ACKNOWLEDGEMENT

Competition among different sectors is growing more around the global. Every field needs some practical knowledge and work experience. We are thankful to college “LDRP-ITR” for granting permission to make Project on “document management system”.

I am thankful to my head of department Dr. Mehul Barot and our Internal Guide prof. Yogesh vaghela for being instructor in our entire journey right from the start till the completion of this project by sharing their valuable time with us.

We hope you will satisfy with our project.

Vrajesh patel (222SBEIT30011)

Vraj patel (222SBEIT30010)

Hasit patel (222SBEIT30009)

Chaitya nayak (222SBEIT30008)

ABSTRACT

The rapid digital transformation has necessitated the development of efficient systems to manage and organize digital documents. This paper presents the design and implementation of a Document Management System (DMS) leveraging the strengths of PHP, MySQL, and React.js. The proposed DMS aims to facilitate seamless document storage, retrieval, and management for organizations of varying sizes.

The backend of the system is built using PHP, a robust server-side scripting language, which ensures secure and efficient processing of requests. MySQL, a reliable and high-performance relational database management system, is used to manage and store metadata and document information. The frontend is developed using React.js, a popular JavaScript library for building user interfaces, ensuring a responsive and dynamic user experience.

Key features of the system include user authentication and authorization, document version control, full-text search capabilities, and role-based access control. The DMS provides a user-friendly interface for uploading, downloading, and sharing documents. It also incorporates audit trails and logging to ensure accountability and transparency.

The integration of PHP, MySQL, and React.js in this Document Management System offers a scalable, secure, and efficient solution to the challenges of document management. This paper details the system architecture, design considerations, and implementation strategies, along with an evaluation of the system's performance and usability.

LIST OF FIGURES

TABLE OF CONTENT

1 INTRODUCTION

- **INTRODUCTION**
- **SCOPE**
- **PROJECT SUMMERY AND PURPOSE**
- **OBJECTIVES**

1.1 INTRODUCTION

In today's digital age, the efficient management of documents is crucial for organizations to maintain productivity and ensure the security of sensitive information. The sheer volume of digital documents generated daily necessitates a robust and scalable Document Management System (DMS) to streamline the processes of document creation, storage, retrieval, and sharing. This project aims to develop a comprehensive DMS utilizing PHP, MySQL, and React.js, leveraging their strengths to create a seamless and user-friendly system.

PHP, known for its server-side scripting capabilities, provides the backbone of the DMS by handling server-side logic and ensuring secure processing of user requests. MySQL, a powerful relational database management system, is employed to efficiently store and manage document metadata and user information. React.js, a widely adopted JavaScript library for building dynamic user interfaces, offers a responsive and interactive frontend experience for users.

The proposed DMS is specifically designed to manage PDF documents, addressing common challenges faced by organizations, including secure access management and efficient search and retrieval of documents. In this system, only administrators have the privilege to upload files, ensuring centralized control and maintaining the integrity of the document repository. Regular users are granted the ability to search for, download, and view documents, as well as access user logs to track document access and activity.

The system's architecture is designed to be scalable, accommodating the growing needs of organizations as they expand. Its user-friendly interface simplifies document management tasks, allowing users to easily navigate and retrieve necessary documents. The implementation of user authentication and role-based access control ensures that documents are accessed only by authorized personnel, enhancing security and compliance.

This report delves into the detailed design and implementation of the Document Management System, exploring the technical aspects and design considerations that influenced the development process. It also evaluates the system's performance and usability, demonstrating its effectiveness in addressing the document management needs of modern organizations.

1.2 SCOPE

1.2.1 CURRENT SCOPE

The Document Management System (DMS) developed using PHP, MySQL, and React.js is designed to address specific document management needs within an organization. The current scope of the system encompasses the following functionalities and features:

Document Upload and Management (Admin Only):

- **PDF Document Support**: The system exclusively supports the upload and management of PDF files, ensuring a standardized format for all documents.
- **Admin Privileges**: Only users with administrative privileges can upload new documents to the system. This centralized control helps maintain the integrity and consistency of the document repository.
- **Document Version Control**: The system tracks and manages different versions of documents, allowing administrators to update and maintain the latest versions without losing historical data.

User Authentication and Authorization:

- **Role-Based Access Control**: The system employs role-based access control to differentiate between administrators and regular users, ensuring that only authorized personnel can perform specific actions such as uploading documents.
- **Secure Login**: Users must authenticate themselves to access the system, ensuring that document access is restricted to authorized individuals.

Document Retrieval and Search (User Access):

- **Search Functionality**: Users can search for documents using various criteria such as keywords, document titles, and metadata, enabling efficient retrieval of necessary documents.
- **Download Capability**: Users can download documents from the system for offline use, ensuring accessibility to important information when needed.
- **View-Only Access**: Users can view document details and metadata without modifying the content, preserving the integrity of the documents.

Audit Trails and User Logs:

- **Activity Logging**: The system maintains detailed logs of user activities, including document access, downloads, and search queries. This feature enhances transparency and accountability within the organization.
- **User Log Access**: Users can view logs of their activities and track document interactions, promoting self-monitoring and compliance with organizational policies.

User-Friendly Interface:

- **Responsive Design**: The frontend, built using React.js, offers a responsive and intuitive user interface, ensuring a seamless user experience across various devices and screen sizes.
- **Simplified Navigation**: The interface is designed for ease of use, allowing users to quickly locate and access the documents they need.

Security and Compliance:

- **Data Encryption**: The system ensures that all data transmitted between the client and server is encrypted, safeguarding sensitive information from unauthorized access.
- **Compliance with Standards**: The DMS is designed to comply with relevant industry standards and regulations for document management and data protection.

This Document Management System is currently focused on providing a secure, efficient, and user-friendly platform for managing PDF documents within an organization. Future enhancements may include support for additional document formats, expanded administrative functionalities, and integration with other enterprise systems.

1.2.2 FUTURE SCOPE

The Document Management System (DMS) developed using PHP, MySQL, and React.js has significant potential for future enhancements and expansions. The future scope of the system includes the following areas of improvement and new features:

Support for Additional Document Formats:

- **Multi-Format Support**: Expanding the system to handle various document formats such as Word documents, Excel spreadsheets, images, and text files will increase its versatility and usability across different departments and use cases.

Enhanced Document Security:

- **Advanced Encryption Techniques**: Implementing more advanced encryption techniques for document storage and transmission can further enhance the security of sensitive information.
- **Watermarking and Digital Signatures**: Adding features like document watermarking and digital signatures can help in protecting the integrity and authenticity of documents.

Integration with Other Systems:

- **Enterprise Resource Planning (ERP) Integration**: Integrating the DMS with existing ERP systems can streamline document-related workflows and improve overall organizational efficiency.
- **Cloud Storage Integration**: Providing integration with popular cloud storage solutions such as Google Drive, Dropbox, and OneDrive will enable seamless synchronization and access to documents from multiple platforms.

Advanced Search Capabilities:

- **Full-Text Search**: Implementing a full-text search engine to enable comprehensive search capabilities within document contents, not just metadata.
- **AI-Powered Search**: Utilizing artificial intelligence and machine learning to enhance search functionality, providing more accurate and relevant search results based on user queries.

Improved User Experience:

- **Mobile Application**: Developing a mobile application for both Android and iOS platforms to provide users with convenient access to the DMS on the go.
- **Customizable Dashboard**: Allowing users to customize their dashboards with widgets and shortcuts to frequently accessed documents and features.

Collaboration Features:

- **Real-Time Collaboration**: Enabling real-time collaboration on documents, allowing multiple users to view and edit documents simultaneously.
- **Commenting and Annotations**: Adding functionality for users to leave comments and annotations on documents to facilitate better communication and collaboration.

Advanced Administrative Tools:

- **Automated Workflows**: Implementing automated workflows for document approval, review, and archiving processes to reduce manual intervention and improve efficiency.
- **Detailed Analytics and Reporting**: Providing administrators with advanced analytics and reporting tools to monitor system usage, document access patterns, and user activities.

Compliance and Regulatory Features:

- **Regulatory Compliance**: Ensuring that the DMS complies with various industry-specific regulations and standards, such as GDPR, HIPAA, and ISO standards.
- **Retention Policies**: Implementing document retention policies to automatically manage the lifecycle of documents based on organizational or regulatory requirements.

Enhanced User Management:

- **Granular Access Control**: Introducing more granular access control mechanisms, allowing administrators to define permissions at a more detailed level, such as specific document folders or individual documents.
- **Single Sign-On (SSO)**: Integrating with Single Sign-On solutions to streamline user authentication and improve security.

Scalability and Performance:

- **Scalable Architecture**: Continuously improving the system's architecture to handle increasing numbers of users and documents without compromising performance.
- **Performance Optimization**: Regularly optimizing the system to ensure fast and efficient document processing and retrieval.

The future enhancements and expansions outlined above will significantly enhance the capabilities of the Document Management System, making it a more powerful and comprehensive solution for document management within organizations.

1.3 PROJECT SUMMERY AND PURPOSE

Project Summary

This project entails the development of a Document Management System (DMS) designed to facilitate the efficient management, storage, retrieval, and sharing of PDF documents within an organization. Utilizing a combination of PHP, MySQL, and React.js, the system aims to provide a secure, user-friendly, and scalable solution for document management.

The DMS offers a range of features tailored to meet the needs of both administrators and regular users. Administrators have exclusive privileges to upload and manage documents, ensuring centralized control and integrity of the document repository. Regular users can search for, view, and download documents, as well as access user logs to monitor document interactions.

The system's architecture leverages PHP for server-side scripting, ensuring robust backend processing, while MySQL is used for efficient data management and storage. React.js powers the frontend, delivering a dynamic and responsive user interface. Key functionalities include role-based access control, document version control, audit trails, and a comprehensive search feature.

Purpose

The primary purpose of this Document Management System is to streamline the processes involved in managing digital documents within an organization. By centralizing document storage and providing robust access controls, the DMS aims to improve organizational efficiency, enhance document security, and ensure compliance with regulatory requirements.

1. **Enhanced Document Security**: By implementing role-based access control and user authentication, the system ensures that documents are accessed only by authorized personnel. This helps protect sensitive information and maintain confidentiality.
2. **Centralized Document Management**: With a centralized repository for all PDF documents, the DMS simplifies document storage, retrieval, and management, reducing the risk of document loss or misplacement.
3. **Improved Efficiency**: The system's user-friendly interface and powerful search capabilities enable users to quickly locate and access necessary documents, thereby enhancing productivity and reducing time spent on document-related tasks.
4. **Accountability and Transparency**: Through detailed audit trails and user logs, the system provides comprehensive records of document interactions, promoting accountability and transparency within the organization.
5. **Regulatory Compliance**: The DMS helps organizations comply with industry standards and regulations related to document management and data protection by providing secure storage, access controls, and audit capabilities.

By addressing these key areas, the Document Management System aims to be an essential tool for organizations seeking to optimize their document management processes and enhance overall operational efficiency.

1.4 OBJECTIVES

The primary objectives of the Document Management System (DMS) project are to develop a secure, efficient, and user-friendly platform for managing PDF documents within an organization. The specific objectives include:

Centralized Document Storage:

- To create a centralized repository for storing all PDF documents, ensuring easy access and management.
- To facilitate organized document storage with metadata for quick retrieval and categorization.

Role-Based Access Control:

- To implement a robust role-based access control system that restricts document upload capabilities to administrators while allowing users to search, view, and download documents.
- To ensure that only authorized personnel can access sensitive documents, maintaining document confidentiality and security.

User Authentication and Authorization:

- To provide a secure login system for authenticating users, ensuring that only authorized individuals can access the DMS.
- To differentiate between administrative and regular user roles, granting appropriate permissions based on user roles.

Efficient Document Retrieval and Search:

- To develop an advanced search functionality that allows users to quickly locate documents using keywords, titles, and metadata.
- To ensure that the search feature is intuitive and provides accurate results, enhancing user productivity.

Document Version Control:

- To implement document version control to track changes and maintain a history of document revisions.
- To allow administrators to manage and update document versions without losing historical data.

Audit Trails and User Logs:

- To maintain detailed logs of user activities, including document access, downloads, and search queries.
- To provide transparency and accountability by allowing users and administrators to view logs of document interactions.

User-Friendly Interface:

- To design a responsive and intuitive user interface using React.js, ensuring a seamless user experience across various devices and screen sizes.
- To simplify navigation and make it easy for users to perform tasks such as searching, viewing, and downloading documents.

Document Security and Compliance:

- To implement data encryption for secure transmission and storage of documents, protecting sensitive information from unauthorized access.
- To ensure compliance with industry standards and regulations related to document management and data protection.

Scalability and Performance:

- To design the system architecture to be scalable, accommodating the growing needs of the organization as the number of users and documents increases.
- To optimize system performance for fast and efficient document processing and retrieval.

Future-Proofing and Expandability:

- To build a flexible system that can be easily expanded to support additional document formats and future enhancements.
- To ensure the system can integrate with other enterprise systems and cloud storage solutions, providing seamless data synchronization and access.

By achieving these objectives, the Document Management System will provide a comprehensive solution for managing PDF documents, improving organizational efficiency, and ensuring document security and compliance.

2 TECHNOLOGY AND LITTRATURE REVIEW

- **TOOLS AND TECHNOLOGY**
- **PROJECT PLANNING**
- **PROJECT SCHEDULING**
- **COST ESTIMATION**

2.1 TOOLS AND TECHNOLOGY USED

Backend Technologies:

- **PHP**: A widely-used open-source scripting language that is particularly suited for web development. PHP is used to handle server-side logic, processing requests, and managing interactions with the database.
- **MySQL**: A reliable and high-performance relational database management system. MySQL is used for storing and managing document metadata, user information, and activity logs.

Frontend Technologies:

- **React.js**: A popular JavaScript library for building user interfaces. React.js is used to create a responsive and dynamic frontend, providing a seamless user experience.
- **HTML5**: The standard markup language for creating web pages. HTML5 is used to structure the content of the web application.
- **CSS3**: A stylesheet language used for describing the presentation of a document written in HTML. CSS3 is used to style the web application and ensure a consistent look and feel.

Development Tools:

- **Visual Studio Code**: A powerful, open-source code editor used for writing and editing code in various programming languages, including PHP, JavaScript, HTML, and CSS.
- **Git**: A version control system used to track changes in the source code during development. Git facilitates collaboration among team members and helps manage project versions.
- **GitHub**: A web-based platform for version control and collaboration. GitHub is used to host the project's repository, manage code changes, and facilitate team collaboration.

Web Server:

- **Apache**: A widely-used open-source web server software. Apache is used to serve the web application and handle HTTP requests.

Security Technologies:

- **HTTPS**: Hypertext Transfer Protocol Secure, used to ensure secure communication between the client and server by encrypting data transmission.
- **SSL/TLS Certificates**: Secure Sockets Layer/Transport Layer Security certificates are used to establish a secure connection between the client and server.

Libraries and Frameworks:

- **Bootstrap**: A popular front-end framework for developing responsive and mobile-first web applications. Bootstrap is used to enhance the styling and layout of the web application.
- **Axios**: A promise-based HTTP client for making API requests. Axios is used in React.js for handling HTTP requests to the server.

Database Management Tools:

- **phpMyAdmin**: A free and open-source administration tool for MySQL and MariaDB. phpMyAdmin is used to manage the MySQL database, execute SQL queries, and handle database administration tasks.

Testing and Debugging Tools:

- **Postman**: An API development and testing tool. Postman is used to test API endpoints and ensure proper communication between the frontend and backend.
- **Chrome DevTools**: A set of web developer tools built directly into the Google Chrome browser. Chrome DevTools is used for debugging, profiling, and optimizing the performance of the web application.

Deployment and Hosting:

- **Docker**: A platform for developing, shipping, and running applications in containers. Docker is used to containerize the application, ensuring consistent environments across development, testing, and production.
- **AWS (Amazon Web Services)**: A comprehensive cloud platform offering various services for hosting and managing web applications. AWS can be used to deploy and host the DMS.

By utilizing these tools and technologies, the Document Management System aims to provide a robust, secure, and efficient solution for managing PDF documents within an organization.

2.2 PROJECT PLANNING

2.2.1 Project Development Approach:

The model that is referred for the development of the project is incremental model. It combines elements of the waterfall model applied in an iterative fashion. In this process the phases are same as waterfall but the advantage is that when first phase is done it is incremented and then the other phases are carried with the same cycle. Here in this add ons on each phase can be added according to the need of the client and the project.

Phases are as follows:

1. Communication 2
2. Planning
3. Modeling : Includes Designing
4. Construction
5. Deployment: Feedback , Delivery

Each phases are iteratively carried out. Main reason for using this any other is waterfall has the drawback of iterations, if there is any other requirement added later on then this is not possible to add up in it, Spiral model has disadvantage that it need more manpower and even itis for multiple transactions or multiple tasks handling projects and so does the time consumption is more in it for those projects.

Planning is essential cause multiple software teams works in parallel on different system functions. Scalability should be obtained in any of the project selected but it is not available in waterfall cause of few drawbacks.

2.2.2 MILESTONES AND DELIVERABLES

Month1:- Milestones & Deliverables

Milestones	Deliverables
Study about our web application requirement, planning	Analysis Report
Understand a project definitions and basic terms and logic for Parameter Evaluation.	
Gathering the requirements of the project using different fact finding techniques.	
Still Continue with Requirement’s study.	

Month 2: -Milestones & Deliverables

Milestones	Deliverables
System Analysis	Analysis report
System design including various diagrams	SRS

Month 3 : -Milestones & Deliverables

Milestones	Deliverables
Database creation and procedures	Designing/ coding
Admin module	
Client module	

Month 4 : -Milestones & Deliverables

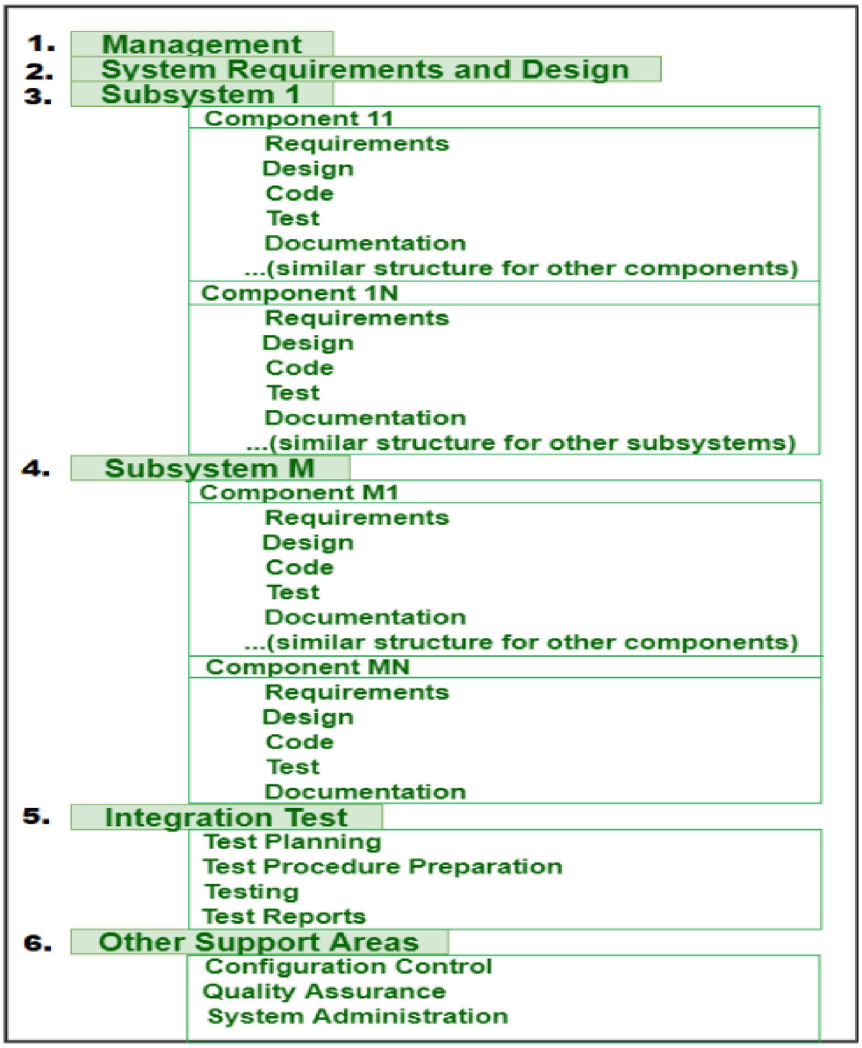
Milestones	Deliverables
Website testing	Testing
Required changes after testing	Designing/coding

2.3 PROJECT SCHEDULING

Project scheduling involves separating the total work in a project into separate activities and judging the time required to complete these activities. Usually, some of these activities are carried out in parallel.

Work Breakdown Structure :

Work Breakdown Structure is used to decompose a given task set recursively into small activity.



1. Management
2. System Requirements and Design
3. Subsystem 1
    - Component 11
        - Requirements
        - Design
        - Code
        - Test
        - Documentation
        - ...(similar structure for other components)
    - Component 1N
        - Requirements
        - Design
        - Code
        - Test
        - Documentation
        - ...(similar structure for other subsystems)
4. Subsystem M
    - Component M1
        - Requirements
        - Design
        - Code
        - Test
        - Documentation
        - ...(similar structure for other components)
    - Component MN
        - Requirements
        - Design
        - Code
        - Test
        - Documentation
5. Integration Test
    - Test Planning
    - Test Procedure Preparation
    - Testing
    - Test Reports
6. Other Support Areas
    - Configuration Control
    - Quality Assurance
    - System Administration

Conventional Work Breakdown Structure

FIG(2.3.1)

2.4 ESTIMATION

2.4.1 Effort estimation

Effort Estimation includes the time required i.e. the dedication for developing the project successfully. Gantt chart includes that each phases or each modules divided so that is according to the plan.

2.4.2 Cost estimation

There are two types of models that have been used to estimate cost: cost models and constraint models. Cost models provide direct estimates of effort. These models typically have a primary cost factor such as size and the number of secondary adjustment factors or cost drivers. Cost drivers are characteristics of the project, process, products, or resources that influence effort. Cost drivers are used to adjust the preliminary estimate provided by the primary cost factor. **COCOMO (Constructive Cost estimation Model)** is a Heuristic Technique which can be used for our project.

Software project can be classified in one of the three categories based on the project complexity.

Software project ab bb cb db Select ORGANIC 2.4 1.05 2.5 0.38

SEMI DETACHED 3.0 1.12 2.5 0.35

EMBEDDED 3.6 1.20 2.5 0.32

Since we are unfamiliar with some aspects of the system like image processing etc. our project falls under the category of semidetached type.

Final step is to select complexity of software project.

Effort = a1*(KLOC) ^a2 PM

Tdev = b1*(Effort) ^b2 Months
Initial COCOMO estimate without cost drivers:
Effort = 3.0*(8.500) ^1.12 = 32.96 PM approx
Tdev = 2.5*(32.96) ^0.35 = 8.49 Months approx.

This doesn’t mean that in 8.49 Months 1 person completes the project nor does it indicate the reverse. Cost drivers have not been taken into consideration.

If it take an estimate of cost incurred:

Consider, per month the developer gets 10,000 Rupees as a trainee. Then the total cost will be 8.49*10,000 = 84,900 Rupees for developing the software.

Transportation, hardware and other cost have not been included.

COCOMO model can be used to estimate the effort, development time and the cost for developing the software. Staffing is very important after these estimates. In this project, there are only four team members and the work is done by all equally in all the phases of development.

3 SYSTEM REQUIREMENT STUDY

- **USER CHARACTERISTICS**
- **HARDWARE AND SOFTWARE REQUIREMENTS**

3.1 USER CHARASTRISTICS

□ **Administrators**

- **Role**: Administrators have the highest level of access and control within the system. They are responsible for managing the overall functionality, security, and maintenance of the document management system.
- **Responsibilities**:
  - Upload and manage PDF documents.
  - Assign and manage user roles and permissions.
  - Monitor and review user activity logs.
  - Ensure system security and compliance.
- **Technical Skills**: Intermediate to advanced knowledge of IT systems, familiarity with document management processes, and basic understanding of security protocols.
- **Usage Frequency**: Regular to frequent use, depending on the number of documents and users managed.

□ **Regular Users**

- **Role**: Regular users have limited access to the system. Their primary interaction is with searching, viewing, and downloading documents.
- **Responsibilities**:
  - Search for specific documents using keywords, titles, or metadata.
  - View and download available PDF documents.
  - Follow organizational guidelines for document usage and handling.
- **Technical Skills**: Basic computer literacy, familiarity with web-based applications, and basic understanding of document management processes.
- **Usage Frequency**: Varies based on individual needs, ranging from occasional to regular use.

□ **IT Support Staff**

- **Role**: IT support staff assist in maintaining the system, troubleshooting issues, and ensuring smooth operation of the document management system.
- **Responsibilities**:
  - Provide technical support to users experiencing issues.
  - Perform routine maintenance and updates.
  - Ensure data integrity and backup processes are followed.
- **Technical Skills**: Advanced technical skills, including knowledge of PHP, MySQL, React.js, and general IT support procedures.
- **Usage Frequency**: Regular, with additional on-demand support as needed.

□ **Compliance Officers**

- **Role**: Compliance officers ensure that the document management system adheres to relevant regulatory and organizational policies.
- **Responsibilities**:
    - Review and audit system activities and logs.
    - Ensure that documents are managed according to compliance requirements.
    - Work with administrators to implement necessary security measures.
- **Technical Skills**: Intermediate knowledge of compliance standards and regulations, basic understanding of IT systems, and familiarity with document management processes.
- **Usage Frequency**: Periodic, with additional reviews based on audit schedules.

□ **Trainers**

- **Role**: Trainers are responsible for educating and training users on how to effectively use the document management system.
- **Responsibilities**:
    - Conduct training sessions and create user manuals.
    - Provide ongoing support and guidance to users.
    - Develop and update training materials as needed.
- **Technical Skills**: Good communication skills, basic to intermediate technical knowledge of the system, and proficiency in creating instructional materials.
- **Usage Frequency**: Periodic, primarily during initial user onboarding and system updates.

3.2 SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements

Server-Side Software:

- **Operating System**:
    - Linux (Ubuntu, CentOS, or similar)
    - Windows Server
- **Web Server**:
    - Apache HTTP Server
    - Nginx (optional)
- **Programming Language**:
    - PHP 7.4 or higher
- **Database**:
    - MySQL 5.7 or higher
- **Version Control**:
    - Git

- **Development Environment**:
    - Visual Studio Code or any preferred IDE
- **Containers (Optional)**:
    - Docker for containerizing the application

Client-Side Software:

- **Web Browser**:
    - Latest versions of Chrome, Firefox, Safari, or Edge
- **Frontend Framework**:
    - React.js 17 or higher
- **JavaScript Libraries**:
    - Axios for HTTP requests
    - Bootstrap for responsive design (optional)

Security Software:

- **SSL/TLS Certificates**:
    - For HTTPS communication
- **Firewall**:
    - To protect the server from unauthorized access
- **Antivirus/Anti-Malware**:
    - To ensure server integrity

Other Software:

- **phpMyAdmin**:
    - For database management (optional)
- **Postman**:
    - For API testing
- **Collaboration Tools**:
    - Slack, Jira, or similar tools for team communication and project management

Hardware Requirements

Server Hardware:

- **Processor**:
  - Quad-core CPU (Intel Xeon or equivalent)
- **Memory**:
  - 8 GB RAM or higher
- **Storage**:
  - 500 GB SSD or higher
  - Additional storage for backups and logs
- **Network**:
  - High-speed internet connection
  - Network Interface Card (NIC) supporting at least 1 Gbps
- **Power Supply**:
  - Uninterruptible Power Supply (UPS) for backup

Client Hardware:

- **Processor**:
  - Dual-core CPU (Intel i3 or equivalent)
- **Memory**:
  - 4 GB RAM or higher
- **Storage**:
  - 100 GB HDD/SSD
- **Display**:
  - Monitor with at least 1366x768 resolution
- **Peripherals**:
  - Keyboard, mouse, and other standard input devices
- **Network**:
  - Internet connection (broadband recommended)

Backup and Recovery:

- **External Storage**:
  - External hard drives or network-attached storage (NAS) for backups
- **Backup Software**:
  - Backup and recovery software to automate backup processes

Additional Considerations:

- **Scalability**:
  - Ensure hardware can be upgraded to accommodate future growth in users and data volume
  - Consider redundant power supplies, network connections, and storage solutions for high availability

4 SYSTEM DIAGRAMS

- **SYSTEM FLOW CHART**
- **USECASE DIAGRAM**
- **CLASS DIAGRAM**
- **ER- DIAGRAM**
- **ACTIVITY DIAGRAM**
- **SEQUENCE DIAGRAM**

4.1 SYSTEM FLOW CHART

User Login and Authentication:

- **Start**: The system starts when a user opens the DMS web application.
- **Input**: User enters login credentials (username and password).
- **Process**: The system sends the login request to the backend server (PHP) which queries the MySQL database to authenticate the user.
- **Decision**:
    - If authentication is successful, the user is granted access.
    - If authentication fails, an error message is displayed, and the user is prompted to try again.
- **Output**: User is logged in or an error message is displayed.

Document Upload (Admin Only):

- **Start**: Admin selects the option to upload a document.
- **Input**: Admin chooses a PDF file to upload.
- **Process**:
    - The system processes the file upload request.
    - The file is stored in a designated directory on the server.
    - Metadata about the file (e.g., title, upload date) is stored in the MySQL database.
- **Output**: A confirmation message is displayed, and the file is made available for users to download.

Document Search and Download:

- **Start**: User selects the option to search for a document.
- **Input**: User enters search criteria (e.g., keywords, title).
- **Process**:
    - The system queries the MySQL database for documents that match the search criteria.
    - A list of matching documents is displayed.
- **Decision**:
    - If the user selects a document, the system initiates the download process.
- **Output**: The selected document is downloaded to the user's device.

User Activity Logging:

- **Process**: For every significant action (e.g., login, document upload, document download), the system logs the activity.
- **Storage**: Logs are stored in the MySQL database, detailing the user ID, action taken, and timestamp.
- **Access**: Admins can view the user activity logs through a dedicated interface.

Logout:

- **Start**: User selects the option to log out.
- **Process**: The system ends the user session.
- **Output**: User is redirected to the login page.

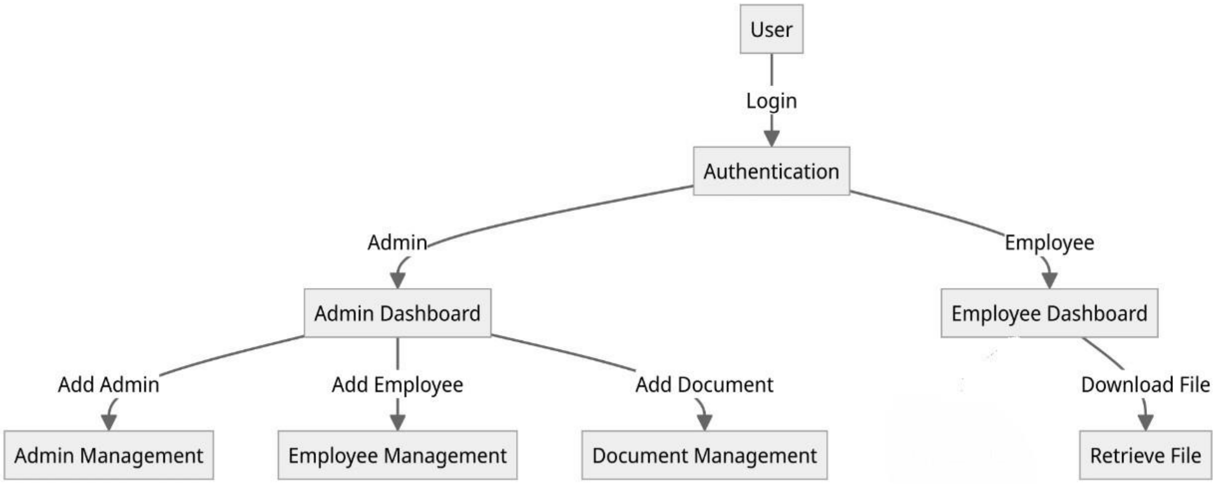


Fig (4.1.1) system flow chart

4.2 USECASE DIAGRAM

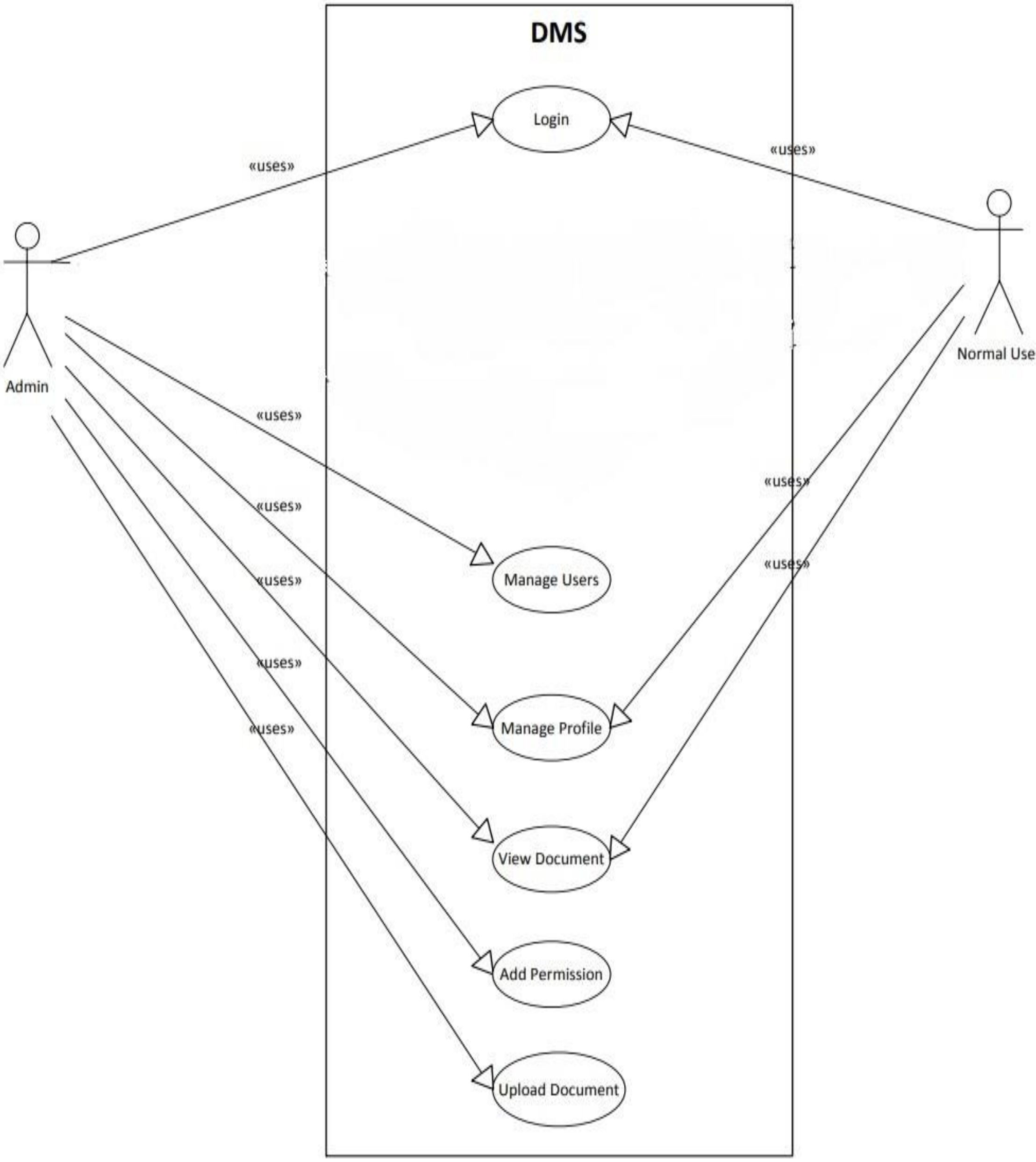
A Use Case Diagram is a visual representation of the interactions between users (actors) and a system, depicting the different ways in which users can achieve their goals using the system. It helps in understanding the functional requirements of a system and defines the scope of the system by identifying the interactions between users and the system.

Key Elements of a Use Case Diagram

1. **Actors**:
   - **Admin**: Manages documents, user permissions, and views logs.
   - **Regular User**: Searches and downloads documents.
2. **Use Cases**:
   - **Login**: All actors must log in to the system.
   - **Upload Document**: Admin uploads documents to the system.
   - **Search Document**: Users search for documents.
   - **Download Document**: Users download documents.
   - **View User Logs**: Admin can review user activity logs.

USECASE DIAGRAM DESCRIPTION

1. **Login**:
   - **Actors**: Admin, Regular User
   - **Description**: All users must authenticate themselves by logging into the system.
2. **Upload Document**:
   - **Actor**: Admin
   - **Description**: Admin uploads new PDF documents to the system.
3. **Search Document**:
   - **Actors**: Admin, Regular User
   - **Description**: Users search for documents using keywords or metadata.
4. **Download Document**:
   - **Actors**: Admin, Regular User
   - **Description**: Users download documents from the system.
5. **View User Logs**:
   - **Actors**: Admin
   - **Description**: Admin can review the logs of user activities.



Fig(4.2.1) usecase diagram

4.3 ER DIAGRAM

An Entity-Relationship (ER) Diagram is a visual representation of the data model of a system, showing how data entities relate to each other within that system. It is widely used in database design to depict the logical structure of databases, including the relationships and constraints that define the data.

KEY ELEMENTS OF ER DIAGRAM

1. **Entities**:
   - Represent objects or things within the system that have a distinct existence.
   - Depicted as rectangles.
   - Example entities for a Document Management System:
     - User
     - Document
     - Role
     - Log
2. **Attributes**:
   - Represent properties or characteristics of entities.
   - Depicted as ovals connected to their respective entities.
   - Example attributes:
     - User: userID, username, password, email
     - Document: documentID, title, uploadDate, filePath
     - Role: roleID, roleName
     - Log: logID, action, timestamp, userID
3. **Relationships**:
   - Represent the associations between entities.
   - Depicted as diamonds connecting entities.
   - Example relationships:
     - A User uploads a Document
     - A Role is assigned to a User
     - A User performs an Action logged in Log
4. **Cardinality**:
   - Defines the numerical relationships between entities.
   - Common types include one-to-one, one-to-many, and many-to-many.
   - Example cardinality:
     - One User can upload many Documents (one-to-many).
     - Each Document is uploaded by one User (many-to-one).
     - One User can have one Role (one-to-one).

ER Diagram Description

- **User**:
  - **Attributes**: userID, username, password, email, roleID
- **Document**:
  - **Attributes**: documentID, title, uploadDate, filePath, userID
- **Role**:
  - **Attributes**: roleID, roleName
- **Log**:
  - **Attributes**: logID, action, timestamp, userID

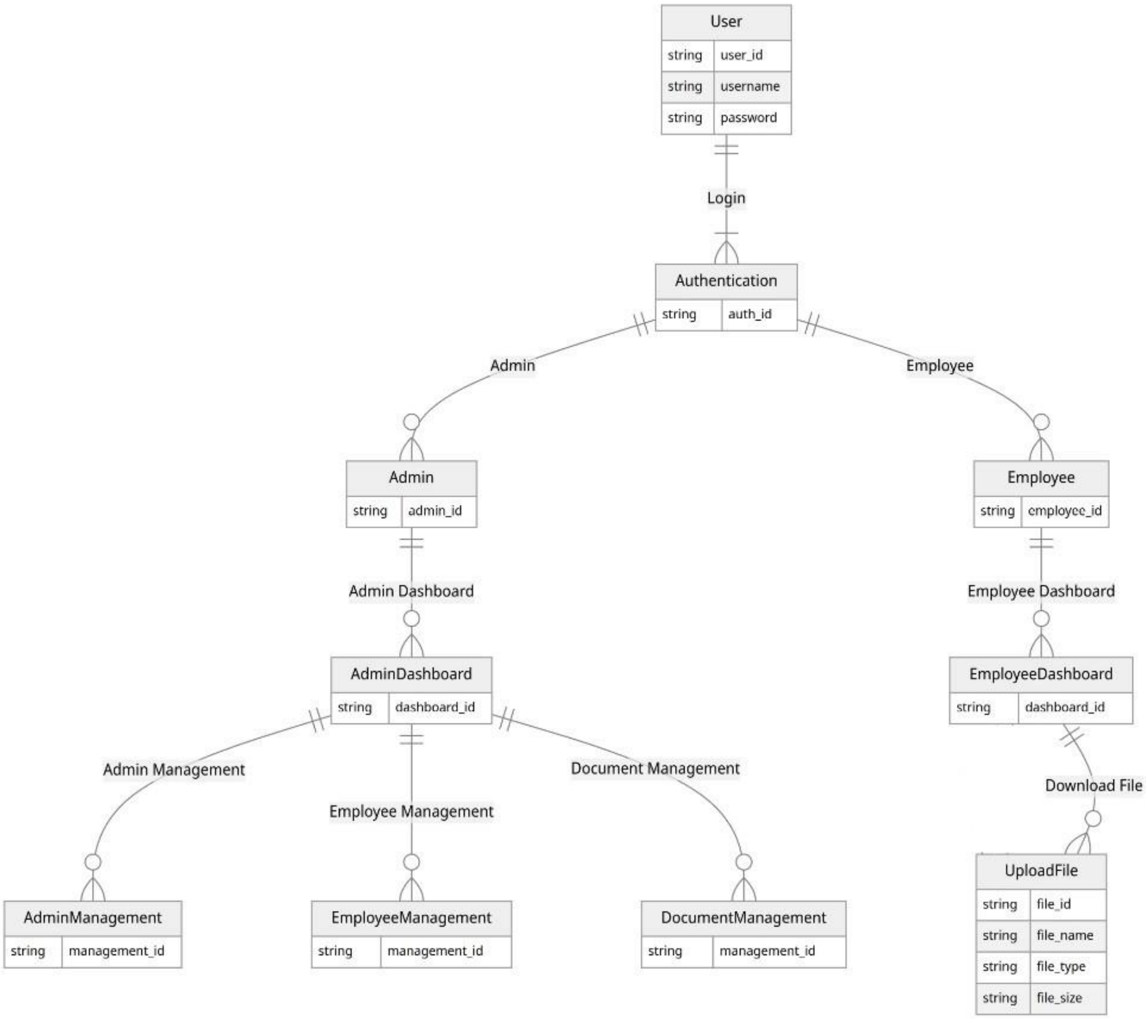


Fig (4.3.1) ER Diagram

4.4 ACTIVITY DIAGRAM

An Activity Diagram is a type of UML (Unified Modeling Language) diagram that illustrates the dynamic aspects of a system by modeling the flow of control or data. It provides a graphical representation of the sequence of activities and the flow of control between them, often used to depict the workflow of a business process or a system function.

KEY ELEMENTS OF ACTIVITY DIAGRAM

Activities:

- Represent specific tasks or actions that need to be performed.
- Depicted as rounded rectangles.
- Example activities for a Document Management System:
  - User Login
  - Search Document
  - Download Document
  - Upload Document
  - View User Logs

Transitions/Flows:

- Represent the flow of control from one activity to another.
- Depicted as arrows.
- Indicate the sequence in which activities are performed.

Start (Initial) Node:

- Indicates the starting point of the workflow.
- Depicted as a filled black circle.

End (Final) Node:

- Indicates the end point of the workflow.
- Depicted as a filled black circle within a larger circle.

Decision Nodes:

- Represent points in the workflow where decisions need to be made, leading to different paths.
- Depicted as diamonds.
- Example decision: Is the user an admin or a regular user?

Fork and Join Nodes:

- Fork node: Splits the flow into multiple concurrent paths.
- Join node: Synchronizes multiple concurrent paths into a single path.
- Depicted as horizontal or vertical bars.

Swimlanes:

- Divide the diagram into sections, each representing different actors or departments responsible for various activities.
- Help in clarifying who performs which activity.

ACTIVITY DIAGRAM DESCRIPTION

Start:

- The initial action where the user opens the Document Management System.

User Login:

- User inputs their username and password.
- Transition: The system validates the credentials.

Role Verification:

- Decision node checks if the user is an admin or a regular user.

Admin Path:

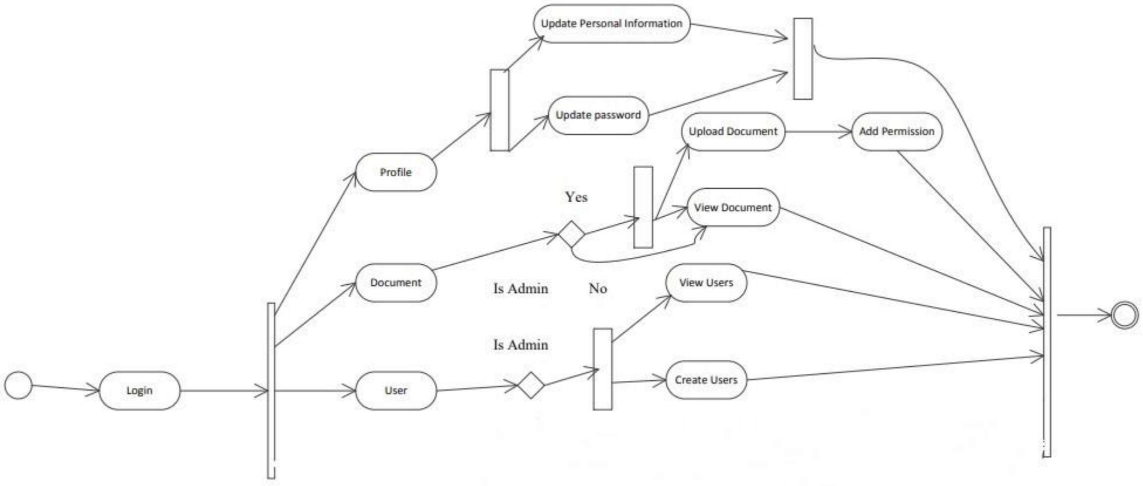
- **Upload Document**: Admin can upload documents.
- **View User Logs**: Admin can view logs of user activities.
- Transition back to role verification or final node for logout.

Regular User Path:

- **Search Document**: User searches for documents.
- **Download Document**: User downloads documents.
- Transition back to role verification or final node for logout.

End:

- The final action where the user logs out and the system ends the session.



Fig(4.4.1) activity diagram

4.5 SEQUENCE DIAGRAM

A Sequence Diagram is a type of UML diagram that represents the interaction between objects in a sequential order. It depicts how objects communicate with each other through messages in a particular sequence, showing the flow of control in a system. Sequence diagrams are particularly useful for detailing the dynamic behavior of a system and for modeling the logic of complex processes.

KEY ELEMENTS OF SEQUENCE DIAGRAM

Actors:

- Represent external entities that interact with the system.
- Depicted as stick figures.
- Example actors for a Document Management System:
    - Admin
    - Regular User

Objects:

- Represent instances of classes or components within the system.
- Depicted as rectangles with underlined names.
- Example objects:
    - User Interface
    - Authentication System
    - Document Repository
    - Log System

Lifelines:

- Represent the existence of an object over a period of time.
- Depicted as vertical dashed lines extending from objects.
- Lifelines are used to indicate the lifespan of an object during the interaction.

Messages:

- Represent communication between objects.
- Depicted as horizontal arrows from one lifeline to another.
- Types of messages:
    - **Synchronous Message**: Indicates a call that waits for a response.
    - **Asynchronous Message**: Indicates a call that does not wait for a response.
    - **Return Message**: Indicates the return of control to the caller object.

Activation Bars:

- Represent the time period an object is performing an action.
- Depicted as thin rectangles on lifelines.

Combined Fragments:

- Represent control structures such as loops, alternatives, and options.
- Depicted as rectangular frames with operators like alt, loop, opt.

SEQUENCE DIAGRAM DESCRIPTION

User Login Sequence:

- **Actors**: Admin, Regular User
- **Objects**: User Interface, Authentication System
- **Sequence**:
  - User enters credentials on the User Interface.
  - User Interface sends credentials to the Authentication System.
  - Authentication System verifies credentials.
  - Authentication System returns authentication result to the User Interface.
  - User Interface displays the result to the user.

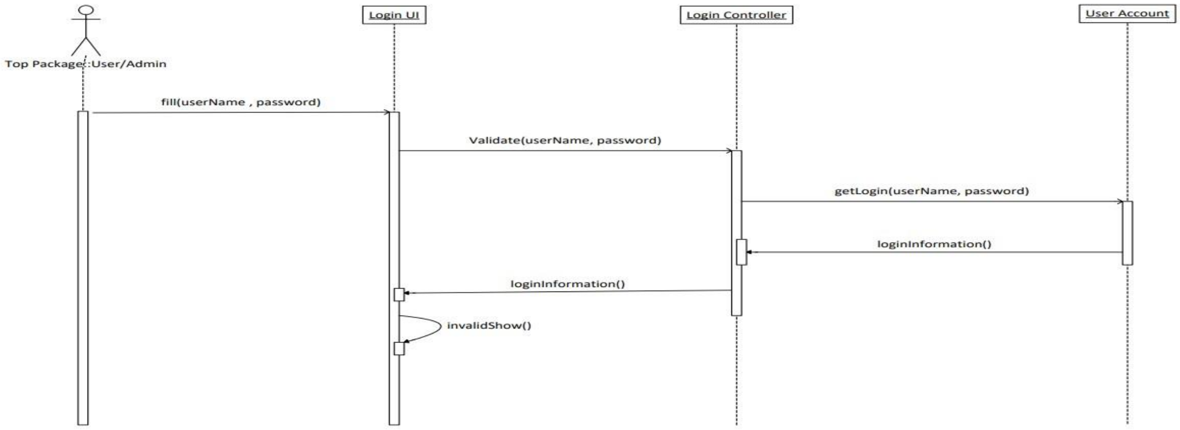
Document Upload Sequence (Admin):

- **Actors**: Admin
- **Objects**: User Interface, Document Repository, Log System
- **Sequence**:
  - Admin selects the upload document option on the User Interface.
  - User Interface sends the document to the Document Repository.
  - Document Repository stores the document.
  - Document Repository returns the result of the upload to the User Interface.
  - User Interface displays the result to the Admin.
  - User Interface sends the upload action to the Log System.
  - Log System records the action.

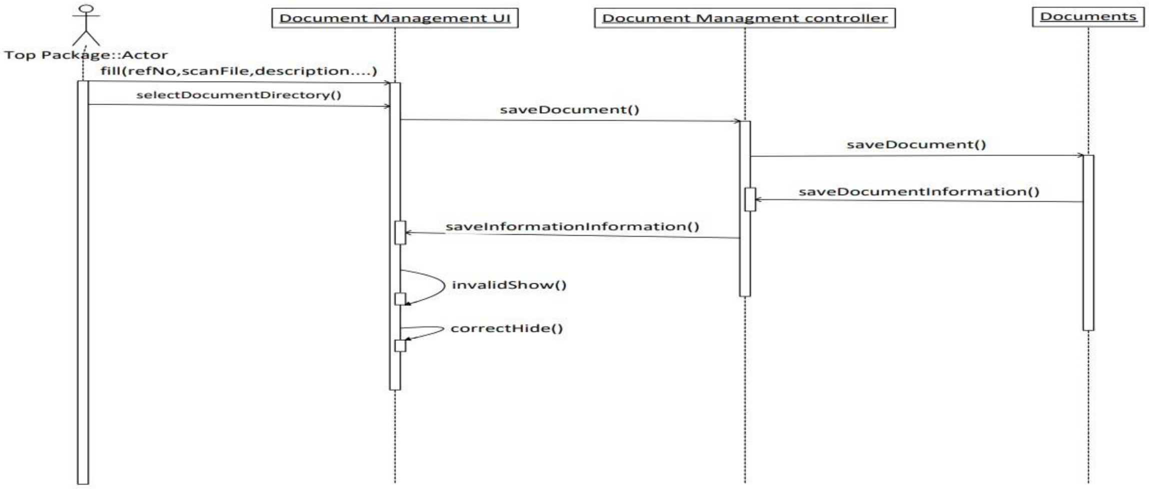
Document Search and Download Sequence (Regular User):

- **Actors**: Regular User
- **Objects**: User Interface, Document Repository, Log System
- **Sequence**:
  - User enters search criteria on the User Interface.
  - User Interface sends the search request to the Document Repository.
  - Document Repository returns search results to the User Interface.
  - User Interface displays search results to the user.
  - User selects a document to download on the User Interface.

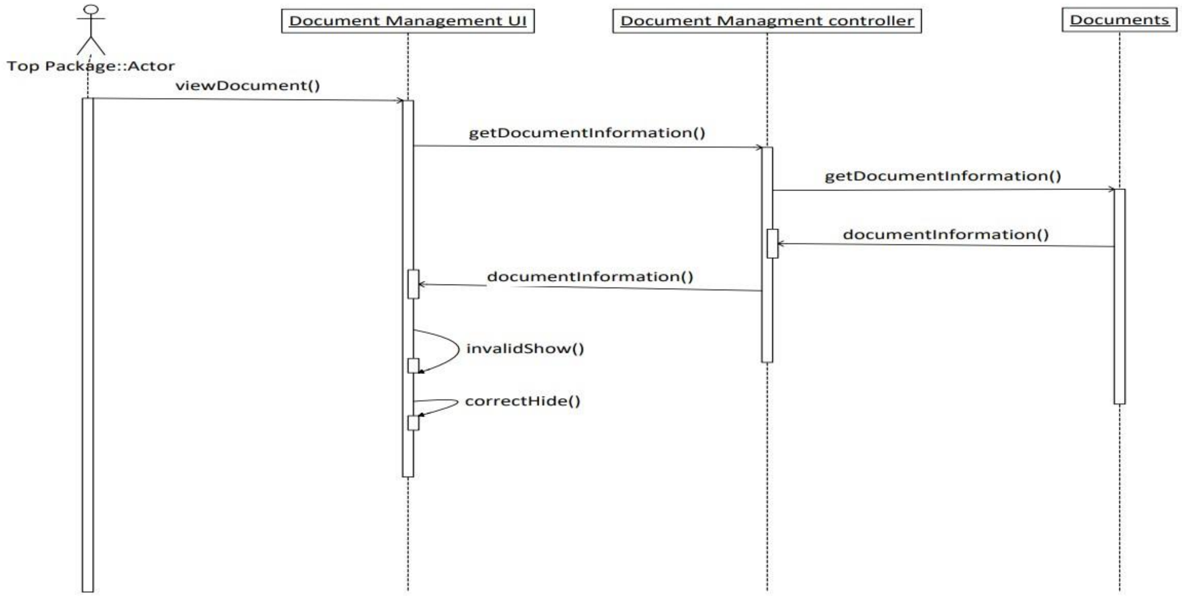
- User Interface sends the download request to the Document Repository.
- Document Repository sends the document to the User Interface.
- User Interface allows the user to download the document.
- User Interface sends the download action to the Log System.
- Log System records the action.



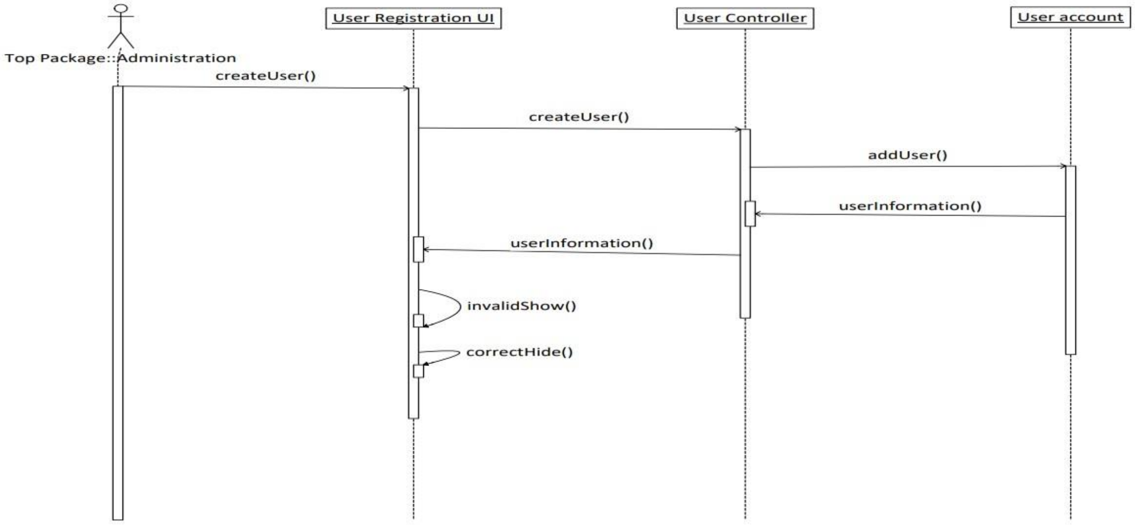
Fig(4.5.1) login sequence diagram


Top Package::Actor
Document Management UI
Document Managment controller
Documents
fill(refNo,scanFile,description....)
selectDocumentDirectory()
saveDocument()
saveDocument()
saveDocumentInformation()
saveInformationInformation()
invalidShow()
correctHide()

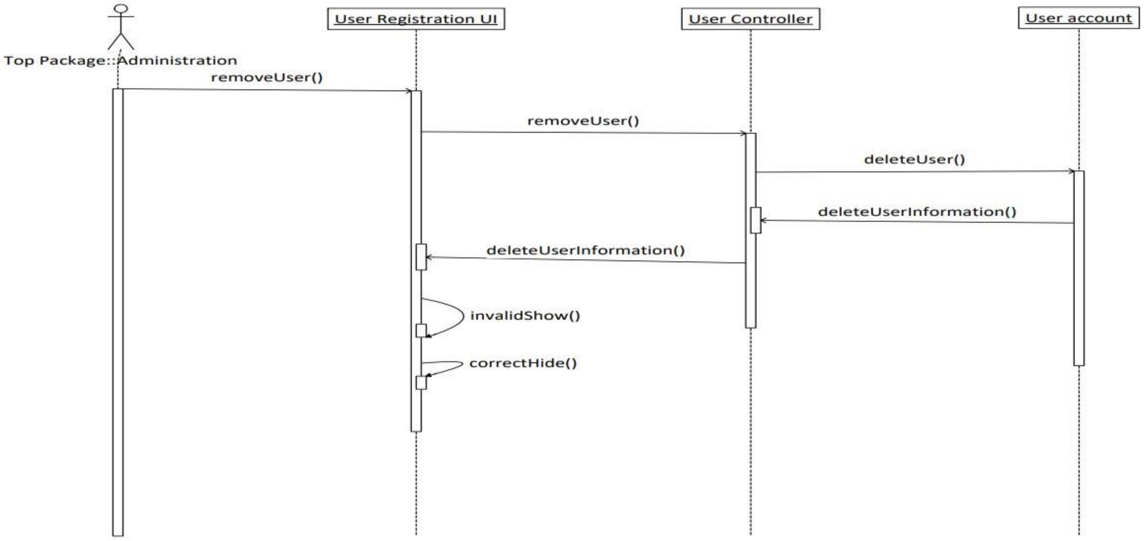
Fig(4.5.2) add document sequence



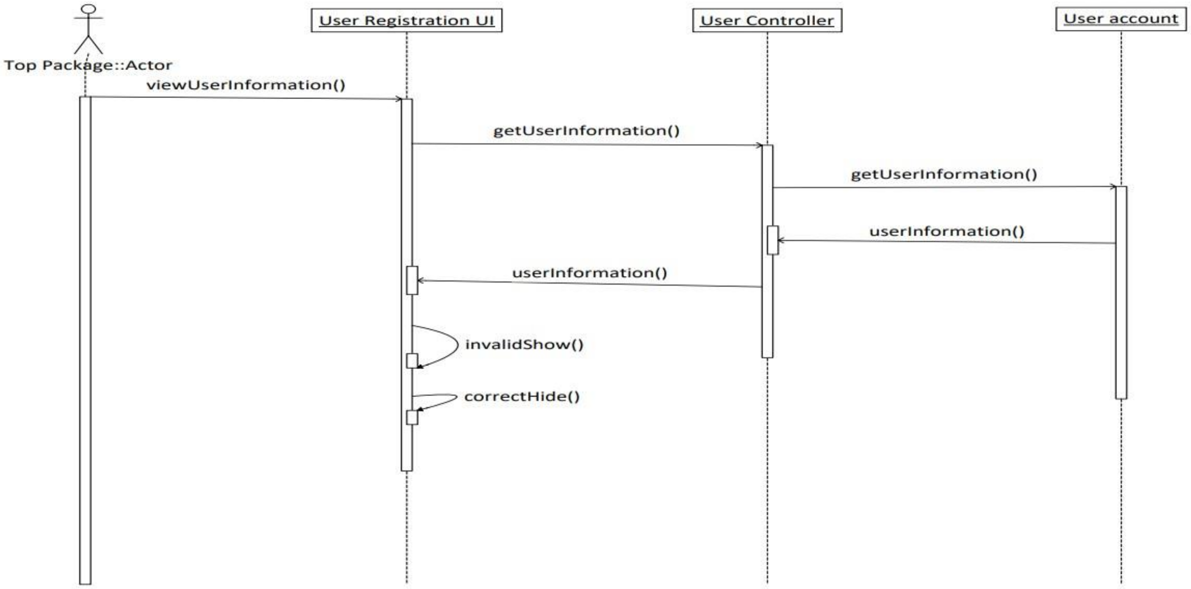
Fig(4.5.3) view document sequence



Fig(4.5.4)add user sequence



Fig(4.5.5) remove user sequence



Fig(4.5.6) view user sequence

5 SYSTEM IMPLEMENTATION AND TESTING

- **SYSTEM IMPLEMENTATION**
- **SYSTEM TESTING**

5.1 SYSTEM IMPLEMENTATION

System implementation is the process of putting a designed system into operation. It involves a series of steps and activities that ensure the system is built, tested, and deployed effectively to meet the requirements and objectives defined in the earlier stages of the system development lifecycle (SDLC). The implementation phase is crucial as it brings the system from the conceptual phase to actual operation, ensuring it performs as intended and provides value to the users.

To implement this web-based system different technology were used. The technology solution includes React js, PHP, HTML with MySQL as the back-end integration database. The choice of these programming languages is based on the features of the languages that make them very appropriate for this Project. HTML, CSS and React js is used (for the front-end interface) and MYSQL (for the backend) and served through a web server, APACHE. The use of HTML and CSS, which is a markup language for information presentation and a styling language respectively, allow for the user-interface to be designed and properly laid out.

To enable dynamic content generation, PHP (a web scripting language) is used to generate dynamic contents based on the user of the system and the corresponding content stored in the backend database which is managed by MySQL. The web server is used to serve the webpages to users when they are needed, and also to interpret the PHP scripting commands contain in the page. Bootstrap is another collection of tools used for creating web application for this project. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components as well as optional JavaScript extensions.

The reason for choosing to use this tool is:

- It is compatible with all major browsers.
- It supports responsive design: Which means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used by personal computer, mobile phone and tablets.

5.2 SYSTEM TESTING

System testing is a techniques used to perform to evaluate the complete system against specific requirements and it is an important and critical stage in software development. Testing plays an important role in the determining the quality and reliability of the application. Different testing methods and tools were used to ensure the functionality and usability of document management. The primary purpose of these tests is to uncover the systems limitations and measure its full capabilities.

Following are testing tools used for all testing types:

5.2.1 Manual test:

Manual Testing is a type of software testing where testers manually execute test cases without using any automation tools. Manual testing is selected because it is the most primitive of all testing types and helps find bugs in the newly implemented system.

Testing were made with different employees found in different offices of the University, such as Executive vice president office, Archive office, store and HRM office. These office users are the one who involved in the requirement gathering and repeatedly interact with. The key concept of having manual testing with those office users help; ensure that the implemented system is error free and it is working to the specified functional requirements.

The process of testing is implemented with repetitive discussion between the system developer and system user.

5.2.2 UNIT TESTING

This test is conducted to check whether the internal logic is functioning properly and program inputs produce valid outputs that compare with the expected results. It is done after the completion of an individual unit. These tests are performing at component level and specific business process, application, and system configuration. In this testing module interface is tested to assure that information properly and correctly flows into and out of the module. This testing involves the testing of data truncation, the structure of the data, and whether the program correctly accepts the input data. The whole validation of the program is encountered in this testing.

Unit testing has been implemented and successfully tested by the system developer, until it reaches a point where a set of methods are ready for the system user. For instance, the system developer is testing a web application.

Following are the various dimensions from which it needs to be tested, each component being a single 'unit' or 'entity'.

- The first page appears on the very first call to a webpage.
- The system notifies the user if he tries to login with only user id or only password.
- System text fields and buttons are aligned properly.
- Links for ‘previous’ and ‘next’ pages works fine the system.
- The system pages navigate according to a proper sequence.
- The search button works for searching any content in the page.
- The CSS styles work as per its design specifications.

5.2.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentations of the document management system. The goal of this testing is to verify the functionality of the proposed system.

Following are the various dimensions from which it needs to be tested, in functionality testing:

- All the links of webpages are working correctly and successful redirect to another page
- All Forms are working as expected and if user doesn’t feel a mandatory field in form an error message is shown.
- All URL pages of the applications are displayed and working fine.
- Login functionality is working with valid inputs.
- Login functionality with invalid inputs displayed error message.
- All data manipulation is working like: Delete/Edit operations
- All elements or objects of the web is working like: Buttons
- Is the system allowing the system user to search documents according to their need?
- Does the system allow to destroy or move records based on their retention policy?

5.2.4 SECURITY TESTING

Security testing is a crucial phase in the software development lifecycle aimed at identifying vulnerabilities and ensuring that the system's data and resources are protected against potential threats. This type of testing involves evaluating the system's ability to withstand malicious attacks and unauthorized access by simulating various attack scenarios. The primary objectives of security testing are to identify security weaknesses, ensure data integrity, confidentiality, and availability, and verify that the system complies with relevant security standards and regulations.

Security testing encompasses several techniques, including penetration testing, vulnerability scanning, security code reviews, and risk assessments. Penetration testing involves ethical hackers attempting to exploit vulnerabilities to gain unauthorized access, thereby identifying weaknesses before they can be exploited by malicious actors. Vulnerability scanning uses automated tools to detect known security flaws in the system. Security code reviews involve a thorough examination of the source code to identify potential security issues such as SQL injection, cross-site scripting (XSS), and buffer overflows. By addressing these security concerns proactively, organizations can ensure robust protection of their systems and maintain user trust.

Following are the various dimensions from which it needs to be tested, in security testing:

- Does the system check access privileges and validated against authorized users
- Is the system Verify the “View Source code” option is disabled and should not be visible to the user
- If functionality is not working? the system displays an error page, instead of displaying any application, server, or database information.
- Does the system Verify the important information like password, and display in encrypted format.
- Is the system verifying passwords in all authentication pages?

5.2.5 USABILITY TESTING

Usability testing is one of the most used methods to define the level of usability of a software product. It is intended to determine the extent an interface facilitates a user’s ability to complete routine tasks. Users are asked to complete a series of routine tasks. Sessions are recorded and analyzed to identify potential areas for improvement to the web-based system.

Following are the various dimensions from which it needs to be tested, in usability testing:

- Is the interface of the system user friendly?
- Is the system attractive (regarding the font size and color combination) for use?
- Is the system easy to understand and to be used?
- Does the elements or objects of the web is easily visible?

Testing result is extracted from the manual testing made with different system testing participants. Repetitive interaction where made with each participants and finally, they give their last response using the checklist found in each testing types.

6 CONCLUSION AND RECOMMENDATIONS

- **CONCLUSION**
- **RECOMMENDATIONS**
- **REFERANCES**

6.1 CONCLUSION

The Document Management System (DMS) project, developed using PHP, MySQL, and ReactJS, is a comprehensive solution designed to streamline document handling and enhance operational efficiency. By centralizing document storage, providing robust search capabilities, and ensuring secure access control, this system addresses the critical needs of both administrators and regular users. Administrators have the exclusive ability to upload documents and manage user logs, ensuring data integrity and security. Regular users can efficiently search for and download necessary documents, thereby improving their productivity.

Throughout the implementation of this DMS, significant emphasis was placed on ensuring security, user-friendliness, and scalability. Rigorous security testing was conducted to protect against potential threats and ensure data confidentiality, integrity, and availability. The intuitive user interface, developed with ReactJS, ensures a seamless and responsive user experience, facilitating quick adoption by users.

The system's current scope provides a robust foundation for future enhancements, including potential features such as advanced search capabilities, integration with other enterprise systems, and the addition of automated document classification and tagging. This project not only meets the immediate needs of document management but also positions the organization for future growth and technological advancements.

In conclusion, the successful development and deployment of this Document Management System demonstrate a significant step forward in efficient document handling and user management. The project's careful planning, rigorous testing, and attention to user needs ensure that the system will serve as a reliable tool for the organization, enhancing productivity and ensuring data security. With the potential for future enhancements, this DMS will continue to evolve, meeting the changing needs of the organization and its users.

6.2 RECOMMENDATIONS

Based on the developed DMS the following recommendations are made for future works.

- Mobile based document management system integrated with Mobile Cloud Computing. Mobile cloud computing is an infrastructure where both The data storage and data processing happens outside of the mobile device. The recommended mobile document management system moves the computing and data storage away from the mobile devices into powerful & centralized computing platforms located in clouds. Then all the records accessed over the wireless connection.
- Multi-lingual – The document capturing and the entire system support Multiple language.
- Thesis and project portal system to need to be integrated with DMS.
- Office 365 is one of a cloud service offered by Microsoft that includes access to most office applications. and additional services like Lync web conferencing, exchange online hosted email for business, skype world minutes for home and additional online storage including OneDrive. The recommendation of this system is to link it with this cloud service. Doing so will help the system to include and use plenty of services as one. There for it can enable advanced document sharing and collaboration environment.

6.3 REFERANCES

Anderson, C. (2012, December 7). Is document control really that important? Bizmanuals OnPolicy. Retrieved September 23, 2012, from http://www.onpolicy.com/2010-12/isdocument-control-really-that-important.html.

Askwall, S. (1985, April). Computer supported reading vs reading text on paper: A comparison of two reading situations. International Journal of Man-Machine Studies, 22, 425–439.

Noyes,M. J., Garland,J.K. (2008, September). Computer- vs. Paper-Based Tasks: Are They Equivalent?. www.princeton.edu, 51,1352–1375 . Retrieved on August 25, 2012, from http://www.princeton.edu/~sswang/Noyesa_Garland_computer_vs_paper.pdf.

Bertolino, A. (2003). Software Testing Research Practice. Retrieved on August 14,2012, from http://www.cis.upenn.edu/~lee/05cis700/papers/Ber03.pdf.