Gage University College

Department of Computer Science

ELECTRONIC DOCUMENT MANAGEMENT SYSTEM

FOR

Gage University College

- Degree of Computer Sience
  Department of Computer Science
  Gage University College
  Date of Submission - July 21, 2025
- Project-URL: https://edmos.net , https://admin.edmos.net

Gage University College

Department of Computer Science

M.Sc. in Computer Science

DESIGN AND IMPLEMENTATION OF ELECTRONIC DOCUMENT MANAGEMENT SYSTEM FOR

GAGE UNIVERSITY COLLEGE

A Project Proposal Submitted to the Department of Computer Science, Gage University College, in Partial Fulfillment of the Requirements for the Degree of Computer Science.

Submitted By:

- FUAD AHMED
  MRCS/023/2014

- FMEKEDELAWIT DAMETEW
  MRCS/020/2014

- SEBIT REATY
  MRCS/029/2014

Adviser:
ZELALEM BELAY (MSC, MA)
Department of Computer Science, Gage University College

Date: July 21, 2025
Addis Ababa, Ethiopia

Abstract

Efficient document management is a foundational requirement for the effective operation of any academic institution. Gage University College, in line with many universities in the region, currently relies predominantly on manual systems for handling its vast and growing volume of documents. This traditional approach is inherently inefficient and poses significant challenges, including time-consuming retrieval processes, high risk of document loss or damage from various threats, inefficient utilization of physical space, difficulties in maintaining accurate version control, and inadequate security measures to protect sensitive information. These issues collectively contribute to reduced administrative productivity, delayed decision-making, and potential compromise of the integrity and accessibility of crucial college records.

This project proposes the design and implementation of a modern, web-based Electronic Document Management Online System (EDMOS) specifically tailored to address the identified challenges and streamline document management processes at Gage University College. The system will provide a centralized, secure, and intuitive platform for managing the complete lifecycle of both electronically generated and scanned documents. It will be architected as a single-page application (SPA) frontend built with **React** and **Vite**, communicating with a robust **Laravel** backend API, with data persisted in a **MySQL** database. The system will feature a public landing page and secure, distinct portals for both administrators and regular users. Key functionalities will encompass comprehensive document capture (uploading electronic files and facilitating the ingestion of scanned documents), intelligent indexing with rich metadata, secure hierarchical storage, efficient search and retrieval capabilities, automatic version tracking, granular role-based access control, a basic internal messaging/workflow notification system, and structured archiving. Employing an Agile methodology, the project prioritizes iterative development and close stakeholder collaboration to ensure the final system is practical, user-friendly, and directly aligns with the operational needs of Gage University College staff. The expected outcome is a scalable, maintainable, and secure EDMOS prototype that demonstrates significant improvements in document accessibility, administrative efficiency, and overall information governance.

Keywords: Electronic Document Management System, Gage University College, Laravel, React Vite, MySQL, Web Application, Document Archiving, Workflow Automation, Information System, Digitalization, Metadata Indexing.

Acknowledgements

I wish to express my deepest gratitude to Gage University College for providing the academic environment and the opportunity to undertake this project. I am particularly indebted to my supervisor ZELALEM BELAY (MSC, MA), for their continuous guidance, invaluable technical insights, constructive criticism, and unwavering support throughout the research and proposal writing process. Their expertise has been instrumental in shaping the direction and enhancing the quality of this project.

My sincere thanks are also extended to the staff members across various departments at Gage University College who generously dedicated their time to participate in interviews and discussions. Their willingness to share their experiences with the current manual system and articulate their requirements for a digital solution has been crucial in grounding this project in the practical needs of the college.

Finally, I am profoundly grateful for the constant encouragement, patience, and support from my family and friends, which have been essential for the completion of this proposal.

Table of Contents

Abstract iii
Acknowledgements iv
Table of Contents v
List of Tables vi
List of Figures vii
List of Acronyms and Abbreviations viii

Chapter One: Introduction 1
1.1 Background 1
1.2 Statement of the Problem 3
1.3 Objectives 4
1.3.1 General Objective 4
1.3.2 Specific Objectives 4
1.4 Scope of the Project 5
1.5 Significance of the Project 6
1.6 Organization of the Document 7

Chapter Two: Literature Review 8
2.1 Overview 8
2.2 Core Concepts in Electronic Document Management 8
2.3 Review of Related Systems and Solutions 10
2.4 Challenges in EDMOS Implementation and the Path Forward 12
2.5 Identification of Gaps and Project Justification 13

Chapter Three: Methodology 15
3.1 Introduction 15
3.2 Development Methodology (Scrum Framework) 15
3.3 Data Collection Methods 17
3.4 System Design and Modeling (UML) 18
3.5 Implementation Technologies and Environment (Laravel, React Vite, MySQL) 20
3.6 Testing Strategy 22
3.7 Ethical Considerations 24

Chapter Four: Project Plan (Budget & Schedule) 26
4.1 Project Budget 26
4.2 Project Schedule 28

References 35

List of Figures

Figure1.1: EDMOS ADMIN Interfaces.....14
Figure 1.2: EDMOS Analytics Interfaces 15
Figure 1.3: Document Handle of EDMOS 18
Figure 1.4: Document Fields for Large and Different Types of Documents..... 20
Figure 1.5: Document Comparison of EDMOS Solution type one..... 21
Figure 1.6: Document Comparison of EDMOS Solution type two..... 22
Figure 1.7: Deployment Diagram of the EDMOS 27
Figure G.1: High-Level EDMOS Use Case Diagram 28
Figure G.2: Activity Diagram for Documents Retrieval..... 29
Figure G.3: Deployment Diagram of the EDMOS 30

List of Acronyms and Abbreviations

API - Application Programming Interface
CSS - Cascading Style Sheets
DB - Database
DMS - Document Management System
ECM - Enterprise Content Management
EDMOS - Electronic Document Management Online System
ETB - Ethiopian Birr
GUC - Gage University College
HTML - Hypertext Markup Language
JS - JavaScript
MVC - Model-View-Controller
MySQL - My Structured Query Language
N/A - Not Applicable
ORM - Object-Relational Mapper
PHP - Hypertext Preprocessor
REST - Representational State Transfer
RDBMS - Relational Database Management System
SDLC - System Development Life Cycle
SPA - Single-Page Application
UAT - User Acceptance Testing
UI/UX - User Interface / User Experience
UML - Unified Modeling Language
URL - Uniform Resource Locator
VS Code - Visual Studio Code
WCMS - Web Content Management System

Chapter One: Introduction

1. Introduction

In the contemporary landscape of higher education, the volume of documents generated and processed daily by institutions like Gage University College (GUC) is substantial and continuously increasing. Historically, GUC has relied heavily on manual, paper-based filing and storage systems. While functional to an extent, these traditional methods are increasingly inadequate for modern administrative demands. They result in slow document retrieval, vulnerability to loss or damage (e.g., from fire, water, or misfiling), inefficient use of physical space, and difficulties in maintaining security and version control. As universities move toward the “paperless office” paradigm, Electronic Document Management Systems (EDMOS) have emerged as structured software solutions to capture, organize, store, and track documents throughout their lifecycle in electronic format. By adopting an EDMOS, an institution can greatly improve information accessibility, strengthen data security, and support compliance, thereby modernizing its administrative processes and preserving institutional knowledge.

Gage University College recognizes the challenges posed by its current manual processes. Common pain points include long retrieval times, risk of losing critical files, space consumed by paper archives, and difficulty in tracking document versions. These issues reduce staff productivity and impede timely decision-making. The proposed EDMOS project is therefore a strategic initiative to address these problems and enhance the operational capacity of the college by providing a centralized, secure, and user-friendly platform for managing both existing scanned documents and newly generated electronic files.

1.1 Background

Effective document management is a foundational requirement for the efficient operation of any academic institution. In universities, documents serve as the official record of activities ranging from student records and research outputs to administrative reports and policy documents. Traditionally, universities like GUC managed these records through decentralized, manual filing systems. Such systems rely on physical distribution, filing, and localized storage, which are increasingly inadequate for a fast-paced, information-driven environment. The limitations of manual systems—slow retrieval, risk of damage or loss, inflexible version tracking, and poor security—hinder effective information governance. Digital technologies and the concept of the “paperless office” have driven a shift toward electronic document management solutions, which promise to improve efficiency and accessibility. An EDMOS provides the infrastructure to capture, store, search, and control documents electronically, and is therefore a strategic tool for modernizing institutional processes and preserving knowledge.

In the contemporary landscape of higher education, effective management of information assets is paramount. Documents, in their various forms, serve as the official record of institutional activities, academic processes, and administrative decisions. The volume of these documents generated and processed daily at universities like Gage University College is substantial and continuously increasing. Traditionally, managing this influx has relied heavily on manual, paper-based systems involving physical filing, distribution, and archival. While these methods have historical precedence, they are increasingly inadequate for the demands of a fast-paced, information-driven environment. The limitations include slow retrieval times, susceptibility to loss or damage, inefficient space utilization, and challenges in implementing robust security and version control.

The advent of digital technologies and the concept of a "paperless" or "less-paper" office have catalyzed a shift towards electronic document management. An Electronic Document Management Online System (EDMOS) provides a structured software solution to capture, organize, store, manage, and track documents throughout their lifecycle in an electronic format. Such systems are designed to improve efficiency, enhance accessibility, strengthen security, and support compliance. Adopting an EDMOS is a strategic move for academic institutions seeking to modernize their administrative processes, improve collaboration among staff and faculty, and ensure the long-term preservation and accessibility of institutional knowledge.

Gage University College, committed to academic excellence and administrative efficiency, recognizes the challenges posed by its current manual document management practices. Long retrieval times, the risk of losing critical files, the space consumed by physical archives, and the difficulty in tracking document versions and access are significant pain points. Implementing a tailored EDMOS is therefore a timely initiative to address these issues directly, improve staff productivity, and enhance the overall operational capacity of the college.

1.2 Statement of the Problem

The prevailing manual document management system at Gage University College is a significant impediment to efficient operations, leading to a cascade of interconnected problems. Key challenges include:

- **Inefficient Information Access and Retrieval:** Locating specific documents within decentralized physical filing systems is a labor-intensive and time-consuming process. This delays access to critical information, hindering timely decision-making and administrative responsiveness.
- **High Risk of Document Loss, Damage, or Misplacement:** Paper documents are vulnerable to physical deterioration, accidental damage (e.g., from fire, water), loss during transit or handling, and misfiling within physical archives. This puts vital institutional records at risk.
- **Suboptimal Use of Physical Space:** The extensive use of physical storage for documents requires significant office space that could be better utilized for academic or administrative activities.
- **Lack of Centralized Storage and Version Control:** Documents are often duplicated and stored locally in various offices, leading to inconsistencies, confusion regarding the most current version, and a fragmented view of institutional information.
- **Inadequate Security and Access Management:** Physical documents are difficult to secure granularly. Restricting access based on roles or specific user needs is challenging, potentially compromising confidentiality and integrity of sensitive information.

- **Opaque and Slow Workflows:** Manual document routing and approval processes are often slow, lack transparency, and make it difficult to track the status of a document or identify process bottlenecks.

These problems collectively reduce staff productivity, increase operational costs, and detract from Gage University College's ability to operate as a modern, agile institution. A dedicated EDMOS is needed to mitigate these risks and introduce systematic, efficient document handling.

1.3 Objectives

The primary goal of this project is to design, implement, and test a functional Electronic Document Management System tailored to the specific requirements and operational environment of Gage University College, thereby replacing the current manual system's inefficiencies with automated and streamlined processes.

1.3.1 General Objective

To successfully design, develop, and deliver a web-based Electronic Document Management System prototype for Gage University College that provides a secure, efficient, and centralized platform for managing electronic and scanned documents throughout their lifecycle.

1.3.2 Specific Objectives

1. To conduct a detailed analysis of the existing manual document management processes at Gage University College and gather comprehensive functional and non-functional requirements through interviews, observation, and document review.
2. To design the system architecture, including the database schema, backend API endpoints (Laravel), and the structure of the frontend user interfaces (React).
3. To implement the backend of the EDMOS using the Laravel framework, developing secure API services for document operations, user management, and system configuration, utilizing MySQL as the database.

4. To develop the frontend user interfaces as a responsive Single-Page Application using React and Vite, creating intuitive interfaces for the public Landing Page, the secure User Portal, and the secure Administrator Portal.
5. To integrate core EDMOS functionalities: secure document upload (supporting various electronic formats and scanned documents), comprehensive metadata indexing and management, hierarchical folder-based storage, advanced search and retrieval, version tracking, role-based access control based on GUC's structure, a basic internal messaging/workflow notification system, and document archiving.
6. To perform multi-level testing, including unit tests, integration tests (frontend-backend API), comprehensive system tests, security vulnerability assessments, and User Acceptance Testing (UAT) with representative GUC staff, to validate the system's functionality, performance, security, and usability.
7. To produce detailed technical documentation for the implemented system and a user guide for the different portals, along with recommendations for potential future system expansion and full-scale deployment.

These objectives follow the project's agile development approach, ensuring that the final system aligns with user needs and project goals.

1.4 Scope of the Project

The scope of this project is defined to focus on the core functionalities necessary for a viable EDMOS prototype within the timeframe of an M.Sc. project, specifically addressing the requirements identified for Gage University College. The system will be a web-based application accessible via standard web browsers and will consist of the following interfaces:

- **Landing Page:** A simple, static page providing an overview of the EDMOS and access points to the login page.
- **User Portal:** A secure, authenticated area for regular GUC staff to upload, index (add metadata), store, organize in folders, search, retrieve, view, download, track versions, and share documents based on their assigned permissions.
- **Administrator Portal:** A secure, authenticated area with administrative privileges to manage users (create, edit, delete), roles (define permissions), departments/offices (structure the organization), system configurations, view system reports (e.g., document counts, user activity), and manage the archiving process.

The system will support the ingestion and management of both **born-electronic documents** (e.g., Word files, PDFs, spreadsheets, images) and **scanned documents** (e.g., PDFs, images), providing mechanisms for adding metadata to both types.

Key functionalities included:

- User Authentication (Login) and Authorization (Role/Department-based Access Control).
- Secure File Upload (electronic and scanned).
- Metadata Management (defining and assigning metadata fields).
- Hierarchical Folder/Category Structure for organization.
- Advanced Search (by metadata and potentially basic full-text for compatible formats).
- Document Viewing and Downloading (with permission checks).
- Version Tracking (recording and accessing previous versions).
- Basic Document Routing/Sharing via an internal messaging system with notifications.
- User, Role, and Department Management (Admin only).
- Basic System Activity Logging/Reporting (Admin only).
- Document Archiving (marking documents as inactive and moving them to a designated archive area).

Functionalities specifically excluded from this project scope due to time and resource constraints, but which represent potential future work:

- Automated Optical Character Recognition (OCR) processing for full-text indexing of scanned images at the point of upload.
- Integration with other existing Gage University College systems (e.g., student information system, HR system).
- Development of a native mobile application.
- Complex, multi-step customizable workflow automation engine with approval queues and digital signatures.
- Tools for bulk import/migration of existing paper or scattered digital document repositories.
- Real-time collaborative document editing within the platform.
- Advanced reporting analytics or dashboard features.

FIGURE 1.1: EDMOS ADMIN Interfaces

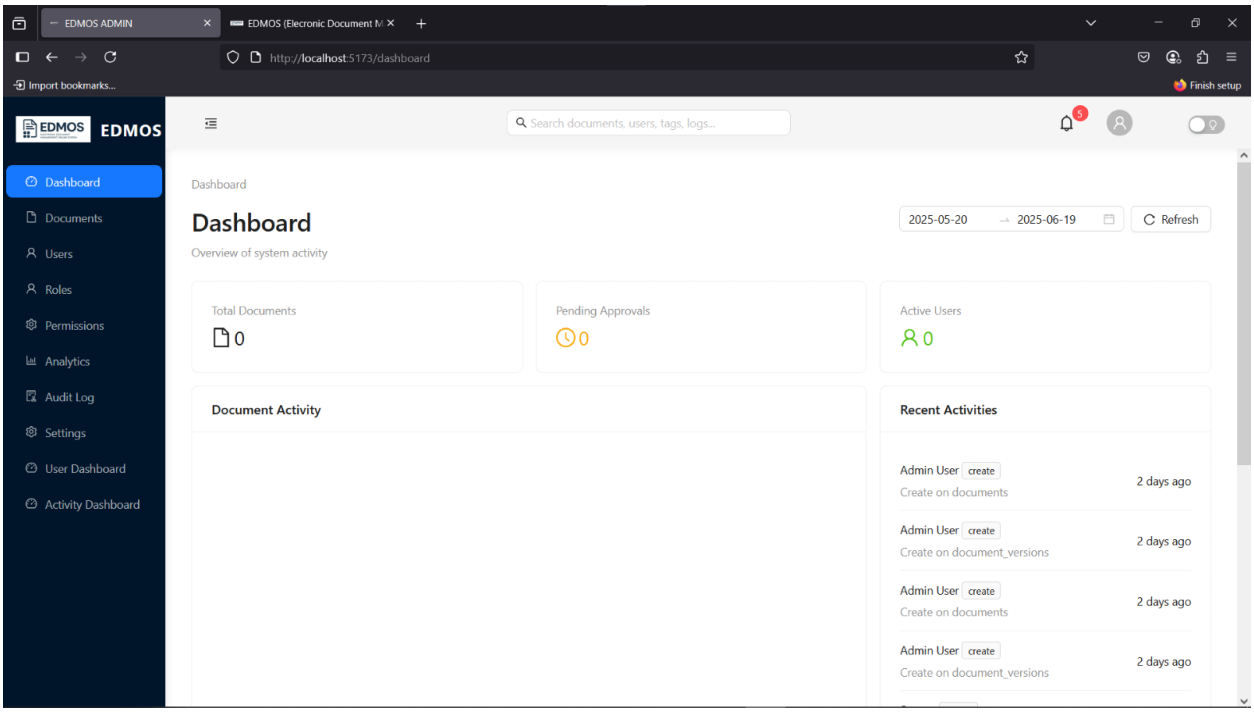
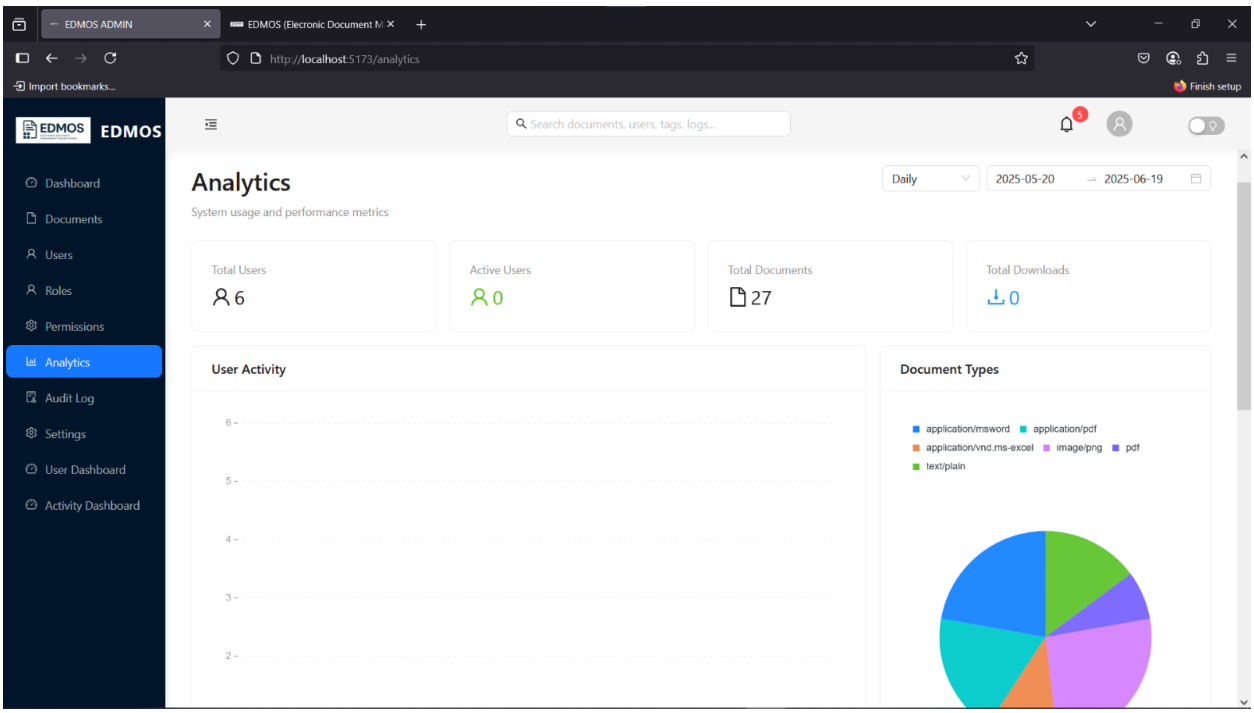


FIGURE 1.2: EDMOS ANALYTICS Interfaces



1.5 Significance of the Project

The proposed Electronic Document Management System, tailored for Gage University College, holds significant potential to improve operations and provide value across the institution. Its significance can be viewed from multiple perspectives:

- **Enhanced Administrative Efficiency:** By automating tasks like filing, searching, and routing, the system will dramatically reduce manual effort and time spent by staff, leading to faster processing of administrative tasks across departments (e.g., student applications, internal memos, HR records).
- **Improved Information Governance:** Centralized, organized storage combined with robust access controls and audit trails enhances the college's ability to manage its information assets effectively, ensuring data integrity, security, and accountability.
- **Increased Accessibility and Collaboration:** Staff will gain quick, secure access to necessary documents from authorized locations. The built-in sharing mechanism facilitates collaboration and information dissemination among relevant personnel.
- **Risk Mitigation:** Electronic storage significantly reduces the risk of permanent loss or damage to vital records compared to vulnerable paper files. The system supports basic archiving, aiding in long-term preservation.
- **Cost and Space Savings:** Over time, reducing reliance on paper will decrease costs associated with printing, copying, physical storage space, and filing supplies.
- **Support for Scanned and Electronic Documents:** The system's ability to handle both types of documents is crucial for transitioning from a paper-heavy environment while still managing newly generated electronic files effectively.
- **Foundation for Future Digital Initiatives:** A successful core EDMOS provides a foundational layer that can potentially be expanded with more advanced features (e.g., workflow, integration with other systems) in the future, supporting the college's broader digital transformation goals.

Ultimately, the project aims to deliver a practical tool that directly addresses current operational challenges and positions Gage University College for more efficient and secure information management in the future.

1.6 Organization of the Document

This proposal is structured into four main chapters, followed by references and appendices, to provide a clear and comprehensive overview of the planned EDMOS project for Gage University College.

Chapter One provides the **Introduction**, establishing the context, identifying the problem addressed by the project, outlining the specific objectives, defining the scope and limitations, and highlighting the significance and potential benefits for the college.

Chapter Two presents a **Literature Review**, synthesizing existing research and industry knowledge related to Electronic Document Management Systems. It covers core concepts, examines various existing solutions (commercial and open-source), discusses common implementation challenges, and articulates the specific gap that this project aims to fill, thereby justifying the proposed approach.

Chapter Three details the **Methodology**, describing the chosen software development lifecycle (Scrum), the methods for gathering and analyzing user requirements (interviews, observation, document review), the system design techniques (UML modeling), the specific technology stack selected for implementation (Laravel, React Vite, MySQL), the strategy for testing the system's quality and security, and the ethical considerations governing the project.

Chapter Four outlines the **Project Plan**, including a preliminary estimation of the project budget and a high-level timeline and schedule for completing the various phases of the project development.

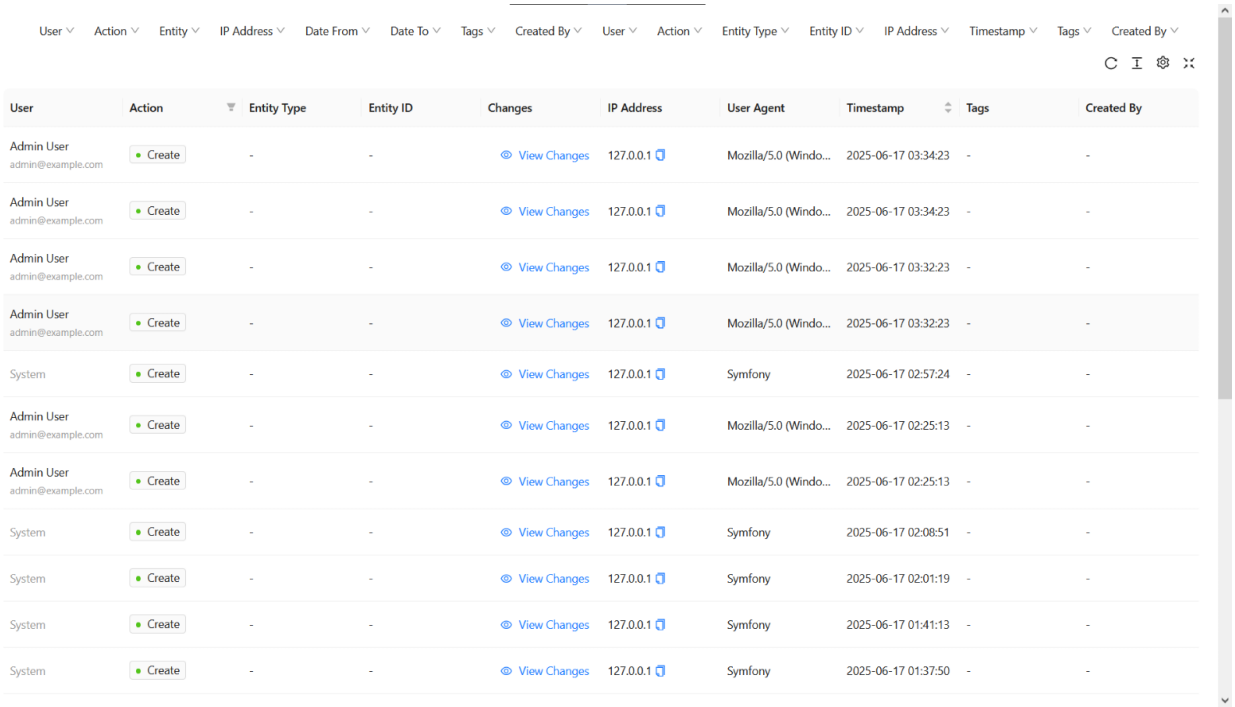
The document concludes with a list of academic **References** cited throughout the proposal and relevant **Appendices** containing supplementary materials such as data collection instruments, detailed design diagrams, and preliminary test checklists.

Chapter Two: Literature Review

2.1 Overview

This chapter delves into the existing body of knowledge surrounding Electronic Document Management Systems (EDMOS). The purpose is to understand the theoretical underpinnings, review successful implementations and available solutions, identify prevalent challenges, and ground the design and implementation approach of the proposed EDMOS for Gage University College within established principles and current practices. This review will cover fundamental EDMOS concepts, explore related technologies, analyze different system types, and examine their application, particularly within academic contexts.

FIGURE 1.3: Document Handle of EDMOS Project

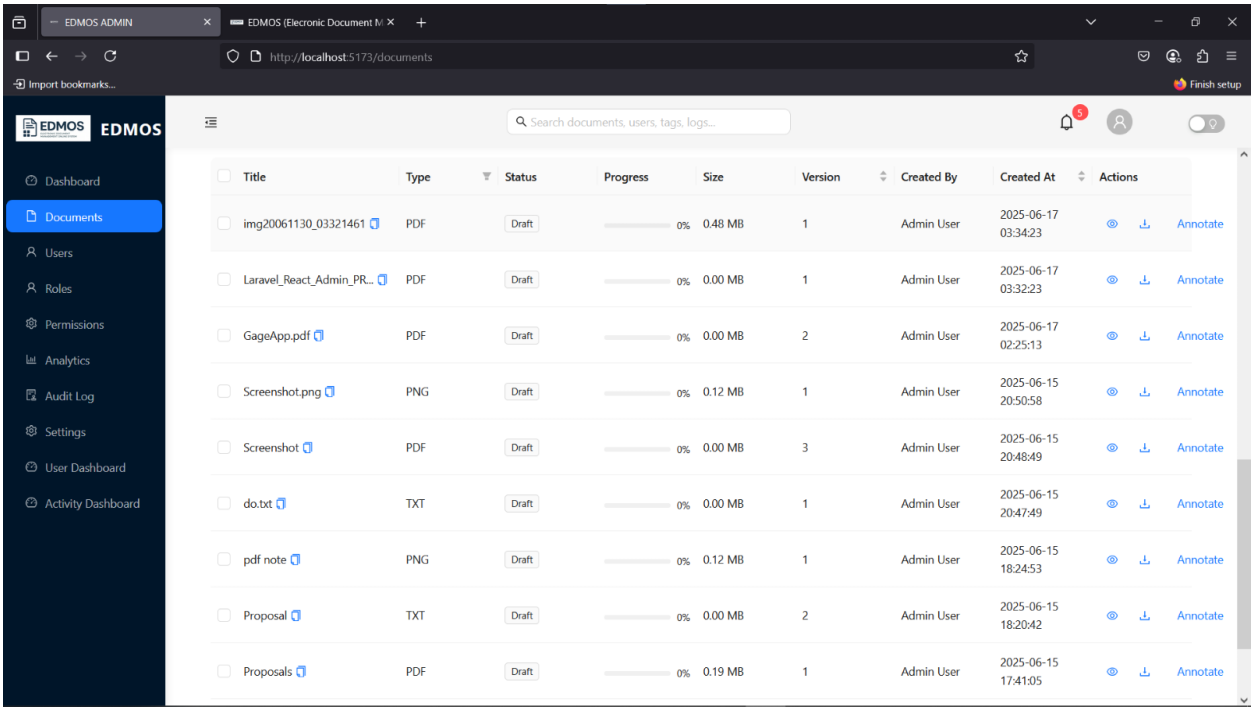


2.2 Core Concepts in Electronic Document Management

- **Definition and Purpose:** Define Electronic Document Management Systems (EDMOS) as systems designed to manage digital information assets. Discuss the primary purpose of EDMOS in facilitating the organization, storage, retrieval, security, and control of documents throughout their lifecycle, aiming to improve efficiency and reduce reliance on paper.

- **Document Lifecycle Management:** Detail the key stages:
    - *Document Capture:* Including methods for ingesting both born-electronic files and converting physical documents through scanning. Discuss considerations for scanning quality and format (e.g., PDF, TIFF).
    - *Indexing and Metadata:* Emphasize the importance of assigning descriptive information (metadata - like author, date, subject, department, document type, keywords, and crucially, for scanned documents, physical location identifiers) to documents for effective searching and organization. Differentiate between metadata indexing and full-text indexing. [Cite a source on metadata's role].
    - *Secure Storage:* Discuss requirements for secure, centralized digital repositories, including considerations for file system vs. database storage approaches and backup strategies.
    - *Version Control:* Explain how EDMOS tracks modifications, allowing users to revert to previous versions and understand the evolution of a document.
    - *Search and Retrieval:* Highlight the need for efficient search capabilities (both metadata-based and full-text search if applicable to document types) to quickly locate documents.
    - *Workflow and Routing:* Describe the function of automating document-centric processes and directing documents to relevant users or stages (e.g., for review or approval).
    - *Security and Access Control:* Discuss mechanisms for ensuring only authorized users can access, modify, or delete documents, including authentication (verifying identity) and authorization (defining permissions based on roles and departments). [Cite a security reference].
    - *Archiving and Retention:* Explain the process of managing inactive documents, potentially moving them to a separate archive and adhering to defined document retention policies.

FIGURE 1.4: Document Fields for a Large and Different Types of Document



2.3 Review of Related Systems and Solutions

Examining existing EDMOS solutions and related projects provides insights into best practices and informs the design choices for the Gage University College system.

- **Commercial Systems:** Proprietary EDMOS (e.g., SharePoint, Documentum) offer comprehensive features and vendor support. They typically target enterprise and large organizations. While feature-rich, they can be costly and may involve vendor lock-in.
- **Open-Source Systems:** Examples include Alfresco Community Edition, Nuxeo, LogicalDOC Community, among others. Open-source systems are cost-effective and flexible, with community-driven support, but they often require significant in-house expertise to install, configure, and maintain.
- In academic settings, EDMOS implementations face unique requirements: handling student records, research data, departmental diversity, and often limited IT resources. Studies (e.g., case studies of university EDMOS projects) highlight

challenges like aligning with academic workflows, ensuring user adoption, and integrating with existing campus systems.

- Other related systems include File Tracking Systems, Enterprise Content Management (ECM), and Web Content Management (WCMS). ECM is broader, encompassing various content types beyond documents, while WCMS focuses on web content. These systems share components with EDMOS (e.g., repositories, workflows) but serve overlapping needs.

FIGURE 1.5: Document Comparison of EDMOS Solution Types

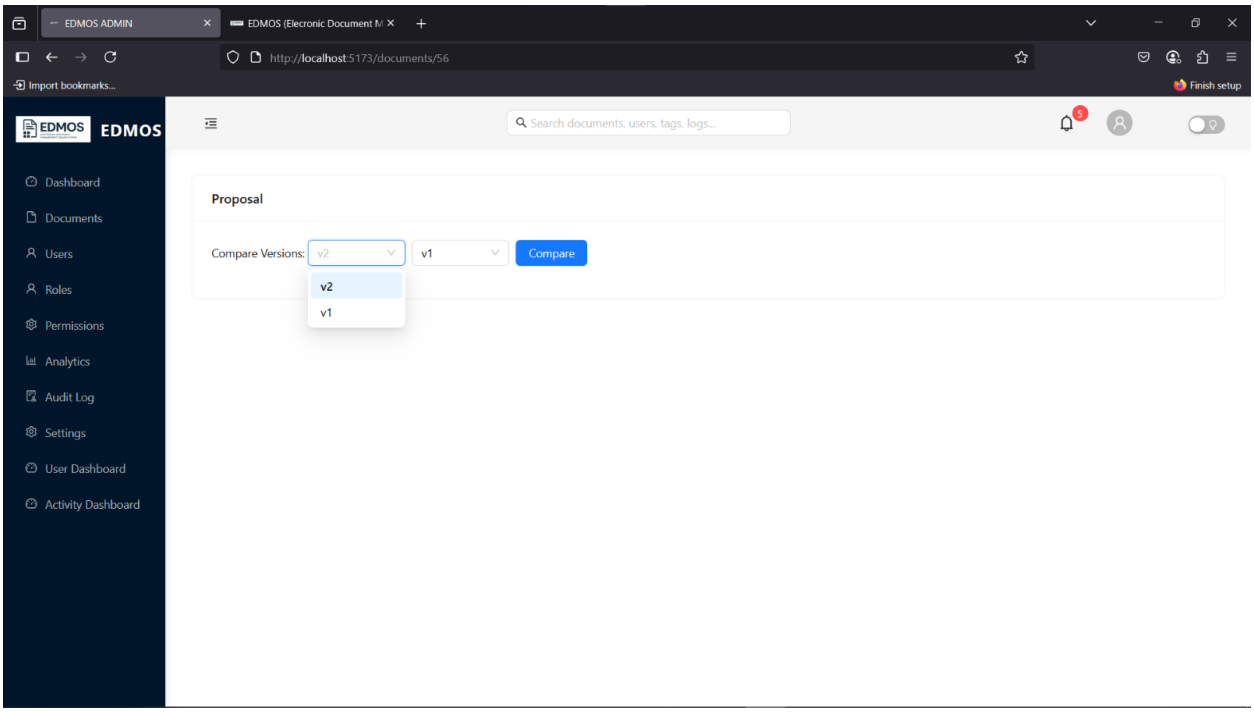
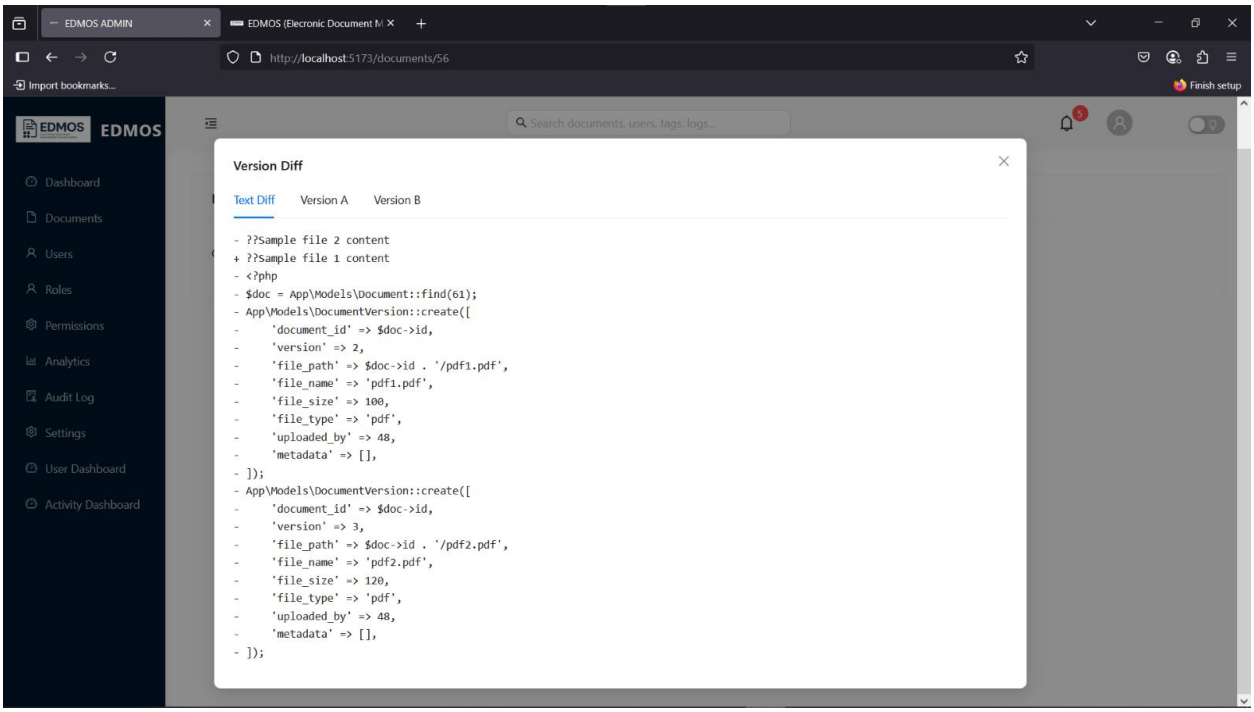


FIGURE 1.6: Document Comparison of EDMOS Solution Types



2.4 Challenges in EDMOS Implementation and the Path Forward

Implementing an EDMOS successfully requires navigating a range of technical, organizational, and human factors. Key challenges frequently cited in literature include:

- **Gathering Accurate and Complete Requirements:** Ensuring the system addresses the diverse and sometimes conflicting needs of various departments and users.
- **Data Migration Strategy:** Planning and executing the transfer of existing paper and digital documents, including data cleansing and indexing.
- **User Adoption and Training:** Overcoming resistance to change, ensuring users are adequately trained, and fostering a culture of digital document management.
- **Integration with Legacy Systems:** Connecting the new EDMOS with existing university information systems (e.g., student portals, HR systems) to avoid data silos.
- **Choosing the Right Technology and Architecture:** Selecting a technology stack that is robust, scalable, secure, and maintainable by the institution's IT resources.

- **Ensuring Data Security and Compliance:** Designing and implementing security measures that protect sensitive data and comply with privacy regulations.

Addressing these challenges requires a well-planned methodology, active stakeholder engagement, and a system architecture that is flexible and built on a solid technical foundation.

2.5 Identification of Gaps and Project Justification

The review of the current situation at Gage University College (Chapter 1) reveals a clear need for a modern document management solution. The literature review highlights that while various commercial and open-source EDMOS exist, selecting and implementing one that perfectly fits an institution's unique structure, workflows, and budget is complex.

Existing challenges at GUC include not only the general inefficiencies of manual systems but also the specific need to manage a mix of currently generated electronic documents and a substantial volume of historical scanned documents, alongside departmental variations in current practices. While off-the-shelf solutions offer broad functionality, they may be either prohibitively expensive (proprietary) or require significant technical expertise for configuration and customization to align with GUC's specific needs and existing structures (open source), potentially leading to a steeper learning curve for administrators and users, and uncertainty regarding long-term support for community editions.

This project is justified as it aims to fill these identified gaps by:

- **Developing a Custom-Fit Solution:** Designing the EDMOS based on the specific, analyzed requirements of Gage University College departments, ensuring core workflows and document types are directly supported.

- **Leveraging a Modern, Flexible Stack:** Utilizing **Laravel** for a structured, scalable, and secure backend API and **React (Vite)** for a dynamic, responsive, and user-friendly frontend interface (Admin, User, and Landing pages). This architecture promotes maintainability and allows for tailored UI/UX.

- **Explicitly Supporting Scanned Documents:** The design will specifically account for incorporating scanned documents, ensuring they can be indexed with relevant metadata (including physical location) and managed alongside electronic files.

- **Focusing on Core GUC Needs:** Prioritizing the implementation of essential EDMOS features (upload, indexing, search, access control, versioning, archiving, basic workflow messaging) deemed most critical by potential GUC users, rather than implementing a vast, potentially underutilized feature set from a generic system.

- **Providing a Clear and Understandable System:** A custom-built system can be designed with a user interface and workflow logic that is intuitive for GUC staff, reducing the training burden.

By developing this EDMOS prototype from scratch using the specified technology stack and an agile approach focused on GUC's requirements, the project offers a tailored, cost-effective (in terms of software licensing), and maintainable solution that directly addresses the college's document management challenges and provides a clear path for potential future expansion.

Chapter Three: Methodology

3.1 Introduction

This chapter outlines the planned approach for executing the Electronic Document Management System project for Gage University College. It details the chosen software development methodology, the strategies for gathering and analyzing requirements, the design techniques employed, the specific technology stack selected for implementation, the plan for testing the system's quality and adherence to requirements, and the ethical considerations guiding the project's execution. This methodology aims to ensure a systematic, efficient, and user-centric development process.

3.2 Development Methodology (Scrum Framework)

The project will be managed and executed using the **Scrum Framework**. Scrum is an iterative and incremental agile methodology suitable for projects where requirements may evolve and close collaboration with stakeholders is beneficial. The project lifecycle will be structured into several time-boxed iterations called sprints, typically lasting (2–4 weeks each). Key elements include:

- **Scrum Roles:** The project will involve the core Scrum roles:
    - *Product Owner:* Represented by the project supervisor and potentially key GUC staff, responsible for defining and prioritizing the product backlog items.
    - *Development Team:* The student researcher, responsible for designing, implementing, and testing the system increments.
    - *Scrum Master:* Could be the student researcher or supervisor, facilitating the process and removing impediments.
- **Scrum Events:** Key Scrum events will include:
    - *Sprint Planning:* At the start of each sprint, to select items from the product backlog and plan how to complete them.
    - *Sprint Review:* At the end of each sprint, to inspect the increment and adapt the product backlog if needed, with stakeholder feedback.
    - *Sprint Retrospective:* After the sprint review, to reflect on the sprint and identify improvements for the next sprint.
- **Scrum Artifacts:** Key artifacts will include:
    - *Product Backlog:* A prioritized list of features, functions, requirements, and fixes for the system.

- *Sprint Backlog:* The subset of product backlog items selected for a specific sprint.
- *Increment:* The sum of all product backlog items completed during a sprint and the value of the previous increments, resulting in a potentially shippable product.

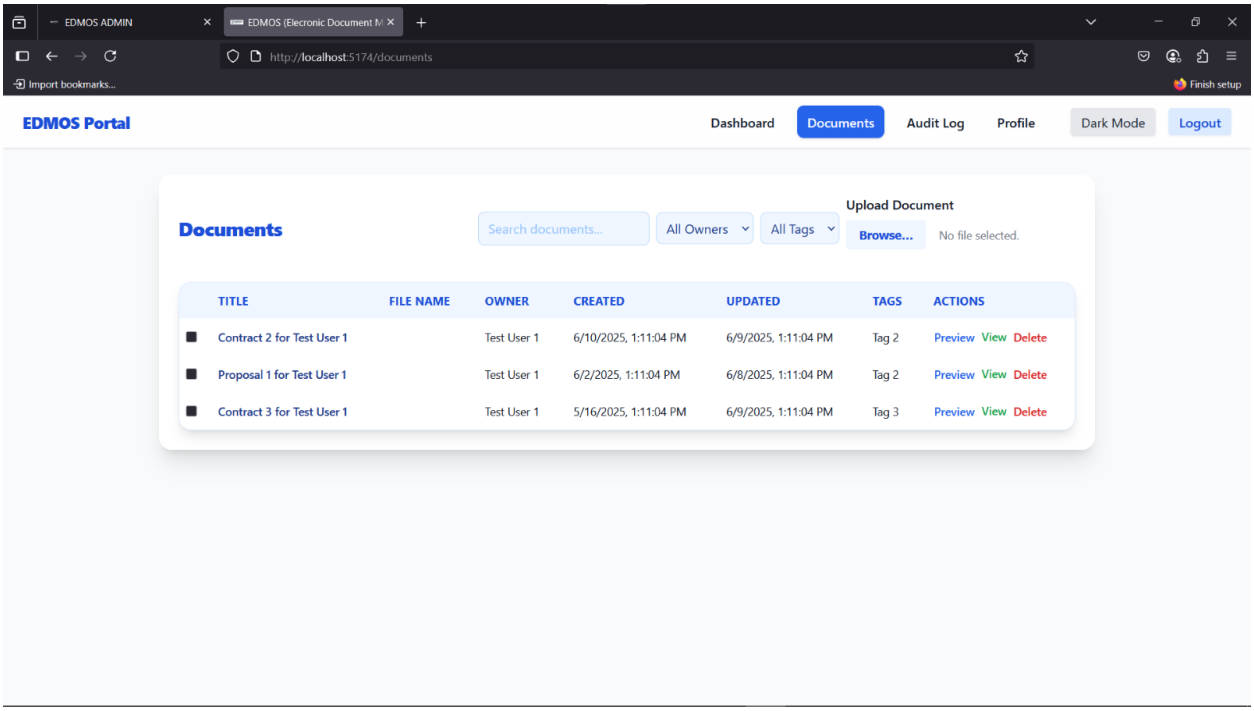
Scrum's iterative nature allows incorporation of feedback from GUC users as early versions of the EDMOS are developed, reducing risk and ensuring alignment with actual needs.

3.3 Data Collection Methods

To ensure the EDMOS meets the genuine needs of Gage University College, a mixed-methods approach combining qualitative techniques will be used to gather requirements:

- **Semi-structured Interviews:** Conduct interviews with key personnel from various administrative departments (e.g., Registrar, Finance, HR, Academic Offices) identified as heavy users of documents. This approach allows for focused exploration of predefined areas (current processes, challenges, desired features) while maintaining flexibility to delve deeper into specific points raised by the interviewees [Cite a source on qualitative interviews].
- **Observation:** Observe the current manual workflows related to document creation, handling, storage, and retrieval within selected offices. This provides practical context and helps uncover tacit knowledge or steps not explicitly mentioned in interviews. Observations will focus on document types, volume, flow, bottlenecks, and storage methods [Cite a source on observational data collection].
- **Document and Form Analysis:** Review samples of important documents used at Gage University College and any existing manual forms, logs, or registers (e.g., physical file indexes, incoming/outgoing mail logs, internal routing slips). This helps in understanding the types of information recorded, potential metadata fields, and existing classification attempts.

FIGURE 1.7: Data Collection Process

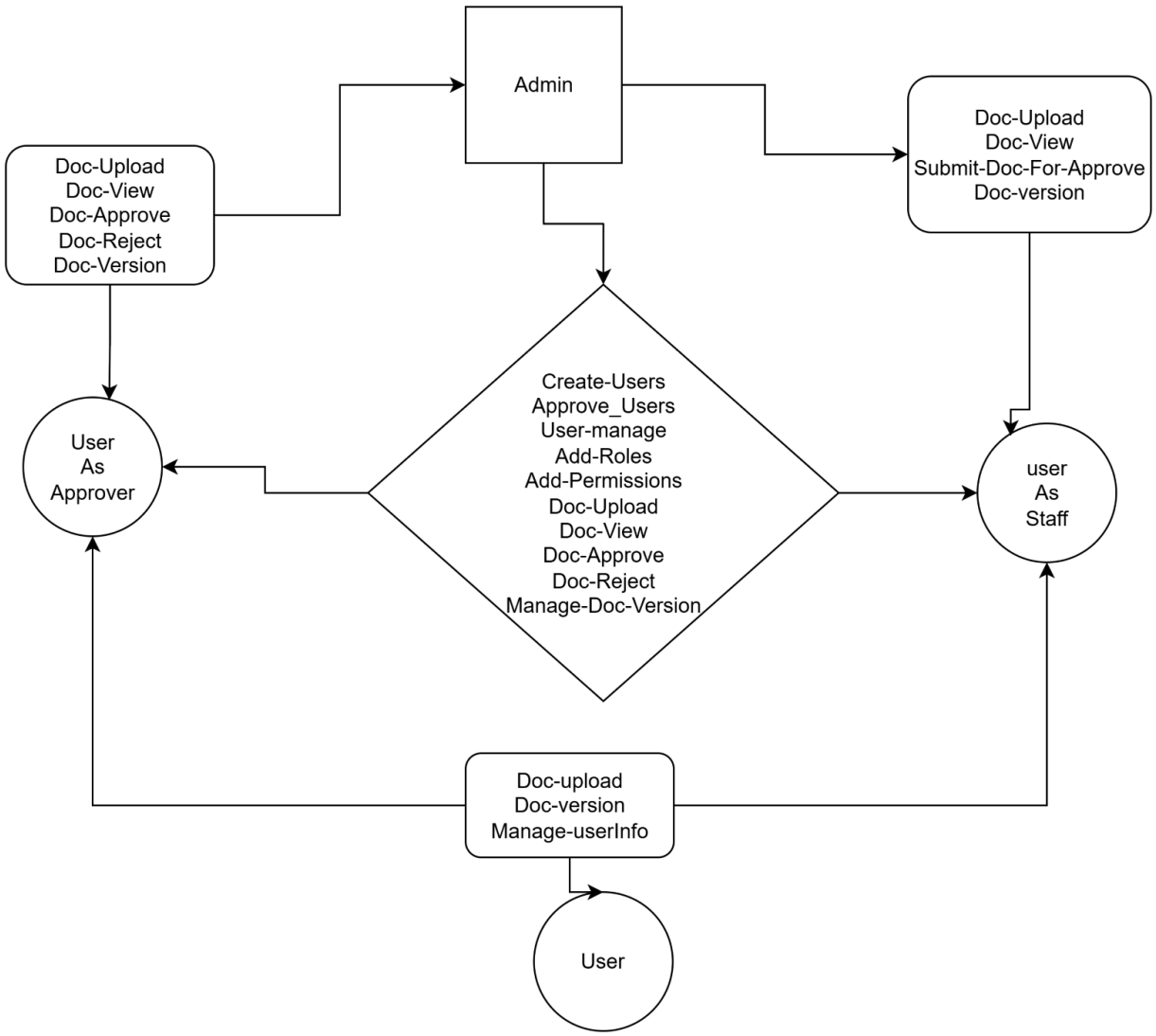


3.4 System Design and Modeling (UML)

The design of the EDMOS will be systematically documented using the Unified Modeling Language (UML) to provide clear blueprints for implementation and communication among stakeholders.

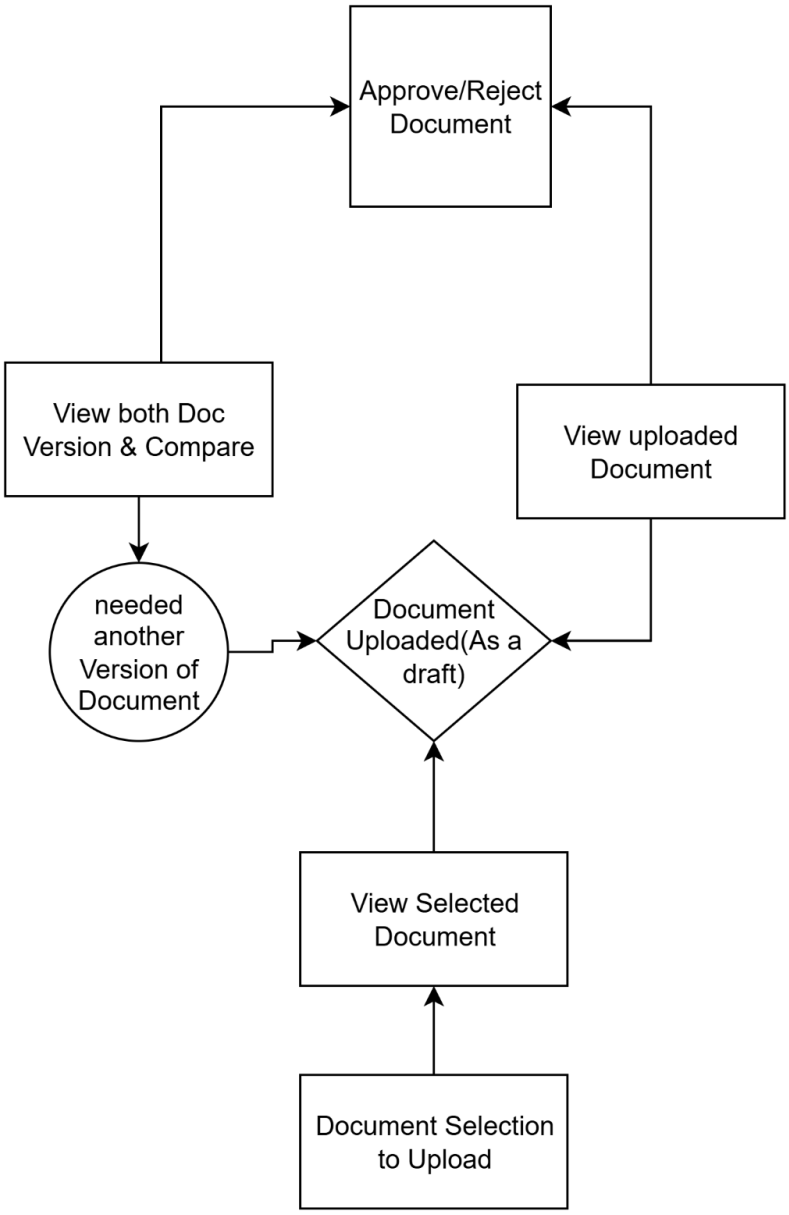
- **Use Case Diagram:** To model the functional scope of the system, illustrating the interactions between the primary actors (Admin, User) and the system's key functionalities (use cases). This diagram will provide a high-level view of what the system will do for different user types.

Figure G.1: High-Level EDMOS Use Case Diagram



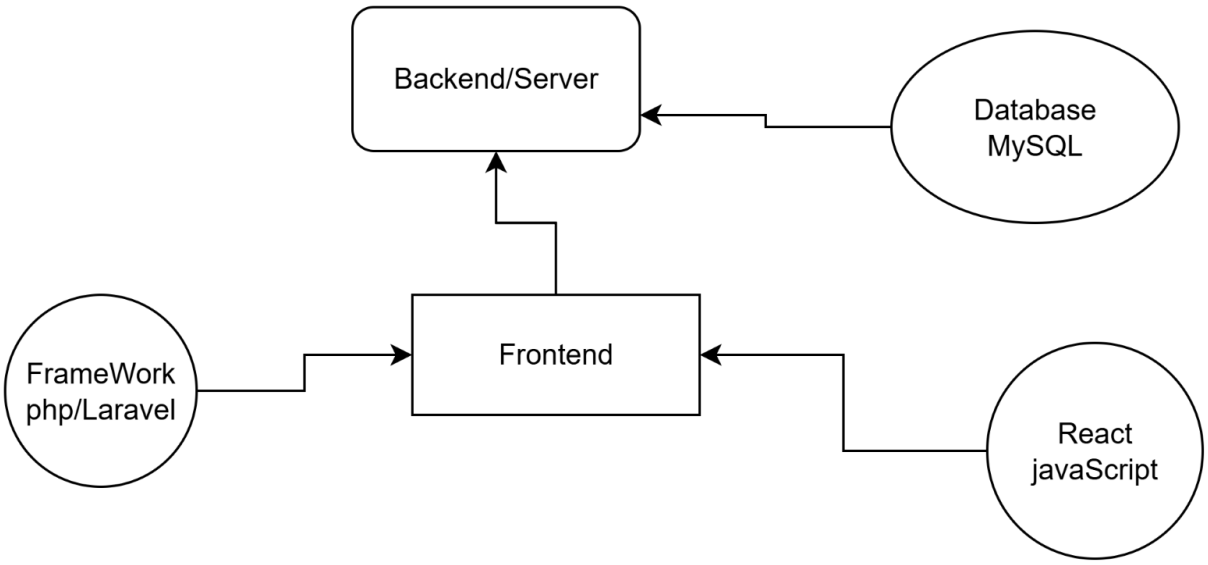
- **Activity Diagrams:** To model the dynamic flow of activities for specific processes within the system, such as the document upload process, the search and retrieval process, or the user registration process.

Figure G.2: Activity Diagram for Document Retrieval



- **Deployment Diagram:** To model the physical deployment of the software components on hardware nodes, showing where the frontend application runs (client browsers), where the backend application runs (web server), and where the database resides (database server).

Figure G.3: Deployment Diagram of the EDMOS



3.5 Implementation Technologies and Environment

The EDMOS will be implemented following a client-server architecture with a distinct frontend and backend, communicating via a RESTful API.

- **Frontend Implementation (React with Vite):**

  The user interface is built as a single-page application using React (with TypeScript). Vite serves as the build tool for fast development and hot module replacement. Tailwind CSS is used for responsive, utility-first styling. React Router manages navigation between pages. Libraries like @tanstack/react-query handle data fetching and caching. The frontend implements features such as document lists with filtering/sorting, bulk actions (delete, export CSV), document upload forms,

preview modals, status indicators, and dark mode. Client-side logic is handled with JavaScript/TypeScript, and unit tests are written with Jest.

- **Technology:** React will be used to build the user interface components. Vite will serve as the build tool, providing fast development server startup and hot module replacement. This allows for building a dynamic and responsive user experience across the different portals (Landing, User, Admin).
- **Styling:** CSS with a framework like Bootstrap will be used to ensure responsive design and a consistent visual appearance.
- **Interaction:** JavaScript will handle client-side logic and interactions.

- **Backend Implementation (Laravel):**

  Laravel (PHP) is used for the server-side application, providing routing, controllers, middleware, and Eloquent ORM for database access. Key backend features include RESTful API endpoints for document CRUD, user and role management, and system settings. Authentication and API token management are handled via Laravel Sanctum, and roles/departments logic is integrated for access control. Laravel Scout is used to provide full-text search capabilities. Maatwebsite's Excel package allows importing/exporting data in spreadsheets. Pest (PHP testing framework) is used for backend unit and feature tests.

  - **Technology:** Laravel, a powerful PHP framework, will be used to build the backend application and expose RESTful API endpoints. Laravel provides features for routing, authentication, authorization (including integrating role/department logic), data validation, and Eloquent ORM for simplified database interactions.
  - **Functionality:** The backend will handle core business logic, processing requests from the frontend, interacting with the database, managing file storage, and implementing security checks.

- **Database (MySQL):**

  A MySQL relational database stores structured data: user accounts, roles, departments, document metadata, folder structures, version histories, messages,

and audit logs. The frontend and backend communicate over HTTP/JSON with RESTful endpoints defined in Laravel.

    - **Technology:** MySQL will be used as the relational database management system to store all structured data, including user information, roles, departments, document metadata, folder structures, version history logs, message data, and audit trails.
- **API Communication:** The frontend and backend will communicate via HTTP requests to RESTful API endpoints defined in the Laravel application. This ensures a clear separation of concerns and modularity.
- **Development Environment:** A local environment will be set up (e.g., XAMPP or Docker) to run the Laravel PHP server and MySQL database. Node.js and npm/yarn are used to manage the React development environment. Version control is managed with Git.

3.6 Testing Strategy

A rigorous testing strategy will be implemented to ensure the quality, reliability, usability, and security of the EDMOS.

- **Unit Testing:** Individual functions, methods, and components in both the Laravel backend (using PHPUnit) and React frontend will be tested in isolation to verify their correct behavior.
- **Integration Testing:** Testing the interactions between different modules, particularly focusing on the communication flow between the React frontend and the Laravel backend API, and between the backend logic and the MySQL database.
- **System Testing:** Testing the fully integrated EDMOS against the specified functional and non-functional requirements (performance, reliability, scalability - within project scope) to verify the system as a whole performs as expected in a simulated production environment.
- **Security Testing:** Conducting tests to identify potential vulnerabilities, including testing authentication and authorization mechanisms, input validation, data sanitization, and potential exposure to common web attacks (e.g., SQL injection, XSS). Leveraging Laravel's security features and performing manual security checks.
- **Usability Testing:** Evaluating the ease of use and intuitiveness of the user interfaces (Landing Page, User Portal, Admin Portal) with potential end-users from

Gage University College. This will involve observing users performing typical tasks and collecting their feedback.

- **User Acceptance Testing (UAT):** Key GUC staff members representing different user roles and departments will be involved in UAT to validate that the system meets their practical needs and workflows in a realistic scenario. Predefined test cases and checklists derived from the requirements will guide these sessions.

3.7 Ethical Considerations

The project is committed to conducting research and development ethically, particularly concerning data handling and engagement with Gage University College staff.

- **Data Privacy and Confidentiality:** All information gathered during requirements analysis, especially details about existing processes and challenges, will be treated with the utmost confidentiality. Any use of realistic or sensitive data for testing purposes will involve anonymization or the creation of synthetic data sets. The implemented system will prioritize data privacy and security features.
- **Informed Consent:** All Gage University College staff members participating in interviews, observations, or user acceptance testing will be fully informed about the project's purpose, how their contributions will be used (anonymously in reporting), and their right to withdraw their participation at any time without penalty. Written or recorded consent will be obtained where appropriate, following GUC guidelines.
- **Transparency:** The nature and scope of the project, including the technology used and how data will be handled within the developed system, will be communicated clearly and transparently to relevant stakeholders.

A formal ethical clearance request will be submitted to the relevant academic or administrative body at Gage University College if required, prior to commencing data collection involving human participants.

Chapter FOUR: Project Plan (Budget & Schedule)

4.1 Project Budget

The estimated budget focuses on direct project costs. Many resources are provided by the university or are open-source, minimizing cash outlay. Key estimates (in Ethiopian Birr, ETB) are:

- **Hardware:** ETB 2,000–5,000 for development machine maintenance or shared server costs.
- **Software:** ETB 0 (all development tools and frameworks are open-source).
- **Internet/Communication:** ETB 2,000–4,000 for additional connectivity needs (e.g., collaboration, deployment testing).
- **Stationery/Printing:** ETB 1,000–2,000 for documentation, proposal copies, and forms.

The **total estimated cost** is approximately ETB 5,000–7,000. These costs mainly cover operational needs. University contributions include supervisor time (no salary cost) and existing infrastructure (labs, networks, servers).

4.2 Project Schedule

The project timeline is organized into phases, totaling (~1–3 months). This outlines the high-level phases and duration:

- **Initiation & Planning:** Finalize proposal, complete literature review, and plan the project.
- **Foundation:** Set up development environment, finalize initial requirements, design core database schema, establish basic UI layout, and implement user authentication/login.
- **Document Core:** Implement document upload, metadata indexing, secure storage, and basic document viewing/downloading.
- **Search & Access:** Develop advanced search by metadata, enforce role-based access control, and build administrative user management interfaces.
- **Workflow Basics:** Add simple document routing/notification mechanisms and basic audit trail logging.

- **Refinement & UAT:** Address feedback from testing, refine UI/UX, and conduct User Acceptance Testing with GUC staff/students.
- **Testing & Polish:** Perform comprehensive system and security testing, tune performance, finalize documentation, and prepare for deployment.
- **Reporting & Submission:** Complete final report/thesis writing and prepare for the final presentation.

The total is ~12 weeks. This schedule is illustrative; actual progress and stakeholder feedback may adjust sprint content and duration as per Agile principles.

References

This project has been developed utilizing a variety of resources, including official documentation, open-source contributions, and general internet resources to ensure robust and well-informed development practices.

Official Documentation

- **Laravel Documentation:** The official source for comprehensive information on the Laravel PHP framework, covering its architecture, features, and best practices.
    - Link: https://laravel.com/docs
- **React Documentation:** The primary resource for learning and understanding the React JavaScript library for building user interfaces, including hooks, component lifecycle, and state management.
    - Link: https://react.dev/
- **Ant Design Documentation:** The official guide for using Ant Design, a design system and React UI library, providing components, design principles, and usage guidelines.
    - Link: https://ant.design/docs/react/introduce
- **Vite Documentation:** The official documentation for Vite, a next-generation frontend tooling that provides a fast development experience with features like instant server start and hot module replacement.
    - Link: https://vitejs.dev/guide/

Development Resources

- **GitHub:** Various open-source repositories and projects hosted on GitHub were consulted for code examples, best practices, and community-contributed solutions.
    - Link: https://github.com/
- **Open Source Contributions:** Insights and solutions from the broader open-source community, including various libraries, packages, and public code snippets, were leveraged to enhance development efficiency and quality.
- **Internet Resources:** General web searches, technical blogs, forums (e.g., Stack Overflow), and online tutorials were utilized for problem-solving, understanding concepts, and exploring alternative implementations.