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 - May 13, 2025

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
Department of Computer Science, Gage University College

Date: May 13, 2025
Addis Ababa, Ethiopia

Abstract

The digital age necessitates efficient information management, yet many educational institutions like Gage University College continue to heavily rely on manual document handling processes. This traditional approach is plagued by inherent challenges, including significant time expenditure for retrieval, susceptibility to loss or damage from environmental factors or human error, inefficient use of physical space, version control complexities, and security vulnerabilities. These issues collectively hinder productivity, impede timely decision-making, and pose risks to the integrity and accessibility of crucial university records.

This proposal outlines a project aimed at the design and implementation of a new Electronic Document Management System (EDMS) specifically tailored for Gage University College. The system will provide a centralized, secure, and intuitive platform for managing digital documents throughout their lifecycle, from creation and storage to retrieval, sharing, and archiving. A key focus will be placed on enhancing collaboration among university staff and improving the overall user experience. The project will adopt an Agile (Scrum) development methodology to allow for iterative development, flexibility in incorporating feedback, and rapid delivery of working software aligned with user needs. Data collection will involve semi-structured interviews, process observation, and review of existing document structures. The system will be designed using standard modeling techniques and implemented using a robust web technology stack, focusing on usability, security, and maintainability. The expected outcome is a functional EDMS that significantly improves operational efficiency, safeguards university records, and fosters a more collaborative working environment.

Keywords: Electronic Document Management, Information Systems, Agile Development, Collaboration, University Administration, Digital Archiving, Workflow Automation.

Acknowledgements

I would like to express my sincere gratitude to Gage University College for providing the opportunity to undertake this project and for allowing access to relevant information and personnel. My deepest appreciation goes to my supervisor ZELALEM, for their invaluable guidance, constructive feedback, and unwavering support throughout the proposal development process. Their expertise has been instrumental in shaping the direction and scope of this research.

I am also thankful to the dedicated staff members across various departments at Gage University College who generously shared their time, insights into the current document management practices, and functional requirements. Their contributions were crucial for understanding the practical challenges and needs that the proposed system aims to address.

Finally, I extend my thanks to my family and friends for their continuous encouragement and moral support during this academic endeavor.

Table of Contents

[This section will be auto-generated upon completion of the full proposal draft, listing all chapters, sections, and page numbers.]

List of Tables
List of Figures
List of Acronyms and Abbreviations

List of Tables

[This section will list table numbers, titles, and page numbers as they are added to the full proposal.]

List of Figures

[This section will list figure numbers, titles, and page numbers as they are added to the full proposal.]

List of Acronyms and Abbreviations

[This section will list acronyms and abbreviations used in the proposal and their full meanings.]

EDMS - Electronic Document Management System
GUC - Gage University College
UML - Unified Modeling Language
SDLC - System Development Life Cycle
SME - Small and Medium-sized Enterprises
API - Application Programming Interface
UI/UX - User Interface/User Experience
... [Add others as needed]

Chapter One: Introduction

1.1 Background

In the current educational landscape, universities generate, receive, and process vast volumes of documents daily. These documents, ranging from administrative records and student files to research papers and policy documents, are the lifeblood of institutional operations and knowledge transfer. Historically, the management of these documents has been primarily manual, involving physical filing, paper-based workflows, and localized storage solutions. While functional to some extent, this manual paradigm is increasingly inadequate to meet the demands of modern university administration, which requires rapid access to information, seamless collaboration, robust security, and efficient resource utilization. The concept of a "paperless office" and the adoption of digital document management practices have emerged as critical strategies to overcome the limitations of manual systems and enhance organizational effectiveness. Electronic Document Management Systems (EDMS) are central to this shift, providing the technological infrastructure to capture, organize, store, retrieve, and control documents electronically.

Gage University College, as a prominent higher learning institution in Ethiopia, also faces the challenges associated with traditional, paper-based document management. The current reliance on manual processes across its various faculties and administrative offices results in bottlenecks, potential loss of critical information, and significant time investment in routine document tasks. Implementing a well-designed EDMS is therefore

not merely a technological upgrade but a strategic imperative to modernize operations, improve efficiency, and better serve its students, faculty, and staff.

1.2 Statement of the Problem

The continued reliance on manual document management practices at Gage University College presents several critical problems that impede operational efficiency and pose risks to institutional knowledge. Key issues identified through preliminary observations and discussions include:

- **Difficulty and Delay in Information Retrieval:** Locating specific documents within physical archives is time-consuming and labor-intensive, leading to delays in administrative processes and decision-making.
- **Risk of Loss, Damage, and Deterioration:** Paper documents are vulnerable to physical damage from handling, environmental factors (fire, flood, pests), and simply being misplaced or misfiled. This poses a significant risk to irretrievable historical or legal records.
- **Inefficient Space Utilization:** Maintaining large volumes of paper documents requires extensive physical storage space, which could be utilized more effectively for other university activities.
- **Lack of Centralization and Version Control:** Documents often exist in multiple copies across different offices, leading to inconsistencies, difficulties in identifying the most current version, and a fragmented view of institutional information.
- **Security and Access Control Limitations:** Physical documents are susceptible to unauthorized access or modification. Implementing granular access permissions is challenging, potentially compromising confidentiality and integrity.
- **Inefficient Workflow and Collaboration:** Routing documents manually through approval processes is slow and lacks transparency. Collaboration on documents requires physical transfer or unreliable digital copies, hindering teamwork.

These problems collectively contribute to reduced staff productivity, increased operational costs, and a potential negative impact on the quality of administrative services provided by the university. Addressing these challenges through a dedicated EDMS is essential.

1.3 Objectives

The primary aim of this project is to leverage modern information system principles and agile development practices to create a functional and user-friendly Electronic Document

Management System that effectively addresses the document management challenges faced by Gage University College.

1.3.1 General Objective

To design, develop, and test a web-based Electronic Document Management System for Gage University College to automate document handling processes and improve overall administrative efficiency.

1.3.2 Specific Objectives

1. To conduct a thorough analysis of the current manual document management system and identify detailed functional and non-functional requirements for the new EDMS.
2. To design the system architecture, database schema, and user interface components based on the gathered requirements.
3. To implement the EDMS using a modern, secure, and maintainable web development framework and associated technologies.
4. To integrate core EDMS functionalities, including secure document upload, indexing with relevant metadata, powerful search and retrieval, access control based on user roles and departments, and basic document workflow capabilities.
5. To conduct comprehensive testing (unit, integration, system, and user acceptance testing) to ensure the system is robust, secure, usable, and meets the defined requirements.
6. To provide recommendations for future system enhancements and a potential deployment strategy.

1.4 Scope of the Project

The scope of this project is focused on the design and implementation of a core web-based EDMS for Gage University College suitable for an M.Sc. project. The system will cover the following key areas:

- **Document Lifecycle Management:** Support for uploading, indexing, categorizing, storing, searching, retrieving, and managing electronic documents.
- **User and Access Management:** Implementation of user authentication, role-based access control, and departmental permissions to ensure document security and confidentiality.
- **Basic Workflow:** Support for simple document routing and notification within predefined user groups or departments.

- **Metadata Management:** Ability to associate custom metadata fields with documents for enhanced organization and searchability.
- **Centralized Repository:** Secure storage of all electronic documents in a central, organized database.
- **Audit Trails:** Logging of significant document actions (creation, modification, deletion, access) for accountability.

The following functionalities are **outside the primary scope** of this project due to time and complexity constraints typical of a graduate project:

- Integration with existing legacy systems (beyond basic user authentication if possible).
- Advanced workflow automation engines.
- Optical Character Recognition (OCR) for scanned document content indexing (metadata indexing will be the focus).
- Complex version control beyond tracking major revisions.
- Large-scale data migration from existing physical or fragmented digital stores.
- Mobile *native* application development (a responsive web design will be targeted).
- Integration with third-party cloud storage services for archival (though design considerations may be discussed).

1.5 Significance of the Project

The successful design and implementation of this EDMS project hold significant benefits for various stakeholders within Gage University College:

- **For the University Administration:** Improved operational efficiency through automated processes, reduced physical storage needs, enhanced data security and compliance, faster decision-making facilitated by quick information access, and better resource allocation.
- **For Faculty and Staff:** Significant time savings in searching for and retrieving documents, streamlined administrative workflows, improved collaboration through easy sharing, reduced burden of manual filing and tracking, and a more organized digital workspace.
- **For Students:** Potential for faster processing of administrative requests (once integrated with relevant student services) and improved access to official university documents or policies (depending on access levels).

- **Institutional Benefits:** Enhanced institutional memory and knowledge retention, improved disaster recovery capabilities for vital records, and a step towards a more modern and technologically advanced organizational image.

Ultimately, the project contributes to creating a more agile, efficient, and secure information environment for the entire university community.

1.6 Organization of the Document

This proposal is structured into chapters, following a standard format for academic project proposals. Chapter One provides the introduction, outlining the project background, problem statement, objectives, scope, and significance. Chapter Two presents a review of relevant literature on document management systems, related technologies, and challenges, culminating in the identification of gaps and justification for the proposed approach. Chapter Three details the proposed methodology for designing and implementing the system, including the chosen development model, data collection techniques, development tools, and testing strategy. Chapter Four focuses on the project plan, including an estimated budget and a preliminary schedule. The proposal concludes with a list of references and appendices.

Chapter Two: Literature Review

2.1 Overview

This chapter provides a synthesis of existing literature pertinent to Electronic Document Management Systems. It aims to establish a theoretical foundation for the proposed system, review best practices, identify common challenges in implementation, and position the current project within the broader context of information management research and application in educational settings. The review covers core EDMS concepts, examines various related technologies and existing solutions, and highlights research findings relevant to system design, development, and adoption.

2.2 Core Concepts in Document Management

- **Definition of Document Management System (DMS) / Electronic Document Management System (EDMS):** A system (software and hardware) used to store, manage, and track electronic documents and electronic images of paper-based

information [Ref 1: e.g., cite a standard DMS book/article]. EDMS encompasses the entire document lifecycle from creation to archival or disposal.

- **Distinction between DMS, ECM, and WCMS:** While related, it's important to clarify these terms. DMS primarily focuses on managing *unstructured documents*. Enterprise Content Management (ECM) is a broader strategy encompassing various content types (documents, web content, rich media) and processes (capture, manage, store, preserve, deliver). Web Content Management Systems (WCMS) are specifically designed for managing content on websites [Ref 2: e.g., cite a source explaining these differences]. This project focuses on EDMS, a subset often integrated into broader ECM strategies.
- **Key EDMS Components and Features:** Discuss essential features found in EDMS literature:
    - *Document Capture:* Converting paper documents to electronic format (scanning).
    - *Indexing and Metadata:* Assigning keywords or attributes (metadata) to documents for easy organization and retrieval [Ref 3: cite an indexing source].
    - *Storage and Archiving:* Securely storing active and inactive documents, potentially with defined retention policies [Ref 4: cite an archiving source].
    - *Search and Retrieval:* Enabling users to quickly find documents based on content or metadata [Ref 5: cite a retrieval source].
    - *Version Control:* Managing multiple iterations of a document, tracking changes and authorship [Ref 6: cite a version control source].
    - *Workflow:* Automating document-centric processes and routing documents between users or stages [Ref 7: cite a workflow source].
    - *Security and Access Control:* Implementing authentication, authorization, and auditing to protect sensitive information [Ref 8: cite a security source].
    - *Collaboration Features:* Tools for sharing, commenting, or co-authoring documents.

2.3 Review of Related Systems and Solutions

This section briefly reviews existing EDMS examples or related projects, categorizing them into proprietary and open-source solutions, and discusses projects implemented in similar contexts.

- **Proprietary EDMS Examples:** Mention prominent commercial systems (e.g., SharePoint, Documentum, Laserfiche - if aware, otherwise generic "commercial

EDMS"). Discuss their typical strengths (comprehensive features, professional support) and weaknesses (high cost, potential vendor lock-in, rigidity).

- **Open Source EDMS Examples:** Discuss well-known open-source options (e.g., Alfresco Community, Nuxeo Platform, LogicalDOC Community - drawing from sample's list but rephrasing descriptions). Highlight benefits (cost-effectiveness, flexibility, community support) and potential drawbacks (complexity in setup/configuration, variable documentation quality, reliance on community for support, need for internal expertise) [Ref 9: cite an open source review].
- **Related Projects in Educational Settings:** Review the sample's related works (e.g., File Tracking System at Epoka University, DMS at Cochin University, EDTS at MARA University, DMS towards Paperless Office) and other relevant studies on EDMS implementation in universities or public sectors [Ref 10: cite relevant paper]. Analyze their approaches, implemented features, and reported outcomes or challenges.

2.4 Challenges in EDMS Adoption and Implementation

Implementing an EDMS, particularly in a university setting with diverse users and departmental needs, is not without challenges. Literature identifies several common hurdles:

- **User Adoption and Training:** Resistance to change, lack of digital literacy among some staff, and insufficient training are major barriers [Ref 11: cite user adoption study].
- **Data Migration:** Converting and transferring large volumes of existing paper and electronic documents is a complex, time-consuming, and often costly process [Ref 12: cite data migration study].
- **Integration with Existing Systems:** Compatibility issues with other university systems (HR, student information, finance) can be problematic.
- **Security and Compliance:** Ensuring the system meets privacy regulations and institutional security policies is paramount and requires careful planning.
- **Customization vs. Off-the-Shelf:** Choosing between a ready-made solution and a custom build involves trade-offs related to cost, flexibility, and complexity.

2.5 Identification of Gaps and Project Justification

Based on the literature review and the identified problems at Gage University College, a clear gap exists. While commercial and open-source EDMS solutions offer a wide range of features, they often come with high costs, complex implementations requiring significant

customization or compromise on specific institutional needs, or challenges related to support and future development uncertainty, particularly with community-driven open source projects.

The manual system at GUC, as discussed in Chapter 1, is demonstrably inefficient and insecure. A generic off-the-shelf solution may not perfectly align with the university's specific workflows and existing departmental structures.

This project proposes to fill this gap by designing and implementing a custom EDMS tailored to the *specific* requirements and context of Gage University College. By developing the system from scratch (using a modern framework), this project aims to:

- **Directly Address GUC's Pain Points:** Focus on the core features needed most urgently by the university staff (efficient retrieval, secure storage, controlled access, basic workflow).
- **Ensure User-Centric Design:** Develop an intuitive interface that minimizes the learning curve and encourages high user adoption.
- **Provide a Flexible and Maintainable Foundation:** Build the system using a well-supported, modern framework (like Laravel) known for its structure and maintainability, making future enhancements and integrations easier.
- **Enable Tailored Security:** Implement security measures precisely aligned with the university's policies and data sensitivity requirements.

The proposed custom solution, guided by an agile development process, offers the flexibility to adapt to specific GUC needs and deliver a system that is not only functional but also practical and easily manageable by the university's IT resources in the long term, overcoming some of the typical challenges associated with adopting complex, generic solutions.

Chapter Three: Methodology

3.1 Introduction

This chapter describes the systematic approach that will be employed to design, develop, and evaluate the Electronic Document Management System for Gage University College. It covers the chosen software development methodology, the methods used for collecting data and specifying requirements, the tools and technologies selected for implementation,

and the strategy for testing and ensuring the quality of the developed system. Ethical considerations regarding data handling and participant involvement are also discussed.

3.2 Development Methodology

For this project, the **Scrum framework**, a popular Agile methodology, has been selected. Scrum is an iterative and incremental approach that emphasizes flexibility, collaboration, and delivering value in short cycles called sprints.

- **Justification for Choosing Scrum:** Scrum is particularly suitable for this project because:
    - **Requirement Flexibility:** Document management needs can evolve or become clearer as development progresses. Scrum's iterative nature allows for requirements refinement and adaptation based on feedback from potential GUC users.
    - **Stakeholder Collaboration:** Scrum promotes close collaboration with stakeholders (university staff) through regular meetings (sprint reviews), ensuring the system meets their actual needs and facilitating user acceptance.
    - **Risk Mitigation:** Delivering working increments frequently allows for early identification and mitigation of risks (technical issues, misalignment with requirements).
    - **Transparency:** Scrum's artifacts (Product Backlog, Sprint Backlog, Increment) and events (Sprint Planning, Daily Scrums, Sprint Review, Sprint Retrospective) provide transparency into the project's progress.

While the reference document used a Modified Waterfall, which allows some overlap, Scrum offers a more dynamic and feedback-driven process, better suited for projects where requirements might need adjustment based on user interaction with early versions.

3.3 Data Collection Methods

A combination of qualitative data collection methods will be used to gain a deep understanding of the current document management practices and to elicit detailed user requirements:

- **Semi-structured Interviews:** Conduct interviews with staff members from key departments (e.g., Registrar, HR, Finance, Academic Deans' offices) involved in significant document handling. This method allows for a flexible conversation

guided by a set of core questions, enabling the interviewer to explore specific points in detail and uncover nuanced requirements and challenges [Ref 13: cite interview method source].

- **Observation:** Observe the current manual document flow and storage methods within selected offices. This provides direct insights into the process, volume of documents, classification schemes (or lack thereof), bottlenecks, and user interactions with documents, complementing the information gathered through interviews [Ref 14: cite observation method source].
- **Document Review:** Analyze samples of different document types and existing forms used for tracking (like the dispatch list mentioned in the reference). This helps in understanding the nature of the documents, required metadata, and existing organizational logic, providing concrete examples for system design.

These methods, used in conjunction, allow for triangulation of data, leading to a more comprehensive and accurate understanding of the requirements.

3.4 System Design and Modeling

The design phase will involve creating detailed models of the system using the Unified Modeling Language (UML). These models serve as blueprints for implementation and facilitate communication among stakeholders.

- **UML Diagrams:** The following UML diagrams will be used:
    - *Use Case Diagram:* To illustrate the system's boundaries and the interactions between actors (users, administrators) and the system's functionalities [Ref 15: cite Use Case source].
    - *Activity Diagrams:* To model the step-by-step flow of activities within key processes (e.g., document upload workflow, user registration process) [Ref 16: cite Activity Diagram source].
    - *Sequence Diagrams:* To show the interactions between objects (or system components) over time for specific scenarios (e.g., user login sequence, document retrieval sequence) [Ref 17: cite Sequence Diagram source].
    - *Class Diagram:* To define the static structure of the system, representing the system's entities (classes), their attributes, operations, and relationships (associations, aggregations, composition, inheritance) [Ref 18: cite Class Diagram source].

    - *Deployment Diagram:* To illustrate the physical deployment of the software components on hardware nodes (e.g., client machines, web server, database server) [Ref 19: cite Deployment Diagram source].
- **Modeling Tool:** A UML modeling tool like Enterprise Architect or draw.io will be utilized to create these diagrams efficiently.

3.5 Implementation Technologies

The system will be implemented as a web application using a modern and widely supported technology stack:

- **Backend Framework: Laravel (PHP Framework):** Chosen for its robust features, expressive syntax, large community, strong security features (like built-in CSRF protection and secure hashing), and adherence to the MVC pattern, which promotes organized and maintainable code [Ref 20: cite Laravel source].
- **Programming Language: PHP:** A widely used server-side scripting language well-suited for web development.
- **Database Management System: MySQL:** A popular, open-source relational database known for its performance, reliability, and ease of use, compatible with PHP and Laravel [Ref 21: cite MySQL source].
- **Frontend Technologies: HTML, CSS, JavaScript:** Standard web technologies for structuring content, styling, and adding interactivity.
- **Frontend Framework/Library: Bootstrap:** A widely-used CSS framework for developing responsive and mobile-first websites, ensuring the system is accessible and visually appealing across different devices [Ref 22: cite Bootstrap source].

This combination of technologies provides a solid foundation for building a scalable, secure, and user-friendly web application.

3.6 Testing Strategy

A multi-level testing strategy will be employed to ensure the quality, functionality, usability, and security of the developed EDMS:

- **Unit Testing:** Testing individual code components (functions, methods) in isolation to ensure they perform as expected. This will be done during development by the project team.
- **Integration Testing:** Testing the interaction between different modules or services to ensure they work together correctly.

- **System Testing:** Testing the complete, integrated system against the specified functional and non-functional requirements to verify that it behaves as intended in an end-to-end scenario.
- **User Acceptance Testing (UAT):** Involving selected GUC staff members (end-users) to test the system in a realistic environment. This is crucial for validating usability, workflows, and overall satisfaction against business needs. Checklists derived from requirements will guide UAT sessions [Ref 23: cite a UAT source].
- **Security Testing:** Assessing the system for vulnerabilities, including input validation, authentication/authorization robustness, and protection against common web security threats (e.g., SQL Injection, XSS). Laravel's built-in security features will be leveraged and specifically tested [Ref 24: cite security testing source].
- **Usability Testing:** Evaluating how easy and intuitive the system is for typical users to learn and operate. This will be part of UAT, focusing on the user interface and workflow efficiency.

The testing process will be documented, including test cases, execution results, and defect tracking.

3.7 Ethical Considerations

This project involves collecting information about existing processes and potentially handling sensitive document types. Therefore, ethical considerations are paramount:

- **Informed Consent:** Obtain informed consent from university staff participating in interviews or observations, explaining the purpose of the study, how the data will be used, and ensuring their voluntary participation and right to withdraw.
- **Confidentiality and Anonymity:** Ensure that all data collected from individuals is treated with strict confidentiality. Personal identifiers will be removed or anonymized in any reporting or documentation where possible [Ref 25: cite research ethics guideline].
- **Data Security:** Implement robust security measures in the developed system to protect the university's documents from unauthorized access, modification, or disclosure, aligning with GUC's privacy policies.
- **Transparency:** Be transparent with the university administration and staff about the project's goals, progress, and potential impact on their work processes.

A formal ethical clearance or memorandum will be sought from the relevant university department (e.g., Computer Science or research ethics board) if required, as indicated in the reference material.

Chapter Four: Project Plan (Budget & Schedule)

4.1 Project Budget

This section provides an estimated budget for undertaking the design and implementation of the EDMS for Gage University College as an M.Sc. project. It's important to note that this is an estimate, and actual costs may vary. Many aspects of an academic project leverage existing university infrastructure and free/open-source tools, minimizing direct cash expenditure but requiring significant time investment.

Category	Estimated Cost (ETB)	Notes
Personnel (Student Time)	N/A	Student's time is the primary resource; no direct salary cost.
Personnel (Supervisor)	N/A	Supervisor's time is contributed by the university.
Hardware	5,000 - 10,000	Development machine maintenance, potential shared server allocation cost.
Software Licenses	0	Utilizing open-source tools (Laravel, MySQL, VS Code, Bootstrap, etc.).
Internet/Communication	2,000 - 4,000	Internet access for research, collaboration, and potentially testing deployment.
Stationery/Printing	1,000 - 2,000	Printing proposal copies, documentation, data collection forms.
Incidentals	1,000	Unforeseen small expenses.
Total Estimated Cost	**9,000 - 17,000**	Primarily covers operational expenses.

This budget focuses on the direct costs associated with the student's project work. The significant investment from the university comes in the form of supervisor time, infrastructure (labs, network), and potential server resources for testing/deployment.

4.2 Project Schedule

The project schedule outlines the planned timeline for completing the EDMS development using the Scrum methodology. The project will be broken down into several sprints,

typically lasting 2-4 weeks each. Below is a high-level overview of the phases and estimated duration. A detailed sprint-by-sprint breakdown with specific user stories will be developed during the project planning phase.

Phase	Estimated Duration	Key Activities
Initiation & Planning	2 weeks	Proposal writing, literature review completion, initial project planning.
Sprint 0: Foundation	3 weeks	Setup development environment, finalize initial requirements, design core DB schema, build basic UI layout, implement user authentication/login.
Sprint 1: Document Core	4 weeks	Implement document upload, basic indexing (metadata), secure storage, basic document viewing/downloading.
Sprint 2: Search & Access	4 weeks	Implement metadata-based search, access control logic (roles, departments), basic user management interfaces (for admin).
Sprint 3: Workflow Basics	4 weeks	Implement simple document routing mechanism, notification system, basic audit trail logging.
Sprint 4: Refinement & UAT	3 weeks	Address feedback from initial testing, refine UI/UX, conduct User Acceptance Testing with GUC staff.
Sprint 5: Testing & Polish	3 weeks	Comprehensive system testing, security testing, performance tuning (basic), finalize documentation, prepare for potential deployment.
Reporting & Submission	3 weeks	Finalize project report, thesis writing, prepare for presentation.
Total Estimated Time	**27 Weeks**	Approximately 6-7 months.

This schedule is illustrative and will be adapted based on the progress made in each sprint and feedback received during the development process, consistent with the principles of

the Agile methodology. A more detailed Gantt chart or sprint backlog will be maintained by the project team.

References

[This section will list the sources cited in the proposal, formatted according to IEEE style. Aim for 10-20 credible sources. Examples are placeholders.]

[1] Author(s). "Title of chapter in the book," in *Title of Book*, Edition (if any). Publisher, Year, pp. [page numbers].
[2] J. Smith, M. Jones, and A. Williams, "Document Management Systems: An Overview," *Journal of Information Science*, vol. X, no. Y, pp. P-Q, Year.
[3] B. White, "Indexing Strategies for Digital Archives," *International Journal of Digital Libraries*, vol. A, no. B, pp. C-D, Year.
[4] C. Green, *Digital Preservation and Archiving*. Publisher, Year.
[5] D. Black, "Efficient Document Retrieval Techniques," *Journal of Computer Science Applications*, vol. E, no. F, pp. G-H, Year.
[6] E. Grey, "Implementing Version Control in EDMS," *Software Engineering Journal*, vol. I, no. J, pp. K-L, Year.
[7] F. Brown, *Business Process Management and Workflow Systems*. Publisher, Year.
[8] G. Blue, "Security Considerations for Web-Based Information Systems," *IEEE Transactions on Software Engineering*, vol. M, no. N, pp. O-P, Year.
[9] Open Source Initiative, "The Open Source Definition." [Online]. Available: [URL]. [Accessed Date].
[10] H. Red, "Case Study: EDMS Implementation in a University," *Journal of Educational Technology Systems*, vol. Q, no. R, pp. S-T, Year.
[11] I. Orange, "Factors Affecting User Adoption of New Software," *International Journal of Human-Computer Studies*, vol. U, no. V, pp. W-X, Year.
[12] J. Yellow, "Challenges and Strategies for Data Migration," *Journal of Database Management*, vol. Y, no. Z, pp. AA-BB, Year.
[13] K. Purple, "Conducting Semi-Structured Interviews for Requirements Gathering," *Journal of Systems and Software*, vol. CC, no. DD, pp. EE-FF, Year.
[14] L. Pink, *Qualitative Research Methods: A Practitioner's Guide*. Publisher, Year.
[15] M. Cyan, *UML for Software Engineering*. Publisher, Year.
[16] N. Magenta, "Modeling Business Processes with Activity Diagrams," *Business Process Management Journal*, vol. GG, no. HH, pp. II-JJ, Year.
[17] O. Lime, "Using Sequence Diagrams in System Design," *Journal of Object-Oriented*

Programming, vol. KK, no. LL, pp. MM-NN, Year.
[18] P. Gold, *Database Design: A Practical Approach*. Publisher, Year.
[19] Q. Silver, "Deployment Strategies for Web Applications," *IEEE Software*, vol. OO, no. PP, pp. QQ-RR, Year.
[20] Laravel Documentation. [Online]. Available: [URL]. [Accessed Date].
[21] MySQL Documentation. [Online]. Available: [URL]. [Accessed Date].
[22] Bootstrap Documentation. [Online]. Available: [URL]. [Accessed Date].
[23] S. Bronze, *Software Testing: A Guide to the Quality*. Publisher, Year.
[24] T. Copper, "Web Application Security Testing," *Journal of Cybersecurity*, vol. SS, no. TT, pp. UU-VV, Year.
[25] University Name, "Guidelines for Ethical Research." [Online]. Available: [URL]. [Accessed Date].

Appendices

[This section will include supplementary materials that support the proposal but are not essential for the main body.]

- **Appendix A: Interview Guide:** The set of guiding questions used for semi-structured interviews with university staff.
- **Appendix B: Sample Observation Checklist:** The form used to record observations of the manual document management process.
- **Appendix C: Detailed UML Diagrams:** Full-size versions of the Use Case, Class, Activity, Sequence, and Deployment diagrams.
- **Appendix D: Preliminary Test Case Examples:** Sample test cases for key functionalities (e.g., document upload, search).
- **Appendix E: Ethical Consent Form:** The template for the informed consent form provided to interview participants.

Formatting Standards Checklist

- Font: Times New Roman, 12pt
- Line Spacing: 1.5
- Alignment: Justified
- Page Numbering: Bottom center or right (starting after the Title page)

- Heading Numbering: 1, 1.1, 1.1.1, 2, 2.1, etc.
- Referencing Style: IEEE

Final Notes for the Student:

- This is a structured *draft*. You need to fill in the bracketed placeholders [ ] with your specific information, dates, names, etc.
- **Crucially**, you must expand each section with detailed content. The text provided under each heading is a starting point and outline. You need to elaborate significantly, providing specific examples based on your research and the GUC context.
- The literature review (Chapter 2) requires finding and citing actual relevant academic sources. The placeholder citations [Ref #] need to be replaced with real references.
- The Project Budget (Chapter 4.1) and Schedule (Chapter 4.2) are estimations and high-level. You'll need to detail these more precisely based on your actual project planning.
- The Appendices need to be created and included in the final proposal.
- Ensure consistency throughout the document (e.g., referring to the chosen methodology consistently).
- Review the proposal writing manual provided by your university for any additional specific requirements or nuances unique to their guidelines.
- **Consult your supervisor frequently** during the process! They will provide feedback on your ideas and writing.

Good luck with your project proposal!