supports potential expansion of user types and document management capabilities without compromising system integrity or performance.

This comprehensive implementation ensures robust document management while maintaining security and efficiency. The system's design principles focus on balancing accessibility with security, ensuring appropriate document access while protecting sensitive information through multiple security layers.

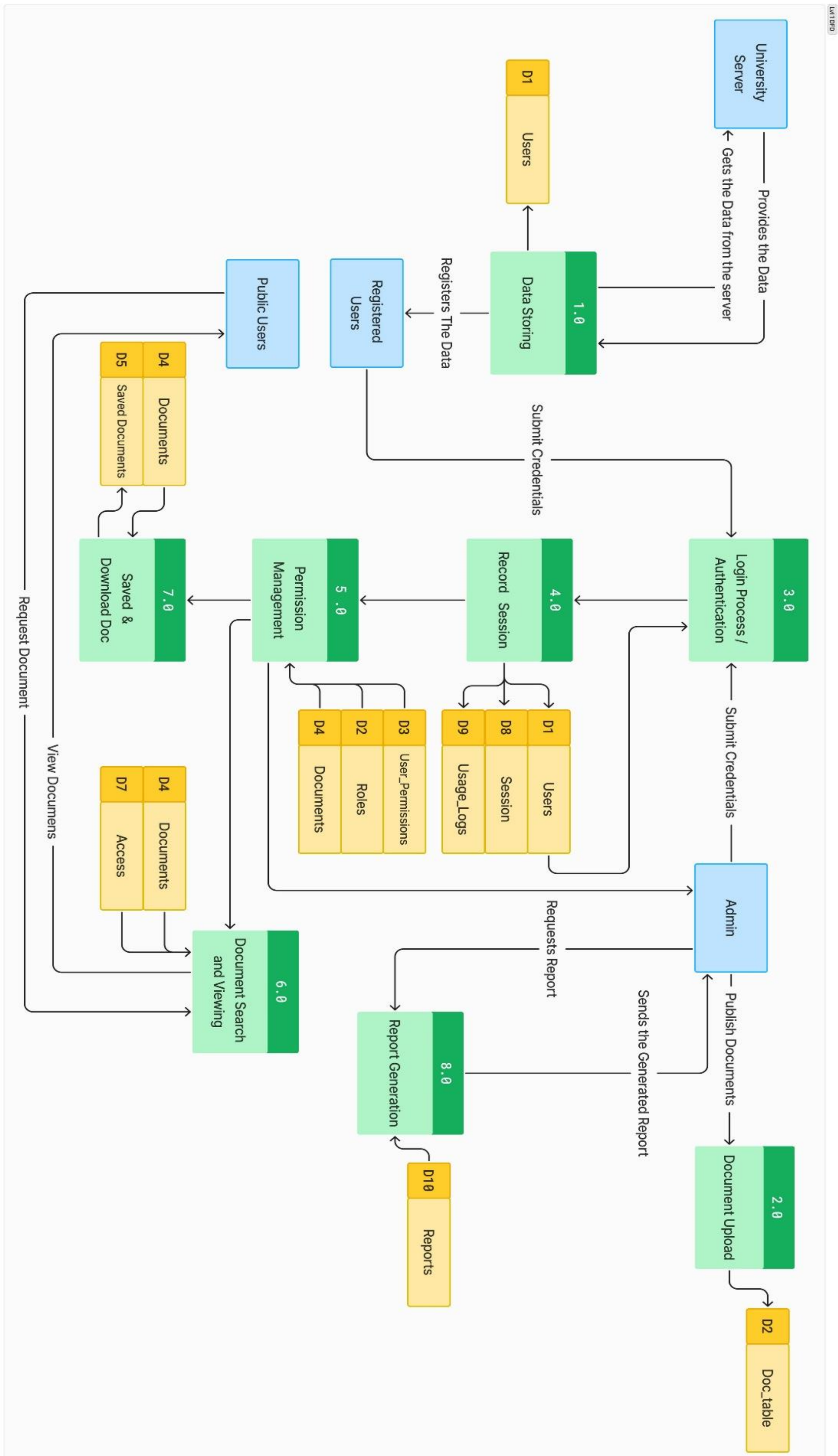
The architecture demonstrates a thoughtful approach to electronic archiving, providing a scalable solution that meets current institutional needs while maintaining flexibility for future expansion. The integration of multiple user interfaces with centralized processing creates an efficient, secure, and user-friendly document management system.

System Architecture Diagrams

Entity Relationship Diagram (ERD)



Level 1 Data Flow Diagram (DFD)



Implementation Details

Meilisearch: Indexing and Search

Indexing Process:
Meilisearch operates as a lightweight, fast search engine by creating inverted indexes for the documents stored in PeAS. When documents are added or updated, their text content and associated metadata (such as author, title, and year) are parsed and tokenized. Each token (word or phrase) is then mapped to the documents in which it appears. This process is fully automated through Meilisearch's API, ensuring that the search index is continuously updated in near real-time as changes occur in the repository.

Search Functionality:
Meilisearch supports advanced full-text search features including:

- **Fuzzy Matching:** Allowing tolerance for minor typos and spelling errors, which enhances the retrieval accuracy for user queries.
- **Ranking Algorithms:** The system uses custom ranking rules that consider factors such as term frequency, proximity, and relevance to quickly present the most pertinent documents.
- **Faceted Search:** In addition to keyword queries, Meilisearch supports filtering based on predefined facets (e.g., document type, publication year), enabling users to refine search results effectively.

For example, when a user enters a query on PeAS, Meilisearch performs a fast lookup on the inverted index and returns results sorted by relevance. This efficiency is crucial for academic users who depend on rapid, accurate search capabilities (Meilisearch Blog, 2023).

Deno's Role in CRUD Operations

Backend Runtime:

PeAS leverages Deno as its backend runtime environment to execute server-side logic. Deno, a modern JavaScript/TypeScript runtime built on the V8 engine, offers inherent security features, such as sandboxing and strict permission controls, that contribute to a more secure and robust system.

CRUD Operations:

Within PeAS, Deno is responsible for handling all Create, Read, Update, and Delete (CRUD) operations through a set of RESTful APIs:

- **Create:** When new academic documents are digitized, Deno processes the incoming data, assigns standardized metadata, and inserts records into the MySQL database. This includes uploading files, parsing content, and ensuring that indexing requests are sent to Meilisearch.
- **Read:** User queries and data retrieval operations are managed by Deno. It fetches the requested document data from MySQL, interfaces with Meilisearch to retrieve search results, and delivers the data back to the frontend in a structured JSON format.
- **Update:** For operations such as modifying document metadata or updating user information, Deno accepts input from authorized users, validates the changes, and updates the corresponding records in the database. Deno's non-blocking, asynchronous I/O model ensures that these updates occur efficiently.

- **Delete:** Deno handles deletion requests by removing records from MySQL and ensuring that any corresponding entries in Meilisearch's index are also purged, maintaining data consistency across the system.

This use of Deno for CRUD operations not only simplifies backend development by using modern language features (including TypeScript for type safety) but also enhances performance and security compared to traditional PHP-based systems (Deno, n.d.; Stack Overflow, 2020).

User Roles and Permissions

PeAS enforces a tiered access model to ensure that different types of users have appropriate permissions:

- **Public Users:**
  Public users are allowed to view document metadata and browse the repository's contents. They can search for academic outputs and access abstracts or summaries, but they are restricted from downloading full documents or making any changes to the content.
- **Registered Users:**
  Registered users (e.g., SPUD students and faculty with authenticated accounts) have enhanced privileges. These users can:
  - Download full document files,
  - Bookmark and annotate documents,
  - Track usage history,
  - Engage in collaborative features like commenting on or rating documents.

The system enforces these permissions via role-based access control mechanisms integrated into Deno's backend, which checks each request against the user's role before performing CRUD operations. This ensures that sensitive academic data is only accessible to authorized users.

- **Administrators:**

  Administrators possess the highest level of access. They are responsible for:

  - Managing user accounts and roles,
  - Overseeing document uploads and deletions,
  - Monitoring system logs and security protocols,
  - Handling maintenance and configuration of the system.

Administrator privileges are tightly controlled and audited to prevent unauthorized modifications, ensuring compliance with institutional policies and data protection standards (Armstrong Archives, 2018).

In summary, Meilisearch provides fast, relevant search results by efficiently indexing document content and metadata, while Deno handles secure, non-blocking CRUD operations for PeAS. The system's user roles and permissions model ensures that only authorized users can perform sensitive operations, maintaining the integrity and security of the academic repository

---

Testing and Evaluation

Performance Testing for Search Speed

Objective:

To ensure that search functionality returns results quickly and reliably, even under high load.

Approach:

- **Load and Stress Testing:** Simulate many concurrent users to verify that the search engine (e.g., Meilisearch) maintains low latency and high throughput.
- **Benchmarking:** Measure response times, throughput, and resource utilization. Tools like Apache JMeter or similar performance testing frameworks can be used.
- **Bottleneck Identification:** Analyze metrics to determine if any part of the system slows down search speed and apply performance optimizations accordingly.

User Acceptance Testing (UAT)

Objective:

To validate that the system meets business requirements and user expectations before full-scale production deployment.

Approach:

- **Realistic Scenarios:** Develop test scripts or use case scenarios based on actual user tasks. This ensures that features such as search functionality and document management work as expected.

- **End-User Involvement:** Engage representative users (or proxies) to run the tests. Their feedback helps uncover usability issues and confirms that the system delivers the intended value.
- **Iterative Testing:** Allow for rounds of testing and feedback, with issues being tracked, resolved, and retested until the system meets acceptance criteria.

Security Considerations

Objective:

To safeguard the system against vulnerabilities and ensure the protection of sensitive data.

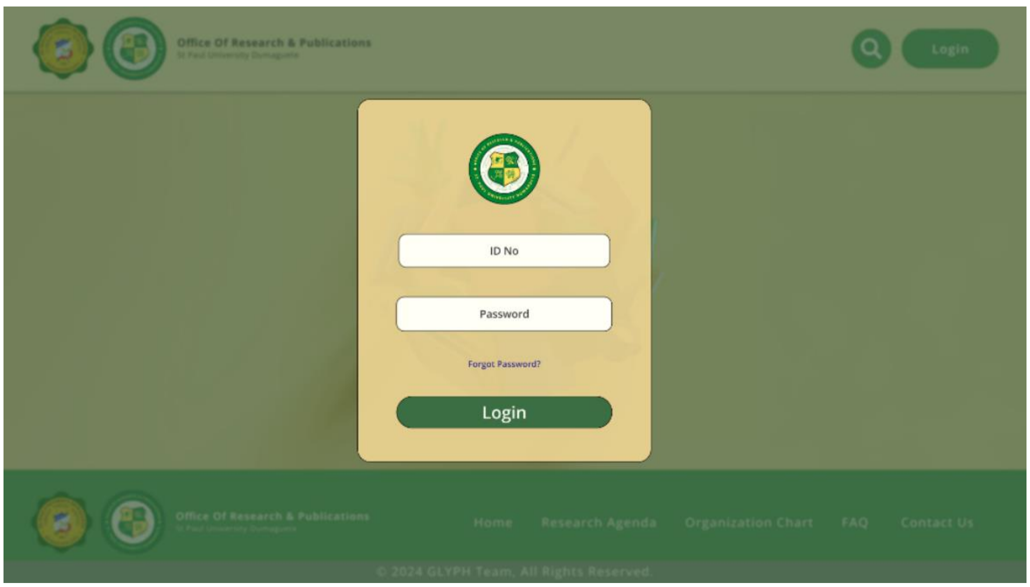
Approach:

- **Penetration and Vulnerability Testing:** Simulate attacks using tools (e.g., OWASP ZAP, Burp Suite) to identify weaknesses in both the frontend and backend.
- **Static and Dynamic Analysis:** Perform code reviews (static analysis) and monitor the running system (dynamic analysis) to catch security issues early.
- **Compliance with Standards:** Ensure that the testing process aligns with recognized security frameworks (e.g., OWASP Testing Guide) and best practices, protecting against unauthorized access and data breaches.
- **Continuous Monitoring:** Implement ongoing security assessments to catch emerging threats, particularly as new features or updates are deployed.

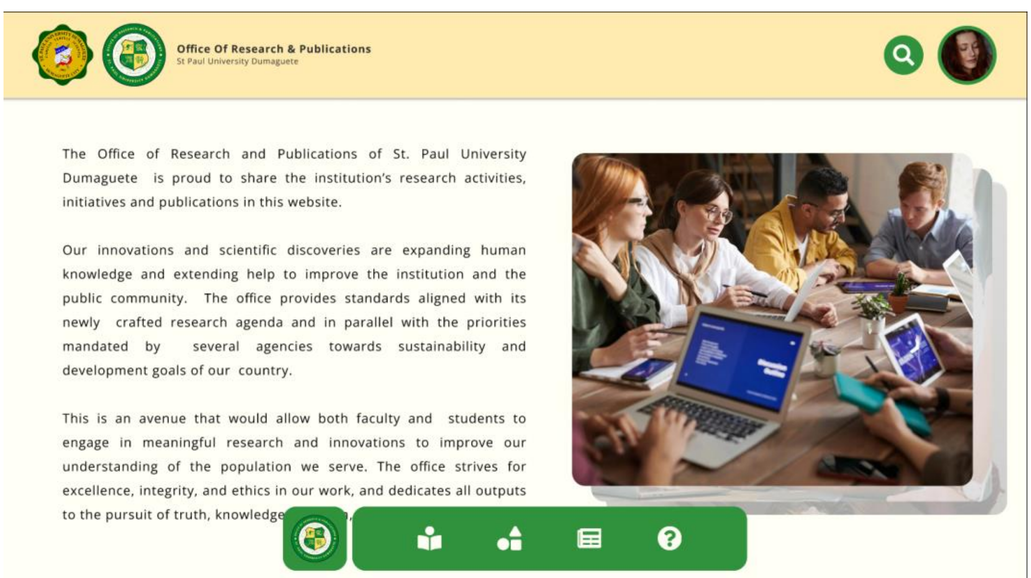
Screenshots of Paulinian electronic Archiving System (PeAS) in action

Users

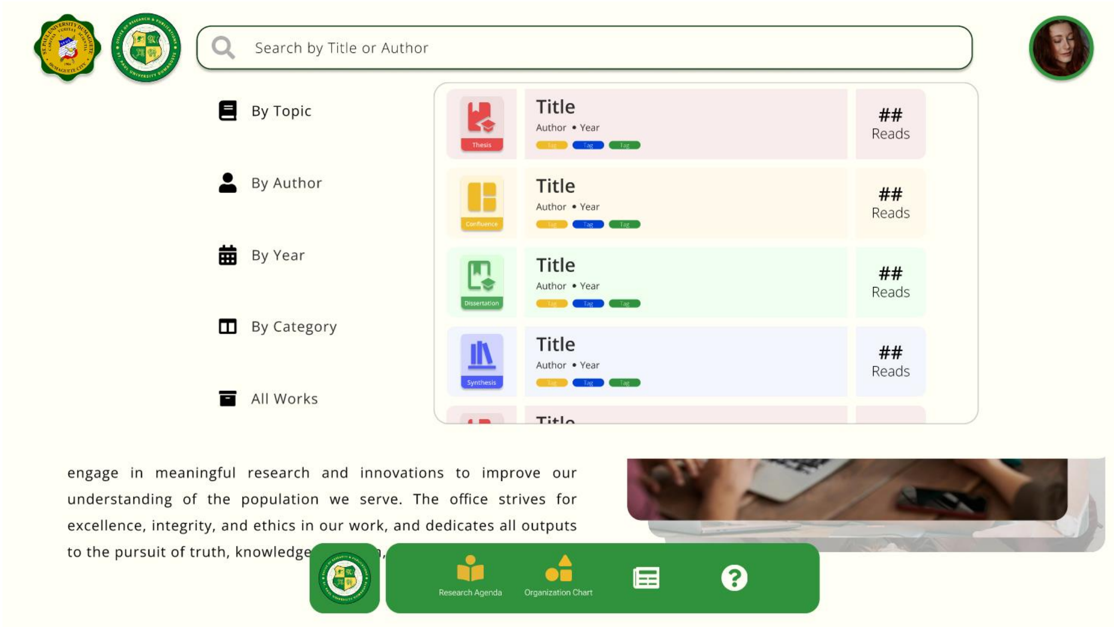
Login Page



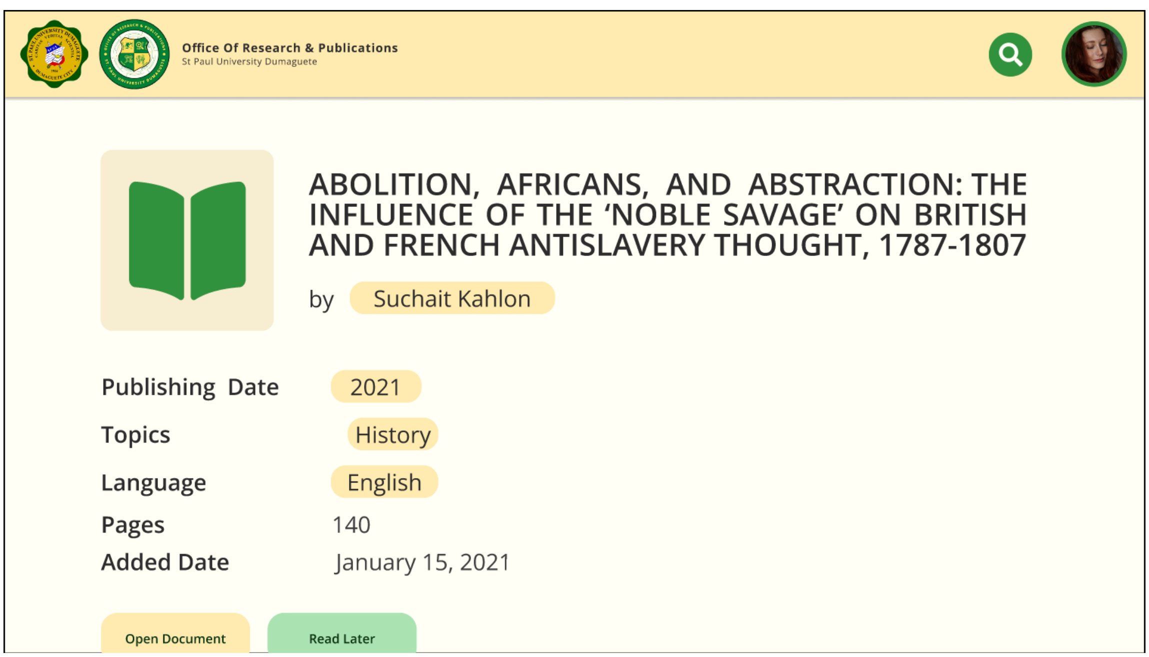
User Logged-in



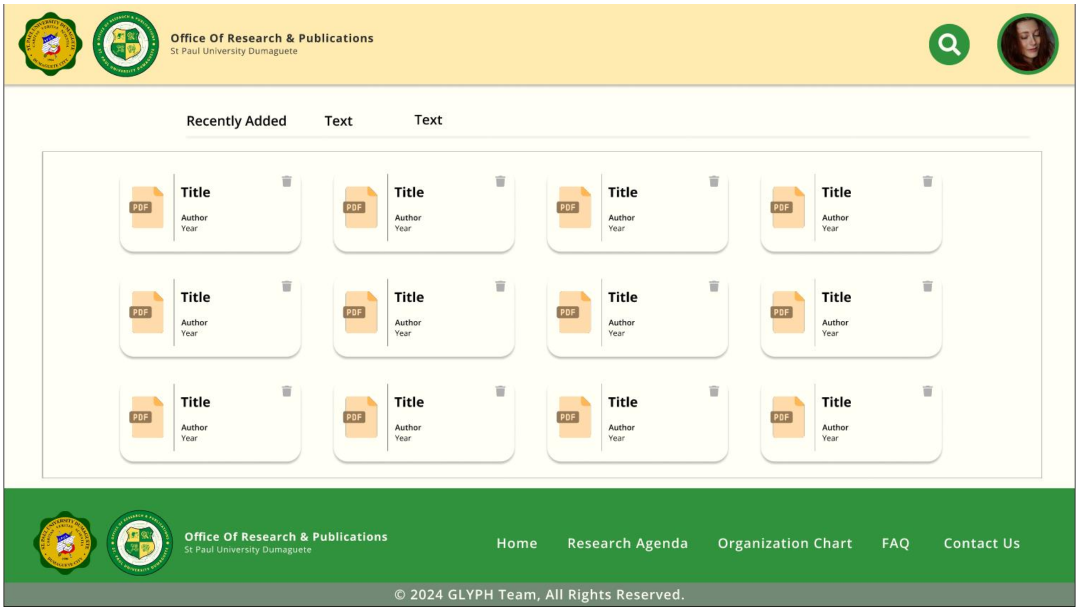
User Searching



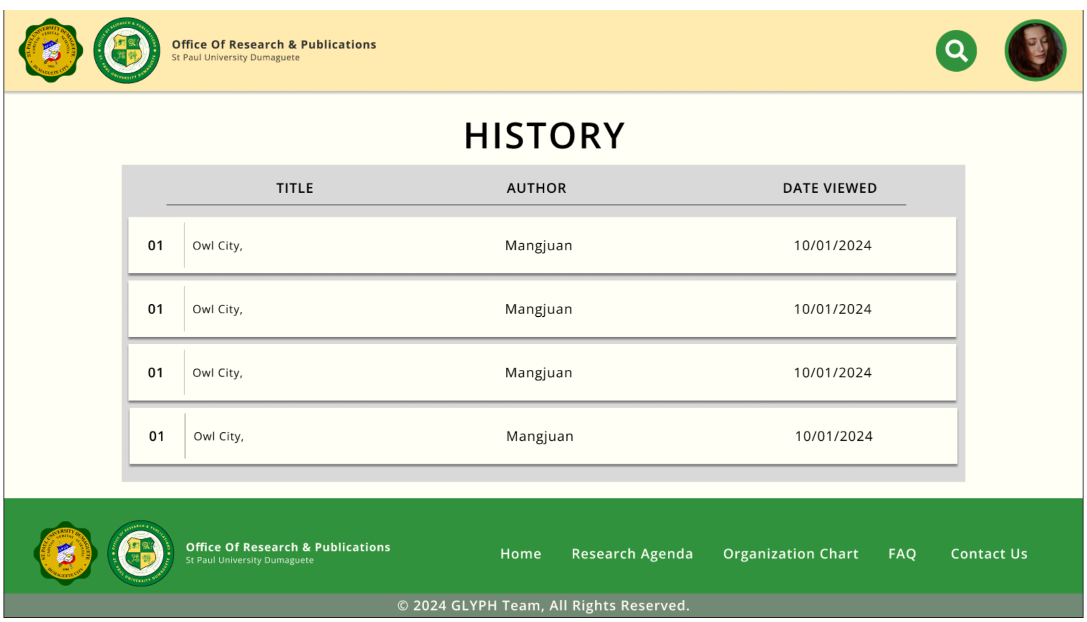
User Opens Document



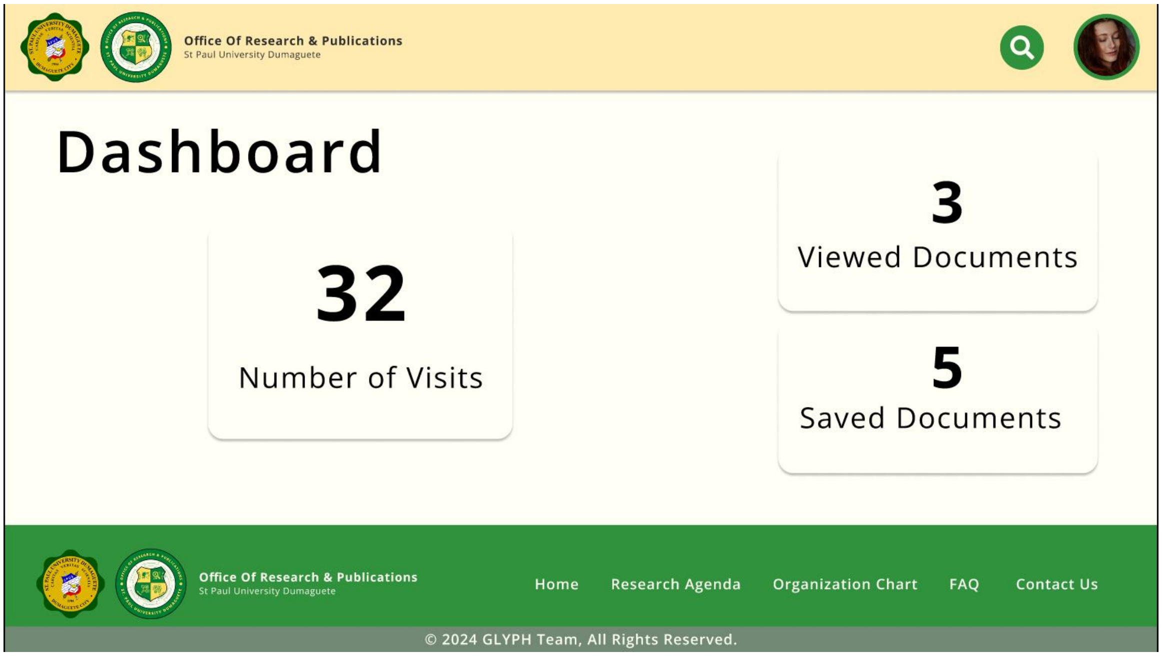
User Recently Added Documents



User History

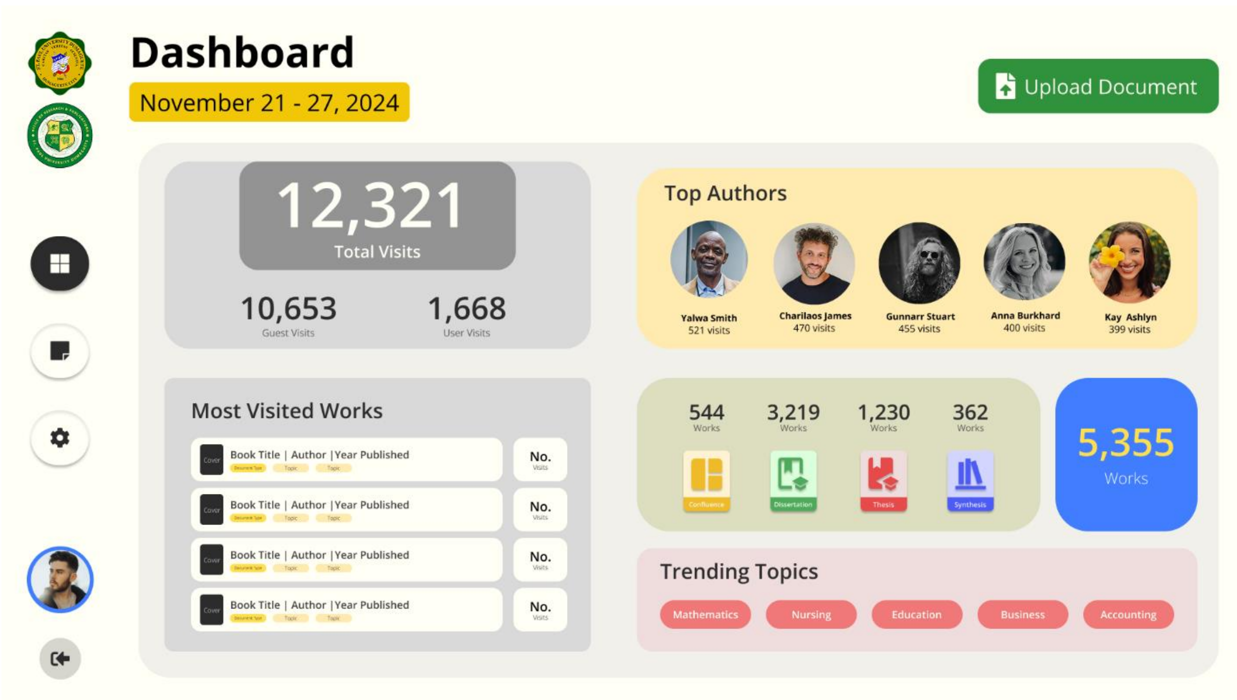


User Dashboard

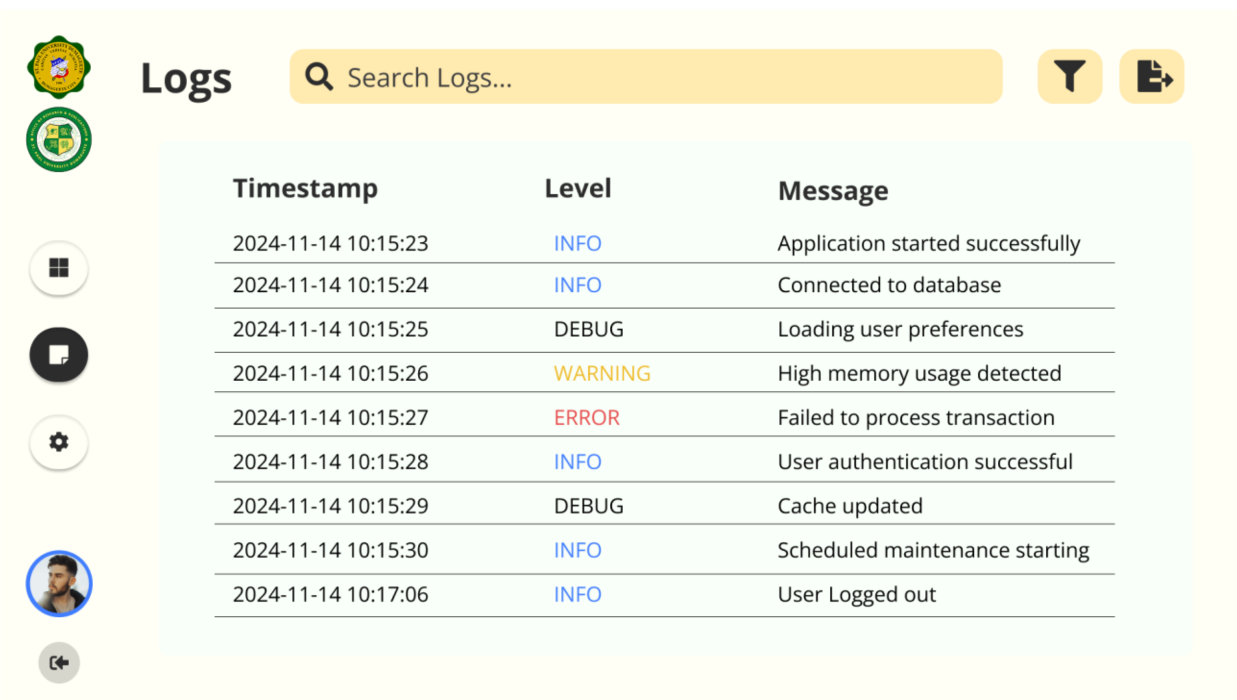


Admin

Dashboard

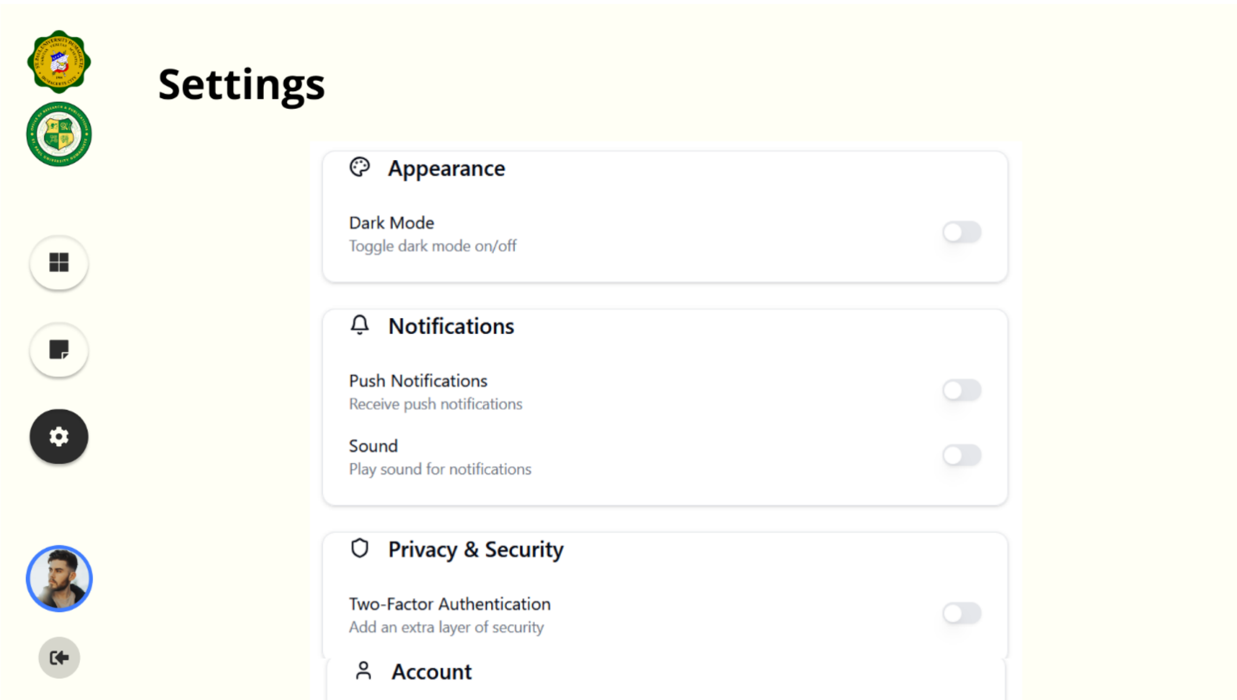


Logs

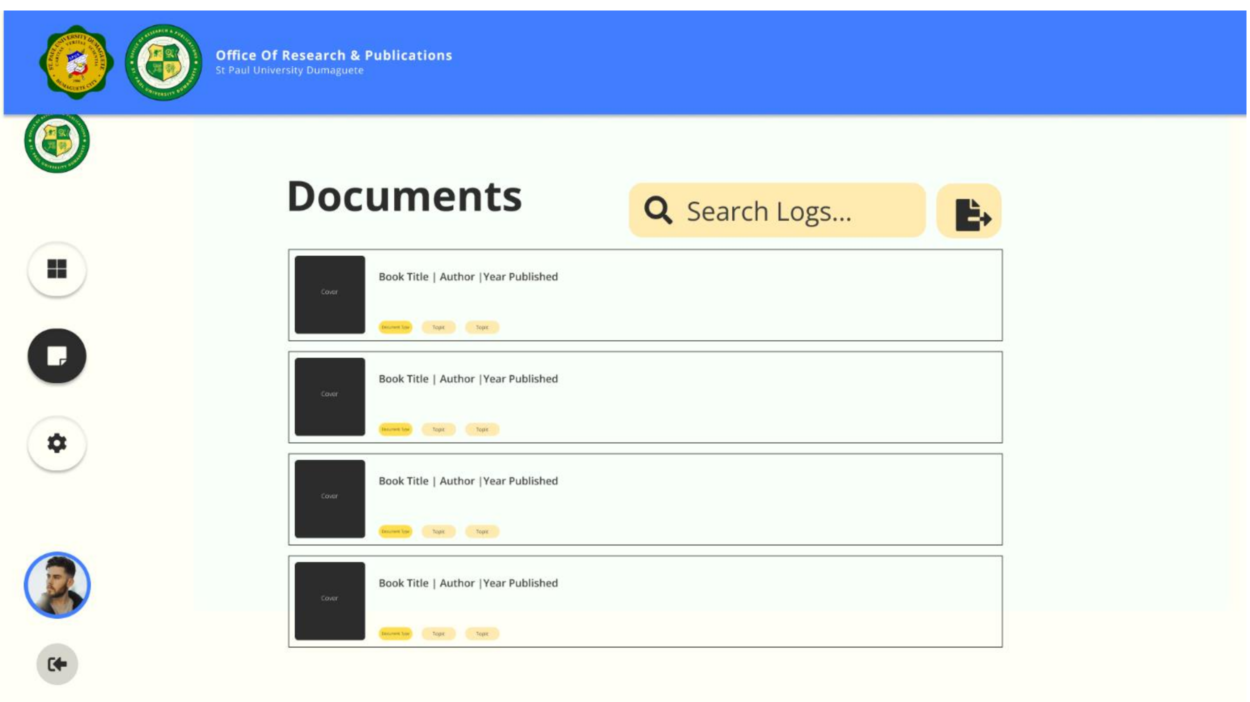


Add New Document

Settings



All Document List



References

Access Corporation. (2025). *Education document management solutions*. Retrieved from https://www.accesscorp.com/industries/education/

Anderson Archival. (2024). *The role of digital archives in education*. Retrieved from https://andersonarchival.com/blog/the-role-of-digital-archives-in-education/

Armstrong Archives. (2018). *Document management for educational institutions*. Retrieved from https://www.armstrongarchives.com/document-management-for-educational-institutions/

Atlassian. (n.d.). *The different types of software testing*. Retrieved February 7, 2025, from https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing

Caluza, L. J. B. (2019). *Development of Electronic Document Archive Management System (EDAMS): A case study of a university registrar in the Philippines* [PDF]. Retrieved from https://www.researchgate.net/publication/334415461_Development_of_Electronic_Document_Archive_Management_System_EDAMS_A_Case_Study_of_a_University_Registrar_in_the_Philippines

Conway, P. (2011). Archival quality and long-term preservation: A research framework for validating the usefulness of digital surrogates. *Archival Science, 11*(2), 99–114. https://doi.org/10.1007/s10502-011-9155-0

Cser, T. (2024, September 23). *User acceptance testing: Complete guide with examples*. Functionize. Retrieved February 7, 2025, from https://www.functionize.com/automated-testing/acceptance-testing-a-step-by-step-guide

DigitalOcean. (n.d.). *MySQL vs. PostgreSQL: Which one is right for you?* Retrieved from https://www.digitalocean.com/community/tutorials/mysql-vs-postgresql-comparison

Digital Preservation Coalition. (2019). *Introduction: Definitions and concepts*. In *Digital Preservation Handbook*. Retrieved from https://www.dpconline.org/

Deno. (n.d.). *Deno – A secure runtime for JavaScript and TypeScript*. Retrieved from https://deno.land/

Marsh, C., Wackerman, D., & Stubbs, J. A. W. (2017). *Creating an institutional repository: Elements for success!*. Retrieved from https://www.serials-librarian.com/

MDN Web Docs. (n.d.). *HTML, CSS, and JavaScript documentation*. Retrieved from https://developer.mozilla.org/en-US/docs/Web

Meilisearch Blog. (2023). *Meilisearch vs. Elasticsearch*. Retrieved from https://www.meilisearch.com/blog/meilisearch-vs-elasticsearch

OWASP. (n.d.). *OWASP testing guide v4*. Retrieved February 7, 2025, from https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf

Peterbe. (2023, January 26). *First impressions of Meilisearch and how it compares to Elasticsearch*. Retrieved from https://www.peterbe.com/plog/first-impressions-of-meilisearch-and-how-it-compares-to-elasticsearch

Stack Overflow. (2020). *Deno performance compared to Node.js*. Retrieved from https://stackoverflow.com/questions/61663266/deno-performance-compared-to-node-js

WMTips. (n.d.). *Deno vs. PHP: Usage and pricing comparison*. Retrieved from https://www.wmtips.com/technologies/compare/deno-vs-php/

Wikipedia contributors. (2025, January). *Software testing*. In *Wikipedia, The Free Encyclopedia*. Retrieved February 7, 2025, from https://en.wikipedia.org/wiki/Software_testing

---